

# Indexación de audio: Reconocimiento de Voz

**< audias >**

Audio, Data Intelligence and Speech

<http://audias.ii.uam.es>

Daniel Ramos Castro

Contrib. de Doroteo Torre, Joaquín González, Alicia Lozano

0

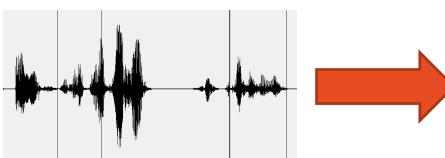
## Contenidos

- **Audio y Voz en Big Data**
- **Speech-To-Text (STT) como usuario**
- **Search-on-Speech como usuario**
- **Comprendiendo el Speech-To-Text**
  - Funcionamiento interno
- **Comprendiendo el Search-on-Speech**
  - Funcionamiento interno
- **Conclusiones**

## Audio y Voz en Big Data

- **Información multimedia (vídeo/audio) cada vez más presente:**
  - Comunicaciones telefónicas (fijas, móviles, call-centers)
  - Medios de comunicación (Radio, TV convencionales)
  - Internet (YouTube, Radio y TV por Internet, VoIP)
  - Comunicación con dispositivos (Siri, Android, Cortana, ...)
- **Información de audio y voz:**
  - Muy desestructurada
  - Muy redundante
  - Muy rica

## Audio y Voz en Big Data

- A partir de un contenido de audio es posible extraer muchísima información:
 
  - Tipo de audio: música/voz/ruidos/silencio
  - Eventos acústicos: ladrido, risa,...
  - Canciones conocidas
  - Idioma y dialecto
  - Quién habla en qué momento
  - Locutores conocidos (y desconocidos)
  - **Qué se dice**
  - Qué emociones expresan
  - ...
- Estas informaciones se pueden extraer y procesar automáticamente

## Audio y Voz en Big Data

- Grandes flujos de llamadas telefónicas
  - Call centers (e.g. 112, organismos públicos, grandes empresas)
  - Watchlists: monitorización actividad (e.g., SiTel)
  - Flujos de llamadas via VoIP (e.g. WhatsApp)
- Datos Broadcast (programas TV y radio – aéreo e Internet-)
  - Detección de publicidad y menciones publicitarias
  - Organización y clasificación de programas
  - Seguimiento de actividad e impacto mediático (políticos, etc.)
- Voz y audio en Internet (YouTube, redes sociales ...)
  - Gran diversidad (adquisición, contenido, entorno ...)
  - Organización de contenidos: clustering, alineamiento, sincronización
  - Detección de eventos acústicos

## Contenidos

- Audio y Voz en Big Data
- Speech-To-Text (STT) como usuario
- Search-on-Speech como usuario
- Comprendiendo el Speech-To-Text
  - Funcionamiento interno
- Comprendiendo el Search-on-Speech
  - Funcionamiento interno
- Conclusiones

## Speech-To-Text (STT) como usuario

- El Speech-To-Text (STT) es un sistema que permite transformar una grabación de voz en un texto que contiene, palabra por palabra, lo que se dice en la conversación



## Speech-To-Text (STT) como usuario

- Aplicación directa en Big Data:
  - Si tenemos audios/vídeos que contienen voz entre nuestros datos...
  - ... bastará con aplicar un STT a los audios/vídeos...
  - ... y ya los tendremos convertidos en textos en lenguaje natural...
  - ... que podremos procesar con técnicas de Procesamiento de Lenguaje Natural o de Recuperación de Información
- Esto nos permitirá analizar documentos que antes eran imposibles de analizar...

## Speech-To-Text (STT) como usuario

- Pero no todo es tan fácil...
  - Los STT (o reconocedores de voz) siguen cometiendo fallos:
    - Incluso los mejores no bajan de tasas de error de en torno a un 10-20% en habla conversacional
    - La tasa de error depende mucho de múltiples factores:
      - El tipo de habla
      - Entorno acústico: reverberación, ruido
      - Idioma: algunos idiomas son intrínsecamente más complicados o la tecnología no se adapta tan bien
      - Locutor: aunque los sistemas sean “independientes del locutor”, locutores con características muy marcadas en su voz pueden ser difíciles de reconocer
      - Emociones: el habla marcadamente emocional sigue siendo un problema
      - Velocidad del habla: habla muy rápida o muy lenta puede dar problemas
      - ...

## Speech-To-Text (STT) como usuario

- ¿Cómo saber entonces qué esperar?
  - La única opción es EVALUAR el reconocedor con los datos reales que se quieren procesar
  - Para ello es necesario:
    - Obtener un conjunto de grabaciones de evaluación
      - Suficientemente grande: 30' o 2000 palabras mínimo
      - Suficientemente variado: distintos locutores, condiciones acústicas relevantes, etc.
    - Transcribir manualmente las grabaciones
      - Requiere un mínimo de 10 veces el tiempo de la grabación
      - Idealmente, cada grabación transcrita por dos anotadores para analizar posteriormente los desacuerdos por otro anotador
      - Así se obtiene la transcripción de referencia (REF)

## Speech-To-Text (STT) como usuario

### ➤ ¿Cómo saber entonces qué esperar?

- La única opción es EVALUAR el reconocedor con los datos reales que se quieren procesar
- Para ello es necesario (ii):
  - Procesar el conjunto de grabaciones de evaluación con el STT
    - Así se obtiene la transcripción reconocida a evaluar (REC)
  - Alinear las transcripciones REF y REC para detectar:
    - Palabras reconocidas correctamente: H (Hits)
    - Palabras sustituidas por otras: S (Substitutions)
    - Palabras borradas: D (Deletions)
    - Palabras insertadas: I (Insertions)

## Speech-To-Text (STT) como usuario

### ➤ ¿Cómo saber entonces qué esperar?

- La única opción es EVALUAR el reconocedor con los datos reales que se quieren procesar
- Para ello es necesario (iii):
  - Ejemplo de alineamiento:
    - REF: porque nadie le entiende lo que esta diciendo
    - REC: porque nade le entiende que esta diciendo
    - Eval: H S H H D H H I S
  - Contar aciertos, sustituciones, borrados e inserciones:
    - $H = 5, S = 2, D = 1, I = 1$

## Speech-To-Text (STT) como usuario

### ➤ ¿Cómo saber entonces qué esperar?

- La única opción es EVALUAR el reconocedor con los datos reales que se quieren procesar
- Para ello es necesario (iv):
  - Contar aciertos, sustituciones, borrados e inserciones:
    - $H = 5, S = 2, D = 1, I = 1$
  - Calcular el Word Error Rate (WER):  $WER = \frac{S + D + I}{N} \times 100.0 \ (\%)$ 
    - N es el número de palabras en REF:  $N = H+S+D$  (En nuestro ejemplo  $N = 8$ )
    - Por tanto en nuestro ejemplo  $WER = (2 + 1 + 1) / 8 \times 100.0 = 50 \%$



## Speech-To-Text (STT) como usuario

### ➤ ¿Cómo saber entonces qué esperar?

- La única opción es EVALUAR el reconocedor con los datos reales que se quieren procesar
- Para ello es necesario (v):
  - Finalmente, interpretar el WER...
  - Un WER del 50% puede parecer mucho peor que lo que ha ocurrido aquí:
    - REF: porque nadie le entiende lo que esta diciendo
    - REC: porque nadie le entiende que esta diciendo
  - Además no todos los errores son igual de graves:
    - El borrado de "lo" no tiene mucha relevancia
    - En cambio el borrado de un nombre propio (ej. "Trump") puede hacer que todo el sistema de big-data o IR posterior falle



## Speech-To-Text (STT) como usuario

### ➤ ¿Cómo saber entonces qué esperar?

- La única opción es EVALUAR el reconocedor con los datos reales que se quieren procesar
- Para ello es necesario (vi):
  - Finalmente, interpretar el WER...
  - El WER puede ser mayor del 100%
    - Si hay muchas inserciones
  - Siempre hay un compromiso entre borrados e inserciones
    - Se puede aumentar uno a costa de reducir el otro
    - Se controla con un parámetro ajustable del reconocedor (word insertion penalty)
    - Tiene un impacto muy importante en el WER

## Speech-To-Text (STT) como usuario

### ➤ Medidas alternativas de rendimiento:

- Phone Error Rate (PER): Idéntico al WER pero operando sobre fonemas en lugar de sobre palabras.
- Sentence Error Rate (SER): Porcentaje de frases erróneas
  - No se contempla borrado ni inserción de frases
  - Suele ser muy alto (próximo al 100%) por efecto acumulativo de errores:
    - Una frase se considera errónea cuando contiene uno o más errores.
- Concept Error Rate (CER): Idéntico al WER pero operando sobre “conceptos” en lugar de sobre palabras
  - Es una evaluación conjunta del STT y de un analizador semántico posterior (NLP)
    - Habitual en sistemas conversacionales que operan sobre estos “conceptos”
  - Los “conceptos” se definen para una aplicación particular
  - Vienen a ser los lemas o raíz de las palabras útiles para la aplicación
  - Suele ser mucho menor que el WER

## Speech-To-Text (STT) como usuario

- Adaptación/Personalización del STT:
  - Como usuario es conveniente saber que un STT contiene:
    - Modelos acústicos (modelan cómo suena cada fonema)
    - Modelo léxico (lista finita de palabras conocidas con sus correspondientes transcripciones fonéticas)
    - Modelo de lenguaje (modela estadísticamente probabilidades de secuencias de palabras del léxico)
  - Todos estos modelos son dependientes del lenguaje, e incluso de la variante dialectal del lenguaje (ej. Español en distintos países e incluso regiones de España)
  - Ninguno de estos modelos puede ser óptimo en todos los entornos
  - Por tanto, en muchas circunstancias es necesaria (o al menos conveniente) una Adaptación / Personalización del STT

## Speech-To-Text (STT) como usuario

- Adaptación/Personalización del STT:
  - Ejemplo de necesidad de personalización:
    - Un reconocedor entrenado con voz de noticias de TV/Radio puede funcionar muy bien...
    - ... pero si se usa para transcribir conversaciones telefónicas espontáneas pasará a funcionar muy mal
      - Por el tipo de habla (formal y principalmente leída vs. informal y muy espontánea)
      - Y también por la calidad del audio (alta calidad vs. calidad media)
    - ... y si se usa para transcribir vídeos de YouTube el resultado puede ser mucho peor

## Contenidos

- Audio y Voz en Big Data
- Speech-To-Text (STT) como usuario
- Search-on-Speech como usuario
- Comprendiendo el Speech-To-Text
  - Funcionamiento interno
- Comprendiendo el Search-on-Speech
  - Funcionamiento interno
- Conclusiones

## Search-on-Speech como usuario

- ¿Qué es el Search-on-Speech?
  - Es un concepto amplio que incluye distintas formas de búsqueda y recuperación de información en repositorios de audio/vídeo que contienen voz humana
  - En general consiste en procesar los documentos de audio/vídeo buscando una query y devolviendo los documentos en los que se encuentra, posiblemente devolviendo información adicional como dónde se han encontrado las queries dentro de cada documento

## Search-on-Speech como usuario

### ➤ Tipos de Search-on-Speech

Tipo de Query	Textual	Resultado	
		Lista documentos	Lista de coincidencias indicando documento + tiempo de cada una
	Spoken Document Retrieval (SDR)	Spoken Term Detection (STD) o Keyword Spotting*	
Voz	Query-by-Example Spoken Document Retrieval (QbE SDR)	Query-by-Example Spoken Term Detection (QbE STD)	

\* STD y Keyword spotting se diferencian en que en STD se procesa el audio una vez y se genera un índice sobre el que posteriormente se puede buscar cualquier palabra. En KS si se cambian las palabras a buscar hay que reprocesar todo el audio.

## Search-on-Speech como usuario

### ➤ Search-on-Speech vs. Speech-To-Text

- Muchas veces la búsqueda en voz se basa en un sistema STT
- Todos los tipos de búsqueda en voz se pueden convertir en un problema de búsqueda en texto:
  - Aplicando un STT a los documentos (y a la query en caso necesario).
  - Sin embargo, esta no es la solución óptima
  - En la práctica se usan técnicas totalmente diferentes o basadas en la tecnología STT pero con modificaciones

## Search-on-Speech como usuario

- ¿Cómo EVALUAR un sistema de búsqueda en voz?
- Los sistemas de búsqueda en voz se evalúan empleando un conjunto de evaluación que incluye:
  - Un repositorio o conjunto de documentos de audio/vídeo que contienen voz humana (mínimo dos horas)
  - Un conjunto de queries (bien de texto o bien de voz) a buscar en el repositorio
    - Es necesario asegurar que haya queries que aparezcan un número suficiente de veces en el repositorio
    - También es conveniente incluir algunas queries que no aparezcan
  - Un etiquetado (manual) de referencia de las queries en los documentos
    - Si sólo nos interesa recuperar documentos (SDR) bastará con indicar qué queries hay presentes en cada documento
    - Si queremos localizar la query dentro del documento (STD) será necesario indicar por cada aparición de cada query en cada documento el instante inicial y final de la query

## Search-on-Speech como usuario

- Métricas de evaluación para sistemas SDR
- A partir de tabla de resultados de clasificación:
  - Por cada combinación query - documento se suma uno a una de estas cuatro posibilidades:

		¿Se ha reconocido la query en el documento?	
		SÍ	NO
¿El documento incluye la query realmente? (en REF)	SÍ	True Positive TP	False Negative FN
	NO	False Positive FP	True Negative TN

- Posteriormente, se calculan métricas con ellas (F-score, precisión/recall)

## Search-on-Speech como usuario

### ➤ Métricas de evaluación para sistemas STD

- STD se diferencia de SDR en que el sistema devuelve todas las ocurrencias de la query que ha encontrado indicando:
  - El documento en el que se ha encontrado
  - Y también el tiempo de inicio y final de cada
- Si aparece más de una vez una query en el documento el sistema tiene que detectarla más de una vez
- Cada detección del sistema puede ser:
  - Hit (acierto o TP) o False Alarm (falso positivo o FP)
- Se permite una cierta tolerancia (0.5 segundos) al comparar los tiempos

## Search-on-Speech como usuario

### ➤ Métricas de evaluación para sistemas STD

### ➤ A partir de los resultados de detección:

- Por cada query detectada se suma uno a una de estas dos posibilidades, y se calculan métricas con ellas

¿Coincide la query detectada en documento y tiempo con la misma query del etiquetado de referencia?

Sí	NO
Hit	False Alarm
H	FA

## Contenidos

- Audio y Voz en Big Data
- Speech-To-Text (STT) como usuario
- Search-on-Speech como usuario
- Comprendiendo el Speech-To-Text
  - Funcionamiento interno
- Comprendiendo el Search-on-Speech
  - Funcionamiento interno
  - Limitaciones
- Conclusiones

## Comprendiendo el Speech-to-Text: Funcionamiento Interno

- Los sistemas de Speech-to-Text incluyen:
  - Un módulo de procesamiento de señal (señal de voz → parámetros)
    - Procesa el audio y lo representa como una secuencia de vectores de parámetros que contienen información relevante, visto en la introducción
  - Un modelo acústico-fonético (parámetros → fonemas)
    - Modela cómo es el sonido (más concretamente los parámetros extraídos por el módulo anterior) para cada fonema (en realidad para cada parte de cada fonema en un contexto fonético dado)
  - Un modelo léxico (fonemas → palabras)
    - Recoge las palabras conocidas por el reconocedor y su transcripción fonética (o sus posibles transcripciones fonéticas)
  - Un modelo de lenguaje (palabras → frases)
    - Modela qué secuencias de palabras son más probables y cuáles menos en un determinado lenguaje (y dialecto, y contexto...)

# Comprendiendo el Speech-To-Text: Funcionamiento Interno

## Tarea Básica: Reconocimiento de Dígitos Aislados

28

28

## Tarea Básica: Reconocimiento de Dígitos Aislados

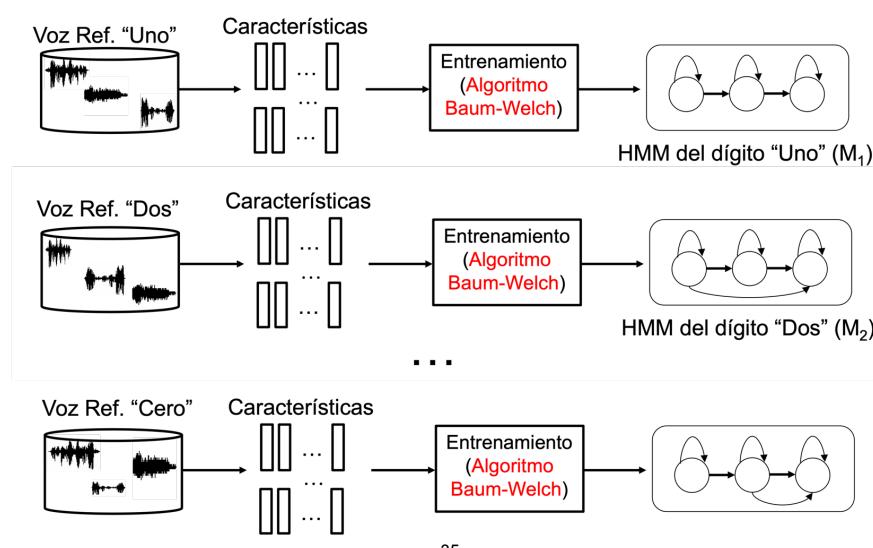
- Reconocimiento de dígitos aislados
  - ▣ Voz pronunciando dígitos (es decir, números: “uno”, “dos”, “tres”...)
- Se dispone de
  - ▣ Muchos fragmentos de voz (base de datos de voz) en los que **conocemos** que se está pronunciando un dígito
    - Por ejemplo: una persona que sabemos que está diciendo “uno”
    - “Voz conocida” o “Voz de referencia”
  - ▣ Un fragmento de voz en el que se supone que se está diciendo un dígito
    - Pero no sabemos cuál es ese dígito
    - “Voz desconocida” o “Voz de prueba” o “test”
- Tarea:
  - ▣ Determinar qué dígito está pronunciándose en la voz desconocida

29

## Reconocimiento de Dígitos Aislados con Modelos Ocultos de Markov (HMM)

- Un modelo oculto de Markov (HMM) es un modelo probabilístico
  - ▣ Contiene toda la información acerca de lo que queremos reconocer
    - Extraída de la señal de voz
  - ▣ Luego veremos qué son estos modelos
    - Por el momento, simplemente hay que saber que resumen la información sobre el dígito a reconocer
- Dos pasos:
  - ▣ Entrenamiento de los modelos HMM (usando voz de referencia)
  - ▣ Reconocimiento del dígito que se pronuncia en la voz de test

## Dígitos Aislados con HMM: Entrenamiento



## Tarea principal

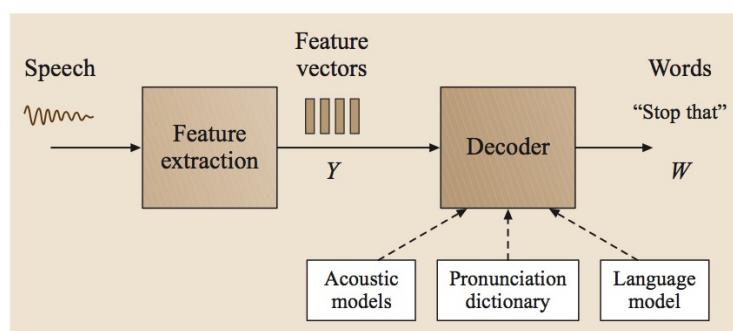
- Dada la secuencia de observaciones  $\mathbf{Y} = \mathbf{y}_1, \dots, \mathbf{y}_T$
- El decodificador busca la secuencia de palabras del léxico  $\mathbf{W} = \mathbf{w}_1, \dots, \mathbf{w}_k$  con mayor probabilidad de haber generado  $\mathbf{Y}$  buscando

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} [p(\mathbf{Y}|\mathbf{W})p(\mathbf{W})]$$

↑                      ↑  
 Modelo                  Modelo de  
 Acústico (HMM)    Lenguaje

## Arquitectura básica

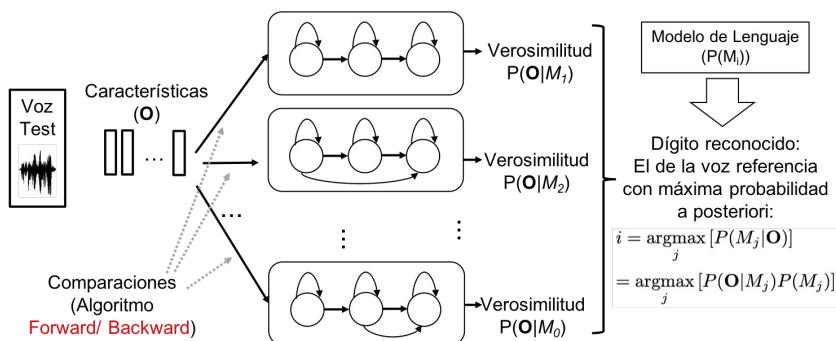
- Modelo acústico (por ejemplo, un HMM)
- Diccionario de pronunciaciones (en la tarea de dígitos aislados no es necesario, porque cada modelo está asociado a un dígito completo)
- Modelo de lenguaje (típicamente n-Gramas)



**Fig. 27.1** Architecture of an HMM-based recognizer

## Ejemplo: Dígitos Aislados HMM

- Imaginemos que cada dígito  $w_i$  está representado por su HMM  $M_i$  (modelo acústico)
- Si un dígito  $w_1$  aparece más que otro  $w_2$  en un idioma determinado...
  - ▣ El modelo de lenguaje asignará  $P(M_1) > P(M_2)$  (o, lo que es lo mismo,  $P(w_1) > P(w_2)$ )
- Por tanto, hay que decidir en base a la probabilidad de palabra vistas las observaciones
  - ▣ Probabilidad a posteriori, o probabilidad de un dígito vistas las observaciones

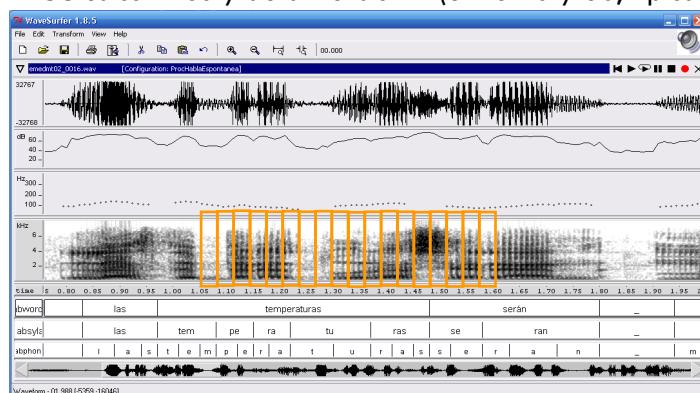


## Comprendiendo el Speech-To-Text: Funcionamiento Interno

Modelo Acústico-Fonético

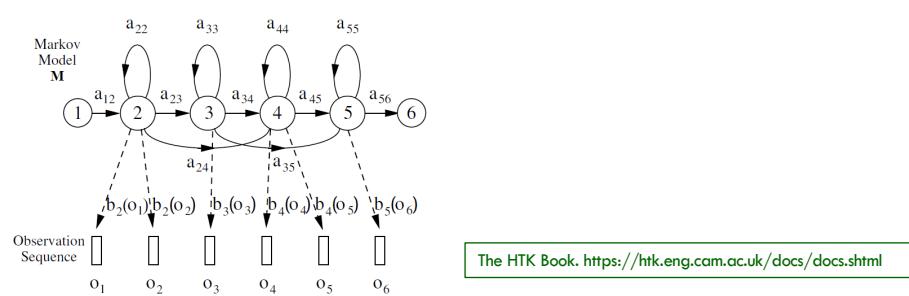
## Modelo Acústico-Fonético

- Observamos la voz a través de ventanas (observacionesMFCC) que representan el espectro, pero no sabemos qué se dice en cada instante
  - También llamadas tramas, parámetros, observaciones o características MFCC
  - Cada vector MFCC es continuo y de dimensión D (entre 20 y 80, típicamente)



## HMM con Observaciones Continuas: Ejemplo

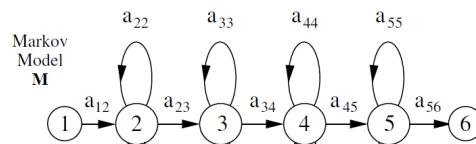
- Modelo izquierda-derecha
  - ▣ Típico en voz (usado para modelar fonemas o palabras/frases cortas)
- Modelo generativo de estados finitos
  - ▣ Si tenemos un modelo entrenado, podemos generar características
  - ▣ Típicamente, un estado o se mantiene en él mismo o transita al siguiente (arquitectura izquierda-derecha, o Bakis)



The HTK Book. <https://htk.eng.cam.ac.uk/docs/docs.shtml>

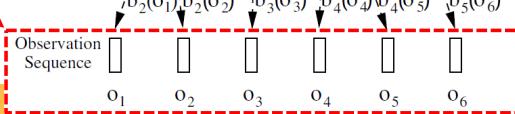
## HMM con Observaciones Continuas: Ejemplo

- ¿Cómo se generan observaciones LPCC/MFCC con un HMM?
  - ▣ El modelo está en el estado “ $i$ ” (círculos con números) en el instante temporal  $n$ , con cierta probabilidad
  - ▣ En cada momento temporal, el modelo **genera** una observación de acuerdo con una función de probabilidad  $b_i(o)$ 
    - Caso continuo:  $b_i(o)$  es una función densidad de probabilidad multidimensional
      - El caso de características (ej. MFCC), que son continuas y vectoriales
  - ▣ Tras ello, el modelo se actualiza, y pasa al estado  $j$  con probabilidad de transición  $a_{ij}$



Características generadas

The HTK Book.  
<https://htk.eng.cam.ac.uk/docs/docs.shtml>



Observation Sequence



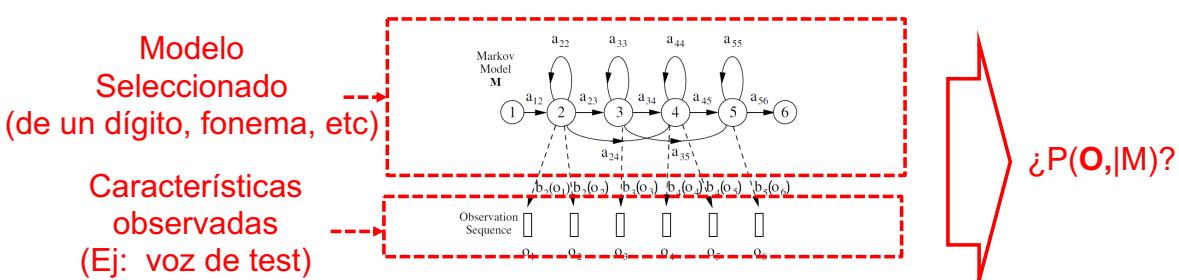
Máster en Big Data y Data Science

Indexación de voz & audio

38

## Modelo Oculto de Markov (HMM)

- Un HMM es un modelo generativo puede usarse para generar nuevas características
- Pero también para el **paso inverso**
  - ▣ Tengo unas características
  - ▣ Quiero saber si es probable que las generara un determinado modelo
  - ▣ Es decir: **cálculo de la verosimilitud**
    - Suponiendo un modelo dado
- Se utiliza para ello el algoritmo forward/backward



Máster en Big Data y Data Science

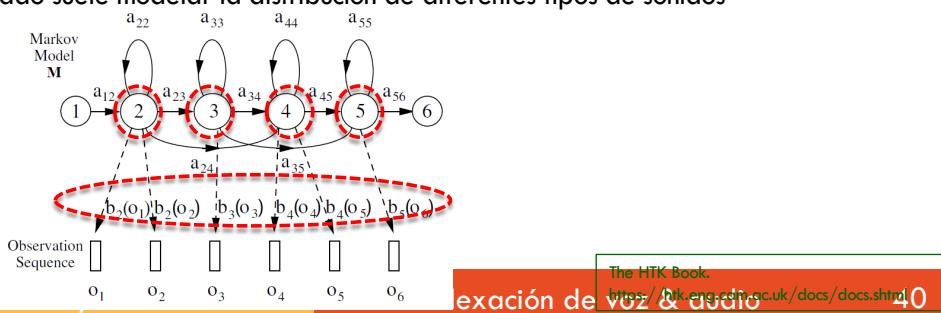
Indexación de voz & audio

39

39

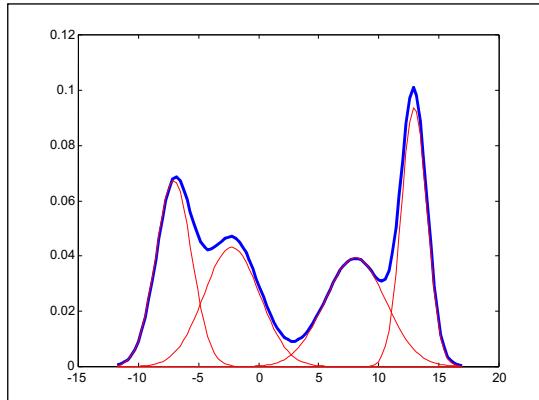
## Función Densidad de Probabilidad en Cada Estado

- En ASR se suele representar como un modelo de mezclas de gaussianas (GMM)
  - ▣ Un GMM representa cómo se distribuyen los MFCC/LPCC en dicho estado
    - En el gráfico inferior, dicha función densidad de probabilidad es  $b_i(o)$ , una para cada estado  $i$  entre 2 y 5 (los estados 1 y 6 no generan observaciones)
    - Intuitivamente, cómo se distribuye el tipo de sonido en un estado determinado
  - ▣ Cada GMM de cada estado es normalmente diferente
    - Porque cada estado suele modelar la distribución de diferentes tipos de sonidos



## GMMs

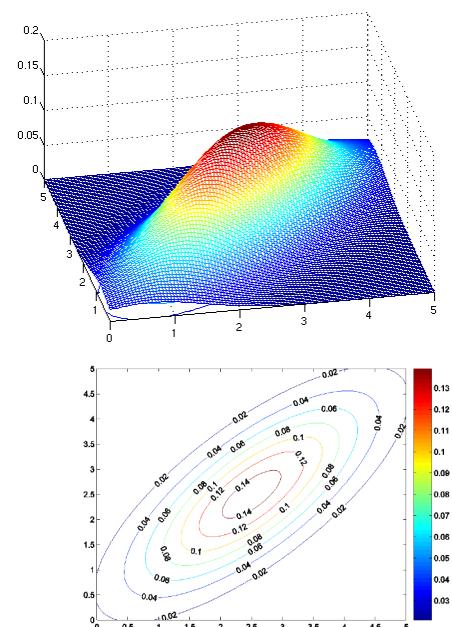
- **Modelo para el espacio de MFCCs: Gaussian Mixture Models (GMMs)**
  - Cada vector en la secuencia de vectores es una “muestra”
  - Con todas las muestras (de un fonema o estado de un fonema) se obtiene un modelo de mezclas de gaussianas GMM
  - Función densidad de probabilidad multidimensional



## Modelo Acústico-Fonético

### ➤ Gaussiana Multidimensional

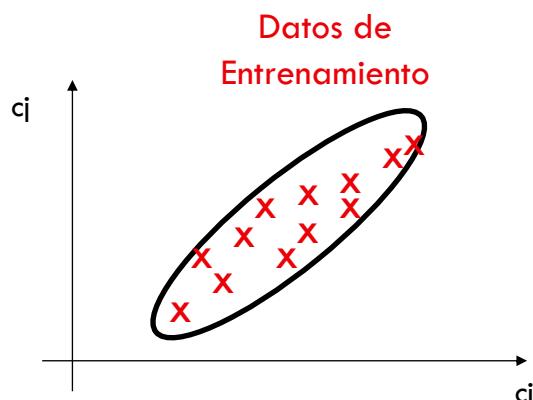
- Función densidad de probabilidad dependiente de dos variables
- Considera la variación de cada variable y la de una frente a la otra (correlación)



## Modelo Acústico-Fonético

### ➤ Gaussiana Multidimensional

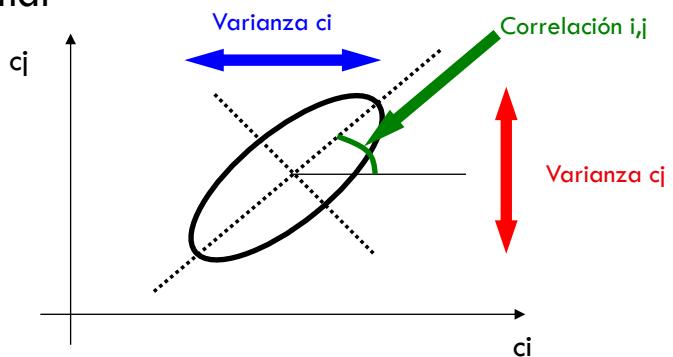
- Calculada a partir de vectores de entrenamiento
- Representación elíptica (curva de nivel)



## Modelo Acústico-Fonético

### ➤ Gaussiana Multidimensional

- Formulación:

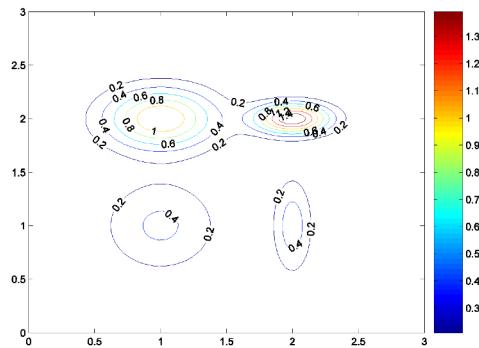
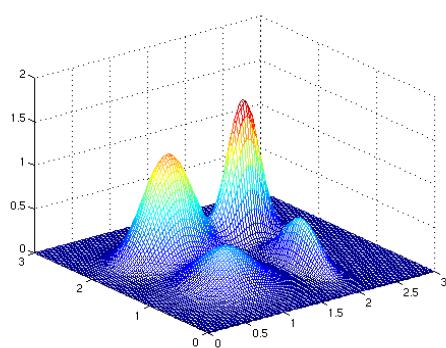


$$g(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} \cdot |\Sigma|^{1/2}} \exp \left[ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \cdot \Sigma^{-1} \cdot (\mathbf{x} - \boldsymbol{\mu}) \right] = N(\boldsymbol{\mu}, \Sigma)$$

## Modelo Acústico-Fonético

### ➤ Gaussian Mixture Models (GMM)

- GMM de  $M=4$  mezclas



## Modelo Acústico-Fonético

### ➤ Modelo GMM (paramétrico)

$x_t$ : MFCC coefficient vector from frame t

Model Mean Vector:

$$\mu_p = \{\mu_{ip}\}$$

Covariance Matrix:

$$\Sigma_p = \{\Sigma_{ip}\}$$

Weights Vector:

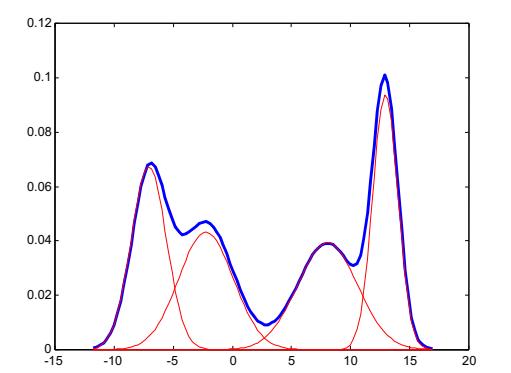
$$\omega_p = \{\omega_{ip}\}, \sum_i \omega_{ip} = 1$$

Model of phone p:

$$\lambda_p = \{\mu_p, \Sigma_p, \omega_p\},$$

$$p(\mathbf{x}|\lambda_p) = \sum_{i=1}^M \omega_{ip} g_{ip}(\mathbf{x})$$

$$g_{ip}(\mathbf{x}) = N(\boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip})$$



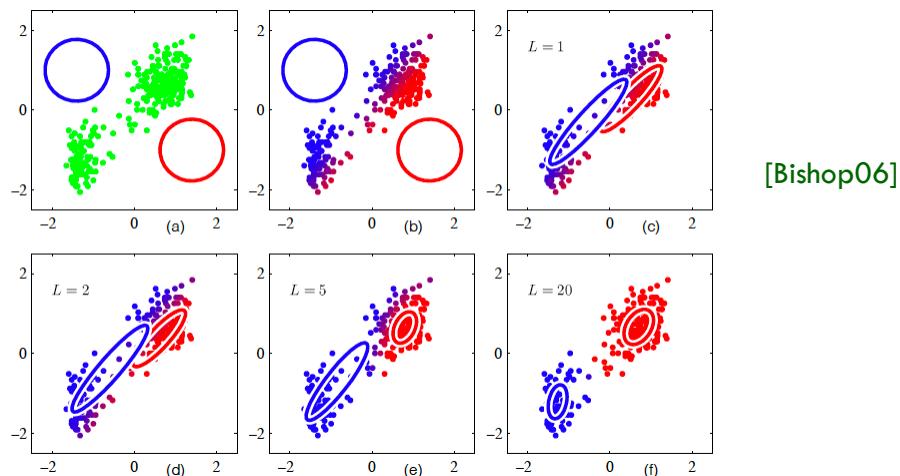
## Modelo Acústico-Fonético

### ➤ Modelo GMM (paramétrico)

- Los parámetros del modelo se obtienen (usualmente) mediante el algoritmo EM (Expectation Maximization)
- EM es una implementación del entrenamiento ML (máxima verosimilitud)
- Algoritmo iterativo
- Cuantas más iteraciones, más se ajustará el modelo a los datos

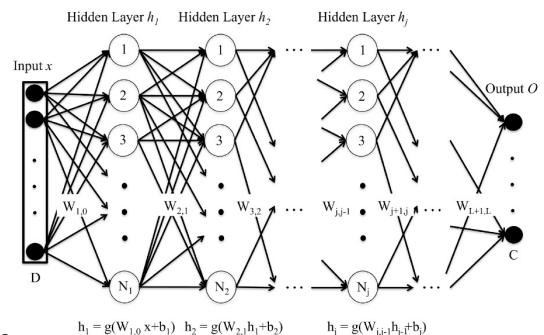
## Ejemplo: Entrenamiento GMM con (EM)

- Ejemplo ( $L$  es el número de iteraciones de EM),  $D=2$  dimensiones,  $M=2$  componentes:



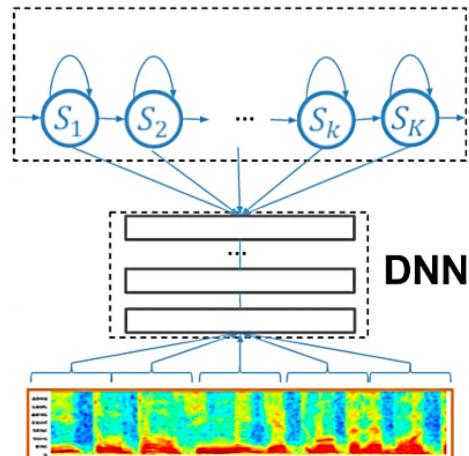
## Modelos Híbridos (HMM/DNN)

- Entrenan primero un sistema HMM/GMM convencional
- Al final sustituyen el GMM por una Deep Neural Network (DNN) entrenada para estimar, por cada frame de entrada, la probabilidad de que el frame se corresponda con cada uno de los estados de los HMMs
  - Ventajas: Las DNNs permiten mejor clasificación y por tanto los sistemas de reconocimiento mejoran notablemente



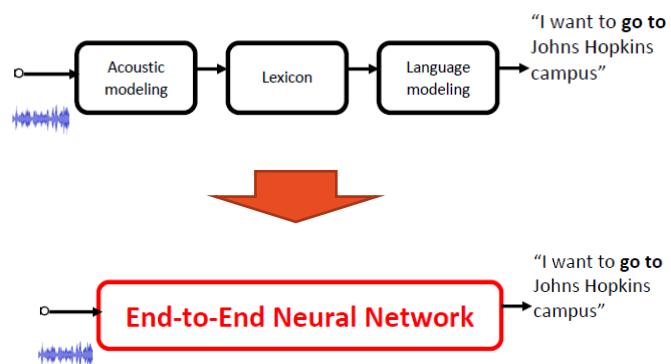
## Modelos Híbridos (HMM/DNN)

- Entrenan primero un sistema HMM/GMM convencional
- Al final sustituyen el GMM de todos los estados por una Deep Neural Network (DNN) entrenada para estimar, por cada frame de entrada, la probabilidad de que el frame se corresponda con cada uno de los estados de los HMMs
- Ventajas: Las DNNs son algoritmos muy potentes si se entrena con datos suficientes, y por tanto los sistemas de reconocimiento mejoran notablemente



## Modelos Neuronales End-to-End

- Sustituyen completamente los modelos HMM
- Emplean una única red neuronal que integra modelo acústico, léxico y modelo de lenguaje
  - Aunque en la práctica se aplican modelos de lenguaje externos a posteriori para mejorar resultados



Ver por ejemplo ESPnet Tutorial @ INTERSPEECH'19  
<https://github.com/espnet/interspeech2019-tutorial>

## Comprendiendo el Speech-To-Text: Funcionamiento Interno

Arquitectura de un reconocedor basado en HMMs:  
Combinando todos los modelos para el reconocimiento

52

52

### Tarea principal

- Dada la secuencia de observaciones  $\mathbf{Y} = \mathbf{y}_1, \dots, \mathbf{y}_T$
- El decodificador busca la secuencia de palabras del léxico  $\mathbf{W} = \mathbf{w}_1, \dots, \mathbf{w}_k$  con mayor probabilidad de haber generado  $\mathbf{Y}$  buscando

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} [p(\mathbf{Y}|\mathbf{W})p(\mathbf{W})]$$

The diagram shows the joint probability formula. Two arrows point upwards from two boxes below to the terms  $p(\mathbf{Y}|\mathbf{W})$  and  $p(\mathbf{W})$  respectively. The left box is orange and labeled "Acoustic model". The right box is green and labeled "Language model".

# Arquitectura

- Unidad básica del modelo acústico: fonema
  - ▣ ~25 fonemas en castellano, ~40 fonemas en inglés

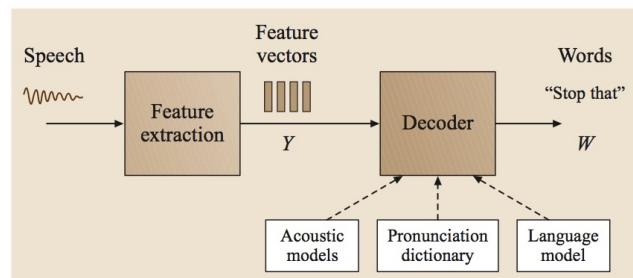


Fig. 27.1 Architecture of an HMM-based recognizer

THE INTERNATIONAL PHONETIC ALPHABET (revised to 2005)											
© 2005 IPA											
CONSONANTS (PULMONIC)											
	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glossal
Plosive	p b		t d		t̪ d̪	c j	k g	q ɣ		?	
Nasal	m	n̪		n		ɳ	p̪ t̪	ɳ̪	N		
Tall	B		r̪						R		
Tap or Flap		v̪	f̪		t̪						
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ɟ	x ɣ	χ ʁ	h ɦ	h̪ ɦ̪
Lateral fricative			ɬ ɺ								
Approximant		v̪	I̪		l̪	j̪	w̪				
Lateral approximant			l̪		l̪	y̪	l̪				

Where symbols appear in pairs, the one to the right represents a voiced consonant. Shaded areas denote articulations judged impossible.

CONSONANTS (NON-PULMONIC)			VOWELS							
Clicks	Voiced implosives	Ejectives								
ʘ Bilabial	b̪ Bilabial	,								
Dental	d̪ Dental/alveolar	p̪ Bilabial								
! (Post)alveolar	f̪ Palatal	t̪ Dental/alveolar								
ǂ Palatoalveolar	g̪ Velar	k̪ Velar								
Alveolar/lateral	g̪ Uvular	s̪ Alveolar fricative								

OTHER SYMBOLS

ʍ Voiced labio-velar fricative	ç Z Alveolo-palatal fricatives
w̪ Voiced labial-velar approximant	ɿ Voiced alveolar/lateral flap

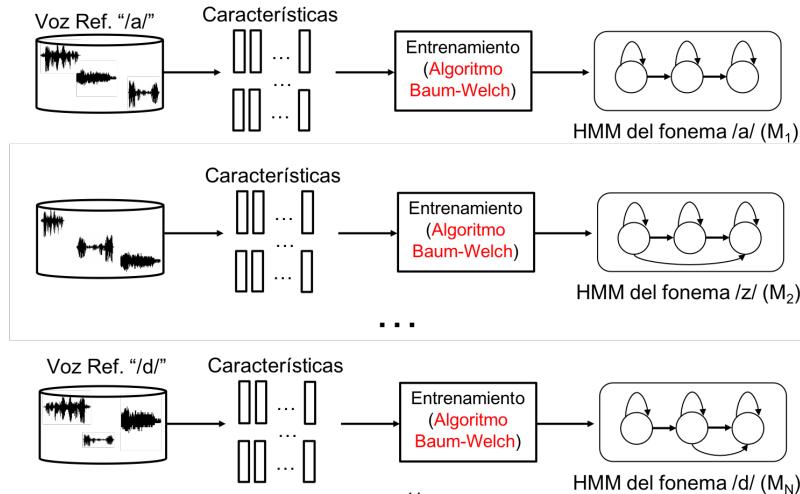
Front      Central      Back

Close i • y — i • u — u  
Close-mid e • ø — e • θ — θ  
Open-mid ɔ — ə — ɑ  
Open a — œ — ɒ — ɑ

Where symbols appear in pairs, the one to the right represents a rounded vowel.

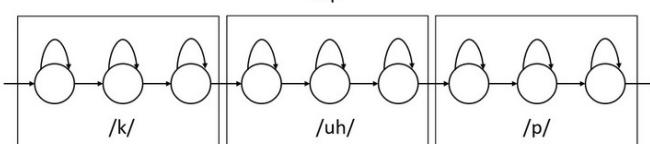
## Modelos Acústicos HMM: Entrenamiento

- Si entrenamos HMMs a nivel de N fonemas, tendremos N modelos HMM:



## Modelos acústicos: de fonemas a palabras

- Varios modelos HMM se pueden concatenar
  - Usando como “conectores” los estados que no emiten ninguna observación
    - Típicamente el inicial y el final en arquitectura Bakis (de izquierda a derecha)  
“cup”
- De ese modo podemos construir...
  - Modelos de palabras con modelos de fonemas utilizando un diccionario de pronunciación
    - Llamado “Continuous Speech Recognition”
  - Modelos de frases concatenando palabras
    - Llamado “Connected Speech Recognition”
- Se puede entrenar el modelo compuesto (también con el algoritmo Baum-Welch, y usando una o varias locuciones lo cual llamamos **reestimación**)



## Modelos dependientes de contexto

- Habla real: alta variación contextual
  - ▣ /a/ diferente en /mano/ y /caso/
  - ▣ Presente más allá de los límites de palabra
- Monofonemas no capturan esta variabilidad
- Solución: modelos fonéticos para cada par de vecinos izquierda-derecha → trifonemas
- Si N fonemas →  $L=N^3$  trifonemas (potenciales)
  - ▣ Castellano:  $\sim 25^3=15625$
  - ▣ Inglés:  $\sim 40^3=64000$

## Modelos dependientes de contexto

- Modelo de la palabra “cup” con HMM de fonemas independientes de contexto
 

“cup”
- Y con modelos dependientes de (modelos de trifonemas)
 

/ay-k+uh/      /k-uh+p/      /uh-p+uh/
- Trifonemas necesitan muchos datos y estrategias de optimización de datos
  - ▣ En realidad hay que entrenar muchísimos más modelos

## Modelos de lenguaje

- N-gramas: la probabilidad de cada palabra depende sólo de las N-1 anteriores

$$p(W) = \prod_{k=1}^K p(w_k | w_{k-1}, \dots, w_1)$$

- En sistemas de gran vocabulario el condicionamiento se reduce a las N-1 anteriores ( $N \sim 2-4$ )

$$p(W) = \prod_{k=1}^K p(w_k | w_{k-1}, w_{k-2}, \dots, w_{k-N+1})$$

- N-gramas se estiman contando ocurrencias del N-grama en corpora de texto

## Estimación de probabilidad de N-grama

- Si  $C(\cdot)$  representa el número de ocurrencias:

$$p(w_k | w_{k-1}, w_{k-2}) \approx \frac{C(w_{k-2} w_{k-1} w_k)}{C(w_{k-2} w_{k-1})}$$

- Si hay escasez de datos → suavizado
- Modelos de lenguaje efectivos para sistemas de gran vocabulario:
  - ▣ Trigramas de palabras

## Decodificación

- La secuencia de palabras más probable se encuentra buscando todas las secuencias posibles de estados derivadas de todas las secuencias posibles de palabras.
- Resolución mediante programación dinámica
  - ▣ Algoritmo de Viterbi

$$\phi_j(t) = \max_i \{ \phi_i(t-1) a_{ij} \} b_j(y_t)$$

- ▣ El backtracking devuelve la mejor secuencia de estados y palabras

## Viterbi en Large Vocabulary Systems

- Implementación directa inabordable
- Construcción de gramáticas de reconocimiento
  - ▣ Representa lo que se puede pronunciar

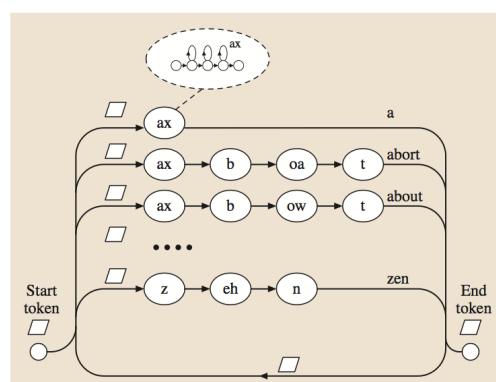


Fig. 27.6 Basic recognition network

## Concepto de “tokens”

- Dada la red, en cada instante  $t$  en la búsqueda:
  - ▣ Cada hipótesis es un camino (alineamiento de estados):
    - Comienza en estado inicial
    - Termina en estado  $j$
    - Loglikelihood acumulado  $\log \varphi_i(t)$
  - ▣ Lo representamos con un token  $\{\log P, \text{link}\}$ 
    - $\log P$ : score (log likelihood)
    - Link: puntero al registro de información histórica
- Cada nodo de la red (estados del HMM) puede almacenar un token
- El reconocimiento continúa propagando los tokens por la red

## Token-passing algorithm

- Los tokens pasan de nodo en nodo
- En cada transición se actualiza el score y el link
- Al salir de una palabra y entrar en otra:
  - ▣ Se actualiza el score con la probabilidad del modelo de lenguaje
  - ▣ En un registro  $R$  se guarda:
    - Copia del token
    - Instante actual
    - Palabra anterior
    - Link del token se apunta a  $R$
- El mejor token en el instante  $T$  en un nodo de salida válido devuelve la secuencia de palabras más probable y sus límites temporales

# Token-passing algorithm

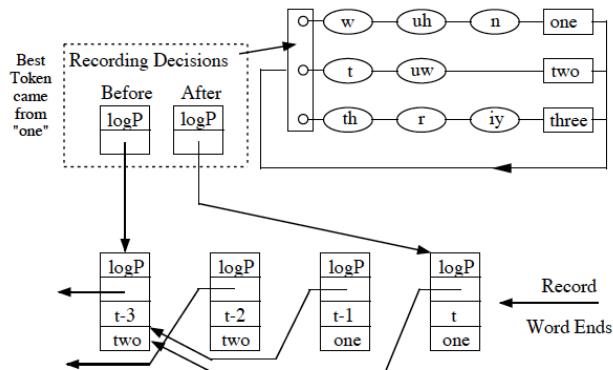


Fig. 1.8 Recording Word Boundary Decisions

## Requisitos en Continuous Speech Rec. (CSR)

- Beam search: sólo se propagan los más probables
  - ▣  $\log P > \text{beam width}$
- 90% CPU en los dos primeros fonemas de cada palabra:
  - ▣ La red debe ser tree structured para compartir los estados iniciales de cada palabra

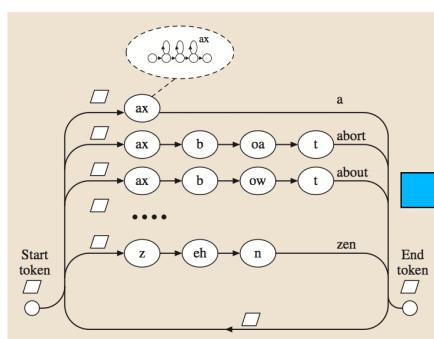


Fig. 27.6 Basic recognition network

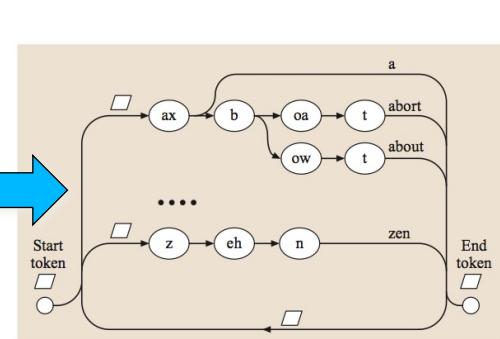
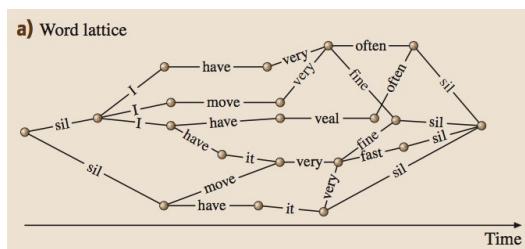


Fig. 27.8 Tree-structured recognition network

## Lattices (celosías)

- Token passing válido para reconocimiento "de un solo paso" (por ejemplo, en tiempo real)
  - ▣ Perdemos todas las hipótesis no-ganadoras
- Para otras aplicaciones conviene guardar múltiples hipótesis para su uso posterior
  - ▣ Reconocimiento que no sea en tiempo real, reconocimiento de término hablado (Spoken Term Detection), etc.
- Word lattices: conjuntos de nodos representando instantes temporales y arcos representando hipótesis de palabras (y sus probabilidades)



## Reconocedores end-to-end DNN

- Se ha propuesto arquitecturas de reconocedores que eliminan completamente la necesidad de los HMM
- Emplean técnicas de DNN (especialmente redes recurrentes) para convertir directamente con una DNN:
  - ▣ Secuencia de vectores de parámetros → Secuencia de fonemas
  - ▣ Sequence-to-Sequence mapping
- Veremos por encima algunas aproximaciones populares
  - ▣ Pero aún en fase de investigación
- Ofrecen muy buenos rendimientos en tareas "pequeñas"
- Pero en la práctica lo más habitual son los modelos Híbridos

## Reconocedores end-to-end DNN

- Connectionist Temporal Classification (CTC)
- RNN Transducer
- Attention-Based Encoder-Decoder (Att)
- Transformer

## Reconocedores end-to-end DNN

- Connectionist Temporal Classification (CTC)
- RNN Transducer
- Attention-Based Encoder-Decoder (Att)
- Transformer

## Connectionist Temporal Classification (CTC)

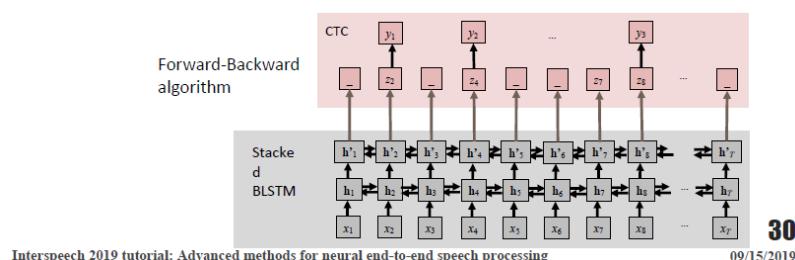
### Connectionist temporal classification (CTC)

[Graves+ 2006, Graves+ 2014, Miao+ 2015]

- Use bidirectional RNNs to predict frame-based labels including blanks
- Find alignments between  $X$  and  $Y$  using dynamic programming

😊 Simple implementation (built-in & cudnn), on-line, fast

😢 Poor performance (conditional independence assumptions), limited applications



## The problem: Sequence Labelling

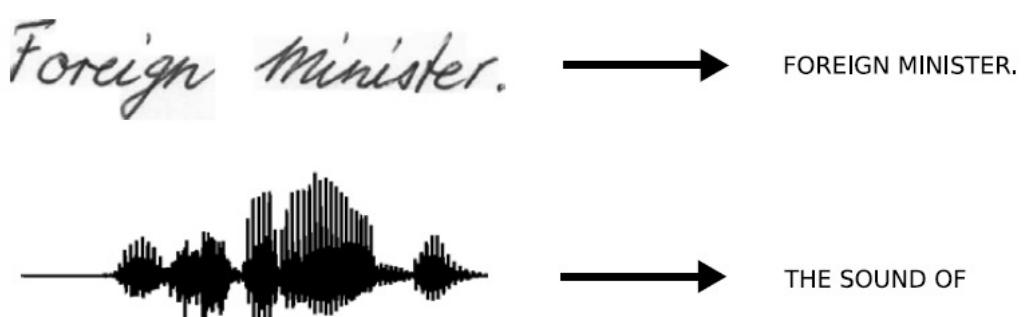


Fig. 2.1 Sequence labelling. The algorithm receives a sequence of input data, and outputs a sequence of discrete labels.

## Why is sequence labelling so hard for Recurrent Neural Networks?

- Standard neural network objective functions are defined separately for each point in the training sequence
- In other words, RNNs can only be trained to make a series of independent label classifications
- This means that the training data must be pre-segmented...
  - That's why Hybrid HMM-DNN systems became popular
    - HMMs can pre-segment data!!
- And that the network outputs must be post-processed to give the final label sequence
  - Transforming frame-by-frame posteriors to final decisions

## What does CTC propose?

- CTC removes the need for presegmented training data and post-processing outputs
- By computing a loss (CTC Loss) that integrates over all possible alignments
- This loss is differentiable, so the RNNs can be trained based on it

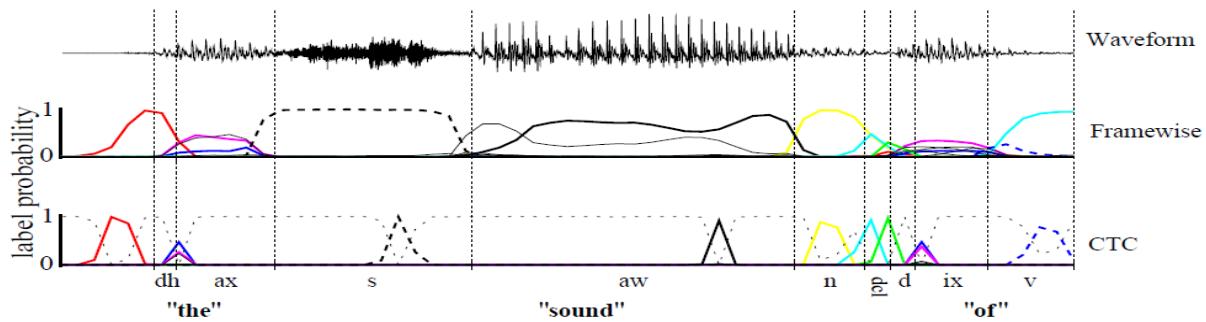
## What does CTC propose?

- CTC removes the need for presegmented training data and post-processing outputs
- By computing a loss (CTC Loss) that integrates over all possible alignments
- This loss is differentiable, so the RNNs can be trained based on it

## Main idea of CTC

- A RNN network has an output for each input time
- But in ASR the number of phones is much smaller than the number of frames
  - ▣ E.g. 10-20 phones/s and 100 frames/s
- The output of the RNN network can predict a phone or no phone (blank, “\_”)
  - ▣ If a language has 45 phones, use 45+1 outputs
  - ▣ Last one means “no output”, represented as “\_”

## CTC Typical Outputs



- Output symbols predicted only at spikes
- With the exception of the blank symbol

## Reconocedores end-to-end DNN

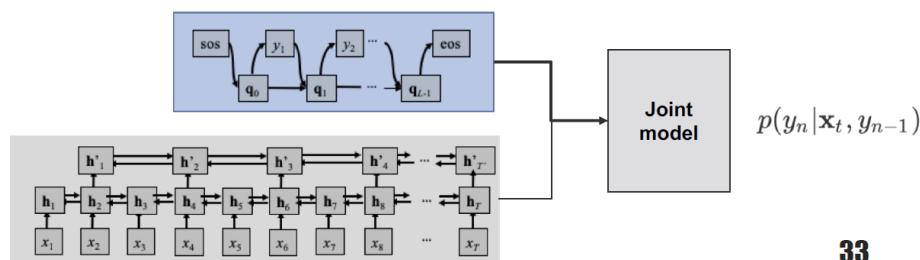
- Connectionist Temporal Classification (CTC)
- RNN Transducer
- Attention-Based Encoder-Decoder (Att)
- Transformer

# RNN Transducer - Overview

## RNN-transducer [Graves+ 2013]

- Extension of CTC by considering previous output dependency
- Combine input RNN and auto-regressive output RNN to provide a joint distribution
  - Joint model can handle this combination

😊 Good performance with reasonable alignment, on-line  
 ☹ Complicated implementation, slow, limited applications



Interspeech 2019 tutorial: Advanced methods for neural end-to-end speech processing

33

09/15/2019

## Reconocedores end-to-end DNN

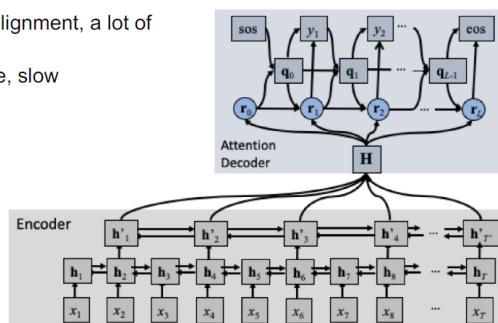
- Connectionist Temporal Classification (CTC)
- RNN Transducer
- Attention-Based Encoder-Decoder (Att)
- Transformer

## Attention-Based Encoder-Decoder - Overview

### Attention-based encoder decoder [Chorowski+ 2015, Chan+ 2016]

- Encoder: acoustic model, decoder: RNN language model, attention: align input and output labels
- No conditional independence assumption

😊 Good performance but too flexible alignment, a lot of applications (ASR, TTS, NMT)  
 😢 Complicated implementation, off-line, slow



Interspeech 2019 tutorial: Advanced methods for neural end-to-end speech processing

09/15/2019



## Reconocedores end-to-end DNN

- Connectionist Temporal Classification (CTC)
- RNN Transducer
- Attention-Based Encoder-Decoder (Att)
- Transformer



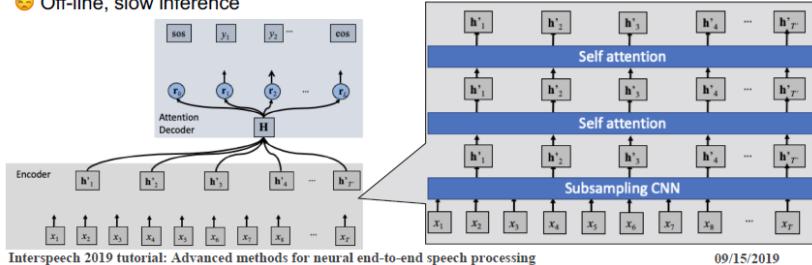
# Transformer - Overview

## Transformer [Vaswani+ 2017, Dong+ 2018]

- Replace all recurrent connections in attention-based encoder-decoder with a self attention block (can capture very long-range dependency)
- All operations across time is well parallelized

😊 Very good performance with reasonable alignment, fast training, a lot of applications (ASR, TTS, NMT), relatively simple implementation

😴 Off-line, slow inference



# Speech Transformer – Original Paper

## SPEECH-TRANSFORMER: A NO-RECURRENCE SEQUENCE-TO-SEQUENCE MODEL FOR SPEECH RECOGNITION

Linhao Dong<sup>1,2</sup>, Shuang Xu<sup>1</sup>, Bo Xu<sup>1</sup>

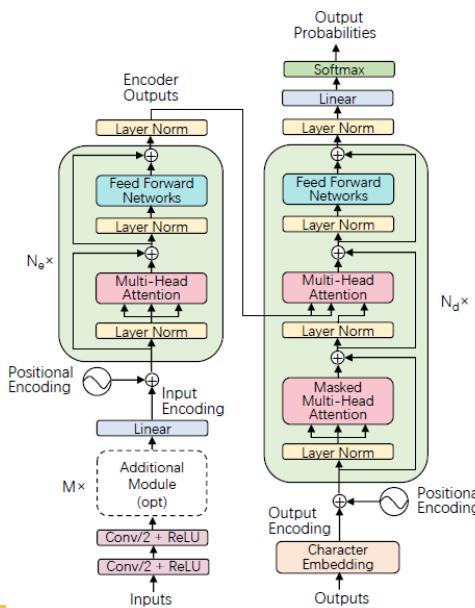
<sup>1</sup>Institute of Automation, Chinese Academy of Sciences, China

<sup>2</sup>University of Chinese Academy of Sciences, China

{donglinhao2015, shuang.xu, xubo}@ia.ac.cn

Dong, L., Xu, S., & Xu, B. (2018, April). Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5884-5888). IEEE.

# Speech Transformer – Main Changes



## Referencias

- Benesty, Shondi and Huang (Eds.), “Springer Handbook of Speech Processing”, Springer-Verlag, 2008  
(Algunas de las figuras de este capítulo han sido tomadas de este libro)

## Contenidos

- Audio y Voz en Big Data
- Speech-To-Text (STT) como usuario
- Search-on-Speech como usuario
- Comprendiendo el Speech-To-Text
  - Funcionamiento interno
- Comprendiendo el Search-on-Speech
  - Funcionamiento interno
- Conclusiones



## Comprendiendo el Search-on-Speech: Funcionamiento Interno

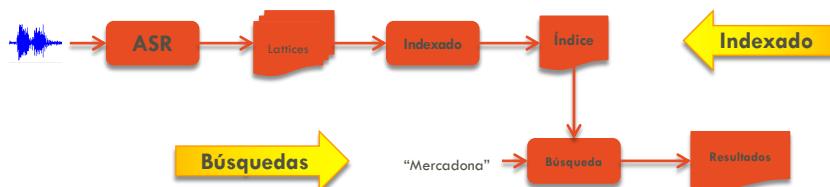
---

Arquitectura de Sistemas Search-on-Speech:  
Text-Based Spoken Term Detection (STD) o Spoken  
Document Retrieval (SDR)

## Arquitectura de Sistemas Text-Based STD o SDR



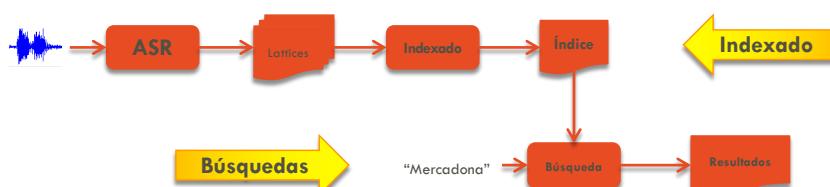
- Spoken Term Detection (STD): encontrar coincidencias de un término en un documento
  - Si hay 5 coincidencias, se devuelven 5 resultados (con el momento temporal de cada uno en el audio)
- Spoken Document Retrieval (SDR): encontrar documentos que contengan la palabra buscada
  - Si hay 5 coincidencias, se devuele el audio entero



## Arquitectura de Sistemas Text-Based STD o SDR



- Objetivo: Conseguir eficacia y eficiencia en búsquedas en voz
  - Dos partes: indexado (offline) y búsquedas (online/offline)
  - Una vez generado el índice se puede buscar cualquier palabra, y la búsqueda es rápida



## Arquitectura de Sistemas Text-Based STD o SDR

- El ASR es un sistema Speech-To-Text general
- En lugar de conservar sólo la salida más probable se conserva todo un lattice con múltiples hipótesis
- Normalmente todo ello se compacta en un WFST (Weighted Finite State Transductor) para acelerar las búsquedas
- Problema: ¿Qué ocurre si se intenta buscar una palabra que no está en el léxico del reconocedor (OOV – Out-of-Vocabulary Word)?

## Arquitectura de Sistemas Text-Based STD o SDR

- El problema de las OOVs
  - Una OOV nunca va a aparecer en un lattice de reconocimiento
  - Por tanto, si no hacemos nada especial, nunca se encontrará
  - Y las OOVs suelen ser palabras importantes (ej. Nombres propios)
- Soluciones:
  - Sistemas híbridos: buscar las OOVs con un sistema similar pero basado en sub-unidades de palabra (fonemas, sílabas)
  - Sistemas basados en proxy-words: en lugar de buscar la OOV buscar secuencias de palabras del léxico que suenen parecidas a la OOV a buscar
  - En cualquier caso, el rendimiento con OOVs es muy inferior al rendimiento con palabras del léxico del reconocedor

## Comprendiendo el Search-on-Speech: Funcionamiento Interno

Arquitectura de Sistemas Search-on-Speech:  
Query-by-Example Spoken Term Detection (QbE-STD) o  
Spoken Document Retrieval (QbE-SDR)

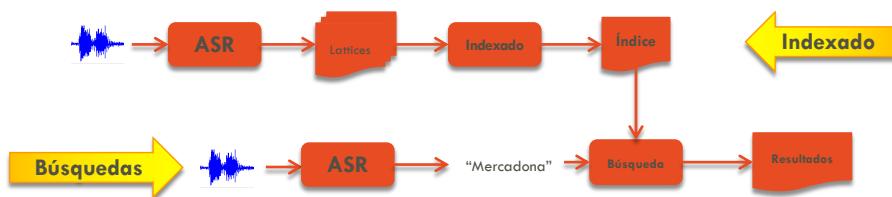
94

94

### Arquitectura de Sistemas QbE - STD o SDR



- La opción más sencilla es convertir la Query hablada en texto y usar un sistema text-based STD o SDR



## Arquitectura de Sistemas QbE STD o SDR

➤ **Ventajas:**

- Incorpora todo el conocimiento del lenguaje de un sistema STT
- Trivial si ya se tiene un sistema Text-based STD o SDR

➤ **Inconvenientes:**

- Acumula errores de reconocimiento
  - Un error de reconocimiento en la palabra a buscar es crítico
- Dependiente del idioma y del dominio (como los sistemas Text-Based STD o SDR)
- Existen otras aproximaciones más independientes del idioma.

## Arquitectura de Sistemas QbE STD o SDR

- Aproximación alternativa: Alineamiento Temporal Dinámico de Subsecuencias (Subsequence Dynamic Time Warping, S-DTW)
  - Transformamos los documentos de audio del repositorio en secuencias de vectores de parámetros
  - Transformamos la query en una secuencia (más corta) de vectores de parámetros
  - Buscamos la query en las secuencias de parámetros del repositorio
  - Devolvemos los resultados ordenados
- **Ventajas:**
  - Es independiente del idioma (si los parámetros lo son)
- **Inconvenientes:**
  - Al incluir menos información del idioma, también es menos preciso que los sistemas basados en un STT

## Contenidos

- **Audio y Voz en Big Data**
- **Speech-To-Text (STT) como usuario**
- **Search-on-Speech como usuario**
- **Comprendiendo el Speech-To-Text**
  - Funcionamiento interno
- **Comprendiendo el Search-on-Speech**
  - Funcionamiento interno
- **Conclusiones**

## Conclusiones

- Hemos revisado dos tecnologías (Speech-To-Text y Search-on-Speech) con múltiples aplicaciones en Big Data:
  - Big data sobre datos de call centers
  - Búsquedas en repositorios multimedia
  - Detección y control de menciones publicitarias contratadas
  - Medida de impacto mediático (políticos, deportistas, empresas)
- La mayor parte de ellas se basan en combinar distintas fuentes de información:
  - Parametrización de la señal de voz
  - Modelado acústico-fonético
  - Léxico
  - Modelo de Lenguaje

## Conclusiones

- Esos modelos dependen fuertemente de:
  - Idioma
  - Dialecto
  - Dominio de aplicación
  - Condiciones acústicas
- Todo ello hace la adaptación a la tarea imprescindible (al menos en Text-based):
  - Adaptación al idioma y dialecto (modelos acústicos, léxico y modelo de lenguaje)
  - Adaptación al dominio (léxico y modelo de lenguaje)
  - Adaptación a las condiciones acústicas (modelos acústicos)
- Una excepción a esta necesidad de adaptación son los sistemas Query-by-Example (QbE) basados en S-DTW
  - Consiguen cierta independencia al idioma
  - Pero a costa de resultados más limitados