UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



**- PROYECTO FIN DE CARRERA -**

# DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO DE HABLA NATURAL INDEPENDIENTE DEL LOCUTOR

**Javier Antón Martín**

**Marzo 2015**

# DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO DE HABLA NATURAL INDEPENDIENTE DEL LOCUTOR

**AUTOR: Javier Antón Martín**
**TUTOR: Daniel Tapias Merino**
**PONENTE: Doroteo Torre Toledano**


**Área de Tratamiento de Voz y Señales**
**Dpto. de Tecnología Electrónica y de las Comunicaciones**
**Escuela Politécnica Superior**
**Universidad Autónoma de Madrid**
**Marzo 2015**

# Resumen

El objetivo de este proyecto es diseñar un sistema de reconocimiento de habla continua de gran vocabulario (LVCRS) utilizando modelos ocultos de Markov (HMM). Una vez conseguido, se procederá a mejorarlo aplicando diversas técnicas, como el incremento del número de gaussianas de los HMM y la optimización del tamaño de ventana de análisis para cada archivo de entrada.

También se intentarán deducir otra serie de parámetros como cuál es la ventana de análisis óptima para un reconocimiento genérico de voz y cuál es el número óptimo de archivos de entrenamiento necesarios. Asimismo, se realizarán pruebas de reconocimiento sobre contenidos audiovisuales obtenidos de Internet, para probar cómo reacciona el sistema en un entorno menos controlado.

Una vez creado, mejorado el sistema y medida su capacidad de reconocer vídeos de Internet, se procederá a integrar sus capacidades en un producto comercial dedicado a la búsqueda avanzada de contendios dentro de vídeos de Internet, utilizando el texto reconocido por el sistema para añadir posibilidades de búsqueda avanzada sobre los vídeos.

## Palabras Clave

Sistema de reconocimiento de habla continua de gran vocabulario, modelos ocultos de Markov, HTK, Julius, frecuencia fundamental, ventana de análisis, MFCC, búsqueda en contenidos audiovisuales.

# Abstract

The goal of this project was the implementation of a large vocabulary continuous speech recognition system (LVCRS) using hidden Markov models (HMM). Once implemented, the next step regarded the improvement of its different parameters, such as the number of gaussians of the HMM or the choice of the analysis window size for each input file.

The best analysis window for generic voice recognition and the optimal number of training files needed for this process were deducted as well.

Once finished, more tests were performed on audiovisual content obtained from Internet in order to verify how the system works in a less controlled environment.

Finally, its abilities were tested so that it can be integrated in a commercial product, used for speech recognition on Internet videos with the aim of using the recognized text to add new advanced search capabilities on these videos.

## Keywords

Large Vocabulary Continuous Speech Recognition, Hidden Markov Models, HTK, Julius, pitch, analysis window, MFCC, audiovisual content indexing.

# Agradecimientos

*En primer lugar, quería mostrar mi más profundo agradecimiento a mi tutor Daniel Tapias, que ha depositado su confianza en mí y me ha brindado la oportunidad de realizar este proyecto que me ha llenado de alegría e ilusión y que se ha convertido además en una transición de la universidad al mundo laboral. Me ha hecho sentirme verdaderamente útil y capaz de lograr metas nuevas en todos los ámbitos de mi vida.*

*También quería dar sinceramente las gracias a Jorge Rico, por enseñarme que con un ordenador todo es posible (incluso configurar una pantalla rebelde), por todos los conocimientos de idiomas, Linux y sabiduría que me ha transmitido, así como todos los momentos de risas que hay un día si y otro también.*

*No habría logrado llegar hasta aquí sin la inestimable ayuda de Juan Manuel Perero, a quien le doy las gracias. Ha estado siempre a mi lado (literalmente) durante este proyecto, tranquilizándome en los momentos de agobio y empujándome en los momentos de debilidad. Luchando codo con codo para hacer nuestros proyectos realidad, riendo cada minuto y diciendo tonterías en "italiano".*

*No me puedo olvidar del resto de compañeros de Sigma-Tax: Cris, Lidia, Martín, Nuria, Mayte, Mari Luz, Julieta y Elena. Siempre me han echado una mano cuando lo he necesitado y han amenizado los días entre cafés, charlas, roscones y risas, y espero que siga siendo así durante mucho tiempo. Muchas gracias a todos.*

*Por supuesto quiero dar las gracias a mis padres, que siempre han estado guiándome, apoyándome, animándome y haciéndome feliz, y más si cabe durante estos maravillosos años de carrera, dándome fuerzas y haciéndome ver que puedo conseguir todo lo que me proponga.*

*Quiero hacer una mención especial a mi siempre compañera Marta Anaya, por tantos años juntos sacando a flote nuestras prácticas mano a mano, aprendiendo uno del otro, dándome tu paz, riendo juntos en una perfecta (y en ocasiones aterradora) sincronía. Gracias.*

*También quería darle sinceramente las gracias a Bruno, mi pilar y punto de apoyo, que siempre ha estado conmigo durante todos estos años en los buenos momentos y en los malos, ayudándome siempre que le he necesitado, compartiendo juntos este periodo de nuestra vida, de café en café y de risa en risa.*

*Sin el resto de mis compañeros y además amigos de la carrera todo este tiempo no habría sido igual. Cada uno habéis dejado un trocito de vosotros en mi, además de los buenos momentos. Por ello, Alicia, Almu, Carol, Eva, Gus, Manu, Miguel, Pascu, Pirata, Rachel, Rosely, Tamara, los Álvaros y los Sergios, gracias.*

*También a los que no habéis estudiado directamente conmigo, pero que tantísimas cosas me habéis aportado durante estos años, Kike, Laura y Paula, muchas gracias por todo.*

*Y por ultimo, pero para nada menos importante, quiero dar las gracias de corazón a Susana Holgado, sin cuyo apoyo seguramente no habría llegado aquí. No sólo ha sabido aconsejarme y guiarme, sino que me ha dado su amistad y ha dado la cara por mi cuando la he necesitado. Gracias Susana.*

*Quiero expresar el más sincero agradecimiento al Banco Santader por la concesión de la "Beca de Prácticas Santander CRUE-CEPYME", que ha servido como complemento a mi formación y contribuido a mi acercamiento al ámbito profesional.*


*También quiero mostrar el agradecimiento a Telefónica I+D por la cesión de los derechos de las bases de datos que se han empleado en el desarrollo del sistema realizado como objetivo de este proyecto.*


*Además quiero dar las gracias al Biometric Recognition Group - ATVS por ofertar desde su grupo este proyecto final de carrera y haberme concedido la oportunidad de realizarlo.*


*Por último, y no menos importante, agradecer a Sigma Technologies la oportunidad brindada de realizar este proyecto, y por supuesto, de la acogida por parte de todos sus miembros y el conocimiento adquirido de cada uno de ellos.*

# Índice general

# Índice de figuras

# Índice de tablas

# Acrónimos

| | |
|---|---|
| **ARPA** | Advanced Research Projects Agency |
| **ASR** | Automatic Speech Recognition |
| **CMS** | Cepstral Mean Substraction |
| **CMU** | Carnegie Mellon University |
| **CVN** | Cepstral Variance Normalization |
| **DCT** | Discrete Cosine Transformate |
| **DFT** | Discrete Fourier Transformate |
| **DNN** | Deep Neural Networks |
| **DTW** | Dynamic Time Warping |
| **FFT** | Fast Fourier Transformate |
| **HMM** | Hidden Markov Model (Modelos Ocultos de Markov) |
| **HTK** | HMM ToolKit |
| **LPC** | Linear Predictive Coding |
| **LVCSR** | Large Vocabulary Continuous Speech Recognition |
| **MFCC** | Mel Frequency Cepstral Coefficient |
| **PLP** | Perceptual Linear Prediction |
| **RAH** | Reconocimiento de habla automático |
| **SNR** | Signal to Noise Ratio |

# Capítulo 1

# Introducción

## 1.1. Motivación

El reconocimiento automático de voz ha tenido una gran evolución en los últimos años gracias a su amplia utilidad, las facilidades que ha introducido en el desarrollo de múltiples tareas y el incremento de la potencia de los sistemas informáticos que lo realizan. Permite liberar las manos de cualquier teclado o dispositivo de entrada, lo que supone una gran ventaja a la hora de introducir estos sistemas en la vida cotidiana.

La posibilidad de obtener una representación simbólica discreta de una señal de voz continua, permite obtener en formato de texto la información vocal pronunciada por el hablante. Esta tarea es realizada en los llamados sistemas de reconocimiento automático de habla (*Automatic Speech Recognition - ASR).* Cada vez más, en este ámbito se tiende al aumento de la complejidad de los modelos, con el propósito de mejorar la precisión para distintas condiciones acústicas y vocabularios extensos.

Al haber tanta variabilidad entre las frecuencias fundamentales de la voz de las diferentes personas y entre sus tractos vocales, entrenar y adaptar de forma correcta un modelo acústico robusto se vuelve una tarea fundamental para afrontar esta labor con éxito.

Así, la motivación de este proyecto será estudiar cómo mejorar la calidad de los modelos acústicos y cómo aprovecharlo en el reconocimiento de contenidos audiovisuales de Internet.

## 1.2. Objetivos

El objetivo fundamental de este proyecto es el desarrollo de un sistema de reconocimiento de habla natural en español. Su evaluación se llevará a cabo de la siguiente forma:

- En Auto-test, mediante archivos de la base de datos de entrenamiento.
- Para test, a partir de una escisión de la primera con archivos que no se han entrenado.
- Otra pequeña base de datos ordenada por la frecuencia fundamental de la voz de sus hablantes.
- Vídeos diversos obtenidos de Internet.

Para alcanzar la meta establecida se fijarán una serie de objetivos parciales:

- En un inicio, se hará uso de la base de datos disponible. Se adaptará al software y la tarea con el fin de generar los modelos acústicos básicos que formen el sistema de referencia.
- Se usarán las herramientas de HTK [1] para el proceso de entrenamiento, y Julius [2] para la etapa de reconocimiento.
- Seguidamente, se estudiarán las mejoras obtenidas al adaptar el modelo acústico al dominio o tarea en cuestión.
- Finalmente, evaluando los resultados y como muestra de su utilidad, se incluirá en una aplicación que emplee este sistema de reconocimiento.

## 1.3. Estructura de la memoria

Esta memoria de proyecto está dividida en los siguientes capítulos:

- Capítulo 1. Introducción: motivación y objetivos del proyecto.
- Capítulo 2. Estado del Arte: sistemas de reconocimiento de voz, arquitectura y reconocimiento con HMMs.
- Capítulo 3. Base de Datos: descripción y caracterización de la base de datos.
- Capítulo 4. Generación del sistema de referencia: metodología de entrenamiento, reconocimiento y resultados.

- Capítulo 5. Optimización del sistema y resultados: mejora del sistema con modelos acústicos adaptados a la frecuencia fundamental y resultados.
- Capítulo 6. Aplicación del sistema: demostración de la aplicación del sistema implementado a una solución comercial.
- Capítulo 7. Conclusiones y trabajo futuro.
- Referencias y anexos.

# Capítulo 2

# Estado del Arte

En este capítulo se proporcionará una visión general de la evolución de los trabajos realizados en el área del reconocimiento de voz hasta nuestros días (sección 2.1). En las siguientes secciones se explicarán la arquitectura de este tipo de sistemas de reconocimiento y sus bloques (sección 2.2), así como una visión detallada de la generación de modelos estadísticos (sección 2.3).

## 2.1. Introducción general del reconocimiento de voz

El reconocimiento de habla natural ha experimentado un intenso desarrollo gracias a los avances que han tenido lugar en el procesamiento de señal, algoritmos, arquitecturas y plataformas de computación.

Desde 1940, los laboratorios de AT&T y Bell se encargaron de desarrollar un dispositivo rudimentario para reconocer voz, fundamentándose en los principios de la fonética acústica, teniendo presente que el éxito de esta tecnología, dependería de su habilidad para percibir la información verbal compleja con alta precisión.

En la década de los 50, el sistema anterior conseguido, permitía identificación de dígitos mono-locutor, basada en medidas de resonancias espectrales del tracto vocal para cada dígito. Siguiendo esta línea, RCA Labs trabajó en el reconocimiento de 10 sílabas. Y es a finales de la década, cuando tanto la University College de Londres como el MIT Lincoln Lab, trataron de desarrollar un sistema de reconocimiento limitado de vocales y consonantes. Esta tarea parecía novedosa por el uso de información estadística y cuyo objetivo era una mejora del rendimiento en palabras de dos o más fonemas.

Fue por la década de los 60, cuando los sistemas electrónicos utilizados hasta el momento, sirvieron de pasarela a los sistemas con hardware específico, en los NEC

Labs de Japón. En esta etapa, cabe destacar tres proyectos notables en la investigación de esta disciplina:

- RCA Labs tenían como objetivo un desarrollo de soluciones realistas para los problemas en la falta de uniformidad de las escalas de tiempo en el habla. Para ello, diseñaron un conjunto de métodos de normalización en el dominio temporal, detectando fiablemente el inicio y fin de discurso.

- En la Unión Soviética, T. K. Vintsyuk, propone el empleo de métodos de programación dinámica para conseguir el alineamiento temporal de parejas de realizaciones. Surge de aquí la técnica *DTW (Dynamic Time Warping)*.

- Por último, en el campo del reconocimiento de habla continua, D. R. Reddy de la Universidad de Stanford, desarrolla el seguimiento dinámico de fonemas, concluyendo su trabajo en un reconocedor de oraciones de amplio vocabulario.

Allá por los años 70, se originan críticas acerca de la viabilidad y utilidad del reconocimiento automático de habla. A pesar de esto, dicha disciplina se adentra en el mundo probabilístico, donde los principales campos de estudio son los siguientes: el reconocimiento de palabras aisladas estuvo fundamentado en el procedimiento de ajuste de patrones, programación dinámica, y más adelante, técnicas *LPC (Linear Predictive Coding)*. Ésta última se empleó exitosamente en la codificación y compresión de la voz, a través del uso de medidas de distancias sobre el conjunto de parámetros LPC. Los primeros intentos de reconocedores de habla contínua y grandes vocabularios los llevaron a cabo IBM, con el dictado automático de voz, ARPA Speech Understanding Research, y la Universidad de Carnegie Mellon, con el exitoso sistema Hearsay I. Finalmente, en los AT&T Labs, se investigó en la dirección de los reconocedores independientes del locutor para aplicaciones telefónicas, finalizando este periodo con la realización de sistemas *ASR (Automatic Speech Recognition)*, favorecida por tarjetas microprocesador.

La década de los 80 se inicia con una base muy asentada en la construcción de sistemas de reconocimiento, a diferencia de los anteriores que sólo reconocía vocablos aislados, ahora tienen la capacidad de tratar con palabras encadenadas fluidamente. Uno de los avances más importante es el paso de métodos basados en comparación de plantillas a otros basados en modelos estadísticos, extendiéndose el uso de los Modelos Ocultos de Markov o HMMs. Éstos experimentaron numerosas mejoras y se situaron como los mejores modelos que capturaban y modelaban la variabilidad del habla.

Las redes neuronales empezaron a tomar peso en este ámbito, y gracias al desarrollo de algoritmos de aprendizaje más eficaces, aparecieron modelos como el *perceptrón*.

Además se llevan a cabo una serie de avances:

- El diseño de unidades de decodificación fonética a partir de la experiencia de fonetistas en tareas de interpretación de espectrogramas.
- La grabación de grandes bases de datos como TIMIT, que permite la comparación de resultados entre diferentes grupos de trabajo.
- El programa DARPA (Defence Advance Research Agency) contribuyó en Estados Unidos, al impulso del desarrollo de sistemas de reconocimiento para habla contínua y vocabularios de gran tamaño con independencia del locutor.
- El desarrollo por parte de la CMU de su sistema SPHINX [3].

En los años 90, continuando con los objetivos ya propuestos anteriormente, se amplían los tamaños de vocabularios y se diversifican los campos de aplicación. Teniendo gran importancia su aplicación sobre línea telefónica, así como los resultados de este reconocimiento en entornos con condiciones adversas y ruido.

Los avances producidos en el ámbito de las tecnologías del habla cada día son más significativos. En el campo del reconocimiento automático de voz, los reconocedores actuales manejan cada vez vocabularios más grandes y reducen las tasas de error, gracias al uso de algoritmos más eficientes, al uso de equipos más potentes y al aumento de complejidad de estos sistemas, con modelados más sofisticados. El amplio grado de aplicación en función de los usuarios y los distintos entornos, hacen que no haya un sistema de reconocimiento de voz universal y sea necesaria su adaptación a las condiciones de funcionamiento y al tipo de aplicación que se requiera.

## 2.2. Arquitectura de un reconocedor automático de habla

Los sistemas de reconocimiento automático de habla (RAH) han sido abordados desde diferentes enfoques, como se ha comentado anteriormente (sección 2.1), siendo los probabilísticos, que emplean la "Teoría de la Decisión de Bayes", la "Teoría de la Información" y las "Técnicas de Comparación de Patrones", los que han aportado los mejores resultados.

Un sistema de estas características (Figura 2.1), tiene la finalidad de extraer de la información acústica contenida en la señal de voz, una representación de todo el conjunto de sonidos pronunciados en formato texto. Para llevar a cabo esta decodificación, existen diferentes técnicas partiendo de un conjunto de patrones que sean comparables con el mensaje de entrada, y devolviendo al final una secuencia de aquellos patrones que con mayor probabilidad representan dicho mensaje.

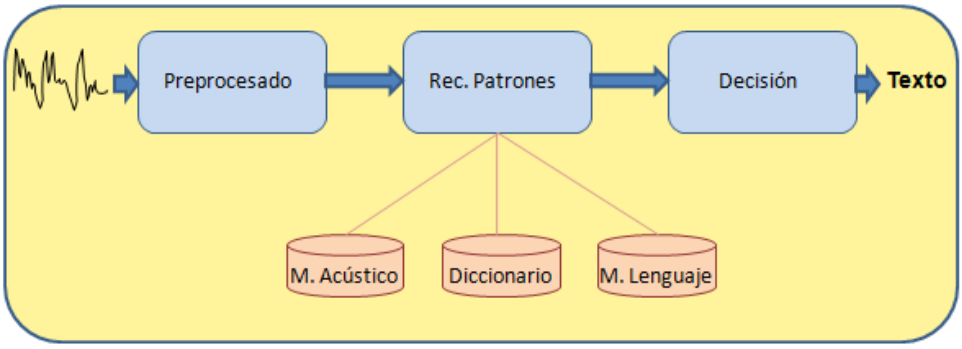


Figura 2.1: Esquema básico de la arquitectura de un RAH.

Una vez dada una visión general del propósito y funcionalidad de este tipo de sistemas de reconocimiento, se centrará la atención de este apartado en los sistemas de reconocimiento de habla continua y de vocabulario extenso *(Large Vocabulary Continuous Speech Recognition - LVCSR),* cuya arquitectura está compuesta por los siguientes bloques:

### 2.2.1. Preprocesado

Este bloque engloba la extracción de características y transformación, procesado de características de robustez al ruido y la estimación de características adaptativas y discriminativas [4].

**Extracción de características:** El papel que desempeña el módulo de preprocesado es extraer a partir de la señal de voz una secuencia de vectores de características acústicas. Ésto es realizado gracias a la transformada rápida de Fourier *(Fast Fourier Transform - FFT)* de la señal de voz dentro de una ventana de análisis, la cual se desplaza un intervalo de tiempo fijo. Las energías de las frecuencias vecinas dentro de cada trama son desechadas mediante un banco de filtros en la escala Mel, siendo las características de éstos inspiradas en el proceso auditivo humano. A la salida de los filtros se aplica un logaritmo y los coeficientes son decorrelados a partir de la transformada discreta del coseno, dando lugar a un vector de coeficientes cepstrales de frecuencia Mel *(Mel Frequency Cepstral Coefficiente - MFCC).*

Posteriormente, estos coeficientes han sido reemplazados teniendo presente una mayor robustez al ruido, basado en coeficientes perceptibles de predicción lineal *(Perceptual Linear Prediction - PLP).*

En este contexto, la extracción de características ha beneficiado dos importantes técnicas: la primera de ellas, el uso de la media basada en el locutor, y la normalización de la varianza de los coeficientes cepstrales. Mientras que la sustracción de la

media cepstral basada en la pronunciación *(Cepstral Mean Substraction - CMS)* es una técnica muy conocida, la normalización de la varianza cepstral *(Cepstral Variance Normalization - CVN)* se ha introducido recientemente. La segunda es la incorporación de contexto temporal entre las tramas, calculando los coeficientes dinámicos o de velocidad y aceleración, también llamados coeficientes delta o delta-delta respectivamente). Éstos son calculados a partir de las tramas próximas dentro de una ventana de aproximadamente de unas 4 tramas de media. Estos coeficientes dinámicos se añaden a los estáticos formando así un vector final.

**Características robustas al ruido:** El ruido ambiente suele contaminar la señal de voz que obviamente afectará posteriormente al reconocimiento, de ahí que se trate este efecto en la etapa de preprocesado. El algoritmo SPLICE *("Stereo-based piecewise linear compensation for enviroments")* fue propuesto para entornos de ruido no estacionario, consistente en la eliminación del ruido por medio de la diferencia entre voz limpia y voz corrupta, asociada a la región más probable del espacio acústico. Otro algoritmo *QE (Quantile-based histogram equalization)* fue desarrollado para compensar distribuciones desalineadas de los datos de entrenamiento y de test.

Ambos fueron evaluados empleando un corpus de The Wall Street Journal, modificando el tipo y los niveles de ruido, pudiendo comprobarse mejoras en entornos experimentales limpios y multicondición.

**Estimación de características adaptativas y discriminativas:** La variación de las características acústicas puede ser observada entre los diferentes locutores o en un mismo locutor. Por ello, existen técnicas para generar un espacio de características canónicas, eliminando esta variabilidad mencionada en la medida de lo posible. Algunos ejemplos de ellas: normalización de la longitud del tracto vocal *(Vocal Tract Length Normalization - VTLN),* tranformación de las características maximizando la verosimilitud bajo un modelo actual *(feature-space Maximun Likelihood Linear Regression - fMLLR),* transformación no lineal de la distribución de los datos de adaptación que será alineada con una distribución normal de referencia.

En la estimación de características discriminativas se usan técnicas como la transformación que permite obtener *offsets* dependientes del tiempo, a partir de una proyección lineal de un espacio de gaussianas posteriores de gran dimensión *(feature-space minimum phone error - fMPE).*

#### 2.2.1.1. Filtro de pre-énfasis

Dado que la señal de voz se ve atenuada según se incrementa la frecuencia, se vuelve necesario aumentar la importancia de las frecuencias más elevadas. Por ello, se analiza la señal de voz mediante un filtro de pre-énfasis. Al ser una señal digital

se suele aplicar un filtro FIR con función de transferencia $H(z) = 1 - a \cdot z^{-1}$ donde $a$ toma el valor de $0,97$.

El cero de transmisión de este filtro varía según el valor de $a$, resultando en un filtro plano cuando $a = 0$ o un cero de transmisión en la frecuencia 0 cuando $a = 1$.

#### 2.2.1.2. Enventanado

El siguiente paso es elegir adecuadamente el enventanado y el solapamiento. Dicho enventanado puede ser realizado con una ventana de tipo Hamming. Se suele elegir por sus cualidades espectrales.

Debido a la forma irregular de la ventana de Hamming, las muestras en los extremos sufrirán una ponderación. Para compensar este efecto lo que se hace es solapar unas ventanas con otras, de forma que se anule. Así, la ventana se toma de 25 ms y el desplazamiento de 10 ms, que corresponde a un 60 % de solape entre ventanas.

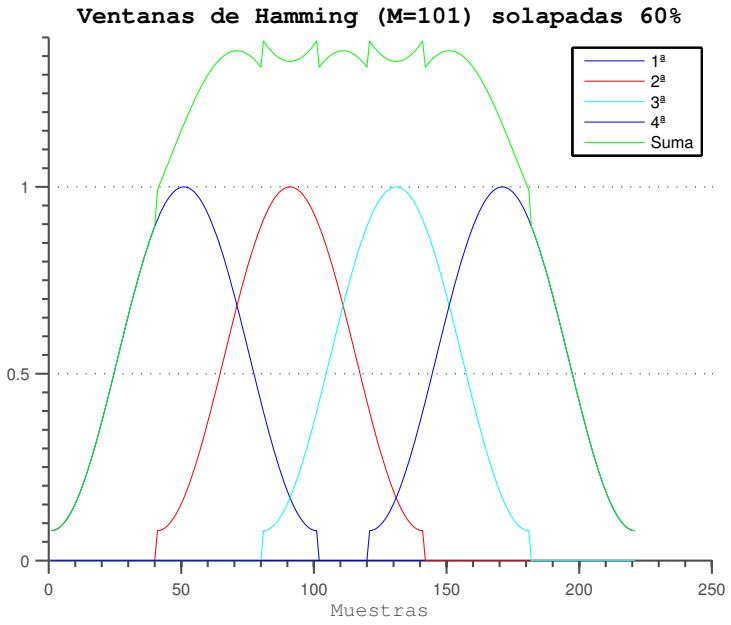


Figura 2.2: Ejemplo de ventanas de Hamming solapadas un 60 %.

#### 2.2.1.3. Transformada discreta de Fourier

Al estar trabajando en un sistema lineal e invariante en el tiempo, se pueden representar algunas de sus propiedades en el dominio de la frecuencia. Una de las propiedades a destacar es que la respuesta a señales sinusoides es otra señal sinusoidal

o una combinación de exponenciales complejas, lo que las hacen muy útiles cuando se trata con señales de voz. Están basadas en secuencias base de exponenciales complejas.

En señales discretas de duración finita se usa la *Transformada Discreta de Fourier* (DFT).

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} \tag{2.1}$$

Pero en la práctica se utiliza el algoritmo de la FFT, que permite realizar la transformada discreta de Fourier y su inversa con un coste computacional mucho menor.

La salida de la transformada de Fourier se filtra con un banco de filtros en la escala Mel (Figura 2.3) , siendo las características de éstos inspiradas en el proceso auditivo humano.

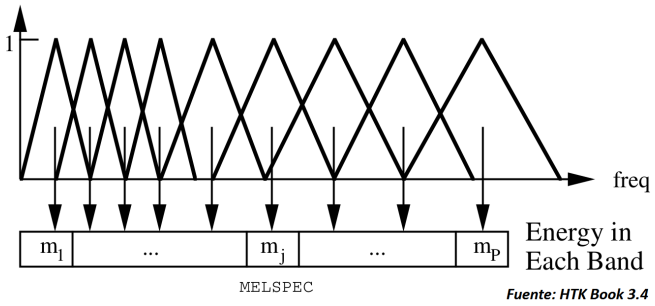


Figura 2.3: Mel-Scale Filter Bank.

#### 2.2.1.4. Transformada discreta del coseno

A la salida de los filtros se aplica un logaritmo y los coeficientes son decorrelados a partir de la transformada discreta del coseno. La transformada discreta del coseno (DCT) y la transformada discreta de Fourier son muy similares entre sí, pero su mayor diferencia radica en que mientras las secuencias base de la DFT son exponenciales complejas, en la DCT son funciones cosenoidales.

$$X(k) = \sum_{n=0}^{N-1} x(n) \cos\left[\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right] \tag{2.2}$$

Dado que la mayor parte de la energía resultante de este tipo de transformada se aglutina en los coeficientes de baja frecuencia, este tipo de transformada es muy útil en aplicaciones de compresión de datos.

#### 2.2.1.5. El dominio cepstral

Todo esto da lugar a un vector de, usualmente, 13 coeficientes cepstrales de frecuencia Mel *(Mel Frequency Cepstral Coefficiente - MFCC)*. Se basan en aplicar un filtro de decorrelación homomórfica (*Cepstrum*) mediante la transformada inversa de Fourier del logaritmo del espectro de potencias, llevando los coeficientes cepstrales al dominio de la *cuefrencia* y obteniendo así coeficientes cepstrales.

Gracias a este filtrado se pueden decorrelar los espectros de los filtros en bandas adyacentes, saltando la limitación que tienen las técnicas de análisis espectral que operan en el dominio de la potencia espectral logarítmica.

Así, los coeficientes cepstrales representan la señal en el tiempo, que coincide con el espectro logarítmico de la potencia.

Como el dominio de la cuefrencia es un dominio homomórfico del dominio temporal, las convoluciones en el dominio temporal se convierten en sumas en el dominio de la cuefrencia, lo cual permite separar las señales de voz de los ruidos convolucionales con los que están mezcladas.

De esta forma, las partes de excitación y envolvente espectral de la voz aparecerán en zonas distintas en el dominio de la cuefrencia, permitiendo separarlo mediante ventanas, lo que se llama *liftering*.

En el dominio de la frecuencia, los análisis en el dominio espectral se llaman LPC (Linear Predictive Coding), MFCC (Mel Frecuency Cepstral Coefficients) y Cepstrum PLP entre otros.

### 2.2.2. Reconocimiento de patrones

El principal motivo de emplear esta técnica es la consistencia de las representaciones de los patrones al definirse claramente un modelo matemático. Éstas pueden servir de referencia a la hora de realizar comparaciones con alto grado de confianza; para ello, serán precisos un conjunto de muestras etiquetadas y una metodología de entrenamiento [5].

#### 2.2.2.1. Etapas

La representación de los patrones puede ser una plantilla (*template*) o un modelo estadístico (HMM), y se aplicará a un sonido, una palabra o una frase. Esta técnica puede dividirse en dos etapas: entrenamiento y comparación.

- **Entrenamiento:** Esta etapa consiste en la construcción de un patrón de referencia asociado a cada palabra o sub-unidad de palabra que se quiere reconocer, basándose en los vectores de características de aquellas unidades empleadas en el proceso de entrenamiento. Existen varias formas de llevar a cabo este proceso:
  - ✧ Entrenamiento casual: se asigna un único patrón de sonido en la generación del patrón de referencia o un modelo estadístico aproximado.
  - ✧ Entrenamiento robusto: se emplean varias versiones de cada unidad a reconocer (provenientes de un mismo locutor) generando así un patrón de referencia promedio o modelo estadístico promedio.
  - ✧ Entrenamiento por *clustering*: se emplean gran volumen de datos, disponiendo de varias versiones de cada unidad (procedentes de un gran número de locutores) y así construir patrones de referencia o modelos estadísticos con alto grado de confianza.
- **Comparación:** Esta etapa está fundamentada, como su propio nombre indica, en la comparación directa entre el vector característico asociado a la señal de voz (a reconocer) y todos los posibles patrones entrenados, con el fin de determinar el mejor ajuste de acuerdo a un criterio establecido. Se definirá una medida de similaridad (distancia) entre vectores característicos a partir de la cual obtener el patrón de referencia mejor ajustado a la señal a reconocer.

#### 2.2.2.2. Modelo acústico

Uno de los elementos fundamentales en la técnica de reconocimiento de patrones es el modelo acústico, que es un conjunto de representaciones estadísticas de los

diferentes sonidos del espacio acústico con el que se está trabajando. Su elaboración se lleva a cabo a partir de un volumen de datos de entrenamiento, consistentes en datos de voz con su correspondiente etiquetado (transcripciones), haciendo posible una asignación de cada sonido a su representación o carácter gráfico.

A continuación, se mencionarán brevemente las técnicas más importantes empleadas para generar estos modelos:

**Hidden Markov Model (HMM)**

Los HMMs son modelos estadísticos empleados en la representación de secuencias de datos espaciales o temporales, este último es el caso de la señal de voz. Estos modelos son la base tecnológica de los sistemas de reconocimiento de voz, sustituyendo desde los años 80 a las técnicas de comparación de patrones como los DTW, que modelaban la voz de forma determinista.

Se considera que el sistema a modelar es un proceso de Markov de parámetros desconocidos, los cuáles serán calculados a partir de los parámetros observables. Este procedimiento ha sido el empleado en este proyecto, por ello, se dedicará una explicación más amplia en la sección 2.3.

**Dynamic Temporal Warping (DTW)**

Consiste en el alineamiento temporal de los parámetros de la locución de test y los parámetros del patrón, como resultado se obtiene la función de menor coste, que alinea ambas locuciones. El amplio abanico existente entre todos los posibles caminos de alineamiento, se verá reducido por un conjunto de límites locales y una serie de limitaciones.

**Vectorial Quantization (VQ)**

Consiste en la representación de las características de las unidades como un espacio vectorial, el cual cuenta con un conjunto infinito de patrones posibles (espacio de características). En este espacio se pretende asignar un conjunto de patrones desconocidos (test) a un conjunto finito de patrones de referencia; de manera que al vector a reconocer se le asigna un vector patrón cuya distancia a él sea mínima.

El espacio representativo quedará dividido en zonas o regiones, donde al vector representativo de esa región se denominará “codeword” (centroide), de forma que los vectores que caigan en dicha región se asignarán a dicho centroide. El conjunto de todos los centroides se denomina “codebook” (muestrario).

Como se puede observar en la figura 2.5, correspondiente a un espacio bidimensional, se asignan aleatoriamente los centroides, representados con un “o”; a su vez,

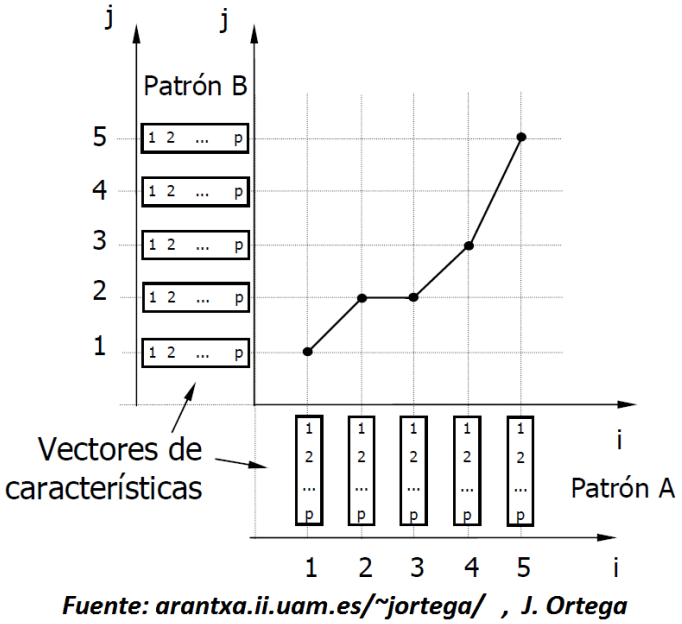


Figura 2.4: Función de alineamiento de DTW.

los vectores de test son representados con una "x" y serán asignados al centroide más cercano, mientras que cada una de las regiones podrían corresponderse con cada uno de los fonemas.

**Deep Neural Networks (DNN)**

Las redes neuronales profundas [6] son una forma alternativa de aprendizaje y de procesamiento automático, basado en el funcionamiento del sistema nervioso. Emplea una red neuronal basada en feed-forward que toma varias tramas de coeficientes como entrada y produce probabilidades a posteriori sobre los estados de HMM como salida. Se caracterizan por poseer un gran número de capas ocultas y son entrenadas usando nuevos métodos que mejoran otros procedimientos aquí ya mencionados.

La conectividad completa entre capas adyacentes, se trata con la asignación de pesos iniciales de baja magnitud y aleatorios, y así se evita que todas las unidades ocultas en una capa tengan exactamente los mismos valores en cuanto a los pesos. En DNNs con gran número de capas y de elementos en cada capa, son modelos más flexibles que son capaces de modelar relaciones altamente no lineales entre entradas y salidas.

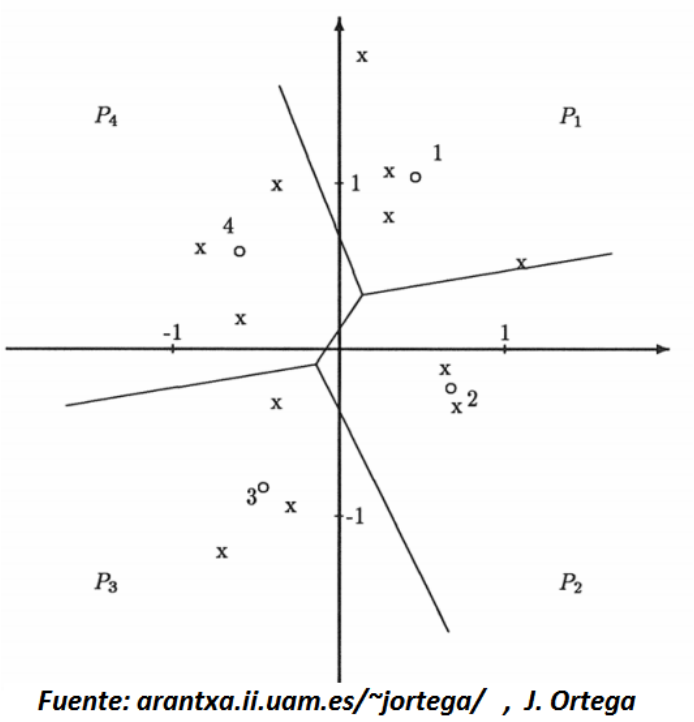


Figura 2.5: Espacio bidimensional dividido en regiones (VQ).

#### 2.2.2.3. Diccionario

El diccionario llamado también *lexicon*, juega el papel de nexo entre la representación del nivel acústico y la secuencia de palabras a la salida del reconocedor. Consiste en un bloque que especifica tanto las palabras conocidas por el sistema, como los significados que construyen los modelos acústicos para cada entrada. Para LVCSR, normalmente el vocabulario es elegido con el objetivo de maximizar la cobertura para un tamaño de diccionario dado, pudiendo contener palabras iguales con más de una pronunciación. Además, la generación del diccionario se puede ver influida por aspectos como el tipo de habla, leída o espontánea [7], siendo recomendable tratar esta variabilidad de las pronunciaciones, y así obtener el mayor rendimiento posible al sistema.

#### 2.2.2.4. Modelo de lenguaje

El modelo de lenguaje permite definir una estructura del lenguaje, es decir, restringir correctamente las secuencias de las unidades lingüísticas más probables. Son empleados en sistemas que hagan uso de una sintaxis y semántica compleja. Su funcionalidad debería consistir en aceptar (con alta probabilidad) frases correctas y rechazar

(o asignar baja probabilidad) secuencias de palabras incorrectas.

Se pueden tener dos tipos de modelos: gramática cerrada de estados finitos y modelo de N-gramas.

**Gramática cerrada de estados finitos**

Este tipo de modelo representa restricciones del lenguaje, permitiendo modelar dependencias tan largas como se quiera. Su aplicación conlleva una gran dificultad para tareas que hagan uso de lenguajes próximos a lenguajes naturales.

**Modelo de N-gramas**

Los modelos estadísticos de lenguaje de N-gramas [8], pretenden predecir la palabra siguiente de manera que se reduzca el espacio de búsqueda sólo a los candidatos más probables. El empleo de la información contextual, permite mejorar aplicaciones ahorrando medios. En idiomas con palabras que tienen la misma pronunciación u homónimas, se precisa claramente de la información contextual, y así mejorar la precisión del sistema.

### 2.2.3. Decisión

Esta última etapa consiste en la toma de decisión a la hora de asignar un patrón de los que se han generado en la fase de entrenamiento del sistema. Para ello, se hará uso de las medidas realizadas en la fase de comparación; es decir, los cálculos de parecido o similitud entre la realización acústica de entrada y el conjunto de modelos conocidos por el sistema. A partir de los valores de similitud obtenidos, el reconocedor debe tomar una decisión acerca de los sonidos que ha generado la señal de voz de entrada.

El teorema de decisión de Bayes expresa la probabilidad condicional de que los sonidos de entrada o combinaciones de ellos (trifonemas) pudiesen ser generados por alguno de los estados que modelan todas las posibles unidades del espacio acústico. Por otro lado, las distribuciones probabilísticas para el modelado del lenguaje, reflejan la frecuencia de aparición de las cadenas de palabras. La decisión estará basada en la mayor verosimilitud obtenida tanto del modelo acústico como del modelo del lenguaje.

Este bloque es uno de los que más relevancia tiene para el diseñador de la arquitectura del sistema de reconocimiento, ya que es la única salida observable por el usuario.

## 2.3. Reconocimiento con HMMs.

Como se explicó brevemente en la sección 2.2, los HMMs se han convertido en la aproximación predominante en el reconocimiento de habla por su algorítmica y los resultados que se obtienen con ellos.

### 2.3.1. Definición y caracterización

Un modelo oculto de Markov es la representación de un proceso estocástico que se compone de dos elementos: una cadena de Markov de primer orden, con un número finito de estados, y un conjunto de funciones aleatorias asociadas a cada uno de los estados. En un instante concreto de tiempo. el proceso está en un estado determinado y genera una observación mediante la función aleatoria asociada. En el instante siguiente, la cadena de Markov cambia de estado o permanece en el mismo siguiendo su matriz de probabilidades de transición entre estados, generando una nueva observación mediante la función aleatoria correspondiente. El observador externo solamente verá la salida de las funciones asociadas a cada estado, sin observar directamente la secuencia de estados de la cadena de Markov.

Tomando por mayor simplicidad la notación de modelos discretos, un HMM está caracterizado por:

- El número de estados en el modelo, $N$. Se denota cada estado como $S_i$, un estado en el instante $t$ como $q_t$; por lo tanto si el sistema se encuentra en el estado $S_i$ en el instante $t$, $q_t = S_i$.

- El número de símbolos observables, $M$. Se denota a cada símbolo observable como $v_j$, la observación en el instante $t$ como $O_t$, y si la observación en el instante $t$ es $v_j$, se tomará $O_t = v_j$.

- La matriz de probabilidades de transición se define como $A = \{a_{i,j}\}$, siendo $a_{i,j} = P\left[q_{t+1} = S_j \,|\, q_t = S_i\right]$ y cumpliéndose que $1 \leq i,j \leq N$

- La distribución de probabilidad de observación en cada estado $j$ como $B = \{b_j(k)\}$, siendo $b_j(k) = P\left[v_k(t) \,|\, q_t = S_t\right]$ y para $1 \leq j \leq N$ y $1 \leq k \leq M$.

- La probabilidad inicial de ocupación de cada estado como $\pi = \{\pi_i\}$, donde $\pi_i = P\left[q_1 = S_i\right]$ y para $1 \leq i \leq N$.

Por convenio, un modelo HMM con todos los parámetros será denotado de la siguiente forma: $\lambda = (A, B, \pi)$.

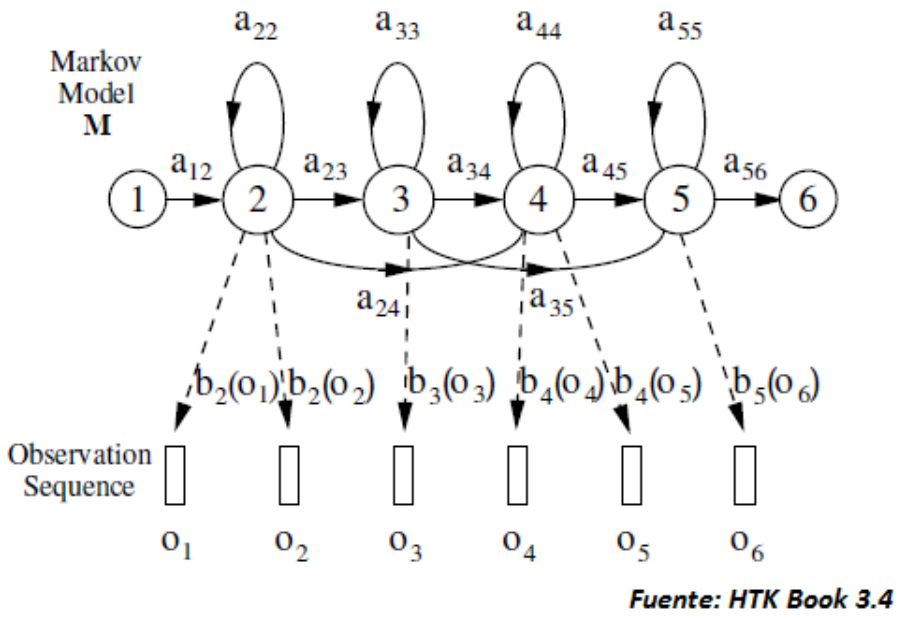


Figura 2.6: Representación gráfica de un HMM de 6 estados.

Dependiendo de la naturaleza de la matriz de distribución de probabilidades de salida $B$, los HMMs se pueden clasificar en varios tipos: modelos discretos, modelos continuos y modelos semicontinuos.

- **Modelos discretos:** En este tipo de modelos, las observaciones son vectores compuestos por símbolos de un alfabeto finito de N elementos distintos. Para cada elemento del vector de símbolos se define una densidad discreta y la probabilidad del vector se calcula multiplicando las probabilidades de cada componente siendo éstos independientes entre sí.

- **Modelos continuos:** Definen las distribuciones de probabilidad en espacios de observaciones continuos, muy conveniente en el ámbito de este proyecto ya que se trata de la señal de voz que es propiamente continua. Se suelen restringir el número de parámetros del sistema para conseguir una mayor manejabilidad de éste y consistencia de las re-estimaciones; para ello, se emplean mezclas de distribuciones paramétricas como gaussianas, para definir las transiciones. Cada estado $x_i$ tendrá un conjunto específico $V\,(x_i,\,\lambda)$ de funciones densidad de probabilidad. Llamando $v_k$ a cada de las funciones de densidad de probabilidad, las probabilidades de las salidas se pueden expresar como:

$$b_i(y) = p\,(y\,|\,x_i,\,\lambda) = \sum_{v_k \epsilon V(x_i,\lambda)} p\,(y\,|\,v_k,\,x_i,\,\lambda)\,P\,(v_k\,|\,x_i,\,\lambda) \qquad (2.3)$$

- **Modelos semicontinuos:** En ellos se modelan distribuciones complejas con un elevado número de mezclas de funciones paramétricas y un gran corpus de entrenamiento. Se compartirán las mismas distribuciones de probabilidad con pesos distintos, entre todos los estados del modelo.

Como se acaba de comentar, la dependencia temporal de la señal de voz permite que los HMMs se adapten muy bien en sistemas de reconocimiento, además del cálculo de probabilidades acústicas gracias a la capacidad de esta técnica a la hora de modelar estadísticamente la generación de voz. Para su uso se tendrán en cuenta dos hipótesis:

1. El análisis localizado de la señal de voz, permitirá la división de ésta en fragmentos, estados, en los que se puede considerar su pseudoestacionariedad [9]. Esto es gracias a que, en la ventana de análisis, la señal mantiene su periodicidad, teniendo presente las transiciones.

2. La hipótesis de independencia de Markov, que enuncia que la probabilidad de observación de que se genere un vector de características, depende únicamente del estado actual y no de elementos anteriores [10].

### 2.3.2. Los tres problemas básicos de los HMMs

Existen tres problemas fundamentales de los Modelos Ocultos de Markov, cuya solución hace de ellos, una técnica de la robustez y utilidad ya mencionada en aplicaciones reales:

- **Evaluación o de puntuación:** Dada una secuencia de observaciones acústicas y un modelo oculto de Markov, ¿cómo calcular la probabilidad de que dicho modelo genere la secuencia de observación vista? Esta probabilidad, $P(O \mid \lambda)$, se determina a partir del algoritmo de forward-backward [11].

- **Decodificación o reconocimiento de estados:** Dada una secuencia de observaciones acústicas y un modelo oculto de Markov, ¿cuál es la secuencia de estados óptima que explique dichas observaciones? La secuencia de estados óptima se consigue gracias a un alineamiento de la secuencia de observación con los estados, mediante el algoritmo de Viterbi [12].

- **Entrenamiento:** Dado un conjunto de observaciones de entrenamiento y un modelo oculto de Markov, ¿cómo se ajustan los parámetros del modelo para maximizar la probabilidad de observar el conjunto de entrenamiento? Este ajuste paramétrico será solucionado con el algoritmo de Baum-Welch [13].

#### 2.3.2.1. Problema de evaluación. Algoritmo Forward-Backward

Partiendo de un modelo HMM definido por $\lambda = (A, B, \pi)$ y la secuencia de observaciones $O = O_1\, O_2 \cdots O_T$, suponiendo que la secuencia de estados es $Q = q_1\, q_2 \cdots q_T$, la probabilidad de la secuencia de observaciones dada una secuencia de estados, viene dada por:

$$P\,(O \,|\, Q,\, \lambda) = \prod_{t=1}^{T} P\,(O_t \,|\, q_t,\, \lambda) \tag{2.4}$$

Con la asunción de la independencia estadística de las observaciones, tenemos:

$$P\,(O \,|\, Q,\, \lambda) = b_{q_1}(O_1)\, b_{q_2}(O_2) \cdots b_{q_T}(O_T) \tag{2.5}$$

Por otro lado, la probabilidad conjunta de $O$ y de $Q$ será:

$$P\,(O,\, Q \,|\, \lambda) = P\,(O \,|\, Q,\, \lambda)\, P\,(Q,\, \lambda) \tag{2.6}$$

Sabiendo que la probabilidad de la secuencia de estados $Q$, puede expresarse como el producto de la probabilidad del estado inicial y de las probabilidades de transición entre estados :

$$P\,(Q \,|\, \lambda) = \pi_{q_1} a_{q_1 q_2} \pi_{q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T} \tag{2.7}$$

La probabilidad buscada se calculará sumando las probabilidades anteriormente definidas para todos los caminos posibles o secuencias de estados:

$$P\,(O \,|\, \lambda) = \sum_{\forall Q} P\,(O \,|\, Q,\, \lambda)\, P\,(Q \,|\, \lambda) = \sum_{q_1, q_2 \cdots q_T} \pi_{q_1} b_{q_1}(O_1)\, a_{q_1 q_2} b_{q_2}(O_2) \cdots a_{q_{T-1} q_T} b_{q_T}(O_T) \tag{2.8}$$

De la expresión anterior se deduce que en el instante inicial $t = 1$, se tiene presencia en el estado $q_1$ con una probabilidad inicial $\pi_{q_1}$ y se genera la observación $O_1$ con probabilidad $b_{q_1}(O_1)$. En un instante de tiempo siguiente $t = 2$, la transición del estado $q_1$ a $q_2$ se producirá con una probabilidad $a_{q_1 q_2}$ y se generará la observación $O_2$ con probabilidad $b_{q_2}(O_2)$. Este proceso se realizará de la misma forma hasta la última transición, es decir, hasta el estado final $q_T$.

El problema de este cálculo directo es que requiere un número muy elevado de operaciones, del $O(2\,T N^T)$, siendo inviable. Esto se soluciona con el algoritmo de forward-backward que realiza cálculos intermedios que emplea a posteriori y que supone una reducción del coste computacional del $O(T N^2)$.

Se llevará a cabo el siguiente procedimiento:

- **Inicialización** de la variable forward, que representa la probabilidad de observar la secuencia parcial hasta el instante $t$ y estar en el estado $S_i$ en dicho instante (ecuación 2.9); y que en el instante $t = 1$, será como muestra la ecuación 2.10.

$$\alpha_t(i) = P\left(O_1 O_2 \cdots O_t,\, q_t = S_i \,|\, \lambda\right) \tag{2.9}$$

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N \tag{2.10}$$

- A través del **método inductivo**, se calculan las variables forward en el instante $t + 1$ a partir de las variables forward en el instante $t$, de las probabilidades de transición y probabilidades de observación. Se realizarán un total de $N + 1$ productos y $N - 1$ sumas.

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(O_{t+1}) \tag{2.11}$$

- El último paso es de **finalización**. La probabilidad deseada se calcula como suma de las probabilidades hacia delante en el último instante posible, $T$:

$$P\left(O \,|\, \lambda\right) = \sum_{i=1}^{N} \alpha_T(i) \tag{2.12}$$

Otro algoritmo es el de backward, que tiene el mismo fundamento pero la probabilidad de observación de una secuencia $i$ y se modela como:

$$\beta_t(i) = P\left(O_{t+1} O_{t+2} \cdots O_T \,|\, q_t = S_i,\, \lambda\right) \tag{2.13}$$

donde $\beta_t(i)$ representa la probabilidad de observar la secuencia parcial $O_{t+1} \cdots O_T$ desde el instante $t + 1$ y estar en el estado $S_i$ en el instante $t$. Se puede calcular la probabilidad empleando tanto el método de forward como de backward, o ambos a la vez que implican una resolución fácil del problema.

Se llevará a cabo el siguiente procedimiento:

- **Inicialización** de la variable backward, teniendo presente que todos los estados son equiprobables, se obtiene la ecuación 2.14; y para el instante $t = T$, quedará como se muestra a continuación (ecuación 2.15).

$$\beta_t(i) = \frac{1}{N} \tag{2.14}$$

$$\beta_T(i) = 1, \qquad 1 \leq i \leq N \tag{2.15}$$

- A través del **método inductivo**, se calculan las variables backward, de derecha a izquierda recursivamente, donde se tiene el mismo coste computacional que en el caso anterior.

$$\beta_t(i) = \sum_{i=1}^{N} \alpha_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \tag{2.16}$$

#### 2.3.2.2. Problema de decodificación. Algoritmo de Viterbi

Existe la necesidad de encontrar la secuencia de estados que explique la secuencia de observaciones dada. Este proceso de decodificación, puede solucionarse de acuerdo a varios criterios.

El primero de ellos puede ser la elección del estado más probable, en cada instante de tiempo. Para ello se tendría que maximizar en cada instante de tiempo la siguiente variable, la cual representa la probabilidad de ocupación de cada estado en el instante $t$:

$$\gamma_t(i) = P\left(q_t = S_i \,|\, O,\, \lambda\right) \tag{2.17}$$

Una forma probable sería calcular las variables forward y backward para calcular la variable anterior:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P\left(O \,|\, \lambda\right)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{2.18}$$

Y por último se tomaría el estado más probable:

$$q_t = \arg \max_{1 \leq i \leq N} \left[\gamma_t(i)\right], \qquad 1 \leq i \leq N \tag{2.19}$$

Este criterio no parece ser el más adecuado debido a que no tiene presente la probabilidad de ocurrencia de la secuencia de estados y, por ejemplo, al tener una probabilidad de transición entre estados nula, $a_{ij} = 0$, podría dar como resultado una secuencia de estados (secuencia de fonemas) que no tuviese sentido.

El segundo criterio, a su vez el más adecuado, consiste en elegir el camino completo de estados con mayor probabilidad global (secuencia de fonemas válida). Se resuelve utilizando el algoritmo de Viterbi. Antes de explicar en qué consiste, se definirá la siguiente variable auxiliar:

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} P\left[q_1 q_2 \ldots q_t = i,\, O_1 O_2 \cdots O_t \,|\, \lambda\right] \tag{2.20}$$

Donde $\delta_t(i)$ representa la mejor puntuación (máxima probabilidad) obtenida a través de una secuencia única de estados $(q_1 q_2 \cdots q_{t-1})$ hasta llegar en el instante $t$, al estado $i$. Una ventaja de este algoritmo es que si se conoce la anterior variable para todos los estados en el instante $t$, se pueden calcular, también para todos los estados, en el instante siguiente $(t+1)$:

$$\delta_{t+1}(i) = \left[\max_i \delta_t(i)\, a_{ij}\right] b_j\left(O_{t+1}\right) \tag{2.21}$$

Además de estas variables, se necesitará disponer también del estado $i$ que maximiza el argumento de la ecuación arriba enunciada. Se llevará a cabo almacenando todos sus valores para cada instante $t$ y cada estado $j$ en otra variable $\varphi_t(j)$.

El proceso completo es el siguiente:

- **Inicialización** de la variable auxiliar en el instante inicial y de la variable de almacenamiento de estados que proporcionan máximos.

$$\delta_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N \tag{2.22}$$

$$\varphi_1(i) = 0 \tag{2.23}$$

- Por **recursión**:

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i)\, a_{ij}\right] b_j\left(O_t\right), \qquad 2 \leq t \leq T, 1 \leq j \leq N \tag{2.24}$$

  Se guardan los valores máximos, que servirán posteriormente para obtener el camino óptimo:

$$\varphi_t(j) = \arg \max_{1 \leq i \leq N} \left[\delta_{t-1}(i)\, a_{ij}\right], \qquad 2 \leq t \leq T,\ 1 \leq j \leq N \tag{2.25}$$

- La parte de **finalización**:

$$P^* = \max_{1 \leq i \leq N} \left[\delta_T(i)\right] \tag{2.26}$$

$$q_T^* = \arg \max_{1 \leq i \leq N} \left[\delta_T(i)\right] \tag{2.27}$$

- El **backtracking** consistirá en realizar el camino desde el instante final al inicial, adoptando aquellos valores que maximizan cada paso de la etapa de recursión,

obteniendo así la secuencia de estados óptima:

$$q_t^* = \varphi_{t+1}\left(q_{t+1}^*\right), \qquad t = T-1, T-2, \cdots, 1 \tag{2.28}$$

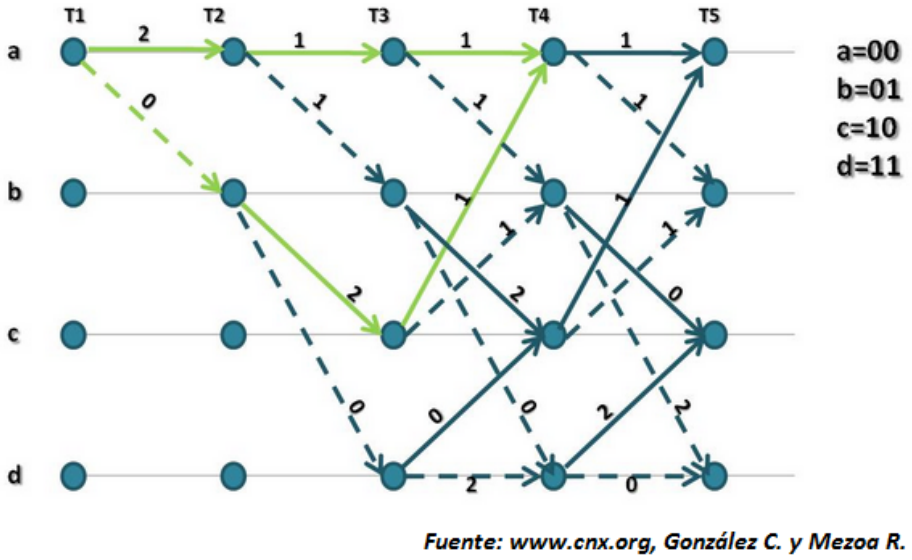


Figura 2.7: Esquema del algoritmo de Viterbi.

Según se muestra la imagen de la figura 2.7, el algoritmo de Viterbi funciona similar al algoritmo forward empleado en la fase de evaluación, teniendo también el mismo coste computacional que éste, del $O(TN^2)$.

#### 2.3.2.3. Problema de entrenamiento. Algoritmo de Baum-Welch

Para la estimación de los parámetros del modelo $\lambda = (A, B, \pi)$ que maximizan la probabilidad de observación $P(O \mid \lambda)$, se utilizará el algoritmo de Baum-Welch, que consiste en un caso particular del algoritmo de Expectation-Maximization (EM) aplicado a los HMMs. Se pretende, mediante una serie de iteraciones y bajo el criterio de máxima verosimilitud (Maximum Likelihood), ir encontrando máximos locales de la probabilidad de observación.

En primer lugar, se comenzará por definir una función auxiliar, que depende de los parámetros anteriores del modelo $\lambda$ y de la nueva estimación de ellos $\bar{\lambda}$:

$$Q\left(\lambda \mid \bar{\lambda}\right) = \sum_{Q} P\left(Q \mid O, \lambda\right) \log\left[P\left(O, Q \mid \bar{\lambda}\right)\right] \tag{2.29}$$

Según el algoritmo de EM, se garantiza que maximizando la función anterior respecto a los nuevos parámetros, se obtendrá una mayor verosimilitud en la siguiente iteración:

$$\max_{\bar{\lambda}} \left[ Q\left(\lambda \,|\, \bar{\lambda}\right) \right] \implies P\left(O \,|\, \bar{\lambda}\right) \geq P\left(O \,|\, \lambda\right) \tag{2.30}$$

Este proceso se repetirá para ir obteniendo nuevos parámetros en cada iteración, para seguir aumentando la verosimilitud hasta el punto en el que el algoritmo converja o el incremento de verosimilitud sea mínimo.

A continuación, se hará una distinción entre los dos pasos del algoritmo:

- En primer lugar, el **paso de Expectation**, tiene como misión calcular los elementos de la ecuación 2.29, que dependen del modelo anterior, principalmente el término $P\left(Q \,|\, O,\, \bar{\lambda}\right)$, es decir, las probabilidades de todas las secuencias de estados dados en el modelo anterior y las observaciones.
  Los parámetros que se van a estimar son:

  - ✧ La probabilidad de estar en el estado $S_i$ en el instante $t$, que vienen dada por la probabilidad de ocupación del estado anteriormente definido, que se podía calcular en función de las variables forward y backward (ecuaciones 2.20 y 2.21).
  - ✧ El número esperado de transiciones desde el estado $S_i$, la cual puede obtenerse a partir de las variables anteriormente mencionadas, de la siguiente forma:

$$\sum_{t=1}^{T-1} \gamma_t(i) \tag{2.31}$$

  - ✧ El número esperado de transiciones desde el estado $S_i$ al estado $S_j$. Para ello, inicialmente se definirá la probabilidad de transición entre ambos estados, y posteriormente se obtendrá el número de transiciones sumando los valores obtenidos:

$$\xi_t(i,\, j) = P\left(q_t = S_i,\, q_{t+1} = S_j \,|\, O,\, \lambda\right) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P\left(O \,|\, \lambda\right)} \tag{2.32}$$

$$\sum_{t=1}^{T-1} \xi_t(i,\, j) \tag{2.33}$$

- En segundo lugar, se tendrá el **paso de Maximization**, consistente en maximizar la función (ecuación 2.29) una vez ya calculados los parámetros en el paso

anterior, y dará lugar a unos parámetros nuevos, siendo los siguientes:

$$\bar{\pi}_i = frecuencia\ esperada\ en\ S_i\ en\ el\ instante\ (t=1) = \gamma_1(i) \tag{2.34}$$

$$\bar{a}_{ij} = \frac{número\ esperado\ de\ transiciones\ desde\ S_i\ a\ S_j}{número\ esperado\ de\ transiciones\ desde\ S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,\, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{2.35}$$

$$\bar{b}_j(k) = \frac{número\ esperado\ de\ veces\ en\ el\ estado\ j\ observando\ v_k}{número\ esperado\ de\ veces\ en\ el\ estado\ j} = \frac{\sum_{t=1,\, s.t. O_i = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{2.36}$$

Los algoritmos descritos anteriormente, que solucionan los tres problemas enunciados al principio de esta subsección, serán empleados en este proyecto, tanto en el proceso de entrenamiento como en el de evaluación del sistema de reconocimiento. La metodología seguida será descrita más adelante en la sección 4 y en ella serán empleadas herramientas de software que hacen uso de estos algoritmos.

# Capítulo 3

# Base de Datos

La base de datos de entrenamiento es uno de los elementos más importantes a la hora de generar un sistema de reconocimiento de habla. La calidad de la misma determina la viabilidad de un buen entrenamiento. En este caso se han utilizado diversas bases de datos cedidas por Telefónica I+D con las que se ha conformado la base de datos final de entrenamiento que se detalla a continuación.

## 3.1. Preparación de los datos

La base de datos consta de un listado de archivos de audio y de un listado de archivos de etiquetas con el mismo nombre pero distinta extensión. Dichos archivos de etiqueta contienen dos partes:

- La transcripción del archivo de audio que es imprescindible.
- Información adicional como puede ser: nombre de la base de datos, identificador de hablante, sexo, idioma, dialecto, SNR del archivo, *pitch* del hablante, fecha y lugar de grabación, eventos fonéticos (como clics, saturación, mala pronunciación...), tipo de ruido de fondo (si lo hubiera).

A partir de estas etiquetas generales se puede extraer el campo de transcripción y adaptarlo a lo que necesita cada programa de entrenamiento. Para el software HTK, el archivo de etiquetas debe tener cada palabra de la transcripción en una línea distinta, y que la última línea sea un símbolo de punto.

Los archivos deben estar transcritos correctamente para evitar entrenar fonemas de forma errónea. Los archivos de audio deben ser fieles a dichas transcripciones, asegurándose de que no están vacíos ni intercambiados.

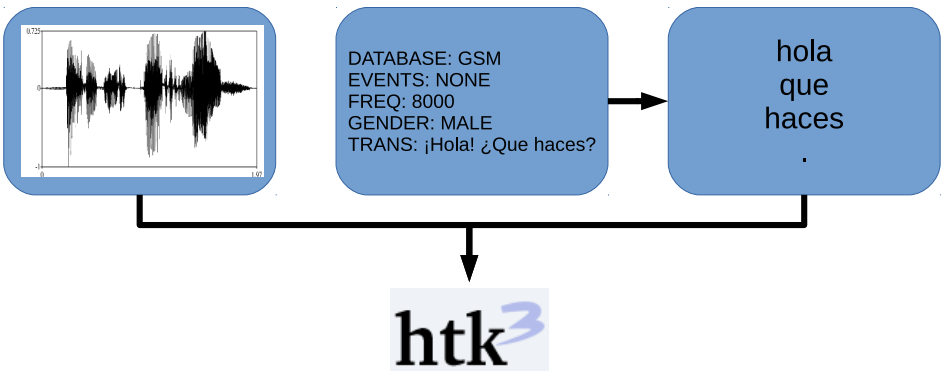


Figura 3.1: Archivos necesarios para HTK.

La riqueza de la base de datos se puede medir por diversos factores:

- El número de archivos con los que cuenta
- El número de frases y palabras distintas de entrenamiento
- El número de palabras y caracteres totales
- El promedio de caracteres por palabra
- El número de horas de audio y cuántas de ellas son de silencio y voz
- La cantidad de frases de hombres y mujeres
- El número de hablantes distintos
- La SNR promedio
- El *pitch* promedio.

Para la generación de la base de datos de entrenamiento se ha partido de bases de datos más pequeñas que se han ido limpiando y uniendo, eliminando archivos con demasiado ruido, incompletos, SNR muy baja, etc. También se ha reducido la tasa binaria de los archivos de audio con velocidad superior para ajustarlos a la calidad telefónica.

Para el entorno de evaluación, de esta base de datos se han extraído 1000 archivos representativos del total de la misma para hacer pruebas de reconocimiento con archivos no entrenados con la SNR promedio de la base de datos. También se han extraído otros 1000 archivos con ruido para hacer pruebas con archivos no entrenados ruidosos. Cada conjunto suma 70 minutos de audio.

Las bases de datos de origen son de diversa índole, siendo aproximadamente el 60 % voz telefónica (tanto de GSM como línea fija) grabada en diversos ambientes (calle, bares, coches, hogares...) y un 40 % voz limpia grabada en estudio.

El idioma de la base de datos es el español de España, conteniendo muestras de casi todas las comunidades autónomas para los distintos acentos.

Así, la base de datos final está en formato WAV, a 8 kHz, 16 bits y 1 canal.

En la siguiente tabla se pueden ver el resto de parámetros de la base de datos utilizada para el entrenamiento:

Tabla 3.1: Características de la base de datos de español.

| Características | Valor |
|---|---|
| Nº Archivos / Frases totales | 244.132 |
| Nº Frases distintas | 19.997 |
| Nº Palabras Totales | 1.063.866 |
| Nº Letras sin espacios | 5.746.114 |
| Promedio letras por palabra | 5,40 |
| Nº Palabras distintas | 11.182 |
| Nº de Horas de audio | 275,64 |
| Nº de horas sin silencios | 124,95 |
| Silencio ( %) | 54,67 |
| Voz ( %) | 45,33 |
| Frases hombres | 141.391 |
| Frases mujeres | 102.700 |
| Hombres ( %) | 57,92 |
| Mujeres ( %) | 42,08 |
| Hablantes distintos | 32.309 |
| Velocidad de muestreo (Hz) | 8.000 |
| Bits por muestra | 16 |
| Canales de audio | 1 |
| SNR Promedio | 33,7 |
| *Pitch* Promedio Hombres (Hz) | 123 |
| *Pitch* Promedio Mujeres (Hz) | 202 |

Otros factores relevantes son:

- La distribución de SNRs, para caracterizar mejor la calidad de los archivos de la base de datos.
- La distribución del *pitch* de los hablantes, que da indicaciones sobre las frecuencias fundamentales de la voz presente en la base de datos. Es un factor más relevante que diferenciar solamente entre hombres y mujeres.

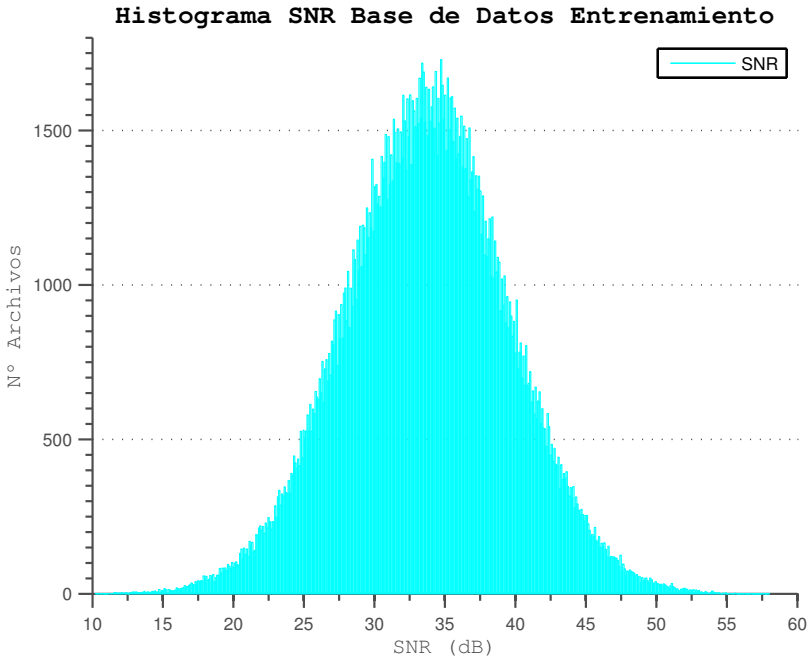


Figura 3.2: Histograma de la distribución de la SNR de la base de datos.

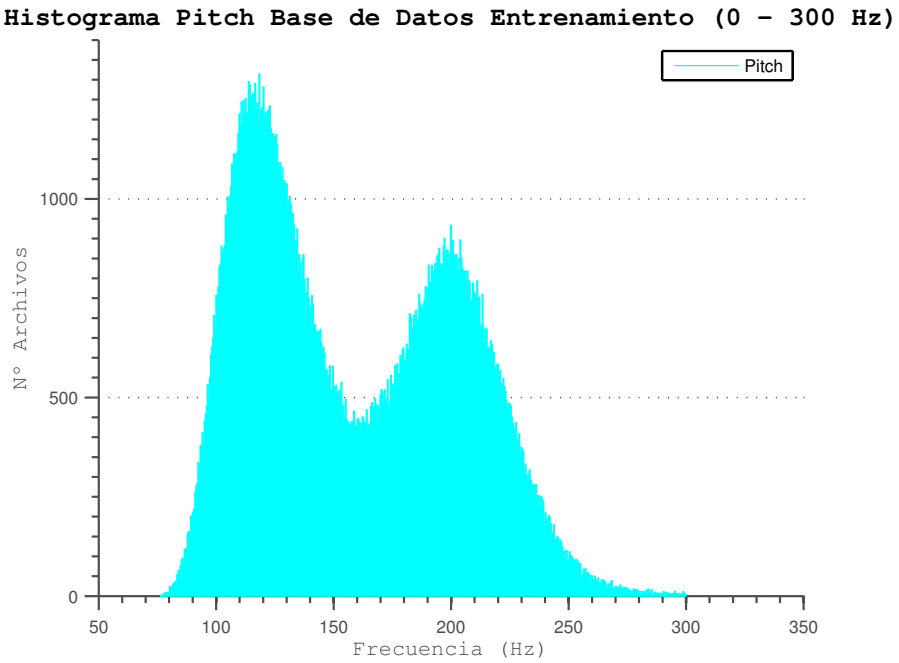


Figura 3.3: Histograma de la distribución del *pitch* promedio de la base de datos.

# Capítulo 4

# Generación del Sistema de Referencia

En este capítulo se detallan los pasos seguidos para la realización del sistema de referencia que se utilizará en las pruebas posteriores.

## 4.1. Introducción

El proyecto consta principalmente de tres partes:

- La primera es la configuración del sistema de entrenamiento y la definición de los procedimientos de entrenamiento que sirvan para estimar los modelos acústicos y de lenguaje iniciales.
- La segunda es utilizar un sistema de reconocimiento con el fin de:
  - ✧ Calcular la tasa de aciertos de los archivos entrenados para verificar si el entrenamiento progresa adecuadamente y tener una referencia del máximo que se puede alcanzar con el sistema.
  - ✧ Reconocer el conjunto de los archivos de test para obtener tasas efectivas en un entorno no controlado.
  - ✧ Introducir cambios en el sistema de entrenamiento y verificar si las nuevas modificaciones introducidas provocan una mejora o empeoramiento en el sistema global.
  - ✧ Decidir, a partir de los resultados experimentales, cuál va a ser el sistema base de entrenamiento.

- Una vez obtenido un sistema base de entrenamiento fiable, efectuar los nuevos experimentos que tienen el objetivo de mejorar el sistema inicial.

Se ha generado desde cero el sistema de entrenamiento a utilizar partiendo solamente de las herramientas del software HTK. Su realización ha llevado en torno al 65 % de las horas dedicadas, sumando el tratamiento de la base de datos, generación de modelos de lenguaje iniciales y generación de modelos acústicos de referencia.

En las tareas de reconocimiento y pruebas efectuadas se han invertido el resto de horas del proyecto, junto con la redacción de este documento.

### 4.1.1. Software utilizado

- **Linux**
  Se ha decidido efectuar este proyecto sobre un sistema operativo Ubuntu-Linux dada la facilidad de conseguir diversas librerías que son necesarias para la compilación y correcta ejecución del software HTK, Julius, y otros programas detallados más adelante.
  Además se ha utilizado el lenguaje "Shell Script (Bash)" para realizar la mayoría de las tareas, ya que permite generar bucles de forma sencilla, encadenar y paralelizar las llamadas a las herramientas HTK y Julius. También se aprovecharon las herramientas de tratamiento de datos y audio que llevan incorporadas la mayoría de las distribuciones de Linux.
  *AWK*, *sed* y *grep*, entre otras herramientas, han sido utilizadas para la limpieza de las etiquetas de la base de datos, para adaptarlas al formato HTK.

- **Sox**
  "Sound eXchange" es la "navaja suiza" del tratamiento de audio. Permite efectuar diversos tratamientos sobre un archivo de sonido.
  En este caso se ha utilizado para reducir la frecuencia de muestreo de los archivos hasta 8 kHz, para contar las horas de voz (excluyendo el silencio) de la base de datos, y durante el filtrado de la misma para detectar y eliminar los archivos vacíos.

- **Praat**
  Herramienta de uso libre de alta potencia en el tratamiento de señales de voz. Además de observar la forma de onda y el espectro de archivos de audio, se ha utilizado para calcular la SNR de los archivos de sonido, dado que permite separar la señal de voz de su silencio. También se ha utilizado en el cálculo del *pitch* (frecuencia fundamental) promedio de las grabaciones de la base de datos.

- **CMU-Cambridge SLM Toolkit**
  Para la generación de parte de los modelos de lenguaje se ha utilizado la herramienta "CMU-Cambridge Statistical Language Modeling Toolkit v2.05".
  Esta aplicación permite generar un modelo de lenguaje con cualquier combinación de N-gramas a partir de un texto y de un archivo de configuración que sirve para señalar los límites de las frases.

- **IRST Language Model**
  Se trata de otra completa aplicación para la generación de modelos de lenguaje con diversos índices de N-grama. También permite la mezcla de modelos de lenguaje y tiene diversos elementos personalizables.

- **HTK**
  Se ha utilizado el software HTK (Hidden Markov Model ToolKit) para la obtención de los modelos acústicos, inspirándose en el proceso básico de entrenamiento de su manual de referencia, HTKBook [1], pero con diversas modificaciones para adaptarlo a la base de datos y a los objetivos específicos del proyecto.


Figura 4.1: Logotipo de HTK 3.

Originalmente fue desarrollado en el "Machine Intelligence Laboratory" del Departamento de Ingeniería de la Universidad de Cambridge. Se trata de una herramienta de uso libre, pero no comercializable, cuya última actualización tuvo lugar en febrero de 2009.
A pesar de ello, es una de las herramientas más potentes de libre uso que existen en el mercado, utilizándose con éxito en tareas de reconocimiento de voz, síntesis de voz, reconocimiento de texto manuscrito y otras muchas tareas relacionadas con el aprendizaje automático.
Internamente se compone de diversas herramientas que permiten desde la generación de gramáticas cerradas, modelos de lenguaje sencillos con bigramas, grabación de archivos, de audio y editores de etiquetas de archivos hasta las herramientas más potentes como *"HERest"* (utilidad que aplica el algoritmo de Baum-Welch) que es la parte central y más costosa de todo el proceso de entrenamiento.
También consta de un reconocedor propio que utiliza el algoritmo de Viterbi,

"*HVite*", que también sirve para realizar alineamientos forzados que mejoran la calidad del entrenamiento al alinear los fonemas de los ficheros y asignarles marcas de tiempo.

- **Julius**
  Para el reconocimiento se decidió prescindir, tras un periodo de prueba, de las herramientas básicas incluidas en el HTK (*HVite* y *HDecode*) y utilizar en su lugar otra herramienta más moderna, potente, rápida, libre, actualizada y con más funcionalidades que es *Julius*.


Figura 4.2: Logotipo del software Julius.

Se trata de un reconocedor LVCSR (Large Vocabulary Continuous Speech Recognition, o reconocedor de habla continua de gran vocabulario) de código abierto y de alto rendimiento cuya particularidad consiste en efectuar dos pasadas de reconocimiento, utilizando en la primera bigramas (pares de palabras con una probabilidad de ocurrencia asociada) y en la segunda trigramas (tríos de palabras) de derecha a izquierda o entidades superiores de N-gramas hasta decagramas.
Con diccionarios de hasta 60.000 palabras se aproxima al tiempo real en la mayoría de las situaciones. A pesar de que el tiempo real no es objetivo de este proyecto, dicho programa ha ayudado a reducir el tiempo de finalización del mismo.
Utiliza modelos acústicos en formato ASCII de HTK y modelos de lenguaje en formato ARPA y soporta distintos tipos de HMMs, como los de estados compartidos y los de gaussianas compartidas.
Fue desarrollado inicialmente en la universidad de Kyoto y ahora es mantenido por el Instituto Nagoya de Tecnología.

## 4.2. Proceso de entrenamiento

En esta sección se detallan los pasos seguidos para generar los modelos acústicos y de lenguaje que se utilizarán en los experimentos.

### 4.2.1. Discriminación de los archivos de audio

Para evitar que el proceso de entrenamiento modele en exceso fonemas con demasiado ruido de fondo o ruidos explosivos como clics y golpes, se ha realizado un purgado previo de la base de datos en la que se eliminan todos los archivos con etiquetas que corresponden a archivos excesivamente mal grabados, aquellos con una SNR menor de 10 dB y los que tenían archivos de audio vacíos o incompletos.

### 4.2.2. Generación de archivos ".lab"

La base de datos de Telefónica tiene dos tipos de ficheros: los .wav que contienen los datos de la forma de onda y los .info que contienen los metadatos asociados a los ficheros de forma de onda. Por cada fichero .wav hay un fichero .info.

Lo que se debe hacer es extraer el campo de transcripción de los archivos ".info" que acompañan a los ".wav". Hubo que prestar especial atención a la codificación interna de los ficheros. Nuestro sistema utiliza por defecto el sistema de codificación UTF-8 y los ficheros venían en ISO-8859-1. Es recomendable convertirlos a la codificación utilizada en la propia terminal de Linux para facilitar el funcionamiento de HTK.

Después se coloca cada palabra de la transcripción en una línea diferente y se termina el archivo con un punto, dado que éste es el formato que HTK utiliza.

bocadillo
de
atún
.

### 4.2.3. Generación del diccionario

Para generar el diccionario de entrenamiento se separan todas las palabras de las transcripciones en líneas diferentes y se ordenan por orden alfabético eliminando duplicados.

Posteriormente, se fonetizan siguiendo las normas del castellano. En este proyecto de fin de carrera se ha utiliza un conjunto de 28 fonemas diferentes para el español.

Como el sistema de entrenamiento utiliza el modelo de pausa corta “sp” se añade este al final de cada palabra.

El diccionario toma la siguiente forma:

abadejo a bb a dd e x o sp
abandono a bb a n d o n o sp
...
zurrar z u rr a r sp
zurrón z u rr o n sp
silence sil
!ENTER sil
!EXIT sil

Las palabras "silence" "!ENTER" "!EXIT" son símbolos especiales. *HVite* usa "silence" para el alineamiento forzado y tanto *HVite* como Julius y otros reconocedores usan "!ENTER" Y "!EXIT" para indicar el comienzo y final de frases.

### 4.2.4. Generación de los archivos de transcripciones y listas

HTK precisa que todas las transcripciones estén juntas en un archivo de extensión “.mlf”, comenzado por la etiqueta “#!MLF!#” seguido del nombre de archivo entre comillas y la transcripción como en los archivos “.lab”:

#!MLF!#
"*/001.lab"
bocadillo
de
atún
.
"*/002.lab"
visitó
la
ciudad
de
cuenca
.

Después, aplicando los diccionarios, la herramienta *HLEd* (HTK Label Editor) separa cada palabra en sus fonemas:

```
#!MLF!#
"*/001.lab"
sil
b
o
k
a
dd
i
ll
o
sp
dd
e
sp
a
t
u
n
sp
sil
```

Por otro lado, se generan todas las listas de archivos que precisa HTK para el entrenamiento: las de archivos ".wav", archivos .mfc y archivos ".lab".

### 4.2.5. Generación del modelo de lenguaje

Para generar el modelo de lenguaje, primero se recopilan todas las frases de la base de datos de entrenamiento (3.1) en un único archivo y se edita para que comiencen y terminen con las etiquetas "!ENTER" y "!EXIT".

Inicialmente se utilizaba la herramienta *HBuild* de HTK para construir modelos de lenguaje, pero sólo permite generar modelos de lenguaje de bi-gramas. Por ello, a continuación, se procedió a utilizar la herramienta CMU-LM y se generó un modelo de lenguaje acotado (en el que todas las palabras a reconocer están en el modelo y no hay ninguna palabra del modelo fuera de las frases a reconocer) basado en trigramas

para el reconocimiento en Auto-test y de test.

### 4.2.6. Extracción de características

Tras realizar diversas pruebas iniciales, se decidió efectuar una extracción de características mediante Mel Frequency Cepstral Coefficients, de ahora en adelante MFCCs.

La principal ventaja de utilizar coeficientes cepstrales es que normalmente están decorrelados, lo que permite que se puedan utilizar covarianzas diagonales en los HMMs.

Para el cálculo de los MFCCs se empleó una ventana de Hamming con pre-énfasis y normalización de energía. El tamaño de la ventana de análisis es de 25 ms con un desplazamiento de 10 ms.

Los MFCCs son generados mediante la herramienta de HTK, *HCopy* que efectúa dicha extracción de características generando un archivo con el mismo nombre que el ".wav" pero con extensión ".mfc". Estos archivos son necesarios para el entrenamiento mediante *HERest* y sirven para acelerar el reconocimiento al no tener que calcularlos de nuevo cada vez.

Los ficheros de audio están en formato WAV a 8 kHz y los vectores así generados tienen 39 componentes.

### 4.2.7. Generación de los HMM

A continuación se detallan cada uno de los pasos que se han seguido para conseguir los HMM finales que se utilizarán en la fase de pruebas.

#### 4.2.7.1. Generación de mono-fonemas

El primer paso consiste en crear un archivo prototipo que contenga un ejemplo del vector de características a utilizar, una representación de cuántos estados va a tener el modelo (tres en nuestro caso más uno inicial y uno final) así como la forma de la matriz de transiciones.

A continuación, dado que no se dispone de un alineamiento inicial de los ficheros de audio con sus marcas de tiempo, se llama a la función "*HCompV*" de HTK que se encarga de transformar este modelo prototipo inicial en otro en el que se añaden todas las medias y las varianzas globales que se han extraído de toda la base de datos de entrenamiento y establece todas las gaussianas a este valor. De disponer de un alineamiento inicial se podría utilizar la función *HInit*.

Una vez obtenidos estos datos iniciales, se genera el MMF copiando el modelo del *proto* en cada uno de los fonemas que utilizamos.

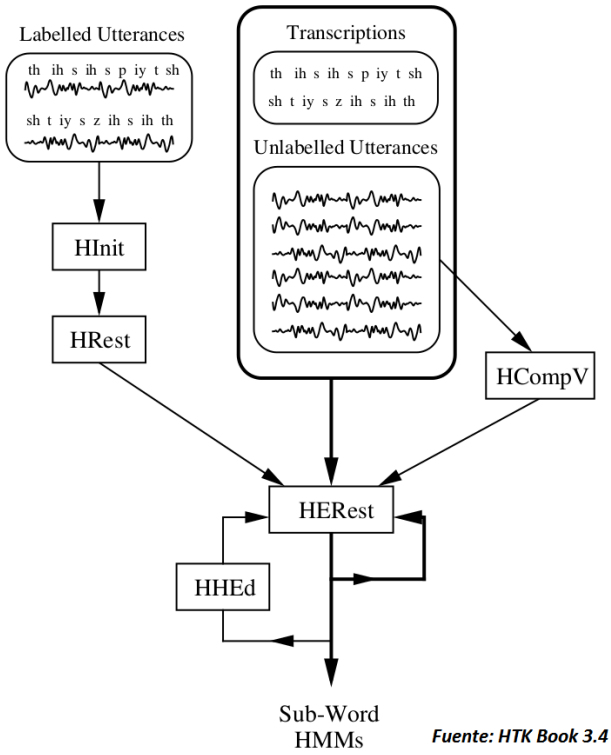


Figura 4.3: Entrenamiento de HMMs de fonemas.

Se genera también a partir del archivo proto y del archivo *'vFloors'*, calculado por *HCompV*, el archivo *'macros'* que se utiliza para, en cada nueva reestimación, indicar la forma del archivo *'hmmdefs'* a generar.

A continuación, ya se puede comenzar la reestimación de los modelos mediante la herramienta *HERest*, que aplicando el algoritmo de Baum-Welch [1], se encarga de calcular los modelos para cada uno de los fonemas que se ha definido a partir de los datos de entrenamiento.

#### 4.2.7.2. Optimización de los modelos de silencio

Para hacer el sistema más robusto ante posibles ruidos impulsivos, se ligan los estados 2 a 4 y 4 a 2. Esto permite que sean estados individuales los que absorban los ruidos impulsivos. También, ligar el estado 4 al 2 permite que esto ocurra sin salir del modelo.

Este ligado se realiza mediante un *script* que se envía a la herramienta *HHEd*.

A la par, se añade un modelo de pausa corta de un estado llamado 'sp', que comparte su estado central con el fonema 'sil' pero que se trata de un "tee-model",

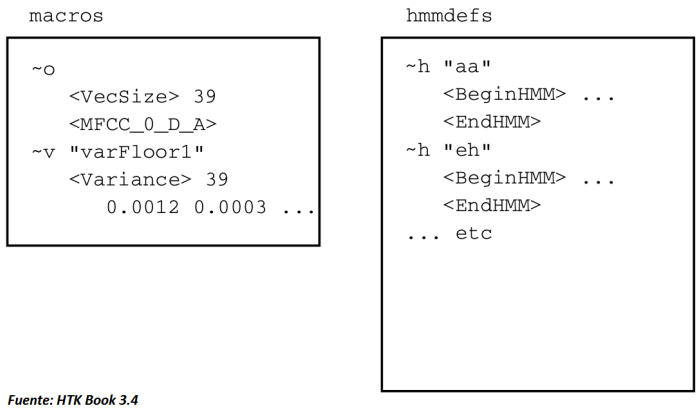


Figura 4.4: Ejemplo de archivo de 'macros' y de 'hmmdefs'.

que es aquel que tiene una transición directa de la entrada a la salida.

Después se vuelven a reestimar los modelos utilizando la herramienta *HERest*.

#### 4.2.7.3. Alineamiento forzado y entrenamiento inicial

Una vez se dispone de los modelos de silencio mejorados, se pasa a realizar un alineamiento forzado. Esto sirve, aprovechando los modelos que ya se han logrado generar, para alinear a nivel de fonema la ocurrencia de cada uno de ellos dentro de cada archivo de audio y extraer marcas de tiempo para mejorar así el trabajo que realiza *HERest*.

Dicho alineamiento forzado se efectúa mediante la herramienta de HTK "*HVite*", que permite hacer un reconocimiento inicial de los archivos de la base de datos para extraer las marcas de tiempo de cada fonema. El archivo alineado de salida se llama "aligned.mlf".

Una vez efectuado dicho proceso, se vuelven a reestimar los modelos.

#### 4.2.7.4. Generación de tri-fonemas

El siguiente objetivo es generar un set de tri-fonemas dependientes del contexto.

En primer lugar, se utiliza la herramienta *HLEd* para expandir el archivo de etiquetas recién alineado en sus posibles tri-fonemas. Además, genera una lista de todos los tri-fonemas vistos en las frases de entrenamiento.

La representación final de los fonemas se define como "word-internal" y toma la siguiente forma:

sil h+o h-o+l o-l+a l-a sp m+u m-u+n u-n+d n-d+o d-o sp

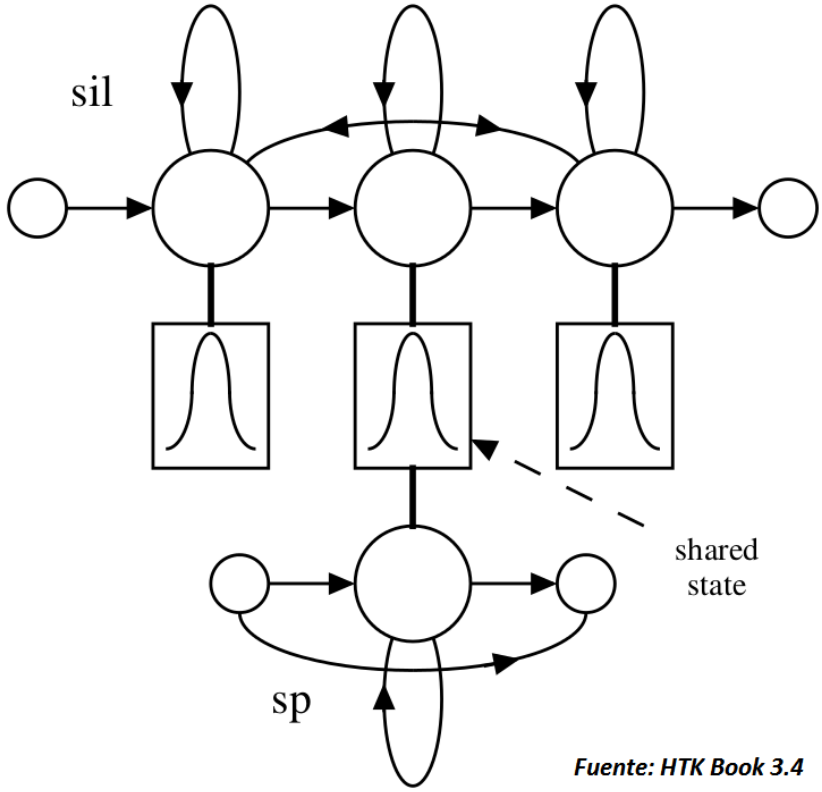


Figura 4.5: Mejora del modelo de silencio.

Aunque para algunos reconocedores se podrían listar sólo los tri-fonemas encontrados en las frases de entrenamiento, al utilizar en las pruebas iniciales *HDecode*, se optó directamente por generar todos los tri-fonemas posibles de las combinaciones de los mono-fonemas sin 'sp' y sin 'sil'. Esta lista que contiene todos los posibles tri-fonemas se llama "fulllist".

Mediante un script que se aplica a la herramienta *HHEd* se procede a generar todos los tri-fonemas mediante la clonación de su fonema central, volviendo a reestimar con *HERest* a continuación. En este punto ya se ha conseguido una lista de HMMs continuos de tri-fonemas dependientes del contexto.

#### 4.2.7.5. Ligado de estados de tri-fonemas

Con el fin de reducir el coste computacional y de caracterizar mejor los tri-fonemas de los que no se disponen datos de entrenamiento suficientes, se procede a efectuar un ligado de estados.

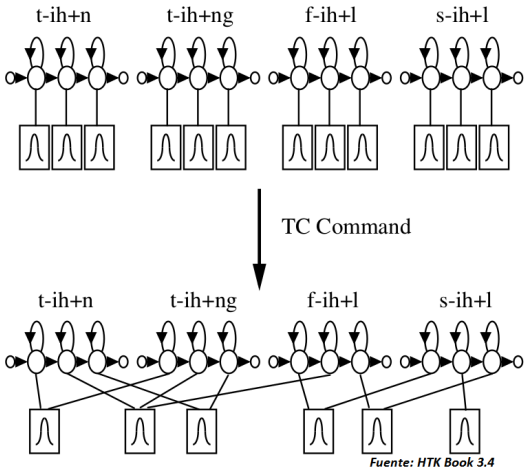


Figura 4.6: Ejemplo de modelo de estados ligados.

Aplicando un árbol de decisión, se unen los estados similares de los tri-fonemas que no tienen realizaciones suficientes a aquellos que tienen más realizaciones. Ajustando el umbral 'RO' en el script de *HHEd*, se puede ajustar el nivel de clusterización. Un valor muy alto puede causar que todos los tri-fonemas se acaben asociando a sus mono-fonemas iniciales, lo que evidentemente provocaría reconocimientos más veloces pero en principio de peor calidad.

Dejando este umbral a 0 se eliminan sólo los estados que no han sido entrenados, aproximándolos a sus estados más parecidos. En función de cada base de datos este factor puede variar por lo que es recomendable realizar un barrido en un rango de valores de esta variable para encontrar su óptimo.

**mktri.hed**

```
CL list
TI T_a {(*-a+*,a+*,*-a).transP}
TI T_b {(*-b+*,b+*,*-b).transP}
...
```

**tree.hed**

```
RO 150 stats

QS "L_NonBoundary" { *-* }
QS "R_NonBoundary" { *+* }
QS "L_Silence" { sil-* }
QS "R_Silence" { *+sil }
QS "L_Vocales" { a-*, e-*, i-*, o-*, u-* }
QS "R_Vocales" { *+a, *+e, *+i, *+o, *+u }
...
```

#### 4.2.7.6. Incremento paulatino del número de gaussianas

Como paso final, para mejorar la calidad del sistema, lo que se realiza es un incremento del número de gaussianas.

Una gaussiana puede no ser suficiente para modelar todas las posibles realizaciones de ese fonema. Por ello, HTK permite incrementar el número de gaussianas para mejorar el modelo, incrementando así la tasa de reconocimiento.

El procedimiento utilizado para incrementar las gaussianas ha sido la herramienta *HHEd* con el comando 'MU' (Mixture Up). Se incrementa en uno el número de gaussianas y luego se realizan una o más reestimaciones con *HERest*. Después, se repite el proceso hasta llegar al número de gaussainas requeridas.

Tras varias pruebas que se detallan en el punto siguiente 4.3 se ha llegado a la conclusión de que, para nuestra base de datos, el número óptimo de gaussianas es de 16.

## 4.3. Reconocimiento de verificación

Una vez terminado el proceso de entrenamiento, se procede a la evaluación de la calidad del sistema generado.

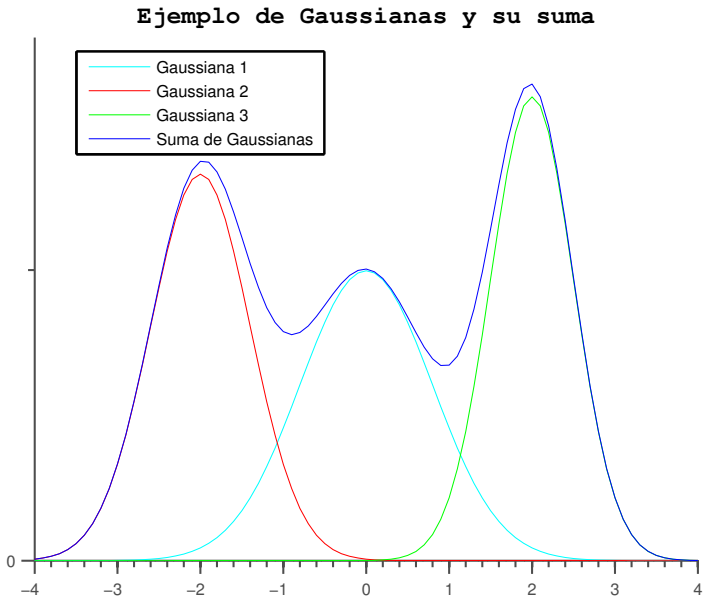


Figura 4.7: Ejemplo de suma de varias gaussianas.

Como se ha mencionado anteriormente, los primeros pasos en la evaluación y mejora de este reconocedor se dieron utilizando las herramientas básicas incluidas *HVite* y *HDecode*. Se abandonaron por su pobre rendimiento tanto en velocidad como en tasas obtenidas, cambiándolos por otro reconocedor más versátil, moderno y en constante evolución como Julius.

Julius permite utilizar los modelos acústicos generados mediante HTK, por lo que se puede hacer una evaluación directa de la calidad de los mismos. Dispone de infinidad de parámetros a configurar, siendo muy importante ajustarlos a la forma en que se hayan entrenado en HTK.

Tiene la posibilidad de extraer directamente el vector de características de los archivos ".wav", lo que permite tratar cualquier archivo sin tener que efectuar la extracción de este vector mediante la herramienta *HCopy* cada vez que se varíe el tamaño de ventana u otros parámetros.

Se ha realizado un script para convertir la salida de Julius a formato HTK con el fin de poder utilizar el programa *HResults* de HTK para evaluar la salida.

Dicho programa da una salida en formato estándar métrico US NIST FOM.

```
---------------------------- Overall Results -----------------
SENT: %Correct=72.40 [H=724, S=276, N=1000]
WORD: %Corr=95.83, Acc=92.12 [H=6619,D=55,S=233,I=256,N=6907]
============================================================
```

Siendo "SENT - Correct" la tasa de frases correctas totales:

$$\%Correct = \frac{H}{N} \times 100\,\% \tag{4.1}$$

"WORD - Corr" la de palabras correctas:

$$\%Corr = \frac{H}{N} \times 100\,\% \tag{4.2}$$

"WORD - Acc" es la precisión, teniendo en cuenta también el número de inserciones negativamente:

$$Acc = \frac{H - I}{N} \times 100\,\% \tag{4.3}$$

Se ha utilizado un diccionario de unas 12.000 palabras, y un modelo de lenguaje de trigramas acotado y adaptado a las frases, donde no hay palabras desconocidas y todas las palabras que se deben reconocer están en el diccionario.

Se han generado 6 grupos distintos de control para la evaluación del reconocedor. Los 4 primeros son de Auto-test, es decir, se prueba a reconocer archivos que se han empleado para entrenar los modelos acústicos. Los otros dos grupos son de archivos que no forman parte de la base de datos de entrenamiento.

La segmentación se ha hecho entorno a la SNR de los archivos para ver de paso cómo afecta ésta a la calidad del reconocimiento. Además, todas las listas de test están balanceadas, es decir, que contienen un número proporcional de los archivos de cada base de datos de origen. Contienen 1.000 archivos cada una, que se corresponde con aproximadamente 70 minutos de audio:

Una vez definidos los grupos de control, se efectuó un experimento para definir el número optimo de gaussianas a utilizar:

- **Elección del número óptimo de gaussianas:**
  Este experimento se basó en entrenar los HMM incrementando de una en una hasta las 64 gaussianas. Este entrenamiento llevó cerca de 72 horas.
  Una vez generados los modelos, se utilizaron para el reconocimiento todos ellos entre 1 y 64 gaussianas. Se reconocieron los seis conjuntos de archivos y se di-

bujaron gráficas para ver la tendencia.

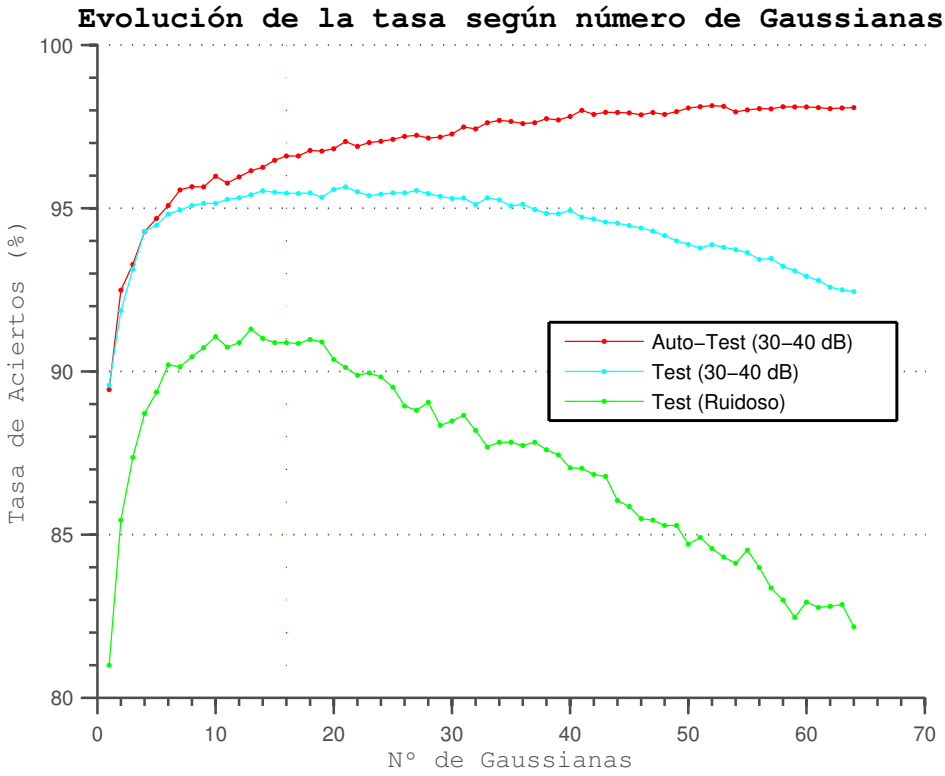


Figura 4.8: Gráfica para la selección de gaussianas.

Aunque en los reconocimientos de Auto-test se observa que incrementa la tasa de acierto siempre que se incrementa el número de gaussianas, al contrastarlo con las bases de datos de test se observa que a partir de las 16 gaussianas la tasa de aciertos comienza a degradarse.

Esto se debe a que al incrementar el número de gaussianas así como las reestima-

Tabla 4.1: Grupos de archivos de prueba.

| Grupo | Lista de Archivos | Nº Archivos | Duración (min.) | Rango SNR (dB) |
|---|---|---|---|---|
| Auto-test | Autotest_10_20 | 1000 | 70 | 10 - 20 |
| | Autotest_20_30 | 1000 | 70 | 20 - 30 |
| | Autotest_30_40 | 1000 | 70 | 30 - 40 |
| | Autotest_40_99 | 1000 | 70 | 40 - 99 |
| Test | Test_30_40 | 1000 | 70 | 30 - 40 |
| | Test_ruido | 1000 | 70 | $< 10$ |

ciones, se están sobre-adaptando los HMM a la base de datos de entrenamiento, lo que provoca que al llegar nuevos locutores o frases no entrenadas se reconocerán peor, y esto dista del objetivo del sistema.
Contrastando todas las gráficas y tablas se decidió tomar 16 gaussianas. Era el número óptimo, en la mayoría de los casos, en que se alcanzaba la máxima tasa de aciertos.
Aunque ocasionalmente 18 o 21 gaussianas dieran tasas similares, se decidió utilizar 16 porque a misma tasa de aciertos el procesado era más rápido.
Para más información consultar el apéndice A.

- **Elección del número óptimo de archivos de entrenamiento:**
  Este experimento se realizó para tratar de determinar cuál es el número óptimo de ficheros de entrenamiento.
  Se entrenaron 26 modelos con distinto número de archivos hasta las 16 gaussianas y se hicieron reconocimientos de prueba para validar la calidad del sistema.
  Se extrajeron las siguientes gráficas de los mismos.

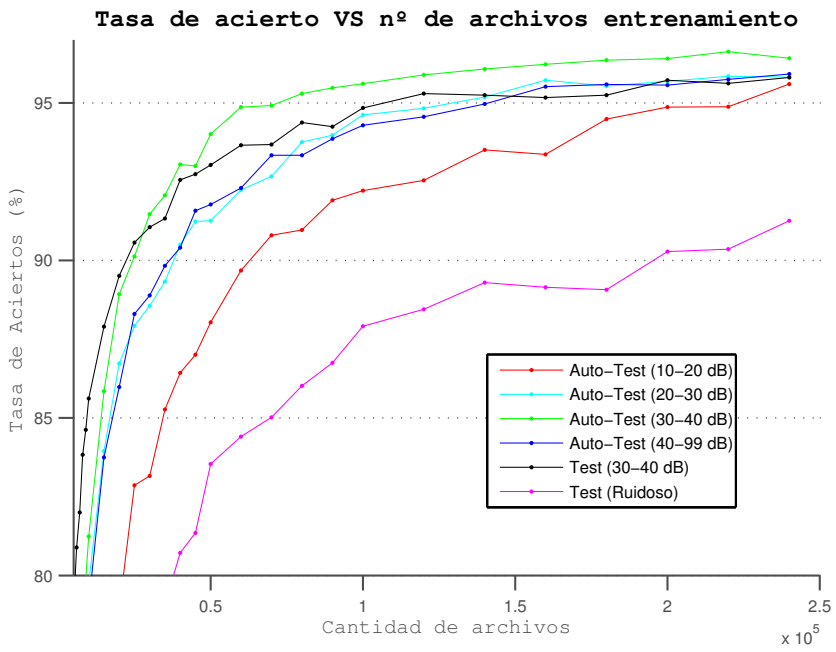


Figura 4.9: Gráfica para la selección del número de archivos de entrenamiento en unidades naturales.

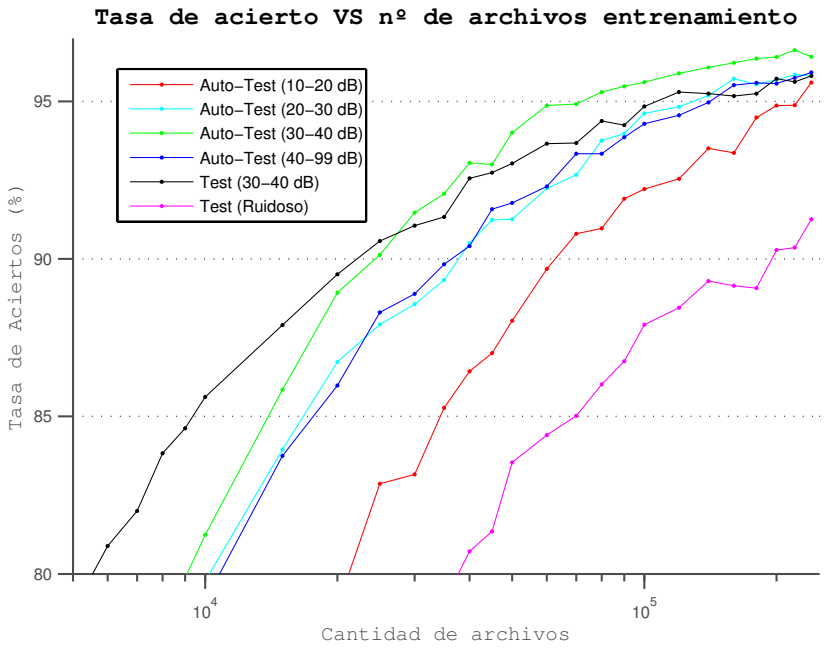


Figura 4.10: Gráfica para la selección del número de archivos de entrenamiento en unidades logarítmicas.

En ellas se puede observar una tendencia logarítmica entre el número de archivos y el incremento de la tasa de aciertos.

Aunque para los archivos de mayor calidad el incremento en la tasa de aciertos tiende a estabilizarse a partir de los 120.000 ficheros de entrenamiento con incrementos cada vez menores en los aciertos. Para los archivos de peor calidad, comprendidos en el grupo "Test Ruidosos" y en el grupo "Auto-test 10-20 dB", todavía se observan grandes incrementos cercanos al 1 % al pasar de 220.000 a 240.000 ficheros de entrenamiento.

Esto sugiere que hay margen para añadir más ficheros de entrenamiento con el fin de mejorar especialmente el reconocimiento en situaciones de mala relación señal a ruido así como en los casos que además de una pobre SNR se dan también eventos fonéticos, como golpes, chasquidos...

Desafortunadamente no se dispone de más archivos de entrenamiento con los que prolongar esta prueba.

Por tanto se deben utilizar todos los archivos disponibles para el entrenamiento.

Para los resultados completos consultar el apéndice B .

Tras estos experimentos iniciales, reconociendo las listas de archivos explicadas

anteriormente se han podido generar diversas modificaciones en el sistema hasta obtener tasas entre el 95 % y el 98 % de acierto en Auto-test y entre el 95 % y 97 % en test. Una vez llegados a este punto, se consideró que el sistema estaba listo para la fase de experimentación.

Los intervalos de confianza se han calculado para el 95 %.

Tabla 4.2: Resultados pruebas test para 16 gaussianas.

| Grupos de archivos | % WORD | % ACC | % SENT |
|---|---|---|---|
| SNR 10-20 dB (Auto-Test) | 95,74 ± 1,04 | 90,28 | 86,26 |
| SNR 20-30 dB (Auto-Test) | 96,18 ± 0,66 | 94,04 | 77,78 |
| SNR 30-40 dB (Auto-Test) | 96,60 ± 0,71 | 93,92 | 77,19 |
| SNR 40-99 dB (Auto-Test) | 95,94 ± 0,72 | 94,38 | 82,23 |
| SNR 30-40 dB (Test) | 95,46 ± 0,89 | 92,27 | 73,47 |
| SNR Ruidosos (Test) | 90,88 ± 1,44 | 83,47 | 76,35 |

# Capítulo 5

# Optimización del Sistema y Resultados

Una vez terminado el sistema de referencia, el siguiente objetivo es mejorarlo. Para ello las pruebas se centrarán en la optimización de la ventana de entrenamiento y reconocimiento para adaptarla de una forma más precisa a la frecuencia fundamental promedio del locutor.

## 5.1. Motivación específica y problema a resolver

Normalmente se utiliza una ventana de 25 ms [1] en reconocimiento de habla continua. Esta ventana permite seleccionar como mínimo dos periodos de los sonidos sonoros de una señal de voz de 80 Hz de *pitch* promedio, tres periodos de una señal de 120 Hz y cuatro periodos de una señal de 160 Hz. Esto es importante ya que, para estimar correctamente los parámetros de los modelos de Markov, es recomendable disponer de al menos uno o dos periodos en el tramo a analizar para que el algoritmo de Baum-Welch pueda extraer correctamente las características de esta señal. Por lo tanto, se puede decir que la ventana de 25 ms está optimizada para los fonemas sonoros de las señales de frecuencia fundamental promedio de 120 Hz.

El problema de este tamaño de ventana comienza a partir de estos 160 Hz de *pitch* (frecuencia fundamental de la voz) promedio, donde ya se empieza a tener más de cuatro periodos de la señal, y por debajo de los 80Hz, donde se tienen menos de dos periodos.

Como se ha dicho en la sección 2.3 se debe garantizar una pseudoestacionariedad durante el proceso de extracción de características, y si se toman demasiados periodos esta condición podría verse afectada [9].

La frecuencia fundamental media de la voz masculina está centrada en los 110-120 Hz. Por contra, tal como se ha visto en el capítulo 3 y en [14], la voz femenina suele estar centrada en 180-200 Hz. Esto significa que una mujer que tenga un *pitch* medio de 240 Hz, provocaría que se tomasen 6 periodos de la señal si se usa una ventana de 25 ms, lo que no garantiza necesariamente que dicho tramo sea pseudoestacionario.

Como se ve en la figura 5.1, en 100 ms se cuentan muchos más periodos para una vocal 'A' en la voz femenina que en la masculina.

Los experimentos presentados en [14] y los realizados para este proyecto demuestran que la tasa de reconocimiento decae rápidamente a partir de los 220 Hz de *pitch*.

De esta problemática surge la idea, ya evaluada en el pasado por diversos autores, de generar por lo menos dos tipos de modelos: uno orientado a la voz masculina y otro orientado a la voz femenina. Con ello se busca evitar interferencias durante el entrenamiento y que cada uno se reconozca mejor por separado.

En este caso, en lugar de discriminar por sexo, se ha decidido abordar el problema diferenciado por la frecuencia fundamental, para evitar los problemas con hombres que tengan un tono muy agudo, o mujeres que lo tengan muy grave. Además, no siempre se sabe si el hablante es hombre o mujer, pero calcular su frecuencia fundamental es un proceso de bajo coste computacional que puede realizarse de forma previa.

Así, el objetivo de este capítulo será:

- Evaluar la mejora que se obtiene al utilizar un reconocedor de voz cuya ventana de análisis esté adaptada al *pitch* medio de la señal de entrada y cómo se comporta cuando no está adaptada.
- Deducir cómo afecta el volumen de datos de entrenamiento de cada tamaño de ventana sobre la calidad de los HMM.

## 5.2. Condiciones experimentales

En esta sección se describen las condiciones experimentales. Se describen las ventanas de análisis elegidas, la base de datos, los modelos de lenguaje y los modelos acústicos que se van a utilizar.

### 5.2.1. Ventanas de análisis

Para efectuar estas pruebas de la forma más sistemática posible, se obtuvo primero el *pitch* medio de cada grabación de la base de datos para poder agrupar los ficheros en grupos a partir de su *pitch*.

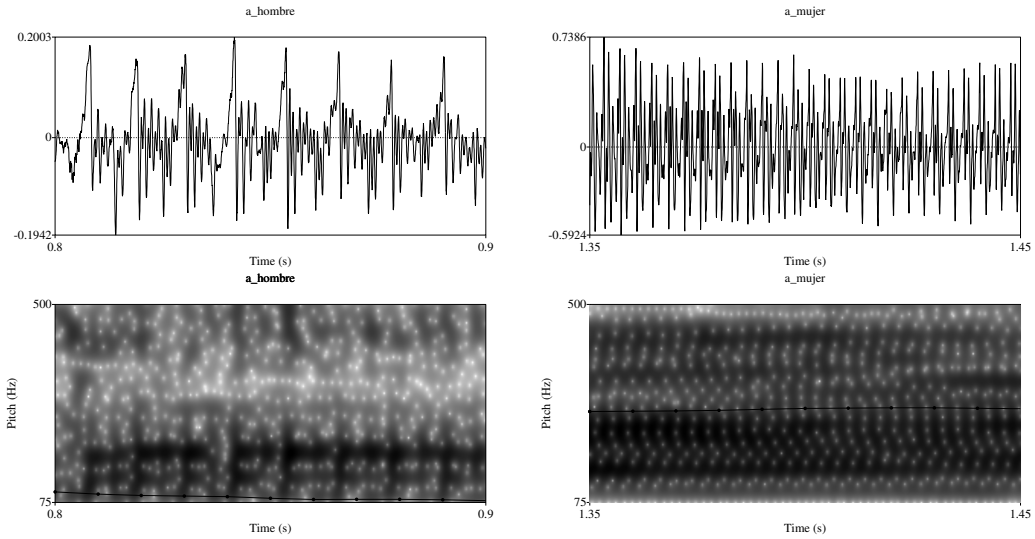


Figura 5.1: Letra 'A' dicha por un hombre y una mujer.

Basándose en los datos extraídos del histograma de la figura 3.3, se dividió el rango de variación del *pitch* medio en cinco tramos cuyas frecuencias centrales se corresponden con cinco tamaños de ventana de análisis.

De mayor a menor coincidirán con: el *pitch* mínimo de los hombres, el *pitch* medio de los hombres, el *pitch* promedio entre hombres y mujeres, el *pitch* medio de las mujeres y el *pitch* máximo esperado para las mujeres.

Los datos se condensan en la siguiente tabla:

Tabla 5.1: Detalle de las ventanas de análisis de los experimentos.

| | | | | | |
|---|---|---|---|---|---|
| Tamaño de ventana asociada (ms) | 37,5 | 25 | 18,75 | 15 | 12,5 |
| *Pitch* medio (Hz) | 80 | 120 | 160 | 200 | 240 |
| Nº muestras ventana (a 8 kHz) | 300 | 200 | 150 | 120 | 100 |
| Nº muestras de desplazamiento del 40 % (a 8 kHz) | 120 | 80 | 60 | 48 | 40 |

### 5.2.2. Bases de datos de experimentos

Tanto la base de datos de entrenamiento como la base de datos de prueba se han dividido en función de su frecuencia fundamental media. Los grupos se han obtenido agrupando los ficheros cuya frecuencia fundamental media tenga una desviación máxima de ±20Hz respecto a la frecuencia central correspondiente de cada ventana de análisis. Se obtienen así cinco grupos de test:

Tabla 5.2: Detalle de las bases de datos de entrenamiento/Auto-test y de test.

| | | | | | |
|---|---|---|---|---|---|
| Tamaño de ventana (ms) | 37,5 | 25 | 18,75 | 15 | 12,5 |
| Rango frecuencia (Hz) | 0-100 | 100-140 | 140-180 | 180-220 | 220-máx |
| BD Auto-test (nº archivos) | 12.000 | 94.400 | 46.700 | 67.400 | 22.600 |
| BD de test (nº archivos) | 250 | 250 | 250 | 250 | 250 |

Además, se utilizan una serie de vídeos extraídos de distintas fuentes de Internet y transcritos manualmente para verificar cómo se comporta este sistema en un entorno menos controlado. Este conjunto está dividido en tres temáticas: economía, deportes y moda. Cada una de ellas cuenta con 70 minutos, que coincide con la duración de los mil archivos utilizados en las pruebas de elección de gaussianas y prueba inicial del sistema descrito en 4.3. Dado que estos vídeos suelen venir muestreados a más de 8 kHz, se extrae el audio y se reduce su frecuencia de muestreo a 8 kHz.

Tabla 5.3: Detalle de la base de datos de vídeos.

| Tema | Locutores | Voz | Dist. micro | Música | Nº Vídeos | Duración |
|---|---|---|---|---|---|---|
| Economía | Uno principal | Masc. | Cerca | No | 3 | 70 min. |
| Moda | Uno/video | Fem. | Lejos | Si | 5 | 70 min. |
| Deporte | Múltiples | Ambos | Cerca | Si | 8 | 70 min. |

### 5.2.3. Diccionarios y modelos de lenguaje

Como se comprobó que todas las palabras que aparecían en la base de datos de test estaban en el diccionario de la base de datos de entrenamiento, compuesto por unas 12.000 palabras de español, se utilizó éste para el reconocimiento de ambas.

Como modelo de lenguaje, se ha utilizado tanto para la base de datos de Auto-test como para la de test el mismo modelo de trigramas utilizado en las pruebas iniciales del sistema base descrito en 4.3.

Se trata de un modelo de lenguaje y de un diccionario acotados, es decir, en los que todas las palabras a reconocer están dentro de ellos y no sobra ninguna que no se halle en las frases. Sirve para ver dónde se sitúa el máximo teórico de nuestro sistema.

También se utilizan un diccionario y un modelo de lenguaje acotados para cada uno de los temas de los vídeos extraídos de Internet por el mismo motivo.

### 5.2.4. Modelos acústicos para los experimentos

Para realizar las pruebas se han generado 21 modelos acústicos distintos. A pesar de la optimización del proceso de entrenamiento mediante la paralelización de 6 CPUs,

la generación de los modelos acústicos es un proceso computacional muy costoso, tardando más cuantos más archivos se utilicen y cuanto menor sea el tamaño de ventana utilizado. Así en promedio, cada modelo entrenado ha llevado aproximadamente 24 horas de CPU.

En cuanto a las pruebas de reconocimiento, también se han paralelizado a 6 CPUs y del mismo modo tardan más cuanto menor es la ventana elegida.

Para la parte de Auto-test se ha reconocido la totalidad de la base de datos, que como se muestra en la tabla 3, cuenta con unas 300 horas de audio. Cada reconocimiento ha durado en promedio 36 horas. Cada CPU ha procesado unas 50h de audio, lo cual indica que todavía se reconoce más rápido que el tiempo real para este tamaño de diccionario (12.000 palabras).

Estos 21 modelos acústicos se encuadran en el marco de cuatro métodos distintos de entrenamiento que se detallan a continuación:

- **Entrenamiento con la base de datos completa:** Donde se entrenan 5 modelos acústicos para cada una de las cinco ventanas utilizando siempre todos los archivos disponibles en la base de datos (244.000).

- **Entrenamiento discriminado por *pitch*:** En esta prueba se ha efectuado una división manual de la base de datos. Para cada tamaño de ventana de entrenamiento, se seleccionan los archivos que tengan entre 2 y 4 periodos al entrenarlos con la misma. Por tanto, habrá archivos que se empleen para entrenar los modelos de distintos tamaños de ventana de análisis. Por otro lado, hay que destacar que este método deja sólo 25.000 archivos para el entrenamiento de la ventana de 37.5 ms, por lo que los resultados obtenidos en estas pruebas podrían ser mucho más fiables de utilizarse un volumen mayor de datos.
  De esta manera utilizamos:

Tabla 5.4: Detalle del entrenamiento discriminativo por *pitch*.

| Frecuencia central (Hz) | 240 | 200 | 160 | 120 | 80 |
|---|---|---|---|---|---|
| Tamaño de ventana (ms) | 12,5 | 15 | 18,75 | 25 | 37,5 |
| Número de archivos | 113.000 | 148.000 | 185.000 | 131.000 | 25.000 |

- **Entrenamiento con menor número de archivos manteniendo la distribución de *pitch* original de la base de datos:** Este entrenamiento con número de archivos reducido, se efectúa para poder hacer una comparación directa con el entrenamiento discriminado por *pitch*. Así se entrena cada modelo

con el mismo número de archivos que en el caso anterior (tabla 5.4), pero en vez de seleccionarlos manualmente se mezclan de forma que, al calcular de nuevo el histograma de su *pitch*, mantenga la misma distribución que la base de datos completa. Así se puede medir la mejora entre utilizar archivos del rango específico de *pitch* a usarlos con la distribución global de la base de datos.

- **Entrenamiento de ventana única óptima:** Por último, lo que se busca es confirmar o desmentir que el tamaño óptimo para una única ventana son los 25 ms utilizados habitualmente. De esta forma se entrenó utilizando todos los archivos con las siguientes ventanas adicionales a las del primer caso: 20, 22.5, 27.5, 30, 32.5 y 35 ms.

## 5.3. Experimentos

En esta sección se describen los experimentos realizados así como las conclusiones que se obtienen de ellos. Los resultados completos de esta sección se hallan en el apéndice C.

### 5.3.1. Descripción de los experimentos

Así se efectuan cuatro experimentos principales:

- **Experimento 1:** Pruebas con distintos tamaños de ventana en Auto-test (apéndice C.1)
  - ✧ Donde se reconocen los tramos de archivos de Auto-test con los HMM estimados en los tres primeros grupos de la subsección 5.2.4.
- **Experimento 2:** Pruebas con distintos tamaños de ventana en test (apéndice C.2)
  - ✧ Donde se reconocen los tramos de archivos de test con los HMM estimados en los tres primeros grupos de la subsección 5.2.4.
- **Experimento 3:** Elección del tamaño óptimo de ventana única en test y Auto-test (apéndice C.3)
  - ✧ Donde se reconocen los tramos de archivos de Auto-test y de test con los HMM estimados en el primer y último grupo de la subsección 5.2.4.
- **Experimento 4:** Pruebas con vídeos de Internet (apéndice C.4)

- ✧ Donde se reconocen los audios de los vídeos con los HMM estimados en los dos primeros grupos de la subsección 5.2.4.

### 5.3.2. Conclusiones de los experimentos 1 y 2

Para entender estos resultados, se toman como referencia lo obtenido con la ventana de 25 ms entrenada con todos los archivos, y se comparan con los resultados más representativos de los otros grupos, aquellos en los que el tamaño de la ventana entrenada coincide con la ventana asignada al archivo que se van a reconocer.

**25ms** Entrenamiento para la ventana de 25 ms con todos los archivos

**Completo** Entrenamiento para múltiples ventanas con todos los archivos

**Específico** Entrenamiento para múltiples ventanas con archivos específicos de esa ventana

**Reducido** Entrenamiento para múltiples ventanas con la misma cantidad de archivos que en el específico, pero siguiendo la misma distribución de *pitch* de la base de datos original

Cabe notar que los resultados se presentan con la tasa de aciertos de palabras (%WORD), que se corresponde con el número de palabras detectadas correctamente dividido entre el número de palabras totales de las transcripciones. El intervalo de confianza se ha calculado para el 95 %.

Tabla 5.5: Resultados de los experimentos de Auto-test, reconociendo toda la base de datos. Los resultados completos están en el apéndice C.1. El número de archivos y los rangos de frecuencias se ven en las tablas 5.1 y 5.2. Para los tres últimos grupos, se representa sólo la ventana adaptada a su tramo.

| Ventana análisis (ms) | 37,5 | 25 | 18,75 | 15 | 12,5 |
|---|---|---|---|---|---|
| Rango *pitch* (Hz) | 0 - 100 | 100 - 140 | 140 - 180 | 180 - 220 | 220 - máx. |
| 25 ms (ref.) (%) | 94,39 | 92,58 | **90,21** | **91,20** | **89,73** |
| Completo (%) | 92,49 | 92,58 | 89,62 | 89,92 | 86,55 |
| Específico (%) | **94,93** | **93,39** | 90,18 | 90,94 | 88,89 |
| Reducido (%) | 84,52 | 91,89 | 89,27 | 89,19 | 85,05 |

Tabla 5.6: Resultados de los experimentos de test, reconociendo los 250 archivos de cada tramo. Los resultados completos están en el apéndice C.2. Los rangos de frecuencias se ven en la tabla 5.1. Para los tres últimos grupos, se representa sólo la ventana adaptada a su tramo.

| Ventana análisis (ms) | 37,5 | 25 | 18,75 | 15 | 12,5 |
|---|---|---|---|---|---|
| Rango *pitch* (Hz) | 0 - 100 | 100 - 140 | 140 - 180 | 180 - 220 | 220 - máx. |
| 25 ms (ref.) ( %) | **85,58** ± 2,53 | 96,38 ± 0,96 | **87,89** ± 2,01 | **95,32** ± 1,45 | **63,73** ± 3,21 |
| Completo ( %) | 82,45 ± 2,70 | 96,38 ± 0,96 | 87,30 ± 2,07 | 94,68 ± 1,71 | 59,68 ± 3,20 |
| Específico ( %) | 84,53 ± 2,60 | **97,00** ± 0,90 | 87,26 ± 2,04 | 94,44 ± 1,82 | 62,89 ± 3,30 |
| Reducido ( %) | 80,99 ± 2,72 | 96,12 ± 1,11 | 86,35 ± 2,09 | 94,25 ± 1,64 | 57,16 ± 3,71 |

Las pruebas realizadas en Auto-test, al ser el caso ideal, permiten saber dónde se encuentra el límite del sistema, mientras qué las pruebas de test indican en que punto se halla realmente el sistema.

Así, de los dos primeros experimentos se desprenden múltiples conclusiones:

- En cuanto a los mejores resultados:
  - ✧ **En Auto-test (caso ideal):** Los mejores resultados se obtienen con la ventana de análisis de 25 ms con entrenamiento selectivo para los tramos de 0 a 100 Hz y de 100 a 140 Hz. El resto de óptimos se consiguen con la ventana de 25 ms con entrenamiento completo. Se observan en la tabla 5.5, donde las filas indican la tasa de aciertos para cada grupo de entrenamiento, y las columnas la ventana de análisis óptima de los archivos a reconocer.
  - ✧ **En test (caso real):** La ventana de análisis de 25 ms siempre es la mejor salvo para el caso de 100 a 140 Hz donde gana la específica de 25 ms. Se observan en la tabla 5.6, donde las filas indican la tasa de aciertos para cada grupo de entrenamiento, y las columnas la ventana de análisis óptima de los archivos a reconocer.
- También se puede observar que la ventana de análisis de 25 ms con entrenamiento completo siempre es la mejor de su grupo de entrenamiento (con todos los archivos) lo que da una pista de que, para una única ventana, la de 25 ms puede ser la mejor.
- Pero la conclusión más importante se extrae al comparar las dos últimas columnas del entrenamiento selectivo con el entrenamiento de distribución uniforme.
  - ✧ En este caso, ambos modelos se han entrenado con el mismo número de archivos, pero en el selectivo estando adaptados a la ventana que se quiere

entrenar y reconocer, y en el segundo siguiendo la misma distribución que sigue la base de datos original.

- ✧ Lo que se puede observar es que a mismo número de archivos de entrenamiento, se observan notables mejoras al utilizar el entrenamiento selectivo:
  - ❏ Auto-test: De entre 10.41 y 0.91 puntos porcentuales. Por orden ascendente de tamaño de ventana las mejoras son de: 3.84, 1.75, 0.91, 1.50 y 10.41 puntos porcentuales.
  - ❏ Test: De entre 5.73 y 0.19 puntos porcentuales. Por orden ascendente del tamaño de ventana las mejoras son de: 5.73, 0.19, 0.91, 0.88 y 3.54 puntos porcentuales.
- ✧ Se observa que la mejora es más destacada en los tramos más extremos, probablemente por corresponderse con los tramos con menor cantidad de archivos de entrenamiento.
- ✧ Estos resultados indican que de poder tener los mismos datos de entrenamiento por cada tramo de *pitch* para efectuar un entrenamiento selectivo, se obtendría una mejora en la tasa de reconocimiento.

### 5.3.3. Conclusiones del experimento 3

Para este experimento se toma como referencia la tasa global obtenida de la suma de todas las palabras correctas dividida del total de palabras para cada ventana de análisis. En el caso de Auto-test se utilizan los 244.000 archivos y para el de test los 1.250 archivos. Los resultados completos por cada tramo de *pitch* se encuentran en la tabla del apéndice 89.

Tabla 5.7: Resultados de test y Auto-test para distintos tamaños de ventana. El resultado es la tasa global de aciertos sobre todas las palabras del análisis.

| *Pitch* central (Hz) | 240 | 200 | 160 | 150 | 133,3 | 120 |
|---|---|---|---|---|---|---|
| Tamaño ventana (ms) | 12,5 | 15 | 18,75 | 20 | 22,5 | 25 |
| Auto-test ( %) | 88,89 | 90,19 | 91,12 | 91,28 | 91,52 | **91,56** |
| Test ( %) | 82,75 | 83,81 | 85,46 | 85,61 | 85,66 | **85,78** |
| *Pitch* central (Hz) | | 109,1 | 100 | 92,3 | 85,7 | 80 |
| Tamaño ventana (ms) | | 27,5 | 30 | 32,5 | 35 | 37,5 |
| Auto-test ( %) | | 91,44 | 91,28 | 91,01 | 90,45 | 89,54 |
| Test ( %) | | 85,75 | 85,29 | 84,84 | 84,00 | 82,41 |

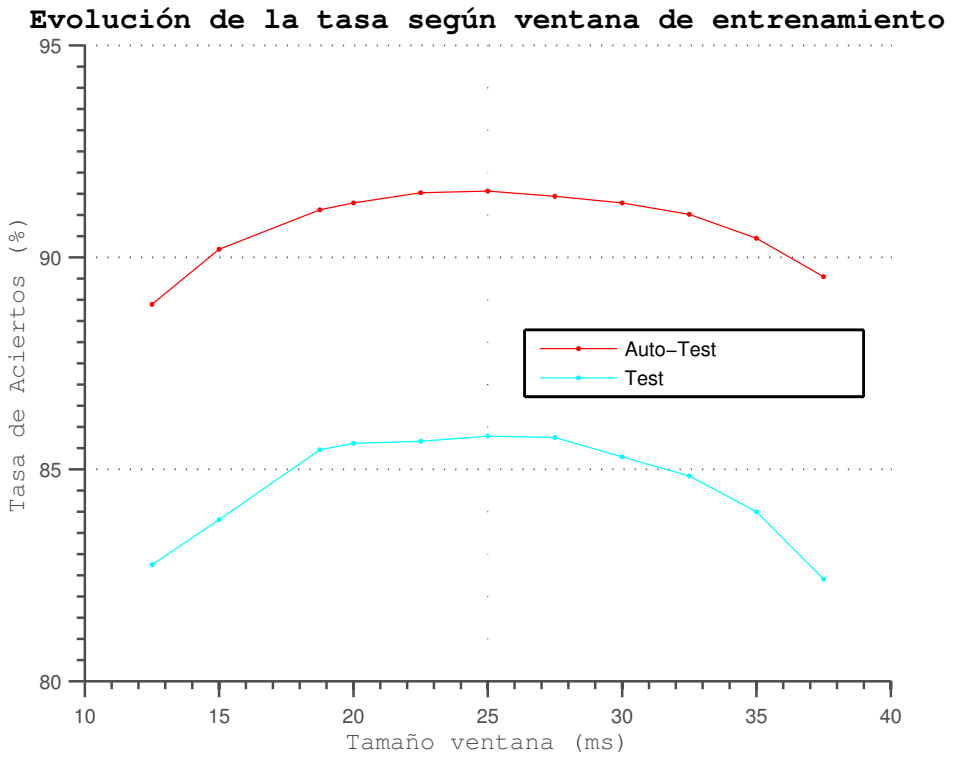


Figura 5.2: Gráfica de los resultados de entrenamiento con múltiples tamaños de ventana utilizando todos los archivos de entrenamiento. Para Auto-test y test.

Tras comparar los resultados obtenidos del reconocimiento con los 11 tamaños de ventana citados en la tabla, se puede observar, tanto en las tablas como en la gráfica que las condensa, la afirmación de que la ventana de entrenamiento y reconocimiento óptima es la de 25 ms. Es seguida, bastante de cerca, por las ventanas contiguas de 22.5 y 27.5 ms.

### 5.3.4. Conclusiones del experimento 4

En este último experimento se pretende probar cómo funciona cada tamaño de ventana en un entorno menos controlado. Se seleccionaron una serie de vídeos de Internet que posteriormente se transcribieron para poder extraer resultados al comparar con el reconocimiento.

Las diferencias con la base de datos radican principalmente en: habla espontánea en lugar de leída, la distancia al micrófono y la música de fondo.

Tabla 5.8: Resultados de las pruebas con los vídeos de Internet con modelos acústicos entrenados con todos los archivos de la base de datos. Tasas en tanto por ciento de aciertos de palabras.

| Ventana de análisis (ms) | 12,5 | 15 | 18,75 | 25 | 37,5 |
|---|---|---|---|---|---|
| Tasa acierto economía ( %) | 48,69 | 51,18 | 51,89 | **52,65** | 43,32 |
| Tasa acierto deportes ( %) | 47,49 | 49,33 | 50,21 | 50,88 | 42,06 |
| Tasa acierto moda ( %) | 24,11 | 25,22 | 26,67 | **28,80** | 26,36 |

Tabla 5.9: Resultados de las pruebas con los vídeos de Internet con modelos acústicos entrenados con archivos específicos. Tasas en tanto por ciento de aciertos de palabras.

| Ventana de análisis (ms) | 12,5 | 15 | 18,75 | 25 | 37,5 |
|---|---|---|---|---|---|
| Tasa acierto economía ( %) | 40,01 | 48,75 | 50,74 | 45,92 | 31,09 |
| Tasa acierto deportes ( %) | 31,05 | 43,01 | 50,29 | **51,75** | 31,09 |
| Tasa acierto moda ( %) | 24,97 | 26,90 | 26,51 | 19,27 | 14,54 |

Tras observar los resultados, se comprueba que la ventana de 25 ms entrenada con todos los archivos sigue siendo en promedio la más adecuada para un análisis de un archivo de voz genérico.

Para los vídeos de economía se ha obtenido un punto más de tasa con la ventana específica de 25 ms. Esto puede ser debido a que en estos vídeos todos los hablantes son varones, por lo que su *pitch* se ajusta más a la gama de 100-140 Hz, que está mejor modelada. En los vídeos de deportes hay tanto hombres como mujeres, mientras que en los de moda sólo hay mujeres (Ver tabla 5.3 ).

# Capítulo 6

# Aplicación del Sistema

Una vez implementado, ajustado, evaluado y optimizado el sistema de reconocimiento de habla natural, foco de este proyecto, se ha introducido en una solución comercial de la empresa.

La solución se encarga de indexar contenidos audiovisuales de Internet con el fin de realizar búsquedas en ellos. El sistema de reconocimiento transcribe en un proceso de segundo plano el audio de los contenidos, generando las marcas de tiempo de cada palabra. Esto otorgará una mayor robustez y fiabilidad a la indexación, obteniendo mejores resultados.

A continuación, se muestra una demostración del funcionamiento de la aplicación y el papel que juega la transcripción de los contenidos audiovisuales en ella.

## 6.1. Demostración

En la demostración se utilizan vídeos correspondientes a diversas noticias extraídas del telediario. Una vez reconocidos e indexados dichos vídeos, se permite la búsqueda en diversas partes de la noticia: en el título y descripción del vídeo, y en el propio audio con su marca de tiempo asociada.

La interfaz muestra un diseño práctico e intuitivo para el usuario (Figuras 6.1 y 6.2). En primer lugar, en la parte superior se dispone de un campo de texto para introducir la cadena completa a buscar, una búsqueda aditiva o disyuntiva, etc. En la parte central, se presenta el visualizador de videos y una barra adicional que marca en color azul cada una de las apariciones de la búsqueda en el instante de tiempo correspondiente. En la parte inferior del vídeo, se mostrará información relevante de la noticia (título, fecha, temática y breve descripción). Por último, a la derecha se mostrará a modo de *scroll*, los resultados de la búsqueda ordenados por orden de

fiabilidad de aparición de esa búsqueda en el contenido del vídeo.

**Ejemplo**

Supongamos como ejemplo que introducimos en la barra de búsqueda la cadena "Presidente del Gobierno" (Figura 6.1), en ese preciso instante aparecerá en primera plana el vídeo que dispone de mayor probabilidad de que aparezca aquello que se ha solicitado.


Figura 6.1: Imagen de la aplicación. Búsqueda concreta.


Figura 6.2: Imagen de la aplicación. Diferentes apariciones en el vídeo.

Puede comprobarse cómo a la derecha aparecerá marcado el vídeo que se está reproduciendo, y en la parte inferior derecha del visualizador se informa de que la búsqueda aparece tanto en el audio como en la descripción de la noticia, apareciendo en ella, la búsqueda marcada en negrita.

Además, como se muestra en la Figura 6.2, la segunda marca de tiempo pasa a un color azul más oscuro, respecto a la Figura 6.1, cambiando el locutor que menciona la cadena "Presidente del Gobierno".

# Capítulo 7

# Conclusiones y Trabajo Futuro

## 7.1. Conclusiones

Para conseguir el objetivo de realizar un sistema de reconocimiento de habla natural orientado a la transcripción de contenidos audiovisuales de Internet, se ha realizado un estudio sobre cómo mejorar el modelo acústico. La realización de este proyecto ha transcurrido por diversas fases.

Al inicio, se realizó una búsqueda de información para situarse en el estado del arte actual, donde se analizaron las características de los sistemas de reconocimiento de voz actuales y en especial de los que aplican modelos ocultos de Markov.

Una vez realizada la documentación pertinente, se comenzó el desarrollo de un sistema de entrenamiento de referencia, que se probó midiendo la calidad de los resultados obtenidos de las pruebas realizadas con el reconocedor.

Por último, se realizaron múltiples optimizaciones sobre todo el sistema. Optimizando el número de gaussianas e incrementando el número de archivos de entrenamiento se logró alcanzar una tasa de acierto adecuada en las pruebas. Después, se realizaron las pruebas con diferentes ventanas de análisis con el fin de mejorar todavía más esta tasa, confirmando que la más adecuada es la de 25 ms.

De esta forma se ha conseguido alcanzar el principal objetivo de este proyecto, que era crear un sistema de reconocimiento de habla continua de gran vocabulario.

Centrándose en los experimentos:

- Se puede observar que hay algunos tamaños de ventana, que entrenados con archivos específicos de su rango de frecuencias, permiten mejorar la calidad del reconocimiento.

- Si se comparan las pruebas de entrenamiento selectivo con las pruebas de entrenamiento reducido (tabla 5.6), se puede deducir que para mismo número de

archivos de entrenamiento la mejora introducida por utilizar una ventana adaptada puede llegar a ser bastante grande. Sin embargo, por lo observado hasta ahora, la ganancia es menor que la introducida por entrenar una única ventana con todos los archivos disponibles. A pesar de ello, se infiere que consiguiendo archivos de entrenamiento de las frecuencias específicas, se puede mejorar el sistema.

- En [15] se ha probado este mismo sistema base en comparación con las soluciones comerciales de las empresas líderes del mercado: Google y Apple.
  Para ellas se han utilizado diccionarios de 138.000 palabras y modelos de lenguaje entrenados con 100 millones de palabras.
  Para tratar fácilmente los ficheros de audio en todas las pruebas se han dividido con una herramienta de la empresa Sigma Technologies.
  Se puede observar que el sistema está a la altura del de Google[1] y sobrepasa en varios puntos al utilizado por Apple[2] en español de España.
  Cabe destacar que tanto Apple como Google cometen muy pocas inserciones en sus textos, tomando la política de descartar la frase si no la ha entendido bien o tiene ruido.

Tabla 7.1: Comparativa de tasas de acierto (%WORD) en el reconocimiento de los tres tópicos con los sistemas de Google y Apple.

| **Tópico** | ΣT | Google | |
|---|---|---|---|
| Economía | 39,23 | 39,67 | 33,94 |
| Deporte | 45,29 | 45,25 | 32,80 |
| Moda | 20,24 | 11,10 | 18,19 |

- Además, se ha logrado probar el correcto funcionamiento del sistema de reconocimiento al incluirlo en una solución comercial que aprovecha sus habilidades para mejorar la búsqueda dentro de contenidos audiovisuales.

[1] Web Speech API Demonstration http://www.google.com/intl/es/chrome/demos/speech.html
[2] Dictado mejorado de Apple en Mac OS X Yosemite. Se desactivaron los comandos de dictado para efectuar la prueba. http://support.apple.com/es-es/HT5449

## 7.2. Trabajo futuro

Una vez concluido este proyecto, y con el fin de seguir mejorando el sistema, aparecen distinas líneas de trabajo en las que se puede continuar investigando:

- Observando la curva obtenida para el reconocimiento con incremento paulatino del número de archivos de entrenamiento, se deduce que todavía se podría mejorar el sistema si se obtuvieran nuevos archivos de entrenamiento, especialmente si éstos son de habla natural y no de habla leída como la base de datos actual.
- Además, habría que centrarse en obtener más datos de baja frecuencia fundamental y alta frecuencia fundamental, y probar cómo mejoraría el sistema al añadir estos nuevos datos.
- Aunque se ha demostrado que el tamaño de ventana óptimo para una única ventana está en 25 ms, sería de gran utilidad realizar un ajuste más fino, entrenando más modelos entre los 22.5 ms y 27.5 ms para poder confirmar más fielmente que la ventana óptima se encuentra en esta posición.
- De cara a optimizar el reconocimiento de contenidos audiovisuales, sería de gran utilidad generar y aplicar modelos de lenguaje robustos y adaptados al tema a tratar. En esta rama ya se ha comenzado a investigar como podemos ver en el proyecto [15].
- Dado que los vídeos que se van a reconocer vienen directamente de Internet y no se puede garantizar su calidad, sería recomendable encontrar nuevos métodos de supresión de ruido para poder así mejorar la precisión del reconocimiento.
- Por último, vista la mejora de en torno al 30 % que suponen las DNN (Deep Neural Networks - Redes Neuronales Profundas) [6] frente a los HMM para la generación de los modelos acústicos en el reconocimiento de habla natural, sería de gran interés poder comparar un sistema hecho con las mismas bases de datos mediante DNNs frente a uno hecho con HMMs, y realizar una evaluación exhaustiva en precisión del reconocimiento y coste computacional.

# Bibliografía


[1] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, *et al.*, *The HTK book*, vol. 2. Entropic Cambridge Research Laboratory Cambridge, 1997.

[2] A. Lee, *The Julius book*. 2010.

[3] P. Lamere, P. Kwok, W. Walker, E. B. Gouvêa, R. Singh, B. Raj, and P. Wolf, "Design of the cmu sphinx-4 decoder.," in *INTERSPEECH*, Citeseer, 2003.

[4] G. Saon and J.-T. Chien, "Large-vocabulary continuous speech recognition systems: A look at some recent advances," *Signal Processing Magazine, IEEE*, vol. 29, pp. 18–33, Nov 2012.

[5] L. R. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*, vol. 14. PTR Prentice Hall Englewood Cliffs, 1993.

[6] G. Hinton, L. Deng, D. Yu, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. S. G. Dahl, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, November 2012.

[7] M. Adda-Decker and L. Lamel, "The use of lexica in automatic speech recognition,"

[8] G. A. Fink, "n-gram models," in *Markov Models for Pattern Recognition*, pp. 107–127, Springer, 2014.

[9] J. González-Rodríguez, D. T. Toledano, and J. Ortega-García, "Voice biometrics," in *Handbook of Biometrics*, pp. 151–170, Springer, 2008.

[10] X. D. Huang, Y. Ariki, and M. A. Jack, *Hidden Markov models for speech recognition*, vol. 2004. Edinburgh university press Edinburgh, 1990.

[11] S. Austin, R. Schwartz, and P. Placeway, "The forward-backward search algorithm," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pp. 697–700, IEEE, 1991.

[12] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *Information Theory, IEEE Transactions on*, vol. 13, no. 2, pp. 260–269, 1967.

[13] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The annals of mathematical statistics*, pp. 164–171, 1970.

[14] C. García and D. Tapias, "La frecuencia fundamental de la voz y sus efectos en reconocimiento de habla continua," *División de Tecnología del Habla Telefónica Investigación y Desarrollo, Revista de Procesamiento de Lenguaje Natural*, vol. 26, pp. 163–168, 2000.

[15] J. M. Perero Codosero, "Desarrollo de un sistema de reconocimiento de habla natural para transcribir contenidos de audio en internet," Master's thesis, EPS-UAM, 2015.

# Apéndice A

# Experimentos de elección de gaussianas

En este apéndice se muestran las tablas completas que han llevado a la elección del número óptimo de gaussianas:

Tabla A.1: Tasas de acierto de palabra ( %WORD) en función del número de gaussianas empleadas durante la fase de entrenamiento del modelo acústico. Archivos de la lista Auto-Test con SNR 10-20 dB.

| **Nº gaussianas** | **% WORD** | **% ACC** | **% SENT** | **Nº gaussianas** | **% WORD** | **% ACC** | **% SENT** |
|---|---|---|---|---|---|---|---|
| 1 | 85,21 | 78,96 | 78,08 | 33 | 96,56 | 91,46 | 86,49 |
| 2 | 88,94 | 83,40 | 80,64 | 34 | 96,59 | 91,10 | 86,47 |
| 3 | 91,04 | 85,49 | 82,21 | 35 | 96,77 | 91,73 | 86,77 |
| 4 | 92,25 | 87,88 | 84,14 | 36 | 96,74 | 91,88 | 86,87 |
| 5 | 92,96 | 89,19 | 86,04 | 37 | 96,74 | 91,70 | 86,57 |
| 6 | 93,85 | 90,40 | 87,36 | 38 | 96,67 | 91,56 | 86,47 |
| 7 | 94,44 | 90,84 | 87,54 | 39 | 96,85 | 91,60 | 86,37 |
| 8 | 94,13 | 89,89 | 86,95 | 40 | 96,81 | 91,17 | 85,57 |
| 9 | 94,34 | 89,79 | 87,35 | 41 | 96,49 | 90,53 | 84,75 |
| 10 | 94,09 | 89,43 | 85,73 | 42 | 96,45 | 89,93 | 83,45 |
| 11 | 94,30 | 89,14 | 85,92 | 43 | 96,74 | 90,49 | 83,47 |
| 12 | 94,39 | 88,83 | 86,07 | 44 | 96,67 | 90,00 | 82,87 |
| 13 | 94,94 | 89,41 | 86,19 | 45 | 96,81 | 89,68 | 82,67 |
| 14 | 95,40 | 89,73 | 86,30 | 46 | 96,80 | 89,68 | 82,29 |
| 15 | 95,32 | 89,99 | 86,36 | 47 | 96,79 | 89,74 | 82,07 |
| 16 | 95,74 | 90,28 | 86,26 | 48 | 96,79 | 89,59 | 81,97 |
| 17 | 95,62 | 90,21 | 85,73 | 49 | 96,86 | 89,30 | 80,95 |
| 18 | 95,74 | 91,02 | 87,06 | 50 | 96,86 | 89,34 | 80,75 |
| 19 | 96,02 | 91,62 | 87,56 | 51 | 96,93 | 89,68 | 81,23 |
| 20 | 95,84 | 91,29 | 86,95 | 52 | 96,71 | 89,71 | 81,41 |
| 21 | 95,84 | 91,29 | 87,35 | 53 | 96,72 | 89,40 | 81,33 |
| 22 | 96,12 | 90,95 | 86,72 | 54 | 96,79 | 89,65 | 80,93 |
| 23 | 96,12 | 90,92 | 86,53 | 55 | 96,90 | 89,76 | 81,43 |
| 24 | 95,91 | 90,41 | 86,56 | 56 | 96,93 | 89,79 | 81,31 |
| 25 | 95,95 | 90,61 | 86,85 | 57 | 97,18 | 90,15 | 81,52 |
| 26 | 96,30 | 90,61 | 86,35 | 58 | 97,21 | 90,28 | 81,70 |
| 27 | 96,44 | 91,15 | 85,94 | 59 | 97,18 | 90,00 | 81,29 |
| 28 | 96,23 | 90,85 | 86,02 | 60 | 97,18 | 89,86 | 81,41 |
| 29 | 96,27 | 90,91 | 85,74 | 61 | 97,36 | 89,79 | 81,11 |
| 30 | 96,28 | 90,75 | 85,69 | 62 | 97,32 | 89,87 | 81,13 |
| 31 | 96,46 | 91,25 | 87,09 | 63 | 97,40 | 90,08 | 80,93 |
| 32 | 96,39 | 91,14 | 86,49 | 64 | 97,40 | 90,08 | 81,43 |

Tabla A.2: Tasas de acierto de palabra ( %WORD) en función del número de gaussianas empleadas durante la fase de entrenamiento del modelo acústico. Archivos de la lista Auto-Test con SNR 20-30 dB.

| **Nº gaussianas** | **% WORD** | **% ACC** | **% SENT** | **Nº gaussianas** | **% WORD** | **% ACC** | **% SENT** |
|---|---|---|---|---|---|---|---|
| 1 | 87,05 | 84,45 | 60,12 | 33 | 96,77 | 94,77 | 78,70 |
| 2 | 90,97 | 88,13 | 66,47 | 34 | 96,75 | 94,72 | 78,90 |
| 3 | 92,06 | 89,11 | 66,53 | 35 | 96,89 | 95,00 | 79,40 |
| 4 | 93,30 | 90,36 | 68,94 | 36 | 96,92 | 95,03 | 79,50 |
| 5 | 93,55 | 90,79 | 70,42 | 37 | 96,98 | 94,94 | 79,10 |
| 6 | 94,21 | 91,36 | 72,21 | 38 | 97,00 | 95,02 | 79,50 |
| 7 | 94,77 | 91,94 | 73,54 | 39 | 96,97 | 95,02 | 79,80 |
| 8 | 94,91 | 92,64 | 74,55 | 40 | 96,97 | 95,05 | 80,00 |
| 9 | 94,94 | 92,57 | 74,60 | 41 | 96,95 | 94,99 | 80,28 |
| 10 | 95,00 | 92,80 | 75,23 | 42 | 97,14 | 95,27 | 80,28 |
| 11 | 95,34 | 92,78 | 75,43 | 43 | 97,03 | 95,17 | 79,60 |
| 12 | 95,51 | 92,99 | 75,68 | 44 | 97,11 | 95,22 | 79,60 |
| 13 | 95,73 | 93,38 | 76,18 | 45 | 97,28 | 95,39 | 80,00 |
| 14 | 95,87 | 93,50 | 76,68 | 46 | 97,31 | 95,32 | 79,98 |
| 15 | 95,99 | 93,91 | 77,48 | 47 | 97,42 | 95,34 | 79,68 |
| 16 | 96,18 | 94,04 | 77,78 | 48 | 97,48 | 95,57 | 80,08 |
| 17 | 95,82 | 93,68 | 77,48 | 49 | 97,50 | 95,56 | 79,98 |
| 18 | 96,06 | 94,17 | 77,80 | 50 | 97,45 | 95,52 | 79,98 |
| 19 | 96,03 | 94,23 | 78,40 | 51 | 97,43 | 95,51 | 80,38 |
| 20 | 96,12 | 94,20 | 78,20 | 52 | 97,32 | 95,38 | 80,28 |
| 21 | 96,32 | 94,28 | 77,88 | 53 | 97,42 | 95,43 | 80,48 |
| 22 | 96,55 | 94,50 | 78,00 | 54 | 97,40 | 95,43 | 80,28 |
| 23 | 96,46 | 94,52 | 78,58 | 55 | 97,34 | 95,40 | 80,18 |
| 24 | 96,47 | 94,41 | 78,50 | 56 | 97,37 | 95,37 | 80,58 |
| 25 | 96,50 | 94,47 | 78,30 | 57 | 97,29 | 95,13 | 79,88 |
| 26 | 96,61 | 94,57 | 78,48 | 58 | 97,28 | 95,18 | 79,68 |
| 27 | 96,51 | 94,50 | 78,58 | 59 | 97,23 | 95,21 | 79,68 |
| 28 | 96,58 | 94,59 | 78,70 | 60 | 97,20 | 95,09 | 79,28 |
| 29 | 96,77 | 94,72 | 79,40 | 61 | 97,17 | 95,15 | 79,38 |
| 30 | 96,81 | 94,84 | 79,10 | 62 | 97,25 | 95,11 | 79,50 |
| 31 | 96,72 | 94,70 | 78,60 | 63 | 97,39 | 95,45 | 80,08 |
| 32 | 96,84 | 94,75 | 78,90 | 64 | 97,43 | 95,52 | 80,08 |

Tabla A.3: Tasas de acierto de palabra ( %WORD) en función del número de gaussianas empleadas durante la fase de entrenamiento del modelo acústico. Archivos de la lista Auto-Test con SNR 30-40 dB.

| **Nº gaussianas** | **% WORD** | **% ACC** | **% SENT** | **Nº gaussianas** | **% WORD** | **% ACC** | **% SENT** |
|---|---|---|---|---|---|---|---|
| 1 | 89,44 | 87,44 | 66,33 | 33 | 97,62 | 95,06 | 79,60 |
| 2 | 92,49 | 90,52 | 72,02 | 34 | 97,69 | 95,16 | 80,10 |
| 3 | 93,28 | 90,41 | 70,30 | 35 | 97,66 | 95,19 | 79,60 |
| 4 | 94,28 | 91,16 | 71,66 | 36 | 97,59 | 95,15 | 79,62 |
| 5 | 94,68 | 91,31 | 73,11 | 37 | 97,62 | 95,16 | 80,12 |
| 6 | 95,08 | 91,95 | 73,43 | 38 | 97,74 | 95,40 | 80,60 |
| 7 | 95,56 | 92,52 | 74,22 | 39 | 97,70 | 95,28 | 80,42 |
| 8 | 95,66 | 92,54 | 74,52 | 40 | 97,81 | 95,42 | 80,62 |
| 9 | 95,65 | 92,72 | 75,73 | 41 | 98,00 | 95,52 | 81,02 |
| 10 | 95,98 | 92,89 | 74,97 | 42 | 97,87 | 95,42 | 80,52 |
| 11 | 95,77 | 92,36 | 74,02 | 43 | 97,94 | 95,69 | 81,21 |
| 12 | 95,96 | 92,68 | 74,57 | 44 | 97,93 | 95,63 | 80,62 |
| 13 | 96,15 | 93,06 | 75,53 | 45 | 97,92 | 95,57 | 80,64 |
| 14 | 96,25 | 93,28 | 75,73 | 46 | 97,86 | 95,51 | 80,74 |
| 15 | 96,47 | 93,72 | 76,63 | 47 | 97,93 | 95,63 | 80,92 |
| 16 | 96,60 | 93,92 | 77,19 | 48 | 97,87 | 95,61 | 80,52 |
| 17 | 96,60 | 94,00 | 77,51 | 49 | 97,96 | 95,74 | 81,14 |
| 18 | 96,77 | 94,06 | 77,99 | 50 | 98,07 | 95,81 | 81,75 |
| 19 | 96,75 | 94,26 | 77,71 | 51 | 98,11 | 95,82 | 81,75 |
| 20 | 96,82 | 94,28 | 78,47 | 52 | 98,14 | 95,81 | 81,54 |
| 21 | 97,04 | 94,54 | 78,59 | 53 | 98,12 | 95,88 | 81,75 |
| 22 | 96,89 | 94,41 | 78,49 | 54 | 97,95 | 95,72 | 81,24 |
| 23 | 97,01 | 94,43 | 78,41 | 55 | 98,01 | 95,90 | 81,75 |
| 24 | 97,05 | 94,60 | 78,69 | 56 | 98,05 | 95,94 | 81,85 |
| 25 | 97,11 | 94,74 | 79,10 | 57 | 98,04 | 95,86 | 81,86 |
| 26 | 97,20 | 94,79 | 79,02 | 58 | 98,11 | 95,98 | 82,26 |
| 27 | 97,23 | 94,68 | 79,32 | 59 | 98,10 | 95,88 | 81,96 |
| 28 | 97,15 | 94,58 | 78,69 | 60 | 98,10 | 95,95 | 82,36 |
| 29 | 97,18 | 94,63 | 78,69 | 61 | 98,08 | 95,88 | 82,06 |
| 30 | 97,27 | 94,67 | 78,69 | 62 | 98,05 | 95,89 | 82,16 |
| 31 | 97,49 | 94,85 | 79,50 | 63 | 98,07 | 95,91 | 82,06 |
| 32 | 97,43 | 94,90 | 79,30 | 64 | 98,08 | 95,89 | 82,26 |

Tabla A.4: Tasas de acierto de palabra ( %WORD) en función del número de gaussianas empleadas durante la fase de entrenamiento del modelo acústico. Archivos de la lista Auto-Test con SNR 40-99 dB.

| **Nº gaussianas** | **% WORD** | **% ACC** | **% SENT** | **Nº gaussianas** | **% WORD** | **% ACC** | **% SENT** |
|---|---|---|---|---|---|---|---|
| 1 | 88,78 | 85,69 | 78,08 | 33 | 96,72 | 95,69 | 86,49 |
| 2 | 91,59 | 89,44 | 80,64 | 34 | 96,67 | 95,47 | 86,47 |
| 3 | 92,31 | 90,39 | 82,21 | 35 | 96,62 | 95,55 | 86,77 |
| 4 | 93,47 | 91,76 | 84,14 | 36 | 96,82 | 95,69 | 86,87 |
| 5 | 93,98 | 92,36 | 86,04 | 37 | 96,99 | 95,87 | 86,57 |
| 6 | 94,52 | 93,15 | 87,36 | 38 | 96,96 | 95,85 | 86,47 |
| 7 | 94,96 | 93,64 | 87,54 | 39 | 96,86 | 95,72 | 86,37 |
| 8 | 95,11 | 93,66 | 86,95 | 40 | 96,84 | 95,77 | 85,57 |
| 9 | 95,27 | 93,82 | 87,35 | 41 | 96,95 | 95,83 | 84,75 |
| 10 | 95,26 | 93,79 | 85,73 | 42 | 97,05 | 95,92 | 83,45 |
| 11 | 95,52 | 94,02 | 85,92 | 43 | 96,91 | 95,62 | 83,47 |
| 12 | 95,40 | 93,83 | 86,07 | 44 | 97,08 | 95,81 | 82,87 |
| 13 | 95,71 | 94,07 | 86,19 | 45 | 97,1 | 95,89 | 82,67 |
| 14 | 95,87 | 94,29 | 86,30 | 46 | 97,05 | 95,87 | 82,29 |
| 15 | 95,88 | 94,29 | 86,36 | 47 | 97,06 | 95,94 | 82,07 |
| 16 | 95,94 | 94,38 | 86,26 | 48 | 97,1 | 96,08 | 81,97 |
| 17 | 96,14 | 94,77 | 85,73 | 49 | 97,2 | 96,07 | 80,95 |
| 18 | 95,99 | 94,58 | 87,06 | 50 | 97,15 | 96,05 | 80,75 |
| 19 | 96,01 | 94,49 | 87,56 | 51 | 97,22 | 96,05 | 81,23 |
| 20 | 96,10 | 94,6 | 86,95 | 52 | 97,15 | 96,03 | 81,41 |
| 21 | 96,16 | 95,09 | 87,35 | 53 | 97,2 | 96,03 | 81,33 |
| 22 | 96,22 | 95,11 | 86,72 | 54 | 97,22 | 96,05 | 80,93 |
| 23 | 96,22 | 95,15 | 86,53 | 55 | 97,24 | 96,03 | 81,43 |
| 24 | 96,32 | 95,28 | 86,56 | 56 | 97,24 | 96,10 | 81,31 |
| 25 | 96,30 | 95,23 | 86,85 | 57 | 97,18 | 95,98 | 81,52 |
| 26 | 96,52 | 95,54 | 86,35 | 58 | 97,25 | 96,09 | 81,70 |
| 27 | 96,54 | 95,45 | 85,94 | 59 | 97,20 | 96,11 | 81,29 |
| 28 | 96,44 | 95,29 | 86,02 | 60 | 97,15 | 96,00 | 81,41 |
| 29 | 96,49 | 95,31 | 85,74 | 61 | 97,20 | 96,08 | 81,11 |
| 30 | 96,41 | 95,24 | 85,69 | 62 | 97,18 | 96,01 | 81,13 |
| 31 | 96,46 | 95,42 | 87,09 | 63 | 97,26 | 96,16 | 80,93 |
| 32 | 96,67 | 95,64 | 86,49 | 64 | 97,23 | 96,18 | 81,43 |

Tabla A.5: Tasas de acierto de palabra ( %WORD) en función del número de gaussianas empleadas durante la fase de entrenamiento del modelo acústico. Archivos de la lista Test con SNR 30-40 dB.

| **Nº gaussianas** | **% WORD** | **% ACC** | **% SENT** | **Nº gaussianas** | **% WORD** | **% ACC** | **% SENT** |
|---|---|---|---|---|---|---|---|
| 1 | 89,58 | 87,61 | 68,97 | 33 | 95,32 | 91,57 | 71,41 |
| 2 | 91,86 | 89,48 | 70,27 | 34 | 95,25 | 91,44 | 71,08 |
| 3 | 93,13 | 90,02 | 70,47 | 35 | 95,06 | 91,25 | 70,51 |
| 4 | 94,29 | 90,78 | 70,80 | 36 | 95,12 | 91,21 | 70,15 |
| 5 | 94,48 | 90,95 | 71,14 | 37 | 94,96 | 90,81 | 69,92 |
| 6 | 94,82 | 91,50 | 72,57 | 38 | 94,84 | 90,50 | 68,98 |
| 7 | 94,94 | 91,74 | 72,14 | 39 | 94,83 | 90,45 | 68,71 |
| 8 | 95,08 | 92,07 | 73,10 | 40 | 94,93 | 90,67 | 68,88 |
| 9 | 95,15 | 92,03 | 72,80 | 41 | 94,72 | 90,30 | 69,15 |
| 10 | 95,15 | 91,81 | 72,47 | 42 | 94,67 | 90,17 | 69,01 |
| 11 | 95,27 | 91,70 | 71,23 | 43 | 94,57 | 90,01 | 68,81 |
| 12 | 95,32 | 91,61 | 71,36 | 44 | 94,54 | 89,87 | 68,37 |
| 13 | 95,41 | 91,96 | 71,51 | 45 | 94,47 | 89,82 | 68,20 |
| 14 | 95,53 | 91,98 | 71,84 | 46 | 94,39 | 89,66 | 67,74 |
| 15 | 95,49 | 92,23 | 73,07 | 47 | 94,30 | 89,55 | 68,00 |
| 16 | 95,46 | 92,27 | 73,47 | 48 | 94,16 | 89,37 | 67,37 |
| 17 | 95,45 | 92,27 | 72,75 | 49 | 94,00 | 89,10 | 67,14 |
| 18 | 95,47 | 92,22 | 73,37 | 50 | 93,89 | 88,90 | 66,97 |
| 19 | 95,33 | 92,03 | 72,87 | 51 | 93,78 | 88,65 | 67,04 |
| 20 | 95,57 | 92,30 | 72,72 | 52 | 93,88 | 88,81 | 66,83 |
| 21 | 95,65 | 92,54 | 73,19 | 53 | 93,80 | 88,56 | 66,73 |
| 22 | 95,50 | 92,27 | 73,12 | 54 | 93,73 | 88,14 | 66,16 |
| 23 | 95,38 | 92,22 | 73,35 | 55 | 93,64 | 88,12 | 65,60 |
| 24 | 95,43 | 92,17 | 73,17 | 56 | 93,43 | 87,50 | 65,13 |
| 25 | 95,47 | 92,24 | 73,32 | 57 | 93,46 | 87,63 | 64,93 |
| 26 | 95,47 | 92,36 | 73,32 | 58 | 93,22 | 87,20 | 64,23 |
| 27 | 95,54 | 92,33 | 73,29 | 59 | 93,08 | 86,86 | 64,29 |
| 28 | 95,45 | 92,13 | 72,89 | 60 | 92,91 | 86,75 | 63,59 |
| 29 | 95,36 | 91,81 | 72,14 | 61 | 92,79 | 86,63 | 63,49 |
| 30 | 95,30 | 91,82 | 72,19 | 62 | 92,58 | 86,35 | 63,05 |
| 31 | 95,31 | 91,75 | 72,09 | 63 | 92,50 | 86,16 | 63,25 |
| 32 | 95,11 | 91,33 | 71,26 | 64 | 92,44 | 86,03 | 62,95 |

Tabla A.6: Tasas de acierto de palabra ( %WORD) en función del número de gaussianas empleadas durante la fase de entrenamiento del modelo acústico. Archivos de la lista Test con archivos ruidosos.

| **Nº gaussianas** | **% WORD** | **% ACC** | **% SENT** | **Nº gaussianas** | **% WORD** | **% ACC** | **% SENT** |
|---|---|---|---|---|---|---|---|
| 1 | 81,00 | 72,14 | 61,43 | 33 | 87,68 | 77,76 | 69,88 |
| 2 | 85,44 | 79,37 | 70,34 | 34 | 87,83 | 77,55 | 69,18 |
| 3 | 87,36 | 80,98 | 71,94 | 35 | 87,83 | 77,41 | 68,98 |
| 4 | 88,71 | 82,82 | 75,25 | 36 | 87,73 | 77,31 | 69,38 |
| 5 | 89,36 | 83,93 | 75,98 | 37 | 87,83 | 76,83 | 69,28 |
| 6 | 90,20 | 85,15 | 77,68 | 38 | 87,6 | 76,38 | 69,18 |
| 7 | 90,14 | 84,63 | 76,95 | 39 | 87,44 | 76,51 | 68,47 |
| 8 | 90,45 | 84,44 | 76,95 | 40 | 87,04 | 75,80 | 67,77 |
| 9 | 90,72 | 84,41 | 76,75 | 41 | 87,03 | 74,90 | 67,30 |
| 10 | 91,06 | 85,15 | 76,75 | 42 | 86,84 | 74,93 | 67,50 |
| 11 | 90,74 | 84,39 | 76,15 | 43 | 86,78 | 75,07 | 67,23 |
| 12 | 90,88 | 84,78 | 76,15 | 44 | 86,04 | 73,83 | 66,23 |
| 13 | 91,30 | 84,97 | 77,15 | 45 | 85,86 | 73,96 | 66,03 |
| 14 | 91,01 | 84,07 | 76,05 | 46 | 85,49 | 73,27 | 65,63 |
| 15 | 90,88 | 83,89 | 76,75 | 47 | 85,44 | 73,03 | 65,13 |
| 16 | 90,88 | 83,47 | 76,35 | 48 | 85,28 | 72,48 | 64,96 |
| 17 | 90,85 | 83,47 | 75,75 | 49 | 85,28 | 71,91 | 63,90 |
| 18 | 90,98 | 83,25 | 75,65 | 50 | 84,70 | 71,27 | 63,70 |
| 19 | 90,90 | 83,65 | 75,25 | 51 | 84,91 | 70,83 | 62,90 |
| 20 | 90,36 | 83,12 | 74,82 | 52 | 84,57 | 69,93 | 62,60 |
| 21 | 90,12 | 82,51 | 74,22 | 53 | 84,31 | 69,59 | 61,90 |
| 22 | 89,88 | 81,93 | 73,52 | 54 | 84,12 | 69,27 | 62,40 |
| 23 | 89,95 | 81,32 | 73,15 | 55 | 84,52 | 69,64 | 62,40 |
| 24 | 89,83 | 81,29 | 72,72 | 56 | 83,99 | 68,93 | 61,30 |
| 25 | 89,51 | 80,37 | 72,02 | 57 | 83,36 | 68,19 | 61,00 |
| 26 | 88,94 | 79,20 | 71,41 | 58 | 82,99 | 67,06 | 59,40 |
| 27 | 88,81 | 79,11 | 71,39 | 59 | 82,46 | 66,38 | 58,70 |
| 28 | 89,05 | 79,95 | 70,88 | 60 | 82,93 | 67,08 | 59,26 |
| 29 | 88,34 | 78,68 | 70,68 | 61 | 82,77 | 65,81 | 58,76 |
| 30 | 88,48 | 78,84 | 70,21 | 62 | 82,80 | 66,63 | 58,76 |
| 31 | 88,65 | 79,29 | 70,18 | 63 | 82,85 | 66,74 | 59,06 |
| 32 | 88,19 | 78,71 | 70,11 | 64 | 82,17 | 65,32 | 57,80 |

# Apéndice B

# Experimentos elección de número de ficheros de entrenamiento

En este apéndice se muestran las gráficas y tablas completas de la elección del número de archivos de entrenamiento:

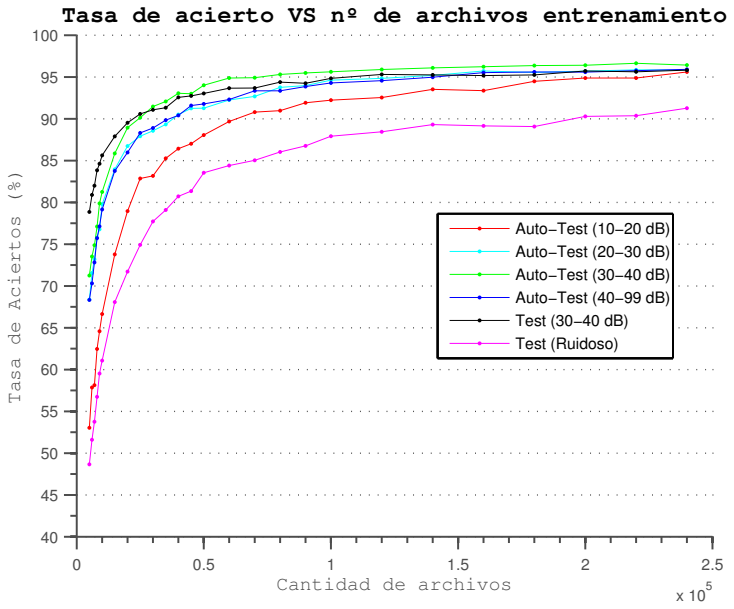


Figura B.1: Gráfica en la que se puede observar la evolución de la tasa de acierto (%WORD) en función del número de archivos con el que se ha entrenado el modelo acústico

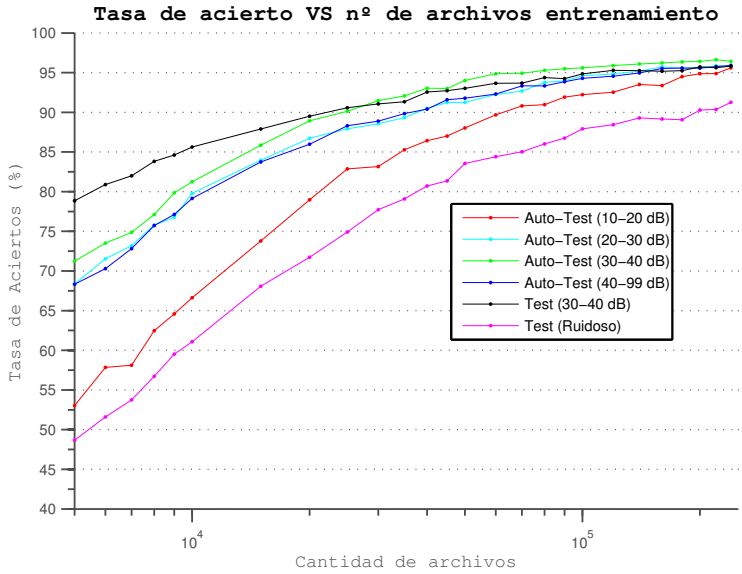


Figura B.2: Gráfica en la que se puede observar la evolución de la tasa de acierto (%WORD) en función del número de archivos con el que se ha entrenado el modelo acústico. Se muestra en escala semilogarítmica para poder apreciar mejor la tendencia de la gráfica.

Tabla B.1: Tasa de aciertos (%WORD) frente a número de archivos de entrenamiento por grupos de archivos de prueba

| **Nº Archivos** | **Auto 10-20** | **Auto 20-30** | **Auto 30-40** | **Auto 40-99** | **Test 30-40** | **Test Ruido** |
|---|---|---|---|---|---|---|
| 5.000 | 53,03 ± 2,66 | 68,36 ± 2,21 | 71,24 ± 2,18 | 68,33 ± 2,29 | 78,85 ± 1,94 | 48,66 ± 2,34 |
| 6.000 | 57,85 ± 2,71 | 71,53 ± 2,18 | 73,52 ± 2,19 | 70,30 ± 2,31 | 80,89 ± 1,84 | 51,59 ± 2,38 |
| 7.000 | 58,13 ± 2,71 | 73,19 ± 2,18 | 74,87 ± 2,16 | 72,82 ± 2,26 | 82,00 ± 1,87 | 53,75 ± 2,37 |
| 8.000 | 62,47 ± 2,65 | 75,78 ± 2,12 | 77,11 ± 2,13 | 75,72 ± 2,27 | 83,83 ± 1,80 | 56,73 ± 2,39 |
| 9.000 | 64,58 ± 2,67 | 76,77 ± 2,12 | 79,87 ± 2,04 | 77,13 ± 2,24 | 84,62 ± 1,73 | 59,51 ± 2,39 |
| 10.000 | 66,63 ± 2,62 | 79,78 ± 2,05 | 81,24 ± 2,02 | 79,15 ± 2,21 | 85,62 ± 1,72 | 61,07 ± 2,41 |
| 15.000 | 73,78 ± 2,50 | 83,95 ± 1,91 | 85,85 ± 1,77 | 83,75 ± 2,06 | 87,90 ± 1,59 | 68,07 ± 2,34 |
| 20.000 | 78,96 ± 2,37 | 86,73 ± 1,77 | 88,93 ± 1,62 | 85,98 ± 1,95 | 89,51 ± 1,55 | 71,72 ± 2,34 |
| 25.000 | 82,86 ± 2,24 | 87,92 ± 1,72 | 90,13 ± 1,46 | 88,30 ± 1,79 | 90,57 ± 1,47 | 74,91 ± 2,24 |
| 30.000 | 83,16 ± 2,20 | 88,56 ± 1,66 | 91,47 ± 1,37 | 88,89 ± 1,71 | 91,06 ± 1,39 | 77,72 ± 2,17 |
| 35.000 | 85,27 ± 2,09 | 89,33 ± 1,60 | 92,07 ± 1,35 | 89,83 ± 1,66 | 91,33 ± 1,31 | 79,09 ± 2,08 |
| 40.000 | 86,43 ± 1,93 | 90,50 ± 1,54 | 93,05 ± 1,24 | 90,41 ± 1,63 | 92,56 ± 1,28 | 80,72 ± 2,06 |
| 45.000 | 87,01 ± 1,97 | 91,24 ± 1,35 | 93,00 ± 1,26 | 91,58 ± 1,52 | 92,74 ± 1,22 | 81,35 ± 2,03 |
| 50.000 | 88,04 ± 1,88 | 91,26 ± 1,38 | 94,01 ± 1,15 | 91,78 ± 1,49 | 93,03 ± 1,24 | 83,54 ± 1,90 |
| 60.000 | 89,68 ± 1,81 | 92,24 ± 1,24 | 94,87 ± 1,02 | 92,30 ± 1,36 | 93,66 ± 1,23 | 84,41 ± 1,86 |
| 70.000 | 90,80 ± 1,68 | 92,67 ± 1,22 | 94,92 ± 0,98 | 93,34 ± 1,22 | 93,68 ± 1,10 | 85,02 ± 1,85 |
| 80.000 | 90,97 ± 1,66 | 93,76 ± 1,14 | 95,30 ± 1,02 | 93,34 ± 1,24 | 94,38 ± 1,12 | 86,02 ± 1,75 |
| 90.000 | 91,91 ± 1,56 | 93,98 ± 1,08 | 95,48 ± 0,89 | 93,86 ± 1,17 | 94,25 ± 1,10 | 86,75 ± 1,72 |
| 100.000 | 92,22 ± 1,51 | 94,62 ± 1,01 | 95,61 ± 0,85 | 94,29 ± 1,14 | 94,84 ± 0,97 | 87,91 ± 1,67 |
| 120.000 | 92,54 ± 1,50 | 94,83 ± 0,95 | 95,89 ± 0,79 | 94,56 ± 1,04 | 95,30 ± 0,97 | 88,45 ± 1,60 |
| 140.000 | 93,51 ± 1,36 | 95,19 ± 0,92 | 96,08 ± 0,80 | 94,97 ± 0,88 | 95,25 ± 0,96 | 89,30 ± 1,55 |
| 160.000 | 93,37 ± 1,29 | 95,72 ± 0,83 | 96,23 ± 0,82 | 95,52 ± 0,83 | 95,17 ± 0,97 | 89,15 ± 1,57 |
| 180.000 | 94,49 ± 1,24 | 95,54 ± 0,79 | 96,36 ± 0,72 | 95,59 ± 0,83 | 95,25 ± 0,99 | 89,07 ± 1,59 |
| 200.000 | 94,87 ± 1,21 | 95,69 ± 0,72 | 96,41 ± 0,72 | 95,57 ± 0,77 | 95,72 ± 0,94 | 90,28 ± 1,48 |
| 220.000 | 94,88 ± 1,19 | 95,85 ± 0,70 | 96,63 ± 0,73 | 95,75 ± 0,71 | 95,62 ± 0,86 | 90,36 ± 1,48 |
| 240.000 | 95,60 ± 1,08 | 95,82 ± 0,65 | 96,42 ± 0,71 | 95,92 ± 0,63 | 95,81 ± 0,83 | 91,26 ± 1,43 |

# Apéndice C

# Experimentos de optimización

En este apéndice se muestran las tablas completas de la optimización del sistema:

## C.1. Resultados experimento 1

Pruebas realizadas con distintas ventanas de análisis en Auto-test, 244.000 archivos de entrenamiento reconocidos:

Tabla C.1: Resultados pruebas Auto-test con entrenamiento de modelos acústicos realizado con todos los archivos (Completo). Tasa de aciertos de palabra ( %WORD)

| Ventana de análisis | 12,5 ms | 15 ms | 18,75 ms | 25 ms | 37,5 ms |
|---|---|---|---|---|---|
| 0 – 100 Hz | 91,66 | 93,07 | 93,89 | 94,39 | 92,49 |
| 100 – 140 Hz | 90,23 | 91,40 | 92,19 | 92,58 | 90,42 |
| 140 – 180 Hz | 86,95 | 88,53 | 89,62 | **90,21** | 88,24 |
| 180 – 220 Hz | 88,68 | 89,92 | 90,91 | **91,20** | 89,25 |
| 220 – máx. Hz | 86,55 | 87,97 | 89,05 | **89,73** | 87,97 |

Tabla C.2: Resultados pruebas Auto-test con entrenamiento de modelos acústicos realizado con archivos específicos de cada tramo (Específico). Tasa de aciertos de palabra ( %WORD)

| Ventana de análisis | 12,5 ms | 15 ms | 18,75 ms | 25 ms | 37,5 ms |
|---|---|---|---|---|---|
| Nº Archivos Ent. | 113.000 | 148.000 | 185.000 | 131.000 | 25.000 |
| 0 – 100 Hz | 73,24 | 86,66 | 91,82 | **95,16** | 94,93 |
| 100 – 140 Hz | 76,88 | 87,46 | 92,35 | **93,39** | 83,82 |
| 140 – 180 Hz | 81,49 | 88,98 | 90,18 | 86,90 | 66,47 |
| 180 – 220 Hz | 90,80 | 90,94 | 90,64 | 76,36 | 40,70 |
| 220 – máx. Hz | 88,89 | 88,60 | 84,57 | 67,52 | 31,87 |

Tabla C.3: Resultados pruebas Auto-test con entrenamiento de modelos acústicos realizado con el mismo número de archivos que el entrenamiento específico pero distribuidos de la misma forma que la base de datos completa (Reducido). Tasa de aciertos de palabra (%WORD)

| Ventana de análisis | 12,5 ms | 15 ms | 18,75 ms | 25 ms | 37,5 ms |
|---|---|---|---|---|---|
| Nº Archivos Ent. | 113.000 | 148.000 | 185.000 | 131.000 | 25.000 |
| 0 – 100 Hz | 90,08 | 92,36 | 93,60 | 93,47 | 84,52 |
| 100 – 140 Hz | 89,17 | 90,77 | 91,96 | 91,89 | 84,31 |
| 140 – 180 Hz | 85,55 | 87,70 | 89,27 | 89,38 | 80,97 |
| 180 – 220 Hz | 87,30 | 89,19 | 90,65 | 90,44 | 82,25 |
| 220 – máx. Hz | 85,05 | 87,05 | 88,76 | 88,62 | 80,02 |

## C.2. Resultados experimento 2

Pruebas con distintas ventanas en test, 250 archivos por tramo reconocidos:

Tabla C.4: Resultados pruebas test con entrenamiento de modelos acústicos realizado con todos los archivos (Completo). Tasa de aciertos de palabra (%WORD)

| Ventana de análisis | 12,5 ms | 15 ms | 18,75 ms | 25 ms | 37,5 ms |
|---|---|---|---|---|---|
| 0 – 100 Hz | 81,62 | 85,09 | 86,15 | 85,58 | 82,45 |
| 100 – 140 Hz | 93,50 | 96,46 | 96,96 | 96,38 | 92,81 |
| 140 – 180 Hz | 85,25 | 86,68 | 87,30 | **87,89** | 83,47 |
| 180 – 220 Hz | 93,71 | 94,68 | 94,65 | 95,32 | 93,71 |
| 220 – máx. Hz | 59,68 | 64,38 | 63,22 | **63,73** | 59,63 |

Tabla C.5: Resultados pruebas test con entrenamiento de modelos acústicos realizado con archivos específicos de cada tramo (Específico). Tasa de aciertos de palabra (%WORD)

| Ventana de análisis | 12,5 ms | 15 ms | 18,75 ms | 25 ms | 37,5 ms |
|---|---|---|---|---|---|
| Nº Archivos E. | 113.000 | 148.000 | 185.000 | 131.000 | 25.000 |
| 0 – 100 Hz | 70,08 | 81,63 | 85,58 | **86,94** | 84,53 |
| 100 – 140 Hz | 87,03 | 93,45 | 96,48 | **97,00** | 91,56 |
| 140 – 180 Hz | 81,32 | 85,00 | 87,26 | 84,77 | 66,68 |
| 180 – 220 Hz | **95,53** | 94,44 | 94,90 | 82,98 | 40,72 |
| 220 – máx. Hz | 62,89 | 62,33 | 59,20 | 30,15 | 15,87 |

Tabla C.6: Resultados pruebas test con entrenamiento de modelos acústicos realizado con el mismo número de archivos que el entrenamiento específico pero distribuidos de la misma forma que la base de datos completa (Reducido). Tasa de aciertos de palabra ( %WORD)

| Ventana de análisis | 12,5 ms | 15 ms | 18,75 ms | 25 ms | 37,5 ms |
|---|---|---|---|---|---|
| Nº Archivos E. | 113.000 | 148.000 | 185.000 | 131.000 | 25.000 |
| 0 – 100 Hz | 83,52 | 83,81 | 85,82 | 86,51 | 80,99 |
| 100 – 140 Hz | 93,42 | 94,48 | 96,27 | 96,12 | 90,55 |
| 140 – 180 Hz | 85,91 | 86,51 | 86,35 | 87,82 | 81,84 |
| 180 – 220 Hz | 92,85 | 94,25 | 94,96 | 95,07 | 90,50 |
| 220 – máx. Hz | 57,16 | 59,76 | 63,28 | 63,57 | 51,74 |

## C.3. Resultados experimento 3

Experimentos para la elección del tamaño óptimo de ventana única. Resultados en Auto-test (244.000 archivos de entrenamiento) y en test para 250 archivos por tramo:

Tabla C.7: Resultados pruebas Auto-test con entrenamiento de modelos acústicos realizado con todos los archivos (Completo). Tasa de aciertos de palabra ( %WORD)

| Ventana de análisis | 12,5 ms | 15 ms | 18,75 ms | 20 ms | 22,5 ms | 25 ms | 27,5 ms | 30 ms | 32,5 ms | 35 ms | 37,5 ms |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 – 100 Hz | 91,66 | 93,07 | 93,89 | 94,20 | 94,25 | **94,39** | 94,27 | 94,13 | 93,85 | 93,29 | 92,49 |
| 100 – 140 Hz | 90,23 | 91,40 | 92,19 | 92,34 | 92,53 | **92,58** | 92,48 | 92,31 | 92,04 | 91,42 | 90,42 |
| 140 – 180 Hz | 86,95 | 88,53 | 89,62 | 89,82 | 90,21 | **90,21** | 90,08 | 89,96 | 89,71 | 89,09 | 88,24 |
| 180 – 220 Hz | 88,68 | 89,92 | 90,91 | 90,98 | 91,19 | **91,20** | 91,06 | 90,85 | 90,57 | 90,07 | 89,25 |
| 220 – máx. Hz | 86,55 | 87,97 | 89,05 | 89,28 | 89,57 | **89,73** | 89,62 | 89,55 | 89,24 | 88,87 | 87,97 |
| Tasa acierto global | 88,89 | 90,19 | 91,12 | 91,28 | 91,52 | **91,56** | 91,44 | 91,28 | 91,01 | 90,45 | 89,54 |

Tabla C.8: Resultados pruebas test con entrenamiento de modelos acústicos realizado con todos los archivos (Completo). Tasa de aciertos de palabra ( %WORD)

| Ventana de análisis | 12,5 ms | 15 ms | 18,75 ms | 20 ms | 22,5 ms | 25 ms | 27,5 ms | 30 ms | 32,5 ms | 35 ms | 37,5 ms |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 – 100 Hz | 81,62 | 83,39 | 85,09 | 86,10 | **86,15** | 85,58 | 85,54 | 85,49 | 85,17 | 83,68 | 82,45 |
| 100 – 140 Hz | 93,50 | 95,09 | 96,46 | 96,39 | **96,96** | 96,38 | 96,38 | 96,79 | 96,28 | 95,09 | 92,81 |
| 140 – 180 Hz | 85,25 | 85,64 | 86,68 | 86,87 | 87,30 | **87,89** | 87,77 | 86,89 | 86,70 | 85,85 | 83,47 |
| 180 – 220 Hz | 93,71 | 93,82 | 94,68 | 95,19 | 94,65 | 95,32 | **95,51** | 95,28 | 95,11 | 94,39 | 93,71 |
| 220 – máx. Hz | 59,68 | 61,10 | **64,38** | 63,48 | 63,22 | 63,73 | 63,53 | 61,99 | 60,96 | 60,99 | 59,63 |
| Tasa acierto global | 82,75 | 83,81 | 85,46 | 85,61 | 85,66 | **85,78** | 85,75 | 85,29 | 84,84 | 84,00 | 82,41 |

## C.4. Resultados experimento 4

Pruebas con vídeos de Internet:

Tabla C.9: Resultados pruebas vídeos con entrenamiento de modelos acústicos realizado con todos los archivos (Completo). Tasa de aciertos de palabra ( %WORD)

| Ventana de análisis | 12,5 ms | 15 ms | 18,75 ms | 25 ms | 37,5 ms |
|---|---|---|---|---|---|
| Economía | 48,69 | 51,18 | 51,89 | **52,65** | 43,32 |
| Deportes | 47,49 | 49,33 | 50,21 | 50,88 | 42,06 |
| Moda | 24,11 | 25,22 | 26,67 | **28,80** | 26,36 |

Tabla C.10: Resultados pruebas vídeos con entrenamiento de modelos acústicos realizado con archivos específicos de cada tramo (Específico). Tasa de aciertos de palabra ( %WORD)

| Ventana de análisis | 12,5 ms | 15 ms | 18,75 ms | 25 ms | 37,5 ms |
|---|---|---|---|---|---|
| Nº Archivos E. | 113.000 | 148.000 | 185.000 | 131.000 | 25.000 |
| Economía | 40,01 | 48,75 | 50,74 | 45,92 | 31,09 |
| Deportes | 31,05 | 43,01 | 50,29 | **51,75** | 31,09 |
| Moda | 24,97 | 26,90 | 26,51 | 19,27 | 14,54 |

# Apéndice D

# Presupuesto

1. **Ejecución Material**
   - Compra de estación de trabajo ..... 2.000 €
   - Material de oficina ..... 200 €
   - Total de ejecución material ..... **2.200 €**
2. **Gastos generales**
   - 16 % sobre Ejecución Material ..... 352 €
3. **Beneficio Industrial**
   - 6 % sobre Ejecución Material ..... 132 €
4. **Honorarios Proyecto**
   - 1800 horas a 15 € / hora ..... 27.000 €
5. **Material fungible**
   - Gastos de impresión ..... 80 €
   - Encuadernación ..... 30 €
6. **Subtotal del presupuesto**
   - Subtotal Presupuesto ..... 29.794 €

7. **I.V.A. aplicable**

   - 21 % Subtotal Presupuesto ..... 6.256,74 €

8. **Total presupuesto**

   - Total Presupuesto ..... 36.050,74 €

Madrid, Marzo de 2015

El Ingeniero Jefe de Proyecto

Fdo.: Javier Antón Martín

Ingeniero de Telecomunicación

# Apéndice E

# Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, del DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO DE HABLA NATURAL INDEPENDIENTE DEL LOCUTOR. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho entorno. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.
Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

**Condiciones generales**

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se

estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado

en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4 % del presupuesto y la provisional del 2 %.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean

oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad “Presupuesto de Ejecución de Contrata” y anteriormente llamado ”Presupuesto de Ejecución Material” que hoy designa otro concepto.

### Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.