UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

PROYECTO FIN DE CARRERA

ADAPTACIÓN DE UN SISTEMA DE BÚSQUEDA DE PALABRAS CLAVE AL CASTELLANO

Junchen Xu

OCTUBRE 2014

ADAPTACIÓN DE UN SISTEMA DE BÚSQUEDA DE PALABRAS CLAVE AL CASTELLANO

AUTOR: Junchen Xu

TUTOR: Doroteo Torre Toledano

Área de Tratamiento de Voz y Señales (ATVS)

Dpto. de Tecnología Electrónica y de las Comunicaciones

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Octubre de 2014

Agradecimientos

Se acaba una etapa importante y especial en mi vida. Han sido años de mucho esfuerzo y sacrificio, pero la recompensa ha merecido la pena. Ha habido momentos de duda e incertidumbre, pero han servido para obtener las más firmes certezas. Y todo esto es gracias a la presencia de tantísimas personas que han estado a mi lado y a las que quisiera darles mi más sincero agradecimiento. Seguro que se me olvida alguien, espero que sepa perdonarme.

En primer lugar quisiera agradecer a mi tutor Doroteo Torre Toledano por concederme esta maravillosa oportunidad de trabajar con él, así como toda su ayuda y paciencia durante todos estos meses. A todos los miembros del grupo de Área de Tratamiento de Voz y Señales por su amabilidad y apoyo desde el primer momento.

Por supuesto tengo que agradecer a mis padres, Jianming y Lidi, a mis tíos, que son como mis segundos padres, y a mis abuelos, todo su amor y cariño. Sabéis que yo no sería nada sin vosotros, y que las pocas cualidades buenas que tengo os lo debo todo a vosotros. Me habéis enseñado a ser persona, a caminar, a correr, a caer y a levantarse. De todo corazón, ¡gracias!

También a todos mis compañeros de carrera con los que he tenido el gusto de coincidir durante estos años de convivencia. En especial a Luis, Patricia, Karim, José y Miriam, por tantos momentos compartidos, de muchas alegrías y no pocos sufrimientos. Por haberme ayudado a superar todos los obstáculos y, cómo no, por saber aguantarme, cosa para nada fácil. A Nasib, compañero infatigable de prácticas y risas, muchas gracias por tu esfuerzo y sentido de humor. A Ricardo y Herrero, tan buenos momentos pasamos y que espero repetir. Y a Pedro, Pablo, Javi, Marmota y Alberto, por esas pachangas de fútbol donde habéis podido comprobar mi reconocida inutilidad con el balón.

Y por último, no podía faltar aquí mi querida "vieja guardia": Nacho, Alex, Chema, David, Pedro, Álvaro, Sergio L. y Sergio G. Hemos crecido juntos, hemos aprendido juntos, nos hemos reído juntos y hemos vivido increíbles experiencias juntos. Pero sobre todo, gracias por estar conmigo en los momentos difíciles, eso tiene un valor incalculable. Desgraciadamente, cada vez es más difícil juntarnos todos, la vida tiene estas cosas, vaya. Pero estoy convencido de que, pase lo que pase, seguiremos riendo juntos.

A todos vosotros, gracias.

Junchen Xu

Resumen

El objetivo de este proyecto es el desarrollo e implementación de un sistema de búsqueda de palabras clave en castellano, partiendo de un sistema ya existente que funciona en inglés y vietnamita. La base de datos utilizada para entrenar y evaluar el sistema es el corpus Fisher Spanish, perteneciente al Consorcio de Datos Lingüísticos, con sede en Estados Unidos.

En primer lugar, se ofrece una breve explicación del sistema de producción del habla humano y el estado del arte en los campos de reconocimiento de voz y de palabras clave.

El sistema completo está compuesto por dos subsistemas que se encargan del reconocimiento de voz y búsqueda de palabras clave, respectivamente. El primero es el encargado de generar los *lattices*, sobre los que el segundo desarrollará la tarea de búsqueda. La modalidad de búsqueda de términos que se utiliza es la llamada Spoken Term Detection (STD).

Para evaluar el rendimiento del sistema, se han realizado pruebas de diferente naturaleza para los dos subsistemas. En esta memoria se recogen los resultados de esas pruebas y las conclusiones obtenidas.

Palabras clave

Búsqueda de palabras clave, detección de términos orales, Fisher Spanish, keyword spotting, lattice, palabras clave, reconocimiento de voz, STD.

Abstract

The objective of this project is to develop and implement a keyword spotting system in Spanish, based on an existing system that works in English and Vietnamese. The database used to train and evaluate the system is the Fisher Spanish corpus, belonging to the Linguistic Data Consortium, which is based in the United States.

First of all, a brief description of the human speech production system is provided, followed by the state of the art in speech recognition and keyword spotting.

The full system comprises two subsystems, responsible of speech recognition and keyword spotting, respectively. The former is in charge of generating lattices, which are used by the latter to search the keywords. The keyword spotting modality used is called Spoken Term Detection (STD).

In order to evaluate the system, different tests have been performed for the two subsystems. The results are captured in this document, as well as the conclusions.

Keywords

Fisher Spanish, keywords, keyword spotting, lattice, speech recognition, spoken term detection, STD.

Índice de contenidos

Índice de figuras

Índice de tablas

1 Introducción

1.1 Motivación

La información y la comunicación son partes esenciales del mundo desarrollado del siglo XXI, hasta tal punto que a la sociedad de hoy en día se le conoce como "la sociedad de la información". Una de las principales características de esta sociedad se encuentra en el hecho de que la cantidad de contenidos multimedia presentes en los medios de comunicación y en Internet, especialmente archivos de audio y vídeo, crece a un ritmo exponencial.

Sin embargo, sigue existiendo una llamativa dificultad a la hora de buscar y acceder a una pieza específica dentro de un amplio catálogo de archivos de audio-vídeo, al contrario de lo que sucede con los contenidos de texto. Esta situación pone de manifiesto la necesidad de un sistema de búsqueda de contenidos de este tipo que facilite de alguna manera esta compleja tarea.

En este sentido, el reconocimiento de palabras clave (*keyword spotting* en inglés) juega un papel cada vez más importante. Los sistemas de búsqueda de contenidos de audio-vídeo basados en esta técnica tendrían una utilidad fundamental en las siguientes aplicaciones:

1. Recuperación de información multimodal en Internet o en bases de datos de archivos multimedia. Un claro ejemplo sería programas de televisión o radio.
2. Búsqueda de información en grabaciones de reuniones o conferencias
3. Servicios de inteligencia, como las escuchas telefónicas.
4. Call-centers, en lo referido al aseguramiento de la calidad del servicio ofrecido.
5. Sistemas de enrutamientos de llamadas en contestadores automáticos.

Para que estas funcionalidades se hagan con un determinado nivel de calidad, es imprescindible que el sistema sea capaz de recibir correctamente la información de entrada, extraer los parámetros característicos de la misma, realizar una búsqueda rápida y efectiva en la base de datos y presentar los resultados al usuario de forma clara y precisa.

Actualmente existen sistemas de búsqueda de palabras clave que funcionan con una eficacia más que notable en diferentes lenguas, donde destaca sobre todo el inglés, una situación consecuente con el hecho de ser el idioma más empleado en el mundo y en la investigación que se realiza en este campo. En cambio, tanto el número de sistemas desarrollados para la lengua castellana como los resultados obtenidos son mucho más modestos, motivo por el cual este proyecto intenta proporcionar una mejora en este sentido.

1.2 Objetivos

El objetivo principal de este proyecto es adaptar un sistema de búsqueda de palabras clave en inglés y vietnamita a la lengua castellana, asegurando el eficiente funcionamiento del original.

El sistema original en cuestión ha participado en la evaluación *NIST Open KWS 2013* [1] y se encuentra en constante evolución de cara a futuras convocatorias de la misma.

El sistema completo, tal como se puede observar en la Figura 1-1, está compuesto por dos subsistemas diferentes que desempeñan funciones bien diferenciadas:

- Subsistema de reconocimiento de voz. Se encarga de procesar los ficheros de audio de la base de datos e intentar reconocer de forma automática la información contenida. Para ello se elaboran y se entrenan un modelo de lenguaje y diferentes modelos fonéticos a partir de la base de datos de entrada. Durante la última fase del entrenamiento se procede a decodificar los grafos entrenados y a extraer los llamados *lattices* fonéticos de los datos de evaluación, que son grafos acíclicos dirigidos que representan las secuencias de fonemas y/o palabras más probables contenidas en el audio de entrada y que se utilizarán en el siguiente subsistema.
- Subsistema de reconocimiento de palabras clave. Se encarga de buscar e identificar todas las ocurrencias de una serie de términos concretos en los ficheros de audio, en este caso en los *lattices* obtenidos anteriormente. Estos *lattices* son convertidos en índices por el subsistema, donde se efectuarán las búsquedas. La Figura 1-2 contiene un ejemplo de *lattice* fonético.

En este proyecto se implementarán todos los elementos necesarios para el funcionamiento de los dos subsistemas. Algunos de estos elementos ya existen en el sistema original, en cuyo caso son adaptados para que funcionen con las características del nuevo sistema (diferente lengua con reglas gramaticales diferentes, particularidades de la base de datos, etc.). Por otro lado, el resto de elementos serán diseñados desde cero.

Una vez que estén disponibles todos los elementos, ambos subsistemas serán entrenados. Durante esta fase, los diversos parámetros disponibles se ajustarán de manera que los resultados sean los mejores posibles.

Al finalizar el entrenamiento, los subsistemas serán sometidos a diferentes pruebas para comprobar su funcionamiento. Los resultados obtenidos serán analizados de manera apropiada y se ofrecerán las pertinentes conclusiones.

La base de datos que se utilizará en este proyecto consiste en una serie de grabaciones de conversaciones telefónicas efectuadas entre locutores hispanohablantes de diferentes nacionalidades, edad, sexo y formación académica.

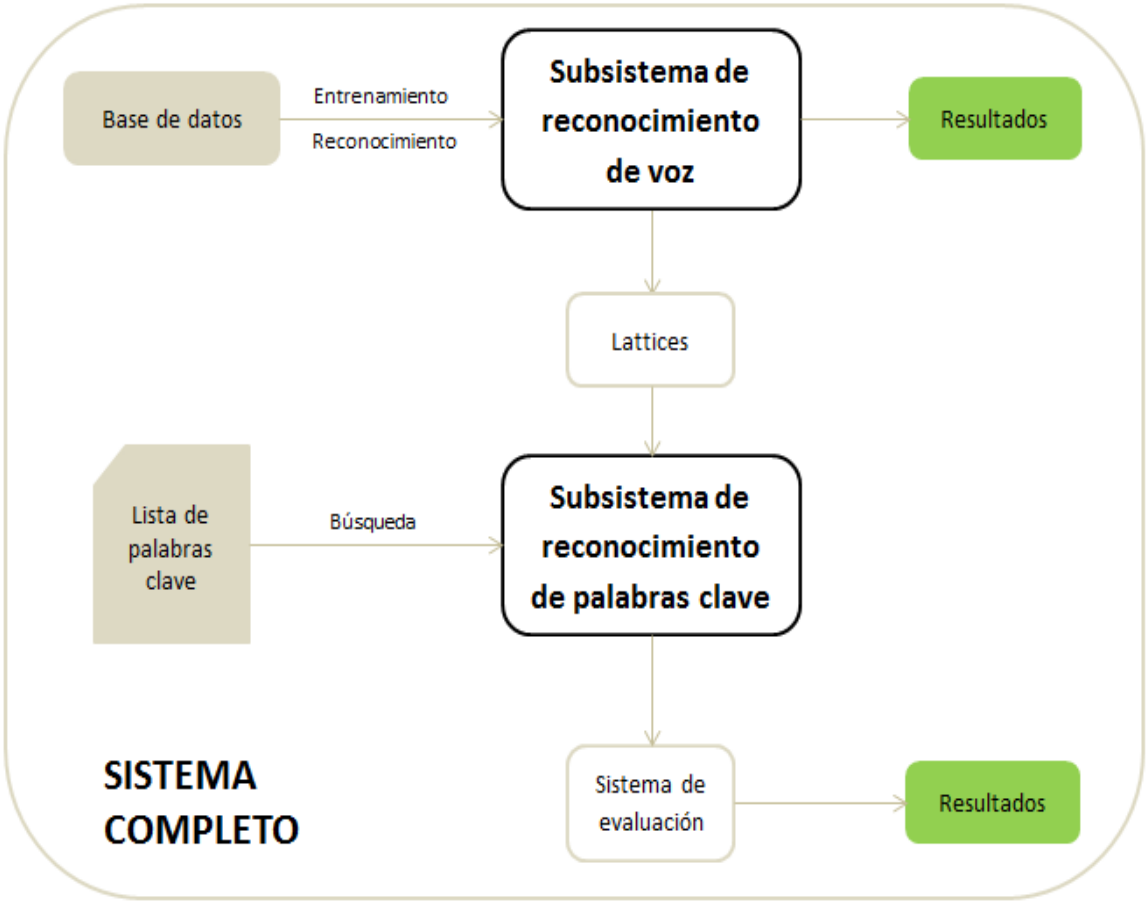

Figura 1-1: Esquema general del sistema completo a desarrollar

Con la intención de comprobar el rendimiento real del sistema y compartir con el resto de la comunidad investigadora los progresos obtenidos en este proyecto, se ha decidido a participar en la evaluación "ALBAYZIN 2014 Search on Speech" [2], convocada dentro del marco de "IberSPEECH 2014, VII Jornadas en Tecnologías del Habla" [3], concretamente en la tarea de Spoken Term Detection (STD). En este sentido, la parte final del proyecto se centra en adaptar y entrenar el sistema con la base de datos empleada en esta evaluación.

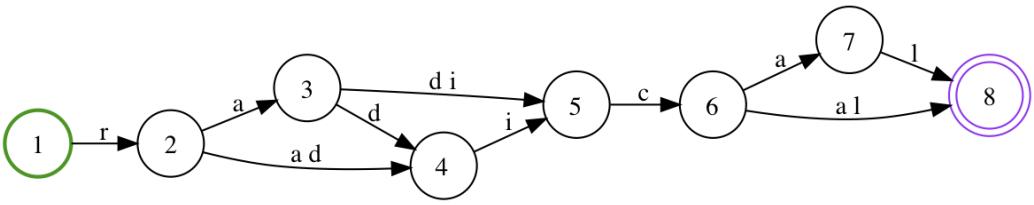

Figura 1-2: Ejemplo de *lattice* fonético

1.3 Organización de la memoria

Esta memoria está estructurada en seis capítulos que se resumen brevemente a continuación.

CAPÍTULO 1: INTRODUCCIÓN

Este capítulo contiene una breve descripción de la motivación y los objetivos de este proyecto.

CAPÍTULO 2: ESTADO DEL ARTE

En este capítulo se desarrolla un detallado estudio del estado del arte de los sistemas y las técnicas que se han utilizado en este proyecto.

En primer lugar se explica brevemente la producción del habla humano, así como algunos conceptos básicos de la fonología y la fonética que sirven para entender el mecanismo de funcionamiento del reconocimiento de voz. A continuación se describe el estado del arte de los dos tipos de sistemas a desarrollar: reconocimiento voz y reconocimiento de palabras clave. Y se enumera en último lugar las técnicas más utilizadas hoy en día y sus aplicaciones más frecuentes.

CAPÍTULO 3: ENTRENAMIENTO DEL SUBSISTEMA DE RECONOCIMIENTO DE VOZ

En este capítulo se explica el diseño, la implementación y el entrenamiento del primero de los subsistemas que componen el sistema completo. Se proporciona una descripción de los elementos más destacados del subsistema, las relaciones entre ellos, la preparación de los datos de entrada, el paso a paso de cómo y con qué técnicas se ha llevado a cabo el entrenamiento, así como los datos de salida intermedios que se producen.

CAPÍTULO 4: ENTRENAMIENTO DEL SUBSISTEMA DE RECONOCIMIENTO DE PALABRAS CLAVE

Este capítulo se centra en el segundo de los subsistemas que componen el sistema completo. Se describe igualmente los elementos principales del subsistema, las relaciones entre ellos, las diferentes fases de entrenamiento y los datos de entrada que se necesitan para ello.

CAPÍTULO 5: PRUEBAS Y RESULTADOS

En este capítulo se detallan las pruebas que se han realizado para comprobar el rendimiento de los subsistemas desarrollados, junto con los resultados obtenidos. Las pruebas se efectúan tanto con la versión inicial del sistema como con la modificada para participar en la evaluación de "ALBAYZIN 2014 Search on Speech".

CAPÍTULO 6: CONCLUSIONES Y TRABAJO FUTURO

Este capítulo contiene un análisis de los resultados obtenidos con las pruebas y las conclusiones a las que se ha llegado.

Y para finalizar, se propone una serie de líneas de trabajo relacionadas con el sistema desarrollado y la tecnología empleada que deberían ser exploradas en trabajos futuros.

2 Estado del arte

2.1 Introducción

El habla es una cualidad inherentemente ligada a la naturaleza del ser humano, posiblemente la más determinante a la hora de diferenciarnos del resto de animales, a la vez que supone una excelente forma de comunicación e inteligencia. Estas características le permiten ser utilizada en numerosos campos hoy en día hasta el punto de que a su alrededor se han desarrollado las llamadas "tecnologías del habla", dedicadas al estudio, análisis, procesamiento y generación de señales del habla mediante el uso de herramientas computacionales. Su desarrollo comenzó a despegar a mediados del siglo XX y su crecimiento ha sido constante desde entonces.

Este progreso viene motivado por diferentes motivos. Primero, los avances en el campo de las tecnologías de la información permiten que hoy en día sea relativamente sencillo crear y mantener bases de datos de archivos de voz de tamaños muy considerables. Gracias a esta ingente cantidad de datos disponibles, junto con la posibilidad de utilizar herramientas de cálculo más potentes, rápidas y eficientes que antes, ha sido posible experimentar con nuevas líneas de investigación, probar algoritmos más complejos y conseguir en definitiva resultados cada vez más eficientes y prácticos.

De entre todas las ramas que forman estas tecnologías, este proyecto centra su atención en el reconocimiento de palabras clave. En este capítulo se expondrán los avances obtenidos hasta ahora en este campo, así como los sistemas más extendidos. Como condición indispensable para un correcto entendimiento de esto, se hará una pequeña introducción al proceso de producción y percepción del habla, sus características y, de forma simplificada, el reconocimiento de voz.

2.2 Proceso del habla humano

En el campo de la fonética, la voz es el término utilizado para caracterizar el sonido del habla.

El proceso de producción del habla propiamente dicho se denomina fonación y se realiza durante la respiración, cuando el aire contenido en los pulmones sale de éstos por simple contracción de la caja torácica y el diafragma y llega a la laringe a través de los bronquios y la tráquea.

Hay tres mecanismos básicos de producción de voz que dan lugar a tres tipos de sonidos diferentes:

- El paso del aire expelido de los pulmones a través de la tráquea puede generar vibraciones en las cuerdas vocales, produciendo de esta manera los sonidos sonoros. Ese aire pasa después a la cavidad bucal o nasal donde se configuran y matizan los diversos sonidos. A la frecuencia de vibración de las cuerdas se le llama frecuencia fundamental, que depende de la presión ejercida al pasar el aire

por las cuerdas y de la tensión de éstas. El rango de frecuencias donde se encuentra la frecuencia fundamental suele ser más amplio en las mujeres que en los hombres, concretamente entre 125 y 225 Hz para la voz femenina y entre 80 y 140 Hz para la masculina [4].

- Las interrupciones parciales en el flujo de aire que sale de los pulmones dan lugar a los sonidos fricativos. Se producen como resultado de un estrechamiento del tracto vocal por el que se hace pasar el aire, dando lugar a una excitación ruidosa debida a las turbulencias que se producen en el estrechamiento.
- Las interrupciones totales dan lugar a una oclusión, que va seguida de una apertura brusca del tracto vocal. Estos sonidos caracterizados por tener una parte oclusiva sin apenas energía seguida de una parte de alta energía (a veces conocida como explosión) se denominan sonidos oclusivos.

De forma simplificada, se puede describir el aparato fonador humano necesario para generar la voz como la combinación de los órganos de respiración (pulmones, bronquios y tráquea); los órganos de fonación (laringe, faringe, cuerdas vocales, resonador nasal y bucal) y los órganos de articulación (paladar, lengua, dientes, labios y glotis). La Figura 2-1 ilustra algunos de estos elementos.

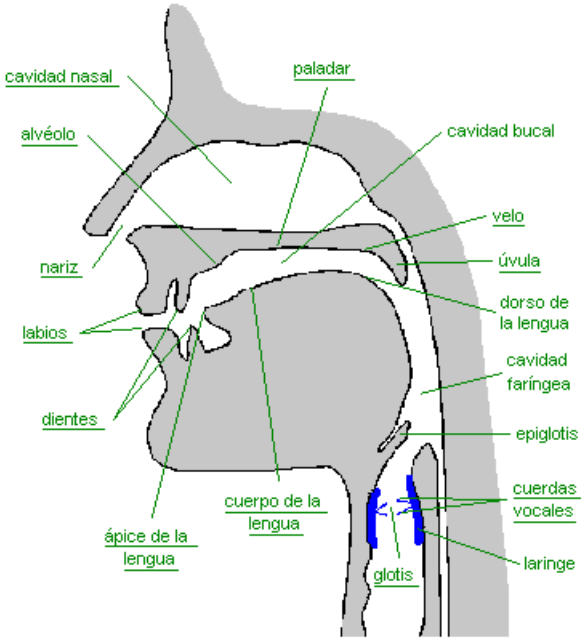

Figura 2-1: Aparato fonador humano [5]

2.2.1 Alófonos y fonemas

Un concepto fundamental a la hora de describir cualquier sistema de reconocimiento de voz es el de fonema. La definición comúnmente aceptada de fonema la describe como la unidad teórica básica necesaria para estudiar el nivel fonético-fonológico de una lengua humana [6]. Dicho de otra manera, un fonema es cada una de las unidades segmentales postuladas para un sistema fonológico que represente los sonidos de una lengua.

Otro concepto importante es el de alófono, que se define como cada uno de los sonidos que en un idioma dado se reconoce como un determinado fonema. La diferencia entre ambos radica en que el fonema pertenece a la lengua, mientras que el alófono, que representa al fonema, pertenece al habla.

La palabra <lana>, por ejemplo, consta de cuatro fonemas, que son (/l/, /a/, /n/, /a/). A esta palabra le corresponden cuatro alófonos en el habla, con la peculiaridad de que éstos pueden variar según el individuo que lo pronuncie, en función de una serie de rasgos fonéticos y articulatorios.

El número de fonemas en cada lengua está claramente establecido (24 en el caso del castellano), aunque puede sufrir variaciones si se introducen cambios en las reglas de pronunciación. Para definir apropiadamente un fonema, se debe tener en cuenta sus tres principales características:

- Diferenciadora: cada fonema posee cualidades que le distinguen de los demás y porta además una intención significativa especial. Como ejemplo, las palabras <cara> (/k/ /a/ /r/ /a/) y rara (/R/ /a/ /r/ /a/) se distinguen semánticamente debido a que /k/ y /R/ son diferentes por sonoridad.
- Indivisible: no se puede descomponer en unidades menores.
- Abstracta: no son sonidos, sino modelos o representaciones de éstos.

2.2.2 Coarticulación

Aunque los fonemas de un determinado idioma deben ser, por definición, claramente diferenciados unos de otros, la forma en que se pronuncian depende en parte de los fonemas anteriores y posteriores debido a inercias del aparato fonador. Dicho de otro modo, un fonema es pronunciado de forma diferente desde el punto de vista acústico, ya que inconscientemente el cerebro humano piensa en los fonemas que vendrán después y prepara los órganos físicos para articularlos correctamente. Del mismo modo, al pronunciar un fonema éste se ve afectado por la posición anterior en que estaban los órganos articuladores. A este fenómeno se le conoce como coarticulación [7] y es la principal razón para que a la hora de modelar los fonemas se empleen habitualmente modelos dependientes del contexto.

2.3 Reconocimiento de voz

Generalmente, el reconocimiento automático de voz (o reconocimiento automático del habla) es considerado como la disciplina encargada de reconocer de forma automática la voz humana, convirtiéndola habitualmente en texto. El principal problema de esta disciplina es el de utilizar de manera adecuada y eficiente conocimientos e informaciones provenientes de diversas fuentes (acústica, fonética, léxica, sintáctica, semántica y pragmática) para poder obtener una interpretación razonable de los mensajes acústicos en cuestión.

Un sistema de reconocimiento de voz es una herramienta computacional con la capacidad de procesar la señal de voz emitida por el ser humano y reconocer de forma automática la información contenida en ésta.

2.3.1 Modelos fonéticos

A la hora de diseñar un sistema de reconocimiento de voz se necesitan dos mecanismos para lograr tal fin: los modelos fonéticos y los modelos de lenguaje.

Un modelo de lenguaje es un modelo estadístico que asigna, tras procesar los datos de entrenamiento, diferentes probabilidades a secuencias de palabras de acuerdo al número de sus ocurrencias en el entrenamiento. Su función consiste en intentar predecir la siguiente palabra en una secuencia de voz.

Por otra parte, un modelo fonético es un modelo matemático que permite discriminar entre los distintos sonidos de una lengua, y se obtiene utilizando locuciones de audio de voz, sus correspondientes transcripciones de texto y software apropiado para modelarlos. El objetivo de un modelo fonético es representar tanto las características del canal acústico (relativas al ambiente donde se realizaron las grabaciones y los dispositivos físicos utilizados) como de la lengua (relativas a los locutores).

En función del nivel de resolución que se quiera obtener con las unidades fonéticas se pueden utilizar diferentes modelos fonéticos. Aunque es habitual, con la intención de reflejar de manera adecuada los efectos de la coarticulación mencionados en la sección 2.2.2, emplear modelos fonéticos dependientes del contexto.

- Monofonemas: son unidades totalmente libres de contexto. Un monofonema tiene en cuenta todas las posibles realizaciones de un fonema independientemente de los que están a su alrededor.
- Bifonemas: son unidades que dependen solo de uno de sus contextos, ya sea el fonema anterior (izquierdo) o el posterior (derecho).
- Trifonemas: son unidades que dependen de ambos contextos a la vez. Como consecuencia, el número de unidades se incrementaría exponencialmente, aunque a cambio proporcionaría resultados mejores en el sistema.
- Trifonemas agrupados: los trifonemas con el mismo fonema central y contextos similares se agrupan a la hora de realizar el entrenamiento como solución a la problemática de que algunos trifonemas, de forma aislada, no ocurren con suficiente frecuencia en los datos de entrenamiento, lo cual resulta en un modelado menos preciso.

El proceso de la creación de estos modelos se conoce como "entrenamiento" y consta de dos fases:

1. Extracción de las características. En esta etapa se extraen los parámetros característicos del sonido en el dominio frecuencial.

2. Entrenamiento y reconocimiento de modelos. En esta etapa se construyen para cada fonema los modelos estadísticos que se utilizarán para reconocer la presencia de estos mismos fonemas en otras locuciones.

2.3.2 Técnicas de entrenamiento de modelos fonéticos

Las técnicas que se pueden emplear en el proceso de entrenamiento de modelos fonéticos son diversas y han ido evolucionando de manera constante. A continuación se listan las más empleadas, dejando constancia que las dos primeras están prácticamente en desuso debido a que las dos últimas ofrecen unas prestaciones significativamente mejores:

- Dynamic Time Warping (DTW) [8]: es un algoritmo que mide la similitud entre dos secuencias que pueden variar en tiempo o en velocidad. Consiste en alinear de forma temporal los parámetros de los modelos y los parámetros de los ficheros a evaluar, obteniendo la función que alinea a ambos, eligiendo la función de menor coste posible. Esta técnica ha sido ampliamente utilizada hasta la aparición de los HMM. En la actualidad encuentra aplicación en problemas específicos como en la conocida como "Query-by-Example" en la que se trata de encontrar un segmento de voz dentro de una base de datos de voz. La Figura 2-2 contiene un ejemplo explicativo de la técnica DTW.

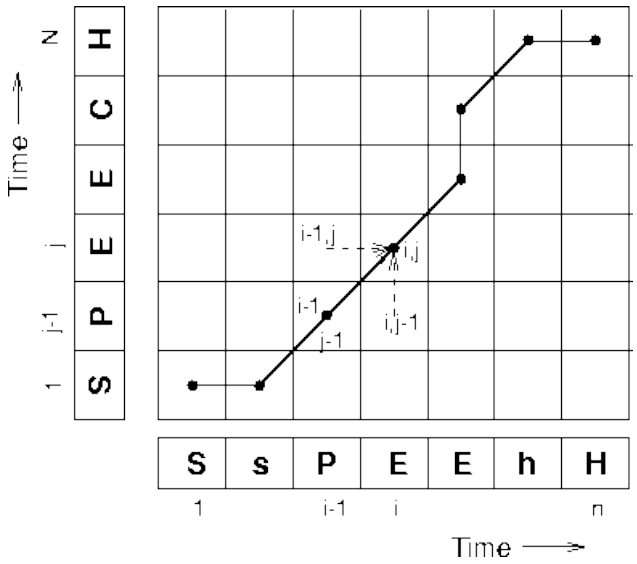

Figura 2-2: Ejemplo explicativo de la técnica DTW

- Vectorial Quantization (VQ) [9]: es una técnica muy utilizada en procesamiento de señales que permite modelar las funciones de densidad de probabilidad mediante el uso de vectores prototipo. Las características de los fonemas se representan en un espacio vectorial, donde cada fonema es un vector llamado "centroide". Al fonema a reconocer se le asigna en cada caso el vector que se encuentre a menor distancia de él. Esta técnica se sigue empleando ampliamente en codificación (o compresión) de voz.

- Hidden Markov Model (HMM): es un modelo estadístico que produce una secuencia de símbolos como salida. Su utilidad se basa en el hecho de que una señal de voz puede ser visto como un proceso de Markov, de manera que se aprovechan sus propiedades estocásticas a la hora de modelar los fonemas. Esta técnica es la base de la herramienta Kaldi y por tanto, la empleada en este proyecto, con lo que su funcionamiento se explicará más en detalle en la siguiente sección.
- Neural Networks (NN) [10]: es una técnica de entrenamiento discriminativo sin hacer suposiciones en cuanto a las propiedades estadísticas de las características. Se emplean en varias etapas del reconocimiento, en la extracción de parámetros o en el modelado de la probabilidad de observación de un vector de estadísticas en un estado de un HMM. En los últimos años esta técnica ha experimentado un gran avance, permitiendo el uso de estructuras multicapa (más de tres), algo prohibitivo hasta hace unos años debido a su exigente coste computacional y a problemas relacionados con la convergencia a mínimos locales no adecuados. Estas nuevas estructuras reciben el nombre de Deep Neural Networks (DNN) [11] [12] y suponen el Estado del Arte en reconocimiento de voz, consiguiendo mejores resultados que modelos basados en Gaussian Mixture Model (GMM) [13]. La Figura 2-4 recoge una red DNN.

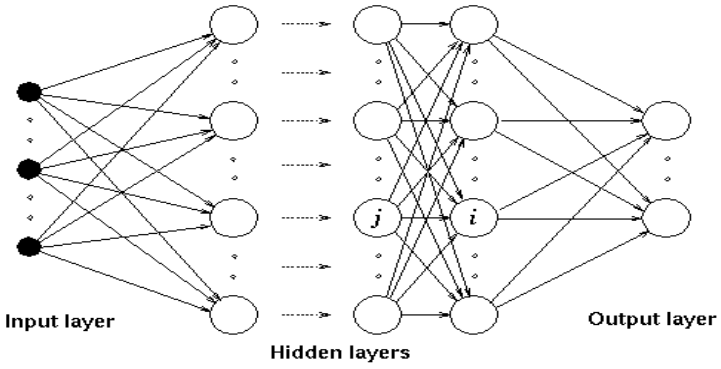

Figura 2-3: Ejemplo de DNN con varias capas ocultas

2.3.3 Modelos Ocultos de Markov (HMM)

Por lo explicado en la sección anterior, queda claro que los DNN son el estado del arte en el campo de reconocimiento de voz. Sin embargo, debido a que su implementación aún no es del todo madura en la herramienta que se va a utilizar en este proyecto, se ha decidido usar los HMM en su lugar. A continuación, se ofrece una explicación más detallada de estos modelos.

Un modelo de Markov es un modelo estadístico donde se asume que el sistema a modelar es un proceso de Markov, lo cual significa que "la probabilidad condicional sobre el estado presente, futuro y pasado del sistema son independientes".

En un modelo oculto de Markov los estados del sistema no son visibles. Cada estado tiene una distribución de probabilidad sobre las posibles salidas. Dicho de otra manera, la

secuencia de salidas de un HMM da cierta información acerca de la secuencia de estados. El objetivo es determinar los parámetros desconocidos u ocultos a partir de los parámetros observables. En la Figura 2-5 se puede observar un HMM.

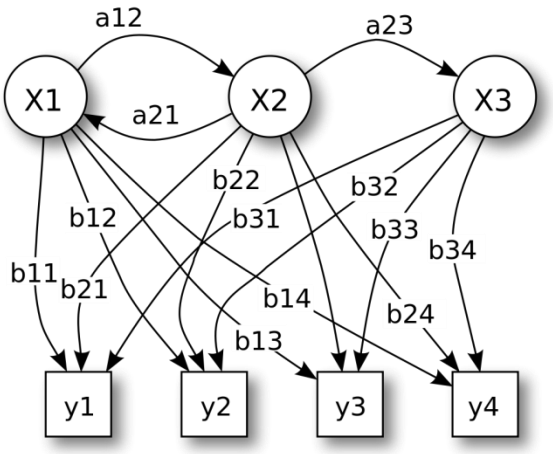

Figura 2-4: Ejemplo de Modelo oculto de Markov (HMM), con estados ocultos y salidas observables

Esta técnica se adapta muy bien a la problemática del reconocimiento de voz [14] [15] [16]. El sonido que se escucha de una grabación de voz es producto de una interacción de diversos factores, como ya se ha explicado en la Sección 2.2. Determinados sistemas de reconocimiento consideran la producción interna de voz como una secuencia de estados ocultos y el sonido resultante como una secuencia de estados observables generada por este proceso.

En un HMM la secuencia de estados observables está probabilísticamente relacionada con el proceso oculto. Las conexiones entre los estados ocultos y observables representan la probabilidad de generar un estado observado sabiendo que el proceso de Markov se encuentra en un determinado estado oculto.

En resumen, un modelo oculto de Markov es la composición de dos procesos estocásticos definidos como:

- Una cadena oculta de Markov, no observable directamente al tratarse del estado real del sistema, que modela la producción del habla teniendo en cuenta la variabilidad temporal.
- Un proceso observable que tiene en cuenta la variabilidad espectral y va tomando valores en el espacio de las características acústicas que resultan del proceso de producción del habla.

La correcta combinación de ambos procesos modela las fuentes de variabilidad de la señal de habla y permite modelar una secuencia de parámetros acústicos que caracterizan los diferentes fonemas.

En este punto se debe mencionar los tres problemas canónicos a la hora de utilizar los modelos ocultos de Markov [14] [15]. Dado que el estudio de estos modelos no es el fin último de este proyecto, simplemente se nombrará estos problemas y los algoritmos que se utilizan como solución a éstos.

1. Problema de evaluación: se considera el problema donde hay un gran número de HMM describiendo diferentes sistemas, y una secuencia de observaciones. El objetivo es saber cuál de ellos tiene la mayor probabilidad de haber generado dicha secuencia. Este problema ocurre en reconocimiento de habla cuando se está usando un gran número de HMM, cada uno de ellos modelando una palabra diferente. Una secuencia de observaciones se forma a partir de una palabra "de entrada", y esta palabra se reconoce mediante la identificación del HMM (palabra de la base de datos) más probable. Se soluciona con el "algoritmo Forward-Backward".
2. Problema de reconocimiento de estados: este problema consiste en encontrar la secuencia más probable de estados ocultos que generaron una salida observada. En muchos casos el interés por los estados ocultos se deben a que representan unidades más pequeñas que los fonemas, de modo que a partir de la secuencia de estados se puede obtener la secuencia de fonemas. La solución a este problema es el "algoritmo de Viterbi".
3. Problema de entrenamiento: consiste en entrenar los parámetros del HMM dada una secuencia de datos de entrenamiento con el fin de maximizar la probabilidad de observar la secuencia dada. Este problema, quizás el más difícil de los tres, se resuelve con el "algoritmo de Baum-Welch".

2.3.4 Aplicaciones

Entre las principales aplicaciones prácticas del reconocimiento de voz se puede enumerar:

- Transcripción automática del habla en texto, disciplina conocida como Speech-To-Text (STT).
- Sistemas de traducción automática.
- Sistemas de subtitulado automático de contenidos multimedia.
- Dictado automático de recetas médicas o textos legales, entre otros.
- Control por comandos con sistemas diseñados para dar órdenes a un ordenador. Suelen reconocer un número muy reducido de palabras pero a cambio requieren un rendimiento casi perfecto.
- Interfaz de usuario en dispositivos informáticos y electrónicos.
- Sistemas diseñados para discapacitados, especialmente con problemas auditivos y visuales.

2.4 Reconocimiento de palabras clave

El reconocimiento de palabras clave, o en inglés Keyword Spotting (KWS), es una parte del reconocimiento de voz, cuya aplicación consiste en identificar palabras concretas dentro de una locución de habla continua. Se trata de algoritmos basados en la búsqueda mecanizada de palabras clave en archivos de audio, de manera que se pueda localizar fácilmente un determinado archivo que contiene una o más palabras clave dentro de un conjunto, así como la o las posiciones del archivo donde aparecen las palabras clave, para su posterior edición o indexación, todo ello con un considerable ahorro de tiempo con respecto a las búsquedas manuales y sin necesidad de escuchar los archivos completos.

2.4.1 Tipos de sistemas de búsqueda de palabras clave

En la actualidad, los sistemas de reconocimiento de palabras clave se clasifican en cuatro tipos:

1. *Keyword Spotting (KWS),* donde la entrada del sistema es una lista de términos, la cual es conocida a la hora de procesar el audio y hace que los reconocedores basados en palabras sean muy efectivos para decidir entre las diferentes hipótesis de detecciones.
2. *Spoken Term Detection (STD),* donde la entrada del sistema es una lista de términos (igual que en KWS) pero ésta es desconocida a la hora de procesar el audio.
3. *Query-by-Example Spoken Term Detection (QbE STD),* donde la entrada del sistema es un ejemplo de consulta acústica por lo que no se puede obtener un conocimiento previo de la transcripción correcta correspondiente a la palabra o al fonema. Este sistema tiene que generar, para cada consulta, un conjunto de ocurrencias detectadas en los ficheros de audio, junto con sus marcas de tiempo (igual que en STD).
4. *Query-by-Example Spoken Document Retrieval (QbE SDR),* donde la entrada del sistema está formada por varios ejemplos de consulta acústica por lo que no se puede obtener un conocimiento previo de la transcripción correcta correspondiente a la palabra o al fonema. Este sistema tiene que generar una puntuación de salida para cada una de las consultas, reflejando la probabilidad de cada una de ellas de aparecer en el fichero de audio, y no se requiere información sobre sus marcas de tiempo.

Este proyecto se va a centrar en el segundo tipo, STD, por lo cual a continuación se van a citar los tipos de sistemas más utilizados [17], de acuerdo a su método de funcionamiento.

2.4.2 Reconocedores de habla continua de gran vocabulario

También conocido con el nombre LVCSR (Large Vocabulary Continuous Speech Recognition) por utilizar un amplio diccionario de la lengua concreta en la que se está buscando las palabras. Con él se realiza un reconocimiento para extraer todas las palabras presentes en la locución, y se obtiene la lista de los términos buscados gracias a un detector de palabras clave.

Este sistema funciona muy bien y de manera muy rápida en el caso de que todas las palabras a reconocer formen parte del vocabulario, cosa que no siempre sucede o que requiere una constante actualización del vocabulario.

Pero como era de esperar, si la palabra buscada no está recogida, entonces será imposible encontrarla. Una palabra de este tipo se conoce como palabra fuera de vocabulario (OOV) y engloba términos como nombres propios, extranjerismos, acrónimos, etc.

Adicionalmente, estos sistemas presentan ciertos problemas relacionados con la robustez, ya que requieren que las condiciones de la grabación y de la voz sean las mismas en entrenamiento y en evaluación para obtener resultados satisfactorios.

Se trata, por tanto, de encontrar algún sistema que pueda responder al problema de las palabras OOV y complemente a un sistema LVCSR.

2.4.3 Modelos de relleno

Antes de explicar estos sistemas, hay que mencionar que durante el proceso de decodificación se propone la secuencia más probable de palabras existentes en el audio. De esta manera, hay que tener en cuenta las palabras clave y cualquier otro tipo de palabra, sonido o efecto que puedan aparecer en el archivo. Por este motivo, los modelos de relleno son utilizados para modelar los intervalos de la grabación donde no estén presentes ninguna de las palabras clave. Como consecuencia de esto, el diccionario de estos sistemas está formado únicamente por las palabras clave que se desea detectar y los mencionados modelos de relleno.

A la salida del sistema reconocedor, tanto las palabras clave detectadas durante la decodificación como los modelos de relleno se someten a unas medidas de confianza previamente establecidas, que hacen la función de umbral. Estas medidas se utilizan para detectar errores de reconocimiento al obtener una puntuación por debajo del umbral. Con este procedimiento, sólo se intenta reconocer unas palabras determinadas, el resto de audio se asigna a modelos de relleno.

Las principales ventajas de esta técnica son su alta precisión y su facilidad de uso en comparación con los sistemas LVCSR. Ofrece, además, la posibilidad de utilizar como unidad de trabajo tanto las palabras como sus sub-unidades: fonemas, grafemas, sílabas y clases fonéticas.

Por contra, este tipo de sistemas tampoco son capaces de identificar las palabras fuera de vocabulario al trabajar con un conjunto de palabras clave previamente definido.

2.4.4 Reconocedores de voz de sub-unidades de palabra

Estos sistemas surgieron como respuesta al principal problema que presentan las técnicas anteriores. La idea clave consiste en que se utiliza las llamadas "sub-unidades" de palabra (fonemas, sílabas, etc.) como unidad de trabajo a la hora de entrenar y evaluar. Estas sub-unidades poseen la particularidad de que son inalterables dentro de una lengua. De este modo, las infinitas combinaciones de secuencias de éstas permitirían obtener cualquier

palabra imaginable de una lengua, solucionando así el problema de las palabras fuera de vocabulario.

El funcionamiento de estos sistemas se estructuran en dos partes: en la primera se realiza el proceso de reconocimiento de voz basado en sub-unidades y utilizando modelos fonéticos para generar un índice; mientras que en la segunda se realiza la búsqueda de los términos, donde a través del detector de palabras clave y las medidas de confianza, el sistema extrae el listado de las palabras solicitadas a la salida.

En términos generales, la búsqueda con esta técnica es mucho más rápida que los casos anteriores, aunque a cambio se pierde precisión. Este inconveniente se puede mejorar si se utiliza conjuntamente con otros sistemas, como los LVCSR.

2.4.5 Aplicaciones

Hoy en día los sistemas de reconocimiento de palabras clave son utilizados en diversidad de campos, como por ejemplo:

- En los call-center, para monitorizar la calidad del servicio.
- En medios de comunicación, para control de impacto mediático o publicitario y detección de menciones.
- Búsqueda de archivos de audio o audiovisuales.
- Procesamiento automático de escuchas telefónicas.

3 Entrenamiento del subsistema de reconocimiento de voz

3.1 Introducción

Este capítulo se va a centrar en el desarrollo del subsistema de reconocimiento de voz. En las siguientes secciones se explicarán paso a paso las diferentes fases que componen el proceso, los ficheros más importantes que se generarán en algunas de las ellas, así como las diferentes técnicas utilizadas.

En la Figura 3-1 se puede observar de manera general cuáles son esas fases y cómo están relacionadas.

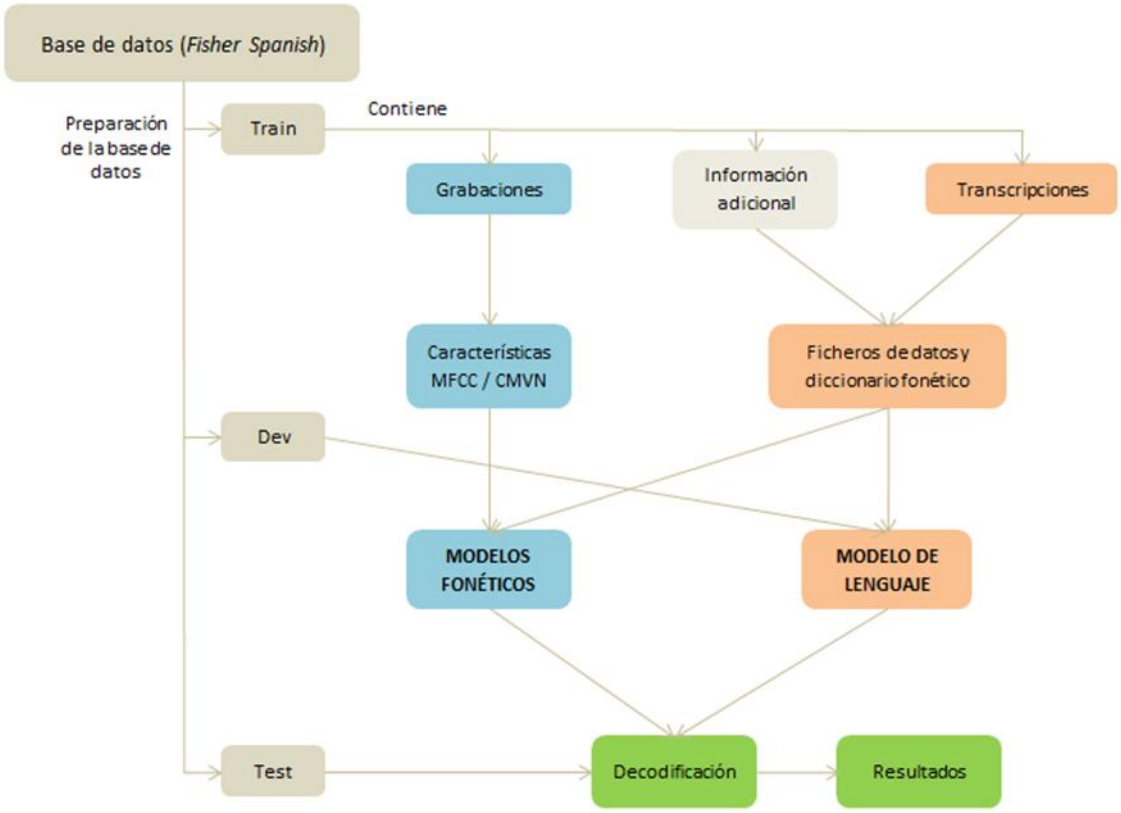

Figura 3-1: Diagrama de bloques del entrenamiento del subsistema de reconocimiento de voz

3.2 Medios disponibles

Antes de proceder a explicar las fases del entrenamiento se va a numerar los medios que se han utilizado para diseñar y desarrollar el sistema completo. Cabe destacar que la base de datos que se empleará para el entrenamiento y la evaluación del sistema no se detallará aquí sino en la sección 3.3, dedicada expresamente al análisis de la misma.

3.2.1 Hardware

El hardware empleado es un ordenador con procesador de 4 núcleos Intel Core i5-2400 de 3.1GHz y 4GB de memoria RAM. Se ha utilizado en el diseño del sistema, así como en las fases iniciales del desarrollo, tales como la preparación de base de datos, entrenamiento del modelo de lenguaje o de modelos acústicos en sus primeras fases, donde los requisitos de potencia son relativamente bajos.

Una vez familiarizado con la naturaleza del sistema y debido principalmente al aumento de potencia exigida en las últimas fases de entrenamiento, se ha utilizado también un equipo informático de mayor potencia, compuesto por un servidor HP Proliant de 24 núcleos de cómputo conectados a un servidor de disco de 12TB en RAID.

Estos medios fueron suministrados por el grupo de investigación ATVS de la Universidad Autónoma de Madrid [18].

3.2.2 Software

3.2.2.1 Herramienta de reconocimiento de voz Kaldi

El medio más importante de los empleados es Kaldi [19], un conjunto de herramientas de carácter libre y de código abierto para la investigación en el campo del reconocimiento de voz que destaca por su potencia, robustez, versatilidad y extensibilidad.

La herramienta utiliza dos conjuntos de librerías externas, uno referente a transductores de estados finitos (usando principalmente las librerías OpenFst [20]) para modelar las transiciones entre estados, y otro de álgebra numérica para los cálculos matemáticos. El acceso a las funcionalidades se realiza a través de ejecutables escritos en lenguaje C++, que se agrupan en una serie de *scripts* para construir sistemas completos con mayor facilidad.

El sistema de este proyecto se desarrollará enteramente con Kaldi, aprovechando estas funciones que ya están implementadas actualmente [21]:

- Extracción de características acústicas
- Modelado de contextos fonéticos de todo tipo y tamaño
- Modelado acústico basado en GMM, HMM o técnicas de adaptación a locutores
- Transformadas de tipo lineal y afín
- Construcción de árboles de decisión fonética
- Creación de grafos de decodificación
- Decodificadores

3.2.2.2 *Scripts* del corpus Switchboard

Entre las librerías incluidas en Kaldi, destaca la presencia de diversos *scripts* que sirven de ejemplo para ilustrar la utilización de la herramienta. Estos *scripts* son diferentes en estructura, en función de los diferentes corpus o bases de datos a los que se aplican. Un

corpus particularmente útil en este caso es el Switchboard [22], formado por conversaciones telefónicas bicanales en inglés.

Debido a las similitudes que existen con el corpus que se va a emplear en este proyecto, los *scripts* de Switchboard son de gran utilidad para el desarrollo del sistema. Por tanto, algunas partes del código a desarrollar estarán basadas en estos ficheros, aplicando los necesarios cambios para adaptarlos a la base de datos y a las características fonéticas del castellano.

3.2.2.3 Herramienta SRILM

Stanford Research Institute Language Modelling Toolkit (SRILM) [23] es un conjunto de herramientas para la construcción y utilización de modelos estadísticos de lenguaje. Lleva en desarrollo desde el año 1995 por el laboratorio de las tecnologías del habla de SRI [24] y está formado por:

- Un conjunto de librerías de clases en lenguaje C++ para la implementación de modelos de lenguaje, estructuras de datos y diversas funciones auxiliares.
- Un conjunto de programas ejecutables construidos sobre las librerías anteriores que llevan a cabo tareas como el entrenamiento de los modelos de lenguaje, su evaluación en datos, etiquetado, etc.
- Una colección de diversos *scripts* que ejecutan otras tareas menores.

Su uso está permitido de manera libre y gratuita para agencias gubernamentales, universidades, instituciones académicas y organizaciones sin ánimo de lucro.

3.2.2.4 Transcriptores fonéticos

El tutor de este proyecto, D. Doroteo Torre Toledano, ha proporcionado una herramienta desarrollada por él mismo que realiza la transcripción fonética de las palabras en castellano. Consiste principalmente en un código en lenguaje Perl que acepta como entrada un fichero de texto con todas las palabras del léxico y produce otro con sus respectivas transcripciones fonéticas. De esta manera, cada palabra de entrada se convierte en una serie de fonemas, cada una de ellas separadas por un espacio en blanco.

En el caso de las palabras en inglés, la herramienta utilizada es el diccionario de pronunciación de la Universidad de Carnegie Mellon: “CMU dict” [25]. Se trata de un diccionario de gran vocabulario que contiene casi 125,000 entradas, junto con sus correspondientes transcripciones.

3.3 Preparación de la base de datos

3.3.1 Introducción

La primera fase del entrenamiento consiste en analizar y preparar la base de datos para extraer la información necesaria que se requerirá en las fases posteriores. La Figura 3-2 ilustra las entradas y salidas de esta fase.

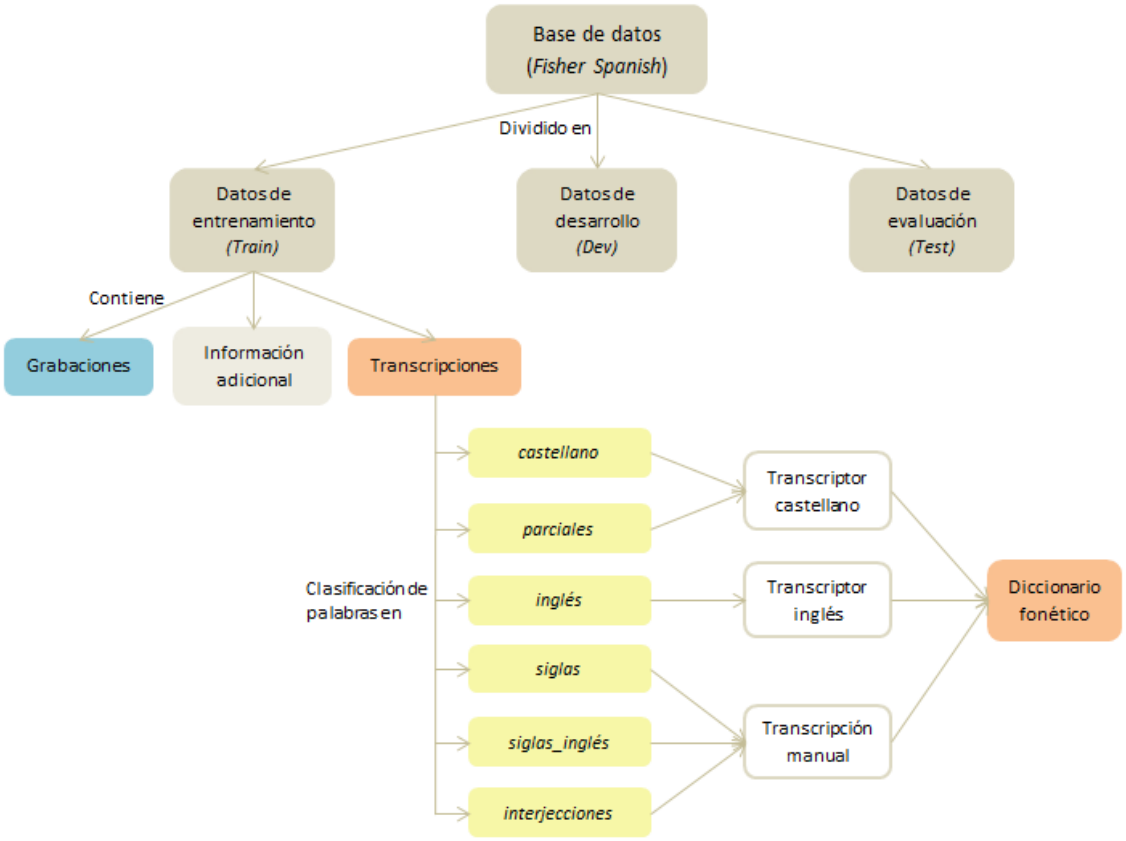

Figura 3-2: Diagrama de bloques de la fase de preparación de la base de datos

Para la realización del proyecto se partirá de una base de datos de conversaciones telefónicas en castellano, concretamente "Fisher Spanish Speech" [26].

Esta base de datos fue realizada por el Consorcio de Datos Lingüísticos (LDC) en Estados Unidos y consiste en aproximadamente 163 horas de conversaciones telefónicas con la participación de 136 locutores de diferentes categorías demográficas, teniendo en cuenta edad, sexo, nacionalidad y formación académica. Con el fin de registrar un mayor rango de vocabulario, los temas tratados por los participantes se seleccionaron arbitrariamente de una lista previamente elaborada, la cual se cambiaba cada 24 horas.

En la colección figuran en total 819 grabaciones diferentes de 10 ó 12 minutos cada una y almacenadas en ficheros con formato NIST Sphere. Cada conversación se compone de dos canales, uno para cada locutor, mientras que la tasa de muestreo empleada es de 8000 muestras por segundo.

Dentro de la base de datos, se proporcionan también ficheros de texto con diferentes informaciones relativas a las grabaciones y a los locutores, como por ejemplo, los números de identificación de cada uno de ellos, qué locutores intervienen en qué grabaciones, etc.

Junto a estas grabaciones se han utilizado también las transcripciones correspondientes a dichas conversaciones y que han sido elaboradas por el mismo consorcio [27]. Éstas se almacenan en ficheros de texto donde en cada línea se incluyen los siguientes elementos: identificador de la conversación, número de canal, marcas de tiempo de inicio y fin, nombre del locutor si se conociera o su sexo en su defecto, número de segmento dentro de la conversación, texto de transcripción e información adicional relevante.

3.3.2 División de la base de datos

Como en cualquier proyecto relacionado con el tratamiento de voz, es necesario utilizar partes diferentes y excluyentes de la base de datos para realizar las distintas funciones a la hora de construir el subsistema. Con este objetivo en mente, se procederá a dividir las grabaciones en las siguientes particiones:

- *Train*: se utilizarán para el entrenamiento inicial del subsistema con la herramienta de reconocimiento de voz Kaldi. Para ello se utilizarán tanto las grabaciones como un diccionario fonético que será elaborado a partir del contenido de las transcripciones.
- *Dev*: se utilizarán para fijar los umbrales del subsistema para mejorar su funcionamiento tras el entrenamiento inicial.
- *Test*: se utilizarán para evaluar el subsistema previamente entrenado.

Para garantizar la no intersección de locutores y/o llamadas entre las grabaciones que quedarán cuadradas en las tres categorías anteriores y teniendo en cuenta el alto número de conversaciones en las que participan algunos locutores (hasta un máximo de 23 y con una media de 12 por locutor), se ha procedido a dividir cada grabación bicanal en dos grabaciones monocanal, conteniendo cada una de ellas únicamente la parte correspondiente a uno de los dos locutores de la grabación original. Esto da como resultado una "nueva" base de datos de 1638 grabaciones, que serán tratadas como independientes a partir de este momento y se distribuirán de la siguiente manera en concreto:

- *Train*: contiene 1348 grabaciones de 112 locutores diferentes (41 masculinos y 71 femeninos), lo cual supone aproximadamente el 80% de la base de datos.
- *Dev*: contiene 146 grabaciones de 12 locutores diferentes (6 masculinos y 6 femeninos), casi el 10% de la base de datos.
- *Test*: contiene 144 grabaciones de 12 locutores diferentes (5 masculinos y 7 femeninos), el 10% restante de la base de datos.

3.3.3 Ecos

Durante el análisis de las grabaciones que se utilizarán para entrenar y a partir de las cuales se elaborarán los ficheros del diccionario, se ha detectado la presencia de ecos en una

pequeña parte de éstas. Estos ecos se producen cuando se filtra la voz de un locutor en el canal correspondiente al otro. Como consecuencia de esto, en una grabación monocanal donde teóricamente debería estar presente la voz de un sólo locutor, aparece también la voz del otro locutor que en un principio no debería. Generalmente, este eco se produce de forma continuada durante todo el tiempo que dura la conversación aunque ocurre a un nivel de volumen sustancialmente menor que la voz del locutor principal.

A través de un análisis preliminar con unos 100 ficheros se ha comprobado la ocurrencia de este fenómeno en 5 de ellos. Al ser este porcentaje un número significativamente pequeño, en principio se procesarán estas grabaciones de forma normal, aunque hay que resaltar lógicamente este hecho de cara a los resultados finales que se obtendrán con el sistema completo.

3.3.4 Elaboración del diccionario fonético

El siguiente paso es elaborar, a partir de las grabaciones y transcripciones de entrenamiento (*Train*), un diccionario fonético compuesto por todo el léxico presente en la base de datos y su transcripción fonética que se utilizará para entrenar el modelo de lenguaje.

3.3.4.1 Descripción de ficheros

Para empezar se analizan las transcripciones que se proporcionan para extraer las palabras, que en función de su clase se almacenan en uno de los seis ficheros siguientes. El conjunto de estas palabras forman lo que se denomina como el "léxico" de la base de datos.

castellano.txt

Todas las palabras en castellano propiamente dichas se incluirán en este fichero, listadas en orden alfabético. En él se incluyen:

- Palabras pertenecientes al léxico normal:

```
árbol
casa
```

- Todas las variantes morfológicas del léxico normal, tales como tiempos verbales, variantes en singular / plural, masculino / femenino o diminutivos:

```
profesor / profesora / profesores / profesoras
pequeñitos
```

- Nombres propios en castellano:

```
Alejandro
Japón
```

- Palabras erróneas o mal pronunciadas:

`preción` (en lugar de presión)

- Extranjerismos recogidos por la Real Academia Española:

```
rock (procedente del inglés)
```

ingles.txt

Al tratarse de una base de datos elaborada en Estados Unidos, y siendo la inmensa mayoría de los participantes personas residentes en dicho país, es comprensible que durante las conversaciones se haya utilizado numerosas palabras en inglés. Dichas palabras se listarán en este fichero e incluirán tanto el léxico normal con sus variantes así como los nombres propios.

```
cold
Seattle
```

parciales.txt

En muchas ocasiones las palabras utilizadas por los locutores no llegan a ser pronunciadas en su totalidad, dando lugar a palabras cortadas o parciales, que serán recogidas en este fichero.

```
expli- (carles)
```

siglas_castellano.txt

Debido a su singularidad, las siglas o acrónimos también se recogerán en un fichero aparte para ser modelados de manera diferente. En este se recogen aquellas que están en castellano.

```
FBI
```

siglas_ingles.txt

Contiene las siglas o acrónimos en inglés debido a que necesitan de un tratamiento diferente.

```
MTV (inglés)
```

interjecciones.txt

Por último, las interjecciones que se utilizan de forma habitual en las conversaciones diarias (hay un total de 25 reconocidas por el LDC para el castellano) también se incluirán en un fichero propio.

```
ay
oh
```

3.3.4.2 Proceso

Para obtener los ficheros anteriormente descritos, se han implementado diversos *scripts* que en su conjunto realizan el proceso completo dividido en las siguientes etapas:

1. Extraer el texto de transcripción de los ficheros proporcionados por el LDC y almacenarlo en un fichero. La información restante, como marcas de tiempo o número de canal, no es necesaria en este caso. Cada línea del fichero resultante contiene un segmento de una conversación cualquiera.
2. Cada segmento de conversación está compuesta por cadenas de caracteres separadas por espacios en blanco. Se divide cada línea en cadenas aisladas de manera que en el fichero resultante cada línea contenga una única cadena, que puede ser una palabra completa o cortada, una etiqueta, etc.
3. Extraer y almacenar cadenas de caracteres que indican palabras en inglés, aprovechando que están incluidas entre etiquetas de la forma *<foreign lang="English"> South </foreign>*. Esto da lugar a dos ficheros, uno con palabras en castellano y otro en inglés.
4. Ciertas palabras están incluidas entre paréntesis, cuando el autor de la trascripción no está completamente seguro de que sean exactamente las pronunciadas por el locutor, debido al bajo volumen de las grabaciones, ruidos de fondo o interferencias. En este caso, se ha decidido tratar estos elementos de forma normal. Por tanto, se extrae las palabras contenidas entre paréntesis para los pasos siguientes.
5. Eliminar los signos de puntuación existentes en las cadenas de caracteres tales como “,”, “.”, “¡”, “!”, “¿”, “?”, etc. a excepción de “-”, que marca el final de una palabra cortada.
6. Eliminar las duplicidades existentes en los dos ficheros.
7. Extraer en un nuevo fichero las cadenas de caracteres que representan palabras cortadas, que se indican al final con un guion “-”.
8. Extraer en un nuevo fichero las interjecciones reconocidas por el LDC.
9. Ordenar alfabéticamente todos los ficheros resultantes, obteniendo de esta manera los ficheros definitivos.

3.3.4.3 Resultados

Los ficheros obtenidos con la aplicación de los pasos anteriores, así como el número de elementos contenidos en cada uno de ellos, son:

- *castellano.txt* contiene casi 25400 entradas
- *inglés.txt* contiene más de 3000 entradas
- *parciales.txt* con casi 2200 elementos
- *interjecciones.txt* con 94 elementos (muchos debidos a una transcripción poco ortodoxa, como por ejemplo `aaahhhhhhh` o `mmmmm`)
- *siglas_castellano.txt* con 85 elementos
- *siglas_ingles.txt* con 92 elementos

3.3.5 Transcripción fonética

El entrenamiento del modelo de lenguaje requiere la descomposición fonética del léxico de entrenamiento para obtener así todos los fonemas presentes, las cuales suponen las unidades gramaticales más básicas. Para ello hay que emplear un transcriptor fonético del castellano que utiliza la correspondencia entre letras y fonemas que se recoge en la Tabla 3-1:

Letra	Fonema	Observación	Letra	Fonema	Observación
á	á		ll	y	
é	é		m	m	
í	í		n	n	
ó	ó		ñ	N	
ú	ú		o	o	
ü	u		p	p	
a	a		q	k	
b	b		r	R	Comienzo de palabra
ch	C		rr	R	
c	T	Delante de [eiéí]	r	r	
c	k		s	s	
d	d		t	t	
e	e		u	-	Detrás de [g] y delante de [ei]
f	f		u	-	Detrás de [q]
g	x	Delante de [eiéí]	u	u	
g	g		v	b	
h			w	u	
i	i		x	ks	
j	x		y	y	Delante o detrás de [aeiouáéíóú]
k	k		y	i	
l	l		z	T	

Tabla 3-1: Correspondencia entre letras y fonemas del castellano para la transcripción fonética

Esta herramienta se ha aplicado a los términos en castellano de la base de datos, tanto los completos como los parciales. Un pequeño ejemplo de la transcripción realizada:

```
abajo → a b a x o
consiguiendo → k o n s i g i e n d o
realizaron → R e a l i T a r o n
```

Para realizar este mismo proceso con los términos en inglés, se ha empleado un diccionario de gran vocabulario denominado “CMU dict” [9], con 125,000 entradas y transcripciones. Para las palabras que no están recogidas en este diccionario, se ha realizado una transcripción manual.

El fichero de salida proporciona la transcripción de las palabras de entrada, pero los fonemas obtenidos son los propios del inglés (en total 39). Es necesario entonces aplicar una conversión adicional a los fonemas del castellano. La Tabla 3-2 detalla la correspondencia utilizada para la conversión:

Fonema inglés	Fonema(s) castellano	Fonema inglés	Fonema(s) castellano
aa	a	l	l
ae	a	m	m
ah	a	n	n
ao	o	ng	ng
aw	au	ow	ou
ay	ai	oy	oi
b	b	p	o
ch	C	r	r
d	d	s	s
dh	d	sh	s
eh	e	t	t
er	er	th	t
ey	ei	uh	u
f	f	uw	u
gg	g	v	b
hh	x	w	u
ih	i	y	i
iy	i	z	s
jh	y	zh	s
k	k		

Tabla 3-2: Conversión de fonemas del inglés a fonemas del castellano

Un pequeño ejemplo de la transcripción realizada:

```
faithful → f e i T f u l
missouri → m i s u r i
```

Por último, las interjecciones y las siglas, tanto en castellano como en inglés, se han transcrito de forma manual, siguiendo las reglas de las dos tablas anteriores. Ejemplo:

```
eh → e
ong (castellano) → o e n e x e
mtv (inglés) → e m t i b i
```

Al finalizar esta etapa, se obtiene por fin un diccionario fonético de carácter definitivo juntando todos los términos presentes en los seis ficheros descritos en la sección 3.3.4.1, así como las transcripciones que les corresponden. El fichero, de nombre

fsp_diccionario.txt, contiene poco más de 30,000 entradas, después de eliminar algunas duplicidades presentes.

3.3.6 Etiquetas

Como en cualquier base de datos compuesta por conversaciones telefónicas, es inevitable que éstas contengan ruidos de fondo en momentos puntuales que puedan alterar la calidad auditiva de las grabaciones, y por consiguiente la del sistema de reconocimiento en su conjunto. Asimismo, existen también otros elementos producidos por los locutores, como las risas o los estornudos, que aunque no supongan información útil propiamente dicha, sí necesitan ser modelados de alguna manera para asegurar que no entorpezcan el reconocimiento de la voz. Estos elementos serán modelados de forma independiente y están debidamente señalados con las etiquetas que se indican a continuación, junto con el número total de veces que aparecen en los archivos analizados:

- <*background*>: ruidos de fondo tales como el televisor, una puerta que se abre o el ladrido de un perro. Aparece en total 8480 veces.
- <*laugh*>: risa. Aparece en total 10957 veces.
- <*breath*>: respiración. Aparece en total 4728 veces.
- <*cough*>: tos. Aparece en total 618 veces.
- <*sneeze*>: estornudo. Aparece en total 16 veces.
- <*lipsmack*>: golpe de labio. Aparece en total 203 veces.

3.4 Modelo de lenguaje del castellano

Una vez analizada y preparada la base de datos en la sección 3.3, la siguiente fase es construir y entrenar un modelo de lenguaje del castellano. Un modelo de lenguaje se puede definir como un mecanismo que permite definir la estructura de un lenguaje, mediante la asignación de probabilidades a secuencias de palabras. El modelo se construye a partir de los datos de entrenamiento (*Train)*, recogiendo las estadísticas de ocurrencia de las diferentes secuencias. En reconocimiento de voz, un modelo de este tipo intentar predecir la siguiente palabra en una secuencia de voz [28].

La Figura 3-3 ilustra la construcción y el entrenamiento del modelo de lenguaje que se desarrollarán en las siguientes secciones.

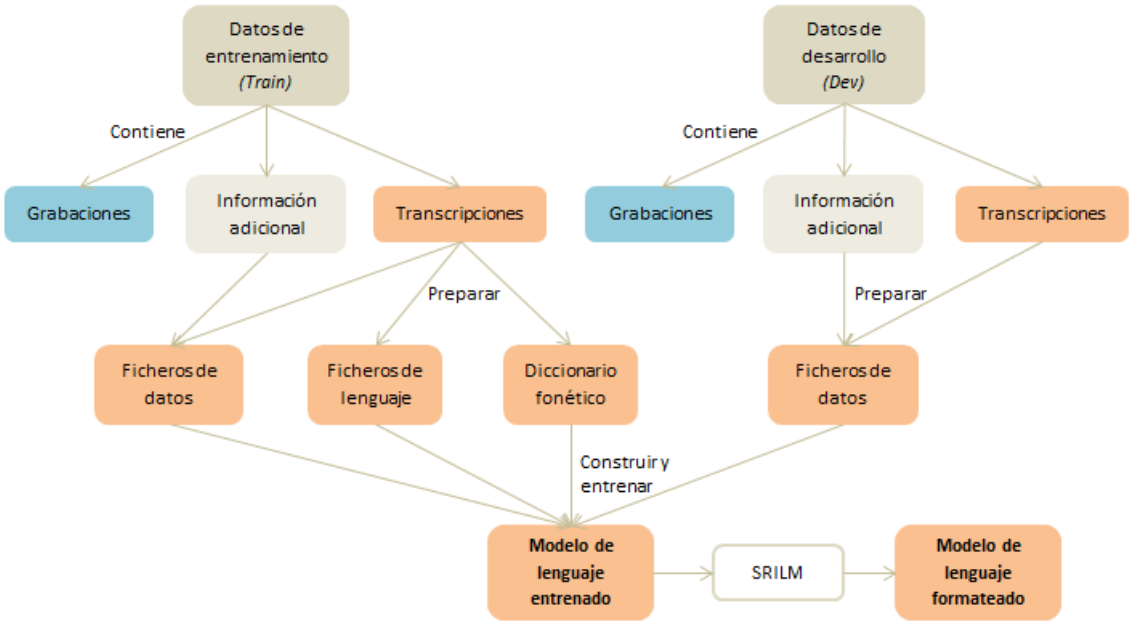

Figura 3-3: Diagrama de bloques de la construcción y entrenamiento del modelo de lenguaje

3.4.1 Ficheros de datos

Para la construcción del modelo de lenguaje es necesario generar una serie de ficheros relacionados principalmente con la estructura de las grabaciones, como la organización de los locutores, la segmentación temporal de las locuciones, etc. Su función principal es establecer de alguna manera relaciones entre todas estas informaciones y facilitar así las tareas de búsqueda o clasificación que ejecutará Kaldi. A la hora de crear estos ficheros, hay que asegurarse de mantener la coherencia entre los contenidos, de forma que los mismos elementos que están presentes en diferentes ficheros sean iguales para no inducir a posibles errores durante el entrenamiento.

Estos ficheros son necesarios para las tres particiones de la base de datos, cada uno de ellos con un uso diferente. Los de entrenamiento (*Train*) para la construcción y entrenamiento del subsistema de reconocimiento, los de desarrollo (*Dev*) para la optimización del modelo de lenguaje y los de evaluación (*Test*) para la evaluación del subsistema.

text

Este fichero contiene dos elementos en cada entrada: <*utterance-id*> y <*text*>. El primero es la identificación de la locución, que puede ser una cadena de texto aleatoria o, como en este caso, basado en información existente sobre locutores y grabaciones para hacer más fácil la organización. El segundo elemento es la propia transcripción de la locución, con presencia de etiquetas cuando corresponda.

```
100315_20050909_210655_26_A_107 si eso es lo que pasa
100853_20051229_182614_906_A_81 muy divertido <laugh> <laugh>
```

segments

Este fichero contiene cuatro elementos en cada entrada: <*utterance-id*>, <*recording-id*>, <*inicio*>, y <*fin*>. El primero es la identificación de la locución; el segundo es la identificación de la grabación a la que pertenece la locución; y los dos últimos corresponden al inicio y al fin de la locución en el tiempo medido en segundos.

```
100315_20050909_210655_26_A_130 20050909_210655_26_A
571.390058972 572.198820556
```

wav.scp

Este fichero contiene dos elementos en cada entrada: <*recording-id*> y <*extended-filename*>. El primero es la identificación de la grabación; el segundo podría ser el nombre real del fichero de la grabación o, como en este caso, el comando a introducir para obtener ese fichero en formato WAV.

```
20050908_182943_22_B
/home/atvs-voz/kaldi-trunk/tools/sph2pipe_v2.5/sph2pipe -f
wav -p ./fisher_spanish/train/audio/20050908_182943_22_B.sph|
```

utt2spk

Este fichero contiene dos elementos en cada entrada: <*utterance-id*> y <*speaker-id*>. El primero es la identificación de la locución mientras que el segundo es la identificación del locutor al que corresponde la locución.

```
100616_20051106_211252_430_A_52 100616
100964_20051123_190439_650_B_10 100964
```

spk2utt (opcional)

La generación y el uso de este fichero es opcional, y contiene dos o más elementos en cada entrada: <*speaker-id*>, <*utterance-id_1*>, <*utterance-id_2*> y sucesivamente. La idea es relacionar al primer elemento, que es la identificación del locutor, con el resto de elementos, que son todas las locuciones que le corresponden.

```
100441 100441_20051001_211346_167_A_1
100441_20051001_211346_167_A_10...
```

spk2gender (opcional)

La generación y el uso de este fichero es opcional, y contiene dos elementos en cada entrada: <*speaker-id*> y <*gender*>. El primer elemento es la identificación del locutor mientras que el segundo es el género del locutor.

```
100573 f
100581 m
```

3.4.2 Construcción

A partir de los ficheros de datos de la partición de entrenamiento (*Train*), es posible construir un modelo de lenguaje inicial que consiste en una serie de ficheros que proporcionan información sobre las palabras que forman el léxico, el conjunto de los fonemas, la topología del modelado HMM y diversa información extra que necesita Kaldi para entrenar el modelo de lenguaje.

phones.txt

Este fichero contiene los fonemas en formatos textual y "entero". Sirve para realizar el mapeo entre los dos elementos y utilizar el formato más conveniente como argumento de entrada en las diferentes funciones ejecutables. En el formato textual, cada fonema presenta diversas variantes teniendo en cuenta el contexto fonético donde aparece: al principio de una palabra (sufijo _B), al final (sufijo _E), en el medio (sufijo _I) o de manera aislada (sufijo _S). Asimismo, las etiquetas mencionadas en la sección 3.3.6 reciben el mismo tratamiento como si fueran fonemas normales y corrientes.

```
background_B 2              e_B 59
background_E 3              e_E 60
background_I 4              e_I 61
background_S 5              e_S 62
```

words.txt

Este fichero contiene todas las palabras del léxico en formatos textual y "entero", con el mismo propósito que en el caso anterior. Se vuelve a incluir las etiquetas en el fichero.

```
<lipsmack> 6
abrirlo 101
rompiste 24807
```

oov.txt

Este fichero incluye solamente una entrada, y se trata de la palabra a la que se mapearán todas las palabras fuera de diccionario (OOV) que el subsistema encuentre durante el entrenamiento.

```
<unk>
```

L.fst / L_disambig.fst

El primer fichero se trata del léxico en formato FST (transductor de estados finitos), con los fonemas como entrada y las palabras como salida. El segundo fichero incluye también, sobre la base del primero, los símbolos de desambiguación.

topo

Este fichero contiene información sobre la topología HMM que se va a utilizar para cada fonema independiente de contexto, especificando los estados HMM y las transiciones entre ellos.

3.4.2.1 Conjunto de fonemas

La información estrictamente relativa al conjunto de los fonemas se encuentra en diferentes ficheros. Muchos de ellos aparecen hasta en tres formatos diferentes aunque contienen la misma información, de los que se utiliza el más conveniente en cada caso según lo requieran las funciones de Kaldi.

- Ficheros *.txt:* en formato textual.
- Ficheros *.int*: en formato "entero".
- Ficheros *.csl*: en formato de lista de elementos separados por ":".

context_indep.txt / **context_indep.int** / **context_indep.csl**

Estos ficheros contienen una lista de fonemas sobre los que se construirán los modelos independientes de contexto, donde no es necesario emplear árboles de decisión para preguntar sobre el contexto. Se trata generalmente de "fonemas no reales", como las etiquetas.

```
laugh_I    /    9    /    1:2:3:4:5:6:7:8:9:10…
laugh_S    /    10   /
```

silence.txt / **silence.int** / **silence.csl**

Estos ficheros contienen la lista de fonemas de silencio. En el caso de este proyecto, son los mismos que en *context_indep.txt*.

```
laugh_I    /    9    /    1:2:3:4:5:6:7:8:9:10…
laugh_S    /    10   /
```

nonsilence.txt / **nonsilence.int** / **nonsilence.csl**

Estos ficheros contienen la lista de fonemas tradicionales que no son de silencio.

```
p_B   /    113   /    110:111:112:113:114…
p_E   /    114   /
```

disambig.txt / **disambig.int** / **disambig.csl**

Estos ficheros contienen los símbolos de desambiguación.

```
#3    /    126   /    124:125:126:127:128…
```

sets.txt / **sets.int**

Estos ficheros contienen conjuntos de fonemas agrupando todas las versiones dependientes de contexto de un mismo fonema.

```
f_B f_E f_I f_S   /   63 64 65 66
```

word_boundary.txt / **word_boundary.int**

Estos ficheros informan sobre la relación existente entre los fonemas y las posiciones que ocupan dentro de las palabras.

```
N_I internal    /     43 internal
N_S singleton   /     44 singleton
```

roots.txt / **roots.int**

Estos ficheros contienen información sobre cómo construir el árbol de decisión. En este caso "shared" y "split" son opciones admitidas por la función de creación de árboles.

```
shared split C_B C_E C_I C_S  /  shared split 31 32 33 34
```

3.4.3 Entrenamiento

El entrenamiento tiene como objetivo mejorar la eficacia del modelo de lenguaje inicial. Para ello se va a utilizar la herramienta SRILM, dada su demostrada eficacia a la hora de trabajar con los modelos *n-gramas.*

Un *n-grama* [29] es un tipo de modelo probabilístico que permite hacer una predicción estadística del próximo elemento de una secuencia de elementos y puede ser definido por una cadena de Markov de orden n-1. Por tanto, un modelo *n-grama* es un tipo de HMM. Las principales ventajas de este modelo son su simplicidad y su facilidad para ampliar el contexto de estudio incrementando el orden *n*.

En reconocimiento de voz, los *n-gramas* se podrían construir en base a fonemas, letras, sílabas o incluso palabras, y el sistema de reconocimiento puede decidir entre varias interpretaciones posibles de lo que ha dicho un locutor en concreto. Por ejemplo, dada una secuencia que empieza por <Esto es posi…> y a partir de un conjunto de datos de aprendizaje, se puede deducir una distribución de probabilidad para el siguiente elemento de esta secuencia: a=0.0003, b=0.5, c=0.0015, etc. donde las probabilidades de todas las posibles letras suman 1.

La medida que se utiliza para evaluar modelos de lenguaje se llama "perplejidad", que se define como 2 elevado a la entropía cruzada [30]. Sin entrar en excesivos detalles matemáticos, se puede decir que la perplejidad indica el número de posibilidades que hay a la hora de predecir el siguiente elemento de una secuencia de fonemas, sílabas o palabras.

Es decir, cuanto menor sea la perplejidad, más eficiente es el modelo de lenguaje en cuestión.

$$ppl = 2^{H(p)} = 2^{-\sum p(x)\, log2\, p(x)},$$

donde H(p) es la entropía de la función de distribución.

En este proyecto, se ha decidido utilizan *n-gramas* de hasta orden 3. Tras llevar a cabo la fase de aprendizaje con los ficheros de datos entrenamiento *(Train)*, se ha utilizado la partición de datos de desarrollo *(Dev)*, que contiene 24.930 locuciones y 146.310 palabras, para evaluar el modelo. Tras varias pruebas y ajustes de parámetros, el mejor resultado de la perplejidad media es de 196,766.

El modelo obtenido tras esta fase tiene un formato interno de SRILM denominado ARPA [31].

3.4.4 Formateado

El modelo de lenguaje en su estado actual, una vez entrenado y optimizado, estaría ya en disposición de ser utilizado en el entrenamiento de los distintos modelos fonéticos. Sin embargo, habría que darle otro formato para que pudiera ser utilizado también a la hora de decodificar esos modelos, dado que Kaldi necesita de un modelo de lenguaje en forma de transductores de estados finitos (formato FST) para construir los grafos de decodificación.

Para ello, se vuelve a utilizar la herramienta SRILM para llevar a cabo la conversión del modelo de lenguaje del formato ARPA al formato FST.

Como resultado de la conversión, los ficheros que componen el modelo de lenguaje son los mismos que los creados en la sección 3.4.2, con un único añadido:

G.fst

Este fichero es la gramática (modelo de lenguaje) en formato FST, donde tanto la entrada como la salida son las palabras.

3.5 Extracción de parámetros característicos

En paralelo al modelo de lenguaje, el otro elemento imprescindible para desarrollar un subsistema de reconocimiento de voz son los modelos fonéticos, aspecto en el que se va a centrar este capítulo a partir de ahora.

Para empezar se procede a extraer ciertas características acústicas de los ficheros disponibles de las grabaciones (es necesario que se conviertan al formato WAV) que servirán de base para construir los modelos fonéticos.

3.5.1 MFCC

Los Mel Frequency Cepstral Coefficients (MFCC) son coeficientes que representan el habla basados en la percepción auditiva humana. Se derivan de la Transformada de Fourier

(TF) o de la Transformada discreta del coseno (DCT) y se diferencia de ellos en que las bandas de frecuencia están representadas en la escala de Mel, de tipo logarítmico. Este tipo de escala es más apropiado para modelar la respuesta auditiva humana que las bandas de frecuencia lineales y permiten un procesado más eficiente de datos.

Kaldi proporciona una función capaz de aplicar de manera directa el siguiente algoritmo para el cálculo de los coeficientes MFCC [32] de una señal:

1. Obtener la Transformada de Fourier cada una de las tramas en que están divididas la señal. En Kaldi las tramas tienes una duración generalmente de 25ms con 10ms de solape entre dos tramas consecutivas.
2. Mapear la energía en cada frecuencia del espectro obtenido sobre la escala de Mel.
3. Tomar el logaritmo de la energía en cada frecuencia de la escala de Mel.
4. Obtener la Transformada discreta del coseno de la secuencia de los algoritmos, como si de una señal se tratara.
5. Las amplitudes del espectro resultante son los coeficientes MFCC.

Es necesario hacer notar que en las siguientes secciones simplemente se mencionarán las funciones de Kaldi que se han utilizado en cada caso, debido a que muchas de ellas se utilizan en diferentes entrenamientos. En el Anexo A se puede encontrar una descripción de lo que realiza cada una de las funciones.

Las funciones de Kaldi utilizadas en este caso son:

```
<extract-segments>
<compute-mfcc-feats>
```

3.5.2 CMVN

En los problemas de clasificación las medias y las varianzas de unos y otros atributos pueden diferir significativamente y hacen que ciertos atributos dominen sobre otros a la hora de realizar determinados cálculos. La normalización consiste en un método de pre-procesamiento que uniformiza estos atributos a media 0 y varianza 1, con lo que se soluciona la problemática anterior y se optimiza el rendimiento de los algoritmos de clasificación.

En el caso de Kaldi se realiza la técnica de normalización denominada Cepstral Mean and Variance Normalization (CMVN) por cada uno de los locutores presentes entre los datos de entrenamiento. Se elimina de esta manera los errores introducidos por las diferencias existentes entra las condiciones de los locutores o de las grabaciones.

La función a utilizar en este caso es:

```
<compute-cmvn-stats>
```

Al finalizar esta etapa, se obtienen las estadísticas CMVN.

3.6 Entrenamiento de los modelos fonéticos en castellano

En esta sección se describen los diferentes tipos de entrenamiento que se van a llevar a cabo para construir los modelos fonéticos. Estos se realizan de forma incremental, dado que cada uno de ellos está basado en el anterior y utiliza alguna técnica nueva que no había sido empleada en ningún modelo anterior. La mecánica en todos ellos es en esencia la misma: alinear el modelo inmediatamente anterior con los datos de entrenamiento (salvo lógicamente en el primer entrenamiento en el que hay que crear un modelo totalmente nuevo) y utilizar después el nuevo algoritmo de entrenamiento. Es de esperar que con cada nuevo entrenamiento mejoren los resultados a la hora de decodificar. La Figura 3-4 ilustra las entradas y salidas de esta fase.

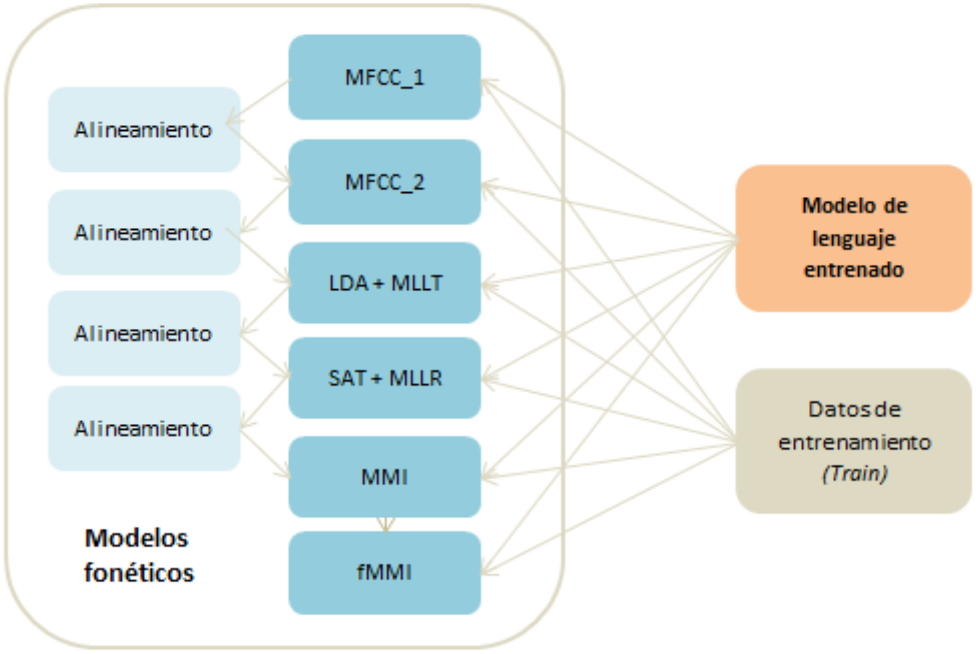

Figura 3-4: Diagrama de bloques del entrenamiento de modelos fonéticos

Kaldi utiliza desde sus inicios los tradicionales modelos GMM a la hora de entrenar los modelos fonéticos, aunque su estructura hace que sea perfectamente extensible a nuevos tipos, como por ejemplo las redes DNN. Los GMM son utilizados como estados de un HMM.

En este punto hay que recordar una vez más que algunos códigos desarrollados para estos entrenamientos están basados en *scripts* existentes del corpus de Switchboard, con los cambios necesarios para adaptarlos a la base de datos utilizada.

3.6.1 MFCC

El primer entrenamiento utiliza como entrada los parámetros característicos MFCC que se han extraído anteriormente y consiste en tres pasos:

1. Entrenamiento de un modelo monofonema, que es independiente del contexto, con características MFCC.
   Las funciones de Kaldi utilizadas son:

```
<apply-cmvn>
<gmm-init-mono>
<compile-train-graphs>
<align-equal-compiled>
<gmm-acc-stats-ali>
<gmm-sum-accs>
<gmm-align-compiled>
<gmm-est>
```

   Entre las diversas opciones configurables se han establecido en 40 el número de iteraciones de entrenamiento a realizar y en 1000 el número de Gaussianas a emplear.

2. Alineamiento del modelo monofonema con los datos de entrenamiento (*Train*) utilizando para ello el modelo de lenguaje.
   Las funciones utilizadas son:

```
<add-deltas>
<compile-train-graphs>
<gmm-align-compiled>
```

3. Entrenamiento de un modelo trifonema, en el que ya se tiene en cuenta el contexto fonético (el fonema anterior y el posterior).
   Las funciones utilizadas son:

```
<acc-tree-stats>
<sum-tree-stats>
<cluster-phones>
<build-tree>
<gmm-init-model>
<gmm-mixup>
<convert-ali>
<compile-train-graphs>
<gmm-align-compiled>
<gmm-acc-stats-ali>
```

Con estos tres pasos estaría terminado el entrenamiento del primer modelo fonético. Para intentar optimizar el rendimiento de las características MFCC, se ha llevado a cabo una nueva iteración de este entrenamiento, pero sin la creación del modelo monofonema ya que esta vez no es necesario empezar de cero.

1. Alineamiento del modelo trifonema con los datos de entrenamiento (*Train*) utilizando para ello el modelo de lenguaje.
2. Entrenamiento del modelo trifonema anterior.

Con esto, se obtiene un nuevo modelo fonético basado en MFCC, previsiblemente más eficiente que el primero.

3.6.2 LDA + MLLT

El siguiente paso es aplicar dos técnicas denominadas Linear Discriminant Analysis (LDA) y Maximum Likelihood Linear Transform (MLLT) sobre la base del modelo con trifonemas.

LDA es un método de transformación lineal que se utiliza para encontrar una combinación lineal de características que definan o separen diferentes clases de objetos o eventos. La combinación resultante podría ser usada para clasificar directamente un objeto en una clase o para reducir la dimensión de la característica antes de la clasificación [33].

Por otro lado, MLLT es una técnica de estimación basada en una transformación lineal de características LDA (como en este caso) o *delta* [34]. En reconocimiento de voz sirve para mejorar la toma de decisión así como proporcionar una pequeña mejora de velocidad a la hora de decodificar.

Este entrenamiento consta de dos fases:

1. Alineamiento del modelo trifonema con características MFCC con los datos de entrenamiento (*Train*) utilizando el modelo de lenguaje.

2. Entrenamiento LDA + MLLT. Las funciones utilizadas en esta fase son:

```
<ali-to-post>
<weight-silence-post>
<acc-lda>
<est-lda>
<acc-tree-stats>
<sum-tree-stats>
<cluster-phones>
<build-tree>
<gmm-init-model>
<convert-ali>
<compile-train-graphs>
<gmm-align-compiled>
<gmm-acc-mllt>
<est-mllt>
<gmm-transform-means>
<compose-transforms>
```

3.6.3 MLLR + SAT

La siguiente técnica a utilizar es un entrenamiento llamado Speaker Adaptive Training (SAT), tras alinear el modelo +LDA+MLLT con Maximum Likelihood Linear Regression (MLLR).

MLLR es una técnica de adaptación al locutor utilizada para adaptar los vectores de medias de las Gaussiana en reconocimiento de voz basado en HMM mediante una transformación afín [35].

Por su parte, SAT es una técnica de entrenamiento adaptado al locutor dado un modelo adaptado por MLLR [36].

Este entrenamiento consta de dos fases:

1. Alineamiento del modelo +LDA+MLLT con los datos de entrenamiento (partición de *Train*) utilizando el modelo de lenguaje. La diferencia con respecto a las anteriores fases de alineamiento es que en esta ocasión se utiliza la técnica MLLR por lo que se utiliza algunas funciones nuevas:

```
<splice-feats>
<transform-feats>
<gmm-post-to-gpost>
<gmm-est-fmllr>
<gmm-est-fmllr-gpost>
```

2. Entrenamiento SAT. Las funciones utilizadas en esta fase son:

```
<splice-feats>
<transform-feats>
<ali-to-post>
<weight-silence-post>
<gmm-est-fmllr>
<acc-tree-stats>
<sum-tree-stats>
<cluster-phones>
<build-tree>
<gmm-init-model>
<convert-ali>
<compile-train-graphs>
<gmm-align-compiled>
<gmm-acc-stats-ali>
<gmm-est>
<gmm-sum-accs>
<gmm-acc-stats-twofeats>
```

3.6.4 MMI

La siguiente técnica de entrenamiento se trata de Maximum Mutual Information (MMI) [37]. La información mutua de dos variables es una cantidad que mide la dependencia mutua de las dos variables, es decir, la reducción de la incertidumbre de una variable aleatoria debido al conocimiento del valor de la otra. El valor máximo de esa información mutua se utiliza como función de coste en el entrenamiento de árboles de decisión.

La principal diferencia con los métodos anteriores es que en este caso no se trata únicamente de que el modelo de cada fonema represente lo más fielmente posible dicho fonema, sino también de que sea capaz de diferenciarse lo más posible de los demás modelos. Por ello se dice que es un entrenamiento discriminativo.

Igual que en los entrenamientos anteriores, éste también consta de dos pasos:

1. Alineamiento del modelo anterior +SAT+MLLR con los datos de entrenamiento (*Train*) utilizando el modelo de lenguaje. Este paso es igual que el de la sección 3.6.3.

2. Entrenamiento MMI. Las funciones de Kaldi utilizadas son:

```
<transform-feats>
<gmm-rescore-lattice>
<lattice-to-post>
<sum-post>
<ali-to-post>
<gmm-acc-stats2>
<gmm-sum-accs>
<gmm-est-gaussians-ebw>
<gmm-est-weights-ebw>
```

3.6.5 fMMI

El último entrenamiento que se lleva a cabo es el feature-space Maximum Mutual Information (fMMI). Dicho de manera simplificada, se trata de aplicar la técnica MMI en un espacio de características [38], que en reconocimiento de voz se define como un espacio abstracto donde cada secuencia de estados se representa como un punto en un espacio de dimensión *n*. Esta dimensión viene determinada por el número de características que describen las secuencias, lo cual permite que objetos similares se sitúen cerca en dicho espacio, facilitando de esta forma la clasificación y el agrupamiento de los objetos.

A diferencia de los entrenamientos anteriores, en este no hace falta llevar a cabo la fase de alineamiento, ya que se reutiliza el del entrenamiento MMI.

Las funciones de Kaldi utilizadas son:

```
<transform-feats>
<fmpe-init>
<gmm-gselect>
<ali-to-post>
<gmm-rescore-lattice>
<lattice-to-post>
<sum-post>
<gmm-fmpe-acc-stats>
<fmpe-sum-accs>
<fmpe-est>
<gmm-sum-accs>
<gmm-est-gaussians-ebw>
<gmm-est-weights-ebw>
```

3.7 Decodificación

La decodificación es una fase que no forma parte del entrenamiento propiamente dicho, pero sirve para examinar los resultados obtenidos tras el proceso. Se utiliza los datos de evaluación previamente separados de la base de datos (partición de *Test*) para evaluar el comportamiento del subsistema desarrollado. En este proyecto, la parte de decodificación se ha llevado a cabo para los seis modelos fonéticos construidos con el fin de verificar que efectivamente con cada nuevo modelo se mejoran los resultados. La Figura 3-5 ilustra esta fase del entrenamiento.

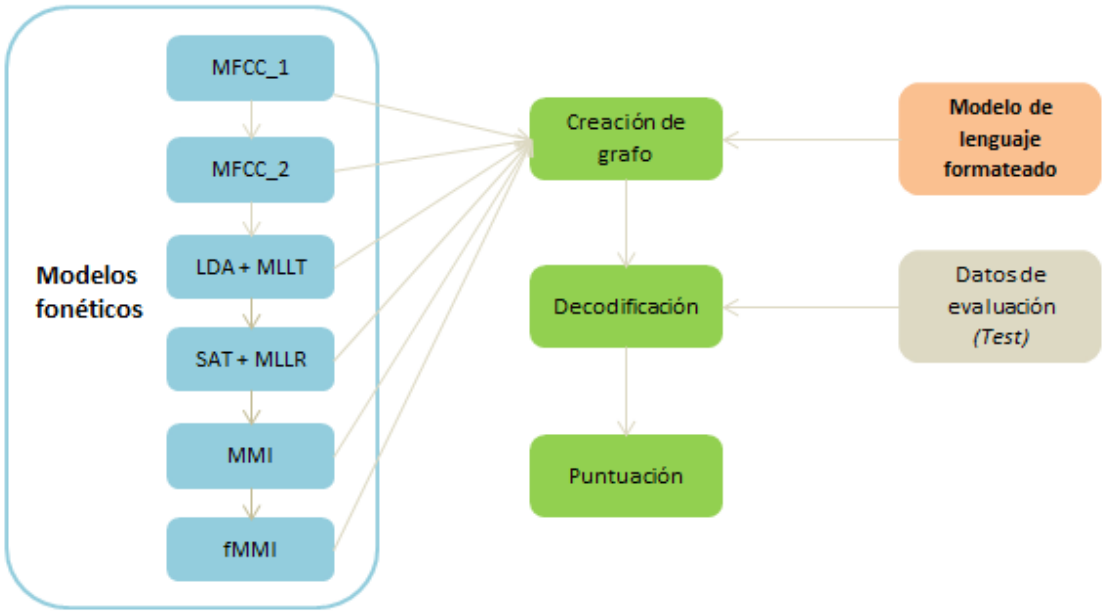

Figura 3-5: Diagrama de bloques de la fase de decodificación

3.7.1 Construcción del grafo de decodificación

En este primer paso se construye un grafo en forma de transductor de estados finitos que recoja todas las características relacionadas con el modelo de lenguaje, el diccionario fonético, la dependencia de contexto y la topología HMM que se han utilizado. Los símbolos de entrada del grafo corresponden a los estados HMM dependientes de contexto y los de salida corresponden a las palabras que se obtendrán tras la fase de decodificación que tendrá lugar después. El proceso detallado se puede encontrar en [39].

Las funciones de Kaldi utilizadas son:

```
<fsttablecompose>
<fstdeterminizestar>
<fstminimizeencoded>
<fstisstochastic>
<make-h-transducer>
<fstrmsymbols>
<addselfloops>
```

3.7.2 Decodificación del grafo

En este segundo paso se procede a decodificar el grafo. Los decodificadores que emplea Kaldi están basados en el "algoritmo de Viterbi" [40]. Se trata, en líneas generales, de una técnica recursiva para estimar la secuencia de estados de un proceso de Markov de estado finito y tiempo discreto, generando como resultado la secuencia más probable, que en este caso corresponde a una hipótesis de fonemas y palabras.

En Kaldi se puede utilizar una variedad de decodificadores, desde algunos muy simples hasta otros más optimizados, aunque más costosos. Los decodificadores son utilizables con cualquier tipo de modelo fonético que acepta Kaldi, aunque ciertas técnicas como MLLR requieren alguna función adicional. Para cada decodificador existen además dos variantes, en función de la extracción o no de *lattices* durante el proceso. Como en este caso los *lattices* serán necesarios para el reconocimiento de palabras clave, se utilizará siempre la variante que soporta la extracción.

Las funciones de Kaldi utilizadas son:

```
<transform-feats>
<gmm-latgen-faster>
<lattice-to-post>
<weight-silence-post>
<gmm-post-to-gpost>
<gmm-est-fmllr>
<gmm-est-fmllr-gpost>
<lattice-determinize-pruned>
<compose-transforms>
<gmm-rescore-lattice>
<gmm-gselect>
```

3.7.3 Puntuación

El último paso de la decodificación consiste en puntuar las hipótesis de palabra obtenidas. El subsistema realiza transcripciones del audio de evaluación en base a los modelos fonéticos y de lenguaje entrenados y las compara con las secuencias de palabras transcritas manualmente (o de referencia).

Las funciones de Kaldi utilizadas son:

```
<lattice-best-path>
<compute-wer>
```

4 Entrenamiento del subsistema de reconocimiento de palabras clave

4.1 Introducción

Este capítulo se centra en el desarrollo del subsistema de reconocimiento de palabras clave.

Para empezar se explicará la preparación de los datos necesarios a la entrada del subsistema. Las dos siguientes secciones versan sobre las dos fases que forman el entrenamiento del subsistema: indexado y búsqueda. La última parte del capítulo detalla la evaluación de la búsqueda y los informes que se generan.

En la Figura 4-1 se puede observar un esquema general del proceso de entrenamiento, así como los ficheros a la entrada y a la salida.

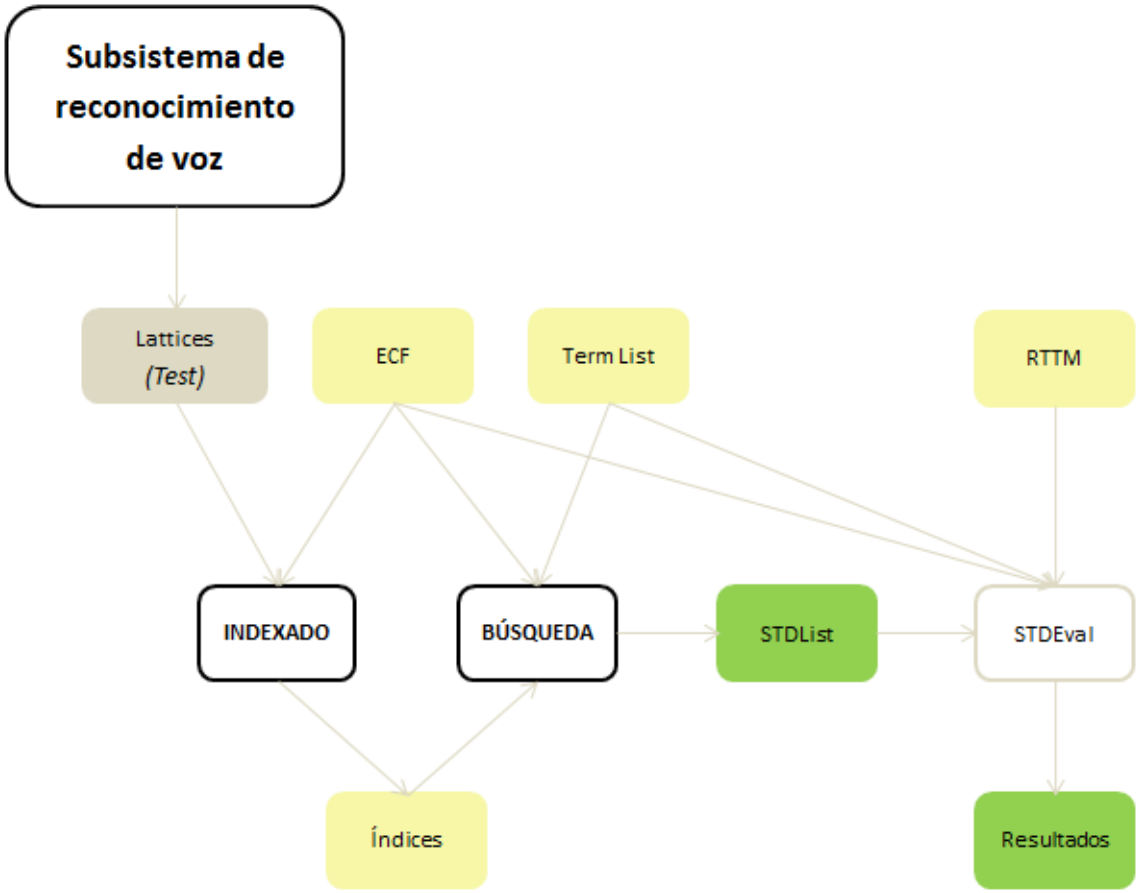

Figura 4-1: Diagrama de bloques del entrenamiento del subsistema de reconocimiento de palabras clave

4.2 Ficheros de datos

Aparte de los *lattices* extraídos durante la decodificación del subsistema de reconocimiento de voz (sección 3.7.2), el subsistema de reconocimiento de palabras clave necesita a su entrada los tres ficheros que se detallan a continuación.

4.2.1 Fichero ECF

ECF es un fichero de texto en formato XML que especifica diferentes aspectos relacionados con los segmentos de audio que serán indexados y buscados en la evaluación.

El nodo principal "ecf" contiene los siguientes atributos:

- *source_signal_duration:* la duración total, en segundos, de todo el audio de evaluación utilizado.
- *version:* identificador del fichero
- *language:* lengua del audio utilizado

Los datos acerca de cada uno de los segmentos de audio se recogen en nodos independientes llamados "excerpt", que se encuentran dentro del nodo principal y contienen:

- *audio_filename:* nombre del fichero utilizado
- *source_type:* el tipo de audio puede ser "cts" (conversaciones telefónicas), "bnews" (noticias) o "confmtg" (reuniones o conferencias)
- *channel:* canal del fichero de audio utilizado
- *tbeg:* instante de comienzo del fichero de audio a procesar
- *dur:* duración del fichero de audio utilizado

4.2.2 Fichero TermList

TermList es un fichero en formato XML que contiene la lista de los términos cuyas ocurrencias debe buscar el subsistema. Generalmente, por término se entiende una secuencia de entre una y cinco palabras [41] [42]. Cada término se identifica con un código alfanumérico que sirve para su rastreo durante todo el proceso.

Los términos están definidos exclusivamente por su representación ortográfica. A la hora de evaluar el subsistema se tendrá en cuenta únicamente la transcripción textual de los ficheros de audio, sin considerar el significado semántico de los términos. De este modo, la palabra <lista> (adjetivo en femenino) es una coincidencia válida con <lista> (sustantivo). Por el contrario, se requiere que la ocurrencia detectada sea una coincidencia exacta con el término a buscar, de manera que <lista> no es una coincidencia válida con <listado>.

El nodo principal "termlist" contiene una lista de nodos "term", además de los siguientes atributos:

- *ecf_filename:* nombre del fichero ECF utilizado y que queda asociado de esta manera

- *version:* identificador del fichero
- *language:* lengua del audio utilizado

Cada nodo "term" contiene un único atributo y un sub-nodo para definir de forma unívoca el término a buscar:

- *termid*: cadena alfanumérica que identifica el término
- *termtext*: sub-nodo con la representación ortográfica del término

4.2.3 Fichero RTTM

RTTM es un fichero que representa la transcripción textual de los ficheros de audio y se utiliza como referencia a la hora de evaluar las ocurrencias detectadas tras la fase de búsqueda. Si una ocurrencia detectada aparece en este fichero, coincide con la representación ortográfica del término y tiene lugar durante el espacio temporal exigido (con unos determinados márgenes de error), entonces será considerada como ocurrencia correcta.

El fichero contiene un conjunto de registros de los diferentes objetos que hay en la transcripción, cada uno de los cuales está compuesto por nueve atributos. Por norma general, un objeto se refiere a cualquier término (lexema) que aparece en el audio o un fragmento donde no aparece ningún término.

- *type:* el tipo de objeto puede ser "segmento", "lexema", "no-lexema", o "metadato"
- *file:* nombre del fichero de audio utilizado
- *chnl:* canal del fichero de audio utilizado
- *tbeg:* instante de comienzo del objeto
- *tdur:* duración del objeto, en segundos
- *stype:* el subtipo de objeto puede ser "lexema", "acrónimo", "interjección", "risa", "tos", "nombre propio", "ruido", "música", etc.
- *ortho:* representación ortográfica del objeto
- *name:* nombre del locutor si se conoce
- *conf:* probabilidad de que la información del objeto sea correcta

4.3 Indexado

En esta primera fase del reconocimiento de palabras clave, el subsistema debe procesar los datos de audio sin conocimiento previo de los términos a buscar (requisito fundamental de la tarea STD). Sólo es necesario realizar esta fase una única vez para cada conjunto de datos.

En reconocimiento de palabras clave, el índice que se utiliza como referencia para detectar las ocurrencias de los términos se obtiene a partir del resultado del proceso de reconocimiento de voz que se ha llevado a cabo previamente, el cual es inherentemente

inexacto (con su tasa de error medida en WER). Para compensar esta deficiencia, se utiliza la técnica de indexación de *lattices*, basada en la asignación de una cierta probabilidad al hecho de si un fragmento de audio ocupa o no una posición determinada dentro del fichero de audio. El proceso detallado, en el cual está basado el algoritmo implementado en Kaldi, se puede encontrar en [43].

En la sección 3.7.2 se describe el proceso de generación de los llamados *lattices* de los datos de evaluación durante la decodificación de audios por el subsistema de reconocimiento de voz. La indexación, desde un punto de vista técnico, consiste en transformar los *lattices* de todas las locuciones desde su forma inicial en transductores de estado finito a una estructura única de transductor de factor generalizado donde el instante de comienzo, el instante de finalización y la probabilidad de cada símbolo de palabra se almacenan como un coste tridimensional. En realidad, este transductor de factor generalizado [44] es un índice en orden inverso de todas las secuencias de palabras contenidas en los *lattices.* Debido a la presencia de la información temporal, el uso de este tipo de estructura es idóneo para la tarea de STD.

Una vez completado el indexado, el audio original no se vuelve a utilizar durante la fase de búsqueda.

Las funciones de Kaldi utilizadas son:

```
<lattice-align-words>
<lattice-scale>
<lattice-to-kws-index>
<kws-index-union>
```

4.4 Búsqueda

En la fase de búsqueda, el subsistema debe detectar las ocurrencias de cada uno de los términos de la lista en los índices obtenidos en la fase anterior. Esta fase se lleva a cabo cada vez que se quiera buscar uno o varios términos.

El sistema de búsqueda que se ha empleado se basa en palabras, de modo que en principio sólo sería capaz de encontrar las palabras que se encontrasen en el vocabulario del reconocedor. En el caso de que la palabra no se encuentre dentro del vocabulario (OOV), en lugar de recurrir a sistemas basados en lattices e índices fonéticos o en general de sub-unidades de palabra, se recurre a un método denominado de palabras “proxy” en el que se buscan palabras del vocabulario próximas fonéticamente a las palabra OOV, y se sustituye la búsqueda de la palabra OOV por la búsqueda de dichas palabras “proxy”. Este método ha demostrado tener una precisión similar a la de las búsquedas basadas en unidades inferiores a la palabra, con la ventaja de no requerir una nueva decodificación, lattices e índices basados en subunidades de la palabra. De este modo, es posible buscar cualquier tipo de término, como nombres propios, extranjerismos, siglas o palabras mal escritas aunque no sean ortodoxas desde el punto de vista gramatical.

Se utiliza el diccionario fonético para cada palabra que forma el término completo. Las palabras que no están presentes en el diccionario se representan fonéticamente por similitudes con otras palabras, teniendo en cuenta su ortografía. Esto se consigue con sistemas estadísticos de conversión grafema-fonema que se entrenan con diccionarios fonéticos amplios.

Después de analizar todas las palabras, comienza la búsqueda propiamente dicha. Se escanea él índice elaborado para encontrar las secuencias de palabras más probables que correspondan a las cadenas de palabras de los términos. Un algoritmo probabilístico realiza la comparación entre ambas secuencias generando una puntuación de confianza.

Técnicamente, para cada término se crea una máquina de estados finitos que lo acepta como entrada y lo compone con el transductor de factor generalizado para obtener todas las ocurrencias del término en los datos de evaluación, junto con la identificación de las locuciones, el instante de comienzo, el instante de finalización y la puntuación que representa la probabilidad de cada ocurrencia.

Todas estas ocurrencias se someten a una decisión de tipo binario (SÍ o NO) comparando su puntuación con un umbral previamente definido [45]. Esto significa que el usuario tiene la opción de establecer el umbral que considere oportuno para optimizar la tasa de detecciones y la de falsas alarmas.

La función utilizada en este caso es:

```
<kws-search>
```

4.4.1 Fichero STDList

Como resultado de esta fase, se obtiene un fichero en formato XML que proporciona los resultados de la búsqueda de términos y los tiempos de ejecución de esta fase. Está compuesto por tres nodos organizados jerárquicamente.

El nodo principal “stdlist” registra las entradas al sistema y los parámetros usados para generar los resultados y contiene cinco atributos:

- *termlist_filename:* nombre del fichero TermList utilizado
- *indexing_time:* el tiempo de indexación es la cantidad de segundos que se ha tardado en procesar todo el audio y producir el índice correspondiente con un único CPU.
- *index_size:* el tamaño, en bytes, del índice construido durante fase de indexado
- *language:* lengua del audio utilizado
- *system_id:* cadena de texto para identificar al sistema utilizado

Dentro del nodo principal se encuentra un conjunto de nodos “detected_termlist”. Cada uno de ellos contiene las salidas del subsistema para un único término. Los tres atributos de cada nodo son:

- *termid:* cadena que identifica al término, tal y como aparece en el fichero TermList

- *term_search_time:* cantidad de tiempo, en segundos, utilizado por un único CPU en encontrar todas las ocurrencias de este término en concreto
- *oov_term_count:* el número de palabras que son fuera de vocabulario (OOV) para el sistema.

Cada nodo "detected_termlist" engloba a su vez una serie de nodos "term", que contiene la localización y la puntuación de cada ocurrencia detectada del término. Los seis atributos son:

- *file:* nombre del fichero de audio utilizado, tal y como aparece en el fichero ECF
- *channel:* canal del fichero de audio donde se ha encontrado el término
- *tbeg:* instante de comienzo del término expresado en segundos
- *dur:* duración del término en segundos
- *score:* puntuación que indica la verosimilitud del término detectado
- *decisión:* decisión binaria (YES o NO) sobre si el término debería haber sido detectado en función de un umbral predefinido

4.5 Evaluación

En esta fase final se examina el rendimiento del subsistema, cuyos resultados de salida se encuentran en el fichero STDList.

Para llevar a cabo esta tarea, el instituto NIST proporciona una herramienta diseñada expresamente para la evaluación de STD: STDEval [46], compuesta por una serie de *scripts* en el lenguaje Perl. Se necesita a la entrada, además del fichero STDList, los ficheros ECF, TermList y RTTM, éste último como referencia para comprobar si las ocurrencias de los términos han sido detectadas de manera correcta.

Entre los informes que la herramienta genera a la salida, los más importantes son:

- *score.occ.txt*: fichero de texto que recoge todas las estadísticas de la evaluación de la búsqueda, tanto globales como particularizadas para cada término. Estas estadísticas engloban:
    - Número de ocurrencias reales
    - Número de ocurrencias correctamente detectadas por el subsistema
    - Número de falsas alarmas (ocurrencias incorrectamente detectadas)
    - Número de falsos rechazos o pérdidas (ocurrencias correctas no detectadas)
    - Probabilidad de falsas alarmas
    - Probabilidad de falsos rechazos
    - Term-Weighted Value (TWV): medida de la calidad de la búsqueda que tiene en cuenta las falsas alarmas y los falsos rechazos de todos los términos. Se da en dos modalidades: Actual TWV (ATWV) con el umbral seleccionado para las detecciones binarias y Maximum TWV (MTWV), con el umbral óptimo a posteriori.

- *score.ali.txt*: fichero de texto que lista todas las ocurrencias para cada término en cada fichero de audio. Aparecen tanto las coincidencias reales existentes en el audio como las detectadas por el subsistema, que serán iguales en caso de que la ocurrencia haya sido encontrada de manera correcta.
- *score.det.png*: imagen con la curva Detection Error Trade-off (DET) que corresponde a la búsqueda. Una curva DET es una representación gráfica donde se contraponen la tasa de falsas alarmas frente a la tasa de falsos rechazos y sirve para ilustrar el funcionamiento general del subsistema.
- *score.cache*: fichero de texto que contiene las coincidencias encontradas en el fichero RTTM de referencia para cada término.

5 Pruebas y resultados

5.1 Introducción

En este capítulo se describe las pruebas realizadas para medir el rendimiento del sistema desarrollado. Como es de esperar, los dos subsistemas se someterán a pruebas diferentes, con medidas de calidad diferentes para evaluar los resultados. Se proporciona en ambos casos una explicación de la medida de calidad, una breve descripción de los datos utilizados en la evaluación y se presentan los mejores resultados que se han obtenido.

5.2 Reconocimiento de voz

5.2.1 Medida de calidad

A la hora de valorar un sistema de reconocimiento de voz, la principal medida es Word Error Rate (WER). Consiste en el porcentaje de palabras incorrectas que se han obtenido con la decodificación, incluyendo inserciones, borrados y sustituciones de una palabra por otra, con respecto al número total de palabras sometidas a la evaluación.

$$WER = \frac{Nº\ Inser\ +\ Nº\ Borr\ +\ Nº\ Sust}{Nº\ Palabras}$$

5.2.2 Subsistema inicial

5.2.2.1 Datos de evaluación

Los datos contenidos en la partición de *Test*, obtenido tras la preparación de la base de datos de Fisher Spanish (ver Sección 3.2), serán utilizados para evaluar el subsistema de reconocimiento de voz. Algunos puntos interesantes de estos datos son:

- La partición está formada por 144 grabaciones, aproximadamente el 10% de la base original.
- Los ficheros contienen un total de 1.020.000 segundos de audio
- En total hay 145.816 palabras

5.2.2.2 Resultados

En la Tabla 5-1 se muestran los resultados obtenidos con el subsistema, con los datos de evaluación de Fisher Spanish.

Modelo fonético	% WER
MFCC_1	61,30
MFCC_2	61,18
+LDA + MLLT	56,98
+MLLR + SAT	52,07
+MMI	50,27
+fMMI	49,88

Tabla 5-1: Resultados del subsistema de reconocimiento de voz desarrollado

Para proporcionar una mejor idea de los resultados obtenidos, se ofrece a continuación una comparación con el estado del arte actual en reconocimiento de voz en castellano con Fisher Spanish [47]. La Tabla 5-2 recoge las características de los datos de evaluación y los resultados de ambos sistemas.

	Subsistema desarrollado	Estado del arte
Nº ficheros	144	50
Nº palabras	145.816	47.896
% WER	49,88	36,50

Tabla 5-2: Comparativa entre el subsistema de reconocimiento de voz desarrollado y el estado del arte actual

5.2.3 Subsistema modificado

Tal y como se ha mencionado en la sección 1.2, el sistema inicial ha sido modificado con la incorporación de nuevos datos para poder participar en la evaluación "ALBAYZIN 2014 Search on Speech". Esta modificación afecta únicamente al primer subsistema, en forma de un nuevo modelo de lenguaje y nuevos modelos fonéticos que han pasado por la consiguiente fase de entrenamiento. El segundo subsistema no sufre cambios y funciona de la misma manera que en el sistema inicial.

Para comprobar mejor el efecto de estos cambios, se ha optado por un enfoque incremental a la hora de introducir los nuevos modelos, en lugar de utilizarlos de manera directa y conjunta. Se han implementado tres versiones diferentes del subsistema de reconocimiento de voz, que han sido probados con los mismos datos y en las mismas condiciones para comparar su funcionamiento:

1. Mod_1: subsistema con modelo de lenguaje y modelos fonéticos iniciales.
2. Mod_2: subsistema con modelo de lenguaje modificado y modelos fonéticos iniciales.
3. Mod_3: subsistema con modelo de lenguaje y modelos fonéticos modificados.

5.2.3.1 Datos de evaluación

La base de datos utilizada en la evaluación "ALBAYZIN 2014 Search on Speech" pertenece al corpus MAVIR [48]. Se tratan de una serie de ficheros de audio monocanal en lengua castellana, con formato PCM y a una frecuencia de muestro de 16KHz. El contenido de cada fichero puede pertenecer a una conversación telefónica, noticias o reuniones/conferencias.

Para el entrenamiento y desarrollo del subsistema la organización de la evaluación ha proporcionado 7 ficheros con unas 5 horas de duración en total. De estos 7 ficheros se ha decidido emplear 5 para el entrenamiento, que junto con los datos de Fisher Spanish han servido para construir los nuevos modelos de lenguaje y fonéticos. La preparación de estos datos se ha realizado de la misma manera que la descrita en las secciones 3.3.4, 3.3.5 y 3.4.1.

Los 2 ficheros restantes, que contienen 20.548 palabras y 6.757 segundos de audio, han sido utilizados para las pruebas.

5.2.3.2 Resultados

En la Tabla 5-3 se muestran los resultados obtenidos con las tres versiones del subsistema modificado.

	% WER		
M. fonético	**Mod_1**	**Mod_2**	**Mod_3**
MFCC_1	99.21	82.02	79.49
MFCC_2	99.04	81.97	78.94
+LDA + MLLT	98.69	80.69	75.68
+MLLR + SAT	98.97	81.23	75.82
+MMI	98.65	80.61	74.59
+fMMI	99.09	83.15	76.69

Tabla 5-3: Resultados del subsistema de reconocimiento de voz modificado, en sus tres versiones, con el corpus MAVIR

5.3 Búsqueda de palabras clave

Las pruebas realizadas con el segundo subsistema tienen como finalidad preparar el sistema para participar en la evaluación "ALBAYZIN 2014 Search on Speech", motivo por el cual los *lattices* a utilizar son los extraídos del subsistema Mod_3, cuyos resultados en las pruebas de reconocimiento de voz son los mejores de las tres versiones modificadas.

5.3.1 Medida de calidad

Para evaluar el rendimiento de un sistema de reconocimiento de palabras clave existen multitud de medidas de calidad, entre los que destacan dos por encima del resto:

- Occurrence-Weighted Value (OWC) [41] se calcula sumando un valor positivo por cada detección correcta y restando un valor por cada detección incorrecta. Todas las detecciones contribuyen de igual manera al resultado final.
- Term-Weighted Value (TWV) [41] se calcula hallando primero las probabilidades de falsa alarma y falso rechazo para cada término por separado, que son usadas posteriormente para calcular valores específicos de cada término y por último promediando estos valores específicos de todos los términos.

$$Value(\theta) = 1 - promedio\,\{Prob_{F.Rechazo}(término, \theta) + \beta * Prob_{FAlarma}(término, \theta)\},$$

donde θ es el umbral de decisión.

Dado que OWV suele tener una mayor varianza, situación que se da cuando un pequeño número de términos ocurren con mucha frecuencia, se ha decidido utilizar TWV como medida de calidad por ser menos susceptible a este fenómeno y por ser un promedio de todos los términos. Para ser exactos, en estas pruebas se ha utilizado MTWV, que se define como el valor máximo de TWV que se alcanza ajustando el umbral de decisión.

5.3.2 Datos de evaluación

Los datos de evaluación utilizados en esta prueba son, una vez más, los pertenecientes al corpus MAVIR. En este caso se mantiene la división de ficheros realizada anteriormente, de manera que el subsistema realizará la búsqueda de términos en dos ficheros, con un total de 20.548 palabras y 6.757 segundos de audio.

La organización de la evaluación ha proporcionado asimismo los ficheros ECF, TermList y RTTM correspondientes a los datos de entrenamiento y desarrollo. Concretamente, la lista de términos contiene 346 entradas a buscar por el subsistema.

5.3.3 Resultados

En la Tabla 5-4 se muestran los resultados obtenidos con el subsistema. Los *lattices* utilizados por el subsistema provienen de la versión del subsistema de reconocimiento de voz que mejores resultados ha ofrecido, es decir, de la versión Mod_3.

Modelo fonético	MTWV
MFCC_1	0.0971
MFCC_2	0.1160
+LDA + MLLT	0.1863
+MLLR + SAT	0.1603
+MMI	0.1845
+fMMI	0.1664

Tabla 5-4: Resultados del subsistema de reconocimiento de palabras clave

5.4 Evaluación "ALBAYZIN 2014 Search on Speech"

Por último, hay que mencionar una vez más que el sistema desarrollado en este proyecto va a participar en la evaluación STD de "ALBAYZIN 2014 Search on Speech", en el contexto de la conferencia "IberSPEECH 2014, VII Jornadas en Tecnologías del Habla", en colaboración con el grupo de investigación GEINTRA [49], de la Universidad de Alcalá de Henares.

En el sistema utilizado para la evaluación final, los modelos de lenguaje y fonéticos serán construidos con los datos del corpus Fisher Spanish y los 7 ficheros de entrenamiento/desarrollo de MAVIR. Concretamente se van a entregar los resultados obtenidos con dos configuraciones distintas del sistema:

- Sistema primario con la configuración que mejor MTWV ha obtenido durante el entrenamiento: +LDA+MLLT.
- Sistema contrastivo con la configuración que mejor WER ha obtenido durante el entrenamiento: +MMI.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Tras analizar los resultados de las pruebas, tanto con el sistema inicial como con la versión modificada para la participación en la evaluación "ALBAYZIN 2014 Search on Speech", es posible extraer una serie de conclusiones acerca del sistema de reconocimiento de palabras clave desarrollado en este proyecto y de la herramienta Kaldi:

- Kaldi ha demostrado ser una herramienta altamente útil a la hora de desarrollar cualquier sistema relacionado con el reconocimiento automático de voz. Gracias a su potencia y versatilidad, es fácilmente adaptable a diferentes condiciones de trabajo y bases de datos sin que eso produzca una alteración muy grande en el rendimiento de los sistemas.
- La base de datos que se utiliza en reconocimiento de voz tiene una importancia capital. Este hecho, que es perfectamente intuitivo y entendible sobre el papel, queda una vez más demostrado con los resultados de las pruebas. Mientras que la versión inicial del sistema presentaba unos resultados muy aceptables (por debajo de 50% WER con el mejor modelo), el simple hecho de que sea evaluado con una base de datos diferente hace que los resultados sean exageradamente peores (en torno a 99% WER con el sistema Mod_1). Esto se debe a la suma de diversas razones, como las condiciones en las que se realizan las grabaciones de audio, el número de locutores en cada fichero, la temática del audio provocando el uso de vocabulario muy específicos o la calidad de las transcripciones textuales.
- Se ha demostrado que la importancia del modelo de lenguaje es relativamente mayor a la de los modelos fonéticos. De esta manera, con la única actualización del modelo de lenguaje y sin modificar los modelos fonéticos (paso del sistema Mod_1 a Mod_2) se ha obtenido una mejora cercana a 20% WER, mientras que uso de modelos fonéticos nuevos (paso del sistema Mod_2 a Mod_3) ha resultado en una mejora menor, de aproximadamente 5% WER. Eso es debido principalmente a que los modelos fonéticos construidos con los datos de Fisher Spanish son bastante completos dado la gran cantidad de audio contenido, con lo cual el cambio de la base de datos ha tenido más efectos en relación a la calidad del modelo de lenguaje.

6.2 Trabajo futuro

Tras analizar los resultados de las pruebas y presentar las conclusiones en la sección anterior, se propone una serie de posibles mejoras a implementar en trabajos futuros basados en este sistema:

- Una base de datos mejor preparada, en el sentido de no contener tanto vocabulario en otra lengua distinta, debería mejorar los resultados obtenidos. En el caso de Fisher Spanish, al estar elaborada en Estados Unidos y por locutores residentes en

el mismo país, es inevitable el uso inconsciente de palabras, expresiones e incluso frases enteras en inglés. Esto trae como consecuencia distorsiones que afectan negativamente a la hora de modelar y entrenar los modelos de lenguaje y fonéticos, dando lugar a una peor prestación del sistema.

- La utilización de redes neuronales a la hora de construir modelos fonéticos debería traer consigo mejoras bastante significativas en los resultados. Además, esta técnica es cada vez más factible en Kaldi dado que este es uno de los principales puntos de desarrollo de la herramienta en estos momentos, con la implementación de diferentes funciones relacionadas y ejemplos de demostración.

Referencias

[1] National Institute of Standards and Technology (NIST), "NIST: Open Keyword Search 2013 Evaluation (OpenKWS13)", Julio 2013, http://www.nist.gov/itl/iad/mig/openkws13.cfm

[2] http://iberspeech2014.ulpgc.es/index.php/albayzin/search-on-speech-evaluation

[3] http://iberspeech2014.ulpgc.es/

[4] C. García, y D. Tapias, "La Frecuencia Fundamental de la Voz y sus Efectos en Reconocimiento de Habla Continua", *Procesamiento del Lenguaje Natural*, vol. 26, pp. 163-168, Septiembre 2000.

[5] http://neivis-viveelmundodelafonoaudiologa.blogspot.com.es/2010/05/aparato-fonador.html

[6] Antonio Quilis, "Tratado de fonética y fonología españolas", Gredos, 1993, ISBN 9788424922474.

[7] M. Branza, y J. L. Suau, "Nociones de fonética y fonología del español", *University of Bucharest*, Febrero 2013.

[8] C. Godin, y P. Lockwood, "DTW Schemes for Continuous Speech Recognition: An Unified View", *Computer Speech and Language*, vol. 3, pp. 169-198, Abril 1989.

[9] F. K. Soong, A. E. Rosenberg, B. H. Juang, y L. R. Rabiner, "Report: A Vector Quantization Approach to Speaker Recognition", *AT&T Technical Journal*, vol.66, pp.14-26, Marzo 1987.

[10] R. P. Lippmann, "Review of Neural Networks for Speech Recognition", *Neural Computation*, vol. 1, pp. 1-38, Marzo 1989.

[11] Li Deng, y Dong Yu, "Deep Learning: Methods and Applications", *Now Publishers*, 2004.

[12] G. Hilton, Li Deng, Dong Yu, y G.E. Dahl, "Deep Neural Networks for Acoustic Modeling in Speech Recognition", *IEEE Signal Processing Magazine*, vol. 29, pp.82-97, Octubre 2012.

[13] D. Povey, L. Burget, M. Agarwal, y P. Akyazi, "Subspace Gaussian Mixture Models for Speech Recognition", *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 4330-4333, Marzo 2010.

[14] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE*, vol. 77, pp.275-286, Febrero 1989.

[15] L. R. Rabiner, y B.H. Huang, "Hidden Markov Models for Speech Recognition", *Technometrics*, vol. 33, pp. 251-272, Agosto 1991.

[16] Mark Gales, y Steve Young, **"**The application of Hidden Markov Models in Speech Recognition"**,** *Foundations and Trends in Signal Processing*, vol.1, pp. 195-304, Enero 2008.

[17] Javier Tejedor, Doroteo Torre, y José Colás, "Estado de arte en Wordspotting aplicado a los sistemas de extracción de información en contenidos de voz", *I Congreso Español de Recuperación de Información*, Junio 2010.

[18] Grupo de investigación Área de Tratamiento de Voz y Señales: http://atvs.ii.uam.es/

[19] http://kaldi.sourceforge.net/

[20] http://www.openfst.org/twiki/bin/view/FST/WebHome

[21] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, y Karel Vesely, "The Kaldi Speech Recognition Toolkit", *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, Diciembre 2011.

[22] Corpus Switchboard. Disponible en Linguistic Data Consortium con referencia LDC97S62. https://catalog.ldc.upenn.edu/LDC97S62

[23] A. Stolcke, J. Zheng, W. Wang, y V. Abrash, "SRILM at Sixteen: Update and Outlook", *IEEE Automatic Speech Recognition and Understanding Workshop,* Hawaii, Diciembre 2011.

[24] http://www.speech.sri.com/projects/srilm/

[25] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[26] Corpus Fisher Spanish. Disponible en Linguistic Data Consortium con referencia LDC97S62: https://catalog.ldc.upenn.edu/LDC2010S01

[27] Transcripciones de Fisher Spanish. Disponible en Linguistic Data Consortium con referencia LDC2010S01: https://catalog.ldc.upenn.edu/LDC2010T04

[28] Craig Trim, "What is a Language Model". Disponible en: http://trimc-nlp.blogspot.com.es/2013/04/language-modeling.html

[29] Christopher D. Manning, y Hinrich Schütze, "Foundations of Statistical Natural Language Processing", MIT Press, 1999, ISBN 0-262-13360-1.

[30] F. Jelinek, R. L. Mercer, L. R. Bahl, y J. K. Baker, "Perplexity, a measure of the difficulty of speech recognition tasks", *Journal of Acoustic Society of America,* vol.62, 1977.

[31] http://www.speech.sri.com/projects/srilm/manpages/ngram-format.5.html

[32] Min Xu, Ling-Yu Duan, Jianfei Cai, Liang-Tien Chia, Changsheng Xu, y Qi Tian, "HMM-Based Audio Keyword Generation", *Advances in Multimedia Information Processing – PCM 2004,* vol. 3, pp. 566-574, 2004.

[33] S. Balakrishnama, y A. Ganapathiraju, "Linear Discriminant Analysis – A Brief Tutorial", Institute for Signal and Information Processing, Mississippi State University.

[34] Mark Gales, "Semi-tied Covariance Matrices for Hidden Markov Models", *IEEE Transactions on Speech and Audio Processing,* vol. 7, pp. 272-281, 1999.

[35] Mark Gales, "Maximum Likelihood Linear Transformations for HMM-based Speech Recognition", *Computer Speech and Language,* vol. 12, pp. 75-98, Mayo 1997.

[36] T. Anastasakos, J. McDonough, y J. Makhoul, "Speaker Adaptive Training: A Maximum Likelihood Approach to Speaker Normalization", *IEEE International*

Conference on Acoustics, Speech and Signal Processing, vol.2, pp.1043-1046, Abril 1997.

[37] L. Bahl, P. Brown, P. de Souza, y R. Mercer, "Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition", *1986 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP),* pp. 49-64, Abril 1986.

[38] R. Hsiao, y T. Schultz, "Generalized Discriminative Feature Transformation for Speech Recognition", *InterSpeech,* 2009

[39] M. Mohri, F. Pereira, y M. Riley, "Speech Recognition with Weighted Finite State Transducers", *Springer Handbook of Speech Processing,* pp. 559-584, 2008, ISBN 978-3-540-49125-5.

[40] G. D. Forney, "The Viterbi Algorithm", *Proceedings of the IEEE,* vol. 61, pp. 268-278, Marzo 1973.

[41] National Institute of Standards and Technology (NIST), "NIST Spoken Term Detection (STD) 2006 evaluation plan". Disponible en http://www.itl.nist.gov/iad/mig/tests/std/2006/

[42] National Institute of Standards and Technology (NIST), "NIST: Open Keyword Search 2014 Evaluation (OpenKWS14)". Disponible en http://www.nist.gov/itl/iad/mig/openkws14.cfm

[43] D. Can, y M. Saraçlar, "Lattice Indexing for Spoken Term Detection", *IEEE Transactions on Audio, Speech and Language Processing,* vol. 19, pp.2338-2347, Noviembre 2011.

[44] M. Mohri, P. Moreno, y E. Weinstein, "General Suffix Automaton Construction Algorithm and Space Bounds" *Theoretical Computer Science,* vol. 410, pp. 3553-3562, Septiembre 2009.

[45] D. Miller, M. Kleber, C. L. Kao, O. Kimball, T. Colthurst, S. Lowe, R. Schwartz, y H. Gish, "Rapid and Accurate Spoken Term Detection", *InterSpeech 2007,* Agosto 2007.

[46] http://www.itl.nist.gov/iad/mig/tests/std/tools/

[47] M. Post, G. Kumar, A. López, D. Karakos, C. Callison-Burch, y S. Khudanpur, "Improved Speech-to-Text Translation with the Fisher and Callhome Spanish–English Speech Translation Corpus", *IWSLT 2013, International Workshop on Spoken Language Translation,* Diciembre 2013.

[48] Corpus MAVIR. Disponible en: http://www.lllf.uam.es/ESP/CorpusMavir.html

[49] Grupo de Ingeniería Electrónica Aplicada a Espacios Inteligentes y Transporte: http://www.geintra-uah.org/

Glosario

Alófono: cada uno de los fonos o sonidos que en un idioma dado se reconoce como un determinado fonema y corresponde a una determinada forma acústica.

ATWV: Actual Term Weighted Value – Valor real ponderado por término. Medida de calidad de sistemas de STD que se obtiene teniendo en cuenta las probabilidades de falsa alarma y falso rechazo de todos los términos. Es igual al MTWV en caso de que el umbral de decisión del sistema sea el óptimo.

CMVN: Cepstral Mean and Variance Normalization - Normalización de la media y la varianza cepstrales. Método de pre-procesamiento que uniformiza una función de distribución a media 0 y varianza 1 que en reconocimiento de voz sirve para eliminar los errores introducidos por los canales de las grabaciones.

C++: Lenguaje de programación de propósito general y orientado a objetos.

Diccionario fonético: lista de palabras en orden alfabético con sus respectivas transcripciones fonéticas.

DNN: Deep Neural Network – Red neuronal profunda. Tipo de red neuronal en el que existen al menos dos capa de "neuronas" entre las capas de entrada y salida que aumenta la capacidad de modelado de la red.

DTW: Dynamic Time Warping - Alineamiento temporal dinámico. Algoritmo utilizado en procesamiento de señales que mide la similitud entre dos secuencias que pueden variar en tiempo o en velocidad a través del alineamiento temporal de los parámetros.

ECF: Experiment Control File – Fichero de control de experimento. Fichero de texto que especifica características relacionadas con los ficheros de audio que serán indexados y buscados por un sistema de STD.

Falsa alarma: detección de una ocurrencia incorrecta por un sistema de STD.

Falso rechazo o pérdida: no detección de una ocurrencia correcta por un sistema de STD.

Fonema: unidad fonológica básica de una lengua que se caracteriza por ser diferenciadora, indivisible y abstracta.

FST: Finite State Transducer – Transductor de estados finitos. Máquina de estados finitos que acepta un conjunto de cadenas a la entrada y genera un conjunto de cadenas a la salida en base a su mecanismo de funcionamiento interno.

GMM: Gaussian Mixture Model – Modelo de mezclas de Gaussianas. Modelo probabilístico basado en la mezcla de funciones de distribución Gaussianas que sirve para representar la presencia de sub-conjuntos dentro de un conjunto.

HMM: Hidden Markov Model – Modelo oculto de Markov. Modelo estadístico donde se asume que el sistema a modelar es un proceso de Markov con estados no observables (ocultos) y cada uno de ellos tiene una distribución de probabilidades sobre las posibles salidas.

Kaldi: conjunto de herramientas de reconocimiento automático de voz, de carácter libre y gratuito y disponible bajo la licencia *Apache.*

Keyword Spotting: Reconocimiento de palabras clave. Campo de investigación de reconocimiento de voz que trata sobre la localización de palabras clave en audio.

Lattice: grafo acíclico dirigido que representa la secuencia de estados más probable en un modelo estadístico; en reconocimiento de voz representa la secuencia de estados, fonemas o palabras más probable tras decodificar los datos de entrada.

LDA: Linear Discriminant Analysis – Análisis discriminante lineal. Método utilizado en reconocimiento de patrones y de señales para encontrar combinaciones lineales de las características que diferencien a dos o más clases de objetos.

MFCC: Mel Frequency Cepstral Coefficients – Coeficientes cepstrales en las frecuencias de Mel. Coeficientes que representan el habla basados en la percepción auditiva humana en una escala de frecuencias logarítmica.

MLLT: Maximum Likelihood Linear Transform – Transformada lineal por máxima verosimilitud. Técnica de estimación de parámetros de un modelo estadístico basada en transformación lineal de sus características.

MLLR: Maximum Likelihood Linear Regression – Regresión lineal por máxima verosimilitud. Técnica de adaptación al locutor utilizada en reconocimiento de voz para adaptar los parámetros mediante una transformada afín.

MMI: Maximum Mutual Information – Máxima información mutua. Técnica de entrenamiento discriminativo basada en la dependencia mutua de dos variables que en reconocimiento de voz sirve para diferenciar lo máximo posible un modelo acústico de otro.

Modelo de lenguaje: modelo estadístico que refleja la estructura de una lengua, mediante la asignación de probabilidades a secuencias de unidades gramaticales de acuerdo al número de sus ocurrencias en la fase de entrenamiento.

Modelo fonético: modelo estadístico que permite representar las características fonéticas de audio de voz, tanto la parte referida al canal acústico como la de los locutores.

NIST: National Institute of Standards and Technology. Organismo federal no regulador de los Estados Unidos cuya función es desarrollar y fomentar medidas, estándares y tecnología para aumentar la productividad y la seguridad económica y mejorar la calidad de vida.

NN: Neural Network – Red neuronal. Modelo computacional utilizado en reconocimiento de patrones que se inspira en el sistema nervioso central de los animales y donde los sistemas se presentan como interconexiones de "neuronas" capaces de calcular valores a partir de los datos de entrada.

OOV: Out-Of-Vocabulary. Palabra fuera de vocabulario, generalmente nombres propios, extranjerismos o acrónimos.

OpenFST: librería para la construcción, combinación, optimización y búsqueda de transductores de estados finitos.

Perl: Practical Extraction and Report Language. Lenguaje de programación de alto nivel, propósito general e interpretado por un intérprete de comandos que toma características del lenguaje C.

RTTM: Rich Text Transcription Mark. Fichero con transcripción textual de ficheros de audio utilizada como referencia a la hora de evaluar ocurrencias detectadas en un sistema de STD.

SAT: Speaker Adaptive Training – Entrenamiento adaptado a locutor. Técnica de entrenamiento utilizada en reconocimiento de voz para maximizar la probabilidad de los datos de entrenamiento para los diferentes locutores.

Script: fichero de órdenes o procesamiento por lotes en forma de archivo de texto plano que es ejecutado por un intérprete de comandos.

SRI: Stanford Research Institute. Organismo americano sin ánimo de lucro dedicado a la investigación y aplicación de ciencia y tecnología en proyectos patrocinados por agencias gubernamentales, compañías comerciales y fundaciones privadas.

SRILM: Stanford Research Institute Language Modelling toolkit. Herramienta desarrollada por SRI dedicada a la construcción y aplicación de modelos estadísticos de lenguaje para su uso en reconocimiento de voz.

STD: Spoken Term Detection - Detección de términos orales. Técnica de reconocimiento de palabras clave donde el audio es procesado sin conocimiento previo de los términos a buscar.

Transcriptor fonético: Herramienta que convierte automáticamente una palabra en los fonemas que la componen, siguiendo las noemas fonológicas-fonéticas de una determinada lengua.

VQ: Vectorial Quantification - Cuantificación vectorial. Técnica de cuantificación utilizada en procesamiento de señales para modelar funciones de densidad de probabilidad de una señal a través de la distribución de vectores prototipo en un espacio vectorial de dimensión *n*.

Anexos

A. Funciones de Kaldi

- **<acc-lda>:** acumular estadísticas LDA basadas en funciones de densidad de probabilidad de un modelo alineado utilizando los posteriores
- **<acc-tree-stats>:** acumular estadísticas para construir un árbol con contexto fonético, a partir de los fonemas independientes del contexto del modelo de lenguaje y las características MFCC normalizadas
- **<add-deltas>:** añadir coeficientes cepstrales del tipo *delta* a las características MFCC. Los coeficientes *delta* son estimaciones de las derivadas de los coeficientes MFCC con el fin de aportar información sobre el comportamiento dinámico de éstos dado que los MFCC sólo contienen información estática.
- **<add-self-loops>:** añadir bucles y probabilidades de transición a un transductor de estados finitos
- **<align-equal-compiled>**: construir una base de alineamiento igualmente espaciada para iniciar el entrenamiento de un árbol
- **<ali-to-post>:** convertir los alineamientos a posteriores. Los posteriores forman una función de distribución de probabilidad de una variable condicionada por las evidencias obtenidas en un experimento. El nombre de "posterior" indica que ya se ha tenido en cuenta las evidencias relevantes al caso particular que se está estudiando.
- **<apply-cmvn>:** efectuar una normalización CMVN con las estadísticas previamente calculadas
- **<build-tree>:** entrenar un árbol de decisión a partir de las estadísticas acumuladas y el modelo de lenguaje
- **<cluster-phones>:** agrupar los fonemas (o los conjuntos de fonemas) del modelo de lenguaje
- **<compile-train-graphs>:** crear los grafos de entrenamiento a partir de un modelo básico inicial y del léxico en forma de transductor de estados finitos
- **<compose-transforms>:** componer dos transformadas, es decir, multiplicar dos matrices de transformación
- **<compute-cmvn-stats>:** calcular las estadísticas de la normalización CMVN indexadas por locutores (podría ser también por locuciones) utilizando las características MFCC y los ficheros que relacionan locutores y locuciones
- **<compute-mfcc-stats>:** calcular los coeficientes MFCC a partir de ficheros con señales de audio
- **<compute-wer>:** calcular estadísticas WER comparando diferentes transcripciones en formato textual o "entero"

- **<convert-ali>**: convertir los alineamientos de un modelo o árbol para usarlos con otro modelo o árbol
- **<est-lda>:** estimar una matriz de transformación LDA usando las estadísticas previamente acumuladas
- **<est-mllt>:** estimar una matriz de transformación MLLT usando las estadísticas previamente acumuladas
- **<extract-segments>:** extraer segmentos de un fichero grande de audio en formato *WAV*
- **<fmpe-est>:** realizar una iteración de entrenamiento en transformadas fMPE
- **<fmpe-init>:** inicializar una transformada fMPE a cero
- **<fmpe-sum-accs>:** sumar estadísticas fMPE previamente acumuladas
- **<fstdeterminizestar>:** eliminar los símbolos *épsilon* de un transductor de estados finitos y realiza la "determinización". Un símbolo *épsilon* se introduce inicialmente para representar el comienzo y el fin de una secuencia al construir el transductor, por tanto, no se tratan de símbolos reales. Una "determinización" de un transductor consiste en construir otro transductor determinista equivalente al inicial, con un único estado inicial y tal ningún par de transiciones salientes de un mismo estado compartan la misma etiqueta de entrada al siguiente estado.
- **<fstisstochastic>:** comprobar si un transductor de estados finitos es estocástico (no determinista) y calcular el error máximo
- **<fstminimizeencoded>:** minimizar el valor de los pesos de un transductor de estados finitos ponderado después de la codificación
- **<fstrmsymbols>:** reemplazar un subconjunto de símbolos de un transductor de estados finitos con símbolos *épsilon*
- **<fsttablecompose>:** realizar la composición de dos transductores de estados finitos
- **<gmm-acc-mllt>:** acumular estadísticas MLLT a partir del modelo LDA entrenado y características MFCC normalizadas
- **<gmm-acc-stats>:** acumular estadísticas a partir de posteriores para entrenamiento de modelos basados en GMM
- **<gmm-acc-stats-ali>:** acumular estadísticas a partir de alineamientos para el entrenamiento de modelos basados en GMM
- **<gmm-acc-stats-twofeats>:** acumular estadísticas para el entrenamiento de modelos basados en GMM, con la particularidad de calcular posteriores de un conjunto de características pero acumular estadísticas de otro conjunto de características diferentes
- **<gmm-align-compiled>:** alinear el modelo con las características MFCC normalizadas.
- **<gmm-est>:** realizar la re-estimación por máxima verosimilitud de modelos basados en GMM

- **<gmm-est-fmllr>:** estimar las transformadas MLLR por locutores (podría ser también por locuciones) a partir de un modelo fonético ya entrenado, de las estadísticas de entrenamiento y de posteriores
- **<gmm-est-fmllr-gpost>:** estimar las transformadas MLLR por locutores (podría ser también por locuciones) a partir de un modelo fonético ya entrenado, de las estadísticas de entrenamiento y de posteriores extraídos de funciones Gaussianas.
- **<gmm-est-gaussians-ebw>:** aplicar el algoritmo "Extended Baum-Welch" a un entrenamiento discriminativo de tipo MMI o MPE
- **<gmm-est-weights-ebw>:** aplicar el algoritmo "Extended Baum-Welch" a los pesos de un entrenamiento discriminativo de tipo MMI o MPE
- **<gmm-fmpe-acc-stats>:** acumular estadísticas para el entrenamiento fMPE de modelos basados en GMM
- **<gmm-gselect>:** calcular los *N* mejores índices de Gaussianas para la poda de un árbol de entrenamiento
- **<gmm-init-model>:** inicializar un modelo basado en GMM a partir del árbol de decisión y las estadísticas acumuladas
- **<gmm-init-mono>:** inicializar un modelo monofonema basado en GMM con la información existente sobre la topología HMM y la dimensión de las características MFCC
- **<gmm-latgen-faster>:** generar *lattices* de un conjunto de datos utilizando modelos basados en GMM
- **<gmm-mixup>:** mezclar dos modelos de tipo GMM
- **<gmm-post-to-gpost>**: convertir posteriores a nivel de estado a posteriores a nivel de funciones Gaussianas
- **<gmm-rescore-lattice>:** reemplazar las puntuaciones acústicas de *lattices* utilizando un modelo diferente
- **<gmm-sum-accs>:** sumar las estadísticas acumuladas para el entrenamiento de modelos basados en GMM
- **<gmm-transform-means>:** aplicar transformada lineal o afín a modelos basados en GMM
- **<kws-index-union>:** unir *lattices* previamente indexados, llevando a cabo la eliminación de símbolos *épsilon,* la "determinización" y la minimización
- **<kws-search>:** buscar palabras clave en los índices
- **<lattice-align-words>:** convertir *lattices* de manera que sus arcos queden alineados con los fonemas o palabras a los que representan
- **<lattice-best-path>:** generar el mejor camino a través de los *lattices,* en forma de transcripciones y alineamientos
- **<lattice-determinize-pruned>:** realizar una "determinización" de *lattices,* conservando únicamente el mejor camino para cada secuencia de entrada

- **<lattice-scale>:** aplicar un escalado a los pesos de *lattices*
- **<lattice-to-kws-index>:** crear un índice invertido de los *lattices* de entrada
- **<lattice-to-post>:** extraer posteriores a partir de *lattices* con el "algoritmo Forward-Backward"
- **<make-h-transducer>:** construir un transductor de estados finitos sin bucles que acepta probabilidades de transición a la entrada y genera fonemas dependientes del contexto a la salida
- **<splice-feats>:** mezclar las características de una trama concreta de la señal con las de las tramas del contexto
- **<sum-post>:** sumar dos conjuntos de posteriores para cada locución
- **<sum-tree-stats>:** sumar las estadísticas previamente acumuladas para construir un árbol con contexto fonético
- **<transform-feats>:** aplicar una transformada a las características MFCC
- **<weight-silence-post>:** ponderar los fonemas de silencio en los posteriores

B. Presupuesto

1) Ejecución Material

- Compra de ordenador personal ..... 1.200 €
- Material de oficina ..... 150 €
- Total de ejecución material ..... 1.350 €

2) Gastos generales

- 16 % sobre Ejecución Material ..... 216 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material ..... 81 €

4) Honorarios Proyecto

- 700 horas a 15 € / hora ..... 10500 €

5) Material fungible

- Gastos de impresión ..... 60 €
- Encuadernación ..... 40 €

6) Subtotal del presupuesto

- Subtotal Presupuesto ..... 12147 €

7) I.V.A. aplicable

- 21% Subtotal Presupuesto ..... 2550,87 €

8) Total presupuesto

- Total Presupuesto ..... 14697,87 €

Madrid, Octubre de 2014

El Ingeniero Jefe de Proyecto

Fdo.: Junchen Xu

Ingeniero de Telecomunicación

C. Publicación

Junchen Xu, Doroteo T. Toledano, Javier Tejedor

<The ATVS-GEINTRA STD System for ALBAYZIN 2014 Search-on-Speech Evaluation>

Enviado a IberSPEECH 2014, "VIII Jornadas en Tecnologías del Habla"

Las Palmas de Gran Canaria, Spain, Nov. 2014

The ATVS-GEINTRA STD System for ALBAYZIN 2014 Search-on-Speech Evaluation

Junchen Xu[1], Doroteo T. Toledano[1], and Javier Tejedor [2]

[1] ATVS-UAM, Escuela Politécnica Superior, Universidad Autónoma de Madrid,
Calle Francisco Tomás y Valiente, 11; 28049 Madrid, Spain
junchen.xu@estudiante.uam.es; doroteo.torre@uam.es

[2] GEINTRA, Universidad de Alcalá, Madrid, Spain
javier.tejedor@depeca.uah.es

Abstract. This paper describes the system developed in a joint effort by ATVS-UAM and GEINTRA-UAH for the ALBAYZIN 2014 Search-on-Speech Evaluation. Among the four different modalities of the evaluation, we have decided to participate only in the Spoken Term Detection (STD) Evaluation. Our system employs an Automatic Speech Recognition (ASR) subsystem to produce word lattices and a Spoken Term Detection (STD) subsystem to retrieve potential occurrences. Kaldi toolkit has been used both for building the ASR subsystem and the STD subsystem. The Fisher Spanish Corpus has been used for training the ASR subsystem. In order to adapt both the acoustic and the language models to the task, the development data provided by the organizers have been added to the Fisher Spanish corpus. Our best ASR result on Fisher Spanish corpus is about 50% Word Error Rate (WER), and about 75% WER on a small part of the development data provided by the organizers. Our best STD result on this part of the development data is an ATWV of 0.1863.

Keywords: Spoken Term Detection, Keyword Spotting, Search on Speech, Automatic Speech Recognition.

1. Introduction

The increasing volume of speech information stored in audio and video repositories motivates the development of automatic audio indexing and spoken document retrieval systems. Spoken Term Detection (STD), defined by NIST as 'searching vast, heterogeneous audio archives for occurrences of spoken terms' [9] is a fundamental block of those systems, and significant research has been conducted on this task [1, 5, 6, 10, 13, 14, 15, 16].

This paper presents the ATVS-GEINTRA STD system submitted to the ALBAYZIN 2014 Search-on-Speech Spoken Term Detection (STD) Evaluation. It is a collaborative work of the ATVS research group from Universidad Autónoma de Madrid and GEINTRA research group from Universidad de Alcalá. Most of the work was conducted by a student (Junchen Xu) under the supervision of the other authors as part of his end of studies project of Telecommunications Engineering.

The submission involves an automatic speech recognition (ASR) subsystem, and an STD subsystem. The ASR subsystem converts input speech signals into word lattices, and the STD subsystem integrates a term detector which searches for putative occurrences of query terms, and a decision maker which decides whether detections are reliable enough to be considered as hits or should be rejected as false alarms.

The ASR subsystem is based on Gaussian mixture models (GMM) and was built using the Kaldi toolkit [12]. The training process largely followed the Switchboard s5 recipe, adapted to use the Fisher Spanish corpus [3] and the training/development materials provided by the organizers. The same tool was used to conduct decoding and produce word lattices.

In previous works [14, 15] we used a proprietary STD subsystem employing an n-gram reverse indexing approach [7] to achieve fast term search. This approach indexed word/phone n-grams retrieved from lattices, and term search was implemented as retrieving n-gram fragments of a query term. Then, the confidence score of a hypothesized detection was computed as the averaged lattice-based score of the n-grams of the detection.

For this evaluation, our goal was to compare our proprietary STD subsystem to the STD subsystem recently provided as part of the Kaldi toolkit [12]. However, due to insufficient time and resources during the evaluation, we were only able to produce scores/detections with the STD subsystem of Kaldi, leaving the interesting comparison for future, post-evaluation work.

We have finally submitted two systems, ATVS-GEINTRA_STD_pri and ATVS-GEINTRA_STD_con1. Both are almost the same system with the only difference that the primary system is designed to optimize the Actual Term Weighted Value (ATWV) while the second system is designed to optimize the Word Error Rate (WER) in the ASR subsystem.

The rest of the paper is organized as follows: Section 2 presents the details of our primary system, including the system description and the detailed description of the database used. Section 3 highlights the differences between the primary and the contrastive systems. Finally, Section 4 provides conclusions and future research directions.

2. Primary System: ATVS-GEINTRA_STD_pri

Our submission involves an ASR subsystem and an STD subsystem, both based on Kaldi. Figure 1 shows our system architecture. Training was conducted using the Fisher Spanish corpus [3] and the training/development data provided by the organizers. The primary system was designed to optimize the ATWV on the training/development data provided by the organizers.

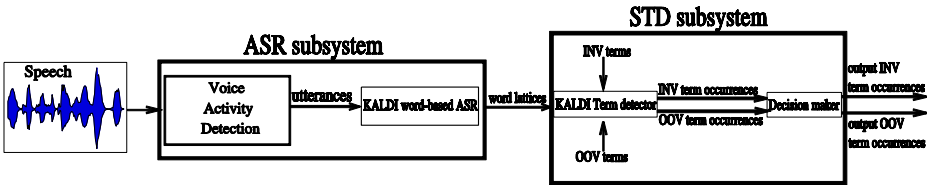

Fig. 1. STD system architecture.

2.1. System description

This section will describe the ASR and the STD subsystems in sequence. Instead of resorting to a hybrid approach using a word-based system to deal with in-vocabulary (INV) terms and a phone-based system to treat out-of-vocabulary (OOV) terms, as in previous works [14,15], we only used the method implemented in Kaldi to deal with OOV words. This method is based on proxy words and consists of substituting the OOV term to search by acoustically similar INV words (proxy words) and searching for these proxy words instead. This method allows dealing with OOV words without having to build two different ASR modules (word-based and subword-based) and correspondingly two different sets of lattices and indices. Details can be found in [12]. Our primary goal with this system was to compare it to our previous hybrid approach [14,15], but we did not have time to do that comparison by the evaluation deadline.

Automatic Speech Recognition Subsystem

The Kaldi toolkit [12] was used to build the ASR subsystem, and we largely follow the Switchboard s5 recipe, except some minor changes in the configurations. Specifically, the acoustic features are 13-dimensional Mel-frequency cepstral coefficients (MFCCs), with cepstral mean and variance normalization (CMVN) applied to mitigate channel effects. We build two context-dependent phonetic acoustic models working directly on MFCCs, corresponding to two training iterations (we refer to these models as MFCC_1 and MFCC_2). The normalized MFCC features then pass a splicer which augments each frame by its left and right 4 neighboring frames. A linear discriminant analysis (LDA) is then employed to reduce the feature dimension to 40, and a maximum likelihood linear transform (MLLT) is applied to match the diagonal assumption in the GMM acoustic modeling. The model trained on these new features is denoted as +LDA+MLLT in the rest of the paper. After this model, the maximum likelihood linear regression (MLLR) and the speaker adaptive training (SAT) techniques are applied to improve model robustness. This model will be referred as +MLLR+SAT. Then a discriminative training approach based on the maximum mutual information (MMI) criterion is adopted to produce better models. Finally, the feature-space based maximum mutual information (fMMI) technique is applied to build the final models.

Based on the acoustic models, a word-based ASR system was built for searching INV terms. OOV terms were searched in the word lattices using the proxy words method implemented in Kaldi. The system uses a 3-gram word-based LM.

An energy-based voice activity detection (VAD) implemented in SoX is used to segment speech signals into utterances. Some heuristics of utterance and silence duration are employed to constraint the VAD process, where the heuristic parameters were optimized on a development set. The segmented utterances are then fed into the decoder implemented in Kaldi, which produces word lattices.

Spoken Term Detection Subsystem

The Spoken Term Detection subsystem uses the keyword search tools provided by Kaldi. A brief description of the process, slightly modified from the one available in the Kaldi webpage is included here for completeness.

Lattices generated by the above ASR subsystem are processed using the lattice indexing technique described in [2]. The lattices of all the utterances in the search collection (speech data) are converted from individual weighted finite state transducers (WFST) to a single generalized factor transducer structure in which the start-time, end-time, and lattice posterior probability of each word token is stored as a 3-dimensional cost. This structure represents an inverted index of all word sequences seen in the lattices.

Given a query term, a simple finite state machine is created that accepts the term and composes with the factor transducer to obtain all occurrences of the term in the search collection, along with the utterance ID, start-time, end-time, and lattice posterior probability of each occurrence.

Finally, the decision maker simply sorts all these occurrences according to their posterior probabilities and a YES/NO decision is assigned to each occurrence.

OOV words are dealt with a method called proxy words, fully described in [4]. It essentially consists of substituting the OOV word to search with INV proxy words that are acoustically similar. The advantage of this method is that it does not require the use of a hybrid approach (word and sub-word models and lattices) as in our previous methods [14,15], being able to deal with OOV words using only a word ASR subsystem and a word-based lattice index. Our goal was to compare both methods in a different task but we could not perform that comparison for the evaluation and it remains as future work.

2.2. Train and development data

The evaluation task involves searching for some terms from speech data in the MAVIR corpus [[48]] that mainly contains speech in Spanish recorded during the MAVIR conferences. Since we did not have a large collection of comparable data, we decided to use a large database in Spanish to train the ASR module. We chose Fisher Spanish corpus [3] which amounts to 163 hours of conversational telephone speech (CTS) recordings (two sides) from 136 speakers. We used the same data for training the acoustic and language models. Since the data in the corpus (CTS) were very different from the data in the evaluation (mainly speech in conferences), we used the training and development data provided by the organizers, along with their transcriptions available in [[48]] to adapt the acoustic and language models.

The Fisher Spanish corpus was separated into mono recordings and divided into the three parts described in Table 1. The time in hours includes all the recorded silences.

Table 1. Partition of the Fisher Spanish corpus.

Part	# Recordings	# Speakers	# Hours
Train	1348	112	~268
Dev	146	12	~28
Test	144	12	~28

For training the ASR subsystem (acoustic and language models), we used the Train part and we evaluated the ASR subsystem using the Test part. The Dev part was used to tune parameters and to evaluate language model perplexity, which was 196.76.

When we started to process the corpus, we used our own rule-based grapheme-to-phoneme conversion module in Spanish to derive the phoneme transcriptions of the words in the lexicon. However, we soon realized that Fisher Spanish corpus had plenty of words in English (it is a Spanish corpus recorded mainly in the U.S.A.), so we had to perform a deeper analysis of the corpus and do *something* with the English words. Table 2 describes the types and amounts of words found in the Fisher Spanish corpus.

Table 2. Type and amount of words found in the Fisher Spanish corpus.

Type	# words
Spanish words	25400
English words	over 3000
Partial words	~ 2200
Interjections	94
Spanish Acronyms	85
English Acronyms	92

For the English words, we decided to use the CMU Dictionary to obtain an English phoneme transcription and define translation rules from English to Spanish phonemes to build the phoneme transcription of the English words using Spanish phonemes. Interjections and acronyms were transcribed manually. In the end, we had a dictionary of about 30,000 terms, fully transcribed with a set of 24 Spanish phonemes.

Besides the phoneme models, we included models for the different types of noise present in the corpus. Table 3 summarizes these types of noise and their absolute frequency in the Fisher Spanish Corpus.

Table 3. Non-speech events present in the Fisher Spanish Corpus that were modeled with independent acoustic models.

Type	# occurrences
<background>	8480
<laugh>	10957
<breath>	4728
<cough>	618
<sneeze>	16
<lipsmack>	203

We also used the MAVIR data provided by the organizers for training/development. The organizers of the evaluation provided 7 recordings that amount at about 5 hours in total. From these 7 files, we decided to use only 5 for training. These 5 files have been added to the Fisher Spanish corpus to adapt a bit the acoustic and language models to the target speech type. The 2 remaining files, which amount at about 2 hours of audio, were reserved to conduct development spoken term detection experiments.

For the final systems submitted we added all the training/development data (7 files) to train the final acoustic and language models used to process the evaluation materials and produce the final results.

2.3. Optimization and results on development data

We conducted initial experiments using only the Fisher Spanish corpus to evaluate our ASR subsystem, and then used the MAVIR training/development data for STD experiments. Here, we report these development and optimization results.

Table 4 summarizes the ASR results obtained on the Test partition of Fisher Spanish corpus (see Table 1) in terms or Word Error Rate (WER) for the different training stages.

Table 4. WER obtained at the different training stages of the ASR subsystem on Fisher Spanish corpus.

Training stage	WER (%)
MFCC_1	61.30
MFCC_2	61.18
+LDA + MLLT	56.98
+MLLR + SAT	52.07
+MMI	50.27
+fMMI	49.88

Our ASR results are not still state-of-the-art. For instance, in [11] a WER of 36.5% is reported on Fisher Spanish corpus, although the partition used for test is different to ours.

After testing our ASR subsystem on Fisher data, we tested it on the 2 files from the MAVIR data provided as training/development that we reserved for STD experiments. Results are presented on Table 5. Initial results (with only Fisher data for training) were very poor. After adapting the language (based on a new dictionary with ~1000 additional words) and acoustic models, results improved, but still reached a WER of about 75%, significantly worse than for Fisher Spanish data. This was something expected due to the mismatch between Fisher Spanish and MAVIR data. We used the language and acoustic models adapted using MAVIR data for the rest of the evaluation. For the final results, we even used all the 7 training/development files to improve adaptation to MAVIR data.

Table 5. WER obtained at the different training stages of the ASR subsystem on MAVIR data for Initial models (trained only on Fisher Spanish data), with LM adapted to MAVIR data and with language and acoustic models adapted to MAVIR data.

Training stage	WER (%) Initial	WER (%) LM adapted	WER (%) LM & acoustic models adapted
MFCC_1	99.21	82.02	79.49
MFCC_2	99.04	81.97	78.94
+LDA + MLLT	98.69	80.69	75.68
+MLLR + SAT	98.97	81.23	75.82
+MMI	98.65	80.61	74.59
+fMMI	99.09	83.15	76.69

We also conducted STD experiments on the 2 files reserved from the training/development data, and obtained the results presented in Table 6 in terms of Maximum Term Weighted Value (MTWV).

Table 6. MTWV obtained at the different training stages of the ASR subsystem on the 2 files of the MAVIR data reserved for STD experiments. As in Table 5, language and acoustic models adapted to MAVIR data used the remainding 5 MAVIR files.

Training stage	MTWV
MFCC_1	0.0971
MFCC_2	0.1160
+LDA + MLLT	0.1863
+MLLR + SAT	0.1603
+MMI	0.1845
+fMMI	0.1664

The system submitted as primary system was the one that optimized the MTWV, which is the LDA+MLLT system using language and acoustic models adapted to MAVIR data. YES/NO decision threshold was set to make ATWV reach the MTWV.

This system was further improved by including the 2 files from the MAVIR training/development data that we initially reserved for STD experiments in the training of the acoustic and language models. We did a final experiment on all the training/development data (7 MAVIR files) using this system. The results we obtained were MTWV=0.6287 and ATWV=0.6233. These results are much better than those obtained before because in this experiment a considerable amount of the material used for test (5 out of 7 files) was also used for training.

3. Contrastive System: ATVS-GEINTRA_STD_con1

The contrastive system we submitted is essentially the same, but optimized for WER instead of for MTWV. This is an MMI system optimized in threshold to make the ATWV meet the MTWV. As with the primary system, we improved the system by including in training the 2 files from the MAVIR training/development data initially reserved for STD experiments. The final experiment using all the 7 MAVIR training/development data achieved an MTWV=0.8327 and an ATWV=0.8155. Again, these results are highly unrealistic due to the re-use of data in training and test.

4. Conclusions and future work

This paper presents the ATVS-GEINTRA systems submitted to the ALBAYZIN 2014 Search on Speech Spoken Term Detection evaluation. Two systems were built. Both involve an ASR subsystem to produce word lattices and an STD subsystem for occurrence detection. Kaldi toolkit has been used to construct both subsystems. The systems were basically the same. The only difference relies on the ASR subsystem

configuration chosen. One system employed a WER optimization-based tuning, while the other simply tuned the whole system towards the STD metric (ATWV). The best system achieved an ATWV of 0.1863 on a subset of the development data.

Future work will focus on the ASR subsystem, whose performance relates to that of the entire STD system in a large extent. For that, a Deep Neural Network-based ASR system should improve the final STD performance, as has been shown in ASR research during the last few years.

5. Acknowledgements

This work has been partly supported by project CMC-V2 (TEC2012-37585-C02-01) from the Spanish Ministry of Economy and Competitiveness.

6. References

1. Abad, A., Rodríguez-Fuentes, L.J., Peñagarikano, M., Varona, A., Bordel, G.: On the calibration and fusion of heterogeneous spoken term detection systems. Proc. of Interspeech. pp. 20-24 (2013).
2. Can, D., Saraclar, M.: Lattice Indexing for Spoken Term Detection. IEEE Trans. On Audio, Speech, and Language Processing, 19(8), pp. 2338-2347 (2011).
3. Fisher Spanish Corpus, Available at Linguistic Data Consortium Catalogue with reference LDC2010S01 (speech) and LDC2010T04 (transcripts), https://catalog.ldc.upenn.edu.
4. Guoguo C., Yilmaz, O., Trmal, J., Povey, D., Khudanpur, S.: Using proxies for OOV keywords in the keyword search task. Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 416-421, (2013).
5. Katsurada, K., Miura, S., Seng, K., Iribe, Y., Nitta, T.: Acceleration of spoken term detection using a subarray by assigning optimal threshold values to subkeywords. Proc. of Interspeech. pp. 11-14 (2013).
6. Li, H., Han, J., Zheng, T., Zheng, G.: A novel confidence measure based on context consistency for spoken term detection. Proc. of Interspeech. pp. 2429-2430 (2012).
7. Liu, C., Wang, D., Tejedor, J.: N-gram FST indexing for spoken term detection. Proc. of Interspeech. pp. 2093-2096 (2012).
8. MAVIR Corpus. Available at: http://www.lllf.uam.es/ESP/CorpusMavir.html.
9. NIST: The spoken term detection (STD) 2006 evaluation plan. National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, 10 edn. (September 2006), http://www.nist.gov/speech/tests/std.
10. Norouzian, A., Rose, R.: An approach for efficient open vocabulary spoken term detection. Speech Communication 57, 50-62 (2014).
11. Post, M., Kumar, G., López, A., Karakos, D., Callison-Burch, C., Khudanpur S.: Improved Speech-to-Text Translation with the Fisher and Callhome Spanish–English Speech Translation Corpus. Proc. of International Workshop on Spoken Language Translation, (2013).
12. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The KALDI speech recognition toolkit. Proc. of ASRU (2011).
13. Szoke, I.: Hybrid word-subword spoken term detection. Ph.D. thesis, Brno University of Technology (June 2010).
14. Tejedor, J., Toledano, D.T., Wang D., Colás, J.: Feature Analysis for Discriminative Confidence Estimation in Spoken Term Detection. Computer Speech and Language, 28(5), pp. 1083-1114 (2014).
15. Tejedor, J., Toledano, D.T., Wang D.: ATVS-CSLT-HCTLab System for NIST 2013 Open Keyword Search Evaluation. LNCS/LNAI Proceedings of IberSPEECH 2014 (to appear).
16. Wang, D.: Out-of-vocabulary Spoken Term Detection. Ph.D. thesis, University of Edinburgh (December 2009).

D. Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de la ADAPTACIÓN DE UN SISTEMA DE BÚSQUEDA DE PALABRAS CLAVE AL CASTELLANO. 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 sistema. 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 Superior 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.