4.1 Introducción

En este capítulo se describe el diseño y desarrollo de un sistema de reconocimiento de habla continua para dominios restringidos. El dominio de trabajo elegido es el de frases que expresan fechas y horas. La razón de elegir este dominio es porque en los Servidores Vocales Interactivos en general, y en los utilizados en esta tesis en particular, preguntar al usuario por una fecha o una hora son interacciones muy frecuentes. Conseguir una fecha o una hora del usuario disponiendo únicamente de un sistema de reconocimiento de habla aislada obligaría a generar un subdiálogo formado por varias preguntas dando lugar a una interacción más larga y pesada para el usuario. Es en estos cuellos de botella donde hay que invertir el mayor esfuerzo para agilizar el servidor vocal, aumentando la rapidez y calidad del servicio.

Aunque el tamaño del vocabulario no será elevado (alrededor de 400 palabras) el sistema que se pretende desarrollar tiene otras dificultades con las que nos debemos enfrentar. En primer lugar, el hecho de abordar el reconocimiento de habla continua obliga a tener en cuenta los efectos de co-articulación entre las palabras que forman la frase. Para ello es necesario utilizar modelos acústicos con información contextual, tanto dentro de una palabra como en sus extremos. Estos modelos acústicos serán modelos de alófono y no de palabra completa debido a que los modelos de palabra completa requieren de una gran cantidad de repeticiones para ser entrenados correctamente, y además, obligaría a entrenar modelos nuevos cada vez que quisiéramos introducir una nueva palabra. Los modelos de alófono permiten un mejor aprovechamiento de los datos acústicos para entrenar. Los modelos de palabras por tanto, se obtendrán por concatenación de los modelos de alófono correspondientes. Un segundo problema de este reconocedor es el hecho de tener que trabajar sobre línea telefónica lo que obliga a tener que hacer frente a una serie de problemas como son la limitación del ancho de banda, la gran variación de la relación señal-ruido, los ruidos inherentes al canal de comunicación y la gran variabilidad de teléfonos existentes. Por último, si pretendemos dar un servicio por teléfono orientado al gran público, el sistema de reconocimiento debe ser independiente del locutor, lo que introduce una gran variedad acústica en la señal de habla. Por otro lado, debido a las limitaciones del modelado acústico utilizado (modelos ocultos de Markov), para afrontar tanto la variabilidad del canal de comunicaciones como del locutor, se deben incorporar modelos de lenguaje que guíen el proceso de decodificación. Estos modelos serán específicos del dominio en el que se esté trabajando, en este caso el de fechas y horas.

Otro aspecto a tener en cuenta es la espontaneidad del habla. En un servicio automático, el reconocedor/decodificador debe hacer frente a una frase del usuario pronunciada como respuesta a una pregunta del sistema. Este tipo de locuciones es bastante diferente al habla leída, conteniendo grandes dosis de espontaneidad. Esta espontaneidad puede alterar de forma importante el habla con la introducción, por parte del usuario, de falsos comienzos, dudas, ruidos, gran variabilidad de la velocidad de locución, y cierta relajación en la pronunciación de algunas palabras o en las estructuras gramaticales utilizadas. En el presente capítulo se trabajará tanto con habla leída como con habla espontánea.

4.1.1 Base de datos

La base de datos utilizada para el desarrollo y evaluación del reconocedor de fechas y horas es la primera versión (1.000 locutores) de la base de datos SpeechDat (Moreno, 1997) desarrollada en la Universidad Politécnica de Barcelona, la misma base de datos que la utilizada en el capítulo anterior. En esta base de datos se dispone de grabaciones de 1.000 locutores a lo largo de la geografía española. En estas grabaciones se recogen gran variedad de locuciones. Para nuestro caso sólo nos quedaremos con los ficheros que contienen locuciones de fechas (etiquetados con D1, D2, y D3) y horas (T1 y T2). El total de ficheros asciende a 5.000. De estos ficheros hay algunos que contienen habla leída (D1, D2 y T1) y otros contienen habla espontánea (D3 y T2). Consideraremos habla leída a aquellas locuciones en las que el locutor no tiene que improvisar la respuesta, es decir, le viene dada y simplemente tiene que leerla. En el caso de habla espontánea ocurre lo contrario, las grabaciones hacen referencia a respuestas del locutor ante preguntas del tipo: ¿Qué día es hoy?, ¿Qué hora es?.

Cada fichero de audio va acompañado de un fichero de texto (con extensión *.eso) en el que se reflejan las características del locutor, de la grabación, así como información de la frase grabada: transcripción de lo que dijo el locutor, frase de referencia y etiquetas de comienzo y fin de la voz.

Para realizar la experimentación hemos agrupado los ficheros en varias listas seleccionándolos de forma aleatoria por llamadas pero manteniendo las siguientes proporciones:

- Entrenamiento de los modelos acústicos. Esta lista contiene 3000 ficheros, 1500 con fechas leídas, 500 con horas leídas, 500 con fechas espontáneas y 500 con locuciones espontáneas de horas. No hemos generado listas independientes para cada tipo de habla (leída y espontánea) debido a que no se disponían de datos suficientes para entrenar habla espontánea por separado.

- Validación y ajuste de parámetros intermedios como la penalización entre unidades o la probabilidad mínima para el suavizado de los modelos de lenguaje. En este caso, tenemos dos listas, una para cada tipo de habla, leída y espontánea. Cada una de las listas contiene 200 ficheros: 100 ficheros contienen locuciones de fechas y 100 de horas.

- Evaluación del sistema de reconocimiento. En este caso también hemos definido dos listas según el tipo de habla (leída y espontánea). Cada una de las listas contiene 800 ficheros mitad con fechas y mitad con horas.

4.1.2 Generación del vocabulario

El vocabulario de reconocimiento con el que se va a trabajar en este dominio contiene 403 palabras. Este diccionario se ha obtenido analizando las 5000 frases contenidas en la base de datos (tanto en los ficheros de entrenamiento como de validación o evaluación). De esta forma, cuando se evalúe el reconocedor no aparecerán

palabras que no estén en el vocabulario. Además de las palabras obtenidas de las 5000 frases, se ha completado el diccionario con nuevas palabras que no correspondían a fechas u horas pronunciadas en la base de datos pero que podrían haber formado parte de una fecha o una hora. Por ejemplo se completó la lista de los días de la semana, los meses del año, las horas del día, etc. y también se añadieron palabras para hacer referencia a festividades (p.e. Navidad, Año Nuevo, etc.) con las que también se puede fijar una fecha, o periodos del día (p.e. Amanecer) con los que se puede determinar una hora aproximada.

4.1.3 Medidas de evaluación

Para evaluar el sistema de reconocimiento desarrollado vamos a considerar las mismas medidas que las descritas en el apartado 3.3.1 pero aplicadas a los errores al nivel de palabra. Estas medidas son los porcentajes (respecto de la frase de referencia) de palabras correctamente reconocidas, palabras sustituidas, palabras insertadas y palabras borradas en la frase hipótesis proporcionada por el reconocedor. A partir de estos porcentajes calcularemos la Tasa de Error (Word Error Rate: WER) y la Precisión de Palabra (Word Accuracy: WA). La Tasa de Error se obtiene como suma de los porcentajes de sustituciones, inserciones y borrados, y la Precisión de Palabra se calcula como el porcentaje complementario de la Tasa de Error. Veamos las siguientes fórmulas:

$$Sus\ (\%) = 100 \times \frac{N_S}{N_T} \qquad Borr\ (\%) = 100 \times \frac{N_B}{N_T} \qquad Inser\ (\%) = 100 \times \frac{N_I}{N_T}$$

$$Tasa\ de\ Error\ (\%) = Subs\ (\%) + Inser\ (\%) + Borr\ (\%)$$

$$Precisión\ de\ Palabra\ (\%) = 100\% - Tasa\ de\ Error\ (\%)$$

donde:

- **N_S:** nº total de sustituciones en las frases de evaluación.
- **N_I:** nº total de inserciones en las frases de evaluación.
- **N_B:** nº total de borrados en las frases de evaluación.
- **N_T:** nº total de palabras en las frases de evaluación.

Para calcular las sustituciones, inserciones, borrados o palabras correctas, se alinean la hipótesis obtenida de la decodificación y la frase de referencia. Este alineamiento se realiza mediante un algoritmo de programación dinámica en el que se fijan una serie de costes. El coste de una palabra correcta es 0, el coste de una inserción o borrado es 1 y el coste de una sustitución es 2 (debido a que una sustitución es equivalente a un borrado más una inserción). En todos los casos se preferirá siempre una sustitución frente a un borrado más una inserción.

En nuestro conjunto de evaluación tenemos 800 frases para habla leída y 800 para habla espontánea con un número medio de palabras de 7,96 y 6,38 respectivamente. Puesto que en esta tesis obtendremos valores de Tasa de Error del 35-30% cuando no se introducen modelos de lenguaje y del orden de 20-15% con modelos de lenguaje, los

intervalos de confianza de los resultados, calculados al 95%, serán respectivamente del 2,34% y 1,96% para el caso de habla leída y, 2,62% y 2,20% para el caso de habla espontánea. Estos intervalos de confianza se han calculado utilizando la fórmula presentada en el apartado 3.3.1.1, donde la variable **p** es ahora la Tasa de Error de Palabra, obtenida con cada uno de los sistemas propuestos.

4.2 Características generales del sistema de reconocimiento

El sistema de reconocimiento responde al siguiente diagrama de bloques:

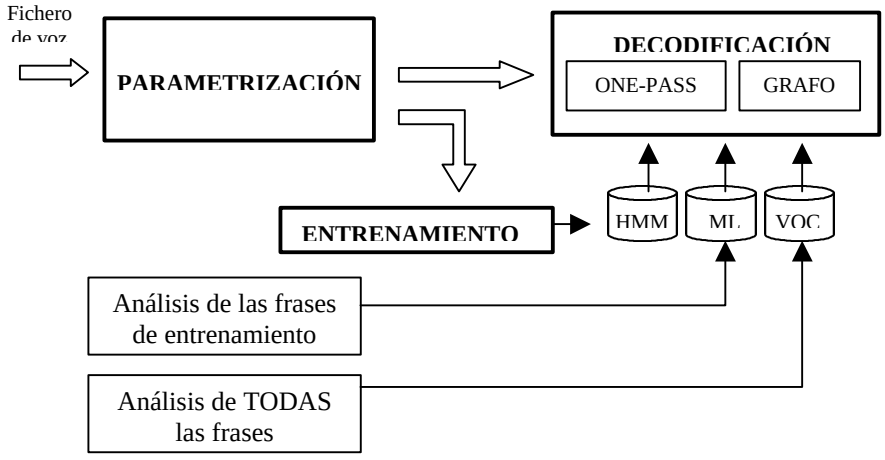

Figura 4-1: Diagrama del sistema de reconocimiento para fechas y horas.

En el proceso de parametrización, al igual que para el sistema de reconocimiento de nombres deletreados, se utilizará la técnica RASTA-PLP (Hermansky at al, 1991) con ventanas de análisis de 25 milisegundos. En este sistema utilizamos también 10 coeficientes cepstral, la energía local de la trama y la primera derivada tanto de los coeficientes cepstral como de la energía (en total 22 parámetros). El proceso de descodificación está formado por dos etapas: la primera de ellas consiste en un algoritmo One-pass del que no obtenemos una única secuencia de palabras sino que obtenemos un grafo o lattice de palabras sobre el cual, en una segunda etapa, aplicaremos modelos de lenguaje más potentes y calcularemos la N mejores secuencias de palabras con bajo coste computacional. El proceso de descodificación utiliza los modelos acústicos (en nuestro caso modelos ocultos de Markov HMM: Hidden Markov Model) obtenidos de una etapa anterior de entrenamiento, un modelo de lenguaje (ML) calculado con las frases de referencia de los ficheros utilizados en el entrenamiento de los modelos acústicos, y el vocabulario (VOC) obtenido del análisis de todos los ficheros.

4.2.1 Modelado acústico a utilizar

Los modelos acústicos utilizados, al igual que para el caso del reconocedor de nombres deletreados (apartado 3.5.1), son modelos ocultos de Markov. Para el

desarrollo del sistema de reconocimiento de fechas y horas consideraremos modelos semicontinuos en lugar de los continuos utilizados en el capítulo anterior. En este caso, la matriz B de probabilidades también se convierte en una función densidad de probabilidad (fdp) aproximada mediante una combinación lineal de un conjunto finito de funciones gaussianas, pero en este caso, el conjunto de gaussianas utilizado es el mismo para todos los estados y constituyen una base finita común. La densidad de probabilidad de un vector 'k' se calculará como la combinación lineal de las aportaciones de cada una de las gaussianas. Los pesos de cada una de las gaussianas son característicos de cada estado. Para simplificar el cálculo, no se utilizan las aportaciones de todas las gaussianas que forman la base finita sino que se utilizan sólo un número reducido. En nuestro caso, el sistema base trabaja con 256 gaussianas y el número de gaussianas consideradas en cada cálculo es 4.

El número de estados del modelo de Markov, N, será 5 aunque presentaremos resultados con 3 y 5 estados para comparar su comportamiento.

Como comentamos en la introducción, trabajaremos con modelos acústicos de alófono. El inventario de alófonos utilizado (45 alófonos en total) se puede consultar en el apéndice C. En este trabajo vamos a considerar tres tipos de modelos de alófono, que son los siguientes:

- ***IC (Modelos Independientes de Contexto):*** son modelos acústicos de cada alófono aislado, en los que no se tiene en cuenta ninguna información del contexto alofónico. Estos modelos se han generado considerando todas las repeticiones de cada alófono (45 modelos, 225 estados y/o distribuciones, considerando 5 estados por modelo).

- ***DC (Modelos Dependientes de Contexto):*** en este caso sí se tiene en cuenta la información del contexto alofónico. Estos modelos se entrenan con todos aquellos alófonos iguales con el mismo contexto anterior y posterior (trifonema). Este tipo de modelo presenta una mayor robustez frente a los efectos de co-articulación pero por otro lado, el número de modelos a entrenar se dispara considerablemente lo que necesita de una mayor cantidad de datos para que se puedan entrenar correctamente. Como veremos más adelante, no se entrenan todos los posibles contextos sino que se realiza una selección en función de su frecuencia de aparición en el conjunto de entrenamiento. Sólo se entrenan modelos específicos para aquellos contextos que aparecen más de un número determinado de veces.

- ***CDC (Modelos Complementarios de los DC):*** estos modelos, uno por alófono, no tienen información contextual, y han sido entrenados con aquellos contextos que no aparecieron un número suficiente de veces como para ser entrenados de forma independiente.

4.2.2 Efectos acústicos modelados

Además de los modelos para los alófonos del alfabeto hemos considerado 6 modelos adicionales: 4 modelos de ruido, un modelo para tramos de voz ininteligible y un

modelo de silencio (SIL). Tanto los modelos de ruido como el modelo de voz ininteligible (que representaremos por "**") han sido entrenados gracias a que en la base de datos están marcados tales efectos, si bien no están delimitados temporalmente, sí que está definida su secuencia en la frase pronunciada. En lo que respecta al modelo de silencio comentar que utilizamos un único modelo para modelar todo tipo de pausas, iniciales, finales y entre palabras.

Los 4 tipos de ruidos modelados corresponden a la clasificación realizada en la base de datos (Moreno, 1997):

- **[fil]:** *pausas rellenas*: hacen referencia a pausas en las que el locutor introduce algún tipo de 'muletilla' para indicar que continúa hablando: uh, um, er, mm.

- **[spk]:** *ruido del locutor*: comprende todo tipo de ruidos producidos por el locutor que no forman parte de la frase: tos, respiración fuerte, dentelleo, risa.

- **[sta]:** *ruido estacionario*: ruidos no intermitentes con una respuesta en frecuencia más o menos plana: ruidos de coche, ruido del canal ó ruido de la calle.

- **[int]:** *ruido intermitente*: ruidos de naturaleza intermitente que típicamente ocurren una única vez, o tienen pausas entre ellos, o cambian el espectro con el tiempo: música, teléfono sonando, timbre ó ruido de puerta.

Por último, comentar que en los modelos de alófono contextuales del apartado anterior no se consideran posibles contextos con los modelos de ruido o con el modelo de voz ininteligible, pero sí con el modelo de silencio. La razón de no considerar estos contextos es la escasez de datos para entrenarlos.

4.2.3 Entrenamiento de los modelos acústicos

Como hemos comentado anteriormente, en el apartado dedicado a la base de datos, vamos a utilizar para entrenar los modelos acústicos un conjunto de 3000 ficheros con fechas y horas pronunciadas tanto de forma leída como espontánea. De esta lista de ficheros se han eliminado ficheros que debido a un error no contienen ninguna fecha u hora, o porque exista en el fichero alguna palabra que ha sido truncada o mal pronunciada por el locutor. Esta reducción de ficheros supone un 7,3% del conjunto de entrenamiento.

Antes de proceder a entrenar los modelos acústicos debemos decidir qué modelos vamos entrenar. Por un lado tenemos los modelos IC de los que obtendremos un modelo para cada uno de los alófonos considerados en el alfabeto. Por otro lado, los contextos a considerar en los modelos DC se definen según el número de repeticiones de ese contexto en los datos de entrenamiento. Para el caso de imponer un mínimo de 100 repeticiones, tenemos 377 modelos contextuales que dan lugar a 1885 estados (considerando 5 estados por modelo). En tercer lugar, los modelos CDC se obtienen considerando los contextos no entrenados de forma aislada, obteniendo un único modelo por alófono.

El proceso de entrenamiento de los modelos acústicos es un proceso iterativo que está formado por los siguientes pasos:

1. Conocida la transcripción de la frase, se construye el modelo acústico de la frase pronunciada mediante la concatenación de los modelos de los alófonos.

2. Utilizando los modelos acústicos obtenidos de la iteración anterior se alinea la secuencia de estados con las tramas de voz mediante el algoritmo de Viterbi y se asigna cada una de las tramas a un estado de la secuencia.

3. Una vez recorridos todos los ficheros, se recalculan los modelos acústicos. Para ello se reestiman:

   - Las gaussianas (media y varianza) que forman la base de funciones con las que se construyen las fdps de cada estado.

   - Los pesos de todas las gaussianas para cada uno de los estados.

   - Las probabilidades de transición entre estados (matriz A), para lo cual hay que analizar las secuencias de estados obtenidas en el alineamiento de los ficheros de voz.

4. Pasaríamos al paso número dos otra vez para realizar un proceso iterativo. Este proceso continúa hasta que el coste medio de alineamiento entre las tramas de voz y los modelos acústicos no varía sustancialmente.

Si no se dispone de unos modelos iniciales o semilla con los que comenzar el proceso, en la primera iteración se asignan las tramas de voz a los estados de forma equidistante. A partir de esta asignación se estiman los primeros modelos y se comienza el proceso iterativo.

El hecho de trabajar con modelos contextuales entre las palabras y el tener incertidumbre sobre el tipo de pausa entre palabras que realmente existe en los ficheros, obliga a introducir nuevas transiciones (posibles caminos) entre los modelos acústicos de alófonos para generar el modelo de cada frase. Estas nuevas transiciones complican el algoritmo de alineamiento entre el modelo de la frase y la secuencia de tramas de voz, pasando de ser un algoritmo de Viterbi a un algoritmo One-pass bastante restrictivo. En la siguiente figura 4-3 podemos observar un ejemplo de las nuevas transiciones que aparecen entre dos palabras.

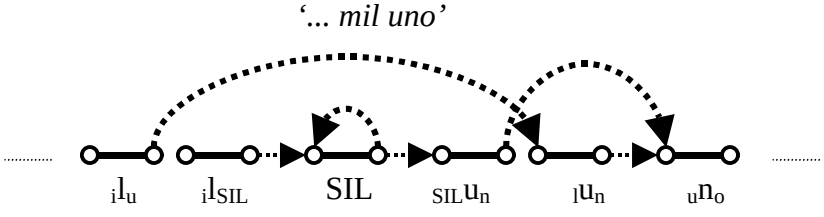

Figura 4-3: Transiciones para permitir variedad de pausas entre palabras.

En la figura 4-3 podemos observar la secuencia de modelos en la unión entre las palabras “mil” y “uno”. Se representa cada modelo únicamente con el estado inicial, el estado final y una línea que los une. “$_{i}l_{u}$” representa el modelo del alófono ‘l’ considerando como contexto las letras ‘i’ y ‘u’. Se puede observar cómo, dependiendo del tipo de pausa, las transiciones son diferentes y los modelos contextuales también. Por último, conviene comentar la transición desde el estado final al estado inicial del modelo de silencio que permite hacer frente a pausas de longitud arbitraria.

Si además se produce un ruido entre estas dos palabras, el número de transiciones aumenta y el modelo se complica. Veamos el siguiente ejemplo:

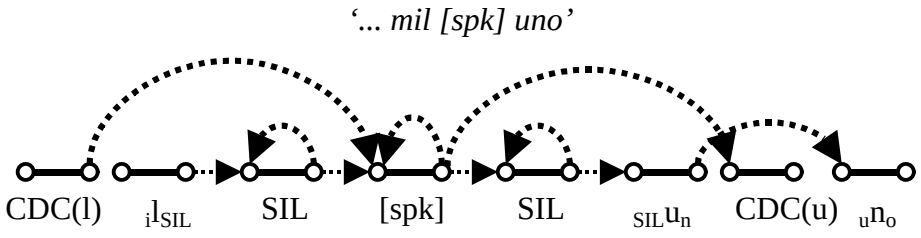

Figura 4-4: Transiciones entre palabras considerando ruidos intermedios.

En este caso tenemos que utilizar los modelos CDC de ‘l’ y de la ‘u’ porque no disponemos de modelos de alófono contextuales con los ruidos considerados. Otro detalle es que en el modelo de ruido también consideramos una transición desde el último estado del modelo hasta el primero para permitir longitudes del ruido arbitrarias.

4.2.4 Algoritmo de decodificación

El proceso de decodificación se realiza en dos etapas: en primer lugar se utiliza un algoritmo One-pass que recorre el espacio de búsqueda formado por todos los modelos de las palabras y genera un grafo o lattice con las secuencias de palabras que acústicamente mejor se ajustan a lo pronunciado. En segundo lugar se realiza un postproceso del grafo para obtener la secuencia de palabras finalmente reconocida.

El One-pass es un algoritmo de programación dinámica ampliamente conocido y utilizado en el campo del reconocimiento de habla continua (Ney, 1984; Ney et al, 1999; Deshmukh et al, 1999), por lo que no nos detendremos en su explicación. Este algoritmo utiliza los modelos acústicos de las palabras del vocabulario para definir un espacio de búsqueda en el que se analizan, para todas las tramas de voz, todos los estados de los modelos. En este apartado comentaremos cómo se generan los modelos de palabra a partir de los modelos de alófono y cómo se resuelve el problema que surge al utilizar modelos contextuales en los extremos de las palabras.

Para formar los modelos de palabra se concatenan los modelos de alófono dependientes del contexto (DC), de forma que los contextos van enlazando con los alófonos contiguos. En el caso de no disponer de un contexto determinado se utilizarían

los modelos complementarios, CDC. Los modelos IC sólo se utilizan para dar soporte a estados concretos, es decir, puede ocurrir que alguno de los estados de algún modelo DC o CDC quede mal entrenado por el escaso número de vectores utilizados para ello, en este caso se utiliza el estado correspondiente del modelo IC para sustituirle.

La utilización de modelos contextuales en los extremos de las palabras, crea el problema de que estos modelos límite dependen de los alófonos iniciales o finales de las palabras posteriores o anteriores. La solución propuesta inicialmente (Ferreiros, 1996; Ravishankar, 1996) consiste en considerar tantos modelos, para los alófonos iniciales y finales, como posibles contextos tengamos. Además, debemos considerar los modelos CDC, por si necesitásemos contextos que no fueron entrenados, y el contexto con el modelo de silencio por si la palabra viniese precedida o seguida de una pausa. En la figura siguiente podemos ver un ejemplo de un modelo acústico para la palabra 'lunes'.

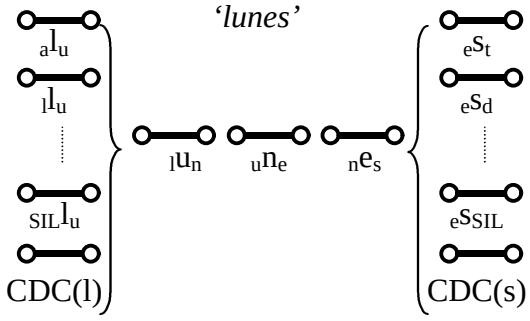

Figura 4-5: Modelo acústico de la palabra 'lunes'.

Dependiendo de las transiciones entre palabras que estemos analizando, se utilizarán unos u otros modelos contextuales y la información se irá almacenando en variables diferentes. Esta duplicidad de los modelos en los extremos repercute en que el algoritmo de One-pass sea más costoso tanto en tiempo como en memoria.

Para el caso del extremo inicial hemos utilizado una aproximación propuesta en (Ravishankar, 1996) que reduce considerablemente la carga computacional. La idea se basa en considerar un sólo modelo inicial en el que las características de los estados cambian según se analicen transiciones con información proveniente de una u otra palabra anterior. Típicamente, en el algoritmo de One-pass se va guardando para cada trama y cada estado, tres datos: la verosimilitud acumulada del camino óptimo hasta ese estado, la *palabra que antecede* a la palabra contenedora del estado analizado y la trama donde se produjo la transición entre la palabra anterior y la actual. De esta forma, a la hora de analizar las posibles transiciones desde estados anteriores hasta el estado actual, podemos utilizar la información acumulada en los estados origen de la transición (en particular la información sobre la *palabra precedente*) para decidir qué características del estado analizado utilizar de forma dinámica. En el algoritmo de One-pass, para cada palabra P y en cada trama, pueden haber varias transiciones posibles desde palabras anteriores $P_{anterior\ i}$. Según el algoritmo de Viterbi, la transición con la mayor verosimilitud acumulada gana: supongamos que es $P_{anterior\ M}$. En este caso, el primer estado de la palabra P infiere automáticamente el último alófono de la palabra $P_{anterior\ M}$

como su contexto izquierdo. A la hora de calcular la probabilidad de generación en el primer estado de la palabra P se considerarán las características del modelo correspondiente al contexto izquierdo definido por la palabra anterior $P_{anterior\ M}$. Este dinamismo en el cálculo de las probabilidades de generación se mantiene a lo largo del primer modelo de alófono de la palabra P. De esta manera se garantiza que para el mejor camino obtenido, se utiliza el mismo modelo contextual en la estimación de la verosimilitud a lo largo de este primer modelo.

Veamos el siguiente ejemplo de la figura 4-6:

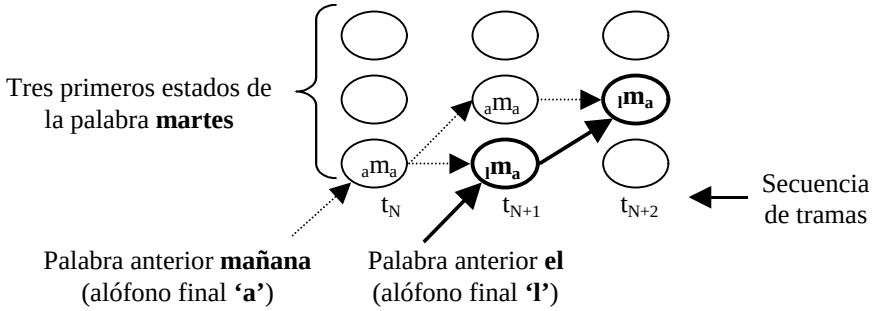

Figura 4-6: Cambio de características de los estados según la palabra predecesora.

Como se puede observar en la figura 4-6, dependiendo de la trama t_N o t_{N+1}, la transición al primer estado de la palabra '*martes*' se realiza desde palabras diferentes, por ejemplo '*mañana*' y '*el*', que tienen alófonos finales diferentes. Según se esté analizando una u otra palabra predecesora, se utilizan unas u otras características del estado (en la figura 4-6 se representa este hecho poniendo el modelo contextual utilizado, $_{a}m_{a}$ ó $_{l}m_{a}$,, sobre el propio estado). La información de la palabra anterior se va guardando a lo largo de los estados para ser utilizada posteriormente. En la trama t_{N+2}, cuando tenemos que analizar las transiciones hacia el segundo estado, debemos tener en cuenta lo siguiente; si estamos evaluando la transición desde el mismo segundo estado, utilizaremos las características correspondientes al segundo estado del modelo contextual $\mathbf{_{a}m_{a}}$, dado que este estado tiene información acumulada habiendo considerado como palabra anterior la terminada en **'a'**. Sin embargo cuando queremos calcular la transición desde el primer estado, y por la misma razón que antes, debemos utilizar las características del segundo estado del modelo $\mathbf{_{l}m_{a}}$. Una vez seleccionada la mejor transición ($_{l}m_{a}$ en este caso), la información se va propagando y el proceso se va repitiendo a lo largo del primer alófono de la palabra.

Por último, comentar que la penalización entre unidades, en nuestro caso modelos de palabra, se ha ajustado utilizando el conjunto de ficheros de validación (ver apartado 4.1.1).

A la hora de hacer el retroceso (backtracking) en lugar de obtener una única secuencia de palabras se obtiene un grafo o lattice de palabras, en el que quedan reflejadas las secuencias de palabras que mejor se ajustan acústicamente a la frase pronunciada por el locutor.

En una segunda etapa, utilizando información más potente que la utilizada en la fase del One-pass, se realiza un postprocesado sobre el grafo de palabra. Esta segunda etapa permite considerar información más potente y a la vez más costosa (modelos acústicos más detallados o modelos de lenguaje de mayor alcance) sin una carga computacional elevada debido a que el espacio de búsqueda se ha quedado reducido considerablemente. Otro aspecto importante de esta etapa, es que permite obtener no sólo la mejor cadena de palabras sino las N mejores con bajo coste computacional. En el apartado 4.4 se describe el algoritmo para la generación y postprocesado del grafo de palabras utilizado en esta tesis.

4.3 Análisis del modelado acústico

En este apartado se describirán los experimentos realizados con el fin de ajustar el diseño de los modelos acústicos a utilizar. En este ajuste probaremos diferentes longitudes de los modelos acústicos y propondremos la técnica de entrenamiento selectivo (selective training) para evaluar la resolución del modelado en función de la cantidad de datos de entrenamiento.

4.3.1 Longitud de los modelos

Considerando un esquema base que consiste en utilizar modelos semicontinuos mediante la técnica de Fuzzy Vector Quantization (Ferreiros, 1996; Ferreiros y Pardo, 1999), con un codebook (o base de funciones) formado por 256 centroides y un número de trifonemas de 377, nos planteamos dimensionar el número de estados por modelo de alófono.

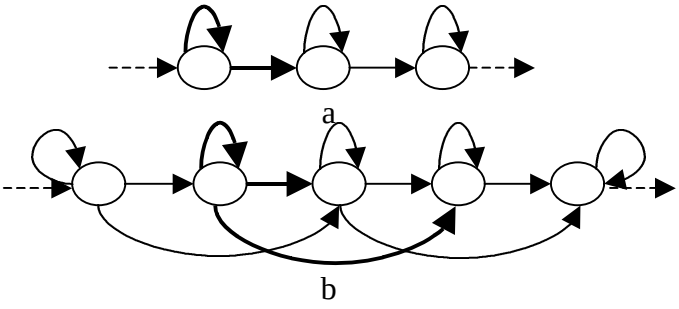

Figura 4-7: Estructuras de modelos acústicos utilizados con 3 y 5 estados respectivamente.

En un principio consideramos un modelo de alófono formado por 3 estados en el que se permiten únicamente transiciones simples entre estados (ver figura 4-7a). Siguiendo la evolución realizada en la Universidad Carneige Mellon, se probó la consideración de modelos de alófono más largos, de 5 estados. En este caso se permite transiciones dobles entre estados como refleja la figura 4-7b.

Aunque consideramos más estados por modelo, seguimos considerando modelos de alófono con un contexto anterior y otro posterior (trifonema) y no modelos con dos contextos anteriores y dos posteriores (pentafonema) (Hain, 1999). Los resultados obtenidos para habla leída y espontánea se presentan en la tabla 4-1.

Habla Leída	Corr (%)	Sus (%)	Inser (%)	Borr (%)	WER (%)
3 estados	68,1	21,4	5,4	10,5	**37,3**
5 estados	73,1	17,6	4,5	9,3	**31,4**
Habla Espontánea	**Corr (%)**	**Sus (%)**	**Inser (%)**	**Borr (%)**	**WER (%)**
3 estados	64,9	24,1	9,7	10,9	**44,7**
5 estados	70,7	19,7	7,3	9,6	**36,6**

Tabla 4-1: Resultados para los dos tipos de habla según el número de estados considerados en el modelo acústico. Se representan los porcentajes de palabras correctas (Corr), sustituciones (Sus), inserciones (Inser), borrados (Borr) y la Tasa de Error (Word Error Rate: WER) (San-Segundo, D., 2001).

Las reducciones relativas de la tasa de error son de un 15,8% para habla leída y de un 18,1% para habla espontánea. Con estas reducciones tan importantes se pone de manifiesto que el modelado con 5 estados es mejor. Existen dos razones principales: en primer lugar el esquema 4-7b permite una mayor flexibilidad en la longitud de los trifonemas y en segundo lugar, permite modelar más efectos acústicos al disponer de mayor número de estados. Como se puede observar, la mayor mejora relativa se obtiene en el habla espontánea donde la variabilidad de efectos es mayor. El tipo de error que mayor reducción ha sufrido son las sustituciones, lo que pone de manifiesto el mayor poder de discriminación del modelado.

Con estos resultados, se pone de manifiesto (como ya era conocido) que el aumento de la resolución de los modelos permite mejorar las tasas de reconocimiento. Esta circunstancia es válida cuando se dispone de suficientes datos para entrenar la resolución definida en el modelado. En nuestro caso, sí disponemos de la cantidad de datos suficiente, como se pondrá de manifiesto en los apartados siguientes. Los experimentos que realizaremos a continuación tomarán como base los modelos de alófono de 5 estados.

4.3.2 Entrenamiento Selectivo

En este apartado probaremos la técnica de entrenamiento selectivo con el fin, en un principio, de aumentar la tasa de reconocimiento. Esta técnica consiste en realizar una selección de los datos con los que vamos a realizar el entrenamiento acústico. En lugar de entrenar con todos los datos disponibles, se seleccionan de forma no supervisada aquellos datos que mejor modelarán los efectos acústicos deseados. En nuestro caso utilizamos como heurístico de selección la verosimilitud por trama, obtenida del alineamiento entre el fichero a considerar y la versión de los modelos acústicos de la iteración anterior. El proceso de entrenamiento selectivo añade las siguientes fases al proceso original descrito en el apartado 4.2.3:

1.- Una vez que se ha realizado el alineamiento entre todos los ficheros con los modelos acústicos de la interacción anterior, dichos ficheros se ordenan de mayor a menor verosimilitud por trama.

2.- Para obtener la versión de los modelos de la nueva iteración, se excluyen de las estimaciones aquellos ficheros con peor verosimilitud por trama hasta eliminar un determinado porcentaje de los ficheros.

En cada iteración se vuelven a alinear todos los ficheros con la versión anterior de los modelos y se vuelve a hacer la selección. De esta forma se permite que ficheros que pudiesen ser inicialmente eliminados puedan volver a ser considerados si con las nuevas estimaciones de los modelos, la verosimilitud por trama adquiere unos valores importantes.

En la figura 4-8, podemos ver la evolución de la tasa de error para los dos tipos de habla al eliminar un determinado porcentaje de ficheros de voz.

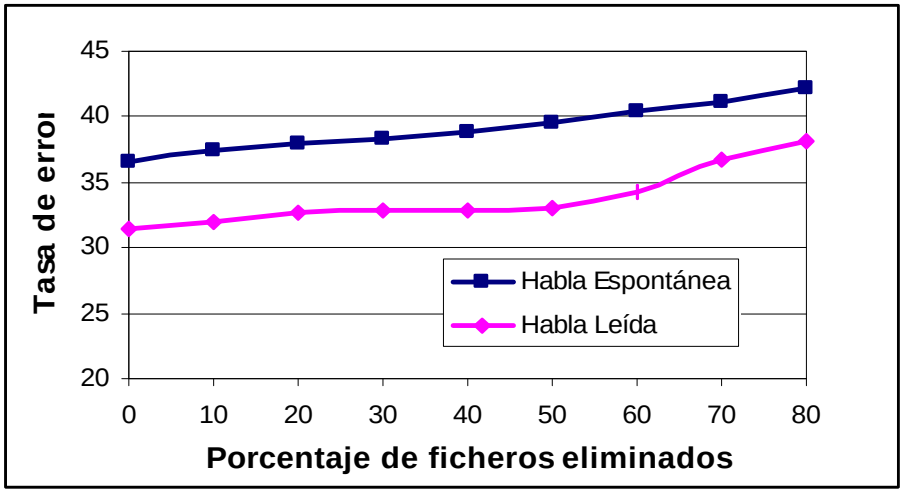

Figura 4-8: Evolución de la Tasa de Error según el porcentaje de ficheros eliminados (San-Segundo, D., 2001).

La primera conclusión que podemos sacar de estos resultados es que el entrenamiento selectivo, para nuestro dominio concreto, no mejora la tasa de error, las dos evoluciones (para los dos tipos de habla) son monótonas crecientes. Esta circunstancia pone de manifiesto que en el grupo de ficheros de entrenamiento no aparecen efectos colaterales, como pueden ser problemas en la grabación o errores de transcripción que no aparezcan en proporciones similares en la lista de evaluación. Este hecho parece bastante probable si se considera que la selección de las listas de entrenamiento, validación y evaluación ha sido de forma aleatoria.

Si nos fijamos en detalle en la evolución de ambas gráficas, podemos observar como la tasa de error apenas se incrementa a medida que vamos quitando ficheros. Al haber eliminado casi un 50% de los ficheros de entrenamiento, la tasa de error solamente se incrementa 2 ó 3 puntos, valores muy cercanos a las bandas de fiabilidad. Al observar este comportamiento deducimos que la resolución del modelado utilizado es demasiado grosera, aun después de haber aumentado el número de estados y transiciones como vimos en el apartado anterior. Es necesario, por tanto, utilizar modelos más detallados

que permitan sacar mejor provecho de los datos de entrenamiento disponibles. En los apartados siguientes nos planteamos aumentar esta resolución del modelado.

4.3.2.1 Aumento del número de trifonemas

El primer paso que probamos ha sido aumentar el número de trifonemas o modelos de alófono dependientes del contexto entrenados de forma independiente. Como comentamos anteriormente (apartado 4.2.4) la selección de los modelos contextuales a entrenar, dependía del número de repeticiones en los datos de entrenamiento. Considerando un valor mínimo de 100 repeticiones, obtuvimos 377 modelos contextuales. Con el fin de aumentar el número de modelos contextuales redujimos a 50 y 15 el número de repeticiones mínimas, obteniendo unas cantidades de trifonemas de 630 y 970 respectivamente.

En la figura 4-9 se puede ver la evolución de la Tasa de Error (Word Error Rate) con el número de modelos contextuales entrenados para los dos tipos de habla (Leída y Espontánea).

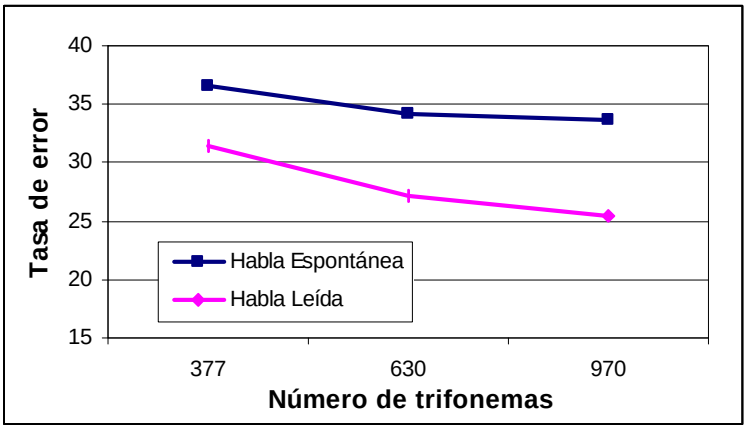

Figura 4-9: Evolución de la Tasa de Error según el número de trifonemas considerado.

Como se puede observar, el aumento del número de trifonemas ha reducido considerablemente el error (19,0% en habla leída y 8,2% en habla espontánea). La reducción del error es sensiblemente menor en habla espontánea debido a que los efectos acústicos son más variables que en habla leída y por tanto, al introducir más modelos de alófonos pero entrenados con menor cantidad de repeticiones, la reducción del error es menor. También se puede observar cómo la reducción de error es más importante al pasar de 377 a 630 trifonemas que la obtenida al pasar de 630 a 970. En este último caso, el número de repeticiones por modelo es menor y dado su peor entrenamiento la reducción del error es también menor.

Con estos resultados podemos concluir que realmente teníamos datos para entrenar modelos más detallados, como dedujimos de la evolución obtenida en el entrenamiento selectivo.

4.3.2.2 Aumento del número de centroides

Como comentamos anteriormente estamos utilizamos modelos semicontinuos mediante la técnica Fuzzy Vector Quantization (Ferreiros, 1996). En esta técnica, para estimar la función densidad de probabilidad de cada estado se utiliza una base de funciones (en nuestro caso 256 gaussianas). Dependiendo del estado concreto, los pesos asociados a cada función de la base de funciones son diferentes. Para calcular la densidad de probabilidad de un vector concreto en un estado determinado, se utilizan las N funciones más cercadas (N=4, en nuestro caso), menor distancia entre el vector de parámetros y la media de la gaussiana. La distancia utilizada es la de Mahalanobis en la que se pondera cada componente por el inverso de su varianza de forma que todas las componentes aporten de igual forma a la distancia sin que las componentes de mayor varianza tengan una repercusión mayor.

En este apartado nos planteamos aumentar el número de centroides o funciones que forman nuestra base con el fin de aumentar la resolución. Los resultados se pueden consultar en la figura 4-10. Para analizar la evolución hemos considerado 128, 256, 512, 1024 y 2048 centroides, y para poder comparar los resultados hemos variado de igual forma el número de funciones (N) con las que se estima la densidad de probabilidad de cada vector, utilizando valores de 2, 4, 8, 16 y 32 respectivamente. El número de modelos contextuales considerado en todos los casos es de 377.

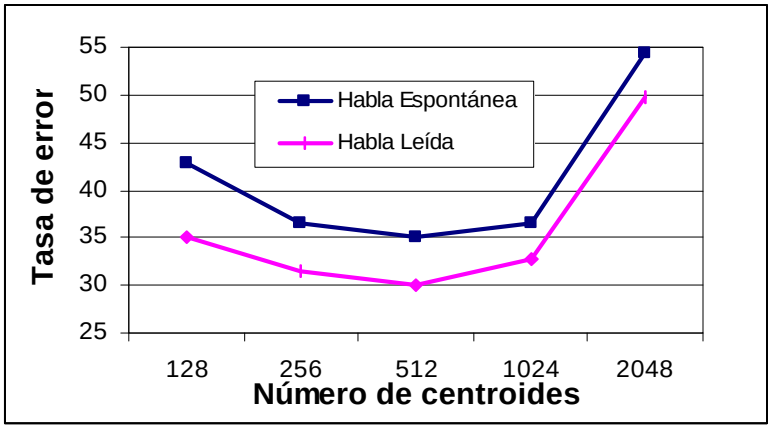

Figura 4-10: Evolución de la Tasa de Error según el número de centroides considerado (San-Segundo, D., 2001).

Como se puede observar, el mínimo se obtiene para 512 centroides lo que nos dice que realmente teníamos datos para entrenar modelos más detallados. Si bien, a partir de 1024, el error comienza a aumentar considerablemente. Este hecho se produce porque para caracterizar la función densidad de probabilidad de cada estado necesitamos entrenar tantos pesos como centroides tengamos. Al aumentar el número de centroides se dispara el número de parámetros a estimar sobrepasando la capacidad de entrenamiento con los datos disponibles.

En este punto nos planteamos cambiar la técnica de modelado semicontinuo. En lugar de caracterizar la función densidad de probabilidad de un estado con todas las gaussianas o funciones base del codebook nos planteamos seleccionar para cada estado M gaussianas de la base de funciones (Duchateau et al, 1998). De forma que la densidad de probabilidad de un vector en un estado, vendrá dada por estas M gaussianas, independientemente del vector que se esté analizando. En esta solución se permite la posibilidad de que puedan haber gaussianas comunes a varios estados, la diferencia residirá por tanto en el peso asociado a esa gaussiana. El valor de M es fijo y por comparación con los anteriores experimentos lo consideraremos de valores 2, 4, 8, 16 y 32 para valores de número de centroides de 128, 256, 512, 1024 y 2048 respectivamente. La evolución de la tasa de error con el aumento del número de centroides se presenta en la figura 4-11.

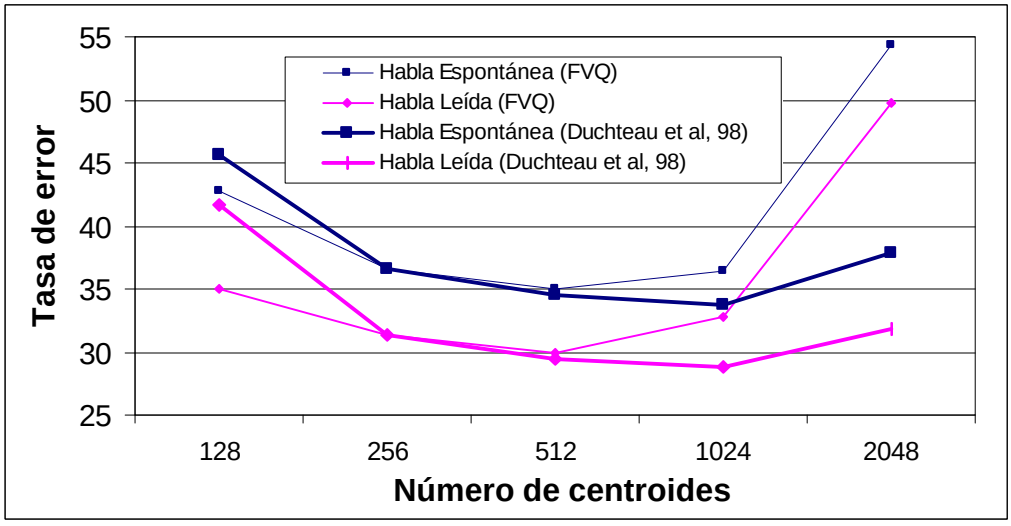

Figura 4-11: Comparación de la evolución de la Tasa de Error para diferentes modelados semicontinuos según el número de centroides.

Como se puede observar, modelando con un número M de gaussianas (Duchteau et al, 1998) nos permite conseguir un nuevo mínimo de la tasa de error, en este caso para 1024 centroides. A partir de este punto, a medida que vamos aumentado el número de centroides, la tasa de error se vuelve a incrementar por el aumento de los parámetros en relación con los datos de entrenamiento disponibles.

Otra forma de conseguir un modelado semicontinuo en el que la fdp de un estado viniese caracterizada por M gaussianas elegidas de entre una base, se puede conseguir partiendo de un modelado continuo (la fdp de cada estado viene reflejada por un conjunto de M gaussianas) y reduciendo el número de parámetros a entrenar mediante la compartición de gaussianas y no de estados completos. De esta forma, se generará una base de funciones común para la definición de las fdps de todos los estados.

De estos dos últimos apartados podemos deducir que el entrenamiento selectivo, aunque no nos ha sido útil para aumentar las tasa de reconocimiento, nos ha permitido evaluar la resolución del modelado utilizado, poniendo de manifiesto la posibilidad de entrenar modelos más detallados.

4.4 Grafo de palabras

En este apartado se describe la generación de un grafo de palabras a partir del proceso de alineamiento dinámico realizado por el One-pass. El algoritmo implementado es una simplificación del propuesto por Ney (Ney, 1994; Ney y Ortmanns, 1999). Este grafo nos permitirá realizar una segunda fase de decodificación en la que se podrán aplicar modelos de lenguaje más potentes (3-gram) y obtener las N mejores secuencias de palabras con bajo coste computacional.

En los experimentos que se presentan en los siguientes apartados hemos utilizado los modelos acústicos base. En estos modelos hemos considerado 377 modelos contextuales y 256 gaussianas que forman la base de funciones con las que representamos la fdp de cada estado mediante la técnica FVQ.

4.4.1 Obtención del grafo de palabras

La idea principal para la obtención de un grafo de palabras es plantear diferentes alternativas de decodificación en aquellas zonas de la señal de voz en las que el reconocedor tiene mayores problemas para seleccionar una u otra secuencia de palabras, debido al gran parecido acústico entre las alternativas. Para la obtención del grafo de palabras debemos anotar todas las posibles cadenas de palabras que sin ser la secuencia óptima, estén cerca, en términos de verosimilitud acústica, a dicha secuencia óptima. Para realizar estas anotaciones debemos añadir más estructuras de almacenamiento al algoritmo de One-pass. Los pasos para la generación del grafo de palabras son los siguientes:

1.- Para cada trama y cada palabra debemos analizar las posibles palabras predecesoras, seleccionando las transiciones entre palabras más probables. H. Ney propone el uso de una estrategia de Beam Search para obtener un número limitado de predecesores (Ney y Ortmanns, 1999). En nuestro caso, al no utilizar Beam Search, hemos definido un número constante de predecesores que hemos denominado COMPLEJIDAD_GRAFO. Otro aspecto importante de la simplificación propuesta es que esta variabilidad se considera únicamente en los saltos entre palabras (entre el último estado de la palabra predecesora y el primer estado de la siguiente) y después se va propagando a lo largo de la palabra, mientras que H. Ney propone que el análisis de las diferentes historias se mantenga a lo largo de todos los estados dentro de una palabra. El hecho de considerar la variabilidad dentro de una palabra supone mantener (mayor memoria) y calcular (mayor tiempo de procesado) en cada estado de la palabra, y para cada posible palabra predecesora, la verosimilitud acumulada hasta ese estado y la trama de tránsito entre las palabras. Este algoritmo requiere de mayor memoria y mayor tiempo de procesado, siendo computacionalmente inviable si no se dispone de una técnica de Beam Search muy ajustada que permita reducir los cálculos como es nuestro caso.

2.- Al final de la señal de voz se realiza un proceso de backtracking en el que se recorren las listas de palabras predecesoras obtenidas en el primer paso, y se va calculando un árbol inverso (partiendo de las últimas tramas) con todas las posibles

secuencias de palabras. Para cada palabra o nodo del árbol debemos calcular la trama inicial y final de la palabra así como el incremento de verosimilitud acústica a lo largo de dicha palabra. En la figura 4-12 podemos observar un ejemplo de árbol generado.

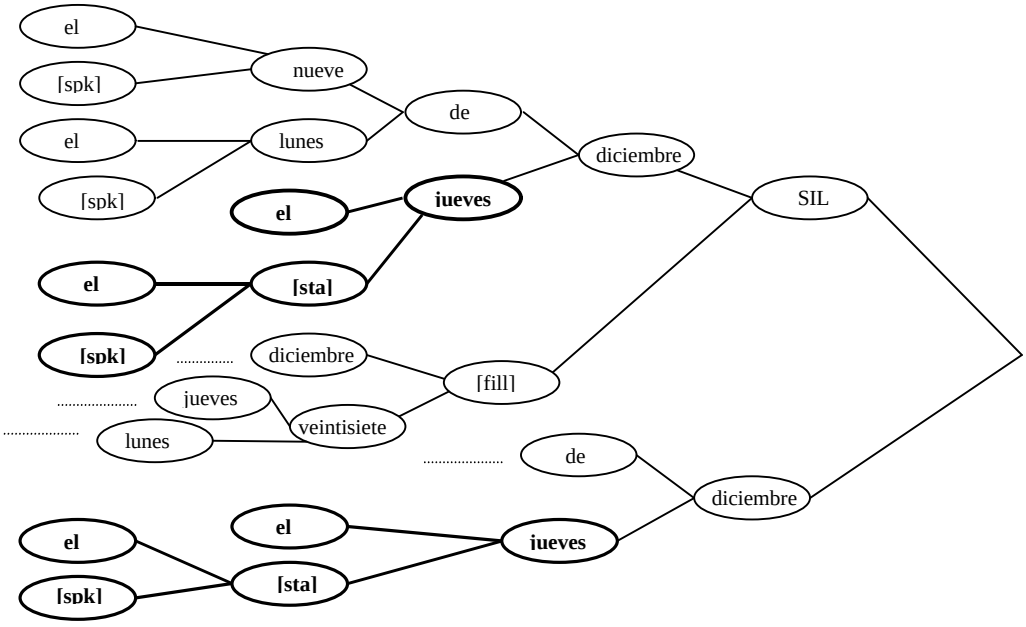

Figura 4-12: Árbol inverso resultado del segundo paso en la generación del grafo de palabras (COMPLEJIDAD_GRAFO=2).

3.- El siguiente paso es agrupar nodos que tengan misma trama inicial y misma trama final, dejando como incremento de verosimilitud del nodo, el máximo de los nodos agrupados. En la figura 4-13, podemos ver el grafo resultante del ejemplo anterior.

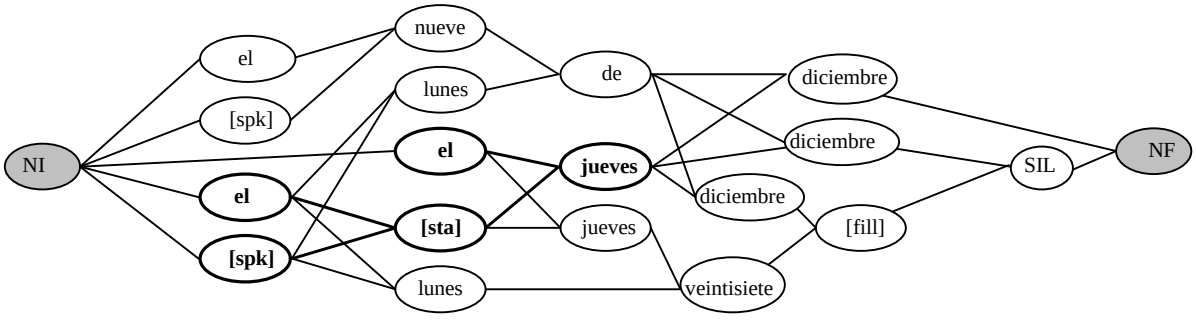

Figura 4-13: Grafo de palabras para el ejemplo de la figura 4-12.

En la figura 4-13 se ha resaltado en negrita a modo de ejemplo ilustrativo, los nodos resultado de la unión de aquellos marcados en la figura 4-12.

A continuación se presenta el pseudocódigo del algoritmo simplificado para la generación del grafo de palabras.

ALGORITMO SIMPLIFICADO PARA LA GENERACIÓN DE UN GRAFO

/* Procesado de izquierda a derecha */

/* Inicialización */

- La verosimilitud (Verosim. Acum.) de los estados iniciales de los modelos de palabra se inicializa a la densidad de probabilidad de la 1ª trama en ese estado.
- La Verosim. Acum. para el resto de estados se inicializa a 0.

/* Para todas las tramas y todos los estados vamos calculando la verosimilitud acumulada */

- **Recorremos los estados iniciales** de todas las palabras calculando la Verosim. Acum. Si se produce una transición entre palabras, entonces almacenamos:
  - Las N mejores **palabras antecesoras** donde N = COMPLEJIDAD_GRAFO.
  - Una única **Verosim. Acum**. obtenida de la mejor transición. No se guardan varias Verosim. Acum. dependiendo de la palabra predecesora.
  - El **número de trama** en el que se ha producido la transición.
- **Recorremos el resto de estados** de las palabras, calculando la **Verosim. Acum.** En cada estado se considera una única Verosim. Acum. (la mejor transición entre palabras). Las transiciones entre los estados van definiendo la manera en la que la información, correspondiente a palabras predecesoras y trama de tránsito, se va propagando.

/* Proceso de Backtraking y obtención del árbol inverso. */

- En la trama final consideramos COMPLEJIDAD_GRAFO posibles palabras final de frase. Para cada una de estas palabras finales calculamos su trama de inicio. Cada trama de inicio, es a su vez, un final de otras COMPLEJIDAD_GRAFO palabras, de las que volvemos a calcular su trama de inicio. Y así sucesivamente hasta llegar a la trama inicial:
  - Para cada palabra, calculamos el **incremento de verosimilitud** (Incr. Verosim.) a lo largo de ella. Este incremento es la diferencia entre la Verosim. Acum. del último estado de la palabra anterior (en la trama de inicio) y la Verosim. Acum. en el último estado de la palabra analizada (en la trama final). También calculamos la **trama inicial y final** del nodo.
  - Una vez realizado este proceso, obtenemos un árbol inverso de palabras de forma que en cada nivel, los caminos se bifurcan en COMPLEJIDAD_GRAFO ramificaciones (Ver figura 4-12).

/* Realizamos la agrupación de nodos que tengan misma trama inicial y final */

- Vamos comparando todos los nodos entre sí de forma que cuando dos nodos tengan las **mismas tramas inicial y final los agruparemos**:
  - El incremento de verosimilitud será el mayor del obtenido en los nodos agrupados.
  - Las transiciones de los nodos agrupados con el resto de nodos se suman; el nuevo nodo resultado contiene todas las transiciones posibles de los nodos agrupados.
- Consideraremos dos nodos adicionales, inicial y final. Todas las palabras que comiencen en la primera trama se unirán con el nodo inicial y todas aquellas que terminen en la última trama de voz se deberán conectar al nodo final.

La agrupación de nodos es un proceso muy importante porque permite reducir considerablemente el número de nodos (en el ejemplo pasamos de 64 nodos a 18), lo

que reducirá el tiempo del procesado posterior que se realice sobre el grafo. Los motivos de la agrupación de nodos se pueden resumir en los siguientes puntos:

- Si dos nodos son iguales (mismas tramas iniciales y finales) la información sobre los nodos predecesores es la misma y por tanto los subárboles completos que se generan desde ese nodo hasta el comienzo de la frase son idénticos y no aportan alternativas nuevas.

- El número de nodos se reduce considerablemente, lo que permite grandes ahorro de memoria y de tiempo de procesado.

La condición de agrupación podría relajarse permitiendo variaciones pequeñas en las tramas de comienzo y final de los nodos agrupados, lo que da lugar a grafos más flexibles.

4.4.2 Procesado del grafo de palabras

Con la obtención del grafo de palabras hemos reducido considerablemente el espacio de búsqueda. Este espacio de búsqueda más pequeño nos permite incorporar fuentes de conocimiento más potentes, como por ejemplo modelos de lenguaje de mayor alcance, sin aumentar enormemente el tiempo de cálculo. A la hora de volver a procesar el grafo, podemos optar por una de las dos opciones que se proponen a continuación:

1.- Sobre cada uno de los nodos del grafo podemos reconstruir el modelo acústico de la palabra y volvemos a realizar un One-pass sobre la estructura del grafo. En este caso podríamos considerar múltiples contextos, tanto al comienzo como al final de cada palabra (apartado 4.2.4) sin que se incremente demasiado la carga computacional.

2.- Utilizar el incremento de verosimilitud a lo largo de cada nodo/palabra (calculada en el proceso de generación del grafo) como heurístico de búsqueda para recorrer el grafo. En este caso, es necesario complementar el heurístico de búsqueda con nuevas fuentes de conocimiento (modelos de lenguaje más potentes) para mejorar la tasa, dado que si utilizásemos sólo el incremento de verosimilitud, el resultado sería el mismo que el obtenido en la primera etapa de reconocimiento.

Habla Leída	**Corr (%)**	**Sus (%)**	**Inser (%)**	**Borr (%)**	**WER (%)**
1ª Opción (modelo)	73,6	17,4	4,3	9,0	**30,7**
2ª Opción (veros. acum.)	73,1	17,6	4,5	9,3	**31,4**
Habla Espontánea	**Corr (%)**	**Sus (%)**	**Inser (%)**	**Borr (%)**	**WER (%)**
1ª Opción (modelo)	70,9	19,9	7,5	9,6	**36,2**
2ª Opción (veros. acum.)	70,7	19,7	7,3	9,6	**36,6**

Tabla 4-2: Resultados para las diferentes opciones de recorrido del grafo. Se representan los porcentajes de palabras correctas (Corr), sustituciones (Sus), inserciones (Inser), borrados (Borr) y la Tasa de Error (Word Error Rate: WER).

En la tabla 4-2 presentamos los resultados para ambas formas de postprocesar el grafo de palabras. Las diferencias son muy pequeñas, menores incluso que las bandas de fiabilidad lo que nos lleva a descartar la primera de las opciones debido a su mayor tiempo de cómputo (del orden de 1.500 veces superior a la segunda). Por último comentar que en el caso de que se quisiese utilizar nuevos modelos acústicos, más potentes, no tendríamos más remedio que optar por la primera de las opciones.

4.5 Modelo de Lenguaje

El modelado de lenguaje en un dominio concreto, como el de fechas y horas, pretende caracterizar a la vez que explotar las regularidades que se producen en el lenguaje natural a la hora de construir frases pertenecientes a ese dominio. Hay muchas formas de modelar el conocimiento gramatical pero en nuestro caso vamos a utilizar gramáticas estocásticas de tipo N-gram (2-gram y 3-gram igual que para el caso del reconocedor de nombres deletreados), que son las más utilizadas hasta la fecha.

4.5.1 Gramáticas N-gram

Las gramáticas de tipo N-grams hacen uso de las últimas N-1 palabras para estimar la palabra que con mayor probabilidad las seguirá en la secuencia. Dependiendo del valor de N, tendremos gramáticas 1-gram, 2-gram, 3-gram, ... Según aumentamos el valor de N (orden de la gramática) mayor es la restricción que se impone y por tanto disminuye la perplejidad en el reconocimiento. Por otra parte, los espacios de búsqueda que se generan son mayores (Colás, 1999) “un modelo de Markov asociado a una unidad de reconocimiento tiene que almacenar diferentes historias en función de las posibles combinaciones de unidades que le pueden preceder. Ello obliga a tener que mantener un determinado número de copias de dicho modelo de forma que se generen copias sin ambigüedad gramatical”. En general, la capacidad de modelado de este tipo de gramáticas es reducida porque modela restricciones locales. Aun así, son de gran ayuda a la hora de orientar el proceso de decodificación. En nuestro caso particular utilizaremos gramáticas 2-gram y 3-gram que nos permitirán reducir considerablemente las tasas de error como veremos en los apartados siguientes.

4.5.2 Incorporación del Modelo de Lenguaje

En nuestro sistema de reconocimiento incorporaremos dos modelos de lenguaje. Un modelo 2-gram en la primera etapa del reconocedor (One-pass) y un modelo 3-gram en la segunda etapa (grafo de palabras).

4.5.2.1 Modelo 2-gram en el One-pass.

Este modelo 2-gram será incorporado en la primera etapa del reconocedor, en el algoritmo de One-pass. En esta etapa se van analizando, para todas las tramas, todos los estados que forman los modelos de las palabras, y se van calculando tanto la verosimilitud acumulada de cada estado como las transiciones entre estados. Pues bien, a la hora de considerar las transiciones entre palabras (entre el último estado de la

palabra predecesora y el primer estado de la siguiente) además de tener en cuenta la verosimilitud acústica se debe tener en cuenta la probabilidad gramatical.

Este proceso tiene el problema de cómo gestionar las unidades que sin tener representación gramatical tienen un modelo acústico asociado, como pueden ser los modelos de ruido o el modelo de silencio. La solución será diferente dependiendo de si lo que pretendemos es predecir la unidad acústica, o por el contrario considerarla como unidad predecesora. En el caso de que nos planteemos predecir un modelo sin representación gramatical, debemos introducir una penalización adicional a la verosimilitud acústica para evitar que esta unidad se inserte de forma frecuente en relación con el resto. En nuestro caso hemos considerado la probabilidad que resulta de suponer todos los modelos acústicos equiprobables (1 / número de modelos acústicos).

Cuando se pretenda predecir una palabra cuyo modelo predecesor es un modelo sin representación gramatical, no podremos aplicar directamente la probabilidad del modelo de lenguaje, puesto que no existe como tal. En ese caso debemos realizar un backtracking parcial hasta encontrar un modelo acústico con representación gramatical y aplicar entonces la penalización asociada. En la figura 4-14 podemos ver un ejemplo.

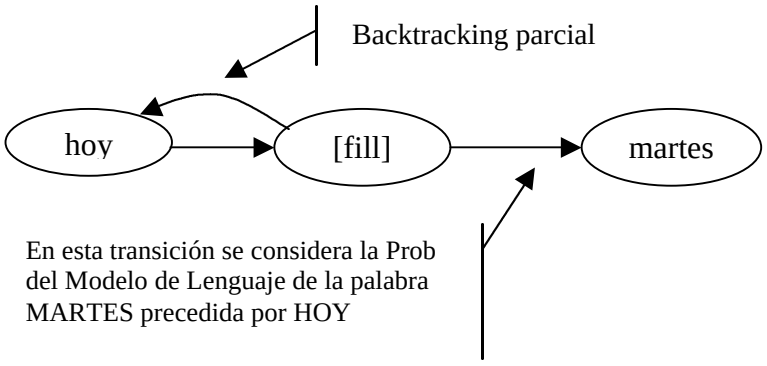

Figura 4-14: Utilización de la probabilidad del modelo de lenguaje considerando un modelo de ruido como modelo predecesor.

En este ejemplo podemos ver cómo para calcular la probabilidad de la palabra “martes” debemos hacer un backtracking, partiendo del modelo de ruido “[fill]” hasta encontrar una palabra “hoy” con representación gramatical. De esta forma, la probabilidad del modelo de lenguaje será la de la palabra “martes” precedida de la palabra “hoy”. Si ocurriera que llegamos al comienzo de la señal de voz sin encontrar ningún modelo con representación gramatical, aplicaríamos la probabilidad de que la palabra analizada fuese comienzo de frase.

4.5.2.2 Modelo 3-gram en el grafo de palabras

La formulación descrita en el apartado anterior no es aplicable cuando queremos incorporar un modelo 3-gram. En este caso, la probabilidad de transición entre dos palabras P_i y P_j, no sólo depende de dichas palabras si no que además es función de la palabra anterior a la P_i. Dicha palabra puede tener varias palabras predecesoras con lo que la probabilidad gramatical no está unívocamente definida. Este problema se

resuelve fácilmente replicando cada palabra tantas veces como posibles predecesoras pueda tener, es decir, creando estados gramaticales diferentes (Colás, 1999).

Debido a la necesidad de duplicar nodos hemos decidido incorporar esta gramática en el grafo de palabras donde el espacio de búsqueda es mucho más reducido y la carga computacional será menor. En la figura 4-15, se muestra un ejemplo de la duplicación de nodos en un grafo de palabras.

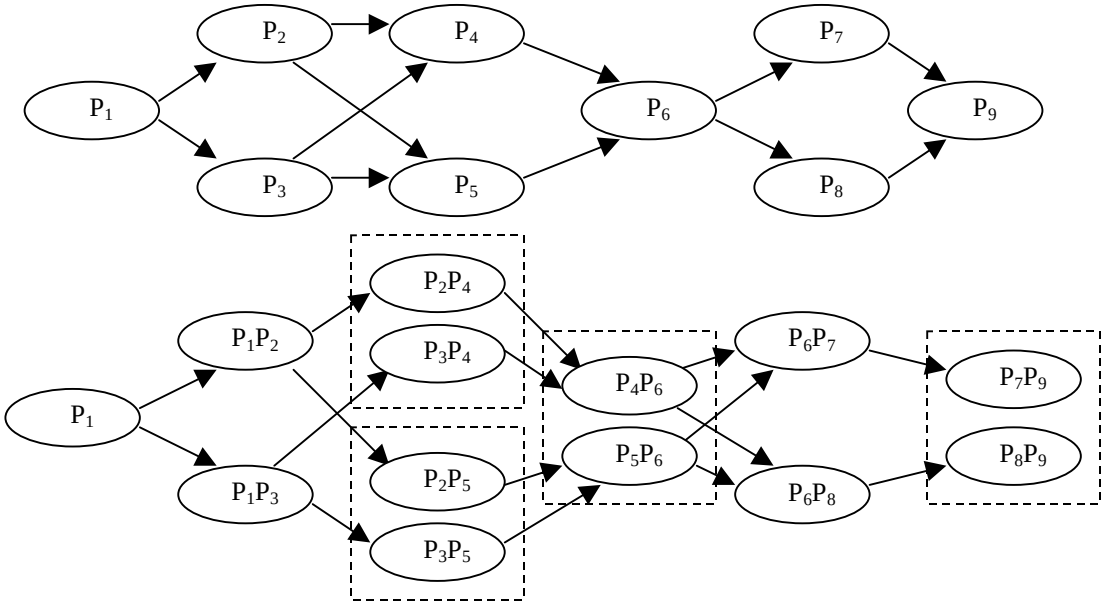

Figura 4-15: Duplicación de nodos para la incorporación de un modelo 3-gram.

Los pasos para la duplicación de los nodos del grafo son los siguientes:

1. Si un nodo P tiene N predecesores distintos (P_i) con i= 1,2,...N en el grafo original, este nodo se debe replicar N veces, etiquetando cada copia con (P_iP), i = 1,2,... ,N respectivamente. En la figura 4-15 se recuadran con líneas discontinuas los nodos duplicados.

2. Si existía un enlace entre el nodo (P_i) y el nodo (P_j), en el grafo nuevo debe haber un enlace entre cada copia de (P_i) y (P_iP_j). Notar que en este proceso, si existe un enlace entre (P_iP_j) y (P_kP_l) entonces $P_j = P_k$.

3. El incremento de verosimilitud a lo largo del nodo (P_i) es el mismo para todas sus copias.

Una vez deshecha la ambigüedad gramatical se puede aplicar directamente el modelo 3-gram. El problema de los modelos acústicos sin representación gramatical se puede resolver de forma análoga al caso anterior. En este caso debemos hacer un backtracking parcial hasta encontrar dos modelos anteriores (en lugar de uno) con representación gramatical. Este hecho produce que haya varias alternativas posibles que se deben evaluar para elegir la mejor. Para evitar esta sobrecarga en la aplicación del

modelo 3-gram, podemos modificar el algoritmo de generación del grafo de forma que sólo se generen nodos que estén representados en el modelo de lenguaje utilizado.

Esta modificación se debe introducir en el paso dos de la generación del grafo (apartado 4.4.1). En el proceso de backtracking para la generación del árbol inverso, al encontramos con un modelo acústico sin representación gramatical, en lugar de generar un nuevo nodo del árbol, seguiremos realizando el backtracking hasta encontrar una palabra que esté representada en el modelo de lenguaje. Una vez encontrada dicha palabra, añadiremos al árbol un nodo con la etiqueta de esa palabra. La trama inicial del nodo será la de la palabra, y la trama final será la de comienzo del backtraking parcial (trama final del primer modelo no gramatical encontrado, ver figura 4-16).

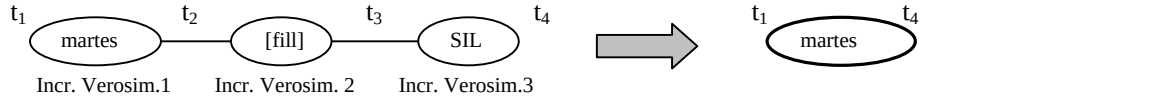

Figura 4-16: Agrupación de varios modelos acústicos en un mismo nodo del grafo.

El incremento de verosimilitud del nodo será la suma de los incrementos a lo largo de los modelos agrupados.

4.5.3 Modelo de lenguaje para Habla Leída

El modelo de lenguaje que vamos a incorporar en el decodificador ha sido generado considerando las frases de referencia de los ficheros de habla leída utilizados en la obtención de los modelos acústicos: 1500 frases que expresan fechas y 500 que expresan horas, en total 2000 frases. Aunque esta cantidad no es muy elevada, nos permitirá caracterizar el dominio elegido de forma suficiente y conseguir tasas de reconocimiento elevadas como veremos a lo largo de este apartado.

Para aprovechar mejor los datos hemos utilizado un modelo N-gram basado en clases. En este tipo de modelo se agrupan varias palabras bajo la misma etiqueta ó clase, de forma que el modelo considere a todas estas palabras como si de un único elemento se tratase. Se deben agrupar palabras bajo una misma clase cuando tengan un comportamiento análogo en las frases utilizadas en el dominio de aplicación. Las clases que hemos definido en nuestro caso han sido las siguientes:

- [días_de_la_semana]: lunes, martes, miércoles, jueves, viernes, sábado, domingo.
- [meses]: enero, febrero, marzo, abril, mayo,..., diciembre.
- [centenas]: ciento, doscientos, trescientos,...,novecientos.
- [decenas]: diez, once, doce, ..., veinte, veintiuno,..., treinta, cuarenta,...,noventa.
- [unidades]: cero, uno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve.
- [estaciones]: primavera, verano, otoño, invierno.

- [ordinales_masc]: primero, segundo, tercero, ..., trigésimo.
- [ordinales_fem]: primera, segunda, tercera, ..., trigésima.

Para la generación de las clases hemos seguido las siguientes recomendaciones:

- Como ya hemos comentado, las clases deben agrupar palabras con comportamiento análogo dentro del dominio considerado.

- Debemos definir tantas clases como regularidades se perciban en las frases, pero garantizando un número mínimo de palabras por clase. Este número mínimo debe ser suficiente como para producir una mejora en la calidad del entrenamiento. Esta mejora será debida al aumento de los datos disponibles para estimar la probabilidad asociada de cada elemento gramatical.

- Se debe evitar en la medida de lo posible palabras que pertenezcan a varias clases.

En una etapa anterior al entrenamiento del modelo, fue necesario sustituir cada palabra por la clase a la que pertenece, deshaciendo las ambigüedades de forma manual. Debido a no disponer de datos suficientes, consideramos las probabilidades de las palabras dentro de una clase como equiprobables. Los resultados de reconocimiento se presentan en la tabla 4-3.

Habla Leída	**Corr (%)**	**Sus (%)**	**Ins (%)**	**Bor (%)**	**WER (%)**
Sin modelo de lenguaje	73,1	17,6	4,5	9,3	**31,4**
2-gram	90,5	5,5	1,6	3,9	**11,1**
3-gram	92,3	4,6	1,2	3,3	**9,0**

Tabla 4-3: Resultados de la incorporación de modelos de lenguaje en Habla Leída.

La reducción relativa del error ha sido del 64,7% para el caso del modelo 2-gram, y del 71,3% para el caso de 3-gram, lo que muestra la potencia del modelo de lenguaje en un dominio restringido como es el de fechas y horas. Al incorporar el modelo 3-gram la reducción respecto al modelo 2-gram no ha sido muy importante. El modelado realizado al nivel de pares de palabras es capaz de capturar muchas de las regularidades existentes en este dominio, dejando poco margen de mejora para modelados más potentes. Aun así, la reducción del error conseguida con el modelo 3-gram respecto el de 2-gram es de un 18,9%.

A la hora de generar el modelo de lenguaje puede ocurrir que existan combinaciones de palabras que aparecen en las frases de prueba que no aparecieron en el entrenamiento por disponer de un número limitado de datos para entrenar el modelo. Por esta razón es necesario realizar un suavizado de las matrices de probabilidades, asignando un valor de probabilidad distinto de cero a aquellas combinaciones no presentes durante el entrenamiento. En los resultados presentados en la tabla 4-3 se utilizó un valor mínimo de probabilidad ajustado con el conjunto de ficheros de validación (ver apartado 4.1.1). En la tabla 4-4, comparamos estos resultados con los obtenidos utilizando la técnica de

suavizado basada en el método de backoff (Katz, 1987). En esta técnica se consideran varios modelos de lenguaje: 3-gram, 2-gram y 1-gram. Para estimar la probabilidad de la secuencia se consulta el modelo más potente (3-gram). En el caso de que no se encuentre información exacta para ese contexto, se pasa a considerar el siguiente de los modelos (2-gram) y así hasta el último modelo (1-gram). A la hora de utilizar un modelo de orden inferior es necesario multiplicar la probabilidad del modelo por un factor de backoff calculado durante la estimación de los modelos. En trabajos anteriores (Rosenfeld, 1994; San-Segundo, D., 2001), podemos ver las fórmulas de cálculo.

Habla Leída	Corr (%)	Sus (%)	Ins (%)	Bor (%)	WER (%)
Prob. min. (2-gram)	90,5	5,5	1,6	3,9	**11,1**
Suavizado Katz (2-gram)	83,8	8,0	2,2	8,15	**18,4**
Prob. min. (3-gram)	92,3	4,6	1,2	3,3	**9,0**
Suavizado Katz (3-gram)	85,1	7,1	1,9	7,9	**16,8**

Tabla 4-4: Comparación de suavizados del modelo de lenguaje para Habla Leída.

Con el suavizado de Katz se permite una mayor flexibilidad. Flexibilidad que no queda bien entrenada si no se disponen de datos suficientes como es nuestro caso (contamos únicamente con 2000 frases). Para un dominio tan restringido como es el de fechas y horas, el considerar un valor umbral de probabilidad bastante restrictivo nos permite obtener mejores resultados. Este fenómeno es el que observamos en el habla leída, donde las frases pronunciadas pertenecen a un conjunto de patrones bastante definidos. Como veremos más adelante este fenómeno no ocurre para el caso de Habla espontánea.

4.5.4 Modelo de lenguaje para Habla Espontánea

A la hora de realizar los experimentos con habla espontánea, el primer paso fue aplicar el modelo utilizado para las frases leídas (apartado anterior). En la tabla 4-5 se presentan los resultados.

Habla Espontánea	Corr (%)	Sus (%)	Ins (%)	Bor (%)	WER (%)
Sin modelo de lenguaje	70,7	19,7	7,3	9,6	**36,6**
Prob. min. (2-gram)	80,2	14,6	5,4	5,2	**25,2**
Prob. min. (3-gram)	78,1	15,1	5,1	6,8	**27,0**

Tabla 4-5 Aplicación del modelo para Habla Espontánea (San-Segundo, D., 2001).

La primera conclusión que podemos sacar de estos resultados es que la reducción del error no ha sido tan importante como en habla leída del 31,1% para 2-gram y 26,3% para 3-gram. La razón de este hecho es que el modelo ha sido entrenado con las frases de referencia utilizadas en habla leída, frases que han sido definidas por el diseñador de la base de datos y no recogen por tanto, efectos derivados de la espontaneidad del habla. Otra prueba de que el modelo de lenguaje no se está ajustando al tipo de frases

pronunciadas en habla espontánea, es el hecho de que el error se incrementa cuando consideramos un modelo más restrictivo como es el 3-gram.

Para validar esta conclusión entrenamos un nuevo modelo de lenguaje con las transcripciones de las 1000 frases espontáneas utilizadas en el entrenamiento acústico. En este caso disponemos de la mitad de datos para modelar un tipo de habla de mayor variabilidad, lo que nos hace pensar que los resultados no serán muy buenos. En la tabla 4-6 se muestran los resultados considerando los dos tipos de suavizado.

Habla Espontánea	Corr (%)	Sus (%)	Ins (%)	Bor (%)	WER (%)
Prob. min. (2-gram)	79,5	12,3	3,5	8,2	**24,0**
Suavizado Katz (2-gram)	79,4	12,5	3,5	8,1	**24,1**
Prob. min. (3-gram)	80,2	12,1	3,4	7,7	**23,1**
Suavizado Katz (3-gram)	80,3	12,0	3,3	7,7	**23,0**

Tabla 4-6: Incorporación del modelo de lenguaje entrenado con 1000 transcripciones de locuciones pronunciadas de forma espontánea.

Como habíamos previsto, los resultados no son tan buenos como en habla leída pero sí que ponen de manifiesto que este modelo se adapta mejor al tipo de habla que estamos considerando: con la mitad de datos para entrenar el modelo, obtenemos menores tasas de error que en la tabla 4-5.

4.6 Consideración de las N mejores hipótesis

En este apartado vamos a analizar el comportamiento del sistema de reconocimiento cuando se consideran varias frases candidato como salida del decodificador. Para evaluar este comportamiento hemos calculado la evolución de la tasa de error con el número de hipótesis. Esta tasa de error se ha calculado considerando la mejor de las hipótesis propuestas (conocida la transcripción).

Esta medida nos da una idea de la mínima tasa de error que se podría conseguir si en un post-procesado posterior (análisis de medidas de confianza o sistema de comprensión por ejemplo), pudiésemos seleccionar la mejor de ellas. Estas hipótesis han sido generadas a partir del grafo de palabras lo que ha supuesto una carga computacional reducida.

A continuación se presenta el pseudocódigo del algoritmo utilizado para generar las N mejores cadenas de palabras a partir de un grafo.

N-MEJORES CADENAS DE PALABRAS A PARTIR DE UN GRAFO

NOTAS:

- Se considerará como heurístico el **incremento de verosimilitud acumulado** a lo largo de los nodos que forman el camino hasta ese punto (Incr. Verosim. ver apartado 4.4.1), junto con la penalización introducida por el modelo de lenguaje 3-gram.
- En cada nodo, se almacenará información de los N mejores nodos antecesores y el Incr. Verosim. Acum. hasta ese punto, dependiendo del nodo antecesor considerado.

/* Procesado de izquierda a derecha */

- Para todos los nodos del grafo realizamos las siguientes acciones:
  - Analizamos la información almacenada en los nodos predecesores del nodo analizado (N incrementos de verosimilitud acumulados por nodo).
  - Obtenemos los N mejores **incrementos a almacenar en el nodo bajo estudio**. Estos valores se obtienen considerando el incremento del nuevo nodo, multiplicándolo por los incrementos acumulados en los nodos predecesores y eligiendo el mayor. Además, el valor obtenido se debe multiplicar por la probabilidad del modelo de lenguaje en la última transición.
  - Además de estos valores debemos almacenar cuál ha sido el **nodo predecesor** del que hemos obtenido cada valor así como la posición en el array donde se guardaba el incremento acumulado del nodo predecesor. De esta forma podemos reconstruir la secuencia de palabras obtenida hasta ese nodo sin más que retroceder por la información almacenada.
  - Se debe comprobar que los Incr. Prob. Acum. y los nodos antecesores almacenados no den lugar a cadenas de palabras idénticas. Para evitar este hecho debemos hacer un backtracking parcial desde cada nodo hasta el nodo inicial para verificarlo.

/* Proceso de backtracking */

- Para cada uno de los N antecesores almacenados en el nodo final se debe realizar un bactracking hasta llegar al nodo inicial obteniendo en cada caso una secuencia de palabras distinta ordenadas de mayor a menor calidad. Este proceso hace uso de la información almacenada en cada nodo.
- El nodo final del grafo no aporta Incr. Verosim. pero sí que introduce una penalización del modelo de lenguaje correspondiente a la probabilidad de que la palabra anterior sea final de frase.

La evolución de las tasas de error, tanto para habla leída como habla espontánea, se presenta en la figura 4-16.

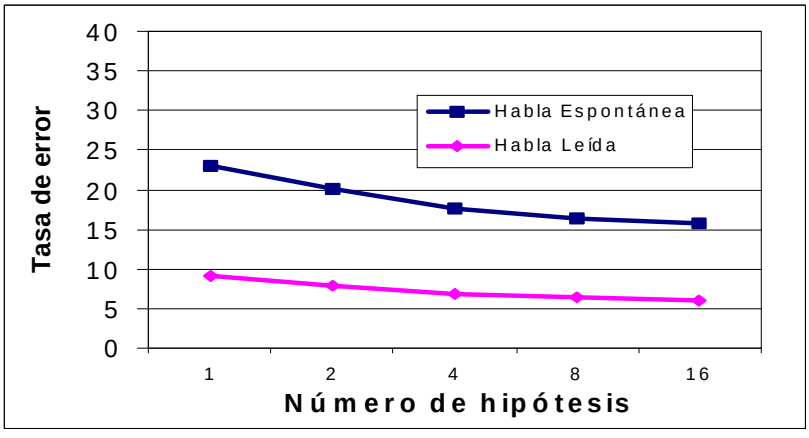

Figura 4-17: Evolución de la tasa de error según el número de hipótesis.

A medida que vamos aumentando el número de hipótesis consideradas, la tasa mínima de error decrece. Como se observa en la figura 4-16 esta tasa de error tiende a saturarse entorno al 16% para habla espontánea y alrededor del 6% para habla leída (N=16). Al aumentar el valor de N llegamos a un punto de saturación de la tasa de error a partir del cuál no se reducirá de forma importante el error. La razón de este comportamiento es que las nuevas hipótesis consideradas se alejan bastante de la mejor hipótesis lo que estadísticamente redunda en una mayor cantidad de errores.

4.7 Conclusiones

Las principales conclusiones que se pueden extraer de este capítulo son las siguientes:

- La utilización de modelos de Markov con 5 estados y transiciones dobles permite una mayor potencia de modelado que redundará en una mejor tasa de reconocimiento siempre que se dispongan de datos de entrenamiento suficientes para estimar estos parámetros.

- El entrenamiento selectivo, aunque no nos ha sido útil para aumentar las tasa de reconocimiento, nos ha permitido evaluar la resolución del modelado acústico utilizado, poniendo de manifiesto la posibilidad de entrenar modelos acústicos más detallados.

- Se describe una simplificación del algoritmo propuesto por Ney para la construcción de un grafo de palabras en ausencia de la técnica de Beam Search. Grafo que permite en una segunda etapa de reconocimiento incorporar modelos de lenguaje 3-gram y calcular varias hipótesis de reconocimiento con bajo coste computacional.

- Se analizan las diferencias entre el habla leída y el habla espontánea proponiendo la utilización de modelos de lenguaje diferentes para cada una de ellas.