3.1 Introducción

Un sistema de reconocimiento de nombres deletreados es un módulo muy útil en el desarrollo de Servidores Vocales Interactivos. Este sistema se puede utilizar como apoyo al reconocimiento de nombres o apellidos en servicios de información telefónica (Lehtinen et al, 2000; Schrâmm et al, 2000; Córdoba et al, 2001) o para la identificación de los nombres de las ciudades en servicios de información y reserva de viajes de tren (Lamel et al, 2000). El reconocimiento de nombres deletreados de forma natural implica el reconocimiento de secuencias de letras conectadas o pronunciadas de forma continua. Esta tarea es bastante complicada, sobre todo por teléfono, debido a la gran confusión entre las letras que forman el alfabeto, la distorsión y limitación del ancho de banda impuesta por el canal de comunicaciones, y la gran variabilidad de las características de los aparatos telefónicos.

En este capítulo se describe el desarrollo de un sistema de reconocimiento de nombres deletreados en castellano. En primer lugar se realiza un análisis de la tarea de deletreo en este idioma y se presentan los resultados obtenidos en trabajos anteriores (San-Segundo et al, 2000b), donde se evalúa el funcionamiento de las principales arquitecturas de reconocimiento utilizadas para otros idiomas (inglés o francés), aplicadas a nuestro caso concreto. Posteriormente, se presenta la arquitectura finalmente utilizada para desarrollar nuestro sistema y los experimentos de ajuste realizados sobre ella. Como veremos más adelante esta arquitectura se basa en una estructura en dos fases: hipótesis y verificación. Finalmente, se evalúa con usuarios reales la mejora conseguida al incorporar el sistema desarrollado en un servicio de información telefónica (Córdoba et al, 2001).

3.2 Tarea de deletreo en castellano

El funcionamiento de un sistema de reconocimiento no sólo depende del tamaño o perplejidad del vocabulario de reconocimiento sino también del grado de similitud existente entre las palabras que forman dicho vocabulario. En la tabla 3-1 presentamos las transcripciones de las pronunciaciones estándar en castellano de las letras. Para ello hemos utilizado el alfabeto fonético internacional (IPA: International Phonetic Alphabet).

En inglés, la mayor dificultad de esta tarea reside en el reconocimiento del conjunto de letras denominado E-set = {B, C, D, E, G, P, T, V, Z} (Loizou et al, 1996). Analizando la tabla 3-1 podemos identificar el conjunto E-set para el caso del castellano = {B, C, CH, D, E, G, P, T}. En este conjunto, los problemas de confusión son muy parecidos a los existentes en inglés: las transcripciones de sus pronunciaciones tienen una estructura muy similar formada por una consonante y el fonema **e.** La única diferencia reside en la consonante que acompaña a la vocal. En este conjunto (E-set), los mayores parecidos acústicos entre letras se producen en los siguientes subconjuntos:

- *Letras B, D, G:* en los tres casos tenemos una consonante oclusiva sorda, o fricativa si no es comienzo de frase. La única diferencia entre las tres

consonantes es el punto de articulación: bilabial, alveolar y velar respectivamente. Generalmente, los castellano–hablantes realizamos muchas pausas entre letras (por la poca costumbre que tenemos a deletrear), lo que hace que la mayoría de las pronunciaciones de estas letras tengan un carácter más oclusivo que fricativo. Esta característica hace que la confusión con el siguiente subconjunto de letras aumente.

Transcripciones de las letras en castellano (IPA)									
A	a	**F**	'e f e	**L**	'e l e	**P**	p e	**V**	'u b e
B	b e	**G**	g e	**LL**	'e ë e	**Q**	k u	**W**	u b e 'd o b l e
C	θ e	**H**	'a t e	**M**	'e m e	**R**	'e r̄ e	**X**	'e k i s
Ch	t e	**I**	i	**N**	'e n e	**S**	'e s e	**Y**	'i g r j e g a
D	d e	**J**	'x o t a	**Ñ**	'e ɲ e	**T**	t e	**Z**	'θ e t a
E	e	**K**	K a	**O**	o	**U**	u		

Tabla 3-1: Transcripciones de las pronunciaciones estándar de las letras en castellano.

- *Letras P, T:* en este caso la consonante es siempre oclusiva y la diferencia vuelve a ser el punto de articulación: bilabial y alveolar respectivamente. Cuando esta oclusión es muy acusada, el ruido de explosión (burst) es muy pequeño y produce que estas letras tiendan a confundirse con la letra E.

En castellano debemos considerar otro conjunto de letras también de gran confusión que denominaremos ExE-set = {F, L, LL, M, N, Ñ, R, S}. En este conjunto, las transcripciones de las letras forman también la misma estructura fonética: **'e _ e**. Estas letras tienen únicamente un fonema diferente (el fonema central), por lo que las diferencias acústicas entre dichas letras son muy pequeñas. Los mayores problemas de discriminación se producen en los siguientes subconjuntos:

- *Letras F y S:* en ambos casos el fonema central es sordo y fricativo, la diferencia está en el punto de articulación: /f/ es labiodental y /s/ alveolar.

- *Letras L y LL:* el fonema central es sonoro y lateral en ambos casos pero el punto de articulación es diferente: /l/ es alveolar y /ë/ palatal.

- *Letras N, M, Ñ*: en este caso también la parte común es el fonema central (sonoro y nasal) pero el punto de articulación es diferente: /m/ bilabial, /n/ alveolar y /ɲ/ palatal. La mayor diferencia reside en la transición entre la nasal y la vocal.

Aparte de estos subconjuntos existen otros dos pares de letras con alto grado de similitud acústica: letras (K, A) y letras (Q, U). En ambos casos, la diferencia es la consonante oclusiva /k/. La duración de esta consonante es sólo una parte muy pequeña de la letra, alrededor de 30 ms (medida obtenida del análisis de 200 ficheros de voz), luego la discriminación es muy difícil. Sin embargo cuando se introducen modelos de lenguaje entre letras, estos pares de letras se diferencian muy bien puesto que al ser

pares (consonante, vocal) tienen comportamientos muy diferentes en cuanto a su orden de aparición en la secuencia de las letras.

Cuando se trabaja con habla continua (sin pausas explícitas entre las palabras que forman la frase), otra fuente importante de errores de reconocimiento es la coarticulación existente entre las palabras, en nuestro caso letras. Este efecto es más peligroso cuando las palabras del vocabulario son más cortas y con pronunciaciones similares como es nuestro caso. En la figura 3-1, podemos ver un ejemplo de errores de reconocimiento cometidos debido al efecto de coarticulación.

NOMBRE DELETREADO:	R	U	B	E	N
TRANSCRIPCIÓN:	'e r̄ e	u	b e	e	'e n e
TRANSCRIPCIÓN RECONOCIDA:	'e r̄ e		'u b e		'e n e
SECUENCIA DE LETRAS RECONOCIDA:	R		V		N

Figura 3-1: Ejemplo de error de reconocimiento debido al efecto de co-articulación.

En este ejemplo podemos ver cómo las pronunciaciones de las letras U y B se han unido para formar la letra V, y la letra E se ha incluido en la pronunciación de la N. Como se puede observar, la secuencia finalmente reconocida dista bastante de la secuencia deletreada.

Otro aspecto importante que se debe tener en cuenta es la necesidad de modelar las pronunciaciones de los nombres alternativos para las letras. Si bien existen unos nombres estándar, cuyas pronunciaciones se presentaron en la tabla 3-1, también existen una serie de nombres alternativos que son ampliamente utilizados por los usuarios para referirse a algunas de las letras. Para desarrollar un reconocedor robusto debemos considerar y modelar acústicamente dichos nombres. Los nombres alternativos más frecuentes se presentan en la tabla 3-2.

Pronunciaciones de los nombres alternativos	
CH	θ e 'a t e
I	'i l a t 'i n a
LL	'e l e 'd o b l e
	'd o b l e 'e l e
R	'e r e
W	'd o b l e 'u b e

Tabla 3-2: Transcripciones de los nombres alternativos más frecuentes para las letras en castellano.

En castellano existe una correspondencia directa entre la escritura de una palabra y su pronunciación (con la excepción de los hiatos que no están acentuados y no forman diptongo). Los castellano–hablantes no estamos acostumbrados a deletrear puesto que habitualmente no lo necesitamos para conocer la escritura de una palabra: se deduce fácilmente a partir de su pronunciación. Por esta razón, en el proceso de deletreo en castellano aparecen con cierta frecuencia los siguientes efectos que se describen a continuación (los porcentajes que se comentan se han obtenido tras el análisis de 200 ficheros utilizados en el entrenamiento de los modelos acústicos y seleccionados aleatoriamente):

- Existencia de pausas grandes entre las pronunciaciones de las letras, y con duración muy variable. En estos ficheros encontramos que en el 83% de las uniones entre letras, el locutor realizó una pausa con una duración media de 0,26 segundos y una desviación típica de 0,24.

- Abundancia de ruidos cometidos por el locutor: tos, respiración fuerte, dentelleo y pausas rellenas como uh, um, er, mm. En el 22,5% de los ficheros se encontró al menos un ruido producido por el locutor. Este porcentaje es muy elevado si tenemos en cuenta que los ficheros considerados pertenecen a la base de datos (Moreno, 1997) en la que los locutores deletreaban nombres que se les daban escritos en un papel. En este caso, la espontaneidad del habla es más reducida y el locutor no tiene que hacer ningún esfuerzo para imaginar la secuencia de letras que forman el nombre, puesto que la tiene escrita.

- Gran cantidad de errores cometidos por el locutor. En el 15,0% de los ficheros el locutor pronunció una secuencia de letras que no coincide con el nombre correspondiente. Porcentaje bastante alto si tenemos en cuenta que los nombres se presentan al locutor de forma escrita, como hemos comentado anteriormente.

3.3 Arquitecturas de reconocimiento

En este apartado vamos a describir el estudio realizado a lo largo de la presente tesis (San-Segundo, et al 2000b), en el que se aplican al castellano varias de las arquitecturas de reconocimiento utilizadas en esta misma tarea para otros idiomas, como el inglés o el francés. En primer lugar, se evaluará una estrategia en dos pasos, después una arquitectura integrada, y finalmente se propondrá una arquitectura de hipótesis y verificación obtenida mediante la combinación de las dos estrategias anteriores. Como veremos, esta tercera solución será la que mejor compromiso nos ofrece entre tasa de reconocimiento y tiempo de proceso, y por tanto, será la utilizada para el desarrollo del sistema de reconocimiento presentado en este trabajo de investigación. La evaluación de las diferentes arquitecturas se ha realizado utilizando un diccionario de 1.000 nombres propios seleccionados aleatoriamente de entre las listas de apellidos y nombres de ciudad más frecuentes en castellano, garantizando que los nombres deletreados por los locutores estén incluidos en dicho diccionario.

Antes de pasar a analizar las diferentes posibilidades, describiremos las medidas de evaluación utilizadas a lo largo del presente capítulo.

3.3.1 Medidas de evaluación

El objetivo del sistema de reconocimiento a desarrollar es obtener el nombre deletreado por el locutor de entre un diccionario de nombres posibles. La medida fundamental de evaluación del sistema completo será la Tasa de Error (TE) cometida por el sistema. Esta Tasa de Error se calculará como el porcentaje de casos en los que nuestro sistema ha fallado, nos ofrece un nombre del diccionario equivocado, en relación con el número de ejemplos con lo que se evalúa.

Generalmente, el sistema de reconocimiento puede estar formados por varios módulos en serie, por lo que se puede calcular la Tasa de Error en puntos intermedios de la cadena. Para alguna de las arquitecturas que se propondrán, existen módulos que no ofrecen como resultado un nombre del directorio, sino una secuencia de letras que pudiera no corresponder con un nombre determinado. Para evaluar estas cadenas de letras consideraremos los porcentajes (respecto de la secuencia de referencia: nombre deletreado) de letras correctamente reconocidas, letras sustituidas, letras insertadas y borradas en la secuencia hipótesis proporcionada por el locutor. A partir de estos porcentajes calcularemos la Tasa de Error de Letra TEL (o Letter Error Rate: LER) y la Precisión de Letra (Letter Accuracy: LA). La Tasa de Error de Letra se obtiene como suma de los porcentajes de sustituciones, inserciones y borrados, y la Precisión de Letra se calcula como el porcentaje complementario de la Tasa de Error de Letra. Veamos las siguientes fórmulas:

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

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

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

donde:

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

Para calcular las sustituciones, inserciones, borrados o letras correctas, se alinean la hipótesis obtenida de la decodificación y la secuencia de referencia (nombre deletreado). Este alineamiento se realiza mediante un algoritmo de programación dinámica en el que se fijan una serie de costes para cada tipo de evento: el coste de una letra 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.

Otra medida de evaluación para este tipo de módulo es el Porcentaje de Cadenas Perfectas (PCP). Este valor representa la proporción de casos en los que la secuencia de letras reconocida corresponde con exactitud con el nombre deletreado por el locutor.

En este capítulo también se presentan resultados de tiempo de proceso (TP) para cada una de las soluciones propuestas. En este caso se darán los resultados en unidades de Tiempo Real, xReal Time (xRT) (Ravishankar, 1996). Una unidad de Tiempo Real corresponde con el tiempo invertido por el locutor para deletrear el nombre. Los resultados presentados se han obtenido utilizando un ordenador Pentium II 350 Mhz con una memoria RAM de 128 Mb.

3.3.1.1 Bandas de probabilidad

A la hora de comparar dos sistemas de reconocimiento o un mismo sistema con varias mejoras/modificaciones, necesitamos disponer de alguna medida para poder asegurar, con cierta fiabilidad estadística, que el sistema de mayor tasa de reconocimiento es mejor que el otro. El método de validación utilizado es el de bandas de probabilidad al 95%. Estas bandas establecen unos límites, inferior y superior respecto de la tasa obtenida, entre los cuales se asegura con un 95% de confianza que el porcentaje real (no el estimado experimentalmente) estará en ese intervalo. Para el cálculo de estas bandas de probabilidad utilizaremos la siguiente fórmula (Weiss and Hasset, 1993):

$$\frac{banda}{2} = 1.96 \times \sqrt{\frac{p \times (100 - p)}{n}} \qquad (3-1)$$

donde **p** es la Tasa de Error al nivel de Letra o de Nombre completo obtenida experimentalmente considerando en la evaluación **n** ejemplos de hipótesis. La Tasa de Error real estará en el intervalo [p-banda/2, p+banda/2] con un 95% de confianza. A la hora de comparar dos sistemas, diremos que tienen comportamientos diferentes cuando sus bandas no se solapen. Para la distribución de ejemplos utilizada en este estudio (San-Segundo et al, 2000b) se consideraron 1200 ficheros de evaluación con una media de 7,6 letras por fichero lo que produce unas bandas de fiabilidad de 1,6% para una Tasa de Error de Letra del 20%, y de 3,4% para una Tasa de Error de Nombre de 10%. A medida que las tasas de error se reducen, las bandas de fiabilidad se estrechan. Por esta razón, si consideramos estas tasas de error como máximas, las bandas de fiabilidad comentadas serán también valores máximos.

3.3.2 Arquitectura en dos niveles

Esta arquitectura consiste en dos pasos de reconocimiento. En primer lugar se obtiene la secuencia de letras que acústicamente mejor se ajusta a la frase pronunciada por el locutor. Esta secuencia de letras se obtiene mediante el algoritmo One-pass. 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). En una segunda fase, con el fin de obtener un nombre del

directorio o diccionario considerado, se compara la secuencia de letras con cada uno de los nombres mediante otro algoritmo de programación dinámica. Este algoritmo considera diferentes penalizaciones para las sustituciones de letras, inserciones, borrados y letras correctas (Fissore et al, 1989). El nombre finalmente seleccionado será aquel que menor distancia presente con la secuencia de letras reconocida. Esta estrategia es similar a la utilizada inicialmente en France Telecom (Jouvet et al, 1993a). En la figura 3-2, podemos ver representada la estructura en dos niveles.

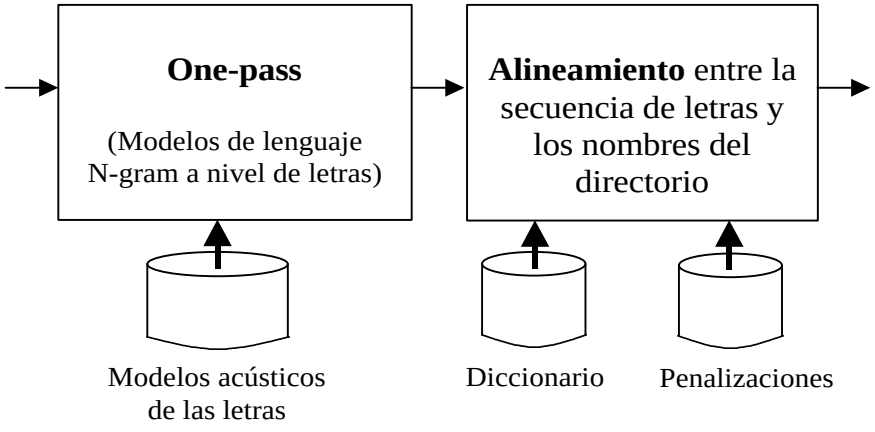


Figura 3-2: Arquitectura en dos niveles: obtención de la secuencia de letras y comparación con los nombres del directorio.

En la tabla 3-2 se presentan los resultados obtenidos para el sistema base y para los casos en los que se consideran modelos de lenguaje 2-gram y 3-gram al nivel de letra en el algoritmo de One-pass. El diccionario está formado por 1.000 nombres diferentes.

	One-pass		**TE (%)**	**TP (xRT)**
	TEL (%)	**PCP (%)**		
Base	20,9	34,3	8,5	1,2
2-gram	18,8	35,4	7,9	1,3
3-gram	11,0	60,4	6,8	3,8

Tabla 3-2: Tasa de Error de Letra (TEL), Porcentaje de Cadenas perfectas (PCP), Tasa de Error total (TE) y Tiempo de Proceso (TP) para el sistema base y considerando modelos de lenguaje en la obtención de la secuencia de letras (San-Segundo et al, 2000b).

Como podemos observar, la reducción importante en el error al nivel de letra (TEL) al introducir el modelo de lenguaje 3-gram no se manifiesta de igual manera en el error final del sistema (TE). La arquitectura utilizada en este caso está formada por dos módulos que se entrenan y optimizan de forma independiente con lo que una mejora importante en uno de ellos puede no repercutir de igual forma en el sistema completo. Más adelante, en el apartado 3.6.4, analizaremos con más detalle este fenómeno.

3.3.3 Arquitectura Integrada

En una arquitectura integrada, la estrategia consiste en construir una gramática muy restrictiva con los nombres del directorio de forma que sólo se permitan secuencias de letras que corresponden con alguno de los nombres. Esta gramática se utiliza para orientar el proceso de reconocimiento realizado mediante el algoritmo de One-pass. En nuestro caso hemos utilizado una estructura en árbol como muestra la figura 3-3.

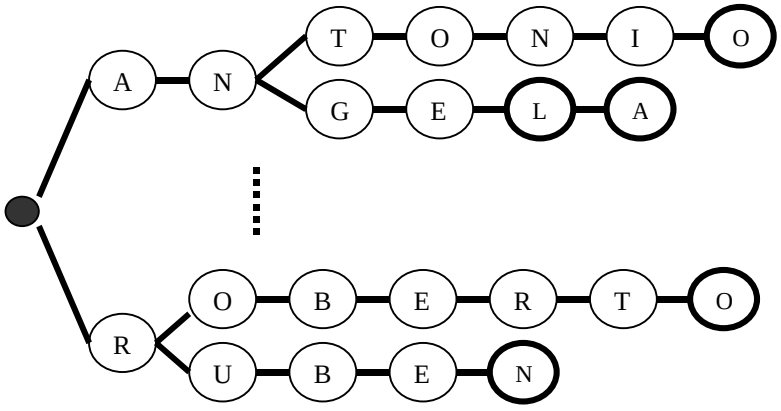


Figura 3-3: Árbol construido con los nombres del directorio.

En negrita hemos resaltado los nodos del árbol que son nodos finales para algunos de los nombres del directorio. Con esta arquitectura, la tasa de reconocimiento conseguida es mayor pero el tiempo de proceso es muy elevado en comparación con la arquitectura en dos niveles.

Uno de los problemas que se plantea en esta estructura es la introducción de los modelos acústicos de ruidos en el algoritmo de reconocimiento. En este análisis (San-Segundo et al, 2000b) se analizaron varias alternativas cuyos resultados se presentan en la tabla 3-3.

	TE (%)	**TP (xRT)**
Sin Modelos de Ruido	5,7	9,2
Considerando Modelos de Ruido al inicio y al final	3,9	9,5
Considerando Modelos de Ruido entre cada par de nodos del árbol	3,1	16,7

Tabla 3-3: Tasa de Error total (TE) y Tiempo de Proceso (TP) para la arquitectura integrada para los casos: sin modelos de ruidos, considerando modelos de ruido en los extremos y considerando modelos de ruido entre cada intersección entre nodos (San-Segundo et al, 2000b).

En primer lugar se muestran los resultados cuando no se considera ningún modelo de ruido. Posteriormente se analiza el caso de considerar modelos de ruidos únicamente en

los extremos, al comienzo y final de la voz, que es cuando aparece una mayor cantidad de ellos. En último lugar, se muestran los resultados al introducir posibles modelos de ruidos para cada transición entre nodos del árbol. El diccionario utilizado para las pruebas está formado por 1000 nombres diferentes. Como se puede observar, considerando únicamente los modelos de ruido al comienzo y al final, podemos hacer frente a gran cantidad de ruidos, consiguiendo tasas de error menores, sin aumentar apenas el tiempo de procesado. Con esta segunda arquitectura, aunque se consiguen tasas de error significativamente menores, los tiempos de procesado han aumentado mucho lo que la hace poco recomendable para su funcionamiento en tiempo real

3.3.4 Arquitectura de Hipótesis y Verificación

Como unión de las dos estrategias anteriores se plantea una arquitectura basada en una estructura de hipótesis y verificación similar a la presentada en (Junqua et al, 1995; Junqua, 1997). Esta arquitectura está forma por dos pasos: en la fase de hipótesis se obtiene la secuencia de letras que mejor se ajusta acústicamente a la secuencia pronunciada, y se compara posteriormente con todos los nombres del directorio mediante un algoritmo de programación dinámica. De esta manera obtenemos los N mejores nombres. Estos nombres se pasan a la fase de verificación donde se construye con ellos una gramática muy restrictiva (un árbol similar al presentado en la figura 3-3), volviendo a realizarse un proceso de reconocimiento pero esta vez sobre esta estructura. En la figura 3-5, se muestra el diagrama de bloques correspondiente.

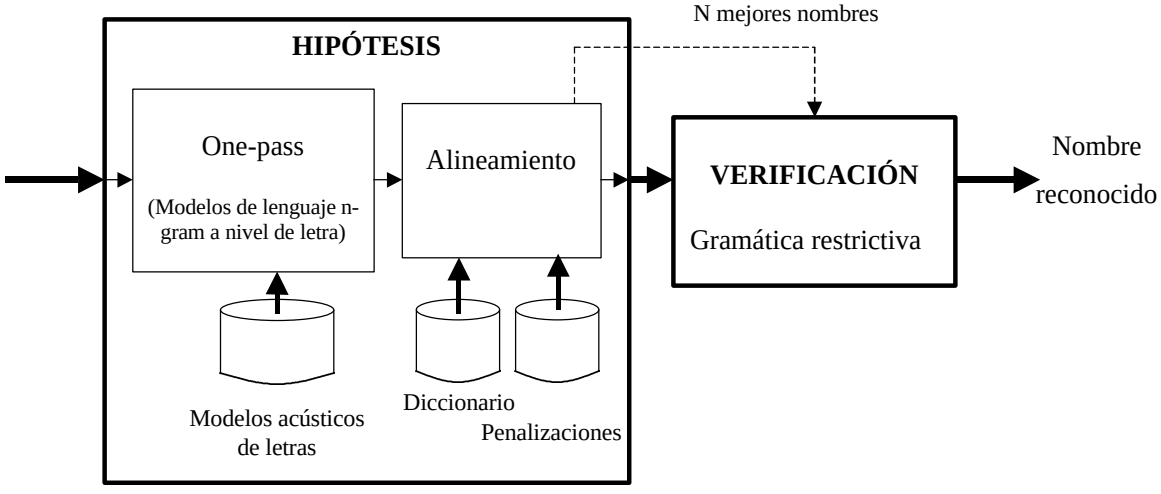


Figura 3-4: Arquitectura de hipótesis y verificación (San-Segundo et al, 2000b).

Como se puede ver, la fase de hipótesis está formada por una estructura de reconocimiento en dos etapas, y la fase de verificación lo forma una estrategia integrada. Esta arquitectura tiene como finalidad conseguir un buen compromiso entre tasa de reconocimiento y tiempo de procesado: en primer lugar, un sistema poco costoso se encarga de realizar una preselección de los nombres más parecidos con los que posteriormente se construye una gramática para la etapa de verificación. En este caso, al utilizar únicamente una selección de nombres del directorio, el árbol construido en la

fase de verificación será mucho menor y el tiempo de procesado muy inferior. Veamos los resultados de reconocimiento en la tabla 3-4.

En esta tabla se analizan también las diferentes posibilidades para la incorporación de los modelos de ruido en la gramática de la etapa de verificación. Al igual que en el caso anterior, el modelado de los ruidos al comienzo y al final permite conseguir buenas tasas de reconocimiento sin aumentar considerablemente el tiempo de procesado.

	TE (%)	**TP (xRT)**
Sin Modelos de Ruido	6,3	2,2
Considerando Modelos de Ruido al inicio y al final	4,5	2,3
Considerando Modelos de Ruido entre cada par de nodos del árbol	3,9	2,9

Tabla 3-4: Tasa de Error total (TE) y Tiempo de Proceso (TP) para la arquitectura de hipótesis y verificación para los casos: sin modelos de ruidos, considerando modelos de ruido en los extremos, y considerando modelos de ruido para cada transición entre nodos (San-Segundo et al, 2000b).

Esta tercera arquitectura basada en una estructura de hipótesis y verificación es la que mejor compromiso ofrece entre tasa de error y tiempo de procesado. Por esta razón es la arquitectura que consideraremos en el desarrollo del sistema de reconocimiento de nombres deletreados desarrollado en la presente tesis.

3.4 Características generales del sistema de reconocimiento utilizado

El sistema propuesto está basado en una arquitectura de hipótesis y verificación similar a la propuesta en el apartado 3.3.4 (San-Segundo et al, 2000b) y a la descrita por Junqua (Junqua, 1997) para el caso del inglés. En el estudio presentado en este capítulo consideraremos la introducción de una nueva topología de HMM (Hidden Markov Models o Modelos Ocultos de Markov) con modelos de silencio contextuales a los modelos de las letras. Por otro lado, generaremos un grafo de letras similar al grafo de palabras, propuesto y descrito en el apartado 4.4, para incorporar modelos de lenguaje N-gram en el algoritmo de decodificación, y obtener las N mejores secuencias de letras como resultado del reconocimiento. Los trabajos de investigación descritos en este capítulo han sido aceptados para ser publicados en Speech Communication (San-Segundo et al, 2002).

3.4.1 Modelado acústico utilizado

Los modelos acústicos utilizados son modelos ocultos de Markov (HMM) (Rabiner et al, 1986, Huang et al 1990). Un modelo oculto de Markov no es más que un autómata

estocástico de estados finito, constituido por un conjunto finito de estados $Q=\{q_1,...,q_i,...,q_L\}$ organizados según una topología determinada, cada uno de los cuales lleva asociada una distribución de probabilidad o función densidad de probabilidad (fdp). Entre los diferentes estados se definen posibles transiciones que llevan asociadas unas probabilidades de transición determinadas.

Un modelo de Markov puede modelar secuencias de vectores característicos $X=\{x_1,...,x_k,...,x_K\}$ pertenecientes a un proceso estacionario, en el que cada segmento de ese proceso puede ser asociado con un estado específico del modelo. Por lo tanto, cuando se utiliza un modelo determinado para modelar una secuencia de vectores $X=\{x_1,...,x_k,...,x_K\}$, esta secuencia es analizada como una sucesión de estados estacionarios $S=\{s_1,...,s_n,...,s_N\}$, N K, con transiciones entre ellos, donde cada uno de estos estados s_n se corresponde con alguno de los estados del modelo de Markov q_i. En la figura 3-5 se puede observar un ejemplo del modelo de Markov que vamos a utilizar. Como se puede ver, el modelo utilizado es lineal formado por N estados en los que permitimos transiciones desde un estado al mismo (qi), al siguiente (qi+1) y una transición doble saltándose un estado (qi+2).

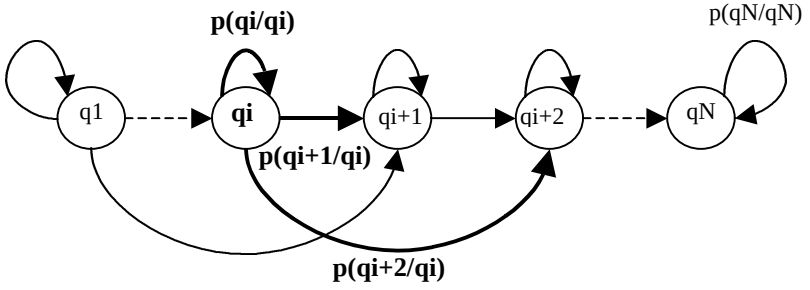


Figura 3-5: Modelo de Markov lineal.

Un modelo de Markov queda determinado por el número de estados N y las siguientes matrices de probabilidades:

- **A** = [Pij, 1≤ i,j ≤ N]; P_{ij}: probabilidad de transitar del estado 'i' al estado 'j'.
- **B** = [Pj(k), 1≤ j ≤ N, 1≤ j ≤ K]; $P_j(k)$ probabilidad de emisión del vector 'k' en el estado 'j'.

En nuestro sistema vamos a utilizar modelos continuos en los que la matriz B de probabilidades se convierte en una función densidad de probabilidad (fdp). Estas fdps se obtienen como combinación lineal de un conjunto de funciones gaussianas. La densidad de probabilidad de un vector 'k' se calculará como la combinación lineal de las aportaciones de cada una de las gaussianas. La razón de utilizar gaussianas como funciones base es porque necesitan sólo dos parámetros (media y varianza) para que queden definidas.

En este sistema utilizaremos modelos acústicos independientes para cada una de las pronunciaciones de las letras. El número de estados considerado en cada modelo es proporcional a la duración media de la pronunciación de cada letra. El modelo más corto es para la letra I con 9 estados, y el modelo más largo con 48 estados corresponde

a la letra W. La razón de utilizar modelos de Markov continuos es porque son los más potentes para hacer frente a la gran confusión existente entre las letras que forman el vocabulario de reconocimiento. El número de gaussianas considerado en cada estado para definir la fdp es proporcional a la cantidad de datos de entrenamiento utilizados. Consideraremos un número mínimo de 3 gaussianas y un número máximo de 9. El proceso de entrenamiento de los modelos acústicos se realizó mediante un proceso estándar: utilizamos el algoritmo de Viterbi para asignar cada trama a un estado de los modelos acústicos y posteriormente se considera la reestimación EM (Expectation-Maximisation) para obtener los parámetros de los modelos en cada iteración. Como punto de partida se utiliza el algoritmo LBG (Linde et al, 1980) para obtener la primera versión de las gaussianas consideradas en cada estado.

3.4.2 Estructura de reconocimiento

En una primera fase se realiza un análisis o parametrización de la señal de voz. Este proceso se va realizando cada 10 ms con ventanas de análisis de 25 ms. Las muestras de voz que corresponden a cada una de las ventanas analizadas las denominaremos tramas de voz. En los experimentos presentados, consideraremos como parámetros de cada trama 10 coeficientes cepstrales, la energía local de la trama y sus respectivas derivadas, tanto de la energía como de los coeficientes cepstrales (en total 22 parámetros para caracterizar cada trama de voz). La parametrización utilizada es la RASTA-PLP (Hermansky et al, 1991) y propuesta por Junqua (Junqua, 1997) donde se puede consultar una descripción detallada de diferentes parametrizaciones aplicadas a la tarea de deletreo en inglés.

En la etapa de hipótesis consideraremos dos fases:

- En la primera fase se aplica el algoritmo de One-pass para obtener la secuencia de letras que mejor se ajusta acústicamente al nombre pronunciado. En esta etapa, consideraremos la introducción de una nueva topología con modelos de silencios contextuales a los modelos de letra, la incorporación de modelos acústicos para los ruidos, la utilización de modelos de lenguaje 2-gram y 3-gram en el espacio de búsqueda, y la obtención de las N mejores cadenas de letras en lugar de una única secuencia. En estos dos últimos puntos, la generación de una grafo de letras tendrá un impacto importante al permitirnos aplicar dichas técnicas con un aumento reducido del tiempo de proceso.

- Una vez obtenidas las N mejores secuencias de letras, el objetivo de la segunda fase es obtener los M mejores nombres del directorio. Para ello, las N secuencias de letras se comparan con todos los nombres del diccionario mediante un algoritmo de programación dinámica. Este algoritmo aplica diferentes penalizaciones para las posibles sustituciones, borrados e inserciones de letras en la cadena. En esta comparación se calcula el mejor camino de alineamiento en el espacio de búsqueda entre secuencia y nombre, así como el coste asociado con dicho camino. Dada una secuencia de letras, se seleccionan los nombres del diccionario con menor coste de alineamiento.

Por otro lado, para entrenar las penalizaciones utilizadas en el algoritmo, cada secuencia de letras (obtenida con el conjunto de validación) se compara con el nombre deletreado y se cuentan las sustituciones, borrados e inserciones producidas (Fissore, 1989). Los valores de las penalizaciones se calculan como el logaritmo del inverso de la probabilidad obtenida para cada uno de los eventos. Esta probabilidad se estima a partir de la cuenta de eventos realizada. Este proceso se repite de forma iterativa hasta que los valores de las penalizaciones no varían sustancialmente. Consideraremos como valores iniciales los siguientes: 1 para las inserciones y borrados, 2 para las sustituciones y 0 para las letras correctas. Un detalle importante es que cada vez que la fase anterior de obtención de las N mejores secuencias de letras sufre alguna modificación, es necesario volver a entrenar las penalizaciones del algoritmo de alineamiento para que aprendan la nueva tipología de errores que aparecen.

En la fase de verificación utilizamos los M nombres del diccionario que más se parecen a las secuencias de letras obtenidas, y construimos con ellos una gramática en forma de árbol (figura 3-3). Una vez construida la gramática se ejecuta el algoritmo de reconocimiento One-pass sobre ella, obteniendo finalmente el nombre reconocido. En esta etapa, el tiempo consumido es muy pequeño debido a dos factores:

- Por un lado, el número de nombres candidato considerados es muy reducido (menor de 50): en el apartado 3.7 se estudia la selección de este número.

- Otra razón es que en esta etapa utilizamos los mismos modelos acústicos que los utilizados en la fase de obtención de la cadena de letras en la etapa de hipótesis. Debido a esto las probabilidades de emisión de cada trama en cada estado ya están calculadas y almacenadas de la etapa anterior.

En nuestros experimentos utilizamos los mismos modelos acústicos en las dos etapas aunque se podrían utilizar modelos más detallados. En la figura 3-6, podemos ver el diagrama de bloques del sistema de reconocimiento.

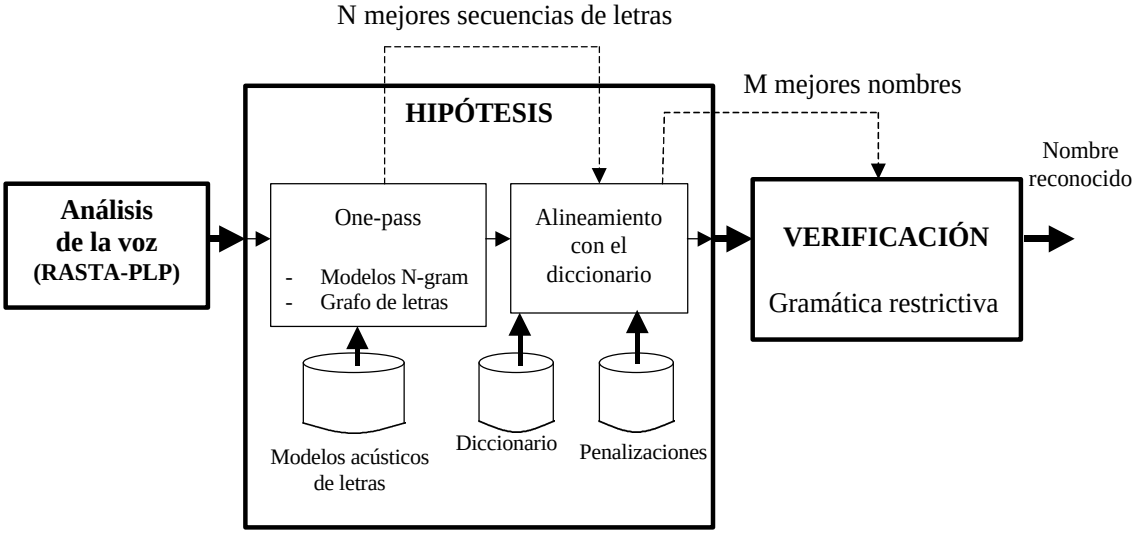


Figura 3-6: Diagrama de bloques del reconocedor de nombres deletreados.

3.5 Base de datos

La base de datos de voz utilizada en los experimentos realizados ha sido SpeechDat en castellano (Moreno, 1997). Esta base de datos ha sido grabada a través de la red telefónica fija en España utilizando 1.000 locutores diferentes. Cada locutor tuvo que deletrear el nombre de una ciudad, un apellido y una secuencia aleatoria de letras (estas secuencias garantizan un número mínimo de ejemplos de entrenamiento para cada letra), disponiendo en total de 22.800 letras en 3.000 ficheros de voz. Las secuencias aleatorias de letras se usan únicamente para el entrenamiento de los modelos acústicos. Por otro lado, de los ficheros con nombres de ciudad y apellidos se han extraído aleatoriamente 600 ficheros (300 locutores diferentes) para generar dos conjuntos de validación diferentes: el primero (300 ficheros, 150 locutores) para ajustar las penalizaciones utilizadas en el algoritmo de alineamiento de las secuencias de letras con los nombres del diccionario, y el segundo (otros 300 ficheros de 150 locutores diferentes), para la evaluación de las diferentes estrategias de reconocimiento. También se seleccionaron otros 300 ficheros de 150 locutores diferentes para realizar el proceso de test final. El resto de ficheros, 2.100, se utilizaron para el entrenamiento de los modelos acústicos[1]. Esta distribución de ficheros la hemos repetido 6 veces (6-Round Robin training). Los experimentos que se presentan corresponden con la media de los obtenidos con cada una de las 6 distribuciones.

En los experimentos consideraremos varios diccionarios con tamaños diferentes: 1.000, 5.000 y 10.000 palabras. Estos diccionarios se han generado seleccionando aleatoriamente nombres de ciudades y apellidos de los directorios de ciudades y apellidos de España. Los nombres de ciudades y apellidos deletreados en la base de datos están incluidos en todos los diccionarios. La confusión media para estos diccionarios es de 0,2, 0,5 y 0,9 respectivamente. Esta medida se calcula como el número medio de pares de nombres del diccionario que difieren únicamente en la sustitución de una letra. Estos valores son una medida de la confusión de los diccionarios (Cole et al, 1991a; Junqua, 1997) y proporcionan una idea de la dificultad de la tarea de reconocimiento. Por ejemplo, para el tercer diccionario (con 10.000 palabras) hay 9.038 pares de nombres que difieren en una letra. Este valor corresponde con una media de 0.9 confusiones por nombre.

Para evaluar la primera fase de la etapa de hipótesis (extracción de la secuencia de letras) y las diferentes soluciones propuestas, consideraremos los porcentajes de letras sustituidas, borradas e insertadas, la Tasa de Error de Letra y el Porcentaje de Cadenas Perfectas (medidas descritas en el apartado 3.3.1). Para evaluar el alineamiento de la secuencia de letras con los nombres del diccionario (segunda fase de la etapa de hipótesis) y la etapa de verificación, consideraremos la Tasa de Error al nivel de Nombre completo. Los intervalos de confianza calculados al 95% para esta distribución

[1] Podemos pensar que esta división especial de los ficheros de la base de datos no garantiza que los experimentos realizados sean independientes del locutor. La razón es que al considerar todas las cadenas aleatorias de letras para entrenar los modelos acústicos, estamos considerando un fichero de cada locutor utilizado en validación y test. En el apéndice B presentamos algunos de estos experimentos excluyendo estos ficheros y demostramos cómo las diferencias en los resultados no son significativas puesto que existe sólo un único fichero de cada locutor y su influencia sobre la tasa es mínima.

de ficheros son de 1,4% para una Tasa de Error de Letra del 30%, y de 3,2% para una Tasa de Error de Nombre de 15%. A medida que las tasas de error se reducen, las bandas de fiabilidad se estrechan. Por esta razón, si consideramos estas tasas de error como máximas, los intervalos de confianza comentados serán también valores máximos. Estos intervalos de confianza se han obtenido mediante la ecuación 3-1.

3.6 Etapa de Hipótesis

Como se comentó en el apartado 3.4.2, la etapa de hipótesis está formada por dos módulos: el algoritmo One-pass para extraer las N mejores cadenas de letras y su posterior comparación con los nombres del diccionario. En este apartado nos centraremos en el ajuste del algoritmo One-pass aunque se presentarán resultados para los dos módulos. Los experimentos presentados en este apartado se han realizado considerando el diccionario con 1.000 nombres.

3.6.1 Sistema de referencia

Como sistema base para la fase de extracción de la secuencia de letras consideraremos un algoritmo One-pass con 35 modelos de letra, considerando tanto las pronunciaciones estándar como las pronunciaciones alternativas, y un modelo simple (tres estados) de silencio o pausa entre letras.

Con el fin de permitir cierta flexibilidad en las duraciones de las pausas hemos introducido una transición hacia atrás en el modelo del silencio desde el último estado hasta el primer estado. Esta transición no es necesaria en el proceso de reconocimiento puesto que permitimos que se concatenen tantos modelos de silencio como sea necesario. El problema surge en la fase de entrenamiento en la que, al no disponer de las marcas de comienzo y final de cada letra, no podemos saber las duraciones exactas de las pausas, y por tanto, la cantidad de modelos de silencio a considerar en el alineamiento. Añadiendo esta transición permitimos que se puedan concatenar de forma no supervisada tantos modelos de silencio como sea necesario. En la tabla 3-5 podemos ver los resultados obtenidos con el sistema de referencia.

Sistema de referencia						
Algoritmo One-pass					**Etapa de hipótesis**	
Sus (%)	**Ins (%)**	**Bor (%)**	**TEL (%)**	**PCP (%)**	**TE (%)**	**TP (xRT)**
20,2	6,5	3,7	30,4	21,3	16,6	0,9

Tabla 3-5: Resultados del sistema de referencia: porcentajes de Sustituciones (Sus), Inserciones (Ins), Borrados (Bor), Tasa de Error de Letra (TEL), Porcentaje de Cadena Perfectas (PCP), y la Tasa de Error (TE) y el Tiempo de Proceso (TP) de la etapa de hipótesis.

Como podemos observar, el porcentaje de sustituciones es bastante elevado debido a la gran confusión entre las letras que forman el vocabulario de reconocimiento. Otro aspecto importante es el porcentaje tan reducido de cadenas perfectas, sólo un 21,3%, lo que muestra la dificultad de reconocer una secuencia de letras que se ajuste

perfectamente al nombre deletreado. Por esta razón, es necesario utilizar el módulo encargado de alinear la secuencia de letras con los nombres del diccionario para incrementar la tasa de acierto al nivel de nombre. En este caso el incremento ha sido considerable, obteniendo una Tasa de Error del 16,6 (tasa de acierto del 83,4%) para el diccionario de 1.000 nombres.

3.6.2 Nueva topología HMM con silencios contextuales

En el proceso de deletreo en general, y entre los castellano–hablantes en particular, las pausas entre las letras pueden cambiar su duración de forma considerable dependiendo fuertemente de los hábitos del locutor en el proceso de deletreo. Para hacer frente a esta característica, hemos considerado una nueva topología HMM con modelos de silencios contextuales incorporados a cada modelo de letra. En esta topología cada modelo de letra dispone de dos modelos de silencio, con tres estados cada uno, para modelar el posible silencio anterior y posterior a la letra. En la figura 3-7 podemos ver un ejemplo para la letra A.

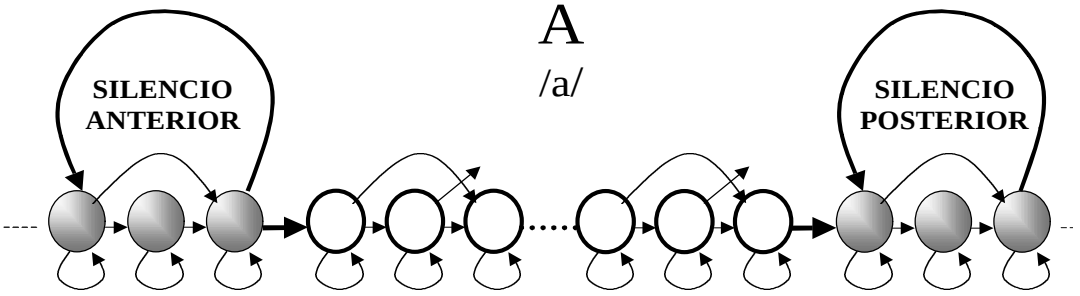


Figura 3-7: Topología del modelo de la letra A con silencios contextuales incorporados.

Al igual que en el sistema de referencia, hemos considerado una transición hacia atrás desde el último estado del modelo de silencio hasta el primer estado con el fin de modelar diferentes duraciones de las pausas. Considerando que tenemos 35 modelos de letra, necesitaremos 70 modelos de silencio de tres estados, 35 modelos anteriores y 35 modelos posteriores. La correcta estimación de estos modelos ha sido posible gracias a que en los ficheros de entrenamiento se dispone de gran cantidad de tramas de silencio entre las letras (ver los efectos del deletreo en castellano, apartado 3.2).

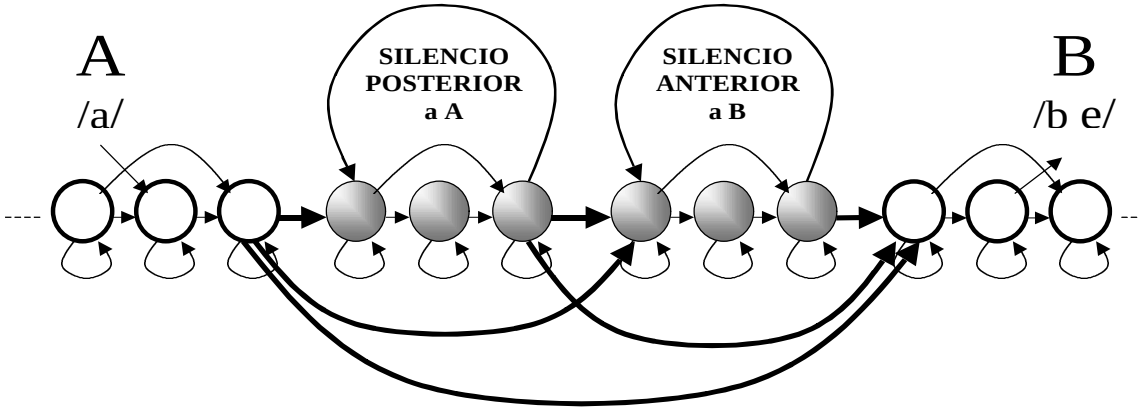


Figura 3-8: Nuevas transiciones entre letras con los modelos contextuales.

En la figura 3-8 se muestran las nuevas transiciones entre modelos de letra que aparecen en el algoritmo One-pass (reconocimiento) al considerar los modelos de silencio contextuales. Los resultados obtenidos con esta nueva topología se presentan en la tabla 3-6. Como podemos ver, al incorporar los modelos de silencios contextuales se ha incrementado ligeramente el tiempo de proceso pero el error al nivel de letra se ha reducido en 6,6 puntos (21,9% relativo) obteniendo una tasa de error de 10,9% al nivel de nombre completo reconocido. Estas reducciones son superiores a las bandas de fiabilidad de estos experimentos con lo que la mejora conseguida es significativa.

Esta mejora tan importante se debe a dos razones fundamentalmente:

- Por un lado, como ya comentamos en el apartado 3.2, el poco hábito de los castellano–hablantes a la hora de deletrear hace que realicemos muchas pausas entre las letras y de gran variabilidad. Según las estimaciones anteriores, en el 83% de las uniones entre letras el locutor realizó una pausa con una duración media de 0,26 segundos y una desviación típica de 0,24. Al considerar varios modelos de silencio contextuales a las letras, el modelado de las pausas resultante es más potente y robusto permitiendo una mejor segmentación de la señal de voz.

- Por otro lado, en el caso de algunos modelos de silencio hemos observado que la probabilidad de la transición hacia atrás es muy baja. Este hecho revela que en algunos casos durante el proceso de entrenamiento, los estados correspondientes al modelo de silencio han pasado a modelar parte de las características de las letras pasando a incrementar la longitud del modelo de la letra. Esta doble funcionalidad ha permitido dar una mayor flexibilidad al modelado acústico permitiendo por un lado modelar las pausas contextuales o incrementar la longitud del modelo de letra según las necesidades concretas de cada letra.

3.6.3 Incorporación de modelos de ruido

Además de la gran variabilidad de las pausas entre letras, otros efectos frecuentes que aparecen en el proceso de deletreo en castellano son falsos comienzos, dudas, pausas rellenas y errores que repercuten en el aumento de la tasa de error obtenida. Por otra parte, al trabajar con señal telefónica, esta puede verse contaminada por gran variedad de ruidos, producidos por la propia línea de comunicación o debidos al ambiente en el que está hablando el locutor; desde una cabina, desde una oficina, desde una habitación con música de fondo,...

Para hacer frente a este tipo de fenómenos hemos considerado 4 tipos de ruidos diferentes y hemos entrenado modelos acústicos específicos para cada uno de ellos. La clasificación realizada ha sido la misma que la propuesta en la base de datos utilizada (Moreno, 1997):

- *[fil]: Pausas rellenas.* Con este modelo se trata de detectar pausas cometidas por el locutor con algún tipo de sonido que trata de mostrar su intención de continuar hablando. Algunos ejemplos son: uh, um, er, ah, mm.

- *[spk]: Ruidos del Locutor.* En esta categoría se engloban todo tipo de ruidos cometidos por el locutor que no corresponden con la cadena de letras pronunciada. Algunos ejemplos pueden ser: ruido con los labios, tos, respiración fuerte, estornudo, click con la lengua y risa.

- *[sta]: Ruido Estacionario.* Esta categoría contiene todo tipo ruido de fondo que no sea intermitente y que tenga más o menos una misma amplitud a lo largo de toda la elocución. Por ejemplo: motor de un coche, ruido de una carretera, ruido del canal de comunicaciones, ruido GSM, entorno de voces de fondo (fiesta o reunión) y ruido de ambiente de la calle.

- *[int]: Ruido Intermitente.* En este modelo se reflejan ruidos con naturaleza intermitente o ruidos cuyo espectro cambia a lo largo del tiempo. Por ejemplo; música, voz de fondo, niño llorando, ring del teléfono, portazo, timbre de una puerta y ruidos de papeles.

El entrenamiento de los modelos acústicos para estos ruidos ha sido posible gracias a que la base de datos SpeechDat dispone de su anotación a lo largo de las transcripciones de los ficheros. Si bien no están delimitados temporalmente sí que está definida su posición dentro de la secuencia de letras lo que permite realizar un entrenamiento no supervisado.

Para evaluar la importancia de este tipo de ruidos analizamos el conjunto de ficheros de entrenamiento y observamos que más del 65,0% de los ficheros contienen uno o más de estos ruidos lo que refleja la gran importancia de su modelado. Como vimos en el apartado 3.2 en el 22,5% de los ficheros aparece algún ruido correspondiente a los tipos [fil] ó [spk]. Los resultados obtenidos considerando estos modelos de ruidos en el espacio de búsqueda son los presentados en la tabla 3-6 donde se pueden comparar con los obtenidos para los casos: sistema de referencia y considerando los modelos de silencios contextuales.

	Algoritmo One-pass					**Etapa de hipótesis**	
Sistema	**Sus (%)**	**Ins (%)**	**Bor (%)**	**TEL (%)**	**PCP (%)**	**TE (%)**	**TP (xRT)**
Sistema de Referencia	20,2	6,5	3,7	30,4	21,3	16,6	0,9
Silencios Contextuales	17,3	4,2	2,3	23,8	27,8	10,9	1.2
Silencios contextuales + Modelos de ruidos	15,9	1,1	2,3	19,3	34,3	8,0	1.2

Tabla 3-6: Resultados del Sistema de Referencia, considerando la nueva topología de HMM con silencios contextuales y considerando los modelos de ruido además de la nueva topología de HMM: porcentajes de Sustituciones (Sus), Inserciones (Ins), Borrados (Bor), Tasa de Error de Letra (TEL), Porcentaje de Cadena Perfectas (PCP), y la Tasa de Error (TE) y el Tiempo de Proceso (TP) de la etapa de hipótesis.

A tenor de los resultados podemos concluir que el modelado de los ruidos se presenta como una técnica complementaria al modelado de las pausas mediante silencios contextuales, permitiendo reducir la tasa de error de letra (TEL) 4,5 puntos (18,9% relativo). Esta mejora se ha debido en mayor medida a la importante reducción del número de inserciones (reducción de 3,1 puntos). Esta disminución del error al nivel de letra ha producido una reducción de 2,9 puntos (26,6% relativo) en el error al nivel de nombre reconocido. Esta mejora en la tasa se ha obtenido sin aumentar apenas el tiempo de procesado. Si bien la mejora del error al nivel de letra es significativa (4,5% con una banda de fiabilidad de 1,4%), la mejora al nivel de nombre completo no lo es (2,9% con una banda de fiabilidad de 3,3%).

3.6.4 Modelos de lenguaje de letras

Cuando el nombre deletreado por el locutor pertenece a una lista finita de nombres (directorio) como es nuestro caso, esta lista nos puede ofrecer información muy útil que podemos incorporar de varias maneras posibles al módulo de extracción de la secuencia de letras (Hild y Waibel, 1997). Una manera muy sencilla de considerar esta información es definiendo modelos de lenguaje (ML) estocásticos de tipo N-gram e incluyéndoles en el espacio de búsqueda. Las gramáticas de tipo N-grams consideran las últimas N-1 letras para predecir la letra que con mayor probabilidad las seguirá en la secuencia. Dependiendo del valor de N, tendremos gramáticas 1-gram, 2-gram, 3-gram, etc... Según aumentamos el valor de N (orden de la gramática) mayor es la restricción que se impone y menor la perplejidad en el reconocimiento, consiguiendo tasas mayores. En los experimentos que se presentan en este apartado hemos calculado modelos 2-gram y 3-gram a partir de los nombres que forman el diccionario de 1.000 nombres. Para los pares o tripletas de letras no observadas en los nombres del diccionario, sus probabilidades correspondientes se suavizaron utilizando un valor mínimo de probabilidad. Este valor se ha ajustado mediante el primero de los dos conjuntos de ficheros utilizados para validación y ajuste de parámetros del reconocedor (ver apartado 3.5).

La incorporación del modelo 2-gram en el algoritmo One-pass es bastante fácil puesto que no es necesario modificar dicho espacio de búsqueda: sólo es necesario considerar la probabilidad del modelo de lenguaje al considerar las posibles transiciones entre los modelos de dos letras. En el caso del modelo 3-gram por el contrario, sí es necesario modificar este espacio de búsqueda: debemos duplicar cada modelo de letra tantas veces como letras pueden precederla con el fin de crear estados gramaticales independientes (Colás, 1999).

En la figura 3-9 se muestra el espacio de búsqueda generado para el caso de aplicar el modelo de lenguaje 3-gram. Los nodos del tipo (X, Y) representan los modelos acústicos de la letra Y con información proveniente de la letra X. En esta figura, también hemos incluido los nodos del tipo (INI, Y) para tener en cuenta las letras que pueden ser primeras letras de secuencia a tenor de los nombres del directorio: representan los modelos acústicos de la letra Y con información proveniente del nodo inicial (INI).

El principal problema que encontramos en esta estructura fue la incorporación en este espacio de búsqueda de modelos acústicos para hacer frente a los diferentes ruidos modelados. La solución ideal sería duplicar los modelos de ruido tantas veces como nodos tengamos en la estructura, posibilitando la existencia de un ruido entre dos letras cualesquiera. Esta solución incrementa enormemente el espacio de búsqueda dando lugar a un aumento considerable del tiempo de proceso.

Considerando los análisis realizados en los apartados 3.3.3 y 3.3.4, y en el trabajo de Kuroiwa (Kuroiwa et al, 1999), analizamos la distribución de ruidos a lo largo de la secuencia de letras. Haciendo un estudio sobre el conjunto de ficheros de entrenamiento pudimos observar que el 67,3% de los ruidos aparecían al comienzo de la secuencia de letras, el 26,2% aparecían al final, y sólo en un 6,5% de los casos los ruidos aparecían entre la pronunciación de dos letras consecutivas. A tenor de estos resultados decidimos incorporar los modelos de ruido únicamente en los nodos de comienzo y final (ver figura 3-9). Con esta solución no incrementamos el espacio de búsqueda de manera importante pero a la vez pudimos hacer frente a una gran cantidad de ruidos (en los apartados 3.3.3. y 3.3.4. fue la solución que obtuvo mejor compromiso entre tiempo de proceso y tasa de reconocimiento).

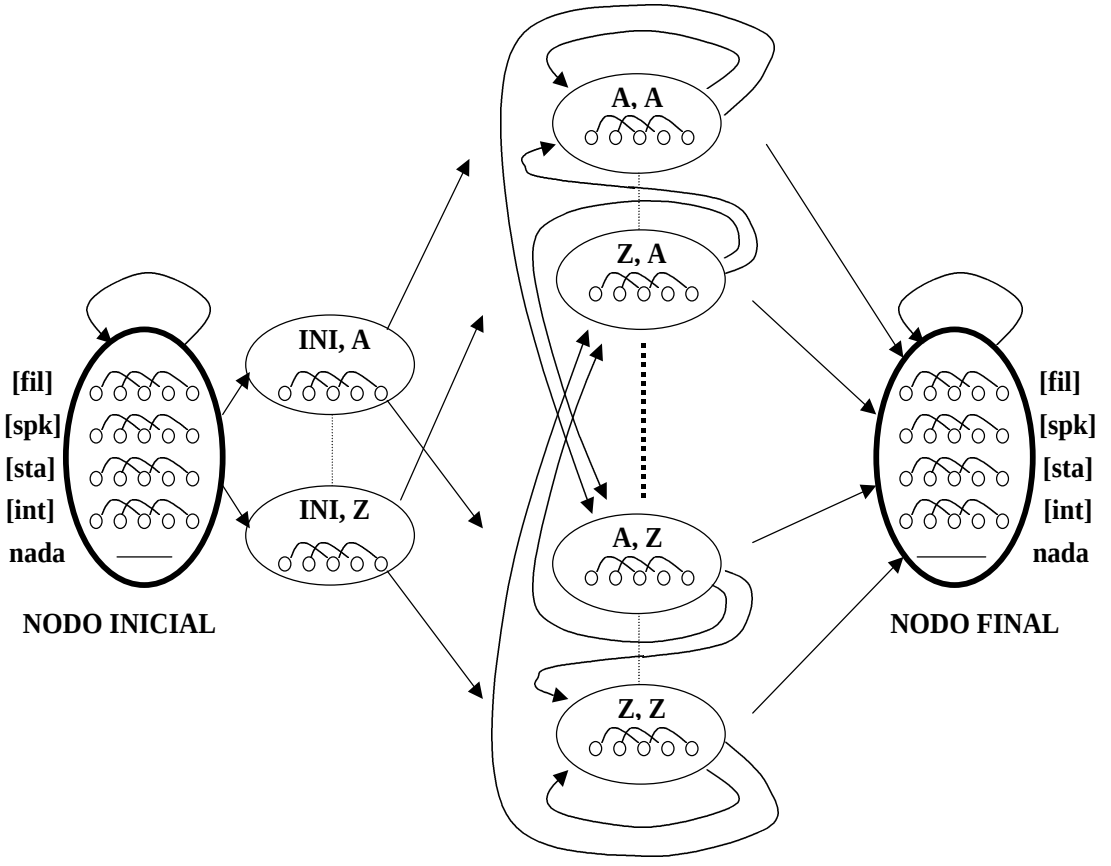


Figura 3-9: Modelos de ruido incorporados en el espacio de búsqueda para aplicar el modelo de lenguaje 3-gram (San-Segundo et al, 2002).

La secuencia de letras reconocida puede empezar o no con un modelo de ruido. Para representar esta posibilidad hemos introducido una línea ("nada") en el nodo inicial. De la misma manera la secuencia puede terminar o no con un modelo de ruido: también hemos representado esta posibilidad mediante una línea ("nada") en el nodo final.

Un aspecto que conviene comentar es que la representación de la figura 3-9 no es del todo completa y debe quedar claro al lector que desde cualquier nodo de la estructura se puede transitar al nodo final, pudiendo haber secuencias que sólo contengan ruidos y no contengan ninguna letra. Por otro lado, en la figura 3-9 no hemos representado los modelos de silencio porque están incluidos en los modelos de letra correspondientes (apartado 3.6.2). Otro detalle importante es que en la representación de los nodos hemos dibujado un único modelo acústico, pero en el caso de letras con varios nombres se debe considerar un modelo acústico diferente para cada nombre asociado a dicha letra.

Los resultados correspondientes a la introducción de los modelos de lenguaje se pueden consultar en la tabla 3-7. Según estos resultados podemos deducir que el modelo 3-gram es más potente que el 2-gram permitiendo tasas de error menores. La reducción del error al nivel de letra es del 40,9% relativo, y el aumento del porcentaje de cadenas perfectas del 70,1% relativo para el modelo 3-gram.

	Algoritmo One-pass		**Etapa de hipótesis**	
Sistema	**TEL (%)**	**PCP (%)**	**TE (%)**	**TP (xRT)**
Sistema de Referencia	30,4	21,3	16,6	0,9
Silencios Contextuales	23,8	27,8	10,9	1,2
Silencios Contextuales + Modelos de ruidos	19,3	34,3	8,0	1,2
SC + MR + 2-gram	18,6	35,4	7,7	1,3
SC + MR + 3-gram	11,0	60,4	6,8	3,8

Tabla 3-7: Resultados del Sistema de Referencia, con la nueva topología de HMM de silencios contextuales, con los modelos de ruido además de la nueva topología de HMM, y considerando los modelos de lenguaje 2-gram y 3-gram: Tasa de Error de Letra (TEL), Porcentaje de Cadena Perfectas (PCP), y la Tasa de Error (TE) y el Tiempo de Proceso (TP) de la etapa de hipótesis.

La reducción significativa de la Tasa de Error de Letra repercute en una reducción muy pequeña de la Tasa de Error al nivel de nombre (no significativa en este caso). Las razones que justifican este comportamiento son las siguientes:

- Por un lado con la introducción del modelo de lenguaje 3-gram hemos reducido considerablemente los errores al nivel de letra, más de un 40% (incrementando más de un 70% el número de cadenas perfectas). Debido a esto, disponemos de menor cantidad de datos para entrenar las penalizaciones de inserciones, borrados y sustituciones de letras que se aplican en el algoritmo de alineamiento utilizado para comparar la secuencia de letras con los nombres del diccionario. Esta reducción de datos de entrenamiento repercute en una peor estimación de las penalizaciones y en

un peor funcionamiento de esta etapa que impide conseguir una mayor reducción de error al nivel de nombre.

- Una segunda razón es que las penalizaciones utilizadas no consideran información contextual, es decir, no se dispone de penalizaciones diferentes dependiendo del contexto. Cuando se incorporan modelos de lenguaje en el reconocimiento, un error producido en una letra puede influir en las letras adyacentes (anteriores o posteriores), luego las penalizaciones independientes del contexto no modelan correctamente este tipo de errores. En la figura 3-10 se muestran dos casos en los que la introducción del modelo de lenguaje produce resultados de reconocimiento diferentes. Cuando no se utiliza el modelo de lenguaje, la letra B se sustituyó por las letras P y D respectivamente, pero la letra E se reconoció correctamente. Al considerar el modelo de lenguaje, el error cometido en la letra B repercute de diferente manera en la letra E posterior: en el primer caso se sustituye por la letra I y en el segundo se borra incluyendo su pronunciación en la de la letra N. En este caso sería necesario el entrenamiento de penalizaciones dependientes del contexto pero esta solución no ha sido posible al no disponer de datos de entrenamiento suficientes dada la buena calidad de las cadenas que se están generando (punto anterior).

	Sin LM	**Con LM**
Referencia	R U B E N	R U B E N
Caso 1	F U P E N	R U P I N
Caso 2	R U D E N	R U D N

Figura 3-10: Variedad de errores producidos en la incorporación del modelo de lenguaje (ML) (San-Segundo et al, 2002)

Por último comentar que la aplicación del modelo de lenguaje 3-gram y la duplicación de nodos requerida ha incrementado el tamaño del espacio de búsqueda dando lugar a un aumento muy significativo del tiempo de proceso. Además este incremento será mayor cuanto mayor sea el diccionario de nombres considerado: al tener más nombres, aparecen nuevas secuencias de letras que dan lugar a nuevos nodos en el espacio de búsqueda. Esta circunstancia hace inviable esta estructura para su funcionamiento en tiempo real.

3.6.5 Obtención de las N mejores cadenas de letras

En este apartado vamos a considerar la opción de generar varias secuencias de letras (en lugar de una única) con el fin de mejorar la tasa de reconocimiento. Para obtener estas N mejores secuencias de letras utilizaremos el algoritmo denominado Pseudo N-Best y descrito en la tesis de Colás (Colás, 1999). En este algoritmo debemos mantener en cada nodo N copias de los modelos acústicos de la letra con el fin de almacenar diferentes historias en el proceso de alineamiento. En la figura 3-11 podemos ver una parte del espacio de búsqueda correspondiente a la transición de cualquier letra a la letra Z.

Un aspecto importante a resaltar es que para considerar que dos historias son diferentes, y por tanto obliguen a mantener caminos paralelos, deben cumplir que las secuencias de letras que generarían esos caminos, desde el inicio de la secuencia hasta el punto de análisis, son diferentes. Para garantizar este hecho debemos realizar un análisis hacia atrás (backtracking parcial) que lo verifique. Este análisis debe ser realizado cada vez que se estudien las posibles transiciones entre letras lo que produce un aumento importante del tiempo de procesado como veremos en los resultados.

En este algoritmo Pseudo N-Best utilizado no se analizan las transiciones entre las N copias para todos los estados que forman los modelos, únicamente se consideran las N copias en la transiciones entre letras. Por esta razón se denomina Pseudo N-best en lugar de Full N-best.

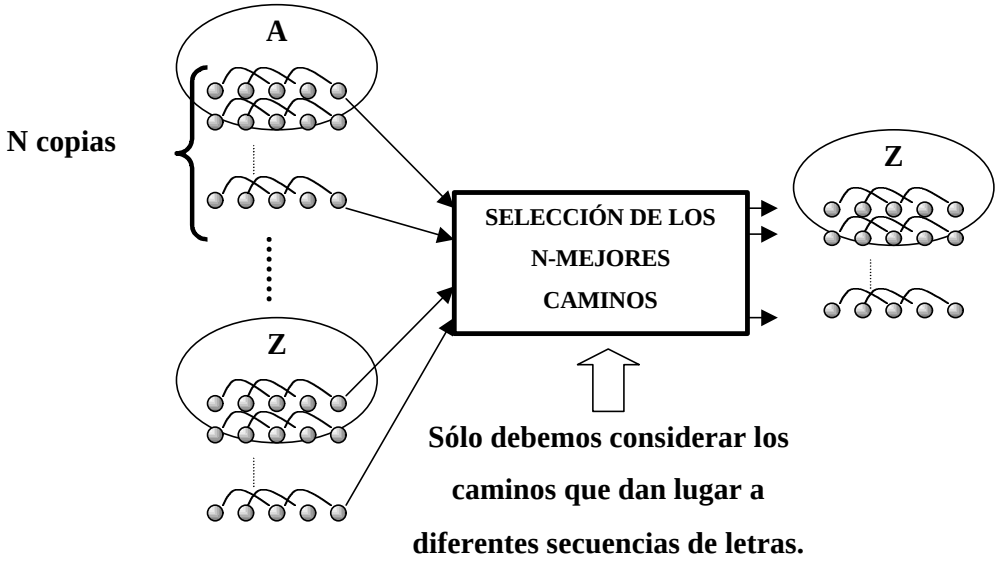


Figura 3-11: Análisis de transiciones entre letras en el algoritmo de generación de las N mejores secuencias de letras.

Para reducir el error al nivel de nombre, utilizando la N mejores cadenas de letras, vamos a seguir las dos ideas propuestas por Jouvet (Jouvet et al, 1993b):

- Considerando que las N mejores cadenas son resultados válidos del reconocedor y que por tanto sus errores son ejemplos válidos de errores en reconocimiento, la propuesta es utilizar las N cadenas de letras para entrenar las penalizaciones usadas en el alineamiento de las cadenas de letras con los nombres del diccionario.

- Por otro lado, suponiendo que las N secuencias son opciones válidas a la hora de elegir el nombre deletreado, calcularemos los mejores alineamientos de las N cadenas (y no sólo de la primera cadena) con los nombres del diccionario.

Los resultados obtenidos se presentan en la tabla 3-8. En estos resultados hemos considerado el modelo de lenguaje 2-gram incorporado en el algoritmo de One-pass.

La Tasa de Error de Letra (TEL) y el Porcentaje de Cadenas Perfectas (PCP) los hemos calculado considerando la secuencia de letras que ofreció un mejor alineamiento (menor coste) con cualquier nombre del directorio.

	Algoritmo One-pass		Etapa de hipótesis	
Valor de N	**TEL (%)**	**PCP (%)**	**TE (%)**	**TP (xRT)**
1	18,6	35,4	7,7	1,3
2	13,8	49,5	6,8	2,1
4	10,9	60,1	5,9	5,1
8	9,0	66,0	5,7	16,6

Tabla 3-8: Resultados obtenidos considerando las N mejores cadenas de letras y el modelo de lenguaje 2-gram: Tasa de Error de Letra (TEL), Porcentaje de Cadena Perfectas (PCP), y la Tasa de Error (TE) y el Tiempo de Proceso (TP) de la etapa de hipótesis.

A tenor de los resultados podemos concluir lo siguiente:

- Las tasas de error, tanto al nivel de letra como al nivel de nombre, decrecen considerablemente al aumentar el número de cadenas consideradas. Al pasar de 1 a 4 cadenas hemos reducido el error un 41,4% al nivel de letra y un 23,4% al nivel de nombre.

- Estas reducciones de las Tasas de Error tienden a saturarse a medida que el número de cadenas aumenta. Por un lado, los errores que aparecen en las cadenas con N elevados comienzan a ser errores forzados y no reflejan comportamientos del propio reconocedor con lo que las penalizaciones quedan peor entrenadas, y por otro lado, estas cadenas ya no se corresponden con alternativas válidas de reconocimiento de la secuencia de letras deletreada puesto que la diferencia acústica comienza a ser importante.

- Al aumentar el número de cadenas utilizadas, el tiempo de procesado se incrementa de forma importante. Este hecho se produce por dos factores. En primer lugar al tener que duplicar N veces los modelos acústicos de cada letra para almacenar historias diferentes, el espacio de búsqueda se incrementa considerablemente y el tiempo de procesado aumenta. Por otro lado, en cada transición entre palabras, al seleccionar los N mejores modelos de letras predecesores, se debe comprobar que los N modelos seleccionados dan lugar a cadenas de letras diferentes, lo que ralentiza aún más la velocidad de reconocimiento. Esta circunstancia hace poco útil la aplicación de este algoritmo para un sistema en tiempo real.

3.6.6 Consideración de un grafo de letras

En este apartado se describe la implementación de un grafo de letras en la etapa de obtención de la cadena de letras. La generación de un grafo tiene como finalidad aplicar modelos de lenguaje tipo 3-gram y obtener las N mejores secuencias de letras sin que el

tiempo de procesado aumente tanto como lo visto en las soluciones propuestas hasta ahora. En el capítulo 4 (apartados 4.4 y 4.5) se propone una simplificación del algoritmo descrito por Hermann Ney (Ney, 1994; Ney y Ortmanns, 1999) para la generación de un grafo de palabras. También se estudian y discuten diversos problemas como la gestión de los modelos acústicos para los ruidos, la incorporación de modelos de lenguaje 3-gram o la generación de las N mejores cadenas de letras. En este apartado analizaremos los resultados al considerar la mejor de las soluciones evaluadas en el capítulo 4 sobre el reconocedor de nombres deletreados.

La principal idea de generar un grafo en lugar de una única secuencia de palabras es representar y extraer posibles alternativas de descodificación en zonas de la voz donde la ambigüedad acústica sea muy elevada. Para la generación del grafo es necesario ir dejando constancia de las posibles secuencias hipótesis cuyas verosimilitudes acústicas sean muy parecidas a la mejor de ellas en esa zona de voz (óptimo local) y que no sobrevivirían en el proceso de recombinación realizado en el algoritmo One-pass. La idea básica es representar estas secuencias de letras o palabras mediante un grafo en el que cada nodo represente una letra. Cada secuencia de letras contenida en el grafo debería ser próxima (en términos de verosimilitud) a la mejor secuencia producida por el One-pass.

Para la generación del grafo de letras se deben considerar los siguientes pasos:

- En primer lugar en el proceso de análisis hacia delante, en cada trama y para cada letra L_i consideramos todas las posibles letras predecesoras (L_j) y seleccionamos las transiciones más probables (L_i, L_j). H. Ney propone el uso de la técnica de Beam Search para limitar el número de predecesores. En nuestro caso no utilizamos esta técnica puesto que el vocabulario de reconocimiento es pequeño (35 letras y 4 modelos de ruido) y de gran confusión acústica. Debido a esto, consideraremos un parámetro que denominaremos COMPLEJIDAD_GRAFO (ver capítulo 4) para limitar el número de predecesores de una letra determinada. Este parámetro nos dará el número de predecesores máximo considerado.

- Al final de la señal de voz, se construye un grafo de letras recorriendo la información obtenida en el proceso hacia delante del One-pass. En este proceso se debe haber anotado las tramas límite de cada letra, sus posibles letras predecesoras y la verosimilitud acumulada a lo largo de dicha letra.

  Un nodo del grafo queda caracterizado por tres parámetros: la letra asociada, la trama de inicio y la trama de final. En el proceso de generación del grafo cuando se encuentran dos nodos de la misma letra con la misma trama inicial y trama final estos nodos se unen manteniendo la mayor de las verosimilitudes consideradas. En algunas situaciones es posible relajar estas condiciones y permitir cierta variación en los límites de las letras.

En este punto existen algunos comentarios que se deben remarcar sobre el algoritmo de generación del grafo de letras:

- En primer lugar, siguiendo los comentarios de H. Ney sobre las palabras cortas como artículos y preposiciones, los modelos de ruidos se han unido a la letra predecesora para formar un mismo nodo del grafo. En la figura 3-12 podemos ver cómo el ruido [spk] se ha unido a la letra R para formar el nodo R.

- Los modelos de ruidos al comienzo del grafo (anteriores a las letras T ó P en nuestro caso) se deben unir a los primeros nodos (T ó P).

- Los modelos de silencio no tienen un tratamiento especial porque se han considerado como parte del modelo de letra (ver apartado 3.6.2)

En la figura 3-12 podemos ver un ejemplo de grafo generado para el nombre TORRA con un valor de COMPLEJIDAD_GRAFO de 2.

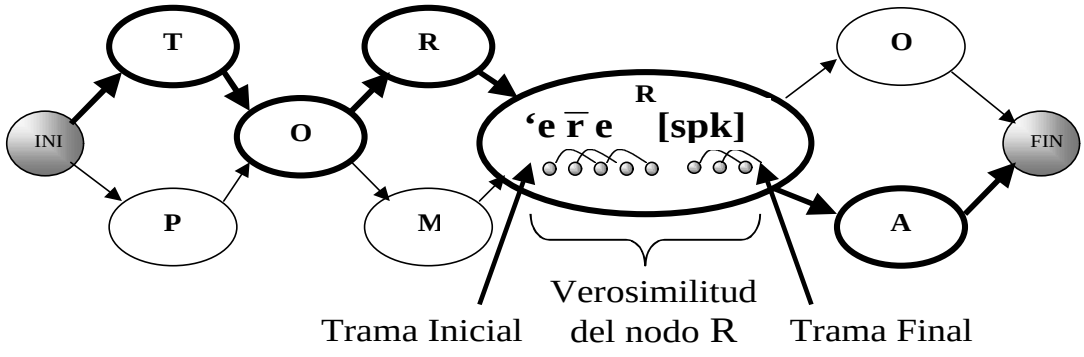


Figura 3-12: Grafo de letras para el nombre TORRA con COMPLEJIDAD_GRAFO=2.

La utilización de un grafo de letras permite aplicar las técnicas anteriores, generación de N secuencias y aplicación del modelo de lenguaje 3-gram, con bajo coste computacional. Las razones son las siguientes:

- Con la generación de un grafo de letras hemos conseguido reducir el espacio de búsqueda considerando únicamente aquellas secuencias de letras que presentan gran parecido acústico con la locución pronunciada por el usuario. De esta forma, el impacto de tener que aumentar el tamaño del espacio de búsqueda para aplicar estas técnicas es mucho menor.

- Por otro lado, al disponer de la verosimilitud acústica acumulada a lo largo de cada nodo del grafo, no necesitamos volverla a calcular y puede ser utilizada directamente como heurístico para recorrer el grafo (ver capítulo 4, apartado 4.4.2 procesado del grafo).

3.6.6.1 Obtención de las N mejores cadenas de letras

En primer lugar veamos los resultados obtenidos en la generación de las N mejores cadenas de letras. En la tabla 3-9 se presentan los resultados para este caso, con los mismos criterios considerados en el apartado 3.6.5: se utilizan las N cadenas para entrenar la penalizaciones del algoritmo de comparación de cadenas con nombres, y

además, se utilizan las N cadenas para obtener el nombre del diccionario que mejor alineamiento ofrece. En el capítulo 4, apartado 4.6 se puede consultar el pseudocódigo del algoritmo para obtener las N mejores cadenas a partir de un grafo de letras.

	Grafo de letras (2-gram)		Etapa de hipótesis	
Valor de N	**TEL (%)**	**PCP (%)**	**TE (%)**	**TP (xRT)**
1	18,6	35,4	7,7	1,3
2	14,6	47,3	6,9	1,5
4	11,1	59,7	5,9	2,2
8	9,5	64,2	5,7	2,3

Tabla 3-9: Resultados obtenidos considerando las N mejores cadenas de letras obtenidas del grafo y considerando el ML 2-gram: Tasa de Error de Letra (TEL), Porcentaje de Cadena Perfectas (PCP), y la Tasa de Error (TE) y el Tiempo de Proceso (TP) de la etapa de hipótesis.

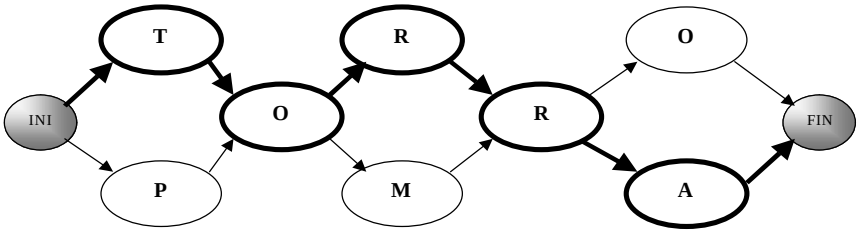
Como se puede ver, aunque en este caso las tasas de error sean ligeramente superiores a las vistas en la tabla 3-8, las diferencias no son significativas. En cuanto al tiempo de proceso la reducción ha sido muy importante. Podemos decir que para N = 4 y N =8 conseguimos las mismas Tasas de Error al nivel de nombre en comparación con los resultados del apartado 3.6.5, con reducciones del 56,9% y del 86,1% en el tiempo de procesado respectivamente. Al igual que en el caso anterior, a medida que aumentamos el valor de N las tasas de error tienden a saturarse.

En nuestros experimentos hemos considerado un valor del parámetro COMPLEJIDAD_GRAFO de dos, para los casos de N=1 y N=2, y de tres para N=4 y N=8. El objetivo de aumentar este valor en el segundo caso ha sido garantizar la obtención de un grafo con suficientes alternativas para poder obtener N cadenas de letras diferentes. Este aumento en la complejidad produce un aumento del tiempo de procesado importante, como que se puede ver en la tabla 3-9 al pasar de N=2 a N=4. En estos experimentos, hemos obtenido un número medio de nodos de 16,3 y 19,2 para los casos de COMPLEJIDAD_GRAFO igual a 2 y 3 respectivamente. De estos valores también podemos deducir que, suponiendo una longitud media de 7,6 letras por nombre, el tamaño de los grafos es muy reducido, ya que produce una ambigüedad de 2,1 y 2,5 nodos por letra reconocida. Estos tamaños tan reducidos del grafo reflejan que el criterio de unión de nodos (misma etiqueta, trama inicial y final) no resulta tan exigente y permite unir gran cantidad de nodos redundantes. En este caso, al disponer de modelos de palabra de gran longitud (número de estados elevado), los límites de las letras obtenidas en reconocimiento quedan mejor definidos que el caso de un reconocedor de habla continua más general como veremos para el caso del sistema desarrollado en el capítulo 4.

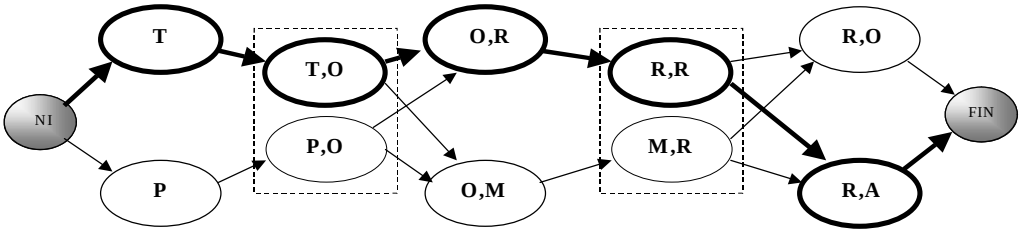
3.6.6.2 Incorporación del modelo de leguaje 3-gram

Una vez visto la mejora en tiempo obtenida para el caso de las N cadenas de letras, debemos analizar si la incorporación del modelo de lenguaje 3-gram en el grafo, en

lugar de en el espacio de búsqueda total, podría obtener reducciones tan importantes. La probabilidad ofrecida por un modelo 3-gram depende de dos letras anteriores a la letra bajo análisis. Dado que en un grafo pueden existir nodos con más de un predecesor, la probabilidad gramatical del nodo no queda definida unívocamente. Este problema se resuelve duplicando nodos para cada predecesor distinto que tenga, es decir, creando estados gramaticales diferentes (ver apartado 4.5.2.2). Veamos el proceso con un ejemplo, figura 3-13.



Grafo de letras original



Grafo modificado para el modelo 3-gram

Figura 3-13: Incorporación en el grafo del modelo de lenguaje 3-gram.

La modificación del grafo se realiza de la siguiente manera:

1. Si un nodo (por ejemplo O) tiene *n* predecesores diferentes (T y P) en el nodo original, este nodo se duplica N veces: (T, O) y (P, O). La primera componente identifica la letra predecesora. En la figura 3-13 se han recuadrado los nodos duplicados con líneas discontinuas.

2. Si desde el nodo O había una transición a los nodos R y M en el grafo original, en el nuevo grafo debe haber una transición desde cada copia de O, (O, R) y (O, M), a los mismo nodos.

3. La verosimilitud acústica acumulada en todas las copias es la misma que la que tenía originalmente el nodo.

En la tabla 3-10 se presentan los resultados al aplicar el modelo de lenguaje 3-gram y la generación de N cadenas de letras.

Si comparamos los resultados para N=1 con los obtenidos en la tabla 3-7 (3-gram) podemos ver como la Tasa de Error al nivel de letra es mayor ahora, mientras que al nivel de nombre es muy similar. Podremos decir que consiguiendo tasas al nivel de nombre similares (sin diferencia estadísticamente significativa) reducimos el tiempo de

proceso un 47,4%. A medida que N aumenta las tasas van mejorando sin aumentar en exceso el tiempo de procesado. Considerando N=8 podemos decir que habiendo reducido un 21,0% el tiempo de proceso hemos reducido las Tasas de Error un 29,0% al nivel de letra y un 25,0% al nivel de nombre. Al igual que en los casos anteriores a medida que N aumenta las mejoras se van saturando.

	Grafo de letras (3-gram)		**Etapa de hipótesis**	
Valor de N	**TEL (%)**	**PCP (%)**	**TE (%)**	**TP (xRT)**
1	14,6	47,2	6,6	2,0
2	11,3	58,4	5,8	2,1
4	9,2	65,3	5,4	3,0
8	7,8	68,5	5,1	3,0

Tabla 3-10: Resultados obtenidos considerando las N mejores cadenas de letras obtenidas del grafo y considerando el ML 3-gram: Tasa de Error de Letra (TEL), Porcentaje de Cadena Perfectas (PCP), y la Tasa de Error (TE) y el Tiempo de Proceso (TP) de la etapa de hipótesis.

Otro aspecto que conviene resaltar es que la incorporación del modelo 3-gram ha incrementado el número medio de nodos en el grafo: 25,1 y 36,1 para COMPLEJIDAD_GRAFO de 2 y 3 respectivamente. Este hecho junto con la necesidad de modificar el grafo, ha producido un aumento del tiempo de procesado respecto los resultados de la tabla 3-9.

Como conclusión de este apartado podemos comentar que la generación de un grafo de letras permite introducir el modelo de lenguaje 3-gram y obtener las N mejores cadenas de letras con bajo coste computacional.

A tenor de los resultados presentados, y considerando un compromiso entre tasa de reconocimiento y tiempo de proceso, hemos considerado como etapa de hipótesis para nuestro sistema la consideración del grafo de letras para calcular las 2 mejores cadenas aplicando un modelo de lenguaje 3-gram.

3.6.7 Análisis de los conjuntos de letras con mayor confusión.

Considerando la configuración final obtenida para la etapa de hipótesis, vamos a analizar los resultados para los conjuntos de mayor confusión descritos en el apartado 3.2. En la tabla 3-11, se presentan los porcentajes de sustituciones (Sus), borrados (Bor) e inserciones (Ins) y la Tasa de Error al nivel de letra para estos subconjuntos de letras. Estos porcentajes se han calculado con las siguientes ecuaciones:

$$Sus(\%) = \frac{N_S}{N_T} \times 100 \qquad (3-2)$$

$$Bor(\%) = \frac{N_B}{N_T} \times 100 \qquad (3-3)$$

$$Ins(\%) = \frac{N_I}{N_T} \times 100 \qquad (3-4)$$

Donde:

- N_S: número de letras, que perteneciendo al subconjunto considerado, son sustituidas por otra letra que puede pertenecer o no a ese mismo subconjunto.
- N_B: número de letras pertenecientes al subconjunto analizado que se borran respecto de la secuencia de letras referencia.
- N_I: número de letras del subconjunto bajo análisis que se insertan en la secuencia de letras reconocida.
- N_T: número total de letras en la secuencia de letras referencia que pertenecen al subconjunto de letras bajo análisis.

	Sus(%)	**Bor(%)**	**Ins(%)**	**TEL (%)**
Total	8,5	1,6	1,2	11,3
E-set	15,5	0,6	1,9	18,0
[B, D, G]	14,1	0,2	1,1	15,4
[P, T, E]	16,5	1,1	2,9	20,5
ExE-set	11,7	1,9	2,1	15,7
[F, S]	7,7	2,9	2,4	13,0
[L, LL]	12,5	1,9	3,1	17,5
[M, N, Ñ]	12,1	0,6	2,3	15,0
[K, A]	1,4	1,6	0,5	3,5
[Q, U]	9,8	1,3	1,4	12,5

Tabla 3-11: Resultados de los errores cometidos en los subconjuntos de letras con mayor confusión acústica: porcentajes de Sustituciones (Sus%), Borrados (Bor%) e Inserciones (Ins%) y Tasa de Error de Letra (TEL).

Como podemos ver el mayor porcentaje de sustituciones se ha obtenido en el caso del subconjunto E-set para las letras P, T y E. En este caso, como ya comentamos en el apartado 3.2 la oclusión del comienzo de la letra tiene una duración muy pequeña que hace que esas letras se confundan de forma importante y además se inserten con bastante facilidad. En el caso del conjunto ExE, el mayor error se ha cometido en el subconjunto [L, LL].

Por último, cabe comentar que las mejores tasas (menores errores) se obtuvieron para los grupos [K, A] y [Q, U]. En estos casos, aunque la confusión acústica es muy

importante, la primera es una consonante y la segunda una vocal, luego el modelo de lenguaje al nivel de letra permite discriminarlas fácilmente.

3.7 Etapa de Verificación

Como comentamos en el apartado 3.4, en la etapa de verificación construiremos una gramática muy restrictiva en forma de árbol con los M mejores nombres del diccionario, seleccionados según su mejor alineamiento con las cadenas de letras obtenidas. Una vez definida dicha gramática se vuelve a ejecutar el algoritmo One-pass obteniendo así el nombre finalmente reconocido. El problema en este caso es definir el número M de nombres a considerar en la gramática.

Cuando se trabaja con sistemas basados en una arquitectura de hipótesis y verificación, es necesario realizar un estudio cuidadoso del número de candidatos M a considerar. Veamos el análisis que proponemos. La Tasa de Acierto al nivel de nombre (TAN) según el número de candidatos M, la podemos descomponer en dos funciones diferentes según la ecuación 3-5.

$$TAN\ (M) = P_H(M) \times P_V(M) \qquad (3-5)$$

Donde:

- P_H (M): es la probabilidad de obtener el nombre correcto entre los M candidatos propuestos por la etapa de hipótesis.

- P_V (M): es la probabilidad de acierto en la etapa de verificación condicionada a que el nombre esté entre los propuestos por la etapa de hipótesis. Es decir, para calcular esta probabilidad únicamente se considerarán los casos en los que la etapa de hipótesis incluyó el nombre correcto entre los M mejores propuestos.

La Tasa de Error al nivel de nombre será igual a 1 menos la Tasa de Acierto.

En la figura 3-14, podemos ver la evolución de las funciones P_H (M) y P_V (M) para el caso de un diccionario de 1.000 nombres.

Como podemos observar P_H (M) y P_V (M) tienen comportamientos opuestos. Al ir aumentando el valor de M la probabilidad de que el nombre correcto se incluya entre los M candidatos propuestos por la fase de hipótesis (P_H) incrementa, mientras que la etapa de verificación presenta una tasa de reconocimiento condicionada menor (P_V) dado que el número de candidatos considerado va aumentando. P_V (M) decrece muy rápidamente al comienzo, con valores de M pequeños, y tiende a estabilizarse a medida que M aumenta. La razón de este comportamiento es la siguiente: en la etapa de hipótesis se obtienen los nombres más parecidos a la secuencia de letras obtenida, con lo que la confusión entre dichos nombres es muy grande. Al añadir un nuevo nombre con M pequeño estamos incorporando mucha confusión en la gramática de la etapa de verificación, y por tanto, P_V (M) decrece rápidamente. Por otro lado al considerar más nombres con un M elevado, estos nombres son bastante diferentes de la cadena

deletreada con lo que la confusión añadida es menor produciendo una estabilización de P_V (M).

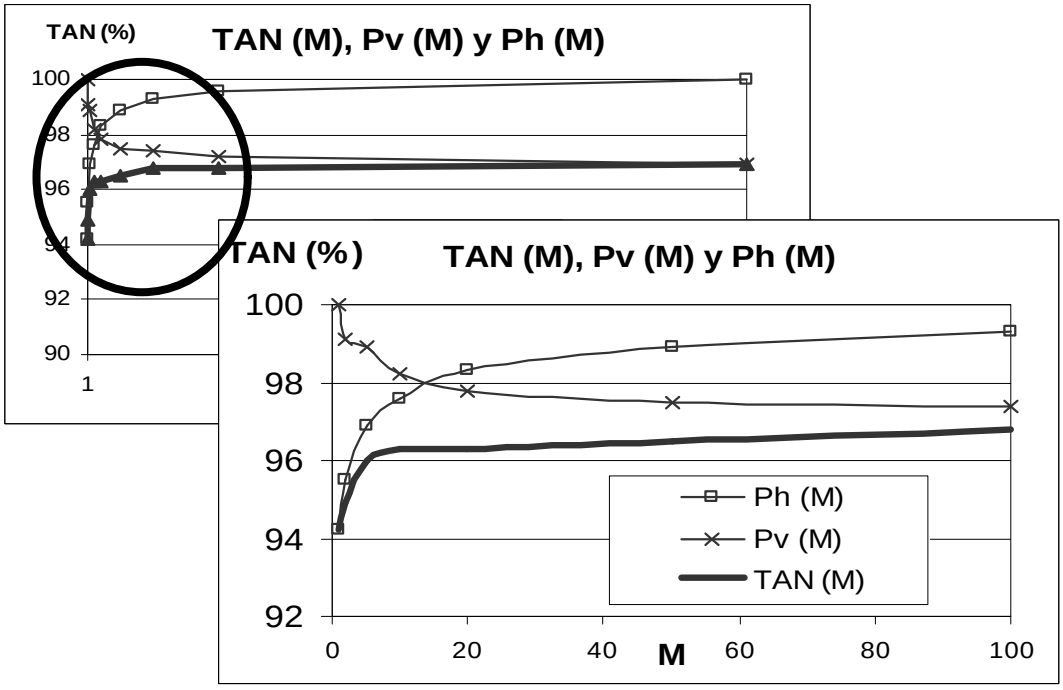


Figura 3-14: Evolución de la Tasa de Acierto (TAN) y de las probabilidades P_H (M) y P_V (M) según M (diccionario de 1.000 nombres) (San-Segundo et al, 2002).

A la hora de elegir el valor de M debemos establecer un compromiso entre la Tasa de Aciertos al nivel de Nombre (TAN) y el tiempo de procesado necesario en cada caso. En la figura 3-15, representamos la evolución de TAN y del tiempo de procesado (TP) con M.

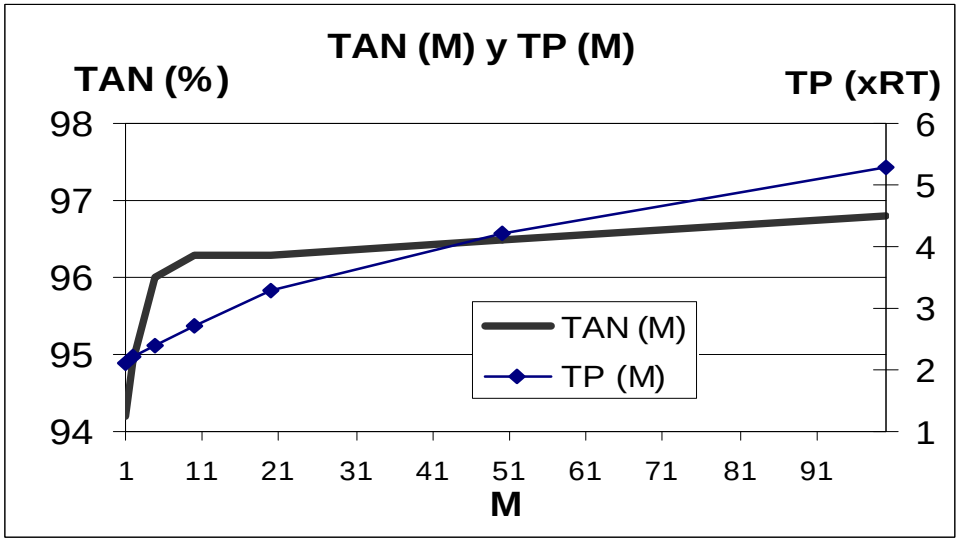


Figura 3-15: Evolución de la Tasa de Acierto (TAN) y el Tiempo de Procesado según M (diccionario de 1.000 nombres).

Generalmente al tener una TAN monótona creciente la elección de M resulta sencilla: cogeremos un valor de M cuanto mayor mejor. En este caso el límite lo debe imponer el tiempo de procesado de forma que no se supere 1 xRT: habría que elegir un M grande para conseguir mayor tasas de reconocimiento pero garantizando que el sistema funcione en tiempo real (tiempo de proceso menor a 1 xRT). En el caso de la figura 3-15, para cualquier valor de M obtenemos un tiempo de procesado superior a 1 xRT y va aumentando casi de forma lineal, con lo que debemos elegir un número M lo más reducido posible que nos garantice una buena TAN. En este caso el número elegido rondaría el valor de M=10. Para este valor de M el tiempo de proceso es 2,2 xRT con lo que habrá que utilizar un ordenador más potente al empleado en estos experimentos (Pentium II 350 Mhz) para hacerlo funcionar en tiempo real.

Pero no siempre la TAN es monótona creciente. Dependiendo de los sistemas que formen cada una de las etapas podemos tener comportamientos diferentes como los que se presentan en la figura 3-16. En el primer caso se puede observar cómo la caída en tasa de $P_V(M)$ es más rápida que la subida de $P_H(M)$ lo que produce un mínimo de TAN. En el segundo caso ocurre lo contrario: la subida de $P_H(M)$ es mayor que la bajada de $P_V(M)$ pero a medida que M aumenta $P_V(M)$ cae por debajo incluso de los primeros valores de $P_H(M)$ con lo que se genera un máximo de TAN alrededor de un M bajo.

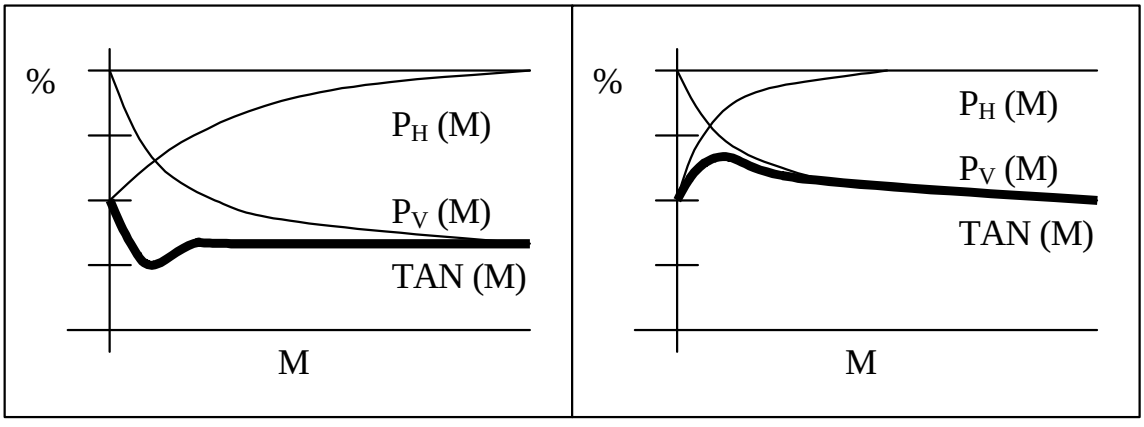


Figura 3-16: Otros ejemplos de evolución de la Tasa de Acierto (TAN) según M.

A la hora de hacer un estudio para mejorar los sistemas basados en una arquitectura de hipótesis y verificación es importante analizar los comportamientos de $P_H(M)$ y $P_V(M)$ para entender el funcionamiento del sistema. De poco sirve optimizar mucho una de las dos etapas si no se hace de acuerdo a su funcionamiento en común. En el primer ejemplo de la figura 3-16, de nada sirve mejorar la etapa de hipótesis si la limitación de TAN nos viene fijada por la etapa de verificación. En este caso es incluso mejor no considerar la etapa de verificación y aceptar como bueno el resultado ofrecido por la etapa de hipótesis. El segundo caso podría ser el resultado de plantear una etapa de verificación que pretenda resolver los problemas de confusión entre los candidatos ofrecidos por la etapa de hipótesis (por ejemplo: entrenamiento discriminativo) consiguiendo que la caída de $P_V(M)$ sea más lenta para M bajos aun a costa de obtener caídas mayores para M grandes. Este comportamiento podría generar un máximo de TAN para un valor de M entorno a 10 que mejorase la TAN conseguida con el sistema

actual desarrollado para ese M. Este estudio se planteará como una línea futura de la presente tesis.

En este punto conviene comentar que, para la gestión de los modelos de ruido, dado que las estructuras en árbol generadas son muy reducidas, hemos considerado modelos acústicos en todas las transiciones entre nodos de letras. Esta solución es la que mayor tiempo de proceso necesita pero a su vez es la que mejores resultados ofrece.

Para los diccionarios con 5.000 y 10.000 nombres, la representación de $P_H(M)$ y $P_V(M)$ es similar a la mostrada para 1.000 nombres en la figura 3-14. Hemos calculado el valor de M como un compromiso entre TAN y tiempo de proceso utilizando el segundo de los conjuntos de validación de datos. Los resultados finales se presentan en la tabla 3-12. En esta tabla se presentan las Tasas de Acierto al nivel de nombre para la etapa de hipótesis (seleccionando el nombre con el mejor alineamiento con las cadenas de letras obtenidas) y para la etapa de verificación que es la que define el comportamiento global del sistema.

Tamaño del diccionario	TAN (Hipótesis)	TAN (Verificación)	M	TP (xRT)
1.000 (0,2)	94,2%	96,3%	10	2,8
5.000 (0,5)	88,7%	92,8%	20	3,4
10.000 (0,9)	86,2%	90,3%	50	4,7

Tabla 3-12: Resultados para los diccionarios de 1.000, 5.000 y 10.000 nombres: Tasa de Acierto de nombre de las etapas de hipótesis y verificación, M y Tiempo de Proceso (TP).

Para poder comparar estos resultados con los obtenidos en el sistema SmartSpell (Junqua, 1997) para la misma tarea en inglés presentamos en la tabla 3-13 los resultados con M=20.

Tamaño del diccionario	TAN (Verificación)	M	TP (xRT)
1.000 (0,2)	96,4%	20	3,2
5.000 (0,5)	92,8%	20	3,4
10.000 (0,9)	90,0%	20	3,5

Tabla 3-13: Resultados para los diccionarios de 1.000, 5.000 y 10.000 nombres considerando M=20: Tasa de Acierto de nombre, M y Tiempo de Proceso (TP) (San-Segundo et al, 2002).

Tamaño del diccionario	TAN (Verificación)	M
491 (0,07)	98,4%	20
3.388 (0,5)	95,3%	20
21.877 (1,8)	90,4%	20

Tabla 3-14: Resultados obtenidos en el sistema SmartSpell (Junqua, 1997) considerando M=20: Tasa de Acierto de nombre y M.

Junqua consideró una base de datos con 4,000 llamadas en las que los locutores deletrearon nombres de pila y apellidos, con y sin pausas entre letras. Más de 1.200 llamadas fueron seleccionadas para entrenamiento, 558 para validación y 491 para test. Todas las llamadas seleccionadas fueron apellidos pronunciados sin pausas entre letras. Además ninguna de las llamadas contenía voces de fondo, ruido de línea o ruidos como clicks de labios o respiraciones fuertes.

Como podemos ver nuestros resultados son ligeramente peores que los presentados para el caso del inglés (Junqua, 1997) pero las tareas no se pueden comparar directamente puesto que en nuestro caso hemos considerado ficheros con todo tipo de ruidos. Como comentamos en el apartado 3.2, en castellano hay una relación directa entre la pronunciación y la escritura de una palabra con lo que no es necesario deletrear para desambiguar. Este hecho da lugar a que los locutores no estén habituados y produzcan falsos comienzos, dudas, pausas rellenas, errores y cambios de velocidad de locución importantes. Hemos evaluado las diferencias en la velocidad de locución en el conjunto de test y hemos obtenido una media de 1,1 l/s (letras por segundo) con una Desviación Típica de 0,24 l/s. La velocidad mínima observada fue de 0,4 l/s y la máxima 2,1 l/s. La confusión media para cada uno de los diccionarios se presenta entre paréntesis.

Hemos desarrollado una versión en tiempo real del sistema sobre un ordenador Pentium III 600 Mhz con 256 Mb de memoria RAM y conectada directamente a la red telefónica. En el siguiente apartado se analizarán los experimentos realizados con este sistema.

3.8 Evaluación de campo

El reconocedor de nombres deletreados ha sido incorporado en un sistema de información telefónica de páginas blancas en castellano con reconocimiento de voz de gran vocabulario sobre línea telefónica. Este sistema se ha desarrollado dentro del proyecto IDAS (Interactive Directory Assistance Service), LE4-8315 (Lehtinen et al, 2000; Córdoba et al, 2001). Algunas de sus características son las siguientes:

- Base de datos representativa con más de un millón de registros.
- 4 vocabularios diferentes: ciudades, nombres propios, apellidos y nombres de empresas. Los vocabularios de ciudades, nombres propios y empresas son de 1.000 palabras mientras que el vocabulario de apellidos es de 10.000.
- El sistema ofrece tanto número de teléfonos de particulares como de empresas.
- En el caso de que el módulo de reconocimiento falle, el sistema pasa directamente la llamada a un operador humano.

En el caso de un número de empresa, el sistema pregunta al usuario la ciudad y la compañía, mientras que para el teléfono de una persona particular pregunta la ciudad, el primer apellido y el nombre de pila. Hemos considerado diccionarios de 1.000 nombres

para las ciudades y empresas, y de 10.000 para los apellidos. Para los nombres de pila hemos considerado un diccionario de 1.000 palabras, pero cuando se reconocen correctamente la ciudad y el primer apellido, el diccionario de nombres propios se queda reducido a menos de 50 nombres. El sistema presenta al usuario tanto el primer como el segundo candidato reconocido. Si ninguno de los dos es confirmado por el usuario, el sistema le pide que deletree el nombre, utilizando el reconocedor de nombres deletreados desarrollado, como última alternativa antes de la intervención del operador humano. En este último caso, únicamente se presenta al usuario el primer candidato para su confirmación. En este servicio no consideramos ningún mecanismo automático para la detección de nombres fuera del diccionario de reconocimiento. Si alguno de los nombres no se ha reconocido correctamente, los datos deben ser completados por un operador quien escribe el nombre correcto y hace progresar la llamada. El usuario no habla directamente con el operador, de esta manera la duración de la llamada es más corta y el operador puede gestionar varias llamadas al mismo tiempo.

Para hacer frente a la larga duración de las pausas entre letras ha sido necesario modificar el módulo de detección de voz relajando las condiciones para la detección de fin de voz. Esta solución incrementa ligeramente el tiempo de respuesta pero garantiza menos del 5% de locuciones cortadas.

Durante un período de dos meses, se recogieron un total de 600 llamadas realizadas por 30 estudiantes de la Universidad. El 50% de las llamadas solicitaron un teléfono de una empresa y el otro 50% solicitó un número particular. Todas las llamadas fueron grabadas y analizadas posteriormente. Los resultados de esta evaluación, tanto para el reconocedor de nombres como para el reconocedor de nombre deletreados, se presentan en la tabla 3-15.

Tamaño del diccionario	**Reconocedor de nombres**		**Reconocedor de nombres deletreados**	**Global**
	1er Cand.	**2do Cand.**		
1.000 (0,3)	62,2%	8,1%	15,7% (52,7%)	86,0%
10.000 (1,1)	32,7%	7,5%	21,7% (36,9%)	61,9%

Tabla 3-15: Resultados de reconocimiento de la evaluación de campo.

Los resultados presentados corresponden con el porcentaje de veces que el sistema obtiene el nombre correcto como primer o segundo candidato del reconocedor de nombres, y como primer candidato del reconocedor de nombres deletreados. En este último caso, se presenta entre paréntesis la tasa de reconocimiento evaluada únicamente con los casos en los que falló el reconocedor de nombres. Para el caso del diccionario de 1.000 nombres, se presentan los resultados medios obtenidos con los diferentes vocabularios (ciudades y compañías) y para el diccionario de 10.000 nombres consideraremos el de apellidos. Como se puede ver, hay una degradación importante de la tasa de reconocimiento debido a que por un lado la confusión de los diccionarios es mayor (representada entre paréntesis) y sobre todo porque el reconocedor de nombres deletreados está funcionando en condiciones adversas, es decir, este reconocedor sólo se utiliza cuando el de nombres falla. Esto significa que el fallo se puede producir por

alguna de las siguientes circunstancias: podría haber un ruido importante de fondo, que el locutor no está acostumbrado a interactuar con sistemas automáticos o que el nombre pronunciado tenga mucha similitud con otros nombres del diccionario.

En esta evaluación, el 39,4% de las llamadas fueron atendidas automáticamente sin necesidad de utilizar el reconocedor de nombres deletreados, en el 19,3% fue necesario la utilización del reconocedor de nombres deletreados para completar la llamada automáticamente y en el resto de llamadas (41,3%) fue necesaria la intervención de un operador humano[2]. Por esta razón podemos concluir que aunque existe una importante degradación de los resultados en comparación con los experimentos de laboratorio, el reconocedor de nombres deletreados nos permite aumentar de 39,4% a 58,9% (39,4% + 19,3%) el porcentaje de llamadas completadas de forma automática.

3.9 Conclusiones

En este capítulo se presenta el desarrollo de un sistema de reconocimiento de nombres deletreados de forma continua por teléfono y se realiza un análisis detallado de la tarea de deletreo en castellano. Para la implementación del sistema se propone una estructura en dos etapas: hipótesis y verificación.

Para la etapa de hipótesis se analizan varias alternativas. En primer lugar se propone la utilización de una nueva topología HMM con modelos de silencio contextuales a los modelos de letra consiguiendo una mejora absoluta de 6,6 puntos en la Tasa de Error de letra en comparación con la utilización de un único modelo de silencio. Por otro lado, se incluyen modelos de ruidos en el espacio de búsqueda reduciendo el error de letra otros 4,5 puntos (de 23,8% a 19,3%) obteniendo una Tasa de Error de 8,0% al nivel de nombre en esta etapa de hipótesis. El modelado de estos ruidos es muy importante sobre todo para el caso del castellano donde no hay hábito a deletrear.

En esta etapa también se incorporan modelos de lenguaje N-gram (2-gram y 3-gram) en el proceso de decodificación. Estas gramáticas se generaron a partir del diccionario de nombres considerado. En este punto se describe una manera eficiente de incorporar los modelos de ruido en el espacio de búsqueda para el modelo 3-gram, obteniendo un error de letra de 11,0% y un 6,8% al nivel de nombre.

Por otro lado validamos para esta tarea la consideración de un grafo de letras (basado en el grafo de palabras propuesto por Ney) como una manera eficiente de incorporar modelos de lenguaje N-gram en el proceso de decodificación y de calcular las N mejores secuencias de letras. Como configuración final de esta etapa de hipótesis se considerará la incorporación del modelo 3-gram y la generación de las 2 mejores

[2] Notar que una llamada se completa automáticamente cuando todos los datos han sido reconocidos de forma correcta. Cada llamada consiste en tres datos (teléfonos privados) o dos datos (teléfonos de empresa). El porcentaje de llamadas atendidas automáticamente se podría calcular multiplicando las tasas de reconocimiento de cada dato. En la tabla 3-15, sólo tenemos los resultados medios para los diccionarios con el mismo tamaño pero no los resultados detallados por diccionario. Por esta razón, los porcentajes de llamadas atendidas automáticamente se obtienen tras el análisis de las llamadas y no multiplicando las tasas de reconocimiento.

cadenas de letras. En este caso se obtuvo un error de letra de 11,9% y un error de nombre de 5,8% considerando el diccionario de 1.000 nombres.

Para la etapa de verificación consideraremos una gramática muy restrictiva generada con los M mejores candidatos propuestos por la etapa de hipótesis. En esta etapa se describe el análisis realizado para calcular M (número de candidatos a considerar) con un buen compromiso entre tasa de reconocimiento y tiempo de procesado. Para la evaluación final del sistema completo utilizamos diccionarios de 1.000, 5.000 y 10.000 nombres, consiguiendo tasas del 96,3%, 92,8% y 90,3% respectivamente. Estos resultados son comparables a los obtenidos por sistemas similares en otros idiomas como en inglés (Junqua, 1997).

Finalmente, demostramos la utilidad del reconocedor desarrollado incorporándole en un sistema de información telefónica. En la evaluación de campo se comprobó que gracias al sistema de reconocimiento de nombres deletreados desarrollado en esta tesis se pudo aumentar la tasa de llamadas automáticamente atendidas de un 39,4% a un 58,7%.