Departamento de Computación, FCEyN, UBA

Procesamiento del Habla

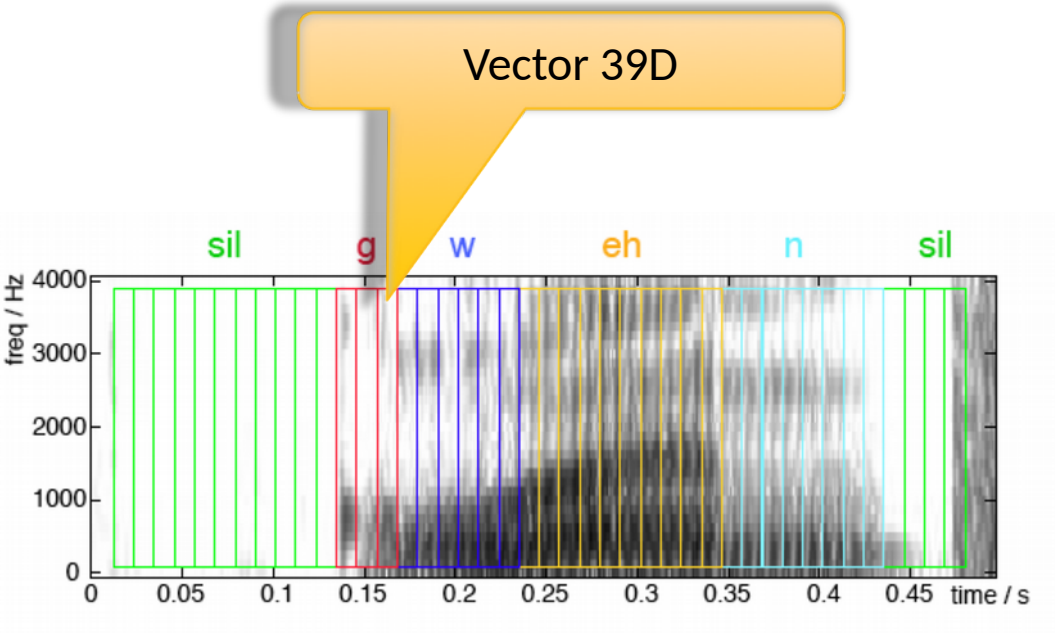
Agustín Gravano

1er Cuatrimestre 2017

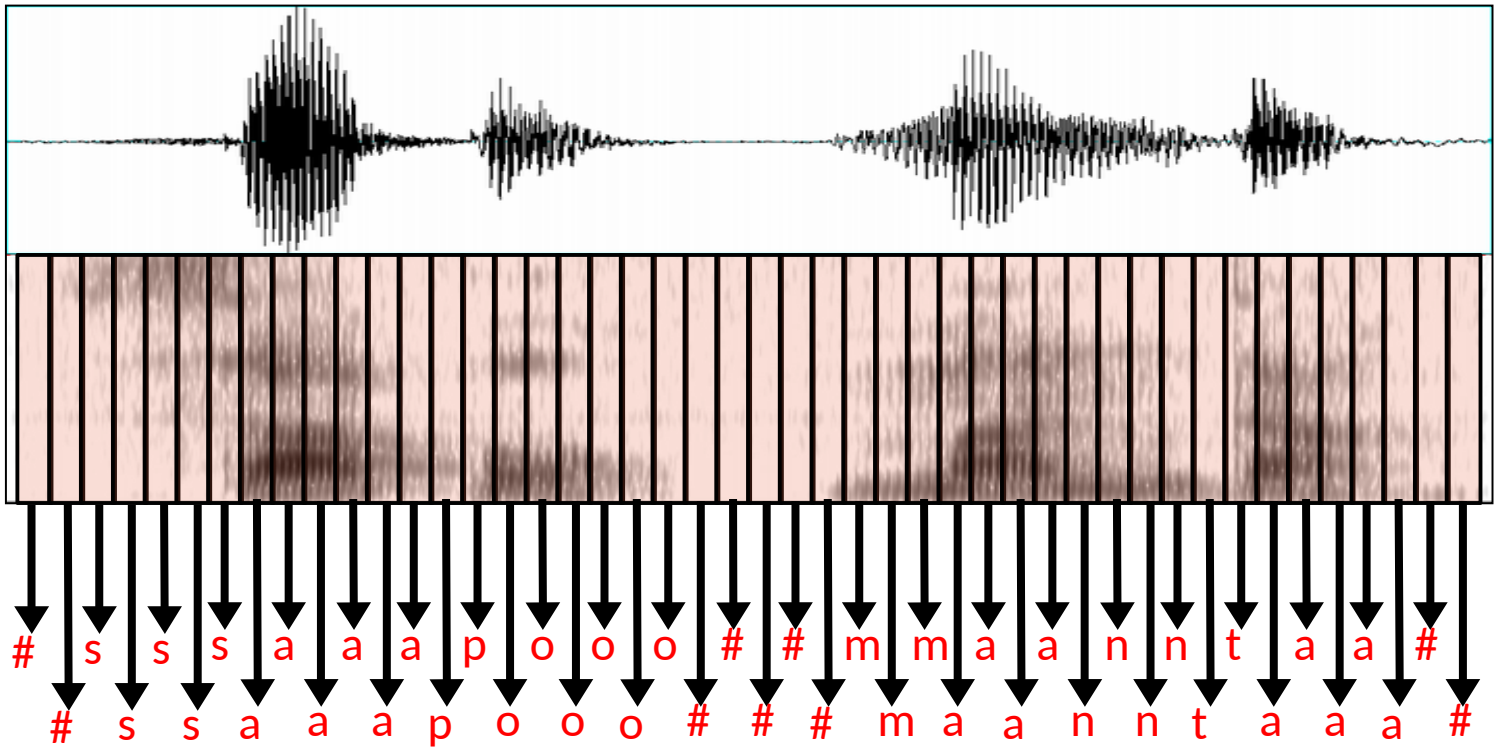
Reconocimiento Automático del Habla (ASR)

Extracción de atributos: MFCC

- Dividir la señal en frames de ~20-30ms, cada ~10-15ms (con superposición no vacía).
- Extraer un vector de atributos de cada frame:
- 12 MFCC + 1 coeficiente de amplitud = 13 atributos
- Delta = 13
- Delta-Delta = 13
- Total: 39 atributos acústicos



Reconocimiento del Habla



Idea: Reconocer el fonema más probable de cada frame, según sus MFCC.

¿Es una buena idea?

- No captura dependencias temporales entre fonemas vecinos (ssssaaeaaaptpooouoo?).
- No considera que hay un léxico (no cualquier secuencia de fonemas es válida) y una sintaxis (hay secuencias de palabras más probables que otras).
- No resuelve cómo pasar de una secuencia de fonemas a una secuencia de palabras.

Reconocimiento del Habla

Formulación probabilística del problema:

Dada una secuencia de observaciones $O = o_1, o_2 \ldots, o_T$, encontrar la secuencia de palabras más probable $\hat{W} = w_1, w_2 \ldots, w_K$.

$$\hat{W} = \underset{W}{\operatorname{argmax}}\ P(W|O)$$

Usando el Teorema de Bayes dos veces, llegamos a:

Modelo acústico

Modelo del lenguaje

$$\hat{W} = \underset{W}{\operatorname{argmax}}\ \frac{P(O|W) \cdot P(W)}{P(O)}$$

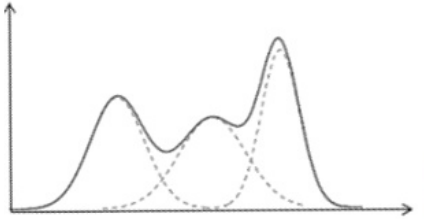
Se puede ignorar porque es independiente de W.

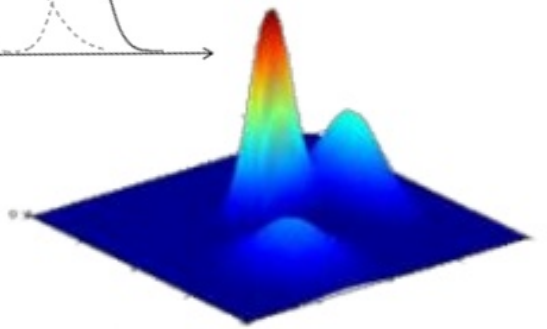
Reconocimiento del Habla

$$\hat{W} = \underset{W}{\operatorname{argmax}} \overbrace{P(O|W)}^{\text{Modelo acústico}} \cdot P(W)$$

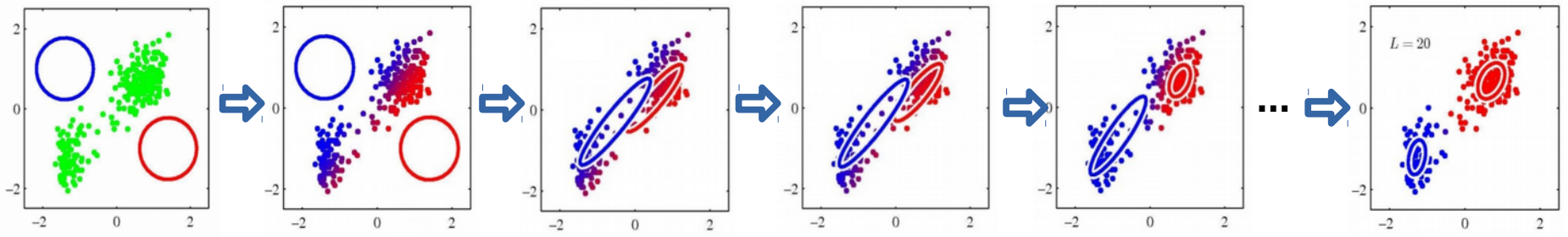
- Modelo acústico: probabilidad de que una secuencia de observaciones *O* haya sido emitida por una secuencia de palabras *W*.
- Hasta el surgimiento de las redes neuronales profundas (por el año 2010): HMM+GMM eran la técnica dominante.

Gaussian Mixture Models (GMM)





- Algoritmo de clustering
- Supone que cada componente sigue una distribución Normal: $\mathcal{N}(\mu, \Sigma)$
- Entrenamiento: método iterativo EM.



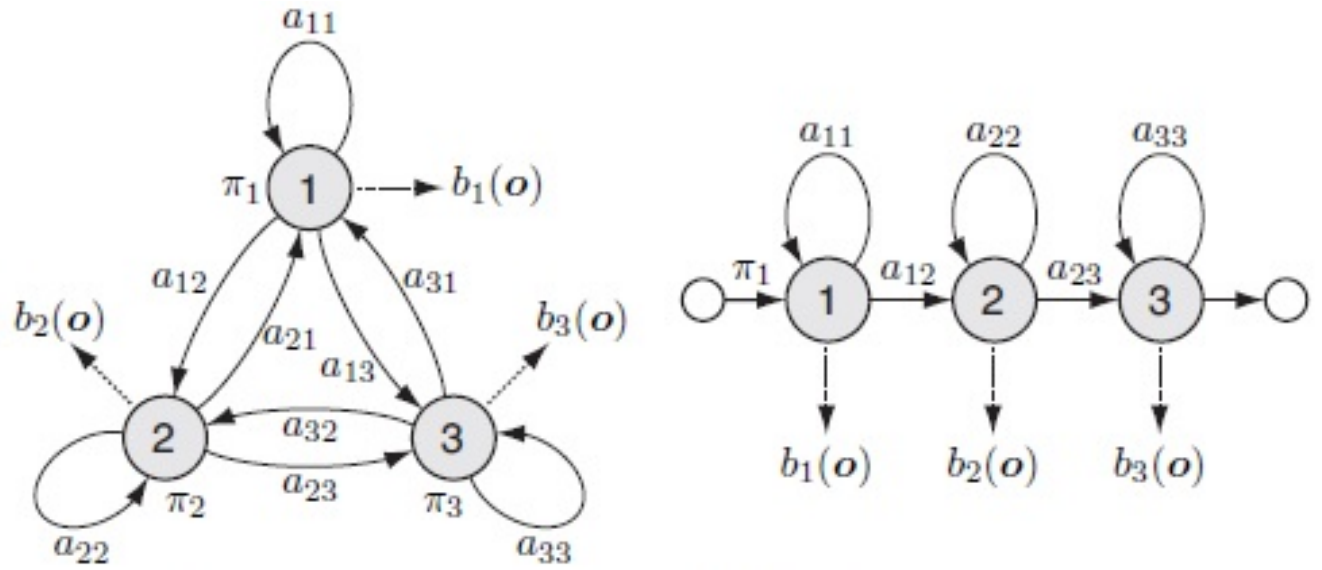
CM Bishop, Pattern Recognition and Machine Learning, Fig 9.1, p426

- Para cada fonema, entrenamos un GMM.
  - Cada componente modela un alófono (ej.: [s] [h] [x] para /s/)
  - La cantidad de componentes se ajusta a los datos.
  - $b_j(o)$: probabilidad de que el fonema *j* emita la observación *o*.

Hidden Markov Models (HMM)

HMM : <*N*, *A*, *B*, *Π*>

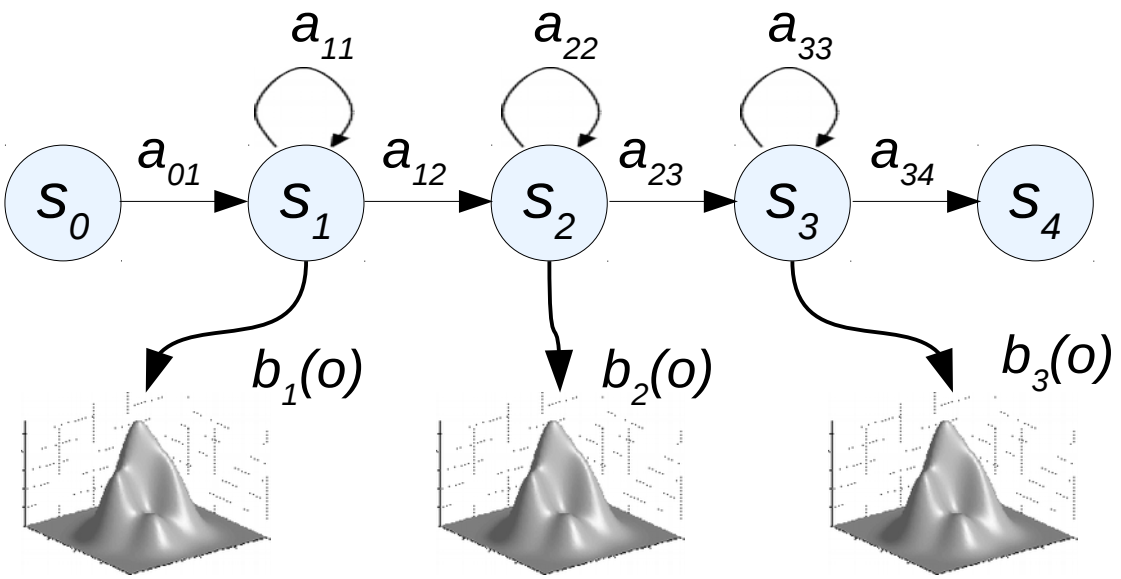
N: cantidad de estados (ocultos)
A=[a_{ij}]: probabilidad de transición del estado *i* al *j*
B=[$b_j(o)$]: probabilidad de emisión de *o* en el estado *j*
Π=[π_i]: probabilidad inicial del estado *i*



(a) A 3-state ergodic model (b) A 3-state left-to-right model

Hidden Markov Models (HMM)

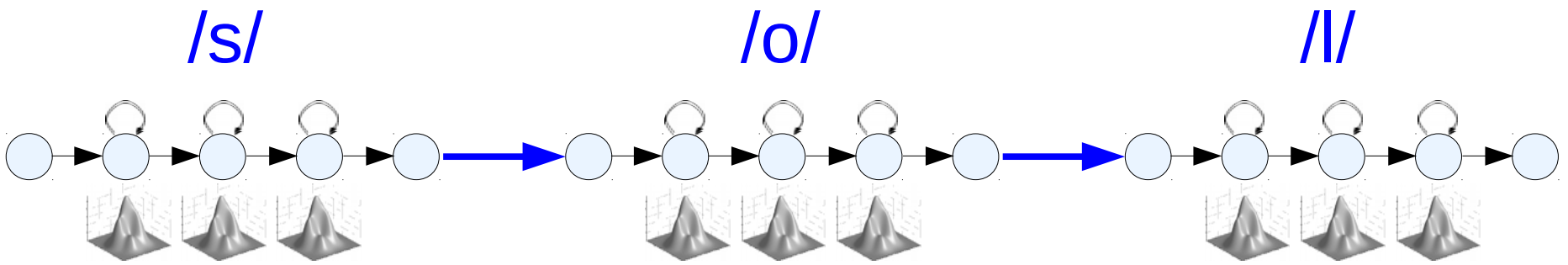
- Un HMM de 5 estados para cada fonema del lenguaje.
  - HMMs “izquierda-a-derecha”.



- Los 3 estados centrales tienen un GMM cada uno, que devuelven la verosimilitud de emitir la observación *o*.

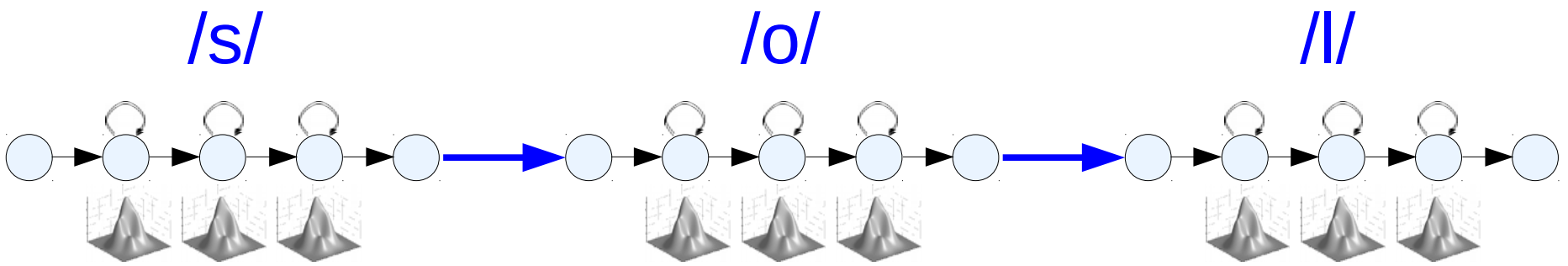
HMM de una palabra

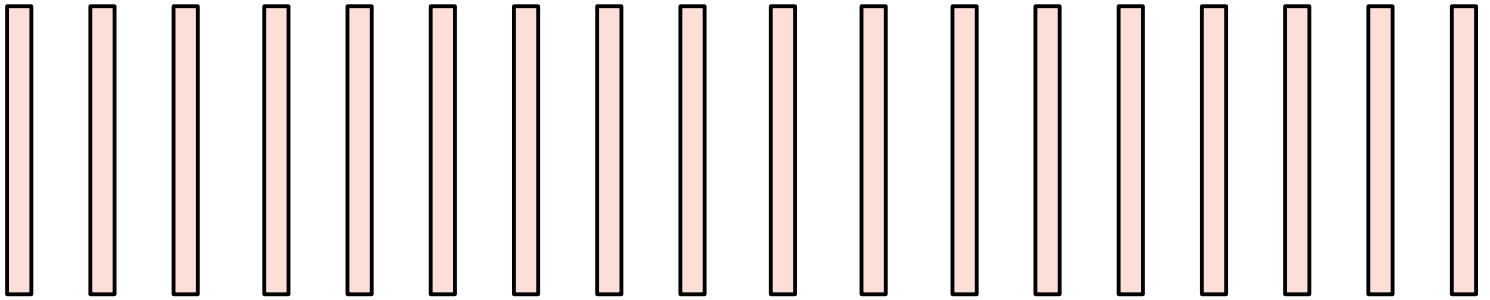
- Para cada palabra del léxico, se construye un HMM concatenando los HMMs de sus fonemas.



HMM de una palabra

- Para cada palabra del léxico, se construye un HMM concatenando los HMMs de sus fonemas.

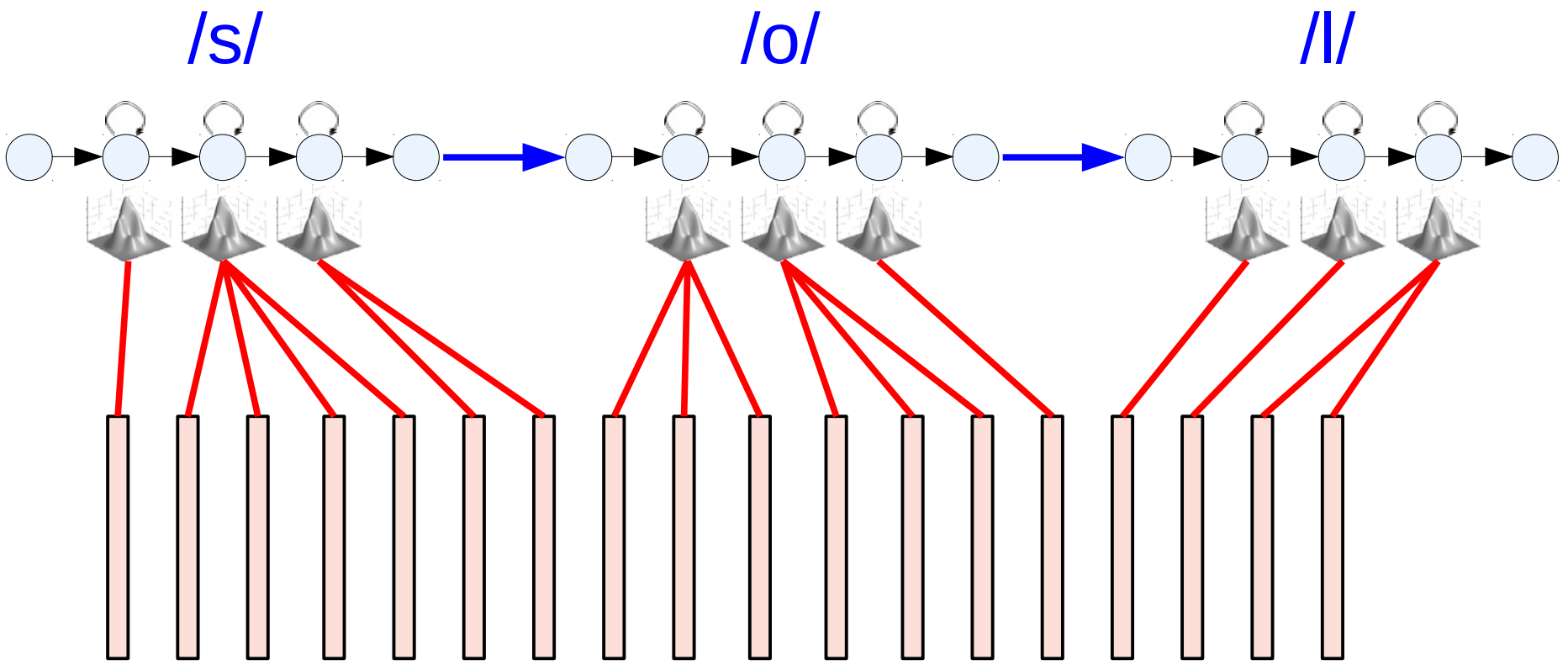




Vectores de atributos acústicos (MFCC): o_1, o_2, o_3, ...

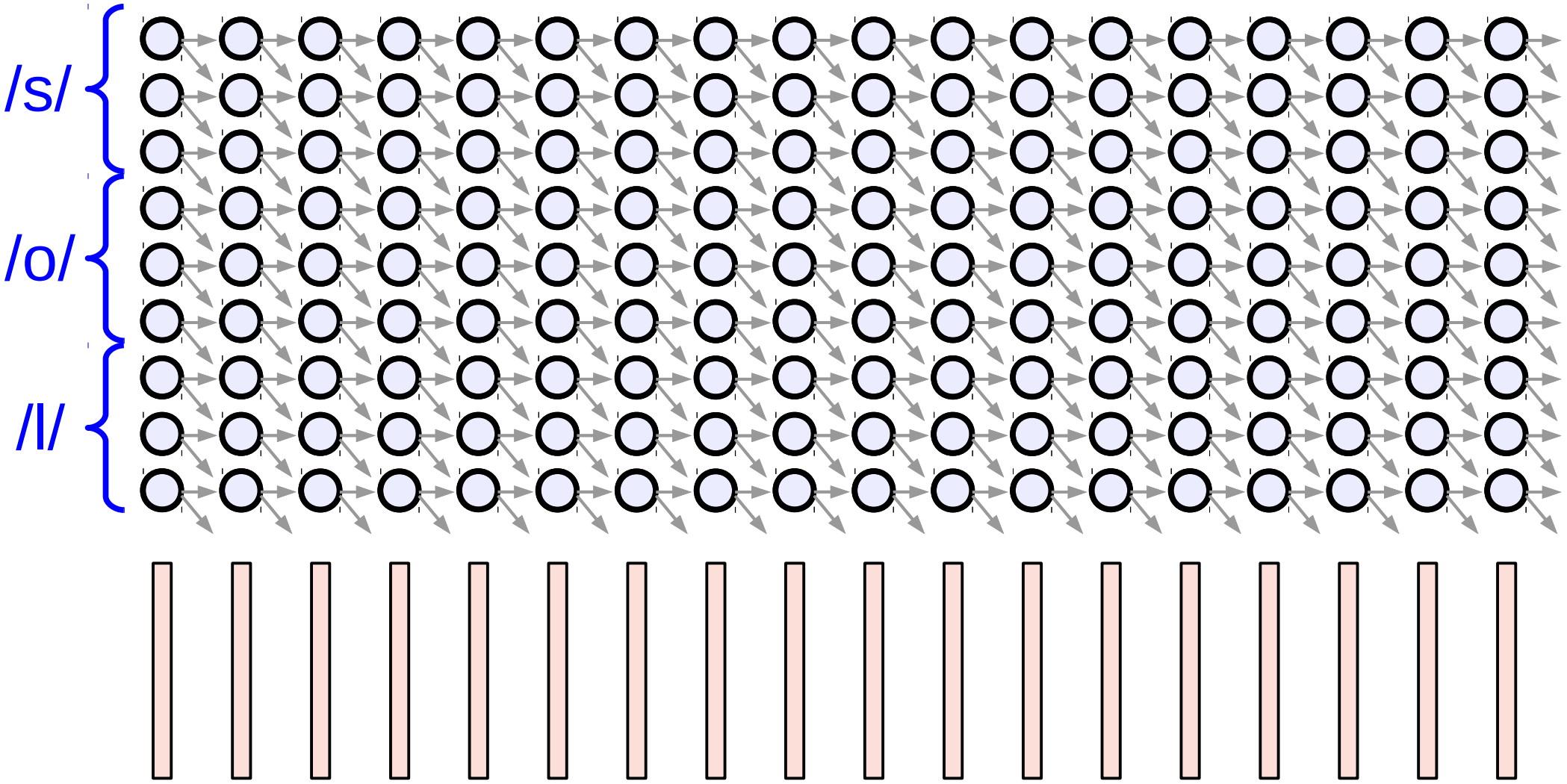
HMM de una palabra

- Para cada palabra del léxico, se construye un HMM concatenando los HMMs de sus fonemas.



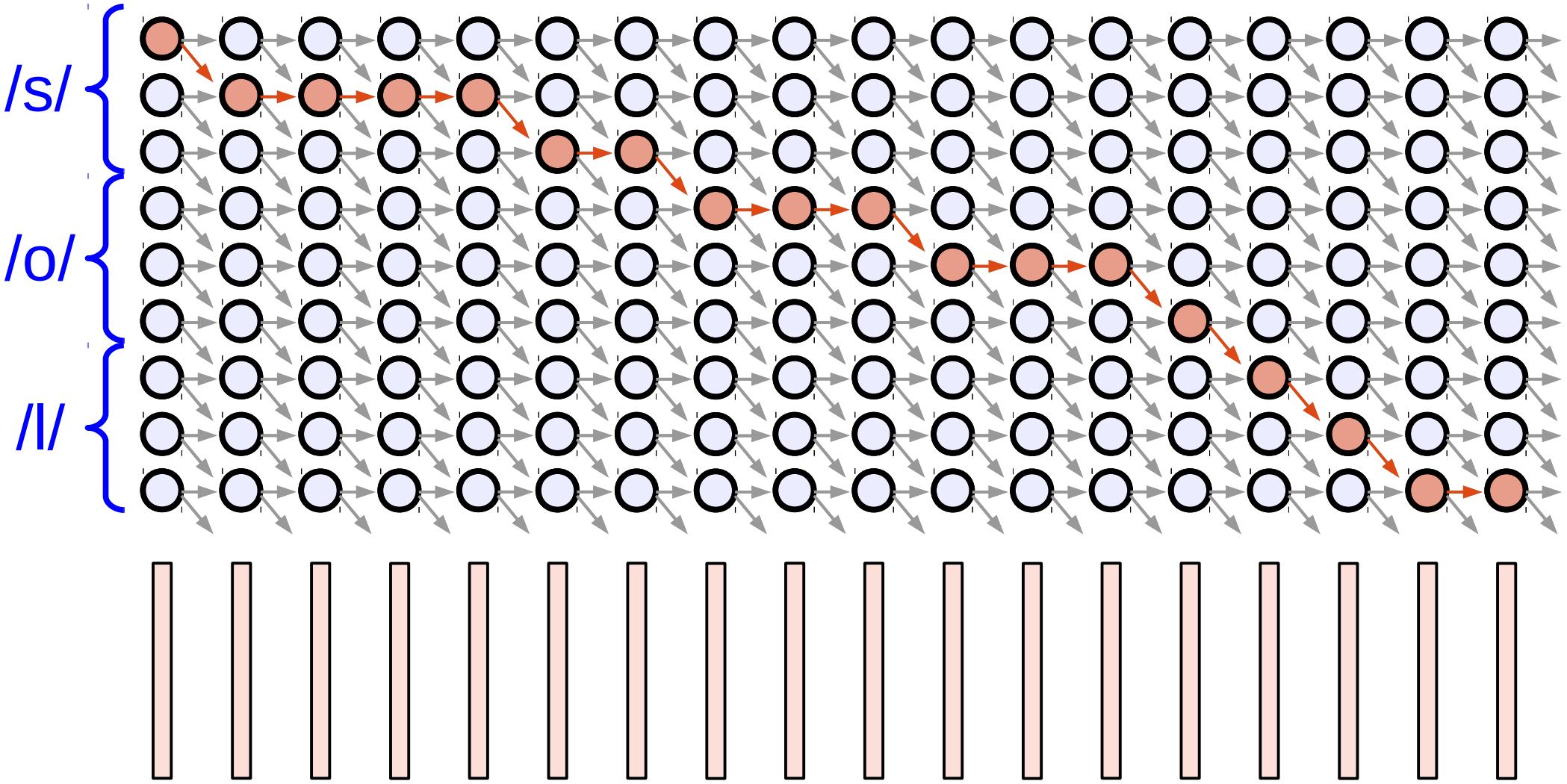
Vectores de atributos acústicos (MFCC): o_1, o_2, o_3, ...

Viterbi en un HMM de una palabra



Vectores de atributos acústicos (MFCC): o_1, o_2, o_3, ...

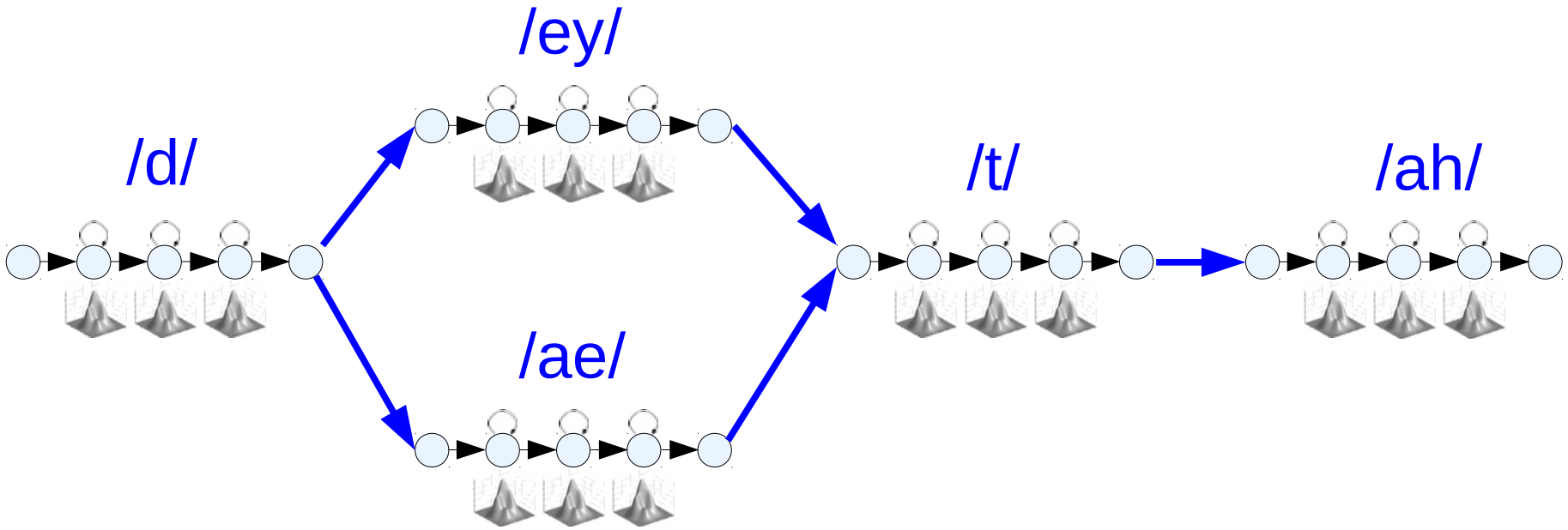
Viterbi en un HMM de una palabra



Vectores de atributos acústicos (MFCC): o_1, o_2, o_3, ...

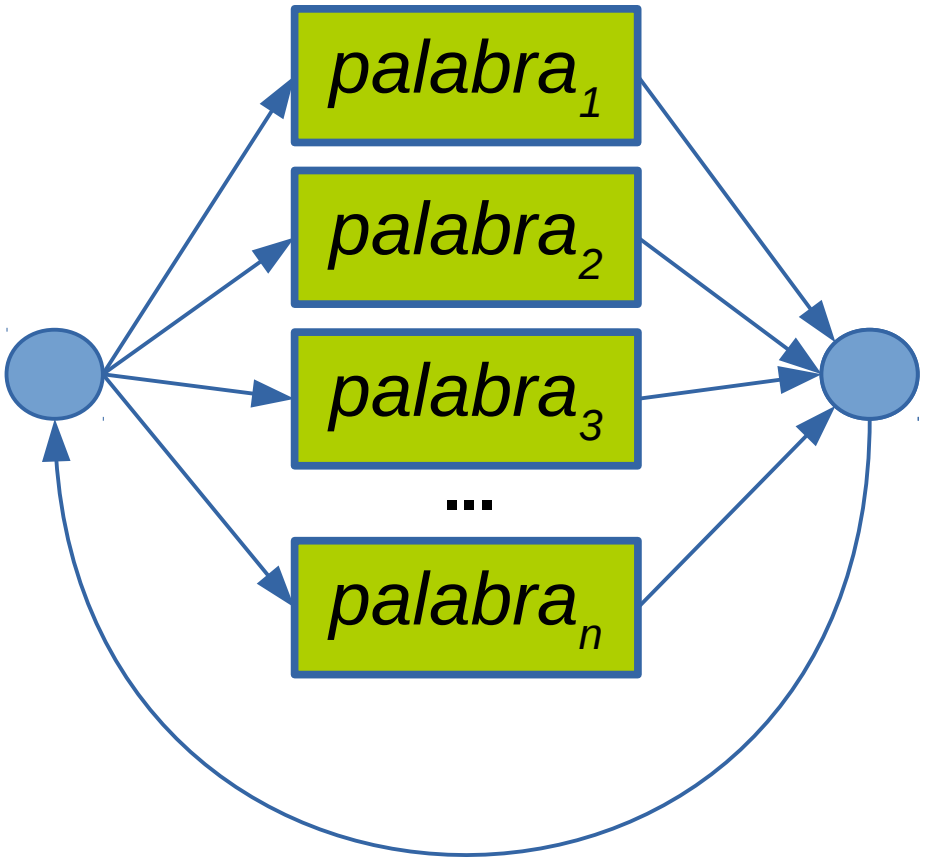
Múltiples pronunciaciones

- Ejemplo (inglés): *data* = /d ey t ah/, /d ae t ah/
- Opción 1: palabras distintas, cada una con su HMM.
- Opción 2: construir HMM con las dos pronunciaciones.



ASR de dominio abierto

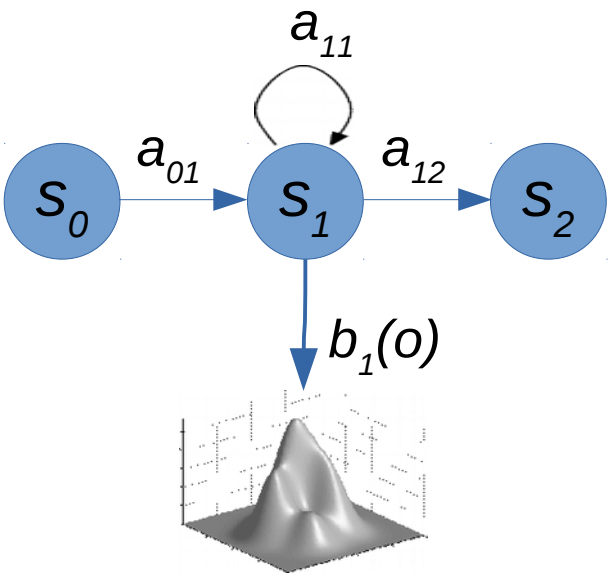
¿Cómo usamos HMMs de palabras para armar un sistema de ASR?



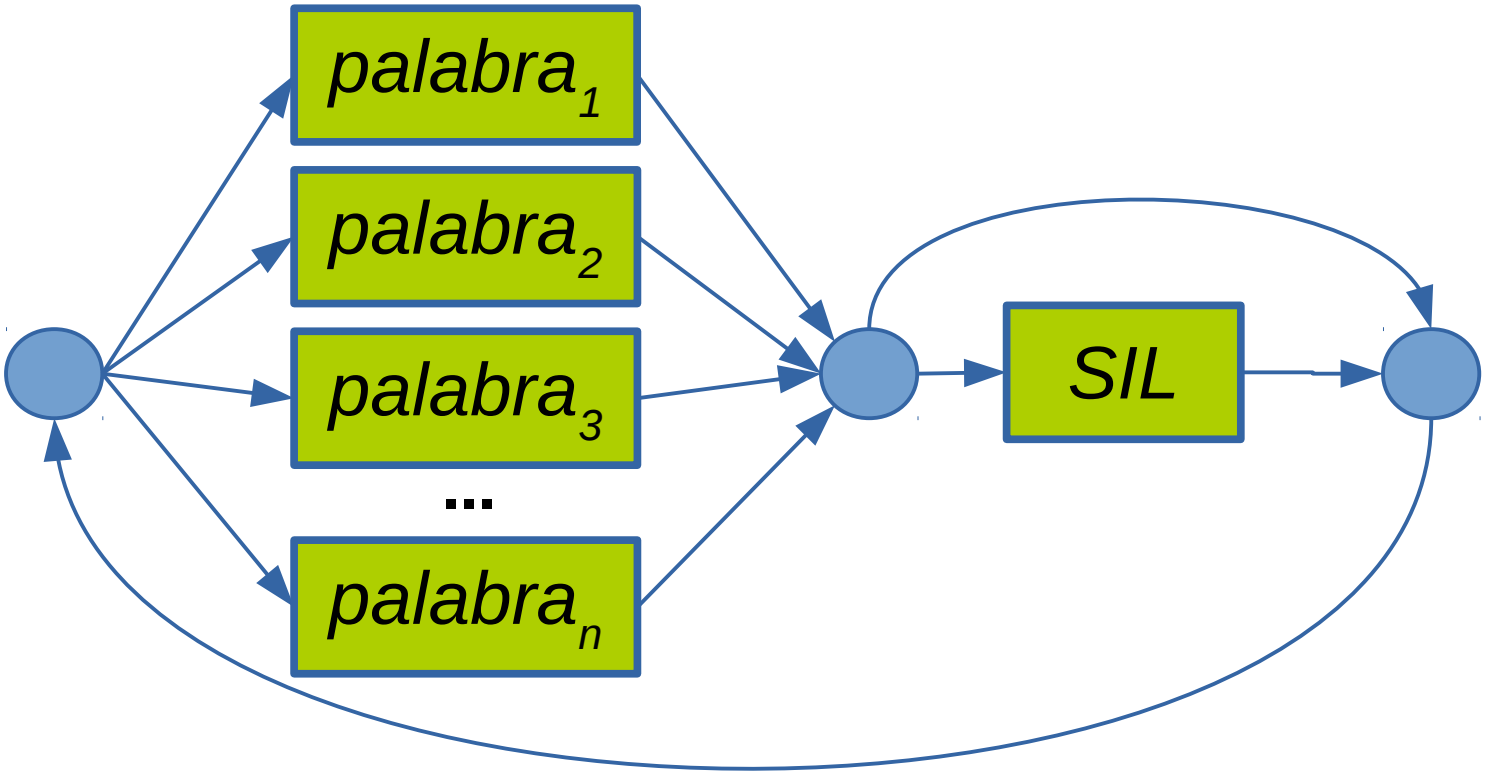
Modelo acústico del silencio

- Para concatenar palabras, debemos permitir que haya silencio entre ellas.
- Creamos un HMM especial para el silencio.

SIL =



ASR de dominio abierto



- Viterbi → Secuencia de palabras más probable, dados el modelo y la secuencia de observaciones (vectores de atributos acústicos).

Entrenamiento de HMM

Ejemplo del Algoritmo Expectation-Maximization (EM)

0) Inicializar modelos acústicos usando un corpus pequeño de grabaciones con alineaciones fonéticas buenas (*"flat start"*).

1) Ajustar datos de entrenamiento (varias horas de grabaciones con transcripciones) al HMM actual.
   - Algoritmo *forward-backward*.

2) Re-estimar los parámetros del HMM: A, B, π.
   - Actualizar las probabilidades de transiciones (A, π) y modelos acústicos (B), para maximizar la probabilidad de los datos de entrenamiento dado el modelo.

3) Ir a 1) y repetir hasta conseguir convergencia.

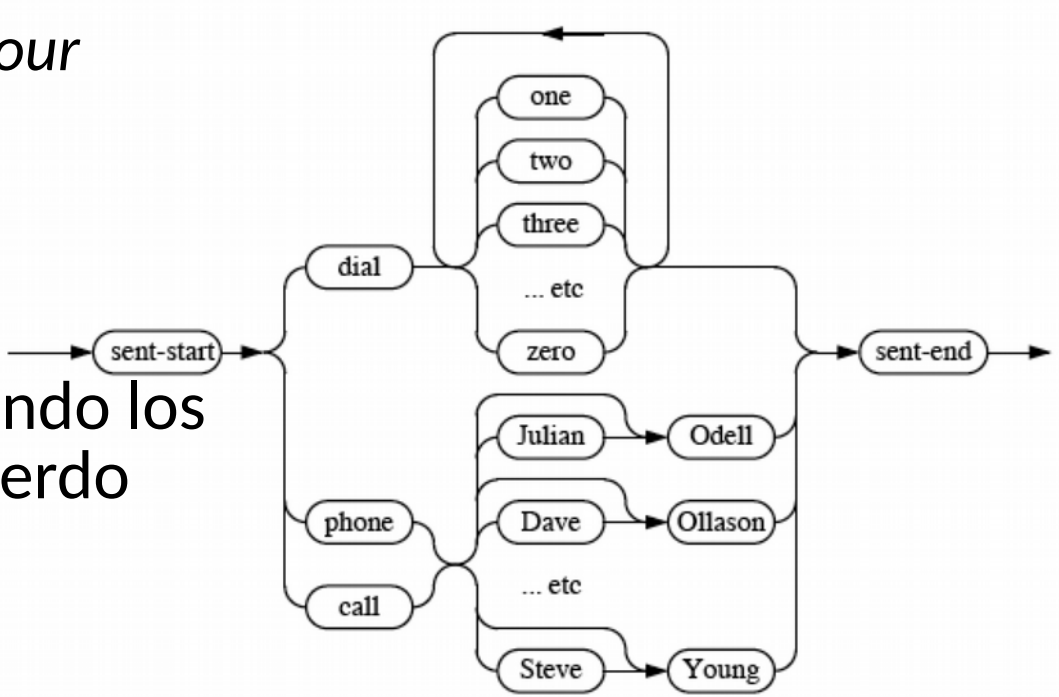
ASR de gramática restringida

- Gramática restringida (p.ej., para un teléfono celular):

```
$digit = ONE | TWO | THREE | ... | ZERO;
$name = [JULIAN] ODELL | [DAVE] OLLASON | ... | [STEVE] YOUNG
( SENT-START ( DIAL ($digit)+ | (PHONE|CALL) $name) SENT-END )
```

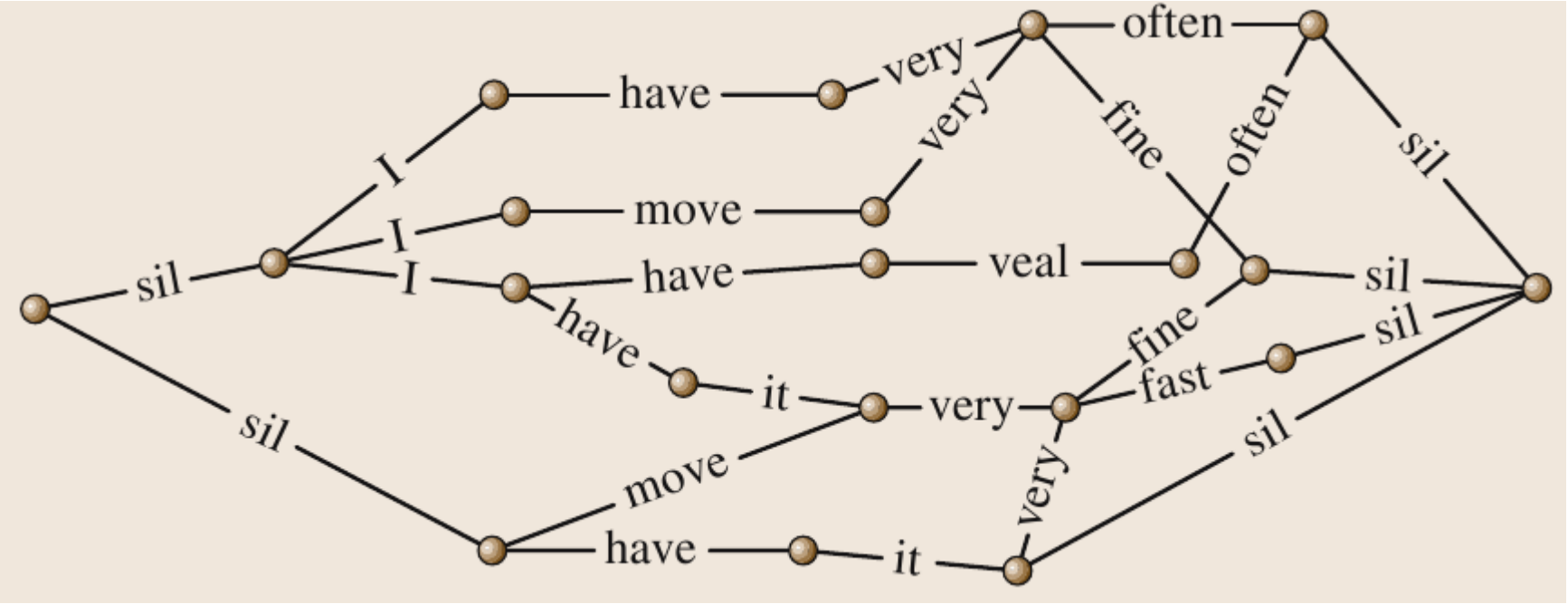
- Ejemplos:
  - *Dial three three two six five four*
  - *Phone Odell*
  - *Call Steve Young*

- Armar un gran HMM combinando los HMMs de las palabras, de acuerdo a la gramática restringida.



Word lattice

- No buscar la mejor solución, sino las *k* mejores.
- Algoritmo: *Multiple-token decoder.*



HMM de Trifonos

- **Problemas**: co-articulación y asimilación.
  - La producción de cada fono es afectada por sus vecinos.
  - Ejemplos: [ala] vs. [ola].
- **Solución:** Usar HMMs de trifonos.
  - Tener, para cada fonema del lenguaje, un HMM por cada posible par de vecinos izquierdo y derecho.
  - Ejemplos: **sil-a+l a-l+a l-a+sil**
    sil-o+l o-l+a l-a+sil
  - N fonemas → $O(N^3)$ potenciales trifonos.
  - Se mapean trifonos lógicos articulatoriamente similares a un conjunto reducido de trifonos físicos.

Reconocimiento del Habla

$$\hat{W} = \underset{W}{\operatorname{argmax}}\ P(O|W) \cdot \overbrace{P(W)}^{\text{Modelo del lenguaje}}$$

- *k* hipótesis generadas por HMM.
- Modelo del lenguaje: Penaliza hipótesis improbables.
- Ejemplo:
  - *el {banco,manco} central anunció la {emisión,emulsión} de nuevas monedas de cincuenta {centauros, centavos}*

Predicción de palabras

Las acciones se ...

Predicción de palabras

Las acciones se derrumbaron esta ...

Predicción de palabras

Las acciones se derrumbaron esta mañana, pese a la baja en las tasas de ...

Predicción de palabras

Las acciones se derrumbaron esta mañana, pese a la baja en las tasas de interés por parte de la Reserva ...

Predicción de palabras

Las acciones se derrumbaron esta mañana, pese a la baja en las tasas de interés por parte de la Reserva Federal, mientras Wall ...

Predicción de palabras

Las acciones se derrumbaron esta mañana, pese a la baja en las tasas de interés por parte de la Reserva Federal, mientras Wall Street volvió a operar por primera ...

Predicción de palabras

Las acciones se derrumbaron esta mañana, pese a la baja en las tasas de interés por parte de la Reserva Federal, mientras Wall Street volvió a operar por primera vez desde los ataques ...

Predicción de palabras

Las acciones se derrumbaron esta mañana, pese a la baja en las tasas de interés por parte de la Reserva Federal, mientras Wall Street volvió a operar por primera vez desde los ataques terroristas del ...

Predicción de palabras

Las acciones se derrumbaron esta mañana, pese a la baja en las tasas de interés por parte de la Reserva Federal, mientras Wall Street volvió a operar por primera vez desde los ataques terroristas del martes pasado.

Predicción de palabras

- En alguna medida, es posible predecir palabras futuras en una oración.
- ¿Cómo hacemos los seres humanos?
  - Conocimiento del dominio.
    - *baja en las tasas de interés*
  - Conocimiento sintáctico.
    - *el <sustantivo>, se <verbo>*
  - Conocimiento léxico.
    - *ataques terroristas, Reserva Federal*

Predicción de palabras

- Parte del conocimiento necesario para predecir las palabras puede ser capturado usando técnicas estadísticas simples.
- En particular, nos interesa la noción de probabilidad de una secuencia de palabras.
- Modelos de *N*-gramas:
  - Usar las *N*–1 palabras anteriores para predecir la siguiente.
  - Unigramas, bigramas, trigramas, …
  - Se entrenan a partir de cuerpos de datos grandes: diarios, libros, Wikipedia, etc.

N-gramas

- ¿Cómo estimamos la probabilidad de una oración?
- Not.: w_1^m es una secuencia de palabras $w_1 \ldots w_m$
- Usando la regla de la cadena, $P(A \wedge B) = P(A \mid B) \cdot P(B)$ tenemos que:

$$P(w_1^m) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1^2) \cdot \ldots \cdot P(w_m|w_1^{m-1})$$

$$= \prod_{k=1}^{m} P(w_k|w_1^{k-1})$$

- Ejemplo:

P("baja en las tasas de interés") =
P("baja") • P("en" | "baja") • P("las" | "baja en") •
P("las" | "baja en las") • ... • P("interés" | "baja en las tasas de")

N-gramas

- Suposición de Markov:
  - La probabilidad de una palabra depende solamente de las N–1 palabras anteriores (*N*-grama).

$$P\left(w_m | w_1^{m-1}\right) \approx P\left(w_m | w_{m-N+1}^{m-1}\right)$$

  - N=2: bigrama $P\left(w_m | w_1^{m-1}\right) \approx P\left(w_m | w_{m-1}\right)$

    P(“interés” | “baja en las tasas de”) ≈ P(“interés” | “de”)

  - N=3: trigrama $P\left(w_m | w_1^{m-1}\right) \approx P\left(w_m | w_{m-2}^{m-1}\right)$

    P(“interés | “baja en las tasas de”) ≈ P(“interés” | “tasas de”)

N-gramas

- Estimar la probabilidad de la oración:
  - *I want to eat Chinese food.*
- $P(I\ want\ to\ eat\ Chinese\ food) =$
  $P(I \mid <start>)\ P(want \mid I)\ P(to \mid want)\ P(eat \mid to)$
  $P(Chinese \mid eat)\ P(food \mid Chinese)\ P(<end> \mid food)$
- ¿Qué necesitamos para estos cálculos?
  - Probabilidad $P(w_m \mid w_{m-1})$ para cada par de palabras.
  - Pre-calculadas de un corpus grande.

Bigramas del BERP Corpus

- BERP (Berkeley Restaurant Project)
  - Consultas de usuarios a un sistema de diálogo hablado.

	I	*want*	*to*	*eat*	*Chinese*	*food*	*lunch*
I	8	1087	0	13	0	0	0
want	3	0	786	0	6	8	6
to	3	0	10	860	3	0	12
eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
food	19	0	17	0	0	0	0
lunch	4	0	0	0	0	1	0

  - $P(want \mid I) = \#(I\ want) / \#(I) = 1087 / 3437 = 0.32$

Smoothing (suavizado)

Todo corpus es limitado. Es inevitable que *N*-gramas válidos (aunque improbables) queden con probabilidad 0 en nuestro modelo.

	I	*Want*	*To*	*Eat*	*Chinese*	*Food*	*lunch*
I	8	1087	0	13	0	0	0
Want	3	0	786	0	6	8	6
To	3	0	10	860	3	0	12
Eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
Food	19	0	17	0	0	0	0
Lunch	4	0	0	0	0	1	0

Técnicas de smoothing: Laplace Smoothing, Good-Turing Discounting

Backoff

- Otro enfoque para evitar el problema de N-gramas con baja frecuencia.

$$P(w_n|w_{n-2}w_{n-1}) = ?$$

- y no tenemos instancias de $w_{n-2}w_{n-1}w_n$

- En vez de concluir *Prob* = 0, hacemos *backoff* a (*N*–1)-gramas:

$$P(w_n|w_{n-2}w_{n-1}) \approx P(w_n|w_{n-1})$$

Resumen de N-gramas

- Es posible capturar las probabilidades de secuencias de palabras mediante técnicas estadísticas simples.
- Suposición de Markov:

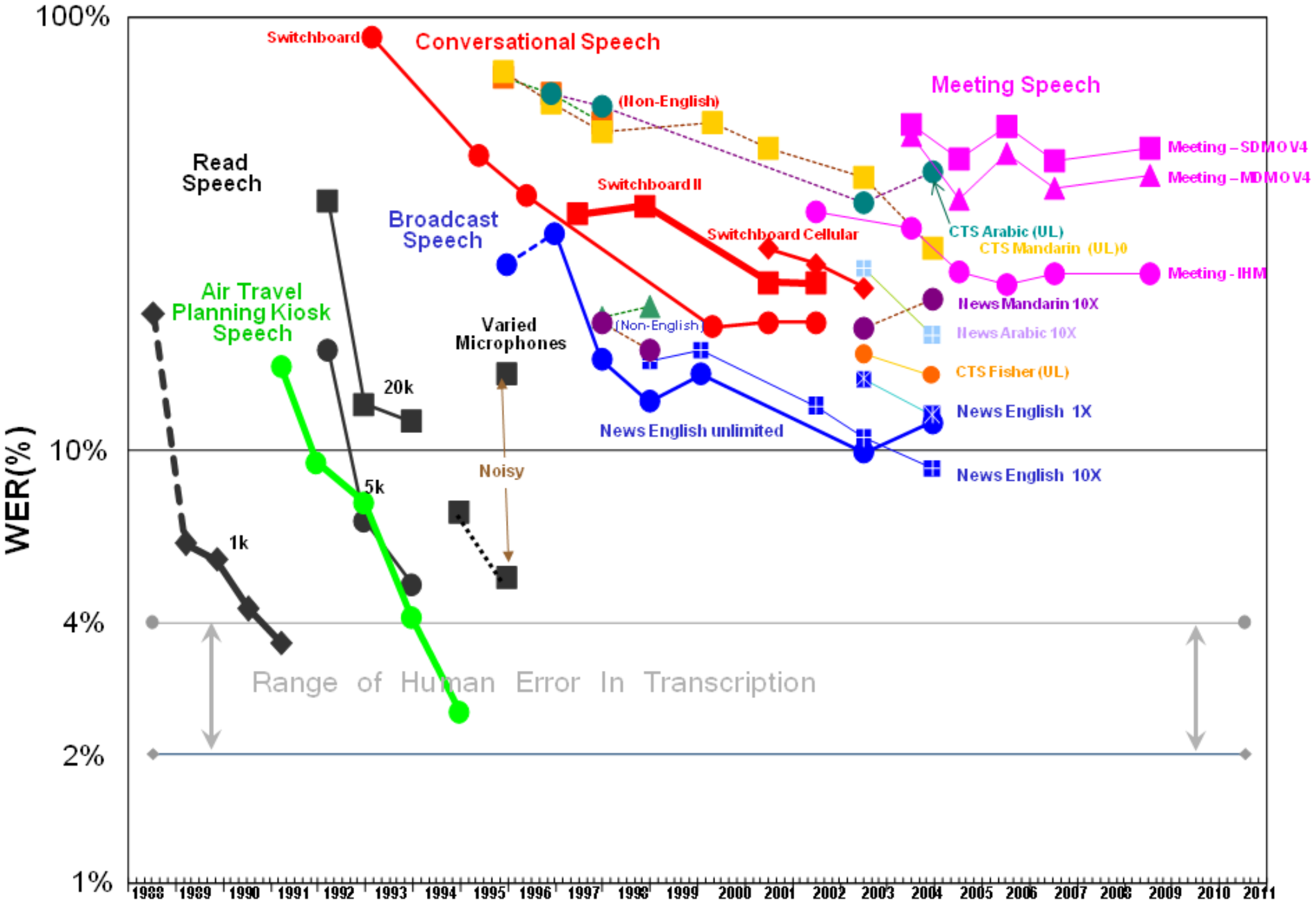
$$P\left(w_m | w_1^{m-1}\right) \approx P\left(w_m | w_{m-N+1}^{m-1}\right)$$

- Técnicas de smoothing y backoff para lidiar con problemas de N-gramas con baja frecuencia.
- Muy usado como modelo del lenguaje en ASR, para combinar la verosimilitud asignada por el modelo acústico:

$$\hat{W} = \underset{W}{\operatorname{argmax}} \overbrace{P(O|W)}^{\text{Modelo acústico}} \cdot \overbrace{P(W)}^{\text{Modelo del lenguaje}}$$

¿Cuán bien funciona el ASR
con técnicas basadas en HMM+GMM?

NIST STT Benchmark Test History – May. '09



Historia de ASR

- Generación 1: 1930s-1940s. Circuitos electrónicos ad-hoc para reconocer palabras aisladas (vocabularios muy reducidos).
- Generación 2: 1950s-1960s. Primeros modelos de lenguaje.
- Generación 3: 1970s-1980s. Dynamic time warping (DTW).
- Generación 4: 1990s-2000s. GMM+HMM.
- Generación 5: 2010s-... **Deep Learning!**

[TABLE 3] A COMPARISON OF THE PERCENTAGE WERs USING DNN-HMMs AND GMM-HMMs ON FIVE DIFFERENT LARGE VOCABULARY TASKS.

TASK	HOURS OF TRAINING DATA	DNN-HMM	GMM-HMM WITH SAME DATA	GMM-HMM WITH MORE DATA
SWITCHBOARD (TEST SET 1)	309	18.5	27.4	18.6 (2,000 H)
SWITCHBOARD (TEST SET 2)	309	16.1	23.6	17.1 (2,000 H)
ENGLISH BROADCAST NEWS	50	17.5	18.8	
BING VOICE SEARCH (SENTENCE ERROR RATES)	24	30.4	36.2	
GOOGLE VOICE INPUT	5,870	12.3		16.0 (>> 5,870 H)
YOUTUBE	1,400	47.6	52.3	

Tomado de Hinton 2012 (Tutorial DNNs & ASR).

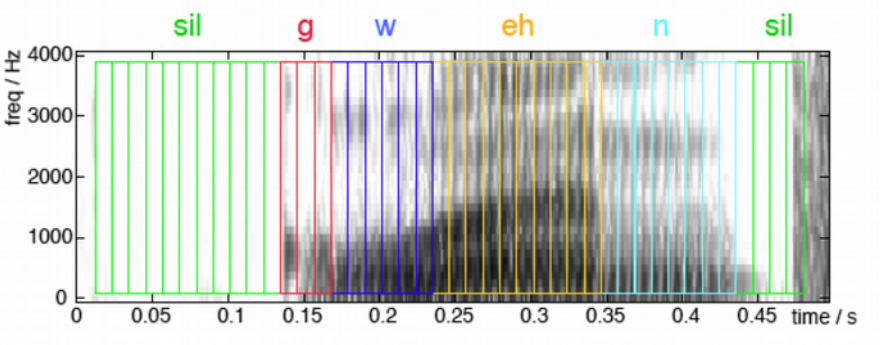
Herramientas para ASR

- **Pocketsphinx**
  - http://cmusphinx.sourceforge.net/wiki/tutorialpocketsphinx
  - Modelos acústicos para el español:

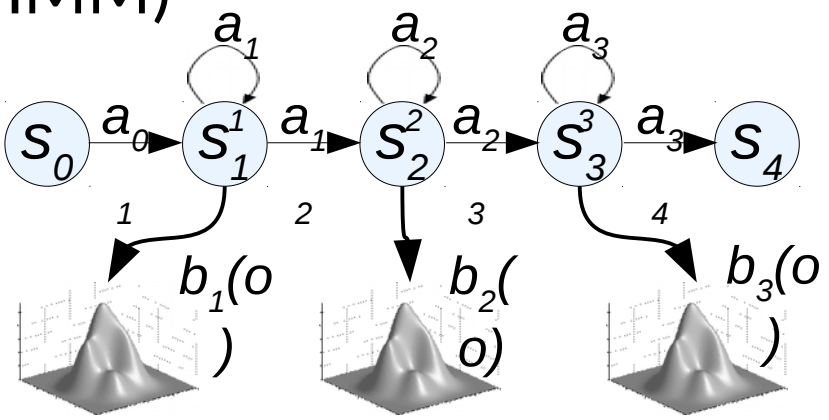
    http://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/Spanish%20Voxforge/
- **HTK** (Hidden Markov Model Toolkit)
  - http://htk.eng.cam.ac.uk/
- **Kaldi**
  - http://kaldi.sourceforge.net/
- APIs para servicios comerciales de ASR:
  - Google, IBM, Microsoft, ...

Reconocimiento del Habla – Resumen

- Atributos: MFCC



- Hidden Markov Models (HMM)



- Gaussian Mixture Models (GMM)
- Modelo del lenguaje, *N*-gramas.
- Presente y futuro: Deep Neural Networks.

Rabiner 1989

Rabiner, Lawrence. "*A tutorial on hidden Markov models and selected applications in speech recognition."* Proceedings of the IEEE 77.2 (1989): 257-286.