UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

RECONOCIMIENTO DE IDIOMA EN VOZ ESPONTÁNEA MEDIANTE RECONOCIMIENTO FONÉTICO MULTILINGÜE EN PARALELO Y MODELADO ESTADÍSTICO DEL IDIOMA

-PROYECTO FIN DE CARRERA-

Alejandro Abejón González
Septiembre de 2007

ACTA DE EXAMEN

NOMBRE DEL ESTUDIANTE:

Alejandro Abejón González

TÍTULO DEL PROYECTO:

Reconocimiento de idioma en voz espontánea mediante reconocimiento fonético multilingüe en paralelo y modelado estadístico del idioma

NOMBRE DEL TUTOR:

Doroteo Torre Toledano

NOMBRE DE LOS MIEMBROS DEL TRIBUNAL:

Presidente: Joaquín González Rodríguez

Vocal: Alejandro Sierra Urrecho

Secretario: Doroteo Torre Toledano

Presidente suplente: Javier Ortega García

Vocal suplente: Ana María González Marcos

FECHA DE LECTURA Y DEFENSA:

Madrid, a de de 2007

CALIFICACIÓN OBTENIDA:

RECONOCIMIENTO DE IDIOMA EN VOZ ESPONTÁNEA MEDIANTE RECONOCIMIENTO FONÉTICO MULTILINGÜE EN PARALELO Y MODELADO ESTADÍSTICO DEL IDIOMA

AUTOR: Alejandro Abejón González
TUTOR: Doroteo Torre Toledano

Área de Tratamiento de Voz y Señales
Dpto. de Ingeniería Informática

Escuela Politécnica Superior
Universidad Autónoma de Madrid
Septiembre de 2007

PALABRAS CLAVE

Reconocedores fonéticos, Reconocimiento de idioma, Habla telefónica espontánea, Modelos Ocultos de Markov, PRLM. PPRLM, HTK, Sphinx , CallFriend, SpeechDat, NIST LRE

RESUMEN

En este proyecto se estudian los sistemas de reconocimiento de idioma en habla telefónica espontánea desde el nivel fonético del idioma. Se construye un sistema PPRLM (Parallel Phone Recognition followed by Language Modelling**)** desde el principio, creando reconocedores fonéticos mediante las bases de datos SpeechDat. Se aplicarán novedosas técnicas en reconocimiento idiomático a nivel fonético y se usarán dos importantes toolkit de tratamiento de voz como son HTK y Sphinx.

En la parte experimental se verán las bondades del sistema de reconocimiento de idioma creado viendo los efectos de algunas variables y la introducción de técnicas diferentes a las de un PPRLM clásico. Realizando varios tests de rendimiento con las bases de datos CallFriend y las evaluaciones internacionales de reconocimiento de idioma del National Institute of Standards and Technology (NIST) del 2003 y 2005, lo cual nos permite realizar una comparativa de nuestro sistema con otros sistemas a nivel internacional. Además el sistema creado las técnicas empleadas se presentará integrado con otros sistemas del grupo ATVS en la evaluación de idioma de NIST en Octubre del 2007.

ABSTRACT

In this project we study phonetic-level language recognition systems for telephone speech. We have built a PPRLM system ((Parallel Phone Recognition followed by Language Modelling**)** from scratch and the phonetic models were built with the database SpeechDat. We have used new techniques and configurations at phonetic level and we used two important speech toolkits: HTK and Sphinx.

Along the experimental part of the project we perform several test in order to analyze the system's behavior, which will be related with the value of some variables and different techniques of classic PPRLM. These tests have been done with database CallFriend and data from international evaluations of language recognition from National Institute of Standards and Technology (NIST) in 2003 and 2005, so we can compare our system on a international level. The new system and techniques will be integrated with other systems of ATVS group for next language evaluation of NIST in October of 2007.

Agradecimientos

Quiero agradecer en primer lugar a mi tutor Doroteo Torre Toledano, sin su apoyo y consejos no hubiera sido posible la realización de este proyecto, junto a él quiero agradecer a Joaquín González la oportunidad de participar en el grupo de investigación ATVS.

También tengo que agradecer la ayuda y apoyo que me han dado todos los miembros del grupo ATVS mientras he realizado este proyecto y en la actualidad., sin ellos hubiese sido muy difícil la realización de este trabajo.

Por último, agradecer a mis compañeros, a mis amigos y familia la compresión, apoyo y ayuda no sólo durante la realización de este proyecto, si no durante todos mis estudios.

Alejandro Abejón González
Septiembre de 2007

Este proyecto ha sido realizado en el Área de Tratamiento de Voz y Señales (ATVS) en la Escuela Politécnica superior de la Universidad Autónoma de Madrid. El proyecto ha sido financiado parcialmente por el Ministerio de Defensa y el Ministerio de Educación y Ciencia con el proyecto TEC2006-13170-C02-01.

Índice

Índice de Figuras

Índice de Tablas

Glosario

Curva DET (compensación por error de detección)
Trazo gráfico de las tasas de error medidas. Por lo general, las curvas DET trazan las tasas de error de decisión (tasa de falso rechazo vs. tasa de falsa aceptación).
DTW (Dynamic Time Warping)
Alineamiento Temporal Dinámico.
GMM (Gaussian Mixture Model)
Modelo de Mezclas Gaussianas
HMM (Hidden Markov Model)
Modelo oculto de Markov
HTK (Hidden Markov Model Toolkit)
Toolkit de creación y tratamiento de modelos HMM diseñado por la universidad de Cambridge (CUED).
MAP (Maximum a Posteriori)
Método de adaptación de modelos independientes de locutor a los distintos locutores.
MFCC (Mel Frequency Cepstral Coefficients)
Coeficientes cepstrales en escala de frecuencias Mel.
MLLR (Maximum Likelihood Linear Regression)
Método de adaptación de modelos independientes de locutor a los distintos locutores mediante transformaciones lineales.
NIST (National Institute of Standards and Technology)
Organismo federal, no regulador, perteneciente a la Cámara de Comercio de los Estados Unidos que desarrolla y promueve medidas, estándares y tecnología para aumentar la productividad, facilitar el comercio y mejorar la calidad de vida.
Phone-SVM
Sistema de reconocimiento de idioma que es semejante a un PPRLM pero que usa para extraer las puntuaciones un sistema SVMs sobre los n-gramas identificados.
PPR (Parallel Phone Recognition)
Sistema de reconocimiento de idioma que además usar un transcriptor fonético tiene un modelo del idioma a reconocer.
PPRLM (Parallel PRLM)
Sistema de reconocimiento de idioma construido con varios PRLM en paralelo
PRLM (Phone Recognition followed by Language Modelling)
Reconocimiento de idioma mediante modelos fonéticos de los mismos
Reconocimiento del habla
Tecnología que permite que una máquina reconozca las palabras pronunciadas. El reconocimiento del habla no es una tecnología biométrica.
Reconocimiento de locutor
Modalidad biométrica que utiliza el habla de una persona, una característica influenciada tanto por la estructura física del tracto vocal del individuo como por las características de comportamiento del individuo, para fines de reconocimiento. Se divide en identificación y verificación de locutor.
Reconocimiento de idioma
Tecnología empleada en el indexado de contenidos multimedia, enrutamiento en servicios de atención telefónica y configuración de sistema. Consiste en determinar ele idioma del los interlocutores.
ROC (Característica de funcionamiento del receptor)

Método para mostrar el rendimiento de precisión medida de un sistema biométrico. La característica ROC en una verificación compara la tasa de falsa aceptación con la tasa de verificación.

Sphinx

Toolkit de tratamiento de voz diseñado por la universidad Carnegie-Mellon.

SVMs (Support Vector Machines)

Método de reconocimiento de patrones.

Tasa de falsa aceptación (FAR)

Estadística utilizada para medir el rendimiento biométrico durante la tarea de verificación. Porcentaje de veces que un sistema produce una falsa aceptación, lo cual ocurre cuando un individuo es erróneamente vinculado con la información biométrica existente de otra persona.

Tasa de falso rechazo (FRR)

Estadística utilizada para medir el rendimiento biométrico durante la tarea de verificación. Porcentaje de veces que el sistema produce un falso rechazo. Ocurre un falso rechazo cuando un individuo no es vinculado con su propia plantilla biométrica existente.

Tasa de igual error (EER)

Estadística utilizada para mostrar el rendimiento biométrico; por lo general, durante la tarea de verificación. La tasa EER es la ubicación en una curva ROC o DET donde la tasa de falsa aceptación y la tasa de falso rechazo son iguales. Por lo general, cuánto más bajo sea el valor de la tasa de igual error, mayor será la precisión del sistema biométrico. Observe, sin embargo, que la mayoría de los sistemas operativos no están preparados para funcionar con la "tasa de igual error", de modo que la verdadera utilidad de esta medida está limitada a la comparación con el rendimiento del sistema biométrico.

Umbral

Valor predeterminado de un usuario para las tareas de verificación o identificación de grupo abierto en los sistemas biométricos. La aceptación o el rechazo de los datos biométricos dependen de si el resultado de coincidencia se encuentra por encima o por debajo de la escala. La escala es ajustable de modo que el sistema biométrico puede ser más o menos estricto según los requisitos de cada aplicación biométrica.

1. INTRODUCCIÓN

1.1. MOTIVACIÓN

El mundo de las comunicaciones ha cambiado mucho, durante los últimos años hemos pasado de un mundo analógico a un mundo digital. Apareciendo nuevas aplicaciones, fundamentalmente multimedia. Todo esto ha hecho necesaria la aparición de técnicas de clasificación de contenidos, tomando una gran importancia los modelos de reconocimiento de patrones, que es en lo que se va a basar principalmente este proyecto, concretamente en el reconocimiento del idioma.

El reconocimiento de idioma es un factor importante en esta clasificación de contenidos así como en procesos de mejora de interfaz con el usuario, ya que se adaptará el contenido en función de ello.

El reconocimiento de idioma comenzó en texto, pero como hemos indicado, la creciente utilización de contenidos multimedia ha hecho necesaria la creación de sistemas de clasificación por lengua, motivando la creación de esta línea de investigación que cada vez está suscitando un mayor interés.

Además muchas de las técnicas empleadas en el reconocimiento de idioma son empleadas en otros tipos de reconocimiento que tienen como protagonista la voz, porque la voz lleva una gran cantidad de información: identidad del locutor, idioma, edad, estado de ánimo, nivel de educación, etc. Por ello el análisis de la voz para cualquiera de estas características posibilita nuevas técnicas para la identificación de otras.

Pero los sistemas de reconocimiento de idioma sobre habla espontánea tienen ciertas limitaciones como son el ruido de las conversaciones y los silencios en las mismas. Además cuanto mayor sea el nivel de reconocimiento requerido mayor tendrá que ser la duración de la conversación.

1.2. OBJETIVOS Y ENFOQUE

El presente proyecto se centra en la problemática de la identificación idiomática en habla conversacional telefónica a nivel fonético, como se ha explicado con anterioridad el desarrollo multimedia ha hecho que los sistemas de detección de idioma en voz espontánea adquieran una labor mucha importancia.

Puesto que el reconocimiento de idioma se hace a nivel de fonético, se hace necesario un estudio del reconocimiento fonético. Además el reconocimiento fonético se puede emplear en diversas técnicas de reconocimiento tanto de locutor como de trascripción automática. En esta memoria se hará una primera introducción al estado del arte y técnicas usadas tanto en reconocimiento fonético (2.2) prestando especial atención al método usado de hidden Markov model (HMM); como de reconocimiento de idioma (2.3.) dando explicaciones de las actuales técnicas empleadas en dicha labor, a la vez que mostramos el sistema de las evaluaciones internacionales de NIST.

Posteriormente describiremos los medios empleados: bases de datos, software y hardware. También se explicará el diseño llevado a cabo para realizar los reconocedores fonéticos de HMM, así como la creación de un sistema PPRLM para la identificación de idioma en habla telefónica espontánea. En la sección 4 mostraremos los resultados tanto de los reconocedores fonéticos, como del sistema PPRLM mostrando diversas técnicas para la mejora de los resultados de identificación de idioma.

Por último, extraemos una serie de conclusiones tanto para los reconocedores fonéticos como para el sistema PPRLM, sobre como influyen diversos factores en los resultados. Además en el caso del sistema PPRLM hacemos una comparativa con otros sistemas desarrollados con anterioridad en el grupo en donde se ha realizado este proyecto (ATVS), y a nivel internacional comparando con los resultados que se obtuvieron en las evaluaciones NIST de 2005 (última evaluación realizada en el momento de realizar esta memoria)

Otro objetivo del nuevo sistema además de mejorar los resultados es hacer un cambio de herramientas empleadas, se pasa de la herramienta HTK (una de las más importantes herramientas en el tratamiento de voz) a Sphinx y herramientas propias del grupo ATVS, con esto posibilitamos nuevas líneas futuras de investigación como se muestra en la sección 6

2. ESTUDIO DEL ESTADO DEL ARTE Y TECNOLOGÍA A UTILIZAR

2.1. INTRODUCCIÓN

Se va a desarrollar un sistema de clasificación de sonidos para la creación de un transcriptor automático de fonemas. Los modelos fonéticos no son más que una particularización de reconocimiento de patrones. Reconocimiento que está adaptado a las características de la voz y en particular al nivel del habla que nos referimos.

La creación de dichos transcriptores automáticos es vital para diversas técnicas, como pueden ser la identificación de locutor dependiente de texto, o como base para reconocedores de alto nivel ya sea para identificación de locutor o de idioma. El reconocimiento de fonemas es la base sobre la que se cimientan las técnicas más empleadas: PRLM y PPRLM que constituyen una parte fundamental de este proyecto, y que son a la vez otras técnicas de reconocimiento de patrones pero a otro nivel.

2.2. ESTADO DEL ARTE EN RECONOCIMIENTO FONÉTICO

2.2.1. INTRODUCCIÓN

Antes de comenzar a explicar el estado del arte de las distintas técnicas de reconocimiento fonético, vamos a describir que es un fonema y cuales son sus características ya que nos ayudará a una mejor comprensión de las técnicas empleadas para su identificación.

Los fonemas son unidades teóricas, postuladas para estudiar el nivel fonético-fonológico de una lengua humana. Entre los criterios para decidir, qué constituye o no un fonema se requiere que exista una función distintiva: son sonidos del habla que permiten distinguir palabras en una lengua. Así, los sonidos /p/ y /b/ son fonemas del español porque existen palabras como /pata/ y /bata/ que tienen significado distinto y su pronunciación sólo difiere en relación con esos dos sonidos (sin embargo en chino los sonidos [p] y [b] son percibidos como variantes posiciónales del mismo fonema). Esto se puede estudiar con más profundidad en Gil y Juana (1989), Llisterri y joaquin (1991) y Trubetzkoy (1939)

Desde un punto de vista estructural, el fonema pertenece a la lengua, mientras que el sonido pertenece al habla. La palabra <casa>, por ejemplo, consta de cuatro fonemas (/k/, /a/, /s/, /a/). A esta misma palabra también corresponden en el habla, acto concreto, cuatro sonidos, a los que la fonología denominará alófonos, y estos últimos pueden variar según el sujeto que lo pronuncie. La distinción fundamental de los conceptos fonema y alófono, está en que el primero es una huella psíquica de la neutralización de los segundos que se efectúan en el habla.

Los fonemas no son sonidos con entidad física, sino abstracciones mentales o abstracciones formales de los sonidos del habla. En este sentido, un fonema puede ser representado por una familia o clase de equivalencia de sonidos (técnicamente denominados fonos), que los hablantes asocian a un sonido específico durante la producción o la percepción del habla. Así por ejemplo, en español el fonema /d/ [+ obstrúyente, + alveolar, + sonoro] puede ser articulado como oclusiva [d] a principio de palabra o tras nasal o pausa larga, pero es pronunciado como aproximante [ð] entre vocales o entre vocal y líquida, así /dedo/ se pronuncia [deðo] donde el primer y tercer sonido difieren en el grado de obstrucción aunque son similares en una serie de rasgos (los propios del fonema).

2.2.1.1. Fono y fonema

Un sonido o fono se caracteriza por una serie de rasgos fonéticos y articulatorios, el número de dichos rasgos y la identificación de los mismos es tarea de la fonética. Un fono es cualquiera de las posibles realizaciones acústicas de un fonema.

La fonología en cambio no necesariamente trata entes claramente distinguibles en términos acústicos. Como realidad mental o abstracta, un fonema no tiene porqué tener todos los rasgos fonéticos especificados. Por ejemplo, en diversas lenguas la aspiración es relevante para distinguir pares mínimos, pero un fonema del español puede pronunciarse más o menos aspirado según el contexto y la variante lingüística del hablante pero en general no está especificado el grado de aspiración. En cambio, en lenguas como el chino mandarín o el coreano un fonema tiene predefinido el rasgo de aspiración.

El número de fonemas de una lengua es finito y limitado en cada lengua al número de alófonos potencialmente definibles, si especificamos rasgos fonéticos muy sutiles, es potencialmente ilimitado y varían según el contexto fonético y la articulación individual de los hablantes. En cuanto al número de fonemas no tiene porqué ser fijo, pudiendo cambiar con el número de especificaciones que se dé para cada fonema. Sin embargo, la mayoría de los análisis del español está en torno a 24 unidades (5 vocales y 19 consonantes), aunque no todas las variedades de español tienen el mismo número de fonemas. Por el contrario, hay otras lenguas como el ruso que llegan a 48 fonemas.

Dada la distinción entre fonema y fono, existe otra forma de concebir un fonema como una especificación incompleta de rasgos fonéticos. Esta relación es de hecho equivalente a la del fonema como conjunto de fonos: el fonema sería el conjunto de rasgos fonéticos comunes a todos los fonos que forman la clase de equivalencia del fonema.

Fijado un conjunto de rasgos fonéticos se pueden definir los sonidos de la lengua. En principio no hay límite a lo fina que pueda ser la distinción que establecen estos rasgos. Potencialmente la lista de sonidos puede hacerse tan grande como se quiera si se incluyen más y más rasgos. Sin embargo el número de fonemas es un asunto diferente, puesto que muchos de los anteriores sonidos serán equivalentes desde el punto de vista lingüístico. Un sistema fonológico es un par $\mathcal{F} = (F, (R))$ donde F es un inventario de fonemas abstractos definidos por unos pocos rasgos del conjunto total (las lenguas

naturales oscilan entre 1 o 2 decenas hasta 4 o 5 decenas de fonemas), y $\mathcal{R}$ es el conjunto de reglas que en función del contexto relativo de aparición de los fonemas definen totalmente los rasgos fonéticos, así el conjunto de reglas puede pensarse como una aplicación del conjunto de secuencias admisibles de fonemas al conjunto de secuencias admisibles de sonidos:

$$\mathcal{R} : P_0(F) \rightarrow P_0(S)$$

Donde $P_0(F), P_0(S)$ representan el conjunto de secuencias finitas de fonemas y el conjunto de secuencias finitas de sonidos.

2.2.1.2. Características

Podemos decir que fonema es una unidad fonológica diferenciadora, indivisible y abstracta.

- **Diferenciadora**: porque cada fonema se delimita dentro del sistema por las cualidades que se distinguen de los demás y además es portador de una intención significativa especial. Por ejemplo, /k-o-t-a/ y /b-o-t-a/ son dos palabras que se distinguen semánticamente debido a que /k/ se opone a /b/ por la sonoridad.

- **Indivisible**: no se puede descomponer en unidades menores. Por ejemplo, la sílaba o el grupo fónico sí pueden fraccionarse. Un análisis pormenorizado del fonema revela que está compuesto por un haz de diversos elementos fónicos llamados rasgos distintivos, cuya combinación forma el inventario de fonemas. El inventario de rasgos distintivos es asimismo limitado y viene a constituir una especie de tercera articulación del lenguaje.

- **Abstracta**: no son sonidos, sino modelos o tipos ideales de sonidos. La distinción entre sonido y fonema ha sido un gran logro en los últimos tiempos en la lingüística.

2.2.2. CREACIÓN DE MODELOS FONÉTICOS

Los modelos fonéticos se crean en dos fases: la primera es la extracción de características de la señal de voz, y la segunda es usar esa característica para identificar los fonemas. Para ello el proceso se divide en varias fases que son las siguientes:

- **Extracción** de las características del sonido del habla. Por semejanza con el funcionamiento del sistema humano la extracción de esas características, que llamaremos parámetros, se realiza en el dominio de la frecuencia. En particular la parametrización que se ha realizado es de tipo Mel-Frequency Central Coefficients (MFCC), la cual se explicará con más profundidad en la siguiente sección.

- **Entrenamiento y reconocimiento** de modelos para cada fonema a identificar. A partir de la extracción de parámetros se construirán una serie de modelos estadísticos con los cuales se identificarán con cierta probabilidad, fonemas en otras locuciones. Por ello, se mide la distancia entre el modelo (conjunto de parámetros que constituye el modelo) y los parámetros de la pronunciación a reconocer. Hay varias técnicas para realizar el proceso

    - **HMM (hidden Markov model)** que se basa en la creación de modelos de fonemas en estados. Éste es el procedimiento seguido, por ello se explicará con más detenimiento en la sección 2.2.4 de este proyecto

    - **DTW** (Alineamiento temporal Dinámico) consiste en alinear de forma temporal los parámetros del archivo de test y los parámetros de los modelos, obteniendo la función que alinea a ambos, eligiendo la función de menor coste posible para dicha adaptación. En la siguiente imagen (Fig. 1) se ve como representar la función de adaptación.

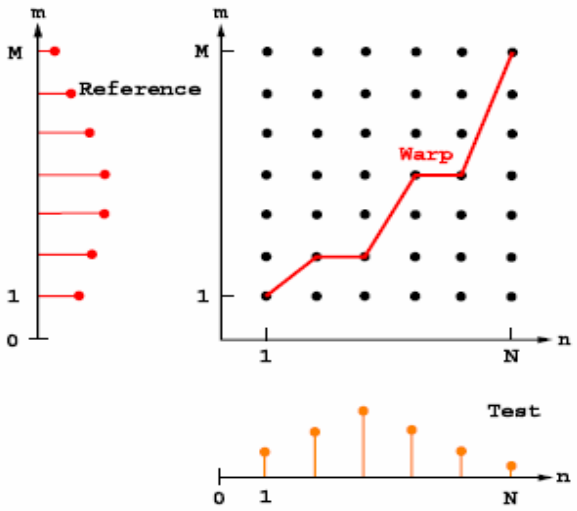

Fig. 1 Función de adaptación de DTW

    - **VQ** (Cuantificación vectorial) [R.O. Duda, 2001] consiste en representar las características de los fonemas como un espacio vectorial de dimensión el número de parámetros. De forma que al fonema a reconocer se le asigna el vector cuya distancia a él sea mínima. Por tanto, los fonemas quedarán representados por unos vectores determinados (centroides) de forma que todos los puntos que caigan en una zona determinada se asignarán a dicho vector. Esto se puede ver por ejemplo en la Fig. 2 en la que el espacio es bidimensional (el número de parámetros que se emplean son dos) y en el que los puntos verdes son los vectores de test, mientras que los rojos son los vectores a los que se asignan (obtenidos de forma óptima durante el entrenamiento), siendo cada una de las regiones los fonemas posibles

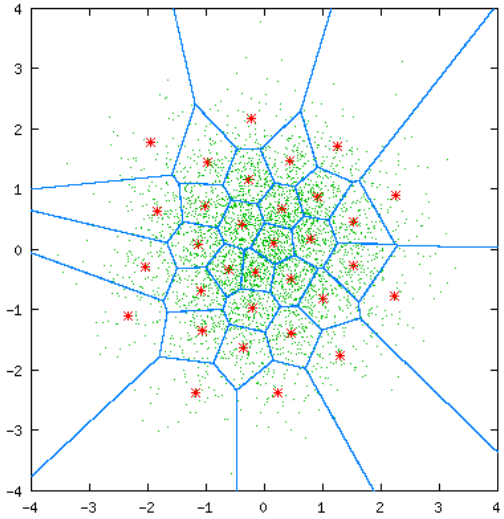

Fig. 2 VQ bidimensional

Este sistema tiene los siguientes beneficios:

- ✓ **Reducción drástica de la capacidad de almacenamiento** para obtener la información del análisis espectral.
- ✓ **Reducción de la complejidad** computacional de distancias.
- ✓ **Representación discreta** de los "sonidos" de voz

Por contra, este sistema tiene lo siguiente desventajas:

- × **Introducción de una distorsión espectral** al representar cada vector espectral por un representante (error de cuantificación). Disminuye si aumentamos el número de centroides (aumento del codebook)
- × **Aumento de la complejidad de cálculo** al vector más próximo con él, según incrementamos el tamaño del coodebook
- × **Problemas de almacenamiento** conforme aumentamos el tamaño del codebook.

2.2.3. MEL-FREQUENCY CEPSTRAL COEFFICIENTS (MFCC)

Esta es la etapa en la que se transforma el audio en una serie de parámetros que representan de forma compacta la información del sonido. Es la extracción de un conjunto de características que se emplean para el reconocimiento de aquello que deseamos. La parametrización es igual para los datos de entrenamiento y para los datos de evaluación.

A imitación de lo que sucede en el aparato auditivo humano, la identificación de los sonidos se hace en el dominio de la frecuencia. Entre las muchas técnicas de parametrización del habla, la más empleada es la de los MFCC. La Fig. 3 muestra el esquema de la extracción de los MFCC.

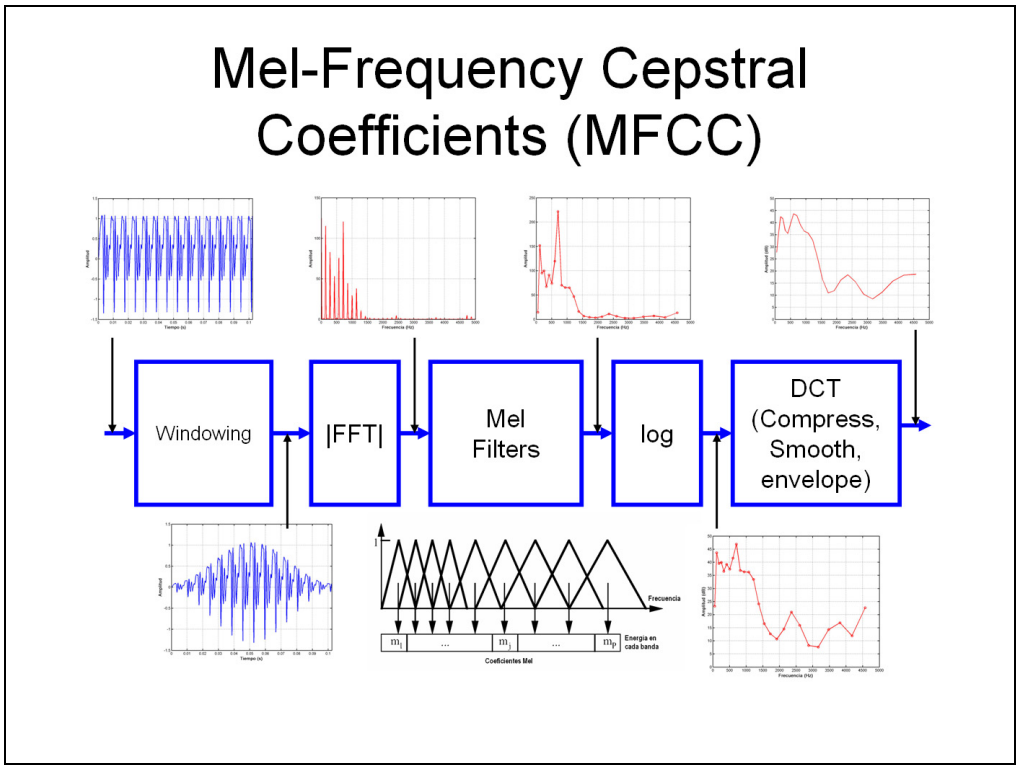

Fig. 3 Proceso de extracción de los Mel-Frequency Cepstral Coefficients (MFCC)

El proceso que se sigue se puede resumir en los siguientes pasos:

1. **Enventanar** la señal en segmentos de 25 ms, con un desplazamiento de 10 ms entre las ventanas

2. **Pasar al dominio de la frecuencia** por medio de la FFT. Después se aplica la señal a un banco de filtros de forma semejante a lo que se hace en el oído humano, concretamente en el oído interno. Con esto conseguimos mayor resolución en bajas frecuencias. El número de filtros que se suele emplear es aproximadamente 40.

3. **Calcular la energía** en cada uno de los filtros, quedando tantos coeficientes como filtros se han utilizado.

4. **Hacer el logaritmo y hacer la DCT,** quedando reducido el número de parámetros a 13 habitualmente. (en Sphinx, herramienta que usamos para la parametrización durante todo el proyecto es de 13).

Los MFCC únicamente representan la envolvente espectral de la señal de voz, obteniendo importantes características identificadoras en el habla. De entre los coeficientes destaca el primero, que es C_0 e indica la energía de la señal.

Para aumentar la información, como por ejemplo la de coarticulación de fonemas, es necesario introducir datos de la velocidad y aceleración de los parámetros. Por ello surgen los MFFC-Delta y los MFCC-Delta-Delta

Los MFCC-Delta se calculan como la variación de los coeficientes MFCC con respecto a un instante de tiempo. Por ello son denominados coeficientes de velocidad (ya que dan los cambios por tiempo) o de primera derivada (Fig. 4). También se suelen emplear los coeficientes MFCC Delta-Delta, estos son llamados coeficientes de aceleración y miden la velocidad de cambio de los MFCC Delta

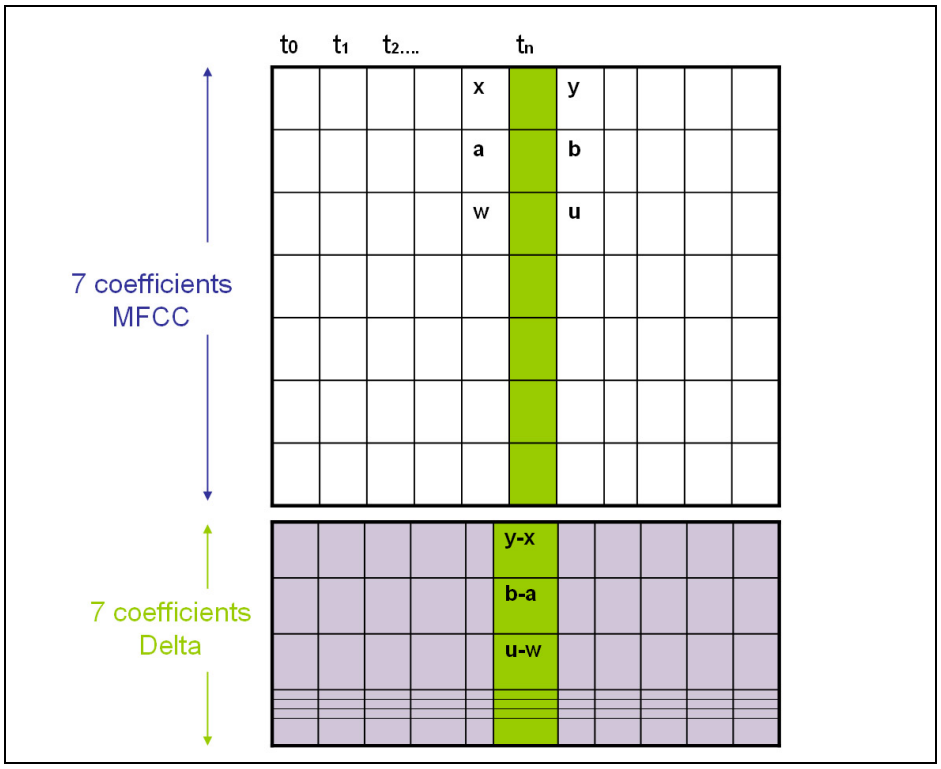

Fig. 4 Una esquematización de los Delta- Mel-Frequency Cepstral Coefficients donde se representa una posible manera de calcular los coeficientes delta.

La parametrización empleada durante todo el proyecto fue realizada mediante dos herramientas de reconocimiento de voz muy conocidas como son Sphinx y HTK, en concreto se hace una parametrización con sphinx y luego se pasa al formato de parámetros de HTK (Fig. 5). La parametrización de HTK es configurable pero la que usamos es de 13 MFCC, 13 –Deltas y 13 Deltas-Deltas

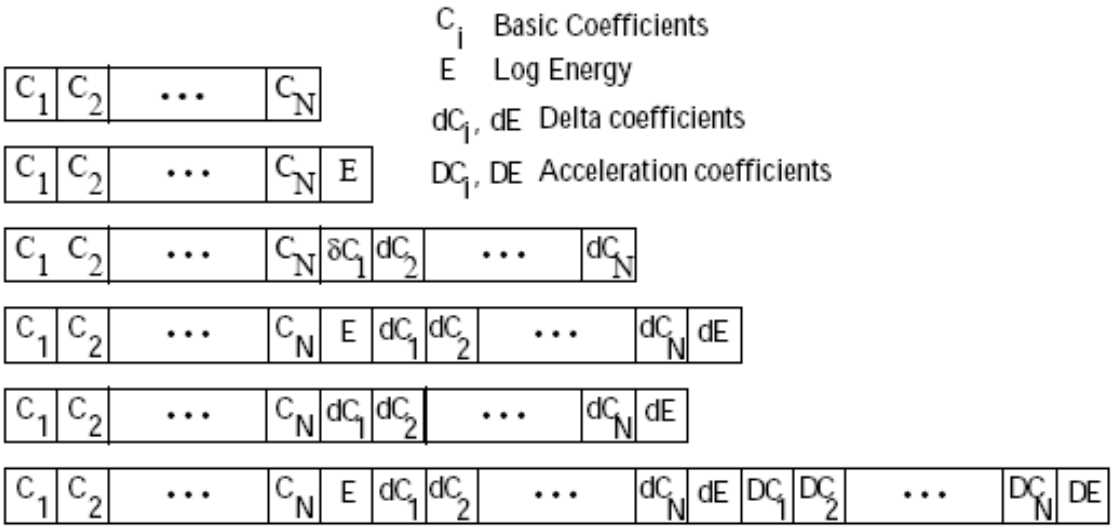

Fig. 5 Organización de coeficientes de un fichero de parámetros de HTK

Por el contrario, los ficheros de parámetros de Sphinx únicamente llevan los 13 coeficientes MFCC, el resto de coeficientes se generan de forma indirecta durante la ejecución del programa.

Existen otras técnicas de parametrización que se usan en reconocimiento de idioma con SVM y GMM pero que no se utilizarán en este proyecto. Una de esas técnicas es SDC (Shifted Delta Cepstrum), que consiste en realizar una combinación semejante a la hecha en los coeficientes MFCC-Delta pero con un mayor número de coeficientes. Dando un mayor cantidad de información sobre la evolución de los sistemas.

$$\Delta c_n(t,i) = c_n(t+iP+d) - c_n(t+iP-d) \qquad (1)$$
$$n = 0, N-1 \qquad i = 0, k-1$$

Donde:
N es el número de coeficientes que se quieren calcular;
d es la distancia que se avanza o retrasa alrededor de cada ventana de referencia;
P: el desplazamiento de cada ventana de referencia con respecto a la anterior;
k: el número de coeficientes a considerar;

En este caso, en un vector de características relativo a un instante t, se tienen en cuenta también esas variaciones alrededor de los instantes que distan P, 2xP,.., kxP muestras sucesivas como se puede ver en Fig. 6. Estos coeficientes, como hemos comentado anteriormente, pretenden tener en cuenta informaciones de coarticulación entre fonemas.

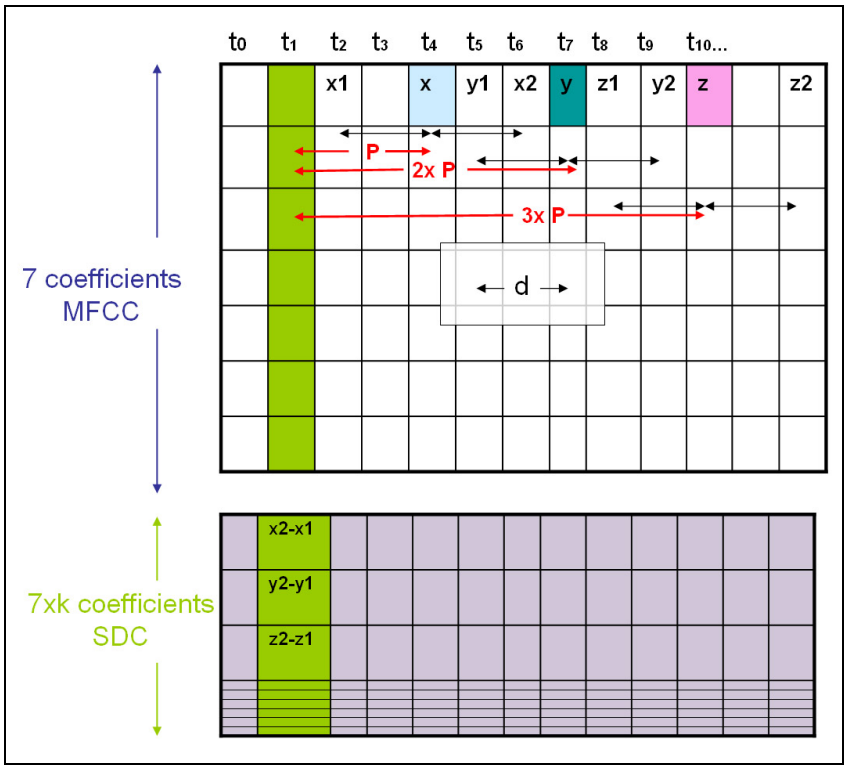

Fig. 6 Creación de los SDC

2.2.4. HMM

2.2.4.1. Introducción

Los hidden Markov models (HMM) son una máquina de estados finita, en la que las observaciones son una función probabilística del estado, siendo un proceso doblemente estocástico. Los HMM pueden ser considerados como una red dinámica bayesiana.

En un modelo observable de Markov, el estado es lo que es directamente visible, por lo tanto, los únicos parámetros que existen son las probabilidades de transición entre estados. Por el contrario, en un HMM el estado no es visible directamente sino que solo son visibles las variables influenciadas por el estado. Cada estado tiene una distribución de probabilidad sobre el símbolo a la salida, en nuestro caso la variable observable la consideramos continua, por tanto, empleamos una función de densidad de probabilidad continua modelada como una mezcla de Gaussinas. Esto es lo que se denomina Continuous–density HMM o CDHMM

Un ejemplo de cadena de Markov observable no relacionado con el habla pero que explica las diferencias entre un HMM y un Modelo de Markov observable es el siguiente:

La cotización en el DowJones, en la que cada estado marca si la cotización ha subido, bajado o no ha cambiado con respecto al día anterior, la representación de este modelo sería como sigue:

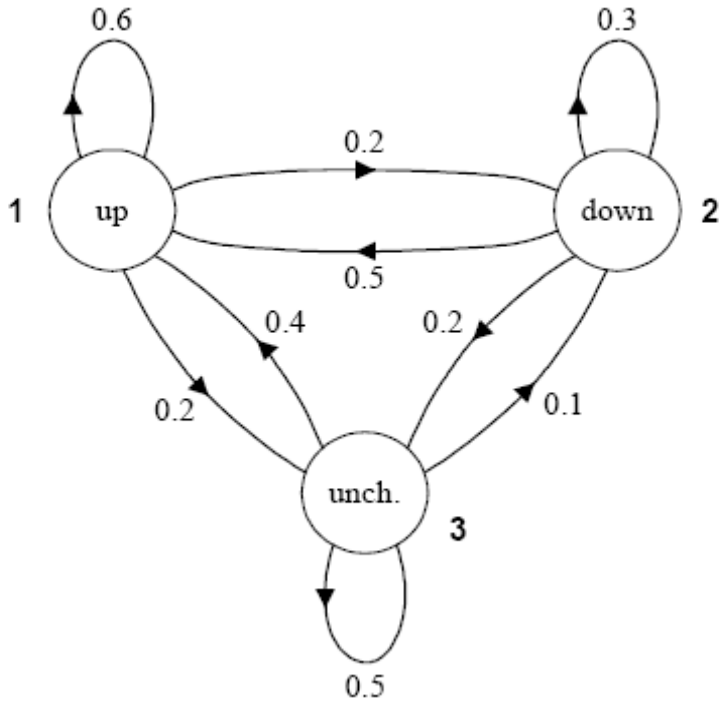

Fig. 7 Esquema de un modelo de Makov observable [Huang et al 2001]

A este gráfico habría que añadir las probabilidades de comienzo en cada uno de los estados. Como se puede ver en este esquema la salida observable es determinista para cada estado, por tanto se sabe en que estado del modelo se está en ese momento.

Por el contrario un modelo de HMM también del Dow Jones nos daría el siguiente esquema Fig. 8, en el que ahora cada uno de los estados es un mercado concreto mientras que las salidas son si suben o bajan o no cambian. Es decir, ahora la salida de cada estado se asigna mediante un proceso estocástico.

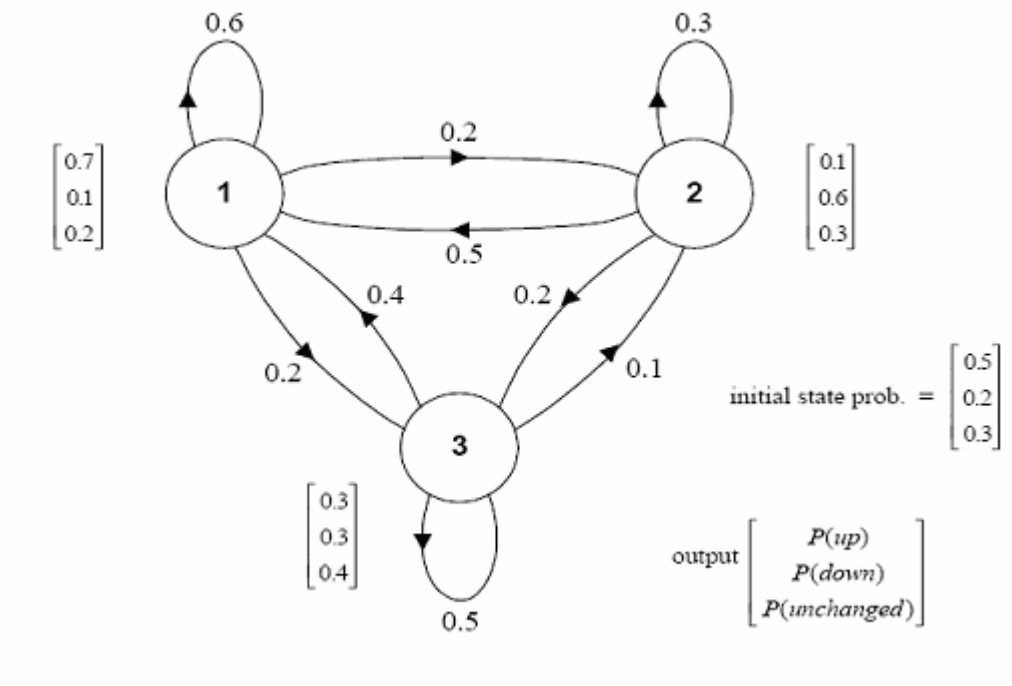

Fig. 8 Esquema de un HMM [Huang et al 2001]

La técnica de HMM se usa en la actualidad en aquellos sistemas en los que el modelado tiene una dependencia del tiempo como pueden ser los sistemas reconocimiento fonético y del habla en general.

Una razón, por la que los HMMs se utiliza en el reconocimiento de fonemas, es que una señal de voz se puede ver como una señal invariante a corto plazo, es decir, uno podía asumir en un corto plazo de unos 10 -20 milisegundos de voz, que se pueda aproximar como un proceso invariante. La voz se podría interpretar así como un modelo de Markov para muchos procesos estocásticos (conocidos como **estados**).

Otra razón por la que los HMMs son populares, es porque pueden ser entrenados automáticamente, siendo factible realizar los cálculos en un tiempo razonable. El reconocimiento fonético es la disposición más simple posible. El modelo oculto de Markov tendrá en cada estado una distribución estadística llamada mezcla de Gaussianas de matriz de covarianza diagonal, que dé una probabilidad para cada vector observado. Cada fonema tendrá una distribución de salida. Un modelo oculto de Markov para una secuencia de fonemas se construye concatenando los modelos ocultos entrenados para los fonemas separados.

El uso de los HMM permite eludir las limitaciones de algunos otros sistemas en el reconocimiento de fonemas como son los siguientes:

- DTW (Alineamiento temporal Dinámico) no hay posibilidad de realizar un entrenamiento estadístico, ya que se realiza comparaciones entre secuencias de vectores de parámetros

- VQ (Cuantificación temporal) asignación dura entre los vectores y la clase que modela. Además tiene que respetar el compromiso entre el tamaño del codebook y el error de cuantificación.

2.2.4.2. Elementos de un HMM

Supongamos un HMM discreto en que las observaciones posibles pertenecen a un conjunto discreto, entonces el HMM vendrá dado por

- N: el número de estados del modelo, donde q_t denota el estado en el instante de tiempo t. Los HMMs que habituales están compuestos por 5 estados. pero tanto el estado 1 como el estado 5 no generan ninguna salida.

  $S=\{s1,s2,...,sN\}$ (2)

- La dimensión del conjunto de observaciones distintas de salida M, es decir el tamaño del alfabeto
  $V=\{v_1,v_2,...,v_M\}$ (3)

- La distribución de probabilidad de transición entre estados $A=\{aij\}$:
  $aij=P(qt=sj \mid qt-1=si)\ 1\leq i,j \leq N$ (4)

- La distribución de probabilidades de emisión de símbolos entre estados
  $B = \{b_j(k)\}$:
  $b_j(O_k)=P(O_k \mid q_t = sj)\ 1\leq j \leq N,\ 1\leq k \leq M$, donde O_k es un símbolo perteneciente a V.

- Distribución del estado inicial $\pi = \{\pi_i\}$:
  $\pi_i = P(q_0 = si)\ 1\leq i \leq N$ (5)

Con todo esto, un HMM se describe como $\lambda = \{A, B, \pi\}$.

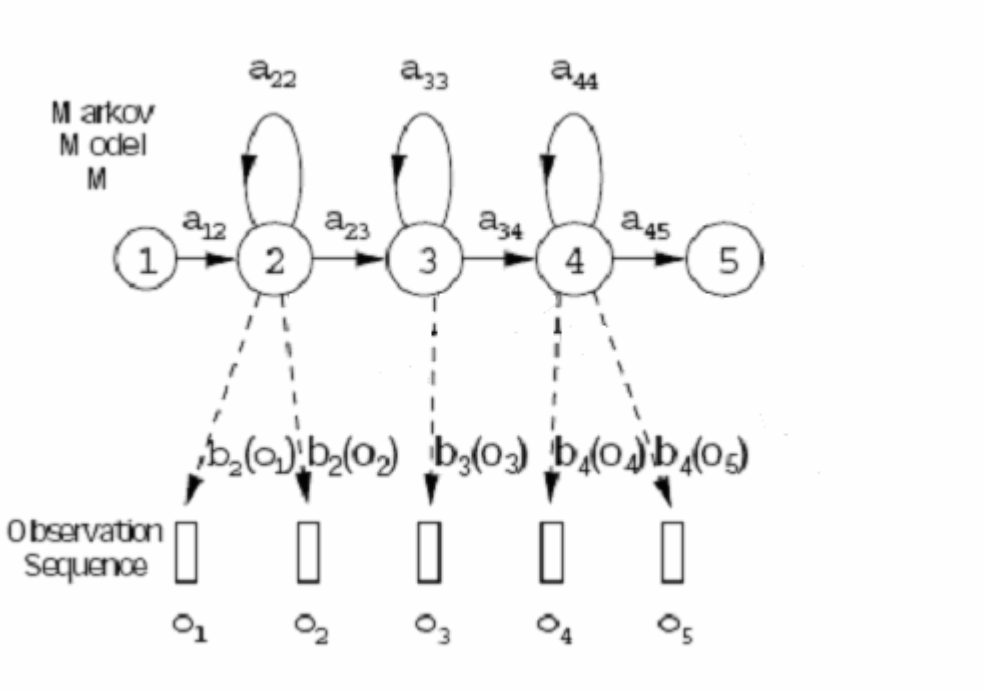

Fig. 9 Esquema general de un HMM de modelo fonético

2.2.4.3. Problemas a resolver para utilizar los HMMs

Los tres problemas que hay que resolver para que los HMM sean útiles son:

1. **Problema de Evaluación**: dada una secuencia de observaciones O={ o_1, o_2, ... o_T} (siendo T la longitud de la secuencia de observación) y el modelo λ={ A, B, Π}, el problema es cómo obtener de forma eficiente P(O|λ), es decir, la probabilidad de obtener una secuencia de observación dado un modelo determinado.

2. **Problema de Decodificación**: dada una secuencia de observaciones O={ o_1, o_2, ... o_T} y el modelo λ={ A, B, Π}, encontrar una secuencia de estados Q={ q_1, q_2, ... q_T) más probable, para la secuencia de observaciones dada.

3. **Problema de aprendizaje:** Como maximizar los parámetros del modelo λ para obtener la máxima P(O| λ) para unas observaciones de entrenamiento O.

Si solucionamos el problema de evaluación, se podría evaluar como de bueno es un modelo HMM para una secuencia de observación. Además podríamos usarlo para hacer reconocimiento de patrones ya que la probabilidad P(O|λ) determina la probabilidad de observación. Si solucionamos el problema de decodificación podremos saber la secuencia de estados óptima para una secuencia de observación. En otras palabras, descubriríamos la secuencia oculta de estados. Por último la solución del problema de aprendizaje nos daría los parámetros de un modelo λ dado una serie de datos de entrenamiento.

Evaluación de HMM – Algoritmo Forward -backward

Para el cálculo de la probabilidad P(O|λ), lo que resulta más intuitivo es el cálculo como la suma de las probabilidades de todas las secuencias de estados:

$$P(O \mid \lambda) = \sum P(O \mid q, \lambda) P(q \mid \lambda) \qquad (6)$$

En otras palabras, enumerar todas las posibles secuencias de estados de longitud T que generen la secuencia de observación O y sumando sus probabilidades según el teorema de la Probabilidad Total

Para ello consideremos una determinada secuencia de estados: $Q=(q_1,q_2,...,q_T)$ donde q_1 es el estado inicial. La probabilidad de la secuencia de observación O dada la secuencia de estados Q es:

$$P(O \mid Q, \lambda) = \prod_{t=1}^{T} P(O_t \mid q_t, \lambda) \qquad (7)$$

Donde se asume independencia estadística de las observaciones. Por lo tanto se obtiene:

$$P(O \mid Q, \lambda) = b_{q1}(O_1) \cdot b_{q2}(O_2) \cdots b_{qT}(O_T) \quad (8)$$

Por otra parte la probabilidad de la secuencia de estados Q se puede expresar como:

$$P(Q \mid \lambda) = \pi_{q1} \cdot a_{q1q2} \cdot a_{q2q3} \cdots a_{qT-1qT} \qquad (9)$$

Que se interpreta como la probabilidad del estado inicial, multiplicada por las probabilidades de transición de un estado a otro.

Sustituyendo los dos términos anteriores en el sumatorio inicial (6) se obtiene la probabilidad de la secuencia de observación:

$$P(O \mid \lambda) = \sum_{Q} P(O \mid Q, \lambda) \cdot P(Q \mid \lambda) = \sum_{q1,q2,...,qT} \pi_{q1} \cdot b_{q1}(O_1) \cdot a_{q1q2} \cdot b_{q2}(O_2) \cdots a_{qT-1qT} \cdot b_{qT}(O_T) \qquad (10)$$

La interpretación del resultado obtenido es la siguiente: inicialmente en el tiempo t=1 nos encontramos en el estado q_1 con probabilidad π_{q1} y generamos el símbolo O_1 con probabilidad $b_{q1}(O_1)$. Al avanzar el reloj al instante t=2 se produce una transición al estado q_2 con probabilidad a_{q1q2} y generamos el símbolo O_2 con probabilidad $b_{q2}(O_2)$. Este proceso se repite hasta que se produce la última transición del estado q_{T-1} al estado q_T con probabilidad a_{qT-1qT} y generamos el símbolo O_T con probabilidad $b_{qT}(O_T)$.

Sin embargo, una primera aproximación al número de operaciones necesarias para calcular P(O|λ) nos da un orden $2TN^T$ operaciones, ya que, para cada T se pueden alcanzar N^T posibles secuencias de estados, haciendo que el problema sea intratable incluso para pequeños valores.

Por fortuna, existe un algoritmo distinto del que antes se ha expuesto, que utiliza los cálculos intermedios para realizar posteriores operaciones de forma que se reducen el número de operaciones. Pasando a ser del $O(TN^2)$, el algoritmo consiste en los siguientes pasos:

1. Inicialización

$$\alpha_1(i) = \pi_i b_i(o_i) \quad 1 \le i \le N \qquad (11)$$

En este paso se inicializan las probabilidades hacia delante como la probabilidad conjunta del estado i y de la observación o_1.

2. Recursión inductiva

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i).a_{ij}\right] b_j(o_{t+1}) \qquad (12)$$

Este paso también denominado paso de inducción, muestra cómo es posible alcanzar el estado j en el instante t+1 desde los N posibles estados en el instante anterior t. Puesto que $\alpha_t(i)$ es la probabilidad conjunta de observar el evento o_1, o_2,...., o_t y de que el estado en el instante t sea i, el producto $\alpha_t(i)a_{ij}$ es la probabilidad conjunta de que se observe la secuencia o_1 ,o_2,...., o_t y de que se alcance el estado j en el instante t+1 a partir del estado i en el instante t. Sumando este producto para todos los N posibles estados de partida en el instante t, se obtiene la probabilidad de estar en j en el instante t+1 para todas las secuencias parciales de observación previas.

3. Finalización

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \qquad (13)$$

El cálculo final de $P(O|\lambda)$ se obtiene como suma de las probabilidades hacia delante en el último instante posible T, es decir, el $\alpha_T(i)$ teniendo en cuenta que por definición

$$\alpha_t(i) = P(o_1 o_2 ... o_t, q_t = i \mid \lambda) \qquad (14)$$

Y que, por tanto, $P(O|\lambda)$ es la suma de las $\alpha_T(i)$.

Otro algoritmo semejante al forward es el backward que consiste en lo siguiente:

La probabilidad de observación de una secuencia en el estado i y con un determinado modelo es:

$$\beta_t(i) = P(O_{t+1}^t \mid q_t = i, \lambda) \qquad (15)$$

Donde $\beta_t(i)$ es la probabilidad de generar una secuencia de observación parcial O_{t+1}^t(secuencia de observaciones desde t+1 hasta el final) dados que el HMM está en el estado i, podemos obtener de forma inductiva:

1. Inicialización:

$$\beta_t(i) = \frac{1}{N} \quad 1 \le \mathrm{i} \le \mathrm{N} \qquad (16)$$

Todos los estados son equiprobables.

2. Inducción:

$$\beta_t(i) = \left[\sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \right] \quad \mathrm{t} = \mathrm{T}-1,....1;\ 1 \le i \le N \quad (17)$$

La relación entre α y β adyacentes se puede observar mejor en la siguiente figura. α se calcula recursivamente de izquierda a derecha mientras β se calcula recursivamente de derecha a izquierda.

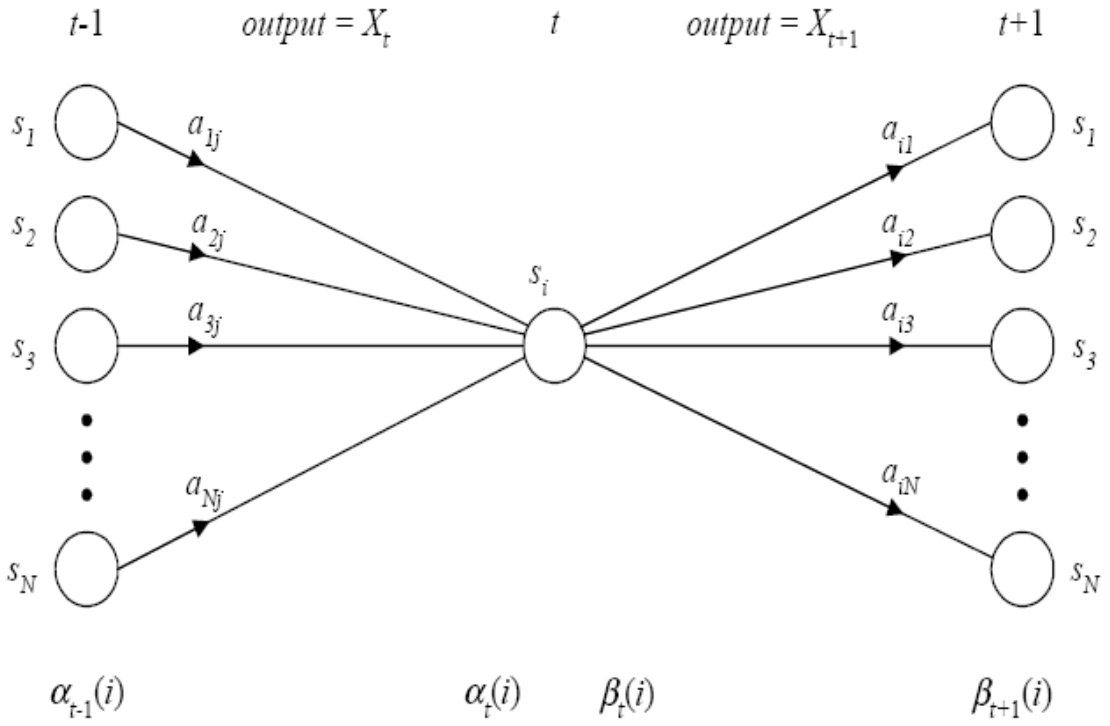

Fig. 10 Se representa tanto el algoritmo foward como backward [Huang et al 2001]

Decodificación HMM- Algoritmo de Viterbi

Decodificar consiste en encontrar la secuencia de estados dada una secuencia de observación, lo que puede ser deseable en muchas aplicaciones de segmentación y reconocimiento de voz.

A diferencia del problema de evaluación para el que se puede dar una solución exacta, existen diferentes maneras de resolver este problema. Esto se debe a que la definición de secuencia óptima no es única, sino que existen varios criterios de optimización.

Un criterio de optimización podría ser seleccionar aquellos estados que tengan individualmente la probabilidad más alta de ocurrencia. Sin embargo, este método no parece el más acertado ya que no tiene en cuenta la probabilidad de ocurrencia de secuencias de estados. Por ejemplo, la probabilidad de transición entre determinados estados es cero (aij=0), este criterio nos podría dar como solución al problema una secuencia de estados que no fuera válida.

Este problema puede resolverse con el algoritmo de Viterbi, que es similar al algoritmo anterior (Forward), con la excepción de que en vez de tomar la suma de valores de probabilidad en los estados anteriores, se toma el máximo de las probabilidades. De esta forma se consigue no solo dar la secuencia de observación más probable sino el camino de máxima probabilidad, consiguiendo la secuencia de estados que da una mayor probabilidad.

Antes de definir los pasos del algoritmo de Viterbi vamos a definir las siguientes variables:

$$\delta_t(i) = \max_{q_1, q_2 \ldots\ldots q_{t-1}} P[q_1 q_2 \ldots\ldots q_t = i, o_1, o_2 \ldots\ldots o_t \mid \lambda] \quad (18)$$

Donde $\delta_t(i)$ sería el mejor candidato (máxima probabilidad) a lo largo de un camino único, en el instante t, que tiene en cuenta la t primeras observaciones y termina en el estado i. Por inducción tendremos

$$\delta_{t+1}(j) = \max_i [\delta_t(i) * a_{ij}] b_j(o_{t+1}) \quad (19)$$

Para recuperar la secuencia de estados debemos seguir el argumento que maximiza la ecuación anterior para cada t y para cada j. Esto lo haremos a través de una tabla de vuelta atrás $\phi_t(j)$.

El proceso completo para encontrar la mejor secuencia será:

1. Inicialización

$$\delta_1(i) = \pi_i b_i(o_i) \quad 1 \le i \le N \quad (20)$$

$$\phi_1(i) = 0$$

Ponemos como los caminos anteriores el 0 para una vez alcanzado el final de este algoritmo al volver por la secuencia más probable no vayamos más para atrás.

2. Inducción

$$\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij} \right] b_j(o_t) \qquad 2 \le t \le T, \qquad 1 \le j \le N \qquad (21)$$

Se guarda aquel camino que tiene mayor probabilidad,

$$\phi_t(j) = \arg\max_{1 \le i \le N} \left[V_{t-1}(i) a_{ij} \right] b_j(o_t) \qquad 2 \le t \le T, \qquad 1 \le j \le N \qquad (22)$$

3. Finalización

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \qquad (23)$$

$$q_T^* = \arg\max_{1 \le i \le N} [\delta_T(i)] \qquad (24)$$

4. Seguimiento hacia atrás del camino óptimo (backtracking)

$$q_t^* = \phi_{t+1}(q_{t+1}^*) \quad t = T-1, T-2 .....1$$

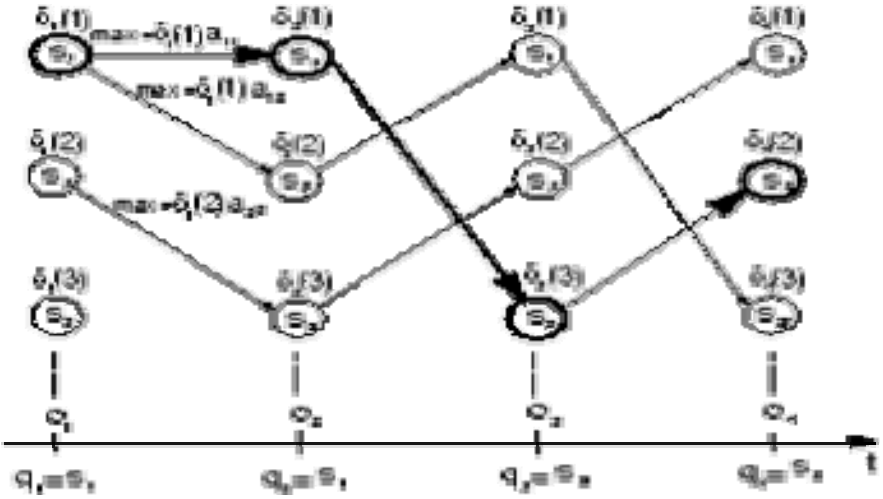

Fig. 11 Esquema del algoritmo de Viterbi

Como se puede observar el algoritmo seguido es muy semejante al de avance hacia delante empleado en la fase de evaluación, y el orden de operaciones también está en torno a $O(TN^2)$.

Aprendizaje de HMM-Algoritmo de Baum Welch

Aquí el problema que tenemos es que queremos estimar los parámetros del modelo λ (A, B, Π) de forma que maximicemos P(O|λ). Sin embargo, no existe ningún método conocido que permita obtener analíticamente el juego de parámetros que maximice la secuencia de observaciones. Por otro lado, podemos determinar este juego de características de modo que su verosimilitud encuentre un máximo local mediante la utilización de procedimientos iterativos como el del método de Baum-Welch, este no es más que un algoritmo E-M aplicado a los HMM; o bien mediante la utilización de técnicas de gradiente.

Un parámetro que debemos definir es el $\xi_t(i,j)$, como la probabilidad de encontrarnos en el estado i en el instante t, y en el estado j en el instante t+1, para un modelo y una secuencia de observación dados

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j \mid O, \lambda) \qquad (25)$$

Utilizando las probabilidades de los métodos forward y backward podemos escribir $\xi_t(i,j)$ con la siguiente formula:

$$\xi_t(i,j) = \frac{P(q_t = i, q_{t+1} = j \mid O, \lambda)}{P(O \mid \lambda)} = \frac{\alpha_t(i).a_{ij}.b_j(o_{t+1}).\beta_{t+1}(j)}{P(O \mid \lambda)} =$$
$$= \frac{\alpha_t(i).a_{ij}.b_j(o_{t+1}).\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i).a_{ij}.b_j(o_{t+1}).\beta_{t+1}(j)} \quad (26)$$

Suponiendo $\gamma_t(i)$ la probabilidad de encontrarnos en el estado i en el instante t, para la secuencia de observaciones completa y el modelo dados; por lo tanto, a partir de $\xi_t(i,j)$ podemos calcular $\gamma_t(i)$ con sólo realizar el sumatorio para toda j, de la forma:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \quad (27)$$

Realizando el sumatorio de $\gamma_t(i)$ para todo t, obtenemos un resultado que puede ser interpretado como el número esperado de veces (en el tiempo) que estamos en el estado i o de manera equivalente, número esperado de transiciones realizadas desde el estado i (excluyendo el instante t=T del sumatorio). De forma análoga, el sumatorio de $\xi_t(i,j)$ en t (desde t=1 hasta t=T-1) puede ser interpretado como el número esperado de transiciones desde el estado i al estado j

Con lo anterior podemos usarlo para la reestimación de los parámetros del HMM λ, quedando:

π_i'=número de veces que permanecemos en el estado i en el instante t=1, $\gamma_1(i)$

$$a_{ij}' = \frac{\text{número esperado de transiciones del estado i al j}}{\text{número esperado de transiciones desde el estado i}} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T}\gamma_t(i)} \quad (28)$$

$$b_j'(k) = \frac{\text{número esperado de instantes en el estado j observando el simbolo } v_k}{\text{número esperado de instantes en el estado i}} = \frac{\sum_{t=1}^{T}\gamma_t(i)\,|_{o_t=v_k}}{\sum_{t01}^{T}\gamma_t(i)} \quad (29)$$

Con estos cálculos obtenemos una re-estimación de los parámetros del modelo obteniendo un nuevo modelo λ'=(A',B', π'). Si el modelo λ definía un punto crítico de la función de máxima verosimilitud en dicho caso tendremos λ'=λ , o bien el nuevo modelo que hace que se cumpla P(O|λ')>P(O|λ), es decir, se ha mejorado el modelo de las secuencias de observación produciéndose con mayor verosimilitud. Por tanto, mejora la probabilidad de observar una secuencia O a partir de un modelo dado hasta llegar a un límite. Pero el principal inconveniente que tiene es que el método **Baum Welch** conduce de forma exclusiva a máximos locales. En la mayoría de los casos de interés la función de verosimilitud es compleja y contiene muchos de estos máximos. Los modelos que se manejan son Continuous-Density HMM con modelos de Gaussianas, las expresiones anteriores deberán pasar al caso continuo.

2.3. ESTADO DEL ARTE EN RECONOCIMIENTO DE IDIOMA

2.3.1. INTRODUCCIÓN

La detección automática del idioma en el habla espontánea consiste en la identificación del idioma partir de muestras de voz. También se puede distinguir entre dialectos, pero es más difícil de conseguir la clasificación automática de los mismos. El habla tiene una serie de características que nos ayudan a distinguir una lengua de otra, esas características se pueden aislar para ser usadas en el reconocimiento automático. Esos parámetros característicos de la lengua para la distinción idiomática se pueden agrupar en 4 grandes grupos que se muestran a continuación:

- *Nivel acústico*: Los distintos idiomas pueden tener patrones acústicos distintos, por ejemplo siendo más nasales, más guturales, etc.

- *Nivel fonético:* Los idiomas difieren también en el conjunto de fonemas que utilizan, así como en la frecuencia de utilización de los distintos sonidos y en la frecuencia de aparición de secuencias de sonidos. A este grupo pertenecen las características que vamos a explotar para el reconocimiento de idioma a lo largo del proyecto. En cierta forma se modela la aparición de una secuencia de sonido para caracterizar un idioma.

- *Nivel prosódico*: También se diferencian por tener distintos patrones prosódicos (duraciones, energía y tono de los fonemas), es decir, cada idioma tiene una entonación característica.

- *Niveles léxico, gramatical y superiores*: Finalmente, y posiblemente lo más importante desde un punto de vista conceptual, los idiomas tienen distintos vocabularios y distintas formas de combinar las palabras. El conjunto de palabras es posiblemente lo más característico de un idioma, de modo que un idioma puede reconocerse como tal si se emplea el vocabulario correcto pero no se emplean los fonemas o prosodia correcta. Pese a ser el nivel que más información sobre el idioma da, en la actualidad son pocas las técnicas que emplean este nivel para realizar la clasificación.

Al igual que ocurre en muchas otras tecnologías que tratan de imitar la percepción y comprensión humanas, en detección de idioma en voz espontánea se está lejos de alcanzar los niveles de precisión que consiguen los humanos entrenados (esta precisión se puede medir mediante estudios preceptúales [Y.K. Muthsami et al 1994a]). Para tratar de acercarse a dicha precisión, un detector de idioma debería idealmente hacer uso de las particularidades en los cuatro niveles del idioma descritos anteriormente. Sin embargo, el cuarto nivel (léxico, gramatical y superiores) resulta muy difícil de manejar porque requiere ser capaz de determinar la secuencia de palabras pronunciada a partir de exclusivamente la voz. En definitiva, para sacar partido de las particularidades del idioma en niveles léxicos y superiores es necesario disponer de un

reconocedor automático de voz con una precisión suficiente y capaz de manejar todos los idiomas que se desee detectar. Considerando que el reconocimiento de voz espontánea no está resuelto de modo satisfactorio ni siquiera en un único idioma sino que sigue siendo un tema de investigación muy activo (como lo atestiguan las recientes evaluaciones competitivas del programa Rich Transcription (RT) del Nacional Institute of Standards and Technology (NIST) [NIST RT 2004y NIST RT 2005] y resultados de de D.T. Toledano et al, [2004] y D.T.Toledano et al, [2005]), lo habitual es que la detección de idioma en voz espontánea se centre exclusivamente en los tres primeros niveles: acústico, fonético y prosódico. La ventaja de centrarse en estos niveles es que permite técnicas de modelado que pueden llegar a ser razonablemente independientes de los idiomas que se desea detectar, lo que proporciona una versatilidad que no se conseguiría con los niveles léxico y superiores.

2.3.1.1. APLICACIONES

Las aplicaciones del reconocimiento automático del habla se centran las tres áreas siguientes:

- **Indexado y recuperación de información** en contenidos de audio y audiovisuales. La creciente proliferación de contenidos multimedia en diversos ámbitos, como las emisiones de radiodifusión o Internet hace necesario la clasificación por idioma de los contenidos de los mismos. Hoy en día se necesita que los buscadores clasifiquen los contenidos multimedia de Internet y se indexen por idioma ya que es un entorno con una gran variedad lingüística.

- **En entornos telefónicos multilingües** tanto automáticos como con operador. En un entorno automático es necesaria la clasificación para que el usuario sea atendido en la lengua concreta desde el principio hasta el final. Por el contrario, para el caso de un operador hacemos que la consulta sea más rápida y eliminamos la necesidad de personal que haga la distinción de idioma. Por tanto, en este medio es importante la detección de idioma para poder hacer un enrutado automático de la consulta.

- **Sistemas de traducción simultanea voz a voz**, ya que en estos procesos es necesario conocer el idioma de los interlocutores. Con la utilización de un sistema de reconocimiento de idioma no será necesaria la configuración de estos sistemas.

Todas estas aplicaciones tienen un interés evidente para grandes empresas relacionadas con servicios telefónicos, con radio y televisión, con buscadores de Internet, y empresas e instituciones dedicadas a la vigilancia y seguridad.

2.3.2. HISTORIA DEL RECONOCIMIENTO DE IDIOMA

Aunque existen algunos estudios anteriores a 1970 [K. Atkinson, 1968] y en las décadas de los 70 y 80 se hicieron algunos estudios [Y.K. Mthusamy et al, 1994b], no fue hasta el 1992 cuando se inicia la investigación clara en este campo con la captura y puesta en publico de la base de datos OGI (Oregon Graduate Institute) [Y.K. Mthusamy et al, 1992]. Este corpus contenía 11 idiomas distintos con grabaciones de 50 segundos de 90 locutores de cada una de las lenguas nativas que se estudiaban. La revolución que supuso este corpus vino fundamentalmente por el hecho de que permitía la comparación entre sistemas, hecho hasta ese momento imposible, puesto que cada uno de los trabajos de distinción se había realizado sobre distintos datos, lo cual no permitía la comparación.

La facilidad de comparación de los resultados, junto con el creciente interés en este campo posibilitó la creación de la primera competición internacional de NIST en 1993 que hizo uso del corpus de OGI, y planteaba un procedimiento de evaluación idéntico para todos los participantes. Por tanto, la creación de OGI y la celebración de la evaluación internacional NIST se pueden considerar el punto de partida en este campo.

Después de la evaluación NIST se comenzó a realizar la captura de lo que se denomina "base de datos CallFriend" [CallFriend copora], que consistía en la creación de una base de datos de 12 idiomas con 60 conversaciones telefónicas entre dos locutores de forma espontánea durante 5 a 30 minutos. Con esto se solucionaba uno de los problemas de OGI que era la cantidad de datos con la que entrenar. Este material se fue empleando para la prueba de los sistemas en las evaluaciones de 1996 [NIST LRE 1996] y 2003 [NIST LRE 2003], siempre conservando el principio de evaluación ciega, es decir, los datos de test no estaban a disposición de los investigadores. Pero en la evaluación de 2005 [NIST LRE 2005] los datos ya pertenecían a otro corpus distinto a CallFriend.

El estado actual de investigación en este campo se resume en los resultados de las evaluaciones de 2003 y 2005 principalmente en los sistemas ganadores de las mismas. Aunque desde ellas se han realizado grandes mejoras algunas de las cuales se recogen en los resultados publicados en ICASSP (IEEE International Conference on Speech and Audio Processing ICASSP) en Honolulu, Hawai, 2007 [ICASSP 2007]. Las técnicas usadas se explicarán con detenimiento más adelante.

El futuro del reconocimiento de idioma está orientado a la detección de dialectos y variedades dentro de dicho idioma. También, se está trabajando en una nueva base de datos como es CallFriend2 que tendrá 30 idiomas. Y en el futuro se emplearán técnicas de más alto nivel al que se está usando en la actualidad como son los PPR que usan la información adicional de los modelos de idioma a nivel fonético, aunque dichas técnicas no parecen funcionar tan bien como se esperaba al menos de momento.

2.3.3. TÉCNICAS DE RECONOCIMIENTO DE IDIOMA

En esta sección vamos a mostrar las principales técnicas de reconocimiento de idioma que se utilizan en la actualidad. En concreto se va a mostrar los principales sistemas ganadores en las evaluaciones de 2003 [NIST LRE2003] y 2005 [NIST LRE 2005], que consisten principalmente en combinaciones de sistemas acústicos como los GMM y sistemas fonéticos como los PRLM y los PPRLM. Estos dos últimos son los sistemas implementados en este proyecto.

2.3.3.1. Gaussian Mixture Models (GMMs)

Los sistemas de reconocimiento de idioma de GMM se basan en el principio de que los idiomas tienen diferentes sonidos y que la frecuencia de aparición de los sonidos es diferente de un idioma a otros.

La realización de esta técnica consiste principalmente en seguir los siguientes pasos:

1. **Extraer características de la voz**, en particular se suele usar la parametrización de MFCC (es la misma que la que se usaba para la creación de los modelos fonéticos) o la SDC.

2. **Modelar los parámetros de entrada para cada idioma** como una mezcla de gaussianas multidimensionales por cada idioma. Cada vector de información o_t para t={1...T} y dado un modelo λ, la probabilidad de observación de dicha secuencia de parámetros vendrá dada por una mezcla de múltiples gaussianas [Marc A. Zissman, 1996]:

$$p(o_t \mid \lambda) = \sum N_m(o_t; \mu_m, \sum_m)$$

Donde μ_m y $\sum_m$ son respectivamente la media y la matriz de covarianza de la gaussiana m; λ es modelo de parámetros $\lambda = \{\omega_m, \mu_m, \sum_m\}$. Con los parámetros del modelos se entrena el algoritmo E-M al hacer varias operaciones con la probabilidad de observación de la secuencia o_t para una componente gaussiana m

$$\begin{aligned}
\bar{w}_m &= \frac{1}{T}\sum_{t=1}^{T} p(\mathbf{o_t}|\lambda) \\
\bar{\mu}_m &= \frac{\sum_{t=1}^{T} p(\mathbf{o_t}|\lambda)\cdot \mathbf{o_t}}{\sum_{t=1}^{T} p(\mathbf{o_t}|\lambda)} \\
\bar{\Sigma}_m &= \frac{\sum_{t=1}^{T} p(\mathbf{o_t}|\lambda)\cdot (\mathbf{o_t}-\bar{\mu}_m)\cdot (\mathbf{o_t}-\bar{\mu}_m)^T}{\sum_{t=1}^{T} p(\mathbf{o_t}|\lambda)}
\end{aligned}$$

3. **Reconocer**, se determina la probabilidad de que los vectores acústicos de la voz a clasificar hayan sido generados por el GMM de cada uno de los idiomas, seleccionando aquel que arroje un valor más alto También se puede detectar la presencia de un idioma comparando con el UBM

Las ventajas de esta técnica están en su relativa sencillez, así como en que no requiere que las locuciones estén etiquetadas fonéticamente, ya que se puede considerar como un HMM con sólo un estado. Su principal limitación es que modela únicamente los vectores de parámetros considerándolos de forma independiente e independientemente del fonema del que provengan: se modela únicamente información puramente acústica y a muy corto plazo (cada vector de parámetros se obtiene típicamente a partir de sólo 25 ms. de voz).

Debido a estas limitaciones el sistema no era competitivo con respecto a los PPRLM y por ello en la evaluación de 1996 estos sistemas estaban claramente por debajo. Pero la inclusión de un nuevo tipo de parametrización como era la de SDC [E. Wong y S. Sridharan, 2002 y P.A. Torres-Carrasquillo et al 2004] que introducía una mayor cantidad de información al expandir el tiempo de la ventana de calculo de los parámetros, hizo que fuesen competitivos e incluso superasen a los PPRLM en la evaluación de 2003.

2.3.3.2. Support Vector Machines (SVMs)

Las máquinas de vectores soporte (SVMs) son herramientas discriminativas genéricas de clasificación de patrones, que en los últimos años han demostrado ser muy potentes, por ejemplo, en reconocimiento de locutores. Extrapolando la experiencia en reconocimiento de locutores, se aplican a la problemática de detección de idioma obteniendo resultados bastante competitivos tanto respecto a las técnicas PPRLM como a las GMM.

La técnica consiste en partir de una serie de puntos que representan los vectores de parámetros del idioma a reconocer y de los idiomas impostores. Lo primero que hacemos es pasarlo a un espacio de dimensión mayor mediante un kernel como el GLDS [W. M. Campbell, 2002], una vez en dicho espacio, se calcula el hiperplano que separa mejor los dos grupos [W.M.Campbell et al, 2006], impostores y legítimos.

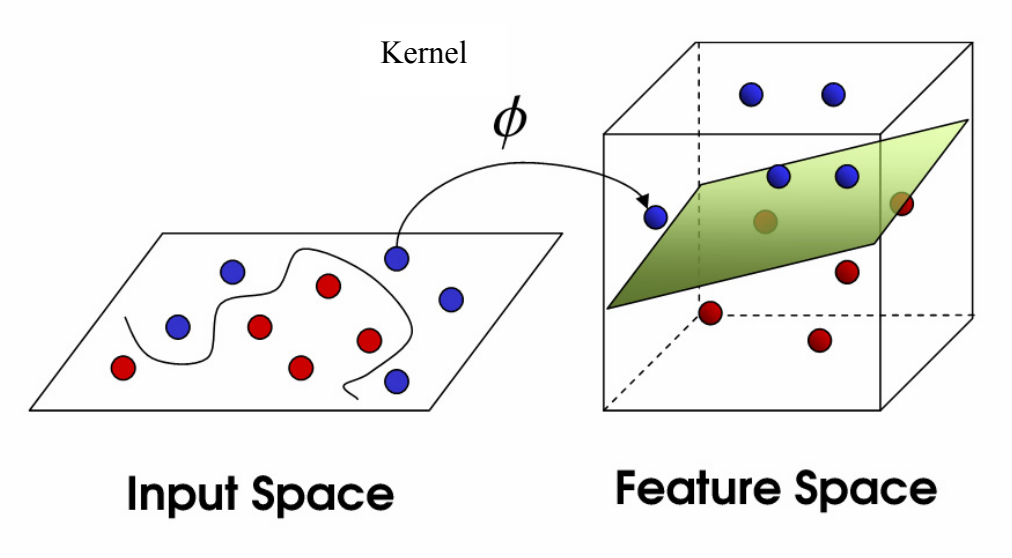

Fig. 12 Proceso de SVM [38]

2.3.3.3. Reconocimiento fonético de idioma: PRLM, PPRLM, PPR.

Las técnicas anteriores son técnicas relativamente nuevas en el campo de reconocimiento de idioma, además dichas técnicas únicamente explotan la distinción entre idiomas a nivel acústico. Las técnicas que vamos a presentar ahora tratan de explotar el nivel fonético de los idiomas, utiliza la información de niveles superiores a los que usaban los métodos antes descritos. Hay que destacar que estos sistemas fueron los que mejores resultados dieron en la evaluación de NIST LRE 1996 y NIST LRE 2005.

Estas técnicas se dividen en dos procesos:

1. **Definir y entrenar los reconocedores fonéticos** que consiste en lo explicado en 2.2, (también se puede consultar L.R.Rabiner [1989] y HTK book). Este paso es el que limita fundamentalmente estas técnicas, ya que precisa la creación de reconocedores fonéticos en varias lenguas, requiriendo la existencia de grabaciones y audio con transcripciones fonéticas. Por suerte, algunas de estas técnicas no precisan que haya reconocedores fonéticos para cada idioma a reconocer.

2. **Modelar estadísticamente el lenguaje** para que se modele la frecuencia de aparición de fonemas y de secuencias de fonemas para un idioma [F. Jelinek et al, 1990]

Según el modelado del lenguaje tendremos las distintas técnicas que se muestran a continuación

Phone Recognition followed by Language Modelling (PRLM)

Consiste en que una vez obtenida la transcripción fonética, se hace un modelo estadístico del lenguaje por medio de la creación de n-gramas. Hay que resaltar el hecho de que el modelo de lenguaje está entrenado con las transcripciones del reconocedor fonético y no con transcripciones fonéticas u ortográficas hechas de forma manual.

A continuación, nos centramos en el segundo paso que se da en estos sistemas, la generación del modelo de idioma (n-gram) ya que el primero (obtención de los fonemas) se ha visto en profundidad antes.

Con la transcripción fonética de la entrada puede ser entrenado el modelo de cada uno de los idiomas, calculando la frecuencia de aparición de los fonemas y de la combinación de los mismos, después en el reconocimiento se calculan también esos datos y en función de la probabilidad en cada modelo se decide a que modelo pertenece. Lo que se cuenta es la ocurrencia de n-grams, que son subsecuencias de n símbolos (fonemas en este caso)

En el entrenamiento lo que se hace es caracterizar por el histograma de aparición del los n-gramas, habiendo uno por idioma y asumiendo que cada idioma tiene diferente histograma [Marc A. Zissman, 1996]. Después se combinan los modelos de idioma dando lugar a un modelo universal (UBM), que en definitiva es un modelo que recoge las secuencias comunes a todos los modelos, si una secuencia de fonemas trascritos se da habitualmente en todos los idiomas dicha secuencia no podría ser tomada en cuenta a la hora de determinar a que idioma pertenece, puesto que no contiene información especifica para la distinción de idiomas.

Una vez calculadas todas las probabilidades de ocurrencia de todos los n-gramas para un modelo concreto y para el modelo de UBM obtendremos la puntuación de la siguiente forma

$$score = \log(\frac{P(x \mid LM_i)}{P(x \mid UBM)})$$

Donde X es la secuencia de n-gramas detectada en el fichero a reconocer. De la formula anterior se deduce que la puntuación que se obtiene depende tanto de lo que se parezca al modelo con el que se enfrenta, como de lo poco que se parezca al modelo genérico UBM, la puntuación será mayor cuando el conjunto de n-gramas del archivo a reconocer se den con alta probabilidad en el modelo de idioma y con baja en el modelo UBM, es decir, en el resto de idiomas. Por tanto, nos esté dando una mayor información de a qué idioma pertenece. Esta operación se puede ver de forma más clara en Fig. 13

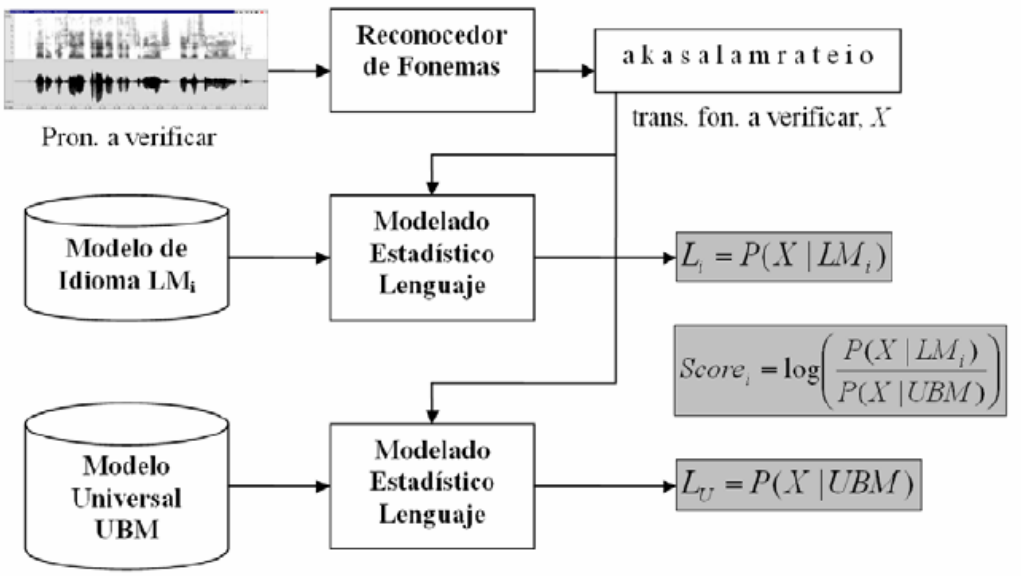

Fig. 13 Esquema de reconocimiento de un PRLM

Parallel PRLM (PPRLM)

Esta técnica es una extensión de la técnica de PRLM, consiste en la construcción de varios PRLM en paralelo, en el que cada uno usa un reconocedor fonético de un idioma distinto [T.J. Hazen y V.W.Zue, 1994], pero que no tienen que ser necesariamente de ninguno de los idiomas a reconocer. Por tanto, este sistema consiste en la obtención de una serie de puntuaciones en cada uno de los PRLM de forma independiente. La puntuación que se dé en cada uno de los PRLM se combina de alguna forma, por ejemplo, mediante una fusión suma.

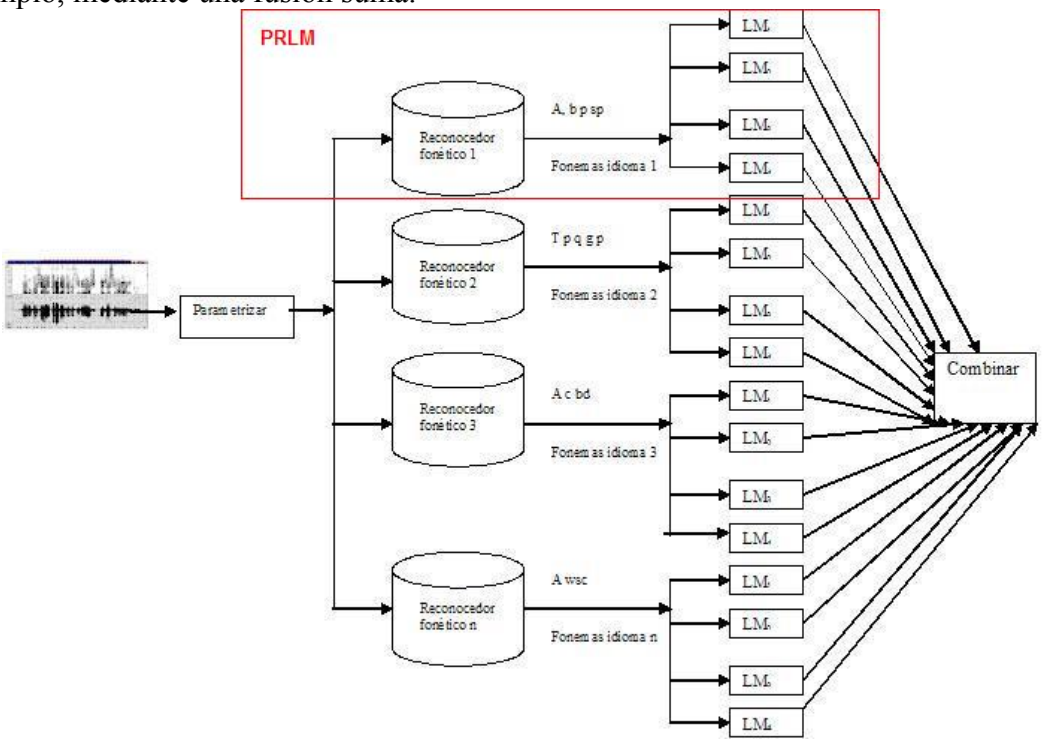

Fig. 14 Esquema de un PPRLM

Esta técnica fue de las primeras que comenzó a usarse en las primeras evaluaciones, pero en la evaluación de 2003 los sistemas de GMM y SVM han conseguido igualar los resultados de PPRLM e incluso superarlos, no así en la del 2005. Además han aparecido variaciones del PPRLM, que en lugar de usar transcripciones fonéticas usan parámetros de GMM o información de lattice de reconocimiento fonético [ICASSP 2007], no como hasta ahora en donde se utiliza el mejor camino dentro del lattice que es lo que da la trascripción.

Parallel Phone Recognition (PPR)

Está técnica es como un PRLM que consiste en aprovechar una de las características no utilizada en los sistemas anteriores, que es que los reconocedores fonéticos funcionan mejor con el idioma que transcriben. El sistema mantiene un reconocedor fonético, pero también tendrá un modelo de idioma (también de n-gramas) adaptado al idioma del reconocedor fonético, es decir, mientras que en un PRLM el modelo de idioma es de post-procesado de la trascripción, en el PPR es un proceso integrado [Marc A. Zissman, 1996]. Este sistema precisa de una gran cantidad de datos de entrenamiento, además de audio con transcripciones, supuesto que se necesita un reconocedor fonético por cada uno de los idiomas a reconocer. Por ello, teóricamente este algoritmo sería el más eficaz de los tres, pero en la realidad es muy difícil de implementar.

2.3.4. PROTOCOLOS, BASE DE DATOS Y PRESENTACIÓN DE RESULTADOS

2.3.4.1. Protocolo de evaluación, evaluaciones NIST

La organización NIST (National Institue of Standards and Technology), organiza regularmente evaluaciones tecnológicas competitivas en tecnologías tanto de reconocimiento de locutor, NIST-SRE (Speaker REcognition), como de idioma NIST-LRE (Language REcognition). Estas evaluaciones constituyen un foro científico y tecnológico que ha impulsado el desarrollo de los sistemas de reconocimiento basados en voz en las últimas décadas.

Dichas evaluaciones intentan cubrir los aspectos más relevantes de los sistemas que se implantan en la realidad, esto ha hecho que en las evaluaciones de idioma, que son en las que nos vamos a centrar, se incremente el número de idiomas y dialectos a reconocer.

Las evaluaciones NIST tienen un carácter abierto, en ellas participan grupos de investigación de todo el mundo. Su intención es establecer condiciones competitivas que permitan determinar el rendimiento de los diferentes sistemas involucrados. Una de las principales finalidades de este tipo de evaluaciones es poder comparar los distintos sistemas, técnicas y configuraciones de cada uno de los integrantes. La comparación de dichos sistemas entre sí ha fomentado la competitividad, obteniendo sistemas con un

grado de madurez suficiente como para funcionar en entornos reales de manera fiable. Entre los grupos de investigación que se suelen presentar está el grupo en el que se ha realizado este proyecto, ATVS que se ha presentado a las últimas evaluaciones de idioma y de locutor

El procedimiento de la evaluación define la medida de rendimiento y los datos sobre los que realizar la evaluación. Es el mismo para todos los integrantes, y viene definido por: datos de entrenamiento, test y datos complementarios (para técnicas de fusión, normalización, etc.).

Las evaluaciones NIST-LRE constan de un número de idiomas y dialectos a identificar. La evaluación está compuesta por diferentes pruebas de distinta dificultad. Estas pruebas irán orientadas a detectar la presencia de un idioma en la grabación de prueba. Existen tres duraciones distintas de segmentos de test, dependiendo de la cantidad de voz que contengan, se clasificarán en segmentos de 3, 10 y 30 segundos. Entre los datos suministrados para la evaluación se encuentran algunos específicos para el entrenamiento de los modelos, pudiéndose ampliar este conjunto de datos por parte de cada participante con datos disponibles públicamente.

Los resultados de la competición de idioma de NIST LRE 2003 y NIST LRE 2005 son los de la siguiente DET, no aparecen los nombres porque NIST no permite la publicación de los resultados identificando a los participantes.

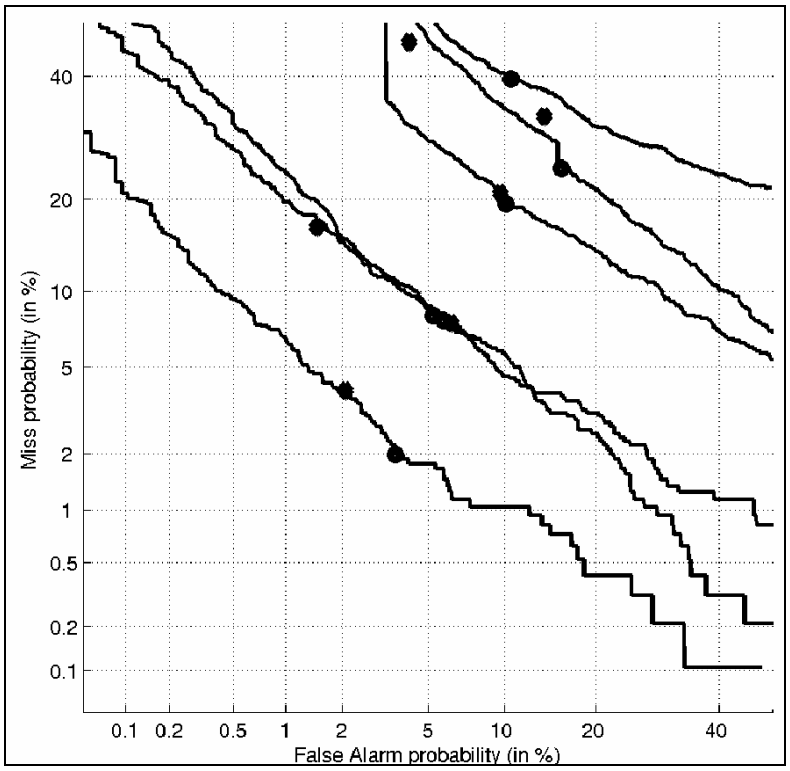

Fig. 15 Curvas DET de los 6 sistemas participantes en la evaluación NIST 2003 de detección de idioma, en la condición de evaluación principal (30 segundos de voz para hacer la detección de idioma).

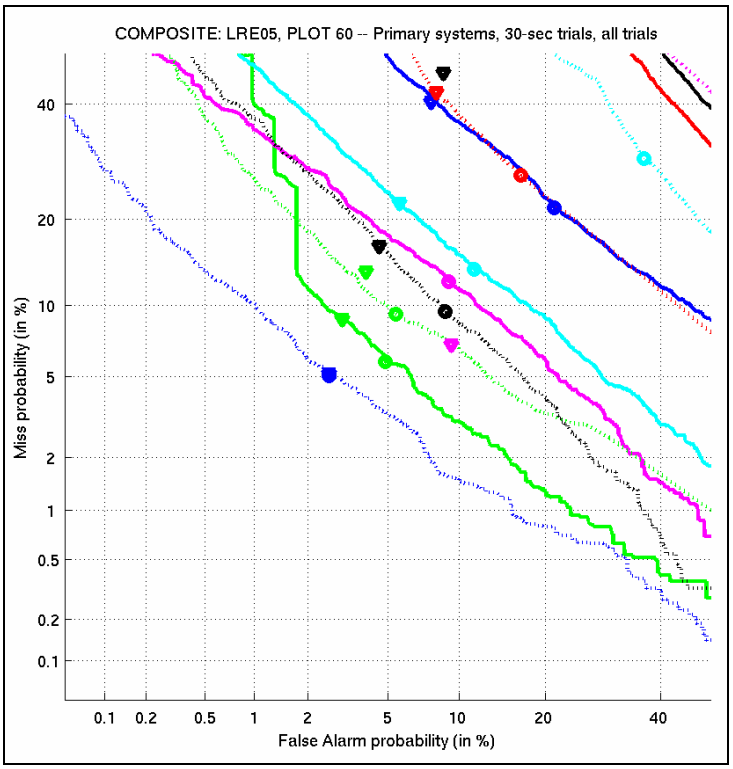

Fig. 16 Curvas DET de los 12 sistemas participantes en la evaluación NIST 2005 de detección de idioma, en la condición de evaluación principal (30 segundos de voz para hacer la detección de idioma) [A.Martin, et al, 1997].

Como se puede ver el EER del mejor sistema de NIST 2003 estaba entorno a 3% y en cambio en NIST 2005 es un poco superior, esto es debido a que a pesar de que los sistemas mejoraron de una a otra evaluación la variabilidad de los ficheros de la evaluación de NIST 2005 hizo que los resultados fuesen peores.

2.3.4.2. Bases de datos

Las bases de datos usadas para el entrenamiento y evaluación de los sistemas de reconocimiento de idioma que aquí implementamos son la base de datos CallFriend, que en un principio se emplea para entrenamiento, para los 12 idiomas que tiene, pero sin tener en cuenta la división entre dialectos. También se usa para test los datos de NIST 2003 y 2005, para entrenamiento los de NIST de 1996, cuando empleamos estos datos solo nos centramos en los 7 idiomas en la evaluación NIST 2005 que son los siguientes:

- Inglés
- Hindi
- Japonés
- Coreano
- Mandarín
- Español
- Tamil

En estas bases de datos no hay distinción de géneros y la información de las grabaciones no es tan detallada como la que tenemos en las bases de datos de SpeechDat cuyo formato y contenido se explicará en 3.1.1 de esta memoria.

El protocolo de evaluación seguido es el de NIST para las evaluaciones de idioma lo cual permite comparar nuestro sistema con otros sistemas en el mundo comparando su precisión con respecto a los otros.

2.3.4.3. Rendimiento de los sistemas de reconocimiento: presentación de resultados

Cuando diseñamos un sistema de reconocimiento de idioma como es el caso, es necesario tener herramientas y procedimientos que permitan ver las bondades del sistema. Esto se consigue por medio de una serie de valores, curvas, etc que servirán al desarrollador tanto para evaluar mejoras, como sistema de comparación de resultados con otros sistemas.

Los sistemas de verificación primero averiguan la puntuación, que es un número que nos da la verosimilitud entre la locución de prueba y el modelo de idioma contra el que se enfrenta. Una vez calculada esa puntuación compara ese valor con el valor umbral para decidir si es o no ese idioma.

En todo sistema de verificación de idioma hay dos posibles errores, que la locución de un idioma se identifique como de otro idioma, se dice que es falso rechazo y se mide con el FRR o False Rejection Rate; o que una locución de un idioma distinto sea aceptada como de ese idioma, se le llama falsa aceptación y se mide con el FAR o False Acceptance Rate.

Así según modifiquemos el valor del umbral tendremos un mayor valor de falso rechazo (si el umbral es muy alto) o de falsa aceptación (el umbral es muy bajo), según variemos el umbral aumentará uno de los errores y disminuirá el otro.

El establecimiento del valor de umbral esta condicionado a unas especificaciones de un punto de trabajo, hay principalmente tres opciones:

- Valor determinado de falso rechazo
- Valor determinado de falsa aceptación
- El punto de error igual, EER (Equal Error Rate), que es el punto donde la curva de falsa aceptación y falso rechazo se cruzan. Este valor es el que se suele utilizar para indicar el funcionamiento del sistema, pero en los sistemas reales el punto de trabajo no suele ser éste.

La falsa aceptación y el falso rechazo se pueden mostrar de forma gráfica. La falsa aceptación es el área que queda por encima del umbral y bajo la función de densidad de impostores, el falso rechazo es el área que queda por de bajo la curva de densidad de usuarios que queda por debajo del umbral

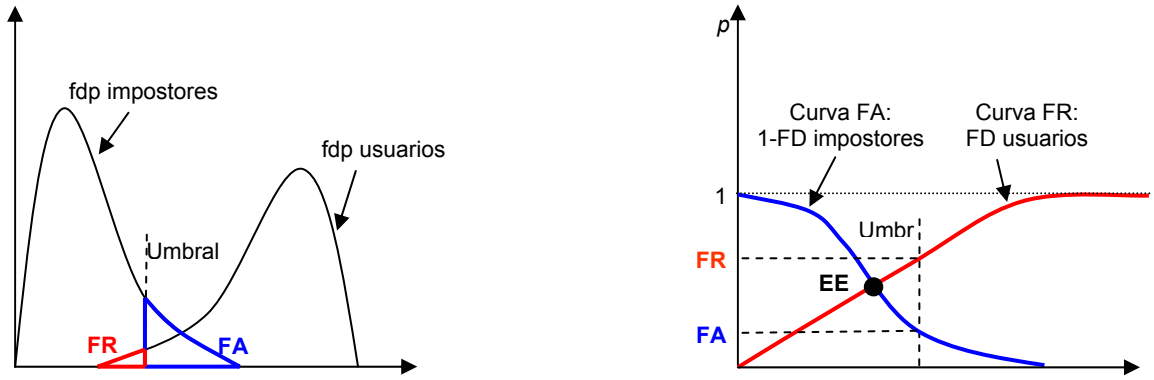

Fig. 17 Densidades y distribuciones de probabilidad de usuarios e impostores

Para representar estos valores otra forma es por medio de curvas ROC (Reciever Operating Characteristic). En estas gráficas se representan la FAR frente a (1-FRR) en función de diferentes valores para el umbral. Una alternativa frente a las curvas ROC, son las curvas DET (Detection Error Tradeoff) que se diferencian de las ROC por el cambio de escala [A.Martin et al, 1997], que hacen que lo que antes era una curva en las ROC ahora sea una recta. Las curvas DET son las que se han usado en el apartado anterior para mostrar los resultados de la evaluaciones de NIST, y será el formato de presentación de resultados en los experimentos realizados. Se muestra a continuación como se relaciona la curva ROC y la DET

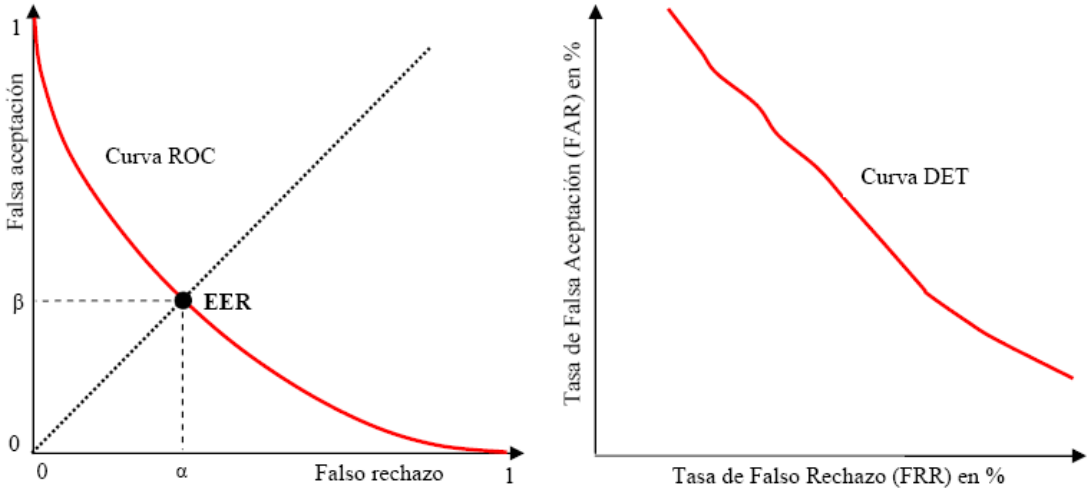

Fig. 18 Relación de curva ROC y DET

3. DISEÑO Y MEDIOS

3.1. MEDIOS DISPONIBLES

3.1.1. BASES DE DATOS

Para la realización de los reconocedores fonéticos necesitamos audio de los idiomas para los que se va a construir un reconocedor fonético, pero no solo necesitamos el audio sino que dicho audio debe estar correctamente trascrito, ya fuera de forma fonética o en forma de palabras. En este último caso también precisábamos de un diccionario fonético de todas las palabras que se pronuncian en el audio y su trascripción como es representado en la Fig. 19

```
WORD  FREQUENCY   PHONEMIC TRANSCRIPTION
A     2381        a
abade 1           a B a D e
abadeek     1     a B a D e e k
abadiñarraren     1     a B a D i J a rr a r e
n
Abadiño     3     a B a D i J o
Abadiñoko   2     a B a D i J o k o
Abadiñon    2     a B a D i J o n
abagunea    1     a B a G u n e a
abal  1           a B a l
Abaltzisketa      1     a B a l ts` i s k e t a
Abaltzisketan     1     a B a l ts` i s k e t a
n
Abandoibar  1     a B a n d o i B a rr
abandoibarrako    1     a B a n d o i B a rr a
k o
abandoibarratik   1     a B a n d o i B a rr a
t i k
abandonatu  1     a B a n d o n a t u
abandonatua 1     a B a n d o n a t u a
```

Fig. 19 Diccionario fonético

Las bases de datos usadas son las del conjunto de SpeechDat, en concreto se han usado las de 6 idiomas:

- Euskera
- Frances marroquí
- Árabe marroquí
- Alemán
- Ruso
- Inglés

La principal característica de estas bases de datos es la gran cantidad de archivos de audio que hay, estando divididos en bloques y sesiones, Además estas bases de datos

están equilibradas en cuanto al género. Aunque cada base de datos pertenece a un solo idioma dentro de las mismas también se hace distinción entre los diferentes dialectos.

Por cada archivo de audio viene un archivo de datos, que nos indica diversos datos aunque los de mayor importancia son los de la trascripción ortográfica y la fonética de lo que el locutor dice. El formato de dichos archivo se puede ver a continuación:

LHD: SAM, 6.0 DBN: SpeechDat_East_Russian_Fixed_Network VOL: FIXED3RU_04 SES: 0800 DIR: \FIXED3RU\BLOCK08\SES0800 CMT: ********** File information ********** SRC: A30800A1.RUA CCD: A1 CRP: BEG: 0 END: 45595 ASS: OK REP: AudiTech Ltd, St.Petersburg, Russia RED: 03/October/1999 RET: 14:06:16 12 CMT: ********** Speech data coding ********** SAM: 8000 SNB: 1 SBF: SSB: 8 QNT: A-LAW CMT: ********** Speaker information ********** SCD: 064500 SEX: F AGE: 47 ACC: TVER CMT: ********** Recording condition ********** REG: MIDDLE RUSSIA ENV: HOME NET: FIXED PHM: ROTARY LBD: CMT: ********** BODY ********** LBR: 0,45595,,,,повторить LBO: 0,22798,45595,[sta] повторить [spk] ELF:	mnem. comments LHD: format name + version ELF: end of label file CMT: comment row DBN: database name VOL: database volume ID SES: session number DIR: signal file directory SRC: signal file name CCD: corpus code CRP: corpus repetition BEG: labelled sequence start position END: labelled sequence end position ASS: assessment code REP: recording place: place, city, country RED: recording date RET: recording time (:SS = :00) SAM: sampling frequency SNB: number of (8-bit) bytes per sample SBF: sample byte order (meaningless with single byte samples, .SNB: 1.) SSB: number of significant bits per sample QNT: quantization SCD: speaker code SEX: speaker sex AGE: speaker age ACC: speaker accent REG: calling region ENV: calling environment NET: telephone network PHM: telephone hand set model LBD: label body keyword LBR: labelling during recording: begin, end, gain, min, max, orthographic text prompt LBO: orthographic labelling: begin, centre, end, orthographic transcription text

Fig. 20 Formato de archivo de datos [SpeechDat Ruso]

A partir de estos archivos descartamos muchos ficheros que tuviesen ruido, indicado en las transcripciones de los mismos. Los ruidos venían indicados con etiquetas como spk (ruido de otro locutor), sta (ruido estacionario)..., pero muchas veces se tuvo que asumir ciertos ruidos ya que si se excluían dichos archivos el número de ficheros disponibles bajaba enormemente. Por tanto, lo primero que se hace es extraer las transcripciones y a partir de ellas descartar de la lista aquellos archivos que por el nivel de ruido o por el tipo de ruido no interesan.

Normalmente estas bases de datos suelen venir divididas en ficheros de test y de entrenamiento, la proporción suele ser de 20 % y 80% respectivamente. En aquellas bases de datos en la que no estaba hecha la división la realizamos siguiendo esa proporción.

Los archivos vienen en formato raw comprimido y codificado en formato ley A, muestreado a 8 KHz y mono. Antes de empezar a usarlo se transforma a archivos .wav, en formato PCM y muestreados a 8 KHz, esto es necesario para poder ser procesado por el parametrizador. Por tanto, antes de empezar a usar los archivos de sonido se deben de transformar a un formato distinto al que venían.

En cuanto las bases de datos empleadas para reconocimiento de idioma lo que se empleaba era las bases de datos CallFriend y los datos de las evaluaciones NIST de el 1996 hasta el 2005, como ya se ha explicado en la sección 2.3.4.2 de esta memoria.

3.1.2. SOFTWARE

El sistema operativo empleado en la realización de este proyecto ha sido Debian y las principales herramientas de software empleadas son HTK y Sphinx.

El Hidden Markov Model Toolkit (HTK) es un toolkit portable usado para la construcción y manipulación de los modelos de Markov. HTK fue usado en principio en aplicaciones de reconocimiento de voz, aunque se ha encontrado otras muchas aplicaciones como la síntesis de voz o secuencias de ADN.

HTK consiste en un conjunto de librerías y herramientas desarrolladas en C. Las sofisticadas herramientas desarrolladas facilitan el análisis de la voz, el entrenamiento de los HMM, el test y extracción de resultados. El software soporta la creación de HMM con distribuciones continuas de mezclas de Gaussinas o por medio de distribuciones discretas pudiendo crear de esta forma complejos sistemas de HMM.

HTK fue desarrollado originalmente en el laboratorio de la inteligencia de máquinas (conocido antes como el grupo “the Speech Vision and Robotics”) del departamento de ingeniería de la Universidad de Cambridge (CUED) donde se ha utilizado para construir grandes sistemas de reconocimiento de habla (véase HTK CUED LVR). En 1993 Entropic Research Laboratory Inc. adquirió los derechos de vender HTK y el desarrollo de HTK fue transferido completamente a Entropic en 1995 en que el laboratorio de investigación de Entropic Cambridge Ltd fue establecido. HTK fue vendido por Entropic hasta que en 1999 Microsoft compró Entropic. Microsoft ahora ha licenciado HTK de nuevo a CUED y está proporcionando la ayuda de modo que CUED pueda redistribuir HTK y proporcionar la ayuda del desarrollo vía el Web site HTK3 [HTK]

El grupo Sphinx está desarrollado por la Universidad Carnegie-Mellon, la Defense Advanced Research Projects Agency (DARPA) financia extensamente el proyecto para estimular la creación de herramientas de discurso y el uso de las mismas, en el reconocimiento de voz, así como en áreas relacionadas incluyendo sistemas de diálogo y síntesis de discurso.

Sphinx ha sido apoyado durante muchos años por la financiación del DARPA y los motores del reconocimiento que se lanzan son los que el grupo utilizó para varios de los proyectos de DARPA y sus evaluaciones respectivas.

La ayuda reciente para el proyecto también incluye Telefónica I + D, Sun Microsystems, y los laboratorios de investigación eléctricos de Mitsubishi.

Los términos en que se licencian para los motores y las herramientas de Sphinx se derivan del DEB, y se basan, particularmente, sobre la licencia para el web server de Apache. No hay restricción contra uso o la redistribución comercial.

Los paquetes que el grupo Sphinx de CMU está lanzando son un sistema razonablemente maduro. Los componentes proporcionan un nivel básico de la tecnología a cualquier persona interesada en crear reconocedores sin el coste de inversión inicial; los mismos componentes están abiertos a la revisión para todos los investigadores en el campo, y se utilizan para la investigación lingüística también. [Carnegie Mellon University SPHINX]

Sphinx al igual que HTK es una serie de herramientas y librerías escritas en C y durante la realización de este proyecto hubo momentos en los que fue necesario modificar dicho código para solucionar algunos problemas que surgieron

3.1.3. HARDWARE

El hardware empleado en el desarrollo de este proyecto fue en un ordenador con procesador Intel Pentium IV y un disco duro externo. Se dispone además de una red interna incluyendo todos los ordenadores del grupo de trabajo, tanto los de uso personal como los de pruebas. Todos estos medios fueron suministrados por el grupo ATVS de la Universidad Autónoma de Madrid (UAM)

3.2. DISEÑO

3.2.1. ENTRENAMIENTO Y EVALUACIÓN DE HMM FONÉTICOS

Antes de la realización de este proyecto en el grupo ATVS únicamente estaban los reconocedores fonéticos entrenados con TIMIT y con Albayzin, bases de datos que tienen aproximadamente 300 locutores y que es habla microfónica leída, también se habían hecho algunos reconocedores con OGI pero con muchos menos datos de entrenamiento que los de las bases de datos anteriores.

El objetivo en esta parte del proyecto es la creación de unos modelos fonéticos, los cuales serán entrenados y probados con las bases de datos SpeechDat, que a diferencia de TIMIT y Albayzin tienen un mayor número de datos de entrenamiento como se pude ver en el número de locutores, por ejemplo, en el ruso llegan a 2500, además es habla telefónica espontánea siendo más realistas que las anteriores. Dichos modelos se han realizado con un entrenamiento de modelos HMM de 5 estados por fonema. El entrenamiento se hizo con HTK, pero el reconocimiento en la parte de test se ha realizado tanto con HTK como con Sphinx. Por tanto, el objetivo de estos experimentos es doble; la creación de modelos fonéticos de mayor calidad a los que había, y la adaptación de los mismos para ser usados tanto por HTK como por Sphinx, analizando cual de las dos herramientas se adapta mejor a nuestro interés.

El procedimiento seguido para la creación y testado de cada uno de los reconocedores fonéticos es el siguiente:

1. **Parametrizar** todos los ficheros de audio de la base de datos.

    a. Cambiar el formato del audio, que suele estar en formato raw y con compresión ley A (por ser habla telefónica), esto obliga a cambiar de formato a wav.

    b. Parametrizar con Sphinx, son archivos con 13 coeficientes MFCC por ventana.

    c. Cambiar el formato de parametrización para que lo pueda usar HTK, para ello se añade los coeficientes MFCC-Delta y los MFC-Delta-Delta. Los ficheros de parámetros pasan a tener 39 coeficientes por cada ventana.(Fig. 5)

2. **Obtener transcripciones fonéticas:** en las bases de datos usadas traen la trascripción de lo que se decía en cada uno de los archivos, y usando los diccionarios fonéticos que venían, se generaron las transcripciones fonéticas reales, tanto para los ficheros de entrenamiento como para los ficheros de test. En este último caso, son las que se utilizan para comprobar la fiabilidad de los reconocedores.

3. **Entrenar los modelos fonéticos**: el entrenamiento de los modelos se realiza con HTK, empleando un modelado de HMM para cada fonema en la que cada uno de los estados del fonema es representado por un determinado número de mezclas de gaussianas. Esta fase de entrenamiento se divide a su vez en las siguientes fases:

    a. Entrenar los modelos fonéticos considerando que únicamente hay un solo tipo de silencio que será el valido tanto para el comienzo y final de locución como para pausa corta. Los modelos de este tipo se crean de una sola Gaussiana y se re-estiman 4 veces.

    b. Copiar el modelo de silencio generando uno para comienzo y fin de locución y un tercero para la creación de la pausa corta. Este último se modifica de forma que la probabilidad de salto del primer estado al último sea del 90% mientras que el salto al siguiente estado es solo del 10% (ver Fig. 21)

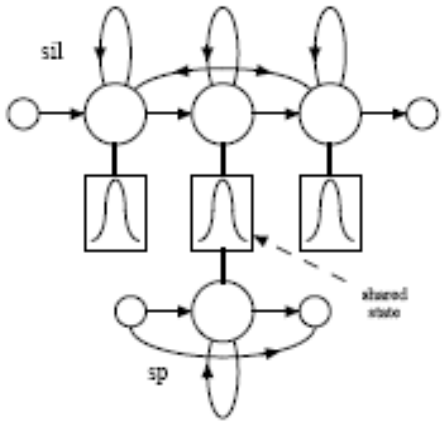

Fig. 21 esquema de estados de la pausa corta [HTK book]

c. Se prosigue añadiendo Gaussianas y por cada Gaussiana añadida, se re-estiman los modelos cuatro veces. Este proceso se repite hasta que llegamos a que cada estado está representado por 20 Gaussianas.

d. Una vez entrenados se vuelve a poner el modelo de pausa corta para que en el primer estado vaya con una probabilidad de 1 al siguiente estado en lugar de que tenga alguna probabilidad de ir al último estado.

4. **Construir gramática,** en la que se indican los posibles fonemas que se pueden dar en el reconocedor y dando igual probabilidad de aparición a cualquiera de ellos. En estos experimentos no se usa el conocimiento del nivel léxico-semántico de la lengua, no se tiene en cuenta que después de un determinado fonema haya más probabilidades de que aparezca uno concreto, como sucede en la realidad, donde la aparición de un fonema hace que sea más probable la aparición de otro, ya que se está realizando reconocimiento fonético no de palabras.

5. **Probar los modelos** comparando las transcripciones fonéticas que se obtienen con los mismos y las que se pronuncian realmente. Para probar las transcripciones que se obtienen con dichos modelos se emplea tanto HTK como Sphinx, para ello hay que transformar los modelos de HTK a Sphinx. En ambos sistemas para los parámetros de entradas se elige la secuencia de fonemas de mayor probabilidad. Además dichas herramientas tenían ciertos parámetros para **regular él número de fonemas insertados**, los cuales se eligen de forma que los resultados sean óptimos (el mayor número de aciertos posibles con el menor número de inserciones posibles)

Este proceso se pude resumir en el siguiente esquema:

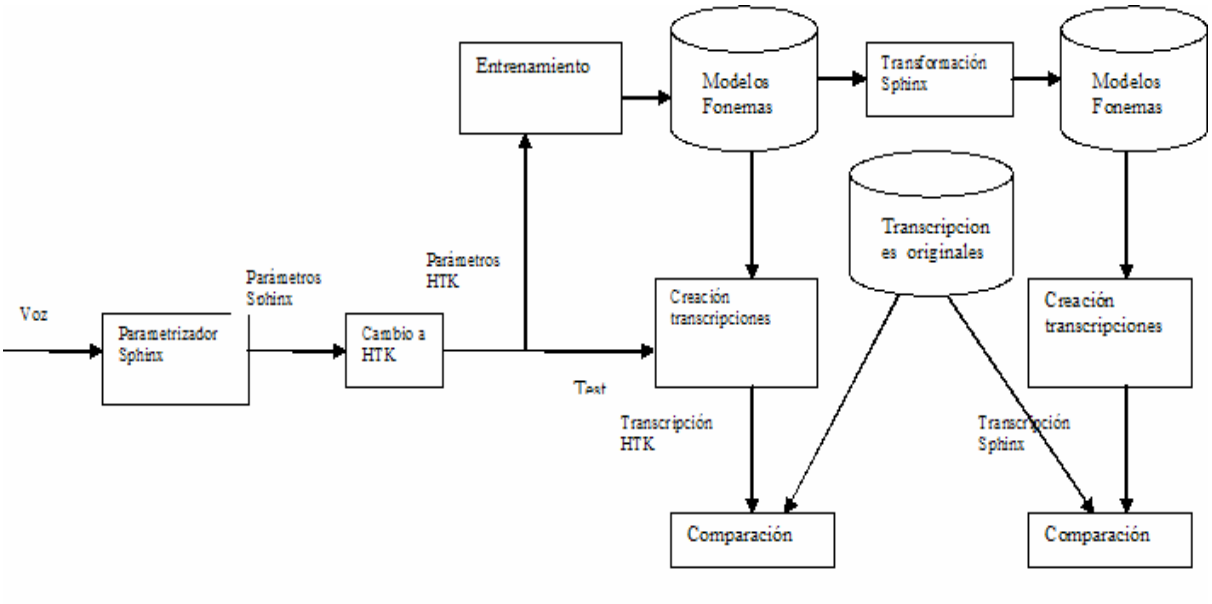

Fig. 22 Esquema de realización de experimentos

La evaluación de los reconocedores consiste en indicar cuanto se parece la transcripción obtenida a partir de los archivos de test y las transcripciones originales. En ambos casos se utiliza un comando de HTK que da la comparación como se ve en Fig. 23 y que usa el modelo de medida estandarizado por el National Institue of Standards and Technology de EEUU

```
====================== HTK Results Analysis =======================
  Date: Wed Nov 15 09:34:46 2006
  Ref : labels1.mlf
  Rec : results_HTKformat_2006_9_C.txt
------------------------ Overall Results --------------------------
SENT: %Correct=5.31 [H=338, S=6026, N=6364]
WORD: %Corr=59.84, Acc=58.29 [H=88386, D=36865, S=22447, I=2299, N=147698]
```

Fig. 23 Formato de resultados

En nuestro caso el valor de "SENT" no tenía sentido ya que habitualmente éste indica cuantas de las frases reconocidas coinciden con la original y en un reconocimiento fonético esto suele ser muy bajo siempre. Por el contrario, los datos de la fila "WORD" en nuestro caso no se refieren a palabras sino a fonemas. Los otros valores que aparecen son los siguientes:

- **H**: número de fonemas correctos
- **D**: número de fonemas borrados
- **I**: número de inserciones de fonemas
- **N**: número de fonemas totales en la trascripción original
- **S**: número de fonemas sustituidos

- **%Corr** es el porcentaje de fonemas correctos $\%Corr = \frac{H}{N}100$
- **%Acc** es la precisión del sistema $\%Acc = \frac{H - I}{N}100$

En nuestro caso, lo que buscamos es un sistema con un alto valor de %Corr, pero sin descuidar el valor del %Acc. De forma que el sistema tenga una gran tasa de aciertos, pero que dicha tasa no sea debida a la introducción de un gran número de fonemas espurios. En estos cálculos no se tienen en cuenta la identificación de los silencios ni del tipo de silencio.

Además de estos resultados también se mostraba la matriz de confusión que tiene la forma que se puede observar en la Fig. 24, en la que se observa la confusión de unos fonemas con otros. También se observa el número de inserciones y de borrados de cada fonema.

Fig. 24 Matriz de confusión

3.2.2. GENERACIÓN Y EVALUACIÓN DE UN SISTEMA PPRLM

El objetivo es la creación de un sistema PPRLM, de forma que sea competitivo con el resto de los sistemas que existen en la actualidad para la identificación de idioma, pudiendo presentar dicho sistema combinado con otros a la evaluación de idioma de NIST 2007. Se partía de un sistema base de PPRLM como el que se puede ver en el artículo de Alberto Montero-Asenjo et al. [2006].

El sistema de partida tenía unos reconocedores fonéticos entrenados con OGI; y tras estudiar el funcionamiento del sistema con modelos fonéticos de 10 y 20 Gaussianas en cada estado de cada modelo de fonema, se hacía una combinación de resultados de los modelos de 10 y 20 Gaussinas..En este sistema los pasos que se dan para realizar el reconocimiento son los siguientes:

1) Parametrización MFCC (Fig. 5) con el estándar ETSI ES 202 050

2) Realizar el reconocimiento fonético con los modelos que había antes de la realización del proyecto, los entrenados con TIMIT, Albayzin y los de OGI

3) **Construir un PRLM** con cada uno de los reconocedores fonéticos:

   a **Entrenar** un modelo de Universal Background Model (UMB) [D. Reynolds et al. 200] y los modelos de cada idioma a reconocer con HTK, a partir de las transcripciones fonéticas de los archivos de entrenamiento de CallFriend

   b **Enfrentar los archivos de test a los modelos construidos**, obteniendo unas puntuaciones según lo cerca que estén, el peso que se da al UBM es de 0,6 y para el modelo concreto es de 0,4 [Alberto Montero-Asenjo et al. 2006]. Estos pesos se averiguaron de forma empírica. Con esas puntuaciones creamos las curvas DET de cada idioma y la global, además de calcular el EER del sistema para su caracterización.

   c **Normalizar con TNorm** que consiste en calcular para cada archivo de test la media de las puntuaciones con los otros modelos de idioma y su varianza. Una vez calculadas, para normalizar hacemos la siguiente operación:

$$New\_score_i = \frac{Score_i - \mu}{\sigma} \text{ donde } \mu = \frac{1}{\mathrm{N}} \sum_{j=1}^{N} Score_j \text{ y la } \sigma = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (Score_j - \mu)^2}$$

4) Fusionar los PRLM, en concreto lo que se hace es una fusión suma de los PRLM, con lo que obtenemos un PPRLM.

5) Realizar el TNorm de los resultados fusionados

6) Generar las curvas y calculo del EER de todo el sistema general

Para mejora el sistema base antes descrito, realizamos unos profundos cambios en el mismo, que son los siguientes:

- **Parametrización MFCC pero hecha con Sphinx**, transformándola a formato HTK

- **Aplicar un detector de voz**, el cuál detecta la voz en función de los parámetros MFCC, y en particular de la energía, si es ruido o es voz el audio parametrizado. En función de esta detección se varían los ficheros de parámetros para que solo contengan las ventanas de voz. En teoría, esto mejorará el comportamiento de los reconocedores fonéticos, que insertarán menos fonemas espurios, y reduce el coste computacional total, puesto que los segmentos de silencio no se procesan más.

- **Realizar reconocimiento fonético con reconocedores de mayor calidad**. En concreto se cambian los reconocedores de TIMIT y OGI por reconocedores de mayor calidad como los SpeechDat, cuya creación se ha explicado en 3.2.1. En concreto los reconocedores fonéticos que se usan ahora son los 7 siguientes:

  - Inglés (entrenado con SpeechDat)
  - Árabe (entrenado con SpeechDat)
  - Francés (entrenado con SpeechDat)
  - Alemán (entrenado con SpeechDat)
  - Euskera (entrenado con SpeechDat)
  - Ruso (entrenado con SpeechDat)
  - Español (entrenado con Albayzin) [A,Moreno et al. 1993]

- **Eliminar del sistema HTK y sustituirlo por Sphinx y programas propios.** Este cambio se debe a dos motivos: la licencia de HTK no permite ser usado en sistemas de producción reales mientras que Sphinx sí, y el otro motivo es abrir nuevos horizontes de investigación en el grupo como son los PRLM basados en lattices o Phone-SVM [W.M.Campbell et al, 2007]

Como consecuencia de los cambios indicados con anterioridad, los experimentos se dividen en 3 grandes grupos:

1. Utilizando a la hora de parametrizar un detector de voz y parametrizando y extrayendo la trascripción fonética con Sphinx, pero usando para la creación de los modelos HTK al igual que a la hora de reconocer el idioma.

2. Analizar la influencia del detector de voz, introduciendo el mismo en el sistema base.

3. Eliminación de HTK de todo el proceso

4. PRUEBAS Y RESULTADOS

4.1. PRUEBAS Y RESULTADOS DE RECONOCIMIENTO FONÉTICO

Las pruebas realizadas en este punto consistieron en el entrenamiento de los reconocedores fonéticos con SpeechDat y la evaluación de los mismos, pero antes del entrenamiento propiamente dicho hay una fase de limpieza y adaptación de transcripciones fonéticas de la base de datos a un formato entendible por HTK. En esto se tardaba de media una semana, puesto que la limpieza de archivos corruptos en algunos casos se debía hacer a mano. Por otro lado, la fase de entrenamiento de los modelos se automatizó por medio de la creación de una estructura de experimentos, pero el proceso duraba de media una semana y media e incluso en algunos casos, como en el reconocedor de SpeechDat de ruso, el entrenamiento duro 20 días en un Pentium IV a 2,3 GHz.

Vamos a mostrar los resultados de la fiabilidad de los reconocedores fonéticos que se han construido, así como una comparativa de reconocimiento con Sphinx y con HTK, relacionando todos estos resultados con lo que suele ser habitual en reconocimiento fonético.

A continuación, mostramos la variación del tanto por ciento de acierto conforme vamos aumentando el número de mezclas de Gaussianas, que constituye cada uno de los estados de los modelos fonéticos, empleando para dichos datos HTK y siendo los resultados de los seis reconocedores fonéticos construidos:

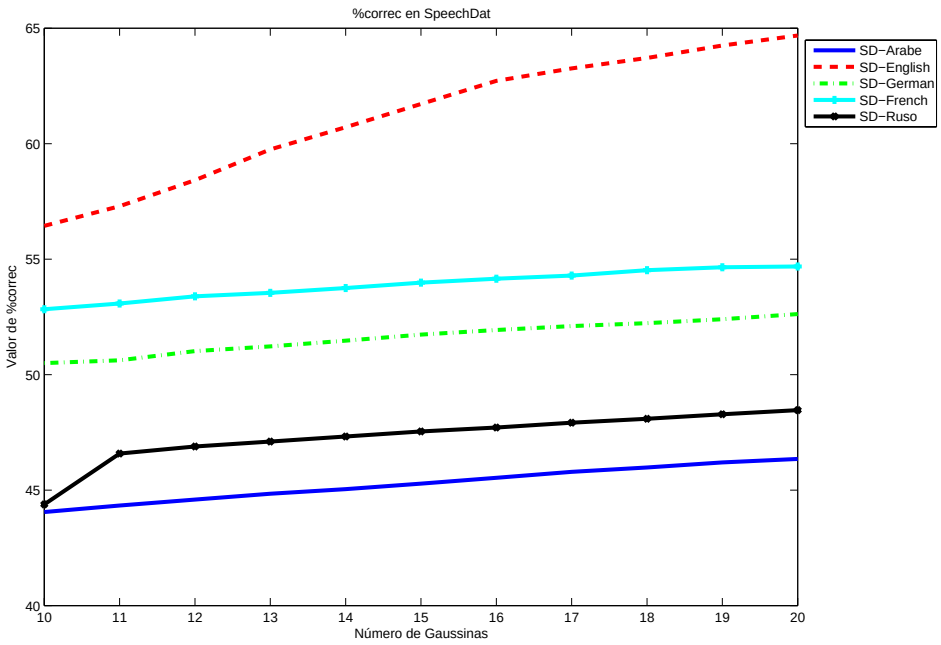

Fig. 25 Curvas de % corr con diferentes Gaussinas y para diferentes reconocedores

Como se puede ver en la Fig. 25, conforme va aumentando el número de Gaussianas va incrementando el nivel de acierto. El que mejor resultados obtiene es el de inglés.

Otro dato que es importante, es el nivel de exactitud de los modelos, es decir, que el porcentaje de aciertos sea alto pero que esto no sea consecuencia de muchas inserciones. Esto se puede observar en la siguiente Fig. 26 para los mismos reconocedores que antes.

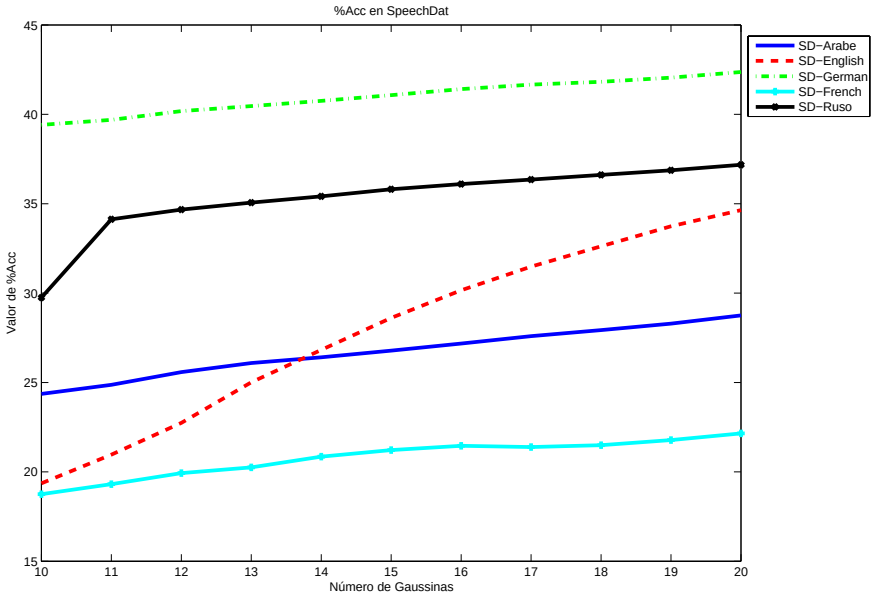

Fig. 26 Nivel de exactitud de los reconocedores fonéticos

Se observa en la gráfica (Fig. 26) que según aumentamos el número de Gaussianas, la exactitud de los sistemas va aumentando. También se ve cómo algunos reconocedores mejoran la exactitud más rápido que otros. Hay que destacar en estos resultados que el inglés tiene una alta tasa de acierto pero que como se puede ver por el valor de %Acc es debida en gran media al gran número de inserciones, por contra, está el ruso cuyo nivel de aciertos no era muy alto, pero el nivel de %Acc si lo es. En la identificación de idioma lo que nos interesa es que tenga un alto nivel de %Acc ya que no es útil un sistema con muchas inserciones espurias.

Los buenos resultados que tiene el reconocedor de ruso en cuanto a valor de %Acc , se explican por el hecho de que esa base de datos es la que mayor cantidad de datos tenía de todas las SpeechDat, aunque en contra tiene que es el reconocedor con mayor cantidad de fonemas a reconocer con 48 fonemas, mientras que por ejemplo, Albayzin tenia sólo 23 fonemas.

También resalta el hecho de que el reconocedor de francés y el de árabe pese a que tienen el mismo número y tipo de fonemas, y que en ambos los locutores son marroquíes, puesto que es francés marroquí y árabe marroquí, los resultados tanto en %corre como en %Acc presentan una gran diferencia entre ellos.

Para los modelos de 20 Gaussianas también se hizo el reconocimiento con Sphinx. La comparación entre ambas herramientas se muestra a continuación:

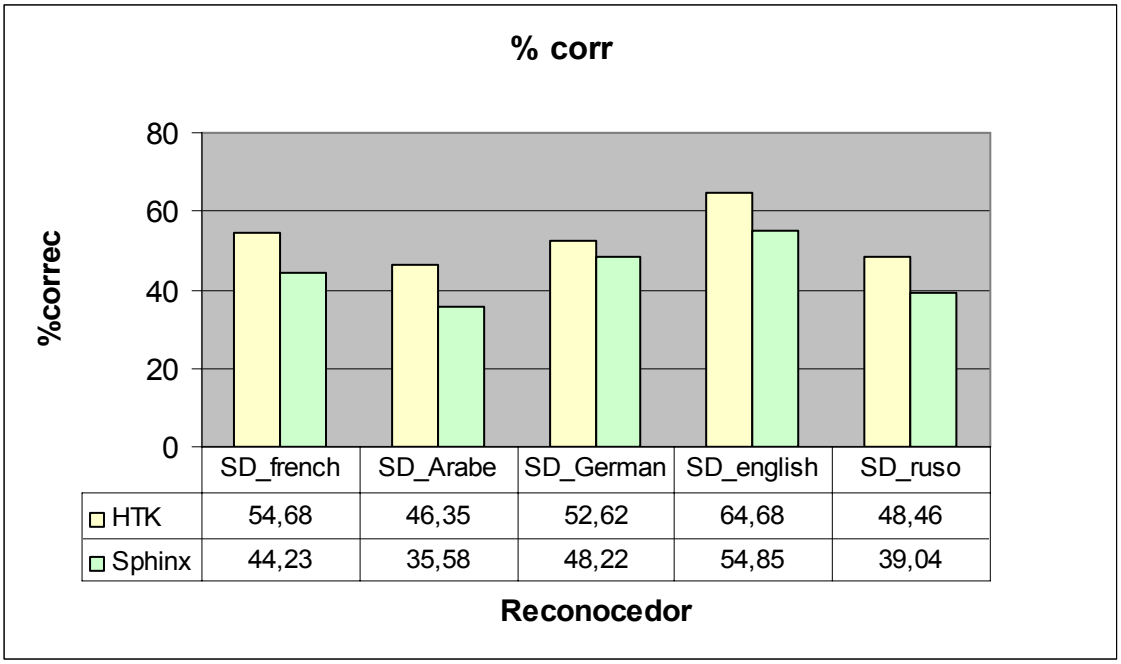

Fig. 27 Comparación de % aciertos de HTK y Sphinx

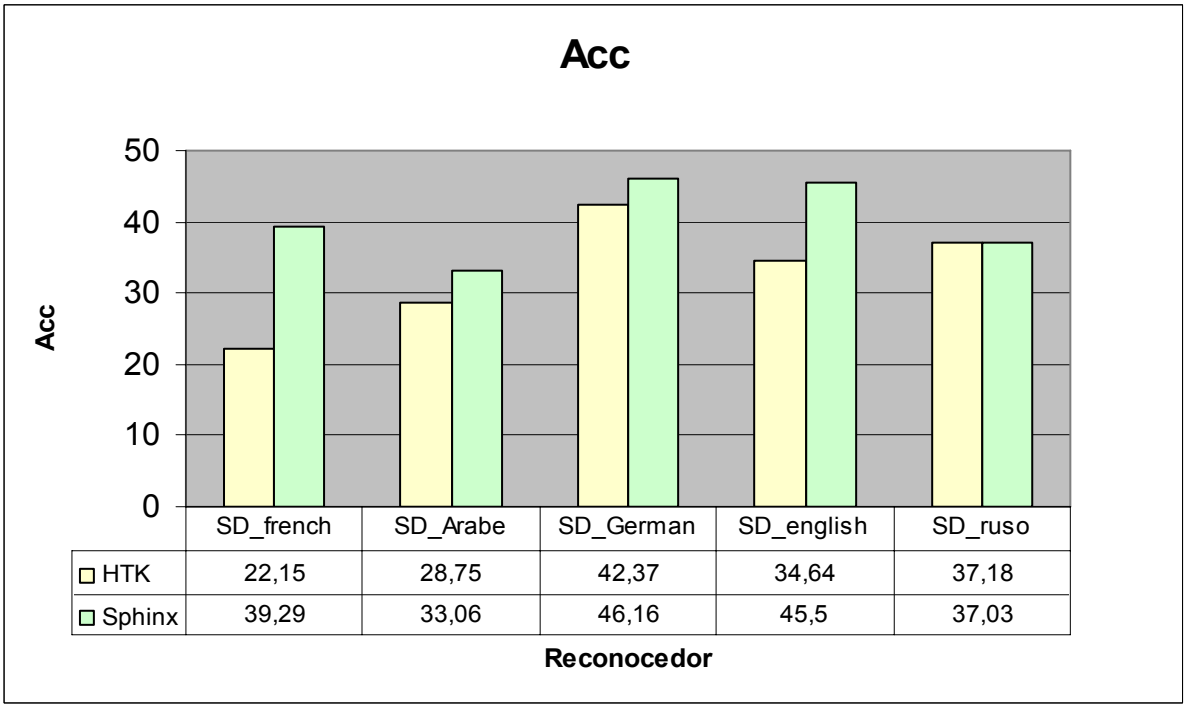

Fig. 28 Comparación de la fiabilidad del reconocimiento con HTK y con Sphinx

Según se observa en la Fig. 27 el reconocedor de HTK tiene una mayor cantidad de aciertos, pero en contra tiene valores de precisión inferiores o muy inferiores en algunos casos a los de Sphinx. (Fig. 28). Para conseguir estos resultados se estuvo probando, tanto en HTK como después en Sphinx, diferentes valores del "Word Insertion Penalty". Este parámetro introduce una penalización por cada fonema que se inserta, de esta forma, se consigue reducir el número de fonemas espurios en las transcripciones. El reducir la cantidad de dichos fonemas es de vital importancia para el reconocimiento de idioma en un sistema a nivel fonético como es el PRLM. Hay que destacar, que en el caso de Sphinx pese a que la documentación del mismo indicaba que el valor del "Word Insertion Penalty" suele ir entre 0,2 y 0,7, en la prueba se llego hasta un valor de 11 y se encontró que el óptimo del sistema estaba en 9 para la gran mayoría de los reconocedores. Por tanto, superamos los teóricos valores óptimos de este parámetro, según Sphinx, obteniéndose unos valores considerablemente mejores tanto de %corre como de %Acc. La necesidad de superar los valores aconsejados del parámetro proviene del hecho de que estamos empleando Sphinx para reconocer fonemas en lugar de su uso habitual que es reconocer palabras completas.

Los resultados presentados con anterioridad si los comparamos con los que podemos encontrar en la actualidad, en los resultados de N. Morales Mombiela [2007] y Li Deng et al. [2006], son bajos. Un primer motivo es que en este trabajo, por cuestión de simplicidad, nos hemos restringido a emplear modelos fonéticos independientes del contexto que no modelan la coarticulación que se da entre fonemas adyacentes, mientras que lo habitual es emplear modelos fonéticos dependientes del contexto. Otro segundo motivo es que, además del modelo HMM del mismo tipo que los presentados anteriormente, se suele emplear un modelo de idioma de bigramas que hace que mejoren los resultados. En nuestro caso el modelo de idioma de bigrama era incompatible con la utilización de los modelos en un sistema PPRLM, por lo que decidimos no emplearlo. Todo ello explica que los resultados presentados en este proyecto sean muy inferiores a los presentados en otros trabajos de reconocimiento fonético, como el de N. Morales Mombiela [2007] que alcanza un 70% de precisión en TIMIT y el de Li Deng et al. [2006], que obtiene un 71,43 % Acc gracias al uso de la técnica HTM .

La gran ventaja de estos reconocedores con respecto a los reconocedores existentes en el grupo con anterioridad a este proyecto (los de TIMIT, Albayzin y OGI) es que están entrenados con más datos, además esos datos son de habla telefónica espontánea. Por tanto, se tienen reconocedores de mayor calidad para el reconocimiento de idioma [Matejka Pavel, et al. 2005] y adaptados a la evaluaciones de NIST que son de habla telefónica espontánea.

4.2. PRUEBAS Y RESULTADOS DE RECONOCIMIENTO DE IDIOMA

Antes de mostrar y explicar los resultados del nuevo sistema PPRLM construido, vamos a mostrar los resultados y características del sistema de partida, nuestro "baseline". El sistema de partida era un clásico PPRLM pero con la principal diferencia con respecto a otros sistemas PPRLM en que usaba para realizar la transcripción fonética reconocedores fonéticos de diferente número de Gaussinas y luego fusionaba sus resultados de reconocimientote idioma. Los reconocedores fonéticos usados en este sistema fueron entrenados con Albayzin, TIMIT y OGI. (Para una descripción más profunda de este sistema leer 3.2.2 de este proyecto y Alberto Montero-Asenjo et al. [2006]).

Los resultados obtenidos por este sistema cuando se entrena con todo CallFriend y reconociendo para un subconjunto de NIST LRE 2003 se presentan en la Fig. 29 .

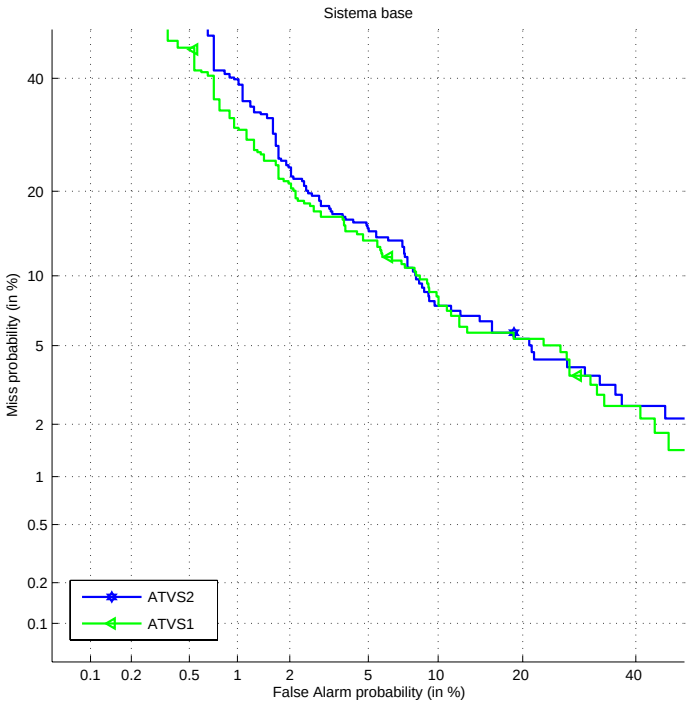

Fig. 29 Sistema base del que se partía. Resultados para NIST2003 sobre los 7 idiomas de 2005

Tanto en los resultados anteriores como en los siguientes, el sistema ATVS2 es el de reconocedores de 10 Gaussinas por estado y el de ATVS1 es la combinación de los reconocedores de 10 y 20 Gaussinas por estado.

El resultado para el sistema entrenado también con CallFriend, pero reconociendo NIST LRE 2005 es la Fig. 30 y tiene un EER del 16,38%. Aun estando lejos del sistema ganador de esa evaluación, era un sistema competitivo a nivel internacional.

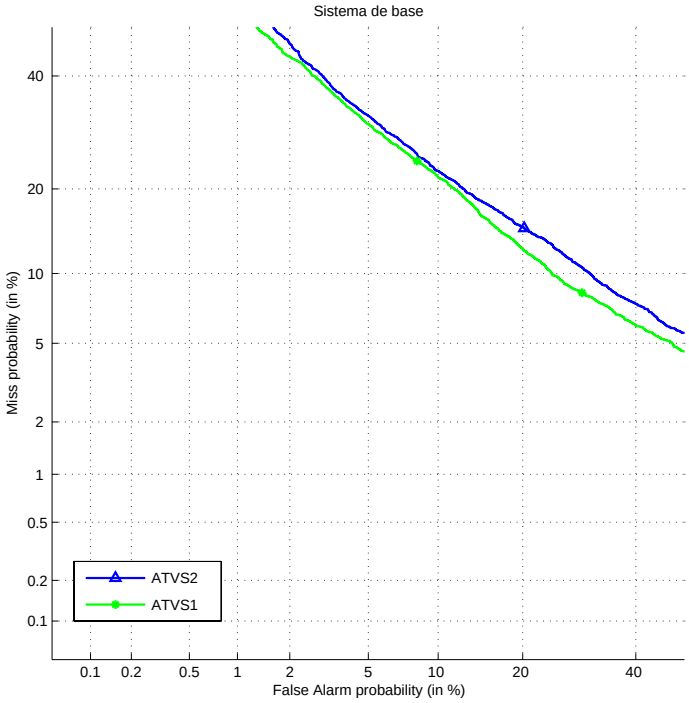

Fig. 30 Sistema bases del que se partía. Resultados para NIST2005

4.2.1. EXPERIMENTOS CON DETECTOR DE VOZ Y PARAMETRIZACIÓN Y RECONOCIMIENTO FONÉTICO DE SPHINX

En estos experimentos vemos los efectos de la introducción de algunos de los cambios que se realizaron sobre el sistema base. En concreto los cambios son los siguientes:

- **Cambia la parametrización,** ahora se hace con Sphinx y se pasa a formato de HTK.

- **Se introduce el detector de voz** mejorando el coste computacional, ya no se procesan los segmentos de silencio o ruido.

- **Se sustituyen los reconocedores** de TIMIT y OGI por los entrenados con SpeechDat, que son de mayor calidad, debido a que están entrenados con más datos y esos datos son de habla telefónica espontánea.

Estos cambios obligaron a cambiar la estructura de los experimentos hechos con el sistema base. Además durante la realización de estos estas pruebas se descubrieron varios fallos en el software de Sphinx que obligo a depurar el mismo y a modificarlo.

Para la evaluación del sistema creado se hicieron varias pruebas que son las siguientes:

CALLFRIEND

Probamos el sistema con los archivos de evaluación de la base de datos CallFriend., Estos resultados tienen una cierta relevancia, puesto que como se ha indicado con anterioridad, las evaluaciones de NIST 1996 y 2003 fueron realizadas con archivos de estas bases de datos.

A continuación mostramos el comportamiento de reconocimiento global de los sistemas para los 7 idiomas de NIST2005 en cada uno de los sistemas PRLM (incluyendo TNorm) en cada uno de ellos y el efecto que conseguimos al hacer la fusión suma de los mismos:

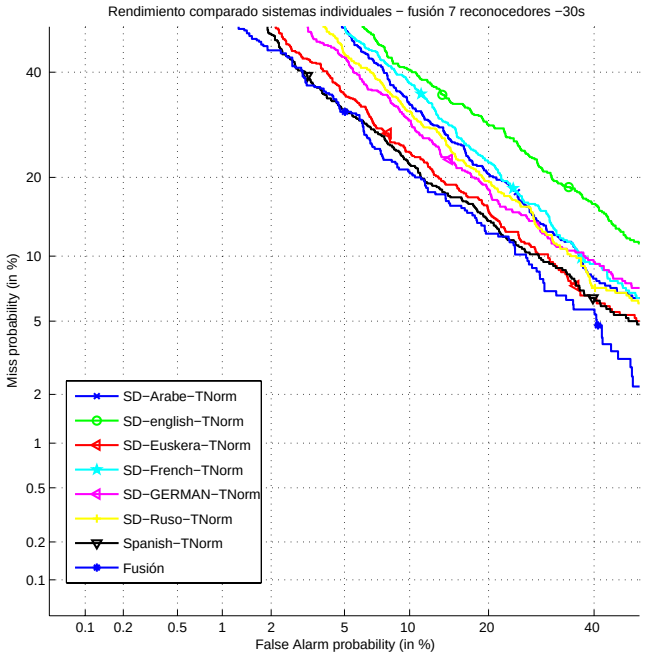

Fig. 31 Fusión para prueba de CallFriend para ficheros de 30s

De la gráfica anterior deducimos, que la mejora obtenida con la fusión es ligera en comparación con alguno de los PRLM usados, en concreto con el de español, cuyo comportamiento es prácticamente igual al sistema fusionado. Esto lo que nos indica es que la información aportada por cada PRLM esta muy correlada entre ellos, por tanto, la fusión no tiene una gran mejora en el reconocimiento de idioma

PRLM	EER-30s
SD-Arabe	20.40
SD-english	23.89
SD-Euskera	16.85
SD-French	21.30
SD-German	17.59
SD-Ruso	19.44
Spanish	15,87
Fusión	15,87

Tabla 1 EER de cada uno de los PRLM en la prueba de CallFriend

En la siguiente tabla podemos ver un análisis más en profundidad del reconocimiento por idioma, dando la tasa de EER para cada uno de los idiomas y el del sistema global:

Idioma	EER-30 s
Inglés	4,44
Hindi	24,81
Japonés	15,18
Coreano	13,33
Mandarín	6,67
Español	10,37
Tamil	13,33
Global	13,33

Tabla 2 EER de CallFriend

Como se puede ver en los valores de EER, el hindi es el de mayor tasa de error del conjunto. Esto es debido a que por las características de ese idioma se confunde con los otros y las puntuaciones obtenidas son menores. Por contra, tenemos idiomas como el inglés que se confunde menos con el resto, por lo que tiene un EER muy bajo, de igual forma pasa con el mandarín.

Para ver más en detalle cómo se identifican los 45 ficheros de test en cada uno de los casos, a continuación mostramos las matrices de confusión:

30-s							
inglés	hindi	japonés	coreano	mandarín	español	tamil	
38	1	0	2	1	3	0	**inglés**
2	13	7	0	0	10	13	**hindi**
0	1	23	16	0	1	4	**japonés**
0	1	3	35	1	2	3	**coreano**
2	1	1	1	40	0	0	**mandarín**
1	0	4	1	0	39	0	**español**
0	1	2	1	2	2	37	**tamil**

Tabla 3 Matriz de confusión en Callfriend para 30s

Se puede observar en la tabla anterior, que el idioma que peor se identifica es el hindi ya que se confunde con el español y el tamil. Por el contrario tenemos al mandarín o al español, que identifican como propios la mayoría de sus archivos, pero en el caso del español le perjudica que muchos de los archivos de hindi se identifiquen como de español, tiene una alta tasa de falsa aceptación.

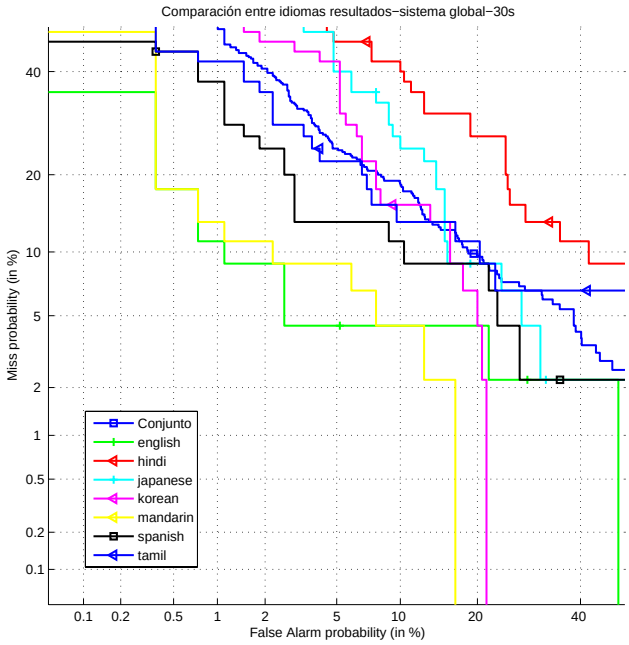

Fig. 32 Resultado por idioma para fichero de 30s en CallFriend

Como se puede ver en la Fig. 32 el comportamiento para la identificación del mandarín es globalmente mejor que el comportamiento en la identificación del inglés, aunque éste último se comporta mejor en cuanto al EER.

Lo principal de esta prueba es que se vio que el sistema funcionaba bien dando resultados competitivos con pruebas anteriores hechas con el sistema base. Además se empezaban a notar las ventajas de los cambios realizados, ya que mientras que el tiempo de parametrización no había variado apenas, el tiempo de transcripción se había disminuido notablemente. Además el uso de Sphinx hacia que a la vez que se realizaba la transcripción daba un fichero con más información que es el fichero de lattices, a lo que tenemos que unir el que ya no se procesen los silencios de las conversaciones.

Otra percepción que sacamos de los mismos, es que a pesar de que cada sistema por separado tenía un mínimo de EER de 15,87% para los ficheros de 30s, luego con la fusión obtuvimos un mínimo de 13,33% como comportamiento global.

Este experimento se hizo como primera aproximación a resultados más importantes como son los de las evaluaciones NIST. Además, como ya se ha explicado con anterioridad, los datos CallFriend son los que se emplearon en las evaluaciones de NIST LRE 1996 y NIST LRE 2003. Por tanto, el EER daba una idea de la potencia del sistema que se ha construido.

En esta prueba también se realizo un estudio de cuando era más conveniente realizar el TNorm y se llego a la conclusión que la obtención de los mejores resultados era como se hacia en el sistema base, es decir, hacer el TNorm de las puntaciones obtenidas en cada PRLM y una vez fusionado se realiza de nuevo el TNorm de la puntuaciones.

NIST2003

Esta prueba se hace con objeto de comparar el rendimiento con el sistema base de partida. En concreto no se realiza para toda la evaluación NIST 2003 sino que se realiza para un subconjunto de 40 archivos de cada uno de los 7 idiomas de NIST 2005 y a diferencia de CallFriend estos archivos tienen la característica de que ya están eliminados los segmentos de silencio, por tanto, se van a filtrar por segunda vez al pasar por el detector de voz. Se siguen los mismos pasos indicados en 3.2.2 ; tampoco cambia la cantidad de datos de entrenamiento que vuelven a ser los archivos de desarrollo de CallFriend, es más, gracias a la prueba anteriormente explicada ya teníamos los modelos de idioma y el UBM que emplearíamos en esta prueba. Por tanto únicamente se tuvo que parametrizar y segmentar los archivos de test de NIST LRE 2003

Los primeros resultados que mostramos son los de cada PRLM por separado y ya aplicado TNorm y la fusión de dichos sistemas sin aplicar TNorm.

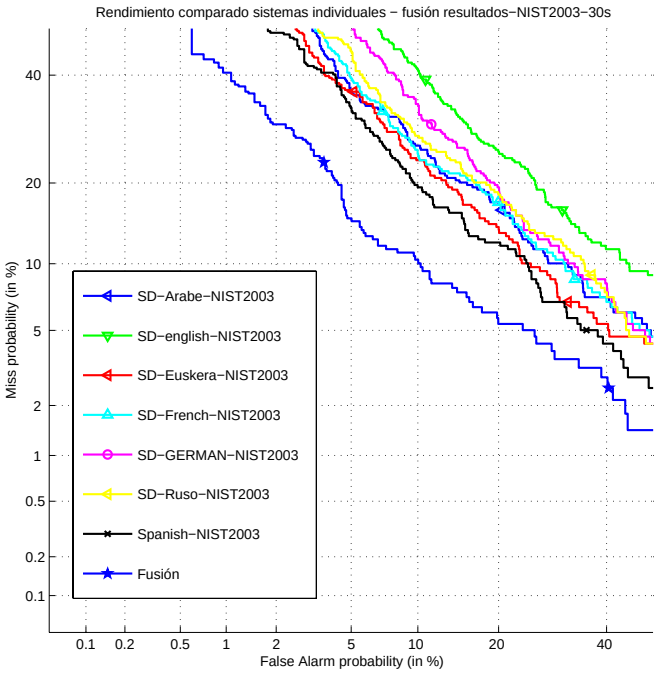

Fig. 33 Fusión de los 7 PRLM con TNorm en NIST 2003

A diferencia de los resultados obtenidos en el experimento anterior, aquí la fusión suma de los sistemas PRLM provoca una mejora considerable del rendimiento del sistema en todos los puntos de actuación.

PRLM	EER-30s
SD-Arabe	18.52
SD-english	23.30
SD-Euskera	16.49
SD-French	18.28
SD-German	19.71
SD-Ruso	19,00
Spanish	15.05
Fusión	10,04

Tabla 4 EER de los PRLM en la prueba de NIST 2003

En este caso la fusión se ha comportado mejor que en el caso de CallFriend ya que la mejora obtenida como se puede ver en la gráfica es considerable.

Una vez hecha la fusión y realizada también TNorm a los datos ya fusionados, los resultados por idioma a reconocer son los siguientes:

Idioma	**EER-30 s**
Inglés	3,33
Hindi	13,39
Japonés	15,00
Coreano	7,50
Mandarín	6,28
Español	7,53
Tamil	6,69
Global	8,90

Tabla 5 EER NIST2003

30-s							
inglés	**hindi**	**japonés**	**coreano**	**mandarín**	**español**	**tamil**	
33	1	1	0	3	0	1	**inglés**
1	30	5	1	0	0	3	**hindi**
0	2	30	6	0	0	2	**japonés**
0	0	5	35	0	0	0	**coreano**
0	1	2	3	31	1	2	**mandarín**
0	2	3	1	0	33	1	**español**
0	4	0	0	0	1	35	**tamil**

Tabla 6 Matriz de confusión de NIST2003

Comparando la Tabla 6 y la Tabla 3 vemos que en este caso la identificación del hindi ha mejorado con respecto al caso anterior al igual que la mayoría de los idiomas, por contra, idiomas que antes se identificaban muy bien como por ejemplo el inglés, ahora ha empeorado su identificación pero estos casos son los mínimos por ello el comportamiento del sistema mejora.

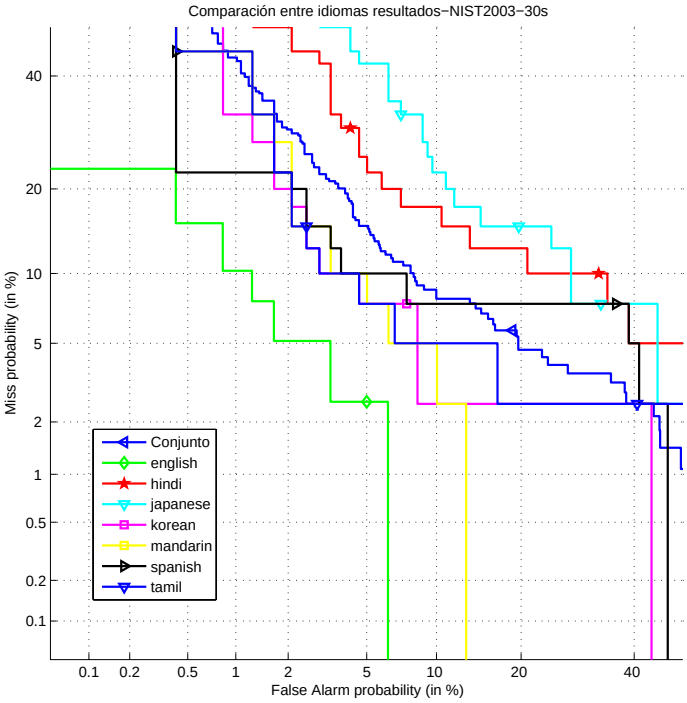

Fig. 34 Resultados por idioma y global para NIST 2003

A la vista de la Fig. 34 y comparándola con la Fig. 32 se observa que el comportamiento global del reconocimiento del inglés sigue siendo de los mejores por el contrario el mandarín que era uno de los que mejor comportamiento tenía, ha empeorado, Lo importante a observar en este caso es la mejora experimentada por el sistema tras la fusión respecto al experimento de Callfriend.

Como se puede comprobar, el nuevo sistema (creado por nosotros con el detector de voz y nuevos reconocedores fonéticos entrenados con SpeechDat como principales mejoras) es mejor que el sistema de partida que tenía un EER de 9,14% . Ahora hemos obtenido un EER del 8,9 %, mejorando por tanto un 2,6%. Destacando, como ya hemos indicado con anterioridad, que el nuevo sistema requiere un menor tiempo de cómputo debido al uso del detector de voz y de Sphinx en la transcripción.

En esta prueba también se ve qué efecto tiene el hecho de no fusionar uno de los reconocedores. En concreto se ve cuál era la precisión del reconocimiento si quitábamos el modelo de español que no había sido entrenado con una base de datos SpeechDat, sino con Albayzin. Se prescinde del reconocedor fonético que está entrenado con habla que no es de telefónica espontánea, por tanto, se prescinde del reconocedor que menos adaptado está a las grabaciones a transcribir Los resultados obtenidos fueron los siguientes:

Idioma	EER-30 s
Inglés	5,12
Hindi	12,97
Japonés	15,00
Coreano	8,78
Mandarín	9,20
Español	10
Tamil	7,5
Global	9,62

Tabla 7 EER NIST2003 sin reconocedor de Albayzin

30-s							
inglés	hindi	japonés	coreano	mandarín	español	tamil	
32	2	1	0	3	0	1	inglés
1	30	4	2	0	0	3	hindi
0	4	28	6	0	0	2	japonés
0	2	5	33	0	0	0	coreano
2	1	2	4	26	3	2	mandarín
0	3	2	1	0	33	1	español
0	4	0	0	0	1	35	tamil

Tabla 8 Matriz de confusión para NIST2003 sin reconocedor de Albayzin

Al comparar Tabla 8 con la Tabla 6 se observa que la mayoría de los idiomas han empeorado con respecto a cuando empleaba Albayzin, su identificación es ligeramente peor, las proporciones de error entre ellos se mantienen y la variación global no es muy significativa.

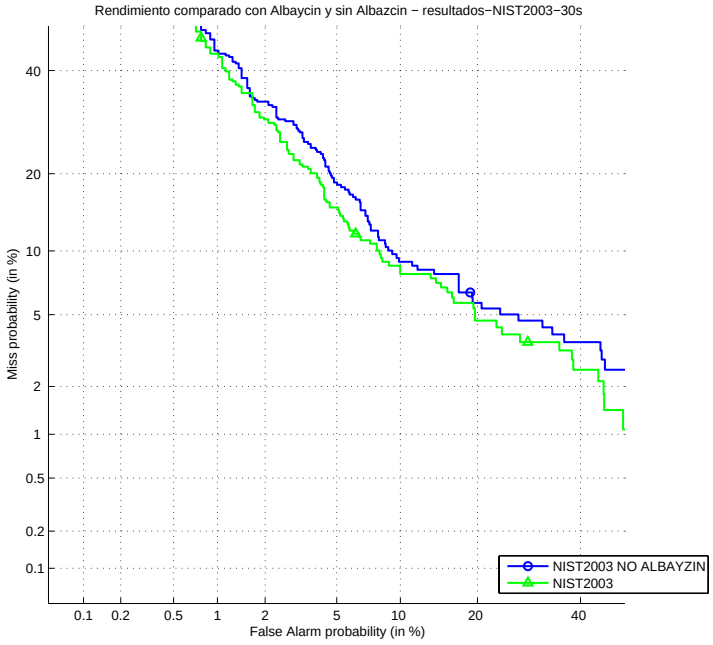

Fig. 35 Comparación en NIST 2003 de la influencia del reconocedor de Albayzin

Como se puede observar en la gráfica, el rendimiento del sistema empeora en este caso con respeto al sistema original, aunque ese empeoramiento no es muy apreciable, por lo que, en determinadas circunstancias, puede considerarse la eliminación de un reconocedor para reducir el coste computacional.

La principal valoración que extrajimos de esta prueba es que en un sistema PPRLM la introducción de más sistemas PRLM provocan la mejora de los resultados, pero esa mejora puede que no compense por el coste computacional que provoca la inclusión de dicho sistema. Por tanto, hay que buscar el equilibrio entre el número de sistemas PRLM a fusionar y la ganancia que ello supone.

NIST2005

Ésta es la prueba más importante de todas, ya que es la que permite comparar el sistema con el resto de sistemas a nivel mundial. Es la última evaluación internacional hecha hasta el momento de escribir esta memoria. Además dadas las condiciones de variabilidad de los archivos de la misma, la evaluación que se hace tras la prueba otorga una buena percepción de si el sistema es factible a nivel práctico, se analiza la robustez frente a la variabilidad de los archivos.

El número de archivos por idioma es diferente, viene definido en NIST LRE 2005. Son los siguientes:

Idioma	**Número de locuciones de 30s**
Inglés	990
Hindi	143
Japonés	365
Coreano	314
Mandarín	973
Español	611
Tamil	183
Otros	84

El hecho de que en la prueba aparezca el conjunto otros quiere decir que hay una serie de archivos que son impostores para todos los archivos de idiomas a reconocer. Esta situación hace que la evaluación tenga más dificultad que las que hemos realizado anteriormente.

Para realizar esta prueba utilizamos los mismos modelos de idioma y de UBM que en las dos pruebas anteriores, parametrizando y pasando por el detector de voz los datos de test de la evaluación de NIST2005. Así que los resultados que se muestran aquí están por debajo de lo realmente óptimo de lo que el sistema puede funcionar, ya que en la evaluación de NIST 2005 podemos emplear también los archivos de test de CallFriend y los archivos de test de NIST LRE 2003, pero en esta prueba no se han empleado.

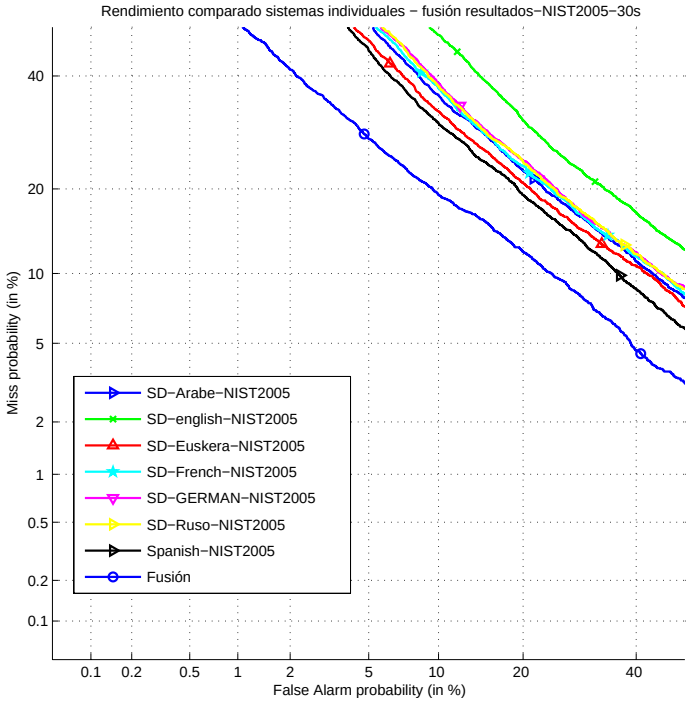

Fig. 36 Fusión de los 7 PRLM con TNorm en NIST 2005

PRLM	EER-30s
SD-Arabe	21,58
SD-english	25,54
SD-Euskera	20,46
SD-French	21,69
SD-German	22,41
SD-Ruso	22,16
Spanish	19,54
Fusión	15,32

Tabla 9 EER de los PRLM para NIST2005

El comportamiento de la fusión es muy bueno ya que, como ve en la gráfica, el rendimiento del sistema tras dicha fusión es mucho mejor que en cada uno de los sistemas por separado. La mejora obtenida es semejante a la que se obtuvo cuando los archivos de test eran los de NIST2003.

El resultado final por idioma es el siguiente:

Idioma	EER-30 s
Inglés	16,26
Hindi	25,17
Japonés	17,59
Coreano	11,26
Mandarín	13,42
Español	11,94
Tamil	13,82
Global	14,10

Tabla 10 EER NIST 2005 por idioma

30-s							
inglés	hindi	japonés	coreano	mandarín	español	tamil	
669	68	42	17	93	32	69	**inglés**
1	67	6	10	22	13	24	**hindi**
3	18	244	27	41	16	16	**japonés**
2	11	65	184	38	5	9	**coreano**
32	30	44	34	808	13	11	**mandarín**
21	23	51	9	12	469	26	**español**
2	10	5	7	17	13	129	**tamil**

Tabla 11 Matriz de confusión de NIST 2005

Como se observa en esta tabla el idioma que mejor se identifica es el coreano, auque debido a que el número de archivos de test de cada idioma es distinto resulta más complicado interpretar estos resultados que en los experimentos anteriores.

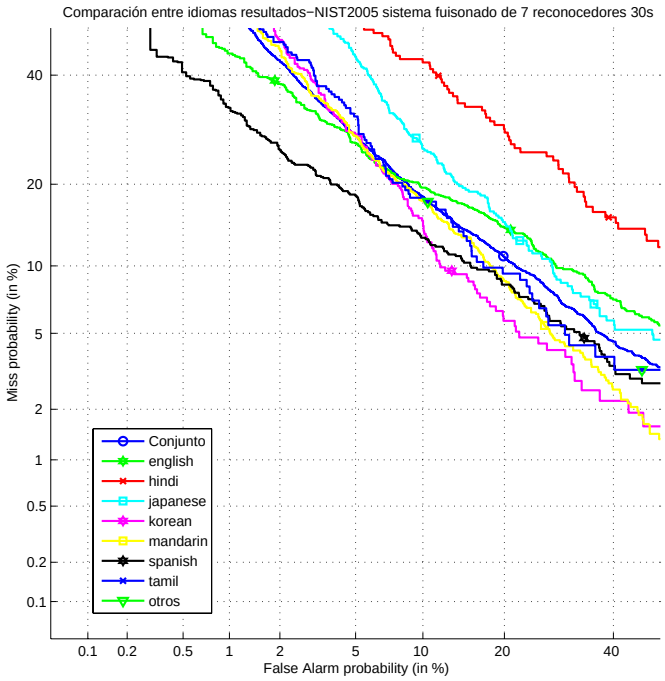

Fig. 37 Comparación por idiomas para NIST 2005

En vista de los resultados la mejora que hemos obtenido con este nuevo sistema es espectacular, puesto que el sistema base tenia un EER del 16,38 % y este sistema está en un EER de 14,1 %. Por tanto, se ha producido una **mejora del sistema del 13,88%**. A lo que hay que unir la mejora de tiempo de procesado por la introducción del detector de voz, que evita procesar segmento de silencio, y de la transcripción con Sphinx. La mejora de estos resultados provocó que parte de la descripción de este sistema se publicase en el articulo D.T. Toledano et al. [2007] de Interspeech, del que también es autor el autor de esta memoria.

Con el fin de afianzar las conclusiones extraídas en la prueba hecha con NIST LRE 2003 al quitar Albayzin, repetimos la prueba pero esta vez poniendo como archivos de test los de NIST LRE 2005. Puesto que ahora lo realizamos con la evaluación al completo las conclusiones que extraigamos de dicha prueba serán más fiables.

Los resultados de dicha prueba se muestran en las siguientes tablas y gráficas:

Idioma	EER-30 s
Inglés	16,16
Hindi	25,07
Japonés	16,44
Coreano	11,78
Mandarín	12,65
Español	11,29
Tamil	13,66
Global	14,34

Tabla 12 EERs para NIST 2005 sin reconocedor Albayzin

30-s							
inglés	hindi	japonés	coreano	Mandarín	español	tamil	
649	77	38	25	89	40	72	inglés
3	66	5	14	15	15	25	hindi
1	19	240	27	42	15	21	japonés
2	9	51	201	36	6	9	coreano
27	23	43	44	805	18	12	mandarín
19	27	49	13	12	460	31	español
3	11	4	7	17	11	130	tamil

Tabla 13 Matriz de confusión NIST 2005 sin reconocedor Albayzin

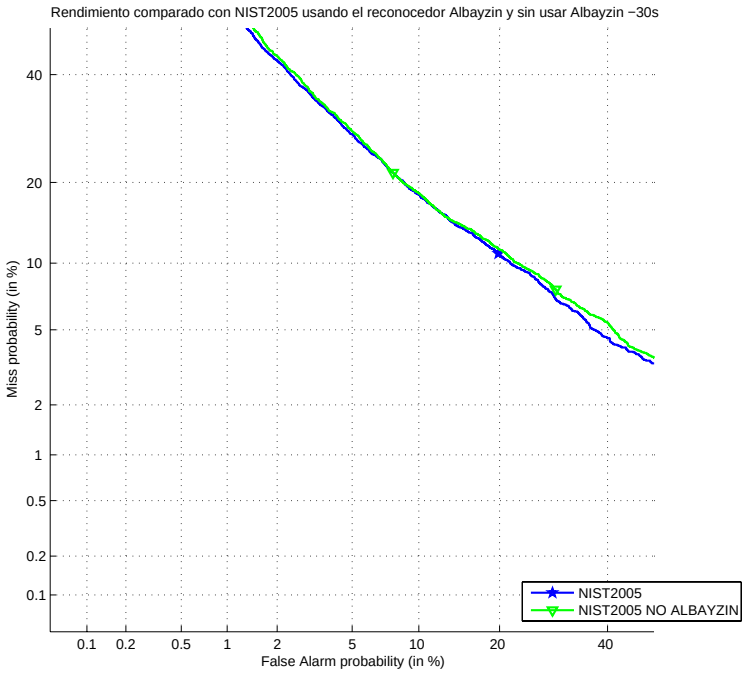

Fig. 38 Comparación NIST2005 con y sin Albayzin

Como se puede observar, el introducir o no en el sistema el reconocedor fonético entrenado con Albayzin no hace variar en exceso el funcionamiento del mismo. Por lo que se puede deducir que la mejora introducida es debida fundamentalmente a los reconocedores fonéticos entrenados con SpeechDat, o al uso del detector de voz. El efecto de esto último se verá en los siguientes experimentos. También se concluye como ya comentamos anteriormente, que la mejora introducida por un reconocedor más es despreciable, por tanto se puede prescindir de él para conseguir una menor carga computacional y consiguiendo una mejora con respecto al sistema base del 12,45%. Ahora esta afirmación tiene una mayor solidez ya que en dos pruebas distintas la conclusión que hemos extraído es la misma.

4.2.2. EXPERIMENTOS PARA ANALIZAR LA INFLUENCIA DEL DETECTOR DE VOZ

El objetivo de estos experimentos es comprobar la influencia del uso del detector de voz; para ello el procedimiento seguido fue el mismo que se seguía en el sistema base [Alberto Montero-Asenjo et al. 2006], y con los mismos reconocedores fonéticos de éste que son los entrenados con OGI, pero introduciendo el detector de voz que empleamos en el sistema final antes de usar los ficheros de parámetros para entrenar o reconocer. De esta forma, los ficheros de entrenamiento y de test son sólo de voz, no incluyen silencios.

Para la realización de esta prueba partimos del sistema base, lo adaptamos para que antes de transcribir fonéticamente se pase un detector de voz que modifica los archivos de parámetros, quitando aquellos parámetros pertenecientes a ventanas donde hay silencio.

Los experimentos se realizaron tanto para un subconjunto de NIST LRE 2003, como para toda la evaluación de NIST LRE 2005.

NIST2003

En esta prueba al igual que en las anteriores los modelos se entrenaron con CallFriend, pero a diferencia de los anteriores entrenamiento, se realizó sobre las transcripciones extraídas de emplear los reconocedores fonéticos de OGI. Los resultados obtenidos para este caso son los siguientes:

Idioma	EER-30 s
Inglés	5,12
Hindi	10
Japonés	17,15
Coreano	10,46
Mandarín	10
Español	10
Tamil	5
Global	10,04

Tabla 14 EERs sistema base con detector de voz NIST 2003

30-s							
inglés	hindi	japonés	coreano	mandarín	español	tamil	
36	0	0	0	3	0	0	**inglés**
0	29	3	3	1	3	1	**hindi**
1	2	27	7	1	1	1	**japonés**
2	0	7	28	1	0	2	**coreano**
1	0	3	3	28	3	2	**mandarín**
3	2	1	2	0	30	2	**español**
3	3	1	0	0	0	33	**tamil**

Tabla 15 Matriz de confusión del sistema base con detector de voz

Comparando la Tabla 15 con la Tabla 6 se observa que el sistema funciona peor que el nuevo sistema con detector de voz y nuevos reconocedores. Aunque hay idiomas que tienen menor tasa de falso rechazo, aumenta la falsa aceptación de los mismos. La mejora de los resultados viene dada por la mejora de los reconocedores.

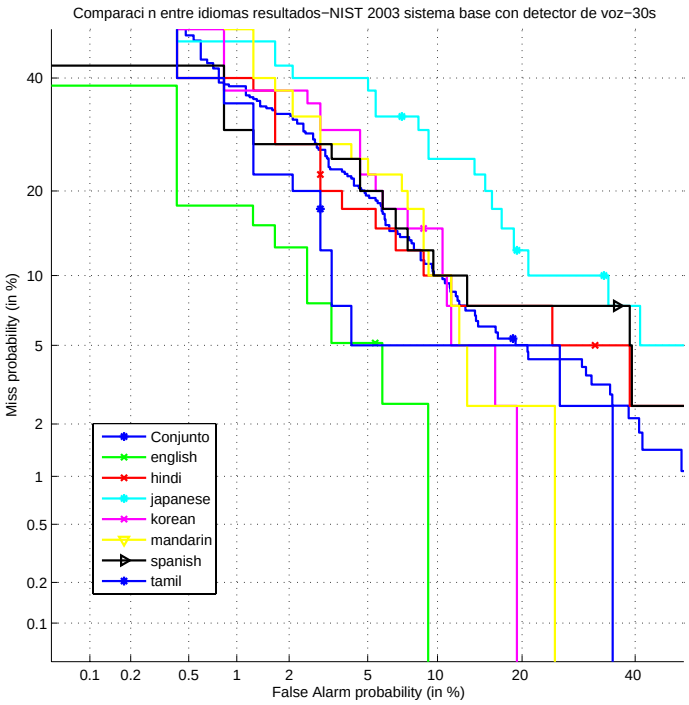

Fig. 39 Comparación por idioma NIST 2003 sistema base con detector de voz

Como se puede ver, el sistema funciona prácticamente igual con el detector de voz, así que lo conservamos porque supone un gran ahorro en coste computacional puesto que los segmentos de no voz ya no se procesarán más.

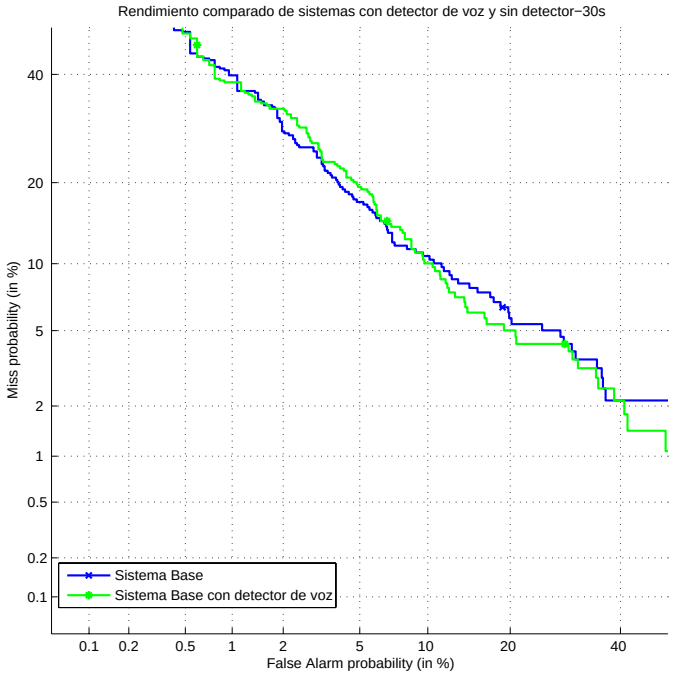

Fig. 40 Comparación sistema con detector de voz y sin detector en NIST 2003

NIST2005

Para dar una mayor fiabilidad a los resultados de la prueba anterior repetimos la prueba anterior pero como archivos de test ahora serán los de la evaluación de NIST LRE 2005, pero se usan los mismos modelos de idioma y de UBM que se construyeron en la anterior prueba. Los resultados de esta prueba se resumen en las siguientes tablas y gráfica:

Idioma	EER-30 s
Inglés	17,39
Hindi	27,27
Japonés	17,81
Coreano	12,74
Mandarín	17,81
Español	15,67
Tamil	18,58
Global	16,38

Tabla 16 EERs sistema base con detector de voz NIST 2005

30-s							
inglés	hindi	japonés	coreano	mandarín	español	tamil	
678	86	66	26	84	26	24	**Inglés**
13	74	4	14	25	10	3	**Hindi**
12	18	227	29	41	34	4	**japonés**
7	14	39	215	36	3	0	**coreano**
72	41	80	45	704	18	12	**mandarín**
33	40	66	24	35	402	11	**español**
13	14	3	9	22	21	101	**Tamil**

Tabla 17 Comparación por idioma NIST 2005 sistema base con detector de voz

Haciendo una comparación de la Tabla 16 y la Tabla 17 con la Tabla 10 y la Tabla 11 vemos que el comportamiento es peor, esto es debido fundamentalmente a que estamos usando reconocedores fonéticos distintos. Puesto que los principales cambios en este acaso es el uso de diferentes reconocedores, se pude concluir que la mejora en el EER del sistema viene dada por la mejora de los reconocedores fonéticos que se emplean en el nuevo sistema, como ya se indicaba en Matejka Pavel ,et al. [2005].

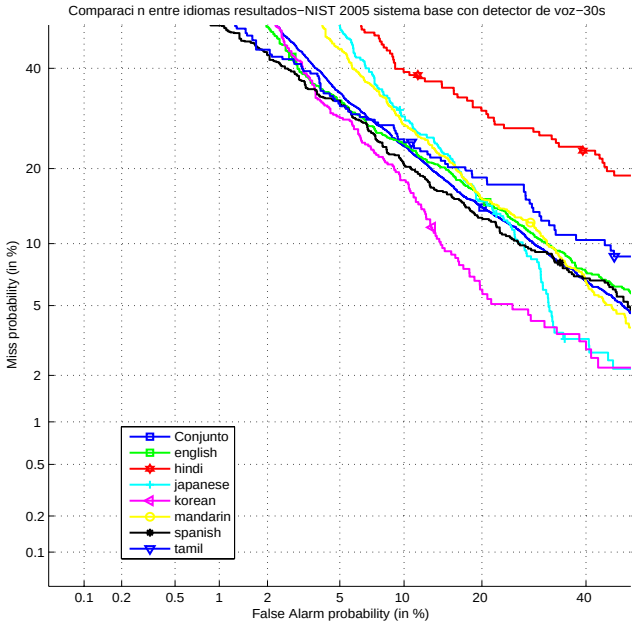

Fig. 41 Comparación por idioma NIST 2005 sistema base con detector de voz

En vista de la comparación hecha en la Fig. 42 entre el sistema base con y sin detector de voz, se extrae que el comportamiento en este caso es ligeramente mejor con el detector de voz que sin él, , sobre todo si nos fijamos en el valor del EER, al contrario de lo que sucedía en la prueba anterior en donde era ligeramente peor. Esto es debido a que la evaluación de NIST 2005 no tenía un detector previo de voz, es decir, que las grabaciones no estaban ya filtradas a diferencia de lo que sucedía en la de NIST 2003 que sí lo estaban

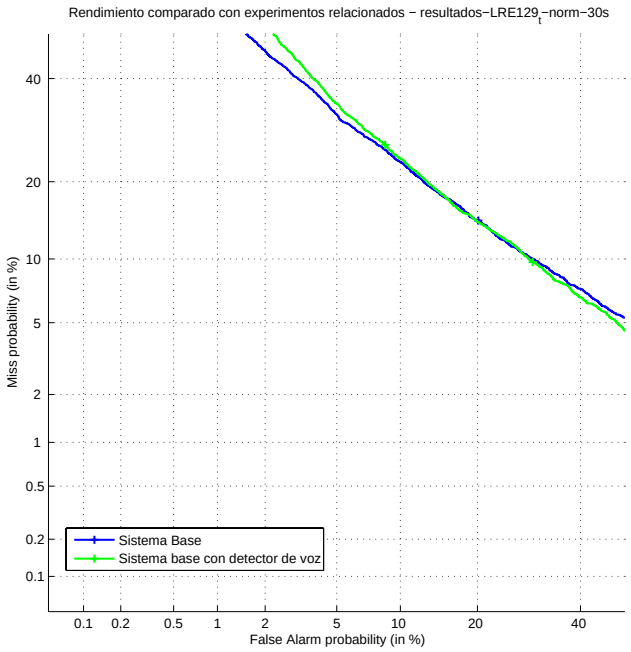

Fig. 42 Comparación sistema con detector de voz y sin detector en NIST 2005

Por tanto, de estos experimentos podemos concluir que el efecto del detector de voz mejora el sistema en cuanto al coste computacional, al no procesarse los segmentos de no voz, obteniendo no obstante resultados de reconocimiento de idioma semejantes a cuando no se empleaba. Estos mismos resultados y conclusiones se publicaron en el articulo de Interspeech de D.T. Toledano et al. [2007].

4.2.3. EXPRIMENTOS SIN HTK

El objetivo es eliminar las herramientas de HTK del proceso de reconocimiento de idioma, para ello se van usar una serie de programas, creados en el grupo ATVS y que se basan en los lattices que se extraen cuando Sphinx hace reconocimiento fonético. Aunque lo que realmente se hace es coger el mejor camino, siguiendo el algoritmo de Viterbi. El objetivo de dejar de usar HTK viene dado por problemas de licencias ya que sólo se puede emplear para uso de investigación pero no para sistemas de utilidad industrial como indicamos en la sección 3.1.2 de esta memoria.

Este es el sistema definitivo, en el cual ya se han aplicado todos los cambios que se indicaban en la sección 3.2.2. Por tanto, el procedimiento que tendremos ahora en el sistema es el siguiente:

1) **Prametrizar** todo el audio a MFCC con Sphinx.

2) **Aplicar el detector de voz**, modificando los archivos de parámetros para tener sólo los parámetros de los segmentos de voz

3) **Transcribimos con Sphinx** tanto los archivos de entrenamiento como los de test, también tendremos los lattices, es decir, el conjunto de transcripciones posibles para esos parámetros con la probabilidad de las mismas.

4) **Calcular los n-gramas**. Con los lattices calculamos un n-grama con el camino más probable, hasta el nivel de trigramas, para todos los archivos de entrenamiento perteneciente a un mismo idioma, dando lugar a los modelos de idioma.

5) **Construir el UBM** como la combinación suma de todos los modelos de idioma.

6) **Construimos un n-grama hasta el nivel 3-grama,** uno por cada archivo de test.

7) **Reconocer**: se comparan el n-grama del archivo de test con el n-grama del modelo de idioma construido, además del modelo de UBM, dando en principio igual peso al UBM que al modelo de idioma, luego se hicieron pruebas para buscar el óptimo.

Las pruebas hechas aquí han utilizado solo los datos de NIST 2005, pero se han realizado pruebas con diferente cantidad de datos de entrenamiento y otras con diferentes pesos entre el modelo de idioma y el modelo UBM.

NIST2005

Los resultados que obtuvimos con este procedimiento para cada uno de los PRLM y la fusión de los mismos para los datos de NIST2005 (resultados con trigramas) son los siguientes:

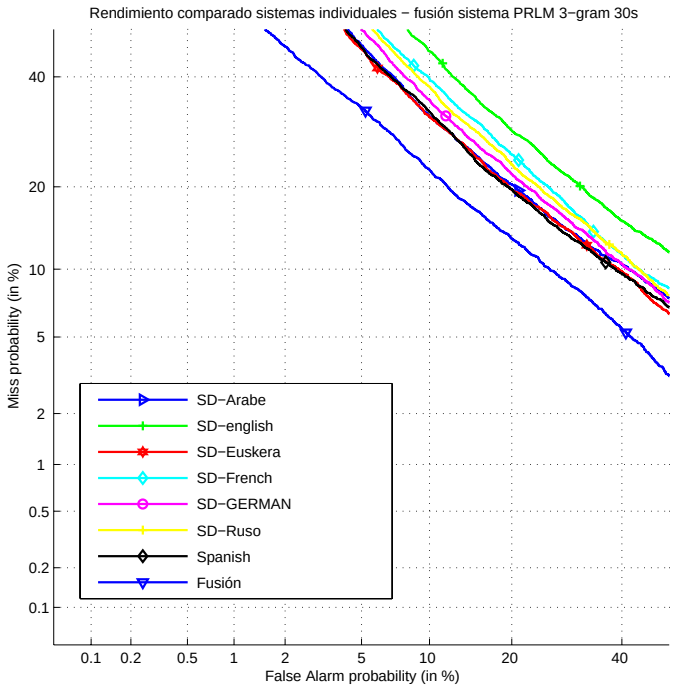

Fig. 43 Fusión de PRLM para NIST 2005 sin HTK

PRLM	EER-30s
SD-Arabe	20,23
SD-english	25,03
SD-Euskera	19,79
SD-French	22,68
SD-German	21,02
SD-Ruso	21,74
Spanish	19,75
Fusión	15,96

Tabla 18 EER de los PRLM para NIST2005 sin usar HTK

Como se observa, la fusión de los sistemas ha sido beneficiosa ya que se consigue mejorar bastante el rendimiento del sistema, esto es debido a que la información que aporta cada uno de los reconocedores es muy diversa y su fusión hace que mejore el sistema. Pero lo más importante es que la gráfica y tabla anteriores son muy semejantes a las obtenidas cuando se empleaba HTK (Tabla 9 y Fig. 36), son resultados prácticamente iguales, que es lo que buscábamos. Porque buscamos la sustitución de HTK por Sphinx y programas propios sin que ello suponga una merma en el rendimiento o en todo caso tener mejores resultados.

Los resultados por idioma a reconocer una vez hecho el TNorm de la fusión son los siguientes:

Idioma	**EER-30 s**
Inglés	15,26
Hindi	27,27
Japonés	16,65
Coreano	13,39
Mandarín	15,02
Español	15,20
Tamil	15,85
Global	15,12

Tabla 19 Comparación por idioma NIST 2005 sistema sin HTK

30-s							
ingles	**hindi**	**japonés**	**coreano**	**mandarín**	**español**	**tamil**	
686	95	47	16	78	15	53	**inglés**
6	64	4	11	26	10	22	**hindi**
3	32	228	25	50	11	16	**japonés**
0	26	72	173	34	2	7	**coreano**
33	45	65	23	789	9	8	**mandarín**
33	52	71	18	22	387	28	**español**
4	21	8	4	17	5	124	**tamil**

Tabla 20 Comparación por idioma NIST 2005 sistema sin HTK

De las tablas anteriores si las comparamos con la Tabla 10 y la Tabla 11, observamos que el comportamiento es prácticamente igual que cuando usamos HTK, como ya se apreciaba en la fusión del sistema.

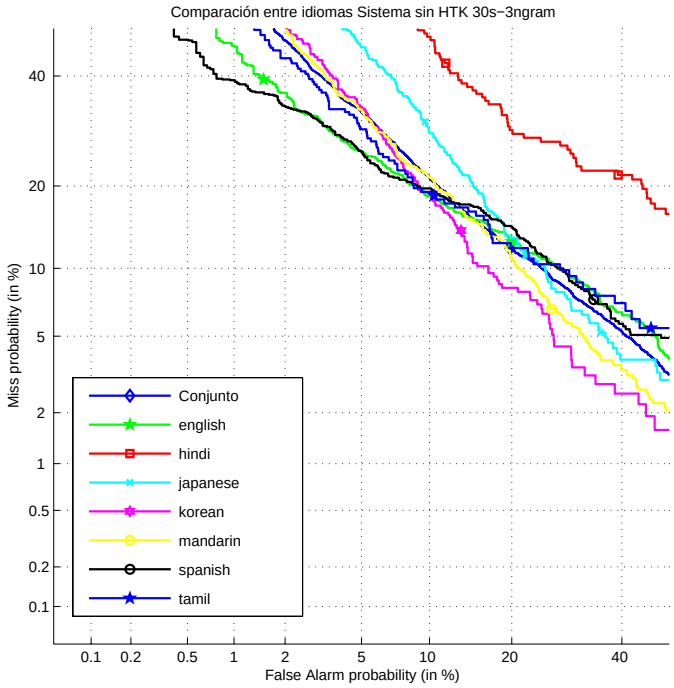

Fig. 44 Comparación por idioma NIST 2005 sistema sin HTK

Cabe destacar por el valor de EER, que el nuevo sistema tiene un rendimiento menor que en el caso de usar HTK, pero este menor rendimiento se debe fundamentalmente a que en esta prueba todavía no están ajustados los pesos entre el UBM y el modelo de idioma, mientras que en HTK si lo estaban. Además también hay que tener en cuenta que los resultados mostrados en esta sección solo usan la información de trigramas, a diferencia de cuando se usa HTK que emplea también la información de los bigramas y los unigramas

Para mejorar el comportamiento, lo primero que hicimos fue introducir más datos de entrenamiento, se introdujeron los datos de evaluación de CallFriend y los datos de la evaluación de NIST 1996, consiguiendo un EER del 14,7%

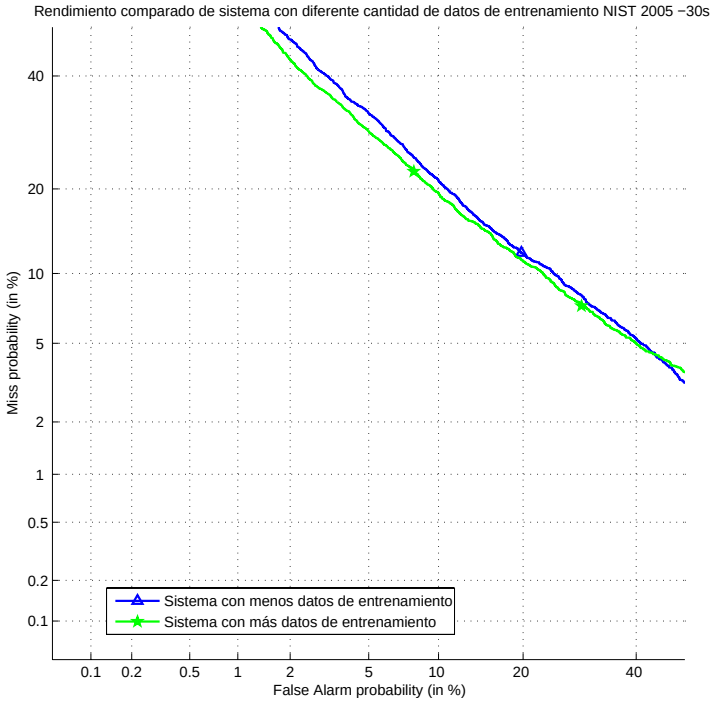

Fig. 45 Comparación de rendimiento entre sistemas con más o menos datos de entrenamiento

Se observa una mejora significativa con el aumento de los datos de entrenamiento. Para seguir mejorando el sistema, lo que se hizo fue un estudio con diferentes pesos entre el UBM y el modelo de idioma a la hora de calcular la puntuación para el fichero de prueba. Hay que destacar que dicho estudio fue posible realizarlo con pruebas con la evaluación completa, en lugar de un subconjunto de la misma, dada la mejora en tiempo de procesado que obtuvimos con este sistema. Además también ayudo a aumentar la velocidad de las pruebas que la creación de los n-gramas de los modelos de idioma y los n-gramas de los ficheros de evaluación sólo se realiza una vez.

Peso UBM	Peso modelo de idioma	EER
50%	50%	14,70
33,3%	66,6%	14,56
66,6%	33,3%	14,09
83,3%	16,6%	14,00
90,9%	9,1%	13,99

Tabla 21 Comparación distintos pesos

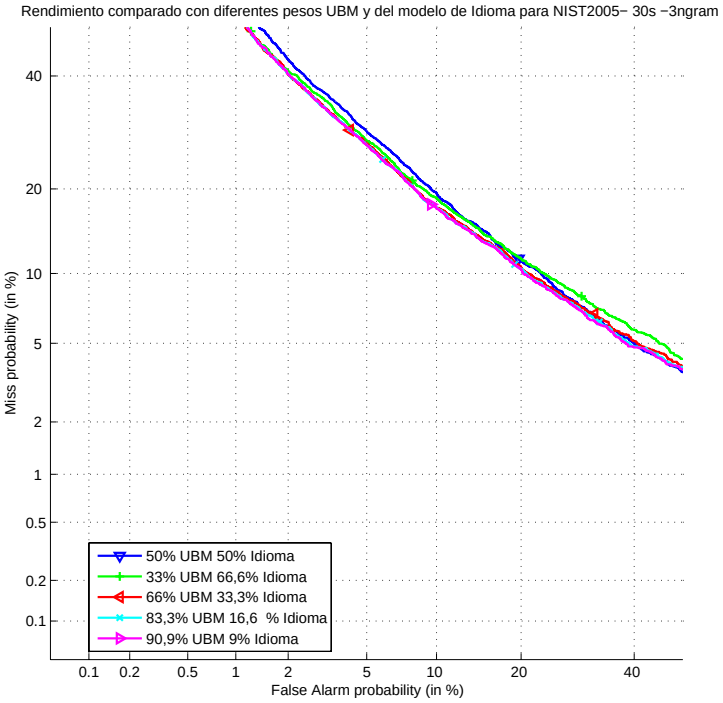

Fig. 46 Comparación diferentes pesos

Los cambios realizados aumentando el número de datos de entrenamiento y poniendo unos pesos adecuados, han hecho que el sistema tenga un rendimiento ligeramente mejor que el sistema que usaba HTK, por tanto, hemos conseguido que se elimine del proceso el uso de HTK que era uno de los objetivos de este sistema, consiguiendo a la vez mejorar el sistema respecto al sistema con HTK y bastante con respecto al sistema de partida publicado en Alberto Montero-Asenjo et al. [2006], mejorando con respecto a este un 14,6 % en EER.

En esta prueba todos los resultados expuestos anteriormente son de trigramas, como se ha indicado con anterioridad, esto es debido a que se estudio la fusión con bigramas y unigramas pero los resultados de la fusión suma apenas mejoraban. En parte, debido a la alta correlación existente entre los resultados obtenidos con unigramas, bigramas y trigramas como se ve en la Fig. 47 las puntuaciones ocupan en los tres casos la misma región. La causa es que la información que contienen los trigramas contiene a su vez la información de los bigramas y de los unigramas.

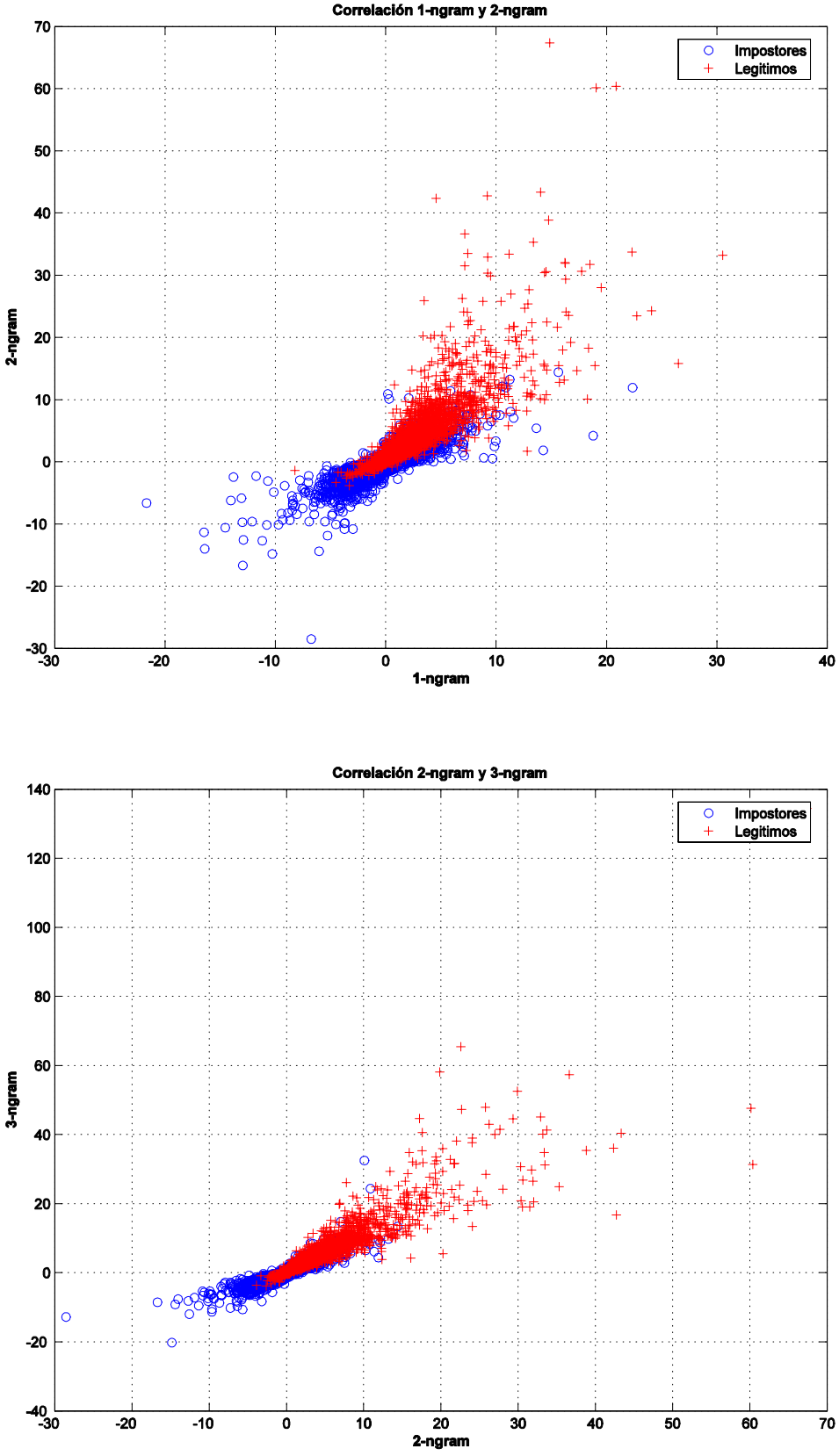

Fig. 47 Relación de puntuaciones entre unigramas, bigramas y trigramas

5. CONCLUSIONES

5.1. CONCLUSIONES SOBRE LOS RECONOCEDORES FONÉTICOS

En estos experimentos se puede ver el comportamiento general de los modelos fonéticos, con respecto al número de Gaussianas y las diferencias entre los diferentes sistemas de reconocimiento.

A la vista de los resultados las conclusiones que extraemos son las siguientes:

- El aumento del número de mezclas de Gaussianas para el modelado de cada uno de los estados, provoca un aumento en el porcentaje de fonemas acertados, de igual forma que se da un mayor incremento en la precisión de los sistemas. Por tanto, **el aumento de las Gaussianas se hace con el fin de aumentar la exactitud del sistema**, siendo capaz de acertar más sin que para ello tenga que introducir fonemas de más, que es el objetivo primordial de un reconocedor fonético.

- De la comparación entre HTK y Sphinx, deducimos que HTK tiene un mayor porcentaje de aciertos, pero que dicho porcentaje es debido al aumento del número de inserciones. Mientras Sphinx obtiene un menor número de aciertos, pero tiene una menor cantidad de inserciones, como consecuencia del ajuste realizado en el factor de penalización de inserción de palabras, cuyo valor se determino después de un estudio. Esta característica hace que para muchas aplicaciones sea mejor usar Sphinx para extraer transcripciones fonéticas, ya que no introduce tantos fonemas espurios como HTK. Esto es especialmente importante en técnicas de reconocimiento idiomático como se ha visto en la construcción de los PRLM.

Las diferencias entre uno y otro idioma vienen dadas por la cantidad de fonemas que se tengan en dicho idioma y las diferencias entre dichos fonemas. De ahí que algunos tengan un porcentaje de acierto superior al 50%, mientras que otros no superan el 40 %.

La principal conclusión es que se han realizado reconocedores fonéticos que aunque pueden ser mejorables, son suficientemente buenos como para desarrollar reconocedores de idioma en el estado del arte como se ha visto en 4.2. Además estos reconocedores fonéticos han supuesto un gran avance con respecto a los reconocedores de los que partimos, TIMIT y Albayzin, tanto por la cantidad de datos de entrenamiento que superan en mucho a los reconocedores antes nombrados; como por las características, ya que es habla telefónica espontánea adaptándose por completo a las características de las grabaciones en las que se realizar las transcripciones fonéticas.

5.2. CONCLUSIONES SOBRE LOS RECONOCEDORES DE IDIOMA

Las conclusiones comunes que extraemos de todos los experimentos realizados sobre el sistema PPRLM construido son las siguientes:

- La fusión de varios PRLM hace que el rendimiento mejore considerablemente y más cuanto más incorrelada este la información que de cada PRLM. Se ha realizado un estudio para determinar que la fusión de más PRLM produce mejores resultados, pero a partir de un cierto número de sistemas fusionados, la mejora que produce la inclusión de otro PRLM no compensa el coste computacional que conlleva. Esto se comprobó viendo la influencia que tenía en la fusión el reconocedor de Albayzin.

- También se vio que el uso de TNorm mejoraba los resultados, determinando que el orden de aplicarlo era en cada sistema PRLM y después de realizar la fusión del sistema.

- El uso de reconocedores fonéticos entrenados con una mayor cantidad de audio como son los de SpeechDat hace mejorar el funcionamiento de cada PRLM, y por extensión el comportamiento global del sistema PPRLM. Esto se observa en los primeros experimentos donde se mejoraba un 13,88% su EER con respecto al sistema base.

- El uso del detector de voz no mejora significativamente el rendimiento de los sistemas en cuanto a valor de EER, como se comprobó al introducir en el sistema base el detector de voz, que mejoraba algo en el caso de NIST2005. Pero se consigue una mejora en el coste computacional del sistema. Esta conclusión y estudio se reflejo junto con parte del sistema que usaba los nuevos reconocedores fonéticos en un articulo de Interspeech 2007 del que el autor de esta memoria es coautor (D.T. Toledano et al. [2007]).

- La conclusión más importante es que hemos mejorado el sistema PPRLM de reconocimiento de idioma y lo hemos puesto a punto para utilizarlo en la evaluación NIST LRE 07,, consiguiendo con el aumento de datos de entrenamiento, y ajuste de pesos entre el UBM y el modelo de idioma que se tenga un sistema con **mejora global del 14,6% en el EER** con respecto al sistema base. Se perfeccionaba el sistema en cada uno de los cambios realizados como se puede ver en la siguiente gráfica:

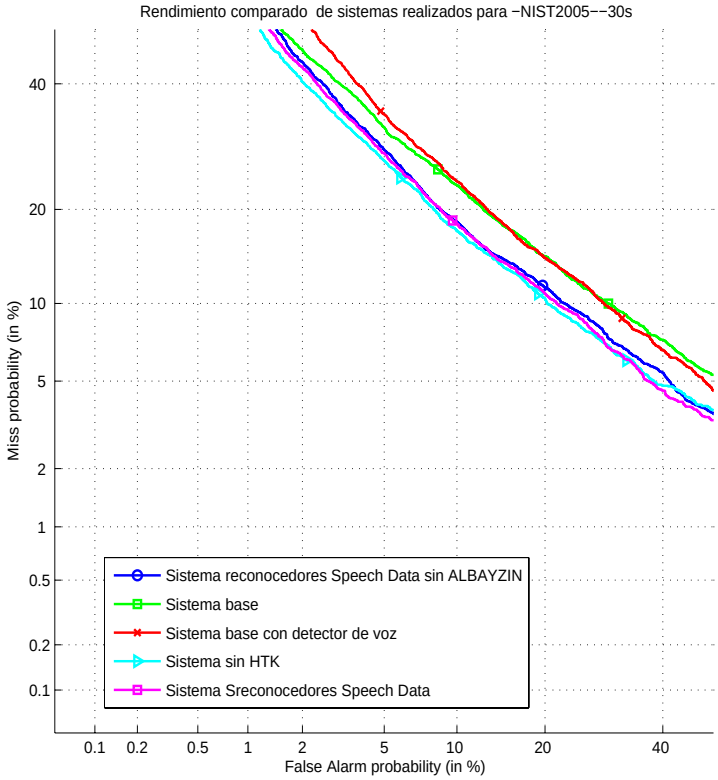

Fig. 48 Comparación de todos los sistemas construidos

- Se ha conseguido sustituir en el sistema el uso de HTK por el uso de programas propios del grupo, los cuales construyen n-gramas a partir del mejor camino del lattice, posibilitando en el futuro técnicas que usen información de lattices que hasta ahora no se han empleado. Consiguiendo además una mejora con respecto al sistema base del 14,6% en EER.

- Los modelos fonéticos y la extracción de lattices y de los n-gramas que se desarrollaron en este proyecto llevo a que se implementase en el grupo una novedosa técnica como es Phone-SVM [W.M.Campbell et al, 2007], obteniéndose unos resultados muy buenos para el reconocimiento de idioma con un EER global de 8,20 %. Esto supone una mejora relativa de en torno al 50% respecto al sistema base de partida.

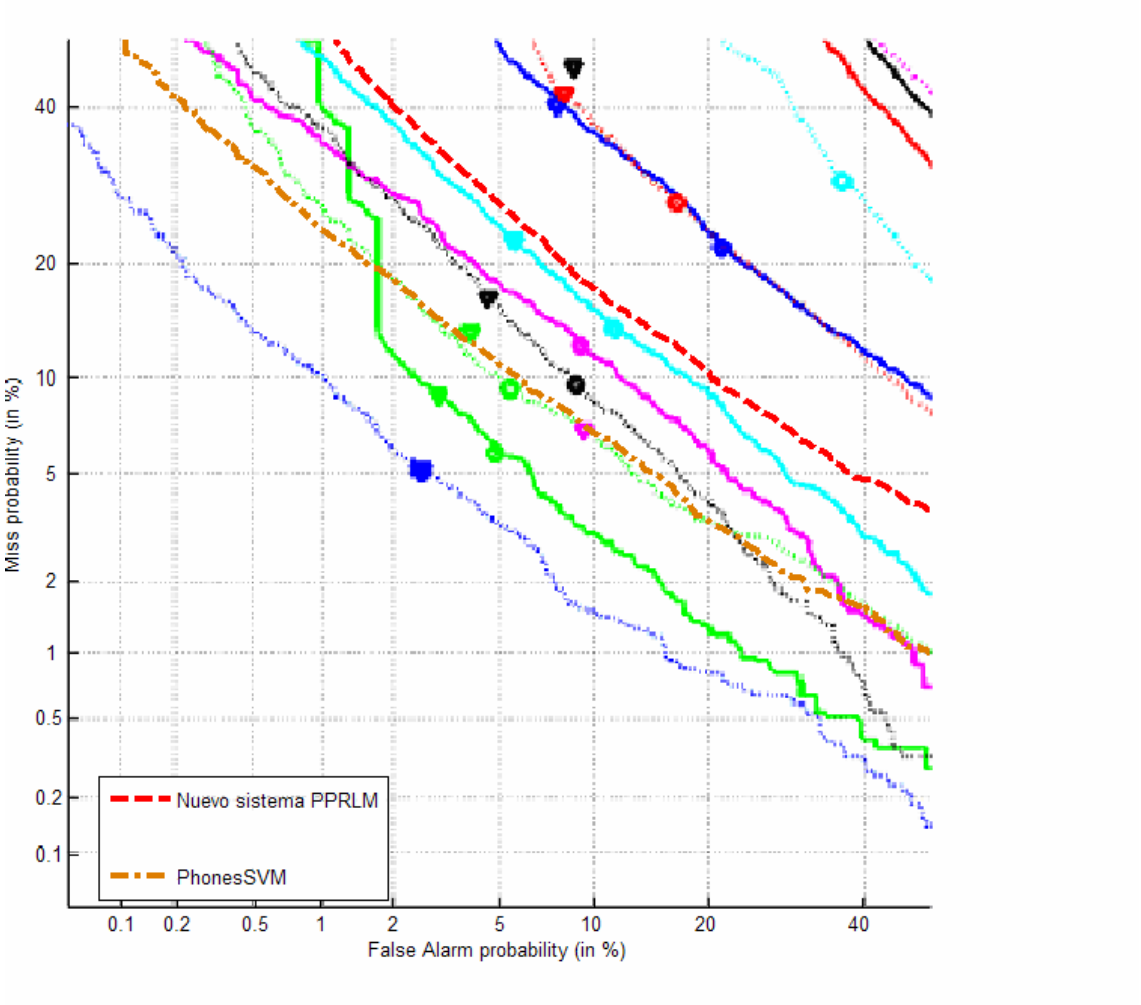

Fig. 49 Comparación del nuevo sistema con respecto al resto de sistemas de la evaluación de NIST 2005.

- Como se puede ver en la gráfica anterior el sistema PPRLM creado sería el 7º del mundo si se hubiese presentado a la evaluación de NIST LRE 2005 y estando en el grupo de los mejores sistemas, pero es que el sistema de Phone-SVM desarrollado a partir de los modelos fonéticos entrenados en este proyecto y los lattices y el programas de generación de n-grmas, hubiese sido el 4º mejor sistema del mundo (y muy próximo al 3º). Por tanto, no sólo se ha desarrollado un sistema competitivo a nivel mundial, sino que alguno de los desarrollos de este proyecto han posibilitado el desarrollo de técnicas aun más competitivas como los Phone-SVM.

6. TRABAJO FUTURO

Con el sistema PPRLM implementado se ha conseguido una mejora del rendimiento del sistema del 14,6 % y el cambio de usar transcripciones al comenzar a usar la información de lattices; y se ha creado unos transcriptores fonéticos de mayor calidad. Por tanto, el trabajo futuro que se impone como fruto de este proyecto se presenta en dos vertientes:

- La creación de mejores reconocedores fonéticos posibilita la creación de mejores sistemas de reconocimiento de locutor dependiente de texto, además de mejores transcriptores automáticos. Por tanto la creación de los reconocedores fonéticos con SpeechDat posibilita la mejora de sistemas que se emplean en actividades distintas a la identificación de idioma. Una línea de trabajo futuro que se abre es en otros sistemas de tratamiento de la información en la voz.

- Por otro lado, en idioma se abren por el nuevo sistema nuevas posibilidades de investigación, ya que la sustitución de las herramientas de HTK por programas propios que leen del lattice que saca Sphinx posibilita nuevos sistemas PPRLM basados en lattices [J.L. Gauvain,, et al. 2004]. O otros sistemas que emplean partes de este sistema es el caso de los Phone-SVM que tan buen resultado han dado. Además la estructura de experimento creada y el sistema en sí, servirá como parte del sistema que se presentara a la evaluación NIST 2007 de idioma.

Además, una futura línea de investigación en el reconocimiento de idioma está apuntando hacia los sistemas de alto nivel, para los cuales es necesario los reconocedores fonéticos que aquí se construyeron; al igual que los principio y conceptos que se han explotado en la creación del PPRLM y las mejoras del mismo.

Por tanto la creación del nuevo sistema pese a basarse en una idea clásica en el reconocimiento de idioma, como son los PPRLM, ha posibilitado nuevas líneas de investigación y ha proporcionado mejoras en otros sistemas distintos al reconocimiento de idioma, además de constituir por si mismos unos sistemas competitivos de cara a la próxima evaluación de NIST LRE 2007.

7. REFERENCIAS

K. Atkinson, *Language Identification from Nonsegmental Cues*, Journal of the Acoustical Society of America, 44:378(A), 1968.

CallFriend corpora, available for purchase from the Linguistic Data Consortium (LDC) on http://www.ldc.upenn.edu/Catalog/byType.jsp#speech, catalog codes: LDC96S46 to LDC96S60.

W. M. Campbell. *Generalized linear discriminate sequence kernels for speaker recognition*. Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vol. 1, pp. 161-164, 2002.

W.M.Campbell, J.P . Campbell, D.A.Reynolds, E. Singer, and P. A. Torres-Carrasquillo *Suport Vector machine for speaker and language recognition*, Coumputer Speech and Language, Vol 20, no 2-3pp 210-229, 2006

W.M. Campbell, Richardson, F. Reynolds, D.A. *Language Recognition with Word Lattices and Support Vector Machines*, in International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vol 4, pp 989-992, 2007-.

Carnegie Mellon University SPHINX speech recognizer, available on http://sourceforge.net/projects/cmusphinx/

The CMU Pronouncing Dictionary, available on http://www.speech.cs.cmu.edu/cgi-bin/cmudict .

R. O. Duda, P. E. Hart y D. G. Store. *Patter Classification*. Wiley. 2001.

ETSI ES 202 050 (v1.1.3): "*Speech processing, transmisión and quality aspects (STQ); Distributed speech recognition; Advanced front-end features extraction algorithm; Compression algorithms*."

J.L. Gauvain, A. Messaoudi, and H. Schwenk, "*Language Recognition using Phone Lattices*", in Proc. ICSLP 2004.

Gil, Juana: *Los sonidos del lenguaje*. Madrid, Síntesis, 1989.

Gleason T.P. Campbell W.M. Reynolds D.A. Singer E., Torres-Carrasquillo P.A., *Acoustic,phonetic,and discriminative approaches to automatic language identification*, in Proc. Eurospeech 2003, Sept. 2003, pp. 1345–1348.

T.J.Hazen and V.W.zue, *Recent improvements in an approach to segment-based automatic language identification* in Proc. ICASSP'94, vol 4, Sept, 1994, pp 1883-1886

Hidden Markov Model ToolKit (HTK), available on http://htk.eng.cam.ac.uk/ .

Huang, Xuedong, *Spoken language processing a guide to theory, algorithm, and system development*, Edit. Prentice Hall PTR año 2001, ISBN 0130226165, pp 377-415

ICASSP 2007 International Conference on Speech and Audio Processing website http://www.icassp2007.org/

F.Jelinek, A,Waibel and K.-F. Lee, Eds Palo alto, CA: Mongan Kaufman, *Self-organized language modelling in for Speech recognotiom* in Readings in Speech Recognition, 1990 pp 450-506

Li Deng, Don Yu, A. Acero, *Structured Speech Modeling*, in IEEE transactions on Audio, Speech and language processing, Vol 14 Nº 5, septiembre 2006 pp 1501.

Llisterri, Joaquim: *Introducción a la fonética: el método experimental*. Barcelona, Anthropos, 1991.

A. Martin, G. Doddington, T. Kamm, M Ordowski, M. Przybocki, *The DET Curve in Assessment of Detection Task Performance*, in Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH), 1997

Matejka Pavel, Schwarz Petr, Cernocký Jan, Chytil Pavel, "*Phonotactic Language Identification using High Quality Phoneme Recognition*", In: Interspeech'2005 - Eurospeech - 9th European Conference on Speech Communication and Technology, Lisbon, PT, 2005, p. 2237-2240.

Y. Muthusamy, R. Cole and B. Oshika, *The OGI Multi-Language Telephone Speech Corpus,* in Proceedings of the International Conference on Spoken Language Processing (ICSLP), 1992, pp. 895-898.

Y. K. Muthusami, N. Jain, R. A. Cole, *Perceptual Benchmarks for Automatic Language Identification*, in Proceedings of the IEEE International Conference on Speech and Audio Processing (ICASSP), Adelaide, Australia, 1994a.

Y. K. Muthusamy, E. Barnard, R. A. Cole, *Reviewing Automatic Language Identification*, in IEEE Signal Processing Magazine, October 1994b, pp. 33-41.

Alberto Montero-Asenjo, Doroteo T. Toledano, Javier Gonzalez-Dominguez, Joaquin Gonzalez-Rodriguez, and Javier Ortega-Garcia, *Exploring PPRLM performance for nist 2005 language recognition evaluation*, in Proceedings of Odyssey06: The speaker and language recognition workshop, 2006.

N. Morales Mombiela, *Robust Speech Recognition Under Band-Limited Channels And Other Channel Distortions*, PhD Thesis June 2007,pp 49

A. Moreno, D. Poch, A. Bonafonte, E. Lleida, J. Llisterri, J. Mariño, C. Nadeu, *ALBAYZÍN Speech Database: Design of the Phonetic Corpus, i*n proceedings of the 3rd European Conference on Speech Communication and Technology *(EUROSPEECH)*. Berlin, Germany, 21-23 September 1993. Vol. 1. pp. 175-178.

NIST 1996 Language Recognition Evaluation website: http://www.nist.gov/speech/tests/lang/1996

NIST 2003 Language Recognition Evaluation website: http://www.nist.gov/speech/tests/lang/2003

NIST 2005 Language Recognition Evaluation website: http://www.nist.gov/speech/tests/lang/2005/

NIST Rich Transcription 2004 Fall Evaluation website, http://www.nist.gov/speech/tests/rt/rt2004/fall/ .

NIST Rich Transcription 2005 Spring Evaluation website, http://www.nist.gov/speech/tests/rt/rt2005/spring/
.
OGI multi language telephone speech," http://www.cslu.ogi.edu/corpora/mlts/.

L. R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, in Proceedings of the IEEE , vol. 77, nº 2, February 1989, pp. 257-286.

D. Reynolds, T. Quatieri, and R. Dunn, *Speaker verification using adapted gaussian mixture models*, Digital Signal Processing, vol. 10, pp 19-41, 2000.

Shen, W., Campbell, W., Gleason, T., Reynolds, D., Singer, E., "Experiments with Lattice-based PPRLM Language Identification", in Proc. IEEE Odyssey 2006, Puerto Rico, June 2006.

P. A. Torres-Carrasquillo, T. P. Gleason D. A. Reynolds*, Dialect Identification using Gaussian Mixture Models*, In Proc. Odyssey: The Speaker and Language Recognition Workshop in Toledo, Spain, ISCA, pp. 297-300, 31 May - 3 June 2004.

D. T. Toledano, E. Campos, A. Moreno, J. Colás, J. Garrido, *Resultados preliminares de decodificación fonética sobre distintos tipos de habla espontánea*, in Proceedings III Jornadas de Tecnología del Habla, 17-19 Noviembre 2004, Valencia, Spain, pp. 227-232.

D. T. Toledano, A. Moreno, J. Colás, J. Garrido, *Acoustic-phonetic decoding of different types of spontaneous speech in Spanish*, in Proceedings of the ISCA Workshop on Disfluency in Spontaneous Speech 2005, 10-12 September 2005, Aix-en-Provence, France.

D. T. Toledano, J. Gonzalez-Dominguez, A. Abejón-Gonzalez, D. Spada, I. Mateos-García y J. Gonzalez-Rodriguez. Improved language recognition using better phonetic decoders and fusion with MFCC and SDC features. Proc. Interspeech. Agosto 2007

Trubetzkoy, Nicolai S.: *Principios de fonología*. Madrid, Cincel. Varias ediciones, 1939

E. Wong and S. Sridharan, *Methods to Improve Gaussian Mixture Model Based Language Identification System*, Proceedings of the International Conference on Spoken Language Processing (ICSLP), 2002.

S. Young et al., *The HTK Book* (for HTK version 3.2.1), available on http://htk.eng.cam.ac.uk/ .
.
Marc A. Zissman, *Comparison Of Four Approaches To Automatic Language Identification On Telephone Speech*. In IEEE Transactions on speech and audio processing, volume 4, pp. 31-44, Jan. 1996.