Universitat Politècnica de València
Departamento de Sistemas Informáticos y Computación

Algoritmos sobre grafos de unidades lingüísticas en procesamiento automático del habla

Trabajo Fin de Máster
Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Presentada por: Marcos Calvo Lance

Supervisores: Dr. Jon Ander Gómez Adrián
Departamento de Sistemas Informáticos y Computación
Dr. Emilio Sanchis Arnal
Departamento de Sistemas Informáticos y Computación

Índice general

Capítulo 1

Introducción

El procesamiento del lenguaje natural es una de las áreas del reconocimiento de formas que se caracteriza por el hecho de que la entrada al sistema codifica unidades lingüísticas. Si esta entrada es una señal acústica que recoge una pronunciación de un locutor hablamos de procesamiento automático del habla. Entre las múltiples tareas asociadas a éste se encuentran:

- Aplicaciones biométricas de identificación y verificación del locutor.
- Detección y establecimiento de fronteras entre unidades subléxicas.
- Decodificación acústico-fonética.
- Localización de palabras clave en fragmentos de audio.
- Transcripción completa de fragmentos de audio.
- Búsqueda de respuestas a preguntas formuladas por un locutor en lenguaje natural.
- Comprensión automática del lenguaje hablado.
- Gestión de diálogo hablado.

Muchas de estas aplicaciones pueden ser enfocadas desde distintos puntos de vista, entre ellos la búsqueda en grafos u otras estructuras derivadas de

éstos, como son los autómatas de estados finitos, los transductores y los modelos ocultos de Markov. En todos estos casos la representación del conocimiento en forma de grafos permite modelar relaciones complejas entre diferentes partes que de otra forma no serían posibles. Además, permite expresar la secuencialidad entre sucesos, ya que si se definen un conjunto de nodos iniciales y otro de nodos finales, cualquier camino de un nodo inicial a uno final representará una sucesión de eventos válida de acuerdo con la estructura del grafo. Esta secuencialidad es totalmente natural en el lenguaje hablado, donde una pronunciación puede verse como una sucesión en el tiempo de eventos físicos producidos por el sistema fonador humano.

En consecuencia, nuestro objetivo en el presente trabajo será abordar algunos de los problemas de procesamiento automático del habla mencionados anteriormente utilizando estructuras de grafos u otras derivadas de éstas.

En primer lugar, trataremos el problema de la detección de fronteras fonéticas. Esta tarea consiste en la identificación de los instantes de inicio y finalización de todos los fonemas contenidos en un fragmento de audio cuya transcripción textual es conocida a priori. La identificación de estas fronteras fonéticas es de especial interés en aplicaciones como la síntesis de voz a partir de un texto escrito (*text-to-speech synthesis* o TTS) y el entrenamiento de modelos acústicos para reconocimiento automático del habla. En este trabajo la aproximación que emplearemos para tratar el problema de la detección de fronteras fonéticas será basándonos en modelos ocultos de Markov.

El segundo problema que se abordará es el de la búsqueda de palabras clave sobre documentos hablados (o *spoken term detection*). Esta tarea cobra especial interés en aplicaciones donde no es importante reconocer exactamente la frase pronunciada por el locutor, sino determinar la presencia o ausencia de cierta palabra o secuencia de palabras en un determinado fragmento de audio. Una de estas aplicaciones es la indexación de grandes bases de datos de audio. De hecho, instituciones como NIST han mostrado interés en esta tarea organizando diferentes competiciones [11]. En este caso, la herramienta que se ha utilizado para abordar esta tarea ha sido cierto tipo de grafos en cuyos arcos se representará información fonética.

Por último, se presentará una aproximación a la comprensión del habla basada en grafos de palabras. El interés de la comprensión automática del habla se encuentra sobre todo en los sistemas de diálogo hablado persona - máquina. En ellos la entrada es una pronunciación que debe ser reconocida y a la que se le debe dar una representación semántica (proceso de comprensión) de forma que el sistema pueda "entender" lo que el usuario ha dicho y actuar en consecuencia (por ejemplo, proporcionando una determinada información). Tradicionalmente se aplica primero el proceso de reconocimiento, de forma que el módulo encargado de tal (ASR) proporciona como salida o bien la mejor frase decodificada (1-best) o bien una representación de las n mejores (n-best), la cual suele ser en forma de retículo (*lattice*). Posteriormente al reconocimiento se aplica el proceso de comprensión, teniendo éste que intentar recuperarse de los posibles errores que el primero haya podido introducir, así como tratar con la variabilidad e incertidumbre introducidas en el caso de haber utilizado alguna representación de las n-best. En este trabajo se expone una aproximación a la comprensión automática del habla que toma como entrada un grafo de palabras, que en nuestro caso se ha generado teniendo en cuenta únicamente información fonética y léxica.

Por tanto, la estructura del resto del presente trabajo se dividirá en 7 capítulos. En el próximo capítulo, se hará una revisión de la aplicación de los grafos y otras estructuras similares al procesamiento del habla, así como del estado del arte de cada una de las tareas concretas que se tratan en este trabajo. A continuación, se tratará la tarea de la segmentación automática a nivel fonético y se expondrá una aproximación basada en modelos ocultos de Markov, los cuales presentan una topología determinada y sus probabilidades fonéticas se han estimado mediante un proceso de *clustering* a nivel acústico. El capítulo 4 estará centrado en la tarea de la localización de términos en documentos hablados, y se presentará una aproximación basada en la utilización de grafos de fonemas. La tarea de la comprensión del habla será la idea central del capítulo 5, en el cual se expondrá una aproximación a ésta que toma como entrada un grafo de palabras que representa una pronunciación y, mediante algoritmos sobre grafos, obtiene como salida una secuencia de conceptos. El capítulo 6 estará dedicado a la exposición de las conclusiones a las que se han llegado y de las líneas de trabajo futuro que pueden seguirse

en cada una de las tres tareas aquí presentadas. Por último, en el capítulo 7 se citan las publicaciones relacionadas con este trabajo en las cuales el autor de éste ha tomado parte.

Capítulo 2

Revisión del estado del arte

2.1. Grafos y tecnologías del habla

Un grafo es una estructura de datos que consta de un conjunto de vértices o nodos y un conjunto de aristas o arcos (según sea el grafo dirigido o no, respectivamente) que los conectan. Estos arcos representan relaciones entre los vértices, y pueden estar etiquetados con algún tipo de información relevante. Además, esta estructura básica es el punto de partida para otros formalismos como los Modelos Ocultos de Markov (HMM, *Hidden Markov Models*), los Autómatas de Estados Finitos y los Transductores, así como las extensiones estocásticas de estos dos últimos.

Debido a su potente capacidad para representar la información, los grafos han sido extensamente utilizados en aplicaciones de tecnologías del habla. Uno de los ejemplos más conocidos es su uso por los sistemas de reconocimiento automático del habla (ASR) para representar de forma compacta una generalización de las n mejores transcripciones de una determinada pronunciación. A este tipo de grafos, donde las palabras están representadas en los arcos y sus nodos representan instantes de tiempo, se les conoce como retículos (*lattices*). Las *lattices* se caracterizan porque no tienen más restricciones en su estructura que ser acíclicas y dirigidas. Además sus arcos tienen asociados como peso la probabilidad acústica dada por el ASR a la palabra correspondiente a su etiqueta o la combinación de esta probabilidad con el

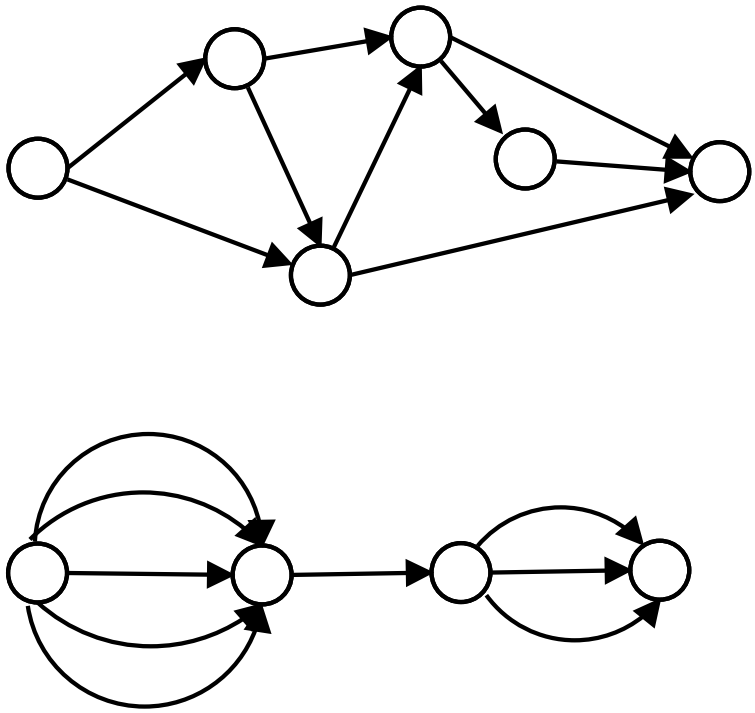

Figura 2.1: Arriba: *Lattice*. Su topología no está definida a priori siempre que sea dirigida y no tenga ciclos. Abajo: *Confusion network*. En este tipo de grafo todos los arcos van de un nodo al siguiente.

score otorgado por el modelo de lenguaje [18, 29, 48].

Un refinamiento en la estructura de las *lattices* nos conduce a las redes de confusión o *word confusion networks* [17, 30], las cuales tienen una topología muy simple, aunque no pierden potencia con respecto a la estructura original. La diferencia en la topología entre estos dos tipos de grafos puede verse en la figura 2.1.

Un tercer tipo de grafo muy útil para la tarea de recuperación de documentos hablados (*spoken document retrieval*) son las *position specific posterior lattices*, como se expone en [7].

Estas representaciones en forma de grafo de la salida de un ASR se han utilizado ampliamente en sistemas que tienen acoplado un reconocedor de voz a su entrada. Por ejemplo, en [5] se presenta una aplicación de las redes de confusión a la traducción automática del habla, de forma que se aprovecha su

topología durante el proceso de traducción. También en [50, 18] se exponen métodos para explotar la simplicidad de la topología de las redes de confusión para mejorar el rendimiento de sistemas de comprensión automática del habla. Como último ejemplo, citaremos que las *lattices* también se han utilizado en tareas de búsqueda de términos o palabras clave en documentos hablados (*Spoken term detection*, STD), como es el caso expuesto en [33], sistema que en la evaluación NIST 2006 sobre STD [11] consiguió el mejor rendimiento en la tarea correspondiente a conversaciones telefónicas en inglés.

Otras estructuras derivadas de los grafos y con una literatura extensísima en procesamiento del lenguaje hablado son los Modelos Ocultos de Markov, los Autómatas de Estados Finitos y los Transductores. Por ejemplo, los primeros son el formalismo típico con el que suele modelarse el comportamiento de las unidades subléxicas en los sistemas de reconocimiento automático del habla, aunque también se han utilizado, entre otros, en síntesis de voz [46] y segmentación fonética [43, 19]. De forma similar ocurre con los autómatas, cuya aplicación tiene cabida en multitud de áreas del procesamiento del habla. Por su interés, destacaremos su aplicación a la representación de modelos de lenguaje estadísticos en sistemas de reconocimiento del habla, como por ejemplo puede verse en [49]. Por último, los transductores también están teniendo presencia en los últimos años en las aplicaciones de las tecnologías del habla. Como muestra, en [41, 42] se expone el paradigma de comprensión automática del habla basado en transductores. Además, en [39] se trata de la traducción de habla también utilizando este formalismo y en [20] se presenta una aproximación basada en transductores estocásticos para la construcción de un gestor de diálogo hablado.

Llegados a este punto, podemos comprobar la gran importancia que tienen los grafos y otras estructuras derivadas de éstos en procesamiento automático del habla. Por otra parte, y dado que en este trabajo nos centraremos en tres de las aplicaciones ya mencionadas donde se utilizan este tipo de estructuras, pasaremos a exponer a continuación una revisión de la literatura publicada en los últimos años sobre ellas.

2.2. Detección de fronteras fonéticas

La detección de fronteras fonéticas, también conocida como segmentación del habla a nivel fonético, es la tarea consistente en la identificación de los puntos de un documento de audio donde el locutor cambia de fonema, siendo conocida la secuencia fonética que éste ha pronunciado. La importancia de esta tarea ha propiciado que se haya abordado desde diversos puntos de vista en los últimos años.

Por ejemplo, tomando como nexo común los Modelos Ocultos de Markov, diferentes autores han hecho diversas aportaciones a la manera de abordarla. En [43] se expone una aproximación basada en HMM donde se aplican reglas para variar la pronunciación esperada de la frase, generando una "red de pronunciaciones". Esta red es procesada mediante una búsqueda por Viterbi teniendo en cuenta la señal vocal, lo cual es utilizado para reestimar las probabilidades de los HMM, repitiendo este proceso hasta su convergencia, lo cual nos da la segmentación buscada. Otros trabajos que utilizan estas estructuras son [19], donde se utiliza una restricción del algoritmo de Baum-Welch para entrenar los HMM, y [37], donde se modifica la topología típica de los HMM para reconocimiento del habla eliminando los bucles sobre algunos estados, lo cual permite modelar mejor la acústica de las fronteras fonéticas controlando a la vez la duración mínima de los fonemas.

Un enfoque diferente es el empleado en [38], donde se utiliza un algoritmo de programación dinámica (DTW) para alinear la pronunciación real con una generada artificialmente mediante síntesis concatenativa de voz. También basado en DTW es el método que se presenta en [13], donde se expone un método para calcular las probabilidades a posteriori de cada uno de los fonemas del lenguaje dado cada uno de los *frames* acústicos de entrada y, una vez obtenidas éstas, se aplica el algoritmo de programación dinámica para obtener las fronteras fonéticas. Justamente lo expuesto en este trabajo, junto con las ideas de [37] serán los puntos de partida para el método de segmentación fonética que se expondrá más adelante.

Existen muchas otras formas de tratar este problema, así como ideas y variaciones al respecto, ya que desde hace muchos años ha sido objeto de

estudio de un gran número de investigadores. Sin embargo, para finalizar con esta sección expondremos otras dos que nos parecen especialmente interesantes. Una de ellas, basada en la minimización del *minimum phone error* [40], es la presentada en [23]. Aquí el objetivo es minimizar el error esperado en la situación de las fronteras fonéticas, utilizando para ello un conjunto de posibles alineamientos fonéticos representados como una *lattice*. La segunda es la expuesta en [36], donde se estudia la idoneidad de diferentes métodos de regresión lineal y no lineal para combinar las fronteras fonéticas calculadas por diferentes sistemas de segmentación automática.

2.3. *Spoken term detection*

La tarea de la detección de términos en documentos hablados (o *spoken term detection* en inglés) tiene como objetivo, según NIST, el procesamiento de grandes bases de datos de audio con el objetivo de encontrar ocurrencias de términos hablados. Estas bases de datos pueden ser de naturaleza muy heterogénea, por lo que en el año 2006, la propia NIST convocó una competición sobre esta tarea (ver [11]) en las que se distinguían las modalidades de *Broadcast News*, conversaciones telefónicas y conferencias. Esta competición se convocó para los idiomas inglés, chino mandarín y dos variedades de árabe, aunque la modalidad de búsqueda de términos en conferencias sólo estuvo disponible en inglés.

Actualmente pueden encontrarse en la literatura varias aproximaciones diferentes para esta tarea. Por ejemplo, en [33] se presenta un sistema basado en *lattices* de palabras obtenidas como salida de un reconocedor para grandes vocabularios. Este sistema fue el que mejor rendimiento dio en la competición anterior para la tarea de conversaciones telefónicas en inglés. Si en lugar de tomar como unidad fundamental las palabras consideramos los fonemas llegamos a la filosofía seguida por ejemplo en [45] (sistema también presentado a la competición convocada por NIST en 2006), donde la salida del reconocedor es una red de fonemas en lugar de palabras y sobre ésta se hace la búsqueda de los términos. También puede utilizarse una aproximación intermedia entre los fonemas y las palabras, como son los fragmentos

de palabras, las partículas o las sílabas, los cuales pueden ser apropiados para cierto tipo de lenguajes con características especiales, como es el caso del chino [32]. Sin embargo, las aproximaciones basadas en palabras tienen un inconveniente con respecto a las que toman como unidad fundamental algún tipo de fragmento de éstas y es que las primeras son totalmente insensibles a las palabras fuera del vocabulario del reconocedor (serían incapaces de detectarlas), mientras que en los basados en unidades subléxicas sí existiría esa posibilidad. También existen aproximaciones híbridas que combinan sistemas basados en unidades léxicas y subléxicas, como es el caso de la expuesta en [53].

También existen varias aproximaciones a la hora de representar el espacio donde deben buscarse los términos objeto de la consulta. Una de ellas son los sistemas basados en *lattices*, de los que ya se han presentado algunos ejemplos. Esta aproximación es muy sencilla, ya que constituye una forma natural de codificar la salida de un reconocedor expresada como sus n-best decodificaciones. Otra forma de representar este espacio es mediante la técnica conocida como "expansión de la consulta" (*query expansion* en inglés), la cual consiste en extender las palabras objetivo añadiendo algunos términos "similares". La medida de esta "similitud" para decidir qué palabras añadir a la consulta original puede ser obtenida por distancia de edición [52], confusión acústica [27] u otras métricas basadas en teoría de la información. Una tercera forma de representar este espacio es la conocida como *soft match*, en la que se utiliza un modelo de error basado en distancia de edición o confusión fonética para calcular la proximidad de un reconocimiento candidato del de un término objeto de la consulta a la forma léxica de éste (por ejemplo, si el término de la consulta fuera la palabra Valencia y el reconocedor diera como forma candidata del término la correspondiente a *Palencia* este modelo calcularía la distancia entre ambas atendiendo a un cierto criterio). Un ejemplo de sistema que utiliza la técnica de *soft match* es el presentado en [33].

Un último detalle al respecto de la detección de términos en documentos hablados es el tratamiento de las palabras de fuera del vocabulario. Ya se ha comentado que las aproximaciones basadas en unidades subléxicas tienen

más potencia en la detección de estas palabras. Sin embargo, este es de por sí un tema bastante amplio que admite largas discusiones sobre algunos de sus aspectos. Para más información es interesante consultar [51].

2.4. Comprensión del lenguaje hablado

La comprensión automática del habla es el proceso por el que, dada una pronunciación emitida por un locutor, se extrae una interpretación semántica de la información contenida en ésta basada en un conjunto de conceptos definido a priori. Los sistemas de comprensión del habla son especialmente útiles en sistemas de diálogo hablado, ya que debe comprenderse la pronunciación de entrada al sistema para que el módulo de gestión de diálogo pueda devolver una respuesta adecuada. Normalmente los módulos de comprensión suelen constar de dos etapas, a saber, extracción de la secuencia de conceptos que aparecen en la frase de entrada y asignación de valores a éstos basándose en las palabras de dicha frase. No hablaremos de la segunda de estas etapas, ya que queda fuera del ámbito del presente trabajo, pero hay que tener presente que es una parte importante para los sistemas de comprensión del lenguaje hablado.

En primer lugar, los sistemas de comprensión automática del habla suelen tener acoplados a su entrada un reconocedor de voz (ASR), el cual les proporciona o bien la mejor transcripción estimada de la pronunciación de entrada (1-best) o bien una representación de las n mejores [18, 50]. Sin embargo, como se discute en [8], el hecho de emplear por ejemplo una *lattice* como entrada al módulo de comprensión hace esta tarea más complicada, ya que hace el espacio de búsqueda de la decodificación semántica correcta todavía más grande. La ventaja, por el contrario, es que entre las posibles transcripciones que estarían representadas dentro de la *lattice* podría estar la correcta, por lo que el proceso de comprensión podría ayudar a recuperar errores cometidos durante el reconocimiento.

En los últimos años se han propuesto varias aproximaciones para la comprensión automática del habla [15, 42] basadas muchas de ellas en métodos

ya aplicados con éxito para otras tareas relacionadas. Por ejemplo, entre los modelos log-lineales, caben destacar los *maximum entropy Markov models* [31] y los *conditional random fields* (CRF) [24]. De hecho, en la experimentación reportada en [15] los mejores resultados se obtienen utilizando los CRF. Otra aproximación es la basada en el marco utilizado en Traducción Automática (SMT, *Statistical Machine Translation*) [28], en la que se utilizan modelos similares a los IBM y a los modelos de frases empleados en este campo de estudio. También las Redes Bayesianas Dinámicas (*Dynamic Bayesian Networks*), que fueron aplicadas con éxito a tareas relacionadas como el etiquetado de actos de diálogo [21], pueden ser aplicadas a la comprensión automática del habla, tal y como puede verse en [15, 25, 26].

Un último enfoque es el basado en transductores estocásticos [15], donde el objetivo es maximizar la probabilidad conjunta de la secuencia de palabras y la secuencia de conceptos. Esta maximización se lleva a cabo efectuando la composición de 4 ó 5 transductores (según sea el caso), los cuales se corresponden con diversos niveles de conocimiento acústico, léxico y semántico. Por último se efectúa una búsqueda al estilo de Viterbi sobre el transductor resultante. Una forma fácil de llevar a cabo esta composición de transductores es utilizando la biblioteca AT&T FSM/GRM [34].

Al igual que en otras áreas del reconocimiento de formas, en comprensión del habla también se han llevado últimamente a cabo intentos de fusionar las salidas obtenidas a partir de varios tipos de clasificadores, utilizando por ejemplo el sistema ROVER [10, 16]. En [8, 15] pueden verse resultados experimentales al respecto.

Capítulo 3

Detección de fronteras fonéticas basada en Modelos Ocultos de Markov

3.1. Definición de la tarea

La detección de fronteras fonéticas, también conocida como segmentación del habla a nivel fonético, es la tarea consistente en la identificación de los puntos de un documento de audio donde el locutor cambia de fonema, siendo conocida la secuencia fonética que éste ha pronunciado. Algunos corpus, como *Albayzin* [35], tienen una parte de las pronunciaciones segmentadas manualmente a nivel fonético por un experto. Sin embargo, este trabajo es caro y tedioso, por lo que es interesante disponer de un método que realice esta tarea de forma automática. La disponibilidad de un corpus en el que se han identificado sus fronteras fonéticas es crucial para aplicaciones como el entrenamiento de modelos acústicos y la síntesis de voz a partir de texto (*text-to-speech synthesis*). En estas aplicaciones es muy importante saber dónde empieza y acaba cada fonema, ya que para cada uno de ellos los fragmentos de audio en los que se está pronunciando dicho fonema se tratan de forma separada.

Por otro lado, podría plantearse la cuestión de si una segmentación fonética obtenida automáticamente podría llegar al nivel de una dada por un experto o si la discrepancia con respecto a ésta sería tan grande que a efectos prácticos sería inutilizable. Para estudiar esta cuestión se han llevado a cabo trabajos como los presentados en [22, 47], en los que se dio la misma base de datos de audio a varios expertos para que la segmentaran manualmente y medir posteriormente el grado de coincidencia de las fronteras fonéticas establecidas. En [47] la coincidencia de estas fronteras en un intervalo de 20 milisegundos fue del 97 %, mientras que en [22] fue del 93 %. Estos resultados nos llevan a dos conclusiones muy importantes sobre la tarea de la segmentación fonética. En primer lugar, el hecho de que exista una discrepancia entre los propios expertos humanos implica que si un sistema automático alcanza un cierto porcentaje de error con respecto a una determinada segmentación de referencia, podría ocurrir que, si ésta hubiera sido dada por otro experto, muy probablemente el porcentaje de error sería diferente. En segundo lugar, esta misma discrepancia constituye una justificación para el uso de sistemas de segmentación automática para aplicaciones prácticas, ya que son más baratos que la obtención de una segmentación manual cuya precisión sabemos que es discutible.

Además, los resultados anteriores justifican el porqué de establecer intervalos de tolerancia, en el caso anterior de 20 ms. Ha quedado demostrado que es muy difícil incluso para los expertos ponerse de acuerdo en el momento exacto en el que termina un fonema y comienza otro, entre otros motivos porque la duración de éstos en el habla "normal" suele ser muy reducida. Por eso es interesante establecer unos márgenes dentro de los cuales las fronteras fonéticas dadas por un sistema automático se consideran correctas. En consecuencia, decir que se establece un intervalo de tolerancia de x milisegundos quiere decir que, si la frontera de referencia está en el instante y, para que la frontera fijada automáticamente se considere acertada deberá estar en el intervalo $[y - x, y + x]$.

3.2. Un método de segmentación fonética basado en modelos ocultos de Markov

El método de segmentación que aquí se propone está basado en la obtención de sucesivos alineamientos mediante un algoritmo de programación dinámica (PD) entre la concatenación de una serie de modelos ocultos de Markov en los que se representa la acústica de los fonemas de la frase que se sabe que ha sido pronunciada y la parametrización de la señal de entrada al sistema. Este método no utiliza información alguna sobre los instantes de inicio y final de los fonemas de las frases de entrenamiento, sino que es suficiente con disponer de la secuencia fonética que se sabe que se ha pronunciado.

3.2.1. Topología y parámetros de los HMM

Como es habitual en los sistemas de reconocimiento automático del habla, cada unidad fonética queda representada mediante un modelo oculto de Markov (HMM). Sin embargo, y siguiendo la idea expuesta en [37], modificaremos la topología clásica de estos HMM para controlar mejor su duración mínima y las transiciones entre ellos. De acuerdo con esta filosofía, los HMM que utilizaremos tendrán dos tipos de estados:

- Estados con bucles (transiciones a sí mismos). Estos sólo podrán ser los estados centrales del HMM.
- Estados sin bucles o *duration control states*. En cada HMM existirá un número B de estos estados a cada lado de los estados centrales que sí tienen bucles. La finalidad de los *duration control states* es marcar la duración mínima de las unidades fonéticas que representa el HMM en el que están, así como representar mejor el fenómeno de transición entre fonemas.

Todos los estados del modelo serán emisores. En la figura 3.1 se muestra un ejemplo de HMM con una topología de 8 estados, de los cuales 6 son *duration control states* y se encuentran 3 a cada lado de los estados centrales

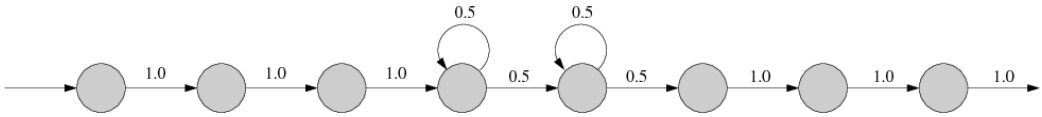

Figura 3.1: HMM de 8 estados con 3 *duration control states* a cada lado. Las probabilidades de transición las que se han utilizado en la experimentación que se expondrá posteriormente.

que sí tienen bucles. Como puede apreciarse en la figura, un estado sea del tipo que sea siempre está conectado con el estado siguiente (a no ser que sea el último, el cual estaría eventualmente enlazado al primer estado del HMM que modele la siguiente unidad fonética) y no se permiten *skips*. Es importante destacar que, de acuerdo con su naturaleza, dos HMM que modelen a dos unidades fonéticas distintas podrían tener topologías diferentes.

Otro aspecto importante de los modelos ocultos de Markov que aquí se proponen son sus probabilidades de emisión $p(x_t|e_i^u)$, donde x_t representa el t-ésimo *frame* acústico de la entrada y e_i^u el i-ésimo estado de la unidad fonética u. Estas probabilidades se calculan de acuerdo con la fórmula 3.1, en la que se asume independencia condicional entre x_t y e_i^u dado a. Las ideas subyacentes a esta fórmula y su uso para segmentación fonética fueron propuestos en [13, 14].

$$p(x_t|e_i^u) = \sum_{a \in A} p(x_t|a) \cdot p(a|e_i^u) \tag{3.1}$$

En esta ecuación, A representa un conjunto de clases acústicas modeladas cada una de ellas por una distribución Gaussiana, mientras que a simboliza una de estas clases. El conjunto de Gaussianas A se calcula como un preproceso previo a la estimación de los parámetros de los HMM. En él se utilizan todos los *frames* acústicos del corpus de entrenamiento, y se lleva a cabo por medio de un proceso de *clustering* paramétrico no supervisado utilizando el criterio de máxima verosimilitud, tal y como se explica en [9]. Por tanto, $p(x_t|a)$ se obtiene simplemente a partir de la conocida fórmula de la probabilidad que una distribución gaussiana otorga a un elemento de su dominio. Cabe destacar que el número de clases acústicas que deben considerarse (es decir, la cardinalidad de A) debe ser mucho mayor que el número de unidades

fonéticas, para modelar así la variabilidad que introducen las diferentes manifestaciones acústicas del tracto vocal, el estado de ánimo del locutor, su acento y timbre, etc.

Por otro lado, el segundo miembro del producto interior al sumatorio, $p(a|e_i^u)$, modela la relación entre los distintos estados de los diferentes modelos de ocultos de Markov que representan a las unidades fonéticas y las clases acústicas anteriormente mencionadas. Evidentemente, estas probabilidades condicionales deben normalizarse de manera que $\sum_{a \in A} p(a|e_i^u) = 1$. La inicialización de estas probabilidades puede hacerse mediante el método de *flat start* o segmentación a partes iguales y su posterior refinamiento puede realizarse tanto de modo supervisado como no supervisado. En el primero de los casos, se necesita un corpus segmentado y etiquetado manualmente; mientras que en el segundo puede utilizarse un método de refinamiento sucesivo de las fronteras fonéticas como el que se propone a continuación. Este proceso de refinamiento también podría utilizarse para reestimar las probabilidades de transición de cada uno de los HMM, pero esta posibilidad no se ha explorado en el presente trabajo. Dicho proceso concluye cuando la posición de las fronteras fonéticas calculadas por el sistema se estabiliza.

3.2.2. Un algoritmo iterativo de refinamiento de las fronteras fonéticas

Para llevar a cabo la reestimación y refinamiento de las fronteras fonéticas, así como de las probabilidades de emisión de los HMM, proponemos un algoritmo iterativo. En cada paso de este algoritmo se calcula el mejor alineamiento entre los *frames* acústicos de entrada y la concatenación de los HMM correspondientes a las unidades fonéticas que se sabe aparecen en la pronunciación mediante un algoritmo de PD, cuyos movimientos posibles dentro de la matriz de programación dinámica se muestran en la figura 3.2. Como puede apreciarse, los movimientos "horizontales" sólo están permitidos en los estados con bucles, ya que implican quedarse en el mismo estado de partida, y los movimientos "diagonales" pueden efectuarse desde todos los estados porque lo que simbolizan es una transición desde un estado al siguiente, lo

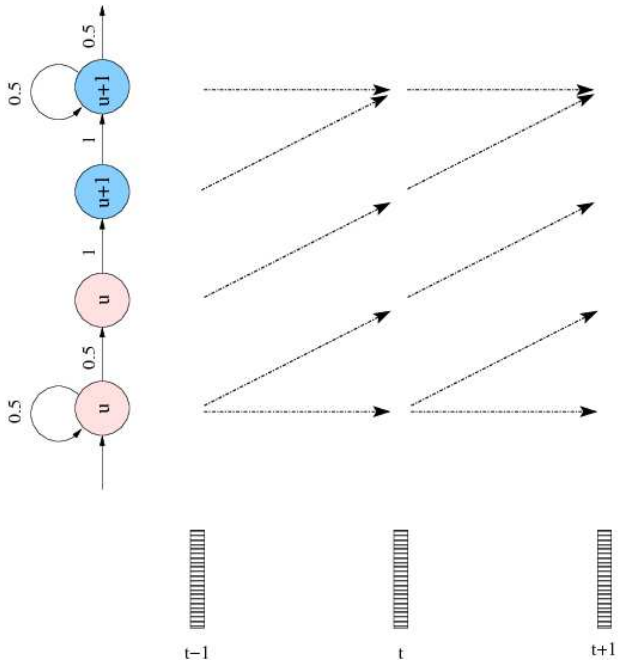

Figura 3.2: Movimientos posibles en nuestro algoritmo de Programación Dinámica para el alineamiento de *frames* acústicas y estados de los HMM. Esta figura se centra en la unión de dos HMM con 1 *duration control state* cada uno y que representan respectivamente a las unidades fonéticas u y $u + 1$.

cual es siempre posible. Los movimientos "verticales" están prohibidos ya que implicarían avanzar de un estado al siguiente sin consumir ningún *frame* (o, equivalentemente, asignar un mismo *frame* a dos estados a la vez), lo cual es inconsistente con la filosofía de los HMM.

Como función de coste en este algoritmo de PD se toman en cuenta tanto las probabilidades de transición de los HMM (ver figura 3.1) como las probabilidades de emisión calculadas de la manera expuesta en la sección anterior.

Tras aplicar de forma completa el algoritmo de programación dinámica, al recuperar el "camino" que proporciona el mejor alineamiento obtenemos también qué *frames* se han asignado a cada estado de cada HMM y, por tanto, podemos inferir cuáles son las fronteras fonéticas inducidas por este

alineamiento. Esta asignación de *frames* a estados nos permite actualizar las probabilidades de emisión, en concreto el miembro $p(a|e_i^u)$ de la ecuación 3.1. Esto es debido a que los *frames* que se han asignado a un estado determinado en el algoritmo de PD que se acaba de completar no tienen por qué coincidir plenamente con los que se asignaron en la iteración anterior (o en la inicialización si es la primera iteración) y por tanto la probabilidad de que una clase acústica se emita en dicho estado cambiará con respecto a la de la iteración anterior. Este razonamiento nos lleva al proceso iterativo de aplicar este algoritmo de programación dinámica de forma sucesiva, con sus correspondientes reestimaciones de probabilidades condicionales, tomando como criterio de parada la convergencia de las fronteras fonéticas estimadas.

3.2.3. Definición de las unidades fonéticas

Una opción muy común en reconocimiento automático del habla y muy intuitiva para la tarea que nos ocupa es asumir que cada unidad fonética se corresponde con un fonema. Por tanto, como cada HMM modela el comportamiento de una unidad fonética, esto implicaría que existe un HMM por cada fonema y que la acústica de cada fonema está representada en un solo HMM.

Sin embargo, no es esta la única opción posible y, dado que lo que se desea establecer son las fronteras entre fonemas, posiblemente exista una definición mejor de cuáles son las unidades fonéticas de interés para la tarea. Teniendo en mente esta idea se propone utilizar como unidades fonéticas adicionales a los propios fonemas las transiciones entre éstos, es decir, modelar la acústica de los fragmentos del habla que suponen la unión entre fonemas mediante HMM adicionales. Este hecho supone una dificultad adicional, ya que, al recuperar el camino sobre la matriz del algoritmo de PD podremos saber cuándo comienza y termina la transición entre dos fonemas, pero esto no nos dará una información directa de dónde situar la frontera fonética entre ellos. La solución que se ha tomado a este respecto es que, dados dos fonemas u y v cuya frontera fonética se desea determinar, y siendo t_1 el instante en el que el algoritmo de PD ha pasado de asignar *frames* a estados de u a asignarlos

a estados de $u + v$ (la transición entre fonemas) y t_2 el instante en el que abandona el HMM de $u + v$ para pasar al de v, la frontera fonética entre u y v se situará en el instante $\frac{t_1+t_2}{2}$.

En la experimentación que se expone a continuación se ha estudiado tanto el uso de únicamente los fonemas como unidades fonéticas como la utilización de éstos más las transiciones fonéticas.

3.3. Experimentación y resultados

Para llevar a cabo una experimentación completa que sirva para comprobar la eficacia del algoritmo de segmentación fonética que aquí se presenta, se han efectuado experimentos sobre dos bases de datos de audio en dos idiomas diferentes: *Albayzin* [35] para el castellano y TIMIT [12] para el inglés. Las características fonéticas de estos dos idiomas son bastante diferentes entre sí, por lo que un buen funcionamiento del algoritmo en ambos casos sería un síntoma de robustez frente a las características fonéticas de la lengua. Por otro lado, el número de fonemas del inglés es mayor que el del castellano.

El corpus fonético de la base de datos *Albayzin* se compone de 6800 pronunciaciones que constituyen alrededor de 6 horas de audio. De ellas, se han utilizado las 1200 frases que están manualmente segmentadas y etiquetadas para test y las 5600 restantes para entrenamiento, de forma que ningún locutor aparece en ambos conjuntos. El motivo de haber hecho esta división es que nuestro algoritmo no necesita para ser entrenado ningún tipo de información sobre los instantes de inicio y final de cada uno de los fonemas de la frase; sólo es necesaria la secuencia fonética. Por el contrario, para comprobar que las fronteras fonéticas se han detectado correctamente sí es útil disponer de un buen número de frases etiquetadas por un experto. De todos modos, como ya se expuso anteriormente, los etiquetados llevados a cabo por un experto humano son segmentaciones subjetivas y, por tanto, propensas a discusión entre los propios expertos.

La base de datos TIMIT es la más utilizada para hacer experimentos sobre segmentación fonética para el inglés. Contiene 6300 pronunciaciones,

que hacen un total de aproximadamente 5 horas de audio, obtenidas a partir de 630 locutores diferentes que representan 8 dialectos distintos del inglés de Estados Unidos. Para la experimentación aquí expuesta se ha utilizado la división de entrenamiento y test sugerida en [12].

Para los experimentos con ambas bases de datos se ha utilizado el mismo tipo de parametrización acústica. Cada *frame* se ha compuesto de 39 valores reales, en concreto: la energía normalizada, los 12 primeros coeficientes cepstrales según la escala de Mel y las primeras y segundas derivadas de cada uno de ellos. Además, estos *frames* se han obtenido aplicando una ventana de Hamming de 20 ms cada 5 ms (frecuencia de submuestreo = 200 Hz).

Como ya se mencionó anteriormente, es muy conveniente establecer un intervalo de tolerancia, de forma que si la frontera fonética calculada por el método de segmentación automática se sitúa dentro de él se considere como correcta, tomándose como fallo en caso contrario. Tradicionalmente en la literatura se exponen los resultados experimentales utilizando un conjunto de tolerancias, lo que da una idea de lo "fina" que es la precisión del método automático con respecto a la segmentación de referencia, así como de la distribución temporal de los errores en la situación de las fronteras. En nuestro caso estudiaremos el porcentaje de fronteras fonéticas correctamente situadas para tolerancias de 5, 10, 15, 20, 30 y 50 ms.

Los parámetros más importantes de nuestro método de segmentación son los referentes a su topología, en concreto el número total de estados del modelo y la cantidad de *duration control states* situados a cada lado de los estados centrales. Por ello, una parte de nuestra experimentación se ha centrado en investigar cómo la topología de los HMM afecta al error en la situación de las fronteras fonéticas, tomando como unidades fonéticas únicamente los fonemas. Aunque a priori cada unidad fonética podría tener asociada una topología diferente, una experimentación completa probando todas las distintas combinaciones de topologías y unidades fonéticas sería inviable, por lo que se ha decidido utilizar en general la misma topología para todas las unidades fonéticas en cada uno de los experimentos. Sin embargo, esto podría implicar que ciertas unidades fonéticas tuvieran una duración mínima excesiva, teniendo en cuenta sus características fonéticas. Por ello, en los

	Albayzin					
Topología	Tolerancia en ms					
$E \times B$	5	10	15	20	30	50
5x1	33.0	58.5	74.9	85.3	94.6	98.7
5x2	36.8	62.6	78.6	87.5	94.7	98.5
6x2	37.1	64.4	80.0	87.9	95.2	98.8
7x0	31.7	58.5	75.5	85.2	94.1	98.3
7x1	33.6	61.0	77.3	85.8	94.4	98.5
7x2	36.2	63.0	78.6	86.9	95.1	98.7
7x3	40.9	67.8	82.1	89.1	95.6	98.9
8x3	40.5	67.5	82.1	**89.5**	**96.2**	**99.2**
9x2	39.8	66.8	81.1	88.5	95.7	98.9
9x3	38.1	66.0	81.5	89.0	96.1	99.2
9x4	**44.0**	**70.3**	**82.8**	89.4	95.8	99.0
10x4	42.5	68.9	82.2	88.9	95.8	99.0

Tabla 3.1: Porcentaje de fronteras fonéticas correctamente fijadas en el corpus Albayzin para diversas topologías y un conjunto de tolerancias.

experimentos realizados con la base de datos *Albayzin* se ha utilizado una topología de 5 estados con 2 *duration control states* a cada lado para las oclusivas sonoras (/*b*/, /*d*/ y /*g*/) cuando la topología general del resto de fonemas ha superado los 5 estados. Del mismo modo, en los experimentos realizados con TIMIT se ha utilizado en todos los casos una topología de 3 estados con 1 *duration control state* a cada lado para todos los fonemas oclusivos (/*b*/, /*d*/, /*g*/, /*p*/, /*t*/ y /*k*/), ya que el silencio preoclusivo se modela como una unidad fonética independiente. En todos los casos y para ambas bases de datos los silencios se han modelado con una topología de 3 estados y ningún *duration control state*.

En las tablas 3.1 y 3.2 pueden verse los resultados obtenidos empleando diversas topologías y tolerancias para los experimentos con *Albayzin* y TIMIT respectivamente. En ellas hemos utilizado la nomenclatura $E \times B$ para expresar la topología de los HMM, la cual representa que el modelo tiene E estados en total y B *duration control states* a cada lado de los estados

	TIMIT					
Topología	Tolerancia en ms					
$E \times B$	5	10	15	20	30	50
5x1	25.5	46.6	62.0	72.7	88.0	97.7
5x2	22.4	43.7	61.9	74.8	89.6	97.8
6x2	**29.5**	**53.6**	69.9	80.3	91.6	97.9
7x0	24.4	44.9	60.8	72.3	88.0	97.9
7x1	24.4	45.2	62.3	74.3	89.5	98.1
7x2	28.5	52.1	68.9	79.8	91.8	98.2
7x3	24.7	47.8	66.6	78.6	91.2	98.1
8x3	27.8	51.9	70.7	**82.7**	93.6	98.5
9x2	28.6	52.2	69.0	79.8	91.6	97.7
9x3	28.2	52.0	**70.8**	82.6	**93.8**	**98.6**
9x4	25.4	49.9	69.3	81.5	92.7	98.2
10x4	26.3	50.1	68.2	79.9	91.6	98.1

Tabla 3.2: Porcentaje de fronteras fonéticas correctamente fijadas en el corpus TIMIT para diversas topologías y un conjunto de tolerancias.

centrales. En todos los experimentos representados en estas tablas la convergencia del algoritmo de segmentación se produce en menos de 20 iteraciones.

Los resultados obtenidos muestran que el uso de *duration control states* permite mejorar el acierto en la situación de fronteras fonéticas para todas las tolerancias estudiadas, aunque este aumento es mayor para tolerancias menores o iguales a 20 ms. Esto también implica que la precisión en la situación de estas fronteras con respecto a la de referencia también es mayor. Por otro lado, puede verse que los porcentajes de acierto obtenidos para el inglés son menores que para el castellano. Esto se debe a características propias de las lenguas estudiadas, y entre otros motivos a que el conjunto de fonemas del inglés es mayor que el del castellano.

Es importante destacar que los resultados obtenidos para la base de datos TIMIT son similares (y en algunos casos mejores) que los reportados por otros investigadores (por ejemplo, [36]) utilizando otras técnicas también basadas en HMM pero empleando frases segmentadas manualmente para

Albayzin					
Tolerancia en ms					
5	10	15	20	30	50
40.6	68.7	83.2	90.5	96.4	99.3

Tabla 3.3: Porcentaje de fronteras fonéticas correctas en el corpus *Albayzin* para un conjunto de tolerancias cuando las transiciones entre fonemas se consideran como unidades fonéticas adicionales.

TIMIT						
Usando	Tolerancia en ms					
manual	5	10	15	20	30	50
No	31.5	55.8	71.0	81.1	92.3	98.2
Sí	44.1	70.3	81.9	88.2	94.8	98.7

Tabla 3.4: Porcentaje de fronteras fonéticas correctamente situadas en el corpus TIMIT para un conjunto de tolerancias cuando las transiciones entre fonemas se consideran como unidades fonéticas adicionales. Dado que en este corpus se dispone de información sobre segmentación manual para entrenar, también se muestran los resultados obtenidos empleando dicha información para inicializar los modelos.

entrenar, información que no es necesaria para nuestro algoritmo y que no se ha utilizado en estos experimentos.

Un segundo juego de experimentos se ha llevado a cabo añadiendo las transiciones entre fonemas como unidades fonéticas adicionales. En este caso se ha utilizado una topología 6×2 para todas las unidades fonéticas, a excepción de las oclusivas, que se han representado mediante un HMM de 4×1 para *Albayzin* y 3×1 para TIMIT. Los silencios se han modelado mediante una topología 3×0 en todos los casos.

En la tabla 3.3 pueden verse los resultados obtenidos para *Albayzin* utilizando el conjunto de unidades fonéticas ampliado. Estos resultados, comparados con los mejores obtenidos en los experimentos anteriores (tabla 3.1) son en general bastante similares, aunque puede apreciarse una leve mejoría para tolerancias mayores o iguales a 15 ms.

Por otro lado, en la tabla 3.4 se presentan los resultados obtenidos en los experimentos con la base de datos TIMIT utilizando el conjunto de unidades fonéticas ampliado. Dado que para este corpus se dispone de información manual para entrenar, también se han realizado pruebas utilizándola para inicializar las probabilidades de emisión. Los resultados muestran, curiosamente, que los porcentajes de acierto obtenidos no utilizando la información manual son mejores para tolerancias menores de 20 ms con respecto a los reflejados en la tabla 3.2. Sin embargo, si se tiene en cuenta la información de la segmentación manual los resultados obtenidos mejoran muy significativamente con respecto a los anteriores, en especial para tolerancias pequeñas.

Capítulo 4

Spoken term detection sobre grafos de fonemas

4.1. Definición de la tarea

La tarea de la detección de términos en documentos hablados (o *spoken term detection* (STD) en inglés) tiene como objetivo, según NIST, el procesamiento de grandes bases de datos de audio con el objetivo de encontrar ocurrencias de términos hablados. A pesar de esta definición, los términos objeto de la consulta suelen presentarse al sistema de detección de forma textual. Por tanto, un sistema de STD sería el equivalente a un motor de búsqueda de pasajes textuales sobre documentos textuales, con la diferencia de que aquí los documentos objetivo de la búsqueda son ficheros de audio, donde hay que determinar la presencia o ausencia de un cierto término especificado textualmente. Esta presentación textual de los términos hace necesario algún tipo de transcriptor que obtenga la secuencia fonética a partir de la representación ortográfica [6]. Además, un término puede estar compuesto por una o varias palabras, así que se debe tratar con la aparición de posibles silencios entre las palabras que conforman el término.

Típicamente, un sistema de detección de términos en documentos hablados se compone de tres módulos conectados entre sí de forma secuencial, tal

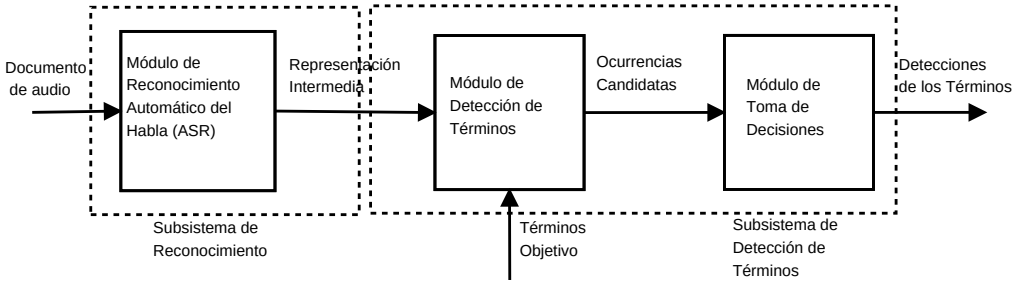

Figura 4.1: Esquema de los módulos de un sistema de *Spoken Term Detection*.

y como se muestra en la figura 4.1. Estos módulos son:

1. Un sistema de reconocimiento de voz (ASR) que proporciona como salida una representación intermedia de la pronunciación. Hablamos de representación intermedia porque para que un ASR sea apto para la tarea de *spoken term detection* no debe facilitar únicamente la mejor transcripción de la pronunciación de entrada, sino varias posibilidades representadas de forma estructurada (por ejemplo, utilizando *lattices*). Además, la salida de este ASR no tiene por qué estar basada en palabras, sino que puede tomar como unidades fundamentales entidades subléxicas tales como los fonemas.

2. Un módulo de detección de términos, el cual es el encargado de encontrar las ocurrencias candidatas de los términos objeto de la consulta.

3. Un módulo de toma de decisiones, cuyo cometido es determinar cuáles de las ocurrencias candidatas proporcionadas por el módulo anterior pertenecen realmente al término especificado.

Normalmente los tres módulos anteriores se agrupan a su vez en dos subsistemas: el subsistema de reconocimiento, formado únicamente por el ASR, y el subsistema de detección de términos, compuesto por la unión de los módulos de detección y toma de decisiones.

4.2. Un sistema de *spoken term detection* basado en grafos de fonemas

Como ya se ha explicado anteriormente, existen varias maneras y filosofías de generar la representación intermedia que posteriormente utilizará el subsistema de detección de términos. La que utilizaremos en el sistema que aquí se propone está basada en fonemas y se representa mediante una estructura de grafo dirigido de forma que:

- Los nodos estén etiquetados con marcas de tiempo (identificadores temporales de *frames* acústicos).
- Los arcos tengan asociada información de los fonemas que pueden haberse pronunciado entre los instantes temporales que representan sus nodos inicial y final, así como la probabilidad a posteriori de dicho fonema durante ese intervalo temporal.

Tras haber construido el grafo correspondiente a una pronunciación se aplica un algoritmo de búsqueda basado en una estrategia de programación dinámica, lo cual permite hallar las ocurrencias candidatas del término objeto de la consulta, así como un mecanismo de toma de decisiones para podar los falsos positivos.

A continuación detallamos en primer lugar el método de construcción de los grafos de fonemas, para centrarnos posteriormente en el método de búsqueda.

4.2.1. El subsistema de reconocimiento

Un esquema general del subsistema de reconocimiento que utilizamos en nuestro sistema de STD es el que se muestra en la figura 4.2. En él pueden distinguirse 4 módulos, cada uno de los cuales proporciona como salida los datos que constituyen la entrada al siguiente. Tomando los módulos por separado tenemos:

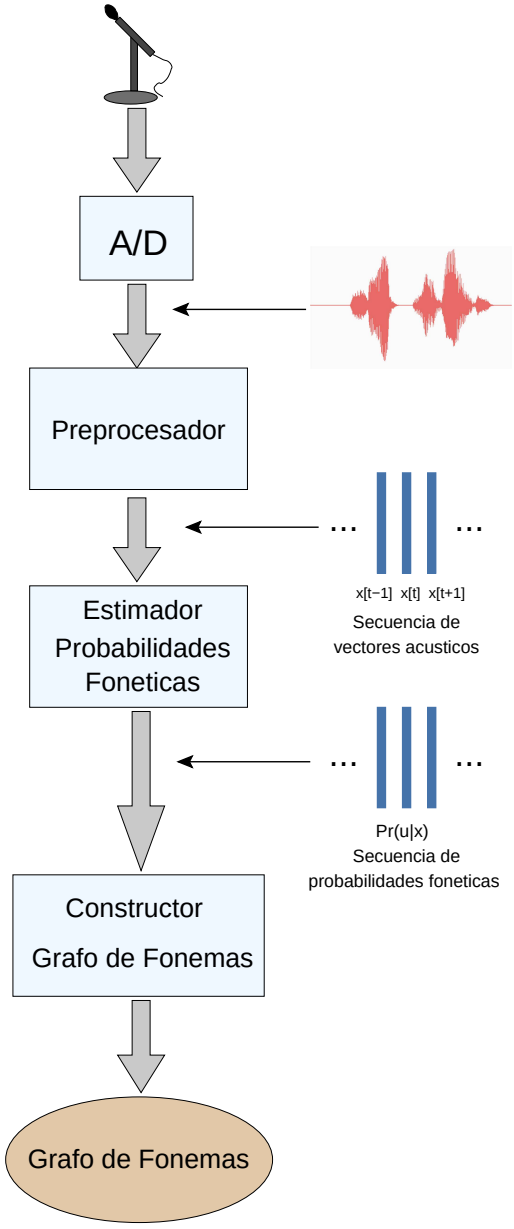

Figura 4.2: Esquema general del subsistema de reconocimiento de nuestro sistema de *spoken term detection*

1. En primer lugar se capta la señal acústica de entrada por un dispositivo adquisidor y se digitaliza (típicamente a 16 KHz) y filtra.

2. A continuación se convierte esta señal en vectores de características (*frames*) útiles para el sistema de reconocimiento. En nuestro caso hemos utilizado la energía, los 12 primeros coeficientes cepstrales en la escala de Mel y las primeras y segundas derivadas de todos ellos, resultando en total vectores de 39 componentes. Para este sistema se ha utilizado una frecuencia de submuestreo de 100 Hz (un *frame* cada 10 ms de audio) aplicando una ventana de Hamming de 20 ms.

3. Por medio de un sistema de estimación de probabilidades fonéticas a posteriori se calcula para todo *frame* x de entrada la probabilidad $p(u|x)$, para toda unidad fonética u. En consecuencia, la salida de este módulo es un vector v por cada *frame* de entrada, cada uno de ellos con tantas componentes como unidades fonéticas se estén considerando y donde $v(ind(u)) = p(u|x)$, siendo $ind(u)$ el índice que representa a la unidad fonética u en el vector y x el *frame* correspondiente al vector v. Al tratarse del cálculo de una distribución de probabilidad condicional, para cada *frame* x, se cumple que $\sum_{u \in U} p(u|x) = 1$.

4. A partir de un análisis de la evolución temporal de las probabilidades a posteriori calculadas en el módulo anterior pueden localizarse los segmentos donde potencialmente cada una de las unidades fonéticas ha podido ser pronunciada. Determinando los puntos de inicio y finalización de estos segmentos y utilizando el algoritmo que se expondrá posteriormente, obtenemos como salida de este módulo un grafo de fonemas que representa la pronunciación de entrada al subsistema reconocedor.

Aunque podría no ser así, a partir de este momento las unidades fonéticas a considerar serán el conjunto de fonemas del castellano más dos especiales que representen respectivamente el silencio largo y la pausa corta.

Estimación de las probabilidades fonéticas

La finalidad del módulo de estimación de las probabilidades fonéticas es calcular, para cada *frame* x y cada unidad fonética u, la probabilidad $p(u|x)$. El proceso de estimación de estas probabilidades sigue la misma idea que el presentado en el capítulo anterior para el cálculo de las probabilidades de emisión de los HMM, y está basado en el expuesto en [13].

Para estimar las probabilidades a posteriori $p(u|x)$ aplicamos en primer lugar la regla de Bayes, obteniendo lo siguiente:

$$p(u|x) = \frac{p(x|u) \cdot p(u)}{\sum_{v \in U} p(x|v) \cdot p(v)} \tag{4.1}$$

En esta ecuación, U representa el conjunto de unidades fonéticas. Podemos asumir que la probabilidad a priori de todos los fonemas es la misma, es decir, que $p(u) = p(v) \forall u, v \in U$. Haciendo esta simplificación tenemos que:

$$p(u|x) = \frac{p(x|u)}{\sum_{v \in U} p(x|v)} \tag{4.2}$$

Para calcular la probabilidad $p(x|u)$ podemos emplear el concepto de clase acústica que se introdujo en el capítulo anterior. Así, una unidad fonética estaría modelada por una o varias clases acústicas. De este modo, dicha probabilidad podría expresarse según la ecuación 4.3, donde A representa el conjunto de todas las clases acústicas.

$$p(x|u) = \sum_{a \in A} p(x|a) \cdot p(a|u) \tag{4.3}$$

Por tanto, sustituyendo en la ecuación 4.2 queda:

$$p(u|x) = \frac{\sum_{a \in A} p(x|a) \cdot p(a|u)}{\sum_{v \in U} \sum_{a' \in A} p(x|a') \cdot p(a'|v)} \tag{4.4}$$

La estimación de los parámetros de las gaussianas que modelan las clases acústicas se lleva a cabo de la misma manera que se explicó en el capítulo anterior para segmentación fonética.

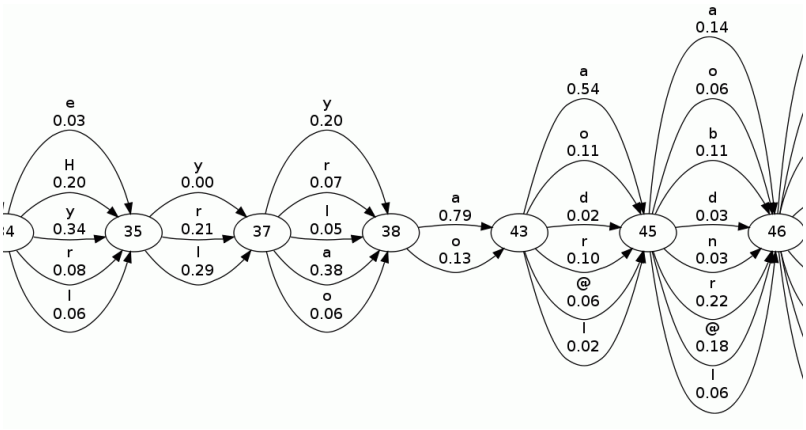

Figura 4.3: Fragmento de un grafo de fonemas.

Construcción de los grafos de fonemas

Como se comentó anteriormente, el objetivo del subsistema reconocedor que estamos presentando es proporcionar como salida un grafo en el que los nodos representen instantes de tiempo y los arcos estén etiquetados con unidades fonéticas y sus probabilidades a posteriori. En concreto, como los intervalos de tiempo que representan dos nodos consecutivos no tienen por qué ser uniformes, la probabilidad asociada a cada uno de los arcos es la media geométrica de la probabilidad a posteriori de la unidad fonética durante dicho intervalo. En la figura 4.3 puede verse un fragmento de un grafo de fonemas con la topología especificada.

Para construir este grafo se ajustan empíricamente dos umbrales: uno para detección y otro para ampliación. Para cada unidad fonética, se observa si su probabilidad supera el umbral de detección en cada uno de los instantes de tiempo. En el momento en que esto ocurre, se crea una hipótesis de segmento fonético que se extiende en ambos sentidos hasta que se cae por debajo del umbral de ampliación. Este proceso de extensión del segmento fonético determina sus puntos de inicio y final, lo que propicia, si no existían previamente, la creación de dos nodos cuyo identificador corresponde a dichos puntos. Como vamos a exigir que en nuestro grafo sólo pueda haber arcos de un nodo al siguiente, se crean tantos arcos como sean necesarios para conectar los nodos inicial y final de ese segmento fonético y se etiqueta cada uno

de ellos con la media geométrica de la probabilidad a posteriori del fonema en el subsegmento inducido por el arco.

El proceso de construcción expuesto en el párrafo anterior no garantiza que existan arcos etiquetados con cada una de las unidades fonéticas entre cada par de nodos consecutivos. De hecho, el grafo que aparece en la figura 4.3 representa justamente el resultado de aplicar el algoritmo descrito. Sin embargo, dado que disponemos de un método para calcular las probabilidades a posteriori de cualquier fonema dado un *frame*, podemos añadir al grafo los arcos que faltan de forma que entre cada par de nodos consecutivos sí haya tantos arcos como unidades fonéticas. A estos arcos se les asigna la media geométrica de la probabilidad a posteriori de la unidad fonética de la misma forma que se hizo en el algoritmo de construcción.

4.2.2. El subsistema de detección de términos

El subsistema de detección de términos toma como entrada el grafo de fonemas generado por el subsistema reconocedor y lo procesa con el objetivo de encontrar ocurrencias del término objeto de la búsqueda. Este procesamiento consiste en, partiendo desde cada uno de los nodos del grafo, intentar ubicar la secuencia fonética correspondiente al término, utilizando un algoritmo de programación dinámica para decidir en cada caso la mejor secuencia de arcos. Este algoritmo tiene también en cuenta un modelo de duración de fonemas que intenta evitar la localización de éstos durante periodos demasiado cortos. El modelo de duración que hemos utilizado está basado en la estimación de un histograma de longitudes de los fonemas, utilizando una resolución de 10 ms.

Como puede intuirse, este método de detección de términos tiene el riesgo de que puede encontrar muchas falsas ocurrencias. Para subsanar este hecho, el módulo de toma de decisiones se ha integrado con el de detección mediante dos tipos de poda. Por un lado, dado que las probabilidades a posteriori con las que están etiquetadas los arcos constituyen unas medidas de confianza, podemos establecer un umbral que represente la medida de confianza mínima que debe presentar una palabra para que sobreviva a esta primera poda.

Una forma de establecer la medida de confianza asociada a una palabra es mediante la media geométrica de las probabilidades asociadas a los arcos que han permitido detectar el término, ponderando cada una de éstas por la duración que representa el arco.

Para aplicar el segundo tipo de poda es necesario disponer de un vocabulario V. Por cada búsqueda de un término en uno de los grafos también se intentan situar en él todas las palabras de V. De esta manera es posible aplicar una poda por *n-best*: una ocurrencia detectada no es descartada si en algún momento del camino que va del nodo de inicio de la detección al final de ésta la hipótesis correspondiente al término de búsqueda ha estado entre las n mejores. En consecuencia, para que una ocurrencia del término objetivo en el grafo se considere válida no debe ser descartada por ninguno de los dos tipos de poda.

Los dos tipos de poda anteriores dependen de algún parámetro, en concreto en el primer caso del umbral mínimo de confianza que debe superarse a nivel de palabra y en el segundo del valor de n. En la experimentación que se expone a continuación se estudia cómo estos dos parámetros afectan al rendimiento del sistema.

4.3. Evaluación experimental

4.3.1. Métricas de evaluación

Para evaluar el rendimiento de nuestro sistema de *spoken term detection* para cada configuración de parámetros relativos a la poda, utilizaremos las medidas estándar conocidas como Precisión y *Recall*. La definición de cada una de ellas dado un término t puede verse en las ecuaciones 4.5 y 4.6. En ellas $Correctas(t)$ representa el número de ocurrencias acertadas para el término t, $Referencia(t)$ el número de veces que t aparece en las transcripciones de referencia y $Detectadas(t)$ es el número total de detecciones del término t emitidas por el sistema.

$$Precision(t) = \frac{Correctas(t)}{Detectadas(t)} \tag{4.5}$$

$$Recall(t) = \frac{Correctas(t)}{Referencia(t)} \tag{4.6}$$

Una manera gráfica de presentar la Precisión y el *Recall* es mediante una curva DET (*Detection Error Trade-off*). En estas curvas y para esta tarea se representa en la Precisión en el eje horizontal y el *Recall* en el vertical (ver las gráficas 4.4, 4.5 y 4.6). Además, cada experimento permite obtener un punto en este sistema de coordenadas, en concreto el correspondiente a los valores de Precisión y *Recall* que se hayan obtenido.

Para evaluar la calidad de un sistema utilizando su curva DET asociada existen varias opciones, entre las cuales se encuentran:

- Determinar el área que encierra la curva. Un sistema será mejor cuanto más se acerque el valor de esta superficie a 1. Gráficamente, un sistema será mejor cuanto más se aproxime su curva DET a los bordes superior y derecho.[1]

- Calcular el valor del EER (*Equal Error Rate*), que indica el punto de funcionamiento del sistema en el que la Precisión es igual al *Recall*. Un sistema se dice que es mejor cuanto más se acerque este valor a 1. Gráficamente este punto está representado por la intersección de la curva DET con la bisectriz del primer cuadrante (recta $y = x$).

En los experimentos que se exponen a continuación hemos basado nuestro razonamiento en la posición del EER, aunque la evaluación mediante área bajo la curva puede hacerse de forma aproximada visualmente analizando las gráficas que se adjuntan.

[1] Aunque en la figura 4.5 la gráfica no toca el borde izquierdo y por tanto parece que no pueda calcularse este área, para el cómputo de ésta se considera que el punto (0,1) (o (0,100) en porcentaje) está unido al punto más a la izquierda de la curva. Lo mismo ocurre con el punto más a la derecha y el (1,0).

4.3.2. Experimentación y resultados

Para los experimentos que aquí se exponen se ha utilizado el corpus fonético de la base de datos *Albayzin* [35]. Se ha generado un grafo de fonemas por cada uno de los ficheros de audio de este corpus y los términos a buscar en cada uno de ellos han sido todas las palabras de este corpus compuestas por más de 6 fonemas. Por otro lado, para evaluar el impacto de los parámetros de la poda en la calidad del sistema se han probado todas las combinaciones entre:

- 29 umbrales entre 1 y 50 que representan el valor de confianza mínimo que debe alcanzar la palabra para ser considerada como válida.
- 17 valores entre 1 y 50 que representan la posición mínima en el ránking considerando todas de palabras del vocabulario del corpus en el que debe quedar en algún momento de su expansión temporal la hipótesis que representa al término de búsqueda para que la ocurrencia candidata sea considerada válida. Es el valor de n en lo que hemos llamado poda por *n-best*.

Debe tenerse en cuenta que para que una ocurrencia candidata se considere válida debe superar los dos tipos de poda. Asimismo, también se han efectuado experimentos sin tener en cuenta la poda por *n-best*, es decir, tomando como válidas todas las hipótesis que superen la poda por medida de confianza.

Para poder estudiar con mayor claridad el efecto de ambos parámetros sobre el rendimiento del sistema se ha obtenido una curva DET por cada valor de n considerado para la poda por *n-best*, por lo que se han dibujado un total de 18 curvas. Además, los valores de Precisión y *Recall* que se han representado en las gráficas han sido los promedios obtenidos considerando todos los términos de test. Los puntos obtenidos se han unido en orden creciente de los umbrales para la poda por medida de confianza.

Si analizamos las figuras 4.4 y 4.5, las cuales representan respectivamente el comportamiento del sistema sin aplicar poda por *n-best* y podando por

4-*best*, vemos que el EER en la primera se sitúa en torno al 65 %, mientras que en la segunda es de aproximadamente el 70 %. Esto nos indica que el método de poda por *n-best* es efectivo. De hecho, en la figura 4.6, en la que se muestran las curvas DET obtenidas para una selección de valores de n junto con la calculada para el experimento sin poda por *n-best*, puede comprobarse que los mejores valores de EER se obtienen para $n = 4$, seguido de $n = 5$ y $n = 6$. A partir de ese valor el EER comienza a decrecer, hasta que con valores mayores que $n = 12$ se consiguen valores equiparables a no hacer poda por *n-best*. Los valores del EER de alrededor del 70 % que se alcanzan utilizando la poda por *n-best* son superiores a los que otros autores obtienen en otros trabajos, utilizando conocimiento léxico y sintáctico pero con diferentes corpora de voz [2, 44].

La comparación de las figuras 4.4 y 4.5 también nos aporta otro dato relevante: para umbrales bajos de la poda por medida de confianza los valores de Precisión obtenidos son mucho mejores utilizando poda por n-best, a pesar de tener una pérdida en el *Recall*. Esto quiere decir que en estos puntos de funcionamiento el sistema es capaz de rechazar mucho mejor los falsos positivos utilizando poda por *n-best*, pero a costa de perder también algunas ocurrencias correctas de los términos de búsqueda.

Por otro lado, en la figura 4.6 puede verse un comportamiento un tanto anómalo del sistema, para todos los valores de n considerados, a partir de umbrales de alrededor del 33 % para la poda por medida de confianza. Para estos valores el *Recall* sigue disminuyendo tal y como se esperaba pero para algunos valores del umbral sorprendentemente la Precisión también disminuye. Esto quiere decir que para dichos umbrales el número de ocurrencias verdaderas de los términos de búsqueda que se deja de detectar es mayor que el incremento de falsos positivos que se rechazan, por lo que el rendimiento del sistema empeora sensiblemente. De todos modos, aunque este comportamiento no es deseable, tampoco tiene demasiada importancia ya que este empeoramiento del rendimiento tiene lugar bajo unas condiciones (valores de los parámetros de los mecanismos de poda) muy lejanas al punto de funcionamiento que para un sistema de este tipo sería razonable, el cual estaría situado alrededor del EER.

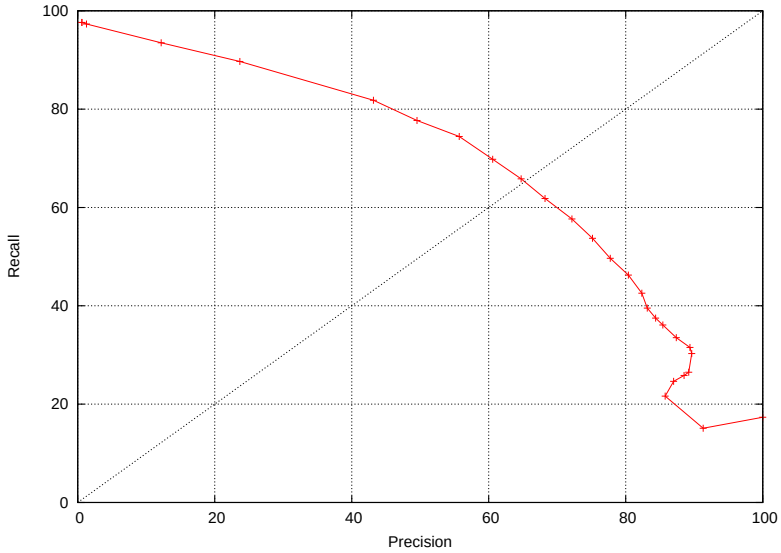

Figura 4.4: Curva DET que representa el *Recall* medio frente a la Precisión media sin aplicar poda por *n-best*.

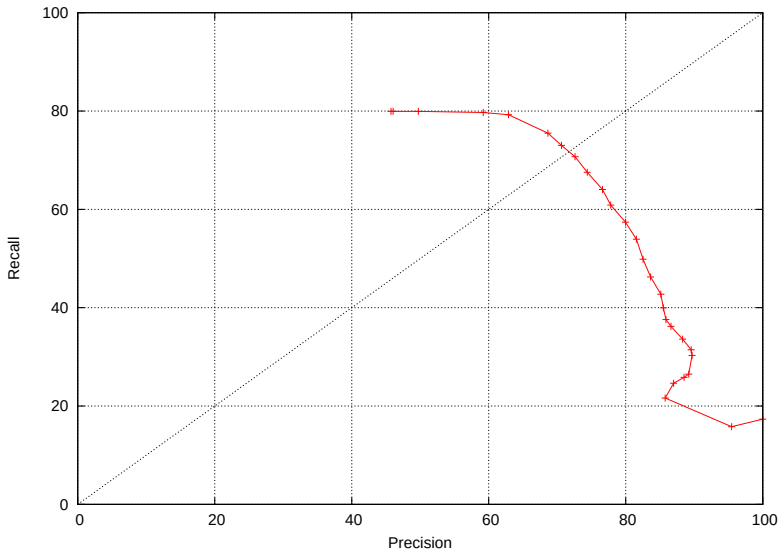

Figura 4.5: Curva DET que representa el *Recall* medio frente a la Precisión media aplicando poda por 4-*best*.

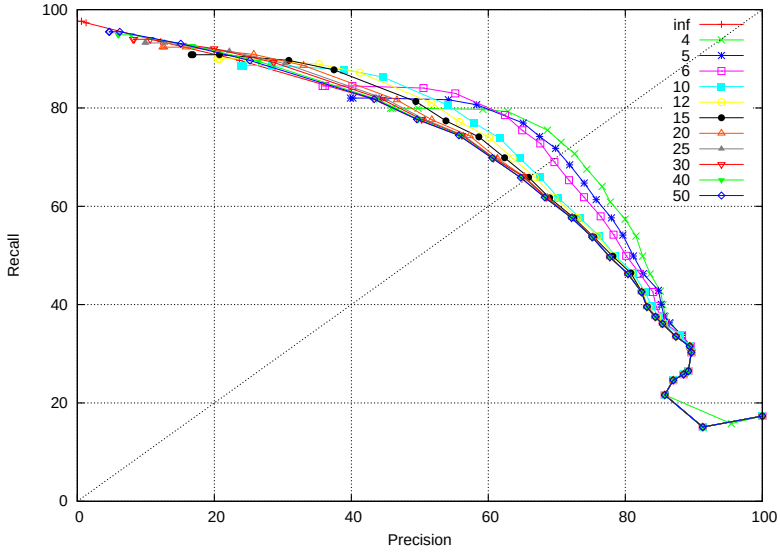

Figura 4.6: Curva DET que representa el *Recall* medio frente a la Precisión media aplicando poda por *n-best* para distintos valores de n. Inf representa que no se ha aplicado este tipo de poda.

Capítulo 5

Una aproximación a la comprensión del habla basada en grafos de palabras

5.1. Definición de la tarea

La comprensión automática del habla es el proceso por el que, dada una pronunciación emitida por un locutor, se extrae una interpretación semántica de la información contenida en ésta basada en un conjunto de conceptos definido a priori. El ámbito donde los sistemas de comprensión del habla tienen mayor aplicación práctica es el de los sistemas de diálogo hablado, en los cuales es crucial que el sistema pueda extraer la información asociada a la pronunciación de entrada (la "comprenda") para devolver una respuesta coherente con lo que ha dicho el usuario. La estructura típica de un sistema de diálogo hablado se muestra en la figura 5.1. Por tanto, asumiendo que al módulo de comprensión le llega una estructura de datos en la que está representada la información que ha extraído el módulo de reconocimiento, la comprensión del habla consiste de dos subtareas, que son:

1. La identificación de la secuencia de categorías semánticas (también llamadas conceptos) y su asignación a secuencias de palabras.

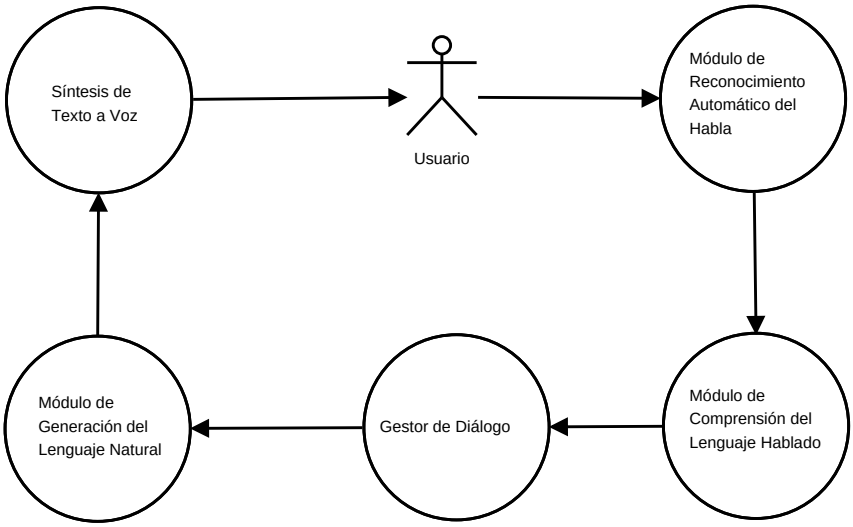

Figura 5.1: Esquema de un sistema de diálogo hablado.

2. La extracción de la información relativa a estos conceptos codificada en las secuencias de palabras asignadas y la construcción de estructuras de datos que representen dicha información.

En el resto del capítulo nos centraremos en abordar la subtarea correspondiente a la identificación de la secuencia de categorías semánticas, aunque la segunda de las subtareas es también de gran importancia.

5.2. Un método para la comprensión del habla basado en grafos de palabras

Desde el punto de vista estadístico, la tarea de la identificación de la secuencia de categorías semánticas puede abordarse utilizando tanto modelos generativos como discriminativos. En el primer caso la distribución de probabilidad que pretende modelarse es la conjunta $\Pr(W, C)$, donde W es la secuencia de palabras que componen la frase y C es la secuencia de conceptos asociada a la de palabras. Como puede intuirse, ambas secuencias no tienen por qué tener la misma longitud, dado que cada concepto puede abarcar más de una palabra. Por otro lado, los modelos discriminativos para comprensión

toman como base la distribución de probabilidad $\Pr(C|W)$. El algoritmo de comprensión basado en grafos de palabras que aquí se presenta está basado en la utilización de modelos generativos, y en concreto en el paradigma basado en transductores de estados finitos.

5.2.1. El paradigma de comprensión del habla basado en transductores

El paradigma de comprensión del habla basado en transductores [15, 42] se fundamenta en la utilización de este formalismo para calcular la secuencia de conceptos $\hat{C}$ tal que cumple la ecuación 5.1, en la que A representa la pronunciación de entrada al sistema.

$$\hat{C} = \underset{C}{\operatorname{argmax}} \Pr(C|A) \tag{5.1}$$

Introduciendo como una variable aleatoria más la secuencia de palabras W y aplicando la regla de Bayes tenemos que:

$$\hat{C} = \underset{C}{\operatorname{argmax}} \frac{\sum_W \Pr(A|W,C) \cdot \Pr(W,C)}{\Pr(A)} \tag{5.2}$$

Dado que no depende del término de la maximización, podemos eliminar el denominador de la fracción de la ecuación 5.2. Además, es posible efectuar la consideración usual de que la suma para toda secuencia de palabras W queda convenientemente aproximada por su máximo. Una última asunción razonable es considerar que la acústica sí depende de la palabra, pero no de la categoría semántica a la que ésta pertenezca. Todas estas simplificaciones nos llevan a la ecuación 5.3.

$$\hat{C} = \underset{C}{\operatorname{argmax}} \underset{W}{\operatorname{máx}} \Pr(A|W) \cdot \Pr(W,C) \tag{5.3}$$

Para calcular la mejor secuencia de conceptos expresada de la forma de la ecuación 5.3 este paradigma propone buscar el mejor camino en un transductor λ_{SLU} resultado de componer 4 transductores:

$$\lambda_{SLU} = \lambda_G \circ \lambda_{gen} \circ \lambda_{W2C} \circ \lambda_{SLM} \tag{5.4}$$

Donde

- λ_G es una representación de la pronunciación de entrada en forma de máquina de estados finitos en la que están especificadas las probabilidades acústicas $p(A|W)$ calculadas por el módulo reconocedor.
- λ_{gen} sustituye algunas de las palabras por sus categorías léxicas (por ejemplo ciudades, horas o números). Representa el conocimiento que se tiene a priori de la tarea a abordar y permite generalizar a partir de los datos de entrenamiento.
- λ_{W2C} convierte secuencias de palabras a conceptos.
- λ_{SLM} codifica un modelo de lenguaje de categorías semánticas, el cual nos permite finalmente estimar la probabilidad conjunta $p(W, C)$.[1]

La búsqueda del mejor camino en el transductor permite obtener la secuencia de conceptos que maximiza la ecuación 5.3, ya que la probabilidad $p(A|W)$ ya viene codificada en λ_G y la composición de λ_{W2C} y λ_{SLM} nos permite calcular $p(W, C)$.

5.2.2. Una aproximación basada en algoritmos sobre grafos

La aproximación que aquí se presenta está basada en la idea de la aplicación de transformaciones sucesivas que propone el paradigma basado en transductores para obtener la secuencia de conceptos que maximiza la ecuación 5.3. La entrada para nuestro método será un grafo de palabras con una topología determinada, que deberá ser suministrado por el módulo de

[1] Realmente no es exactamente esta probabilidad la que se calcula, sino una aproximación teniendo en cuenta que algunas de las palabras han sido sustituidas por sus categorías léxicas.

reconocimiento. Este grafo será procesado en dos pasos (uno de los cuales será la obtención de otro grafo) para obtener como salida la secuencia de conceptos $\hat{C}$. Como consecuencia de la búsqueda de $\hat{C}$ también se consigue la secuencia de palabras $\tilde{W}$, la cual debe interpretarse como la secuencia de palabras que nos permite encontrar la secuencia de conceptos de mayor probabilidad. Hay que tener en cuenta que esta secuencia de palabras no tiene por qué coincidir con la decodificación 1-*best* que proporcionaría un ASR.

A continuación se expondrá en primer lugar la topología que debe cumplir un grafo de palabras para poderse utilizar en el sistema que aquí se propone, junto con la explicación del significado de sus nodos y arcos. Seguidamente, detallaremos los dos pasos en los que se divide el algoritmo de cálculo de la mejor secuencia de conceptos, los cuales se corresponden con la construcción de un grafo de segmentos y su posterior decodificación.

Topología y semántica del grafo de entrada

Para que un grafo pueda utilizarse como entrada para el algoritmo de búsqueda de la mejor secuencia de conceptos debe cumplir las siguientes condiciones:

- Sus nodos han de representar instantes de tiempo y deben estar etiquetados con ellos (son sus identificadores).[2]
- No debe existir ningún nodo cuyo grado tanto de entrada como de salida sea 0. En otras palabras, para todos los nodos debe haber algún arco que entre o salga de él.
- Dados dos nodos con identificadores temporales i y j con $i < j - 1$, existirá un arco del nodo i al nodo j etiquetado con la palabra w si el sistema de reconocimiento ha detectado que dicha palabra ha podido ser pronunciada comenzando en el instante de tiempo i y terminando en el $j-1$. Además, el peso asignado a estos arcos se corresponderá con el *score* acústico que el ASR haya dado a esta ocurrencia de w.

[2]La semántica de los nodos es la misma que en el sistema de *spoken term detection* presentado en el capítulo anterior.

- Para evitar que se dé el caso de que no exista ningún camino en el grafo que una los nodos inicial y final, se permite la existencia de una λ-transición entre cada par de nodos consecutivos (siempre teniendo como origen el nodo anterior), asignándosele a ésta un peso calculado mediante un tipo de suavizado. Se considera nodo inicial del grafo aquél que tiene el menor identificador temporal y final el que tiene el mayor. Asimismo, decimos que los nodos i y j con $i < j$ son consecutivos si no existe ningún nodo etiquetado con k y distinto de estos dos tal que $i < k < j$.

Existen dos diferencias relevantes entre los grafos de palabras aquí expuestos y las *lattices* definidas en el capítulo 2. En primer lugar, en las *lattices* no existen arcos sin palabras asociadas (λ-transiciones), mientras que en estos grafos este método de suavizado los hace más expresivos. Por otro lado, mientras que en la definición general de las *lattices* se permite que los pesos asociados a los arcos puedan haberse calculado teniendo en cuenta el modelo de lenguaje, en los grafos que utilizaremos para nuestro algoritmo la probabilidad que deben tener asociada es únicamente la acústica. De este modo se consigue modelar correctamente la probabilidad $p(A|w)$, siendo A el fragmento de audio correspondiente al intervalo delimitado por los identificadores de los nodos inicial y final del arco y w la palabra asociada a éste. Esta probabilidad es similar a la expresada con el primer tipo de transductor presentado en la sección anterior.

Construcción del grafo de segmentos

Como paso intermedio para la decodificación de la mejor secuencia de segmentos construiremos un grafo, al que hemos denominado "grafo de segmentos", en el que:

- El conjunto de nodos es el mismo que el del grafo de palabras de entrada y mantienen sus identificadores.
- Al igual que en el grafo de palabras, desde un nodo sólo podrán partir arcos que alcancen nodos posteriores en el tiempo, es decir, dado un

nodo i, los destinos de los arcos que partan de él sólo podrán ser nodos j tales que $i < j$.

- Cada arco estará etiquetado con un par (W, c), donde W es una secuencia de palabras y c es el concepto que éstas representan. El peso asociado a cada arco deberá aproximarse a $p(A|W) \cdot p(W|c)$, donde A es el fragmento de audio correspondiente al intervalo limitado por los identificadores de los nodos inicial y final del arco.

- Dados dos nodos del grafo de segmentos i y j con $i < j$, sólo se permite que exista un arco etiquetado con el concepto c cuyo origen sea i y su destino j. La secuencia de palabras que deberá tener asignada será la que maximice la probabilidad $p(A|W) \cdot p(W|c)$ en el intervalo temporal $[i, j[$.

En esta especificación del grafo de segmentos vemos que aparece un nuevo tipo de probabilidad, en concreto $p(W|c)$, donde W es una secuencia de palabras y c un concepto. Para aproximar esta probabilidad puede estimarse un modelo de lenguaje de n-gramas para cada concepto con los datos de entrenamiento disponibles. Para que esto sea posible se necesita que las frases de entrenamiento estén segmentadas en fragmentos que denoten conceptos y se haya adjuntado a éstos la etiqueta del concepto correspondiente.

El grafo de segmentos que pretendemos obtener puede generarse como resultado de un algoritmo de programación dinámica que encuentre, para cada concepto c y cada par de nodos i y j con $i < j$, el mejor camino en el grafo de palabras de entrada que esté formado únicamente por arcos etiquetados con palabras asignadas a c, utilizando λ-transiciones si fuera necesario. En este caso, con mejor camino en el grafo entendemos el camino que maximiza la probabilidad resultado de combinar las probabilidades acústicas $p(A|w)$ expresadas en el grafo de palabras y las probabilidades del modelo de lenguaje del concepto $p(W|c)$, donde $W = w_1 \dots w_n$ es la secuencia de palabras resultante de concatenar las etiquetas w_k de los arcos del grafo de palabras que forman el camino.

Este algoritmo de construcción explota la topología izquierda - derecha del grafo de palabras sobre el que se basa. De este modo las hipótesis que el

algoritmo de programación dinámica va generando sólo dependen de nodos que se han procesado ya y que no se van a volver a considerar. Además, para cada nodo cada uno de los conceptos se considera por separado, explotando así el hecho de que se dispone de un modelo de lenguaje por concepto. Esto permite además que para controlar el número máximo de hipótesis que pueden asociarse a cada par (nodo, concepto) sea posible utilizar una maximización local por estado del modelo de lenguaje.

Al igual que ocurre en la mayor parte de sistemas de procesamiento del habla, se han introducido algunos parámetros que deben ajustarse empíricamente y que permiten mejorar el rendimiento del sistema. En concreto, son cuatro constantes cuyo significado es:

- Un *Grammar Scale Factor*, que es la constante a la que se elevan las probabilidades del modelo de lenguaje con el objetivo de darle a éste más o menos "importancia".
- Un *Word Insertion Penalty*, que es un valor por el que se multiplica la probabilidad acumulada cada vez que se inserta una nueva palabra en un segmento. Este valor puede ser mayor o menor que 1 según se quiera favorecer o penalizar respectivamente la inserción de nuevas palabras.
- El número máximo de λ-transiciones consecutivas sobre el grafo de palabras que se permite que el algoritmo de construcción del grafo de segmentos pueda utilizar.
- Una constante a la que se elevarán los pesos de las λ-transiciones. Este valor pretende regular la importancia de las λ-transiciones con respecto a los arcos etiquetados con palabras, ya que si a los pesos de estos últimos se les aplica un *Grammar Scale Factor* es lógico que a las λ-transiciones también se les aplique una cierta ponderación.

La última de estas constantes implica que se deben replicar (debidamente penalizadas) las hipótesis que llegan a un nodo en los k nodos siguientes, siendo k el número máximo de λ-transiciones que pueden aplicarse. Este hecho incrementa el coste asintótico del algoritmo, pero permite hacer uso de un suavizado que de otra forma no sería posible.

También es interesante observar que este proceso podría estar controlado por algún tipo de poda local a cada nodo y cada concepto, del estilo de un *beam* o un *histogram pruning*. Aunque la implementación del algoritmo que se ha efectuado está lista para soportarla, en los experimentos que se detallarán en la sección 5.3 no se ha aplicado ningún tipo de poda.

Puede observarse que este proceso de construcción del grafo de segmentos tiene una correspondencia con el transductor λ_{W2C} descrito anteriormente, ya que nuestro objetivo aquí es determinar secuencias de palabras válidas en el grafo de entrada que estén asociadas a alguno de los conceptos disponibles. De este modo obtenemos correspondencias entre secuencias de palabras y conceptos enriquecidas con datos sobre sus instantes de inicio y final, lo que nos permite seguir representando la información en forma de grafo.

Destacar también que previamente al proceso de construcción del grafo de segmentos podría utilizarse una generalización por categorías léxicas (números, meses, etc.) al estilo de lo especificado por el transductor λ_{gen}. En este caso los modelos de lenguaje asociados a cada concepto deberían entrenarse teniendo en cuenta dichas categorías léxicas.

Decodificación sobre el grafo de segmentos

Una vez se ha obtenido el grafo de segmentos en el paso anterior, estamos ya en condiciones de buscar la secuencia de conceptos que maximice la ecuación 5.3. Para ello necesitaremos un modelo de lenguaje de conceptos, que puede estar basado en n-gramas, de forma que nos permita estimar la probabilidad $p(C)$, siendo C una secuencia de conceptos. Para construir este modelo deberá disponerse de las secuencias de conceptos correspondientes a las frases del corpus de entrenamiento.

Para entender mejor cómo funciona este algoritmo, descompondremos el segundo término de la ecuación 5.3, quedando:

$$\hat{C} = \underset{C}{\operatorname{argmax}} \max_{W} \Pr(A|W) \cdot \Pr(W|C) \cdot \Pr(C) \tag{5.5}$$

Como se expuso en la definición del algoritmo de construcción, cada arco

del grafo de segmentos está etiquetado con una secuencia de palabras W tal que maximiza la probabilidad $p(A|W) \cdot p(W|c)$ en el intervalo de tiempo inducido por sus nodos de origen y destino, siendo c el concepto asociado al arco. Esta maximización se corresponde con la maximización interna de la ecuación 5.5, considerando un concepto aislado en lugar de una secuencia de conceptos. Gracias al modelo de lenguaje de conceptos podemos estimar la probabilidad de una secuencia de conceptos $p(C)$, y por tanto completar el cálculo anterior y obtener la secuencia $\hat{C}$ que maximiza la ecuación. Esta forma de proceder tiene su correspondencia con el transductor λ_{SLM} presente en el paradigma de comprensión utilizando transductores.

Una forma de encontrar el camino en el grafo de segmentos que maximiza el producto $p(A|W) \cdot p(W|C) \cdot p(C)$ es mediante el algoritmo de Viterbi. De esta manera, para cada nodo del grafo sólo nos quedaremos con la hipótesis de mayor probabilidad para cada estado del modelo de lenguaje de conceptos. Una restricción que se impone sobre dicho camino es que deberá comenzar en el nodo que representa el inicio de la pronunciación y terminar en el último nodo del grafo de segmentos.

Al igual que en el algoritmo de construcción, en este algoritmo de decodificación también se han definido algunos parámetros que deben ajustarse empíricamente para mejorar su rendimiento. En concreto, en este caso estos parámetros son dos y corresponden a un *Grammar Scale Factor* de forma que pueda ajustarse convenientemente la importancia del modelo de lenguaje de categorías con respecto a los pesos del grafo de segmentos y un *Word Insertion Penalty*, el cual permite controlar si queremos favorecer o penalizar la inserción de nuevos conceptos en la secuencia.

Una vez aplicado el algoritmo de Viterbi y obtenido el mejor camino, la concatenación de los conceptos asociados a cada uno de los arcos pertenecientes al camino constituye la secuencia de conceptos $\hat{C}$ que estábamos buscando, es decir, la secuencia de conceptos a los que hace referencia la pronunciación. Además, si concatenamos las secuencias de palabras que dichos arcos llevan asociadas también obtenemos $\tilde{W}$, secuencia que, como ya se comentó, no tiene por qué coincidir con la que daría como resultado un ASR. Por último, si tomamos por separado la información asignada a los arcos del

camino, obtenemos una segmentación de la frase $\tilde{W}$ junto con sus conceptos asociados, lo cual es muy útil para un posterior proceso de extracción de información sobre los conceptos.

5.3. Experimentación y resultados

Para realizar una experimentación que nos permita evaluar el método que se ha presentado, hemos utilizado el corpus DIHANA [1, 3, 4]. Este es un corpus de diálogo de habla espontánea en castellano en el que todos sus diálogos son telefónicos y tratan sobre recuperación de información sobre viajes en tren. Esto implica que los ficheros de audio tendrán calidad telefónica. Algunos datos interesantes al respecto de este corpus se muestran en la tabla 5.1.

Número de turnos	6226
Número de palabras	47222
Talla del vocabulario	811
Media de palabras por turno de usuario	7,6
Número de conceptos	30

Tabla 5.1: Características del corpus DIHANA

Se dispone, además, de una segmentación de este corpus de forma que todas sus frases están divididas en fragmentos correspondientes a conceptos y etiquetados convenientemente. También se dispone de la secuencia de conceptos correspondiente a cada frase de entrenamiento, por lo que es posible entrenar el modelo de lenguaje de conceptos. De las frases disponibles en el corpus se han utilizado 4885 para entrenamiento y 1340 para test.

Por otro lado, todos los modelos de lenguaje estimados (tanto los de palabras correspondientes a un concepto como el de conceptos) han sido modelos de bigramas para cuyo entrenamiento se ha utilizado el suavizado de Witten-Bell con interpolación lineal.

La entrada al sistema para esta experimentación ha estado constituida por 1340 grafos de palabras (uno por cada frase de test), calculados mediante una

extensión del algoritmo presentado en el capítulo 4 para la construcción de los grafos de fonemas. El peso asignado a las λ-transiciones se ha estimado a partir de la confusión acústica observada considerando el conjunto de *frames* correspondientes al intervalo temporal inducido por los nodos inicial y final de la transición. El *Word Error Rate* (WER) del oráculo calculado para estos grafos es del 16,598 %, entendiendo como tal el WER obtenido al buscar la frase de referencia en el grafo, obteniendo en caso de no estar ésta la de menor distancia de edición a nivel de palabra.

Para evitar el sobreentrenamiento, y dado que hay que ajustar 6 parámetros correspondientes a los definidos para los algoritmos de construcción y decodificación del grafo de segmentos, se han tomado 134 frases del conjunto de test (el 10 %) como conjunto de desarrollo. De este modo, se probarán sobre estas frases diferentes combinaciones de valores para los 6 parámetros y aquella configuración para la que se obtengan mejores resultados será con la que se evalúe el sistema utilizando el conjunto de 1340 frases, el 90 % de las cuales permanecerán ocultas hasta el final.

Las medidas de error que se han tomado en estos experimentos han sido el *Word Error Rate* (WER) y el *Concept Error Rate* (CER). La definición de este último es análoga a la del WER pero tomando como unidad fundamental el concepto en lugar de la palabra. A pesar de que se hayan tomado dos medidas diferentes del error debe tenerse en cuenta que la que nos interesa minimizar es el CER, que es la que se corresponde con la secuencia de conceptos que obtenemos a partir de la ecuación 5.3.

Con el objetivo de ajustar el conjunto de parámetros, se han realizado numerosos experimentos divididos en varias tandas, en cada una de las cuales se han probado diferentes configuraciones para estos parámetros. La estrategia empleada ha consistido en fijar en cada tanda un subconjunto de los parámetros a unos ciertos valores y variar el resto en un determinado rango, con el objetivo de intentar mejorar el menor CER que se haya obtenido hasta el momento. Tras cada tanda se determina la configuración para la que se ha obtenido el mejor resultado y, partiendo desde dicha configuración, se repite el proceso anterior. La configuración de parámetros inicial del proceso se ha establecido de forma arbitraria y el procedimiento se ha repetido hasta que

se ha sido posible mejorar el CER.

En las tablas 5.2 y 5.3 se muestran dos subconjuntos de los resultados obtenidos en dos de las últimas tandas de experimentos. En las tablas las abreviaturas GSFc y GSFd representan respectivamente los *Gramar Scale Factors* del algoritmo de construcción y decodificación y WIPc y WIPd los correspondientes *Word Insertion Penalties*. Además maxLambdas indica el número máximo permitido de λ-transiciones consecutivas y penLambda es el factor de castigo al que se eleva el peso de la λ-transición. Hay que tener en cuenta que todos los cálculos se han efectuado en los distintos algoritmos con los logaritmos de las probabilidades, por lo que los valores de los parámetros expresados en las tablas son los que se aplicarían a éstos.

La configuración que obtiene el mejor valor en la tabla 5.3 es la que se tomó como definitiva para esta experimentación. Utilizando esta configuración se ha obtenido con el test completo (1340 frases) un CER=41,827 % y un WER=56,082 %.

A modo de ejemplo, a continuación mostramos toda la información que es capaz de dar nuestro sistema de comprensión como salida. Para este ejemplo se ha seleccionado la frase *"Sí, quería saber el horario del viaje de ida."*, la cual ha sido correctamente reconocida por el sistema al buscar la mejor secuencia de conceptos.

```
Puntuación del mejor camino = -564.886087
Secuencia de segmentos del mejor camino:
        Probabilidad del arco: -57.836728
        Bolsa a la que pertenece el segmento: _afirmacion_ (0)
        s'i
---------------------
        Probabilidad del arco: -133.407823
        Bolsa a la que pertenece el segmento: consulta (6)
        quer'ia saber
---------------------
        Probabilidad del arco: -201.642419
        Bolsa a la que pertenece el segmento: _hora_ (11)
        el horario de un viaje
```

GSFd	WIPd	WER	CER
44	0	62,556	41,080
44	-2	62,556	41,080
44	-4	62,556	40,845
44	-6	62,916	41,784
48	**0**	**62,376**	**40,141**
48	-2	62,196	40,376
48	-4	62,646	41,549
48	-6	62,646	41,549
52	0	62,106	40,610
52	-2	62,196	41,315
52	-4	62,376	41,784
52	-6	62,466	42,019
56	0	62,106	41,549
56	-2	62,196	41,549
56	-4	62,016	41,315
56	-6	62,196	41,315

Tabla 5.2: Resultados obtenidos para una tanda de experimentos en los que se han mantenido fijos los parámetros GSFc=16; WIPc=0; penLambda=2,4 y maxLambdas=36.

```
---------------------
        Probabilidad del arco: -92.865886
        Bolsa a la que pertenece el segmento: tipo_viaje (29)
        de ida
---------------------
```

Un aspecto que debemos comentar es que en algunos casos el algoritmo de decodificación no es capaz de dar ninguna salida, y esto es debido a que en el grafo de segmentos que se construye no existe ningún camino que permita llegar del nodo inicial al final. En concreto, para las 134 frases de desarrollo esto ocurre en 3 de ellas (un 2,24 %) y para el conjunto de 1340 ocurre en 18 (un 1,34 %). Hay varios factores que hacen que este efecto no deseado ocurra. En primer lugar, es debido a que el número máximo de λ-transiciones

WIPc	WIPd	WER	CER
4	4	57,066	40,845
4	3	57,516	40,610
4	2	57,696	40,610
3	4	58,416	41,784
3	3	58,416	41,549
3	2	58,326	41,315
2	4	58,776	39,906
2	3	58,855	39,202
2	**2**	**58,866**	**38,967**
1	4	59,586	39,906
1	3	59,676	39,671
1	2	59,766	40,141

Tabla 5.3: Resultados obtenidos para una tanda de experimentos en los que se han mantenido fijos los parámetros GSFc=16; GSFd=30; penLambda=2,4 y maxLambdas=36.

consecutivas permitidas que se está considerando es insuficiente para que puedan decodificarse todas los grafos de entrada, ya que se ha constatado que a medida que aumenta este valor el número de grafos que no se pueden decodificar es menor. Sin embargo, como ya se comentó, el número máximo de λ-transiciones consecutivas es un factor que afecta al coste computacional del algoritmo, por lo que en este caso se ha optado por un compromiso entre el número de frases no decodificadas y la velocidad del sistema. Por otro lado, es posible que en algunos casos la detección de palabras o silencios les haya asignado unos instantes de inicio o finalización posteriores o anteriores (respectivamente) a los reales. El hecho de que existan frases para las que no se da ninguna salida es un aspecto que debe corregirse en un futuro para disminuir el WER y el CER obtenidos.

Otro hecho relevante que puede observarse en las tablas anteriores que en todos los casos el CER es inferior al WER. Esto indica que, a pesar de que se fallen palabras, en muchas ocasiones puede encontrarse información relativa a los conceptos a pesar de que se cometan errores en las palabras asociadas.

Este hecho es favorable siempre y cuando las palabras que se fallen sean aquellas que no afecten en gran medida al significado de la frase pronunciada, ya que así el posterior módulo de recuperación de información asociada a los conceptos podría funcionar correctamente. Experimentalmente se ha podido comprobar que muchas veces las palabras que se fallan son conjunciones, artículos, etc. (en general, *stopwords*) que aportan poca información. Sin embargo, en otras ocasiones las palabras que no han podido reconocerse sí tienen un significado relevante, y este es un hecho que debería mejorarse para un futuro.

Capítulo 6

Conclusiones y trabajo futuro

En este trabajo se han presentado tres aproximaciones a tres ámbitos del procesamiento del habla como son la detección de fronteras fonéticas, la localización de términos en documentos de audio y la comprensión automática del habla. En los tres casos las aproximaciones aquí descritas están basadas en estructuras en forma de grafo o derivadas de éstas y en algoritmos sobre ellas.

En el caso de la detección de fronteras fonéticas la aproximación propuesta está basada en modelos ocultos de Markov, cuyas probabilidades de emisión se estiman a partir de un proceso de *clustering* paramétrico a nivel acústico y cuya topología tiene ciertos estados a cada lado del modelo en los que no se permiten bucles. Además, la forma de cálculo de las probabilidades de emisión que se propone hace que no sea necesaria una segmentación manual para entrenamiento. Los resultados experimentales presentados son similares a los de otros sistemas que sí utilizan la segmentación manual. Por otro lado, si se utilizan como unidades fonéticas adicionales las transiciones entre fonemas y se emplea la segmentación manual para inicializar el proceso de entrenamiento, los resultados que se consiguen son incluso superiores.

Para esta tarea queda como trabajo futuro la investigación de un método más apropiado que el aquí expuesto para la determinación de las fronteras fonéticas cuando las transiciones entre fonemas se consideran como unidades fonéticas. Asimismo, otra línea que podría explorarse es el hecho de modelar

cada unidad fonética con una topología diferente, y no sólo aquellas que por sus características fonéticas deben tener un menor número de estados.

Por otra parte, para la tarea de *spoken term detection* se ha presentado un sistema basado en grafos de fonemas, cuyos arcos están etiquetados con las probabilidades a posteriori de éstos durante el periodo que abarca. El algoritmo de búsqueda propuesto está basado en una estrategia de programación dinámica y para controlar la tasa de falsos positivos se aplican dos tipos de poda: una basada en la probabilidad a posteriori media del término y otra en la lista de *n-best* hipótesis en un instante de tiempo determinado considerando todas las palabras de un vocabulario que debe proporcionársele al sistema. Los resultados presentados, en términos de EER, son similares a los conseguidos por otros autores, pero en nuestro caso se ha utilizado un corpus de voz diferente. Para comprobar la efectividad de nuestro sistema frente al de otros autores queda como trabajo futuro la experimentación con otros corpora más utilizados en la literatura.

Por último, se ha abordado la tarea de comprensión automática del habla, tomando como entrada un grafo de palabras con una topología determinada, el cual representa una pronunciación. Este grafo se procesa en dos pasos, inspirados en el paradigma de comprensión del habla basado en transductores de estados finitos, a lo largo de los cuales se emplean dos tipos diferentes de modelos de lenguaje. Como resultado de este proceso se obtiene una secuencia de conceptos y, de forma colateral, una secuencia de palabras.

Los resultados experimentales obtenidos hasta ahora parecen presentar un error bastante elevado, aunque hay que tener en cuenta que el objetivo principal es extraer la información semántica de los grafos que proporciona un sistema de reconocimiento, aunque esta salida tenga errores. Hay que considerar que lo expuesto en este trabajo constituye una primera aproximación, por lo que pensamos que este error puede mejorarse bastante. Por ejemplo, una forma de intentar disminuir este error sería solucionando los casos donde el sistema no da ninguna salida, tal vez utilizando también un mecanismo de suavizado basado en λ-transiciones en el grafo de segmentos. Otra forma de intentar minimizar este error y cuya exploración queda como trabajo futuro es la utilización de categorías léxicas para utilizar el conocimiento que a pri-

ori se tiene de la tarea. También queda abierta la posibilidad de utilizar otros tipos de modelos de lenguaje para representar las relaciones entre palabras dentro de un concepto, así como las relaciones entre conceptos.

Capítulo 7

Publicaciones relacionadas

Relacionadas con este trabajo se han editado tres publicaciones en las que el autor ha formado parte. Estas son:

- J.A. Gómez, M. Calvo. Improvements on Automatic Speech Segmentation at the Phonetic Level. *Aceptado para su publicación en CIARP 2011, Pucón, Chile*. Springer, 2011. **CORE C**
- E. Segarra, L. Hurtado, J.A. Gómez, F. García, J. Planells, J. Pastor, L. Ortega, M. Calvo, E. Sanchis. A prototype of a spoken dialog system based on statistical models. *Proceedings of FALA 2010: VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop*, pages 243-246.
- J.A. Gómez Adrián, M. Calvo Lance, E. Sanchis. Localización de Palabras basada en Grafos de Fonemas. *Procesamiento del Lenguaje Natural*, 44:59-66, 2010.

Bibliografía

[1] Nieves Alcacer, María José Castro, Isabel Galiano, Ramón Granell, Sergio Grau, and David Griol. Diseño de un corpus de diálogo: DIHANA. In *Actas de las III Jornadas en Tecnologías del Habla*, pages 131–135, Valencia (Spain), 2004.

[2] A. Amir, A. Efrat, and S. Srinivasan. Advances in phonetic word spotting. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 580–582. ACM, 2001.

[3] José-Miguel Benedí, Eduardo Lleida, Amparo Varona, María-José Castro, Isabel Galiano, Raquel Justo, Iñigo López de Letona, and Antonio Miguel. Design and acquisition of a telephone spontaneous speech dialogue corpus in Spanish: DIHANA. In *Proceedings of LREC 2006*, pages 1636–1639, Genoa (Italy), May 2006.

[4] J.M. Benedí, A. A. Varona, and E. Lleida. DIHANA: Sistema de diálogo para el acceso a la información en habla espontánea en diferentes entornos. In *Actas de las III Jornadas en Tecnología del Habla*, pages 141–146, Valencia (España), 2004.

[5] N. Bertoldi and M. Federico. A new decoder for spoken language translation based on confusion networks. In *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, pages 86–91. IEEE, 2005.

[6] M.J. Castro, S. España, A. Marzal, and I. Salvador. Grapheme-to-phoneme conversion for the spanish language. In *Proceedings of the IX National Symposium on Pattern Recognition and Image Analysis*, pages 397–402, 2001.

[7] C. Chelba and A. Acero. Position specific posterior lattices for indexing speech. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 443–450. Association for Computational Linguistics, 2005.

[8] M. Dinarelli. *Spoken Language Understanding: from Spoken Utterances to Semantic Structures*. PhD thesis, 2010.

[9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

[10] J.G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 347–354. IEEE, 1997.

[11] J.G. Fiscus, J. Ajot, J.S. Garofolo, and G. Doddingtion. Results of the 2006 spoken term detection evaluation. *Searching Spontaneous Conversational Speech*, pages 51–57.

[12] J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallett, and N. Dahlgren. Timit acoustic-phonetic continuous speech corpus linguistic data consortium. *Philadelphia, PA*, 1993.

[13] J. Gómez and M. Castro. Automatic segmentation of speech at the phonetic level. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 883–921, 2002.

[14] Jon Ander Gómez Adrián. Automatic phonetic segmentation. In *Proceedings of FALA 2010*, pages 123–126, 2010.

[15] S. Hahn, M. Dinarelli, C. Raymond, F. Lefèvre, P. Lehnen, R. De Mori, A. Moschitti, H. Ney, and G. Riccardi. Comparing stochastic approaches to spoken language understanding in multiple languages. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1569–1583, 2010.

[16] S. Hahn, P. Lehnen, G. Heigold, and H. Ney. Optimizing CRFs for SLU tasks in various languages using modified training criteria. *Interspeech, Brighton, England*, 2009.

[17] D. Hakkani-Tur and G. Riccardi. A general algorithm for word graph matrix decomposition. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages 596–599. IEEE, 2003.

[18] D. Hakkani-Tür, F. Béchet, G. Riccardi, and G. Tur. Beyond asr 1-best: Using word confusion networks in spoken language understanding. *Computer Speech & Language*, 20(4):495–514, 2006.

[19] D. Huggins-Daines and A. Rudnicky. A constrained baum-welch algorithm for improved phoneme segmentation and efficient training. In *Proc. of Interspeech*, pages 1205–1208, 2006.

[20] L.F. Hurtado, J. Planells, E. Segarra, E. Sanchis, and D. Griol. A stochastic finite-state transducer approach to spoken dialog management. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[21] G. Ji and J. Bilmes. Backoff model training using partially observed data: Application to dialog act tagging. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 280–287. Association for Computational Linguistics, 2006.

[22] A. Kipp, M.B. Wesenick, and F. Schiel. Pronunciation modeling applied to automatic segmentation of spontaneous speech. In *Proceedings of Eurospeech*, pages 1023–1026, 1997.

[23] J.W. Kuo and H.M. Wang. Minimum boundary error training for automatic phonetic segmentation. In *Ninth International Conference on Spoken Language Processing*, 2006.

[24] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289, 2001.

[25] F. Lefèvre. A dbn-based multi-level stochastic spoken language understanding system. In *Spoken Language Technology Workshop, 2006. IEEE*, pages 78–81. IEEE, 2006.

[26] F. Lefèvre. Dynamic bayesian networks and discriminative classifiers for multi-stage semantic interpretation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007.*, volume 4. IEEE, 2007.

[27] B. Logan, J.M. Van Thong, and P.J. Moreno. Approaches to reduce the effects of OOV queries on indexed spoken audio. *IEEE Transactions on Multimedia*, 7(5):899–906, 2005.

[28] K. Macherey, F.J. Och, and H. Ney. Natural language understanding using statistical machine translation. In *European Conf. on Speech Communication and Technology*, pages 2205–2208, 2001.

[29] J. Mamou, B. Ramabhadran, and O. Siohan. Vocabulary independent spoken term detection. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 615–622. ACM, 2007.

[30] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks* 1. *Computer Speech & Language*, 14(4):373–400, 2000.

[31] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 591–598, 2000.

[32] S. Meng, P. Yu, F. Seide, and J. Liu. A study of lattice-based spoken term detection for chinese spontaneous speech. In *Automatic Speech*

Recognition & Understanding, 2007. ASRU. IEEE Workshop on, pages 635–640. IEEE, 2007.

[33] D.R.H. Miller, M. Kleber, C. Kao, O. Kimball, T. Colthurst, S.A. Lowe, R.M. Schwartz, and H. Gish. Rapid and accurate spoken term detection. In *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[34] M. Mohri and F. Pereira Michael. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.

[35] A. Moreno, D. Poch, A. Bonafonte, E. Lleida, J. Llisterri, J.B. Marino, and C. Nadeu. Albayzin speech database: Design of the phonetic corpus. In *Third European Conference on Speech Communication and Technology*, 1993.

[36] I. Mporas, T. Ganchev, and N. Fakotakis. Speech segmentation using regression fusion of boundary predictions. *Computer Speech & Language*, 24(2):273–288, 2010.

[37] K.U. Ogbureke and J. Carson-Berndsen. Improving initial boundary estimation for hmm-based automatic phonetic segmentation. In *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[38] S. Paulo and L.C. Oliveira. Dtw-based phonetic alignment using multiple acoustic features. In *Proceedings of Eurospeech*, pages 309–312, 2003.

[39] A. Pérez, F. Casacuberta, I. Torres, and V. Guijarrubia. Finite state transducers based on k-tss grammars for speech translation. *Finite-State Methods and Natural Language Processing*, pages 297–299, 2006.

[40] D. Povey and PC Woodland. Minimum phone error and i-smoothing for improved discriminative training. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 105–108. IEEE, 2002.

[41] C. Raymond, F. Bechet, R. De Mori, and G. Damnati. On the use of finite state transducers for semantic interpretation. *Speech Communication*, 48(3-4):288–304, 2006.

[42] C. Raymond and G. Riccardi. Generative and discriminative algorithms for spoken language understanding. *Proceedings of Interspeech 2007, Antwerp, Belgium*, pages 1605 – 1608, 2007.

[43] H. Romsdorfer and B. Pfister. Phonetic labeling and segmentation of mixed-lingual prosody databases. In *Proc. Interspeech*, pages 3281–3284, 2005.

[44] M. Saraclar and R. Sproat. Lattice-based search for spoken utterance retrieval. *Proceedings of the HLT-NAACL, 2004*, pages 129–136.

[45] I. Szöke, M. Fapso, M. Karafiát, L. Burget, F. Grézl, P. Schwarz, O. Glembek, P. Matejka, S. Kontár, and J. Cernockỳ. But system for nist std 2006-english. In *Proc. NIST Spoken Term Detection Evaluation workshop (STD'06). Gaithersburg, Maryland, USA: National Institute of Standards and Technology*, 2006.

[46] K. Tokuda, H. Zen, and A.W. Black. An HMM-based speech synthesis system applied to english. In *Speech Synthesis, 2002. Proceedings of 2002 IEEE Workshop on*, pages 227–230. IEEE, 2002.

[47] D.T. Toledano, L.A.H. Gómez, and L.V. Grande. Automatic phonetic segmentation. *Speech and Audio Processing, IEEE Transactions on*, 11(6):617–625, 2003.

[48] M. Tomita. An efficient word lattice parsing algorithm for continuous speech recognition. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'86.*, volume 11, pages 1569–1572. IEEE, 1986.

[49] I. Torres and A. Varona. k-tss language models in speech recognition systems. *Computer Speech & Language*, 15(2):127–149, 2001.

[50] G. Tur, J. Wright, A. Gorin, G. Riccardi, and D. Hakkani-Tür. Improving spoken language understanding using word confusion networks. In *Proceedings of the ICSLP*, 2002.

[51] D. Wang. *Out-of-vocabulary spoken term detection*. PhD thesis, 2010.

[52] M. Wechsler and P. Schauble. Speech retrieval based on automatic indexing. In *Proceedings of MIRO-1995*.

[53] P. Yu and F. Seide. A hybrid word/phoneme-based approach for improved vocabulary-independent search in spontaneous speech. *Proc. ICLSP'04*, 2004.