Informe tecnico DSIC-DD/16/97, DSIC, UPV, 1997

Tema 4, Capítulo 3
Metodologías

Pablo Aibar José Miguel Benedí Francisco Casacuberta María José Castro
Andrés Marzal Federico Prat

1 Introducción

Los sistemas actuales de Reconocimiento Automático del Habla (RAH) están compuestos de varias etapas en las que se pueden utilizar diversas metodologías (Figura 1). De forma muy esquemática, los objetivos de la primera etapa son la adquisición de la señal vocal y la extracción de aquellas características más relevantes (*Preproceso*). La salida de esta etapa es una secuencia de vectores de características o cadenas de símbolos pertenecientes a un alfabeto predeterminado. Las metodologías más usuales en esta etapa provienen, fundamentalmente, del marco del Procesado de Señales y han sido discutidas en el capítulo anterior. La segunda etapa tiene como objetivo el encontrar e identificar en la señal vocal segmentos de voz que estén relacionados con alguna unidad lingüística. Esta etapa es conocida como *Decodificación Lingüística*, y en ella se utilizan modelos acústicos de unidades que pueden ser de nivel subléxico (fonemas, sílabas …) o léxico (palabras). La salida de esta etapa es una (o varias) secuencia(s) de unidades lingüísticas. Inevitablemente, estas secuencias contienen errores debidos, principalmente, a la pobreza del modelado acústico de las unidades utilizadas. La tercera etapa (*Análisis*) tiene como objetivo corregir la anterior secuencia de unidades lingüísticas y/o extraer la información semántica que contenga la pronunciación, empleando para ello conocimiento sintáctico, semántico o pragmático sobre la tarea. En este capítulo se presentan las metodologías que se vienen utilizando en las dos últimas etapas.

Cada una de las etapas descritas requiere la existencia de una serie de modelos que deben ser construidos en una fase de aprendizaje. Este aprendizaje se puede llevar a cabo deductiva y/o inductivamente. En el *Aprendizaje Deductivo* los modelos se construyen a partir del conocimiento que poseen expertos humanos. El *Aprendizaje Inductivo*, por contra, construye automáticamente un modelo de la realidad a partir de ejemplos concretos. Ambas técnicas de aprendizaje pueden participar en diferente proporción en las metodologías que estudiamos en este capítulo.

El problema del RAH se abordó inicialmente mediante la metodología conocida como Alineamiento Temporal No Lineal (ATNL) [1] [2]. En los años setenta esta técnica fue ampliamente utilizada y permitió dar una primera solución al problema del *Reconocimiento de Palabras Aisladas* (RPA), esto es, cada palabra es pronunciada entre pausas. También se abordó mediante esta técnica el problema del *Reconocimiento de Palabras Conectadas* (RPC), esto es, de secuencias de palabras pronunciadas sin pausas intermedias (por ejemplo, secuencias de dígitos). El aprendizaje en sistemas basados en ATNL es fuertemente inductivo. Actualmente, muchos de los sistemas de reconocimiento de palabras aisladas comercializados están basados en esta metodología [3] [4] [5].

La segunda metodología que cabe destacar es la de los *Sistemas Basados en el Conocimiento*, formalizada a finales de los años setenta, si bien muchos de los sistemas propuestos con anterioridad pueden considerarse basados en un mayor o menor grado en esta metodología [6] [7] [8]. En los Sistemas Basados en el Conocimiento, el aprendizaje deductivo es el predominante y en ocasiones el único. A pesar del gran auge que estos sistemas experimentaron inicialmente, la imposibilidad de que los conocimientos de expertos puedan ser representados y utilizados adecuadamente en un computador ha provocado un cierto abandono

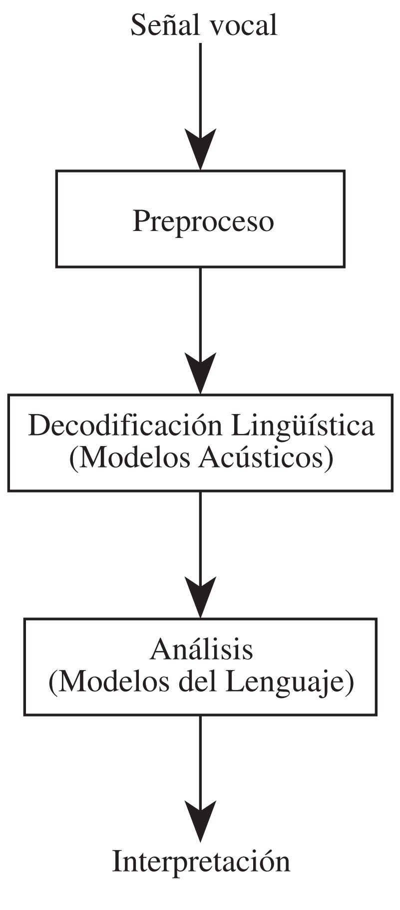

Figura 1: Esquema general de un sistema de Reconocimiento Automático del Habla.

de esta metodología. No obstante, se sigue trabajando con ella ya que, por un lado, permite profundizar en nuestra comprensión del proceso del habla y, por otra parte, se confía en que en un futuro se desarrollen las técnicas necesarias para que dichos conocimientos puedan ser representados y convenientemente tratados.

La tercera metodología, y la más empleada hoy en día, está basada en la Teoría de los Procesos Estocásticos, siendo los *Modelos Ocultos de Markov* (MOMs) el tipo de modelo más utilizado [9] [10]. El éxito de los MOMs reside en que existen poderosos algoritmos para realizar el aprendizaje inductivo de la componente probabilística de los modelos. La utilización de esta metodología en el RAH data de la mitad de los años setenta, pero no es hasta los ochenta cuando se produce su eclosión definitiva. Sus primeras aplicaciones fueron en RPA [11] y RPC. Desde la segunda mitad de los ochenta se está utilizando con gran éxito en el modelado subléxico (de unidades mas pequeñas que las palabras).

Actualmente se está intentando modelar las restricciones semánticas de tareas o aplicaciones concretas con tales modelos. Para representar las restricciones sintácticas, dentro de esta metodología se están utilizando modelos denominados N-gramas [12], que se tratan en el Capítulo 5 de este tema.

La cuarta metodología, basada en la Teoría de los Lenguajes Formales, fue utilizada en un principio básicamente para representar restricciones sintácticas [13] [14]. Aunque las gramáticas que representan al modelo del lenguaje eran dadas deductivamente, con la introducción de la Inferencia Gramatical se pudieron obtener inductivamente a partir de muestras de texto escrito y/o muestras de voz. La Inferencia Gramatical no sólo se ha utilizado para obtener modelos del lenguaje, sino también modelos acústicos estructuralmente complejos de las unidades subléxicas. Esta metodología y la anterior se encuentran estrechamente relacionadas, aunque en la Teoría de los Lenguajes Formales se pone un mayor énfasis en el aprendizaje inductivo de la estructura.

Finalmente, se debe destacar una metodología emergente, la basada en las Redes Neuronales Artificiales [15]. El éxito reciente de estos modelos se debe a la existencia de algoritmos que permiten obtener inductivamente un conjunto de pesos de la red a partir de muestras reales de voz. Inicialmente estos modelos se utilizaron para el RPA; no obstante, en la actualidad se están realizando diversos estudios encaminados a permitir su uso en el modelado acústico subléxico y del lenguaje.

En los apartados siguientes se describen con detalle cada una de estas metodologías y su aplicación al modelado léxico para RPA y/o RPC.

2 Alineamiento Temporal No Lineal

2.1 Comparación entre secuencias acústicas: técnicas de alineamiento

De entre el conjunto de posibilidades que hay a la hora de representar los objetos acústicos tal vez la más intuitiva consiste en la propia secuencia de *vectores de características* (*tramas*) que se obtiene como resultado del preproceso. Una de las principales dificultades que entrañan esta y otras representaciones es la variabilidad temporal existente entre distintos objetos: el complejo proceso de fonación hace que diferentes pronunciaciones de una misma palabra puedan tener duraciones muy distintas aun cuando provengan de un mismo locutor. Y no sólo eso: los diferentes elementos que componen la palabra no se ven afectados de igual manera por una locución más o menos rápida. Por ello, y en la medida de lo posible, un criterio de comparación entre objetos acústicos debería ser independiente de sus duraciones.

La primera aproximación conocida al problema es el *Alineamiento Temporal No Lineal* (ATNL) y de ella nos ocupamos en este apartado. El ATNL se apoya en la estrategia algorítmica de Programación Dinámica y permite establecer comparaciones entre objetos acústicos. Obtiene para ello un "alineamiento" de sus tramas que conduce a una medida de "disimilitud" independiente de las longitudes de los mencionados objetos. El valor de la disimilitud recurre, en última instancia, a la semejanza entre las tramas alineadas, por lo que será necesario disponer de una cierta medida de similitud entre tramas que denominaremos "distancia local" [1].

La comparación de dos objetos acústicos se establece en términos de búsqueda del camino más corto

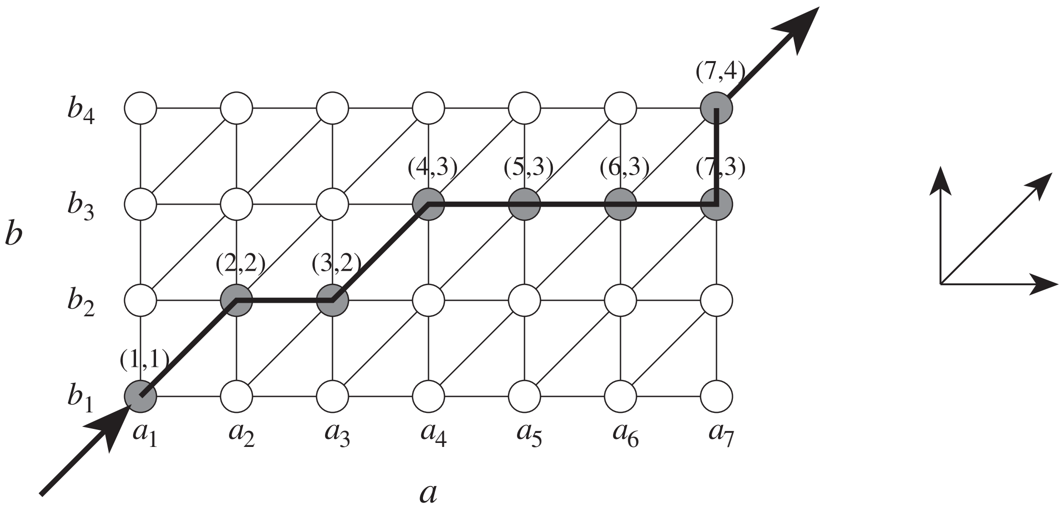

Figura 2: Ejemplo de Alineamiento Temporal No Lineal entre dos objetos acústicos: $a = a_1 a_2 \ldots a_7$ y $b = b_1 b_2 b_3 b_4$, donde cada elemento a_i o b_j es una trama. El grafo de búsqueda es una cuadrícula en la que cada nodo corresponde a un alineamiento entre tramas. A la derecha se muestran las producciones empleadas en la definición de transiciones entre nodos. En trazo grueso se muestra un camino de alineamiento.

en un grafo cuyos nodos se disponen en una cuadrícula (véase la Figura 2). En el eje horizontal se ubican las tramas de uno de los objetos a comparar, a (con un número de tramas igual a $|a|$), y en el eje vertical las del otro, b (de talla $|b|$). Cada nodo de la cuadrícula vendrá etiquetado por un par (i, j) que representa un alineamiento entre las tramas i-ésima de a y j-ésima de b. Las transiciones permitidas entre nodos (arcos) se describen mediante un conjunto de *producciones* (P). Una producción es un par de enteros positivos (x, y) que permite efectuar una transición válida entre un nodo (i, j) y otro $(i+x, j+y)$. A cada producción se le asigna un *peso* (w) que cumple la función de representar la "longitud" de la transición entre dos nodos.

El problema consiste en encontrar aquel camino que partiendo del nodo $(1, 1)$, empleando solo transiciones válidas y acabando en el nodo $(|a|, |b|)$, presenta un menor valor de la suma de distancias locales entre tramas alineadas (ponderadas por el peso de las producciones empleadas). Dicho valor recibe el nombre de "disimilitud".

Formalmente, el problema se puede plantear en términos de evaluación de la disimilitud $D(|a|, |b|)$ donde:

$$D(i, j) = \begin{cases} \infty, & \text{si } (i \leq 0 \wedge j < 0) \vee (i < 0 \wedge j \leq 0); \\ 0, & \text{si } i = 0 \wedge j = 0; \\ \min\limits_{\forall (x,y) \in P} D(i-x, j-y) + w(x, y) \cdot d(i, j), & \text{si } i \geq 1 \wedge j \geq 1; \end{cases}$$

y $d(i, j)$ representa la distancia local entre las tramas i-ésima de a y j-ésima de b.

Por medio de la técnica de Programación Dinámica, la solución al problema se alcanza efectuando una única visita a cada nodo del grafo. Al visitar un nodo (i, j) se obtiene el peso del mejor camino entre $(1, 1)$ e (i, j) consultando únicamente el peso de los mejores caminos que finalizan en nodos desde los que se puede efectuar una transición a (i, j). El coste computacional del ATNL es proporcional al producto de las tallas de los objetos acústicos comparados (y también al número de producciones, aunque éste es muy reducido), lo que lo hace perfectamente asumible en situaciones prácticas.

Para conseguir que la disimilitud sea independiente de las longitudes de las dos pronunciaciones, ésta es *normalizada* (dividida) por la suma de los pesos de las producciones utilizadas en el camino. Ocurre entonces que el método de Programación Dinámica descrito anteriormente sólo es aplicable si la suma de los pesos de las producciones empleadas es igual para todo camino de $(1,1)$ a $(|a|,|b|)$. Esto conlleva severas restricciones en la definición del conjunto de producciones y su ponderación. En la práctica dos familias de conjuntos de producciones son ampliamente utilizados. Los conjuntos de la primera se conocen como "de producciones simétricas" y en ellos la suma de pesos es proporcional a la suma de las longitudes de los dos objetos. Los conjuntos de la segunda familia se denominan "de producciones asimétricas" y la suma es proporcional a la longitud de uno solo de los objetos.

2.2 Reconocimiento de Palabras Aisladas mediante ATNL: Clasificación por Vecindad

El primer problema al que se aplicó la técnica de ATNL fue el Reconocimiento de Palabras Aisladas (RPA), consistente en identificar la pronunciación de una palabra (precedida y sucedida por silencio) perteneciente a un léxico conocido. Para modelar el léxico se utiliza un diccionario en el que cada palabra es descrita por uno o más "prototipos" (pronunciaciones concretas de la palabra).

Generalmente, los prototipos que componen el diccionario son seleccionados a partir de un gran conjunto de pronunciaciones. Se eligen como prototipos de las palabras del léxico aquellas pronunciaciones que se consideran más representativas de cada palabra. Para ello, se pueden aplicar diversos criterios (la mayoría de los cuales están basados en técnicas de agrupamiento) que permiten adecuar el resultado (la composición final del diccionario) a la tarea concreta que se va a abordar [4].

Si consideramos la disimilitud entre secuencias acústicas obtenida por ATNL como base para la definición de una "distancia" entre pronunciaciones y palabras del léxico, podemos aplicar técnicas de clasificación por vecindad (véase el Capítulo 2 del Tema 1) para efectuar el reconocimiento.

Las reglas de clasificación más empleadas en RPA son la del Vecino más Próximo (VP) y la de los k-Vecinos más Próximos (k-VP). Según la regla VP se define la *distancia* de una palabra del léxico a la pronunciación que deseamos reconocer como la mínima disimilitud entre cada uno de los prototipos de la palabra y dicha pronunciación. La pronunciación se cataloga entonces en la clase de la palabra que se encuentra a menor distancia. La regla k-VP, por su parte, define la distancia entre una palabra y la pronunciación desconocida como suma de las k menores disimilitudes entre prototipos de la palabra y la pronunciación (nótese que estamos suponiendo que existen al menos k prototipos de cada palabra en el diccionario). Al igual que en la regla VP, la pronunciación se identifica como perteneciente a la palabra más cercana.

Si bien ambas reglas ofrecen resultados satisfactorios, conviene indicar que, por lo general, la regla k-VP aventaja en prestaciones a la VP, pero fuerza a disponer de k o más prototipos por palabra, aumentando así la demanda de memoria y el coste de cómputo. La conveniencia de emplear la regla k-VP es especialmente notable en aplicaciones multilocutor. Las prestaciones de ambas reglas mejoran al aumentar el número de prototipos por palabra, ya que esto permite considerar un mayor número de articulaciones de una misma palabra. Existe, pues, un compromiso entre prestaciones y eficiencia computacional.

2.3 Reconocimiento de Palabras Conectadas por ATNL: Algoritmos de Dos Pasos y de Un Paso

El problema que sigue en dificultad al de Reconocimiento de Palabras Aisladas es el Reconocimiento de Palabras Conectadas (RPC). En él, la pronunciación que queremos reconocer es una frase compuesta por palabras de cierto léxico conocido que, al igual que en el caso del RPA, viene modelado por un diccionario de prototipos. Las técnicas de Alineamiento Temporal No Lineal también han sido aplicadas a este problema con notable éxito.

El denominado "algoritmo de Dos Pasos" se publicó a finales de la década de los 70 por Sakoe [16] y debe su nombre a la división que efectúa del problema en dos etapas. En la primera etapa se calcula la distancia entre *toda* porción de la pronunciación y *toda* palabra del diccionario. La distancia, como ya indicamos antes, se define en términos de la disimilitud que obtenemos por ATNL. La segunda etapa del algoritmo de Dos Pasos trata de encontrar la mejor división de la secuencia acústica en segmentos que identificamos/etiquetamos con sus respectivas palabras más cercanas. La mejor segmentación de la pronunciación es aquella con menor suma de distancias segmento-palabra. Esta segunda etapa puede plantearse en términos de búsqueda de un camino más corto en un grafo mediante la resolución de una ecuación recursiva abordable también por Programación Dinámica (Figura 3).

El algoritmo de Dos Pasos presenta una elevada complejidad computacional debido al considerable número de disimilitudes segmento-prototipo que deben ser calculadas en el primer paso: lineal con el número de prototipos del diccionario por el cuadrado de la longitud de la pronunciación. Este coste se puede reducir y pasar a ser proporcional al producto del número de prototipos y la talla de la secuencia acústica si se aprovecha el hecho de que al obtener la disimilitud entre un prototipo y un segmento de la pronunciación se obtiene, como subproducto, la disimilitud entre dicho prototipo y todo prefijo del segmento. En cualquier caso, cada disimilitud supone un coste proporcional al producto de las tallas del segmento y del prototipo implicados en la comparación.

En 1984, Hermann Ney publicó un artículo en el que describía con gran nivel de detalle el denominado "algoritmo de Un Paso" [17]. Con su trabajo, Ney contribuyó a divulgar dicho algoritmo que, aunque ya había sido publicado en 1971 por Vintsyuk, no gozaba del reconocimiento que hoy tiene. El algoritmo de Un Paso aplica la técnica de ATNL haciendo posible computar en una sola etapa el mejor alineamiento entre la pronunciación y la mejor secuencia de prototipos. La idea fundamental consiste en formar una nueva cuadrícula en cuyos ejes horizontal y vertical se disponen, respectivamente, las tramas de la frase pronunciada y las de todos los prototipos del diccionario (ver Figura 4). Un nodo de la cuadrícula será una tripleta (i,Ω,j), indicando un alineamiento entre la i-ésima trama de la secuencia y la j-ésima trama del prototipo Ω. El camino que buscamos sobre la cuadrícula partirá de un nodo cualquiera de la forma $(1,\Omega,1)$ y deberá alcanzar otro cualquiera de la forma $(|a|,\Omega',|\Omega'|)$, siendo $|a|$ el número de tramas de la pronunciación y $|\Omega'|$ el del prototipo Ω'. Además de permitir las transiciones entre nodos de la cuadrícula impuestas por el conjunto de producciones, es lícito efectuar transiciones de un nodo $(i,\Omega,|\Omega|)$ a otro $(i+1,\Omega',1)$. Con ello se hace posible que el camino de alineamiento discurra por varios prototipos de modo que cada uno describa un segmento de la pronunciación.

La complejidad computacional del algoritmo de Un Paso es proporcional al producto de la talla de la pronunciación por la suma de las tallas de los prototipos del diccionario, muy inferior, por tanto, a la complejidad del algoritmo de Dos Pasos. Una ventaja adicional de este algoritmo es la facilidad con que permite implementaciones *síncronas con la pronunciación*. Esta sincronía resulta adecuada para el diseño de sistemas que deban proporcionar una respuesta en tiempo real, pues el tiempo que transcurre entre la recepción de tramas consecutivas puede ser invertido en avanzar en el cómputo de la solución. Otra característica importante que explica el interés que ha despertado el algoritmo de Un Paso es la facilidad con que se puede reducir su complejidad computacional en varios órdenes de magnitud por medio de estrategias algorítmicas como la Búsqueda en Haz, a costa, eso sí, de devolver sólo una *aproximación* a la solución óptima.

En contrapartida hay que señalar que el algoritmo de Un Paso resuelve sólo ciertas formalizaciones del problema, que son un subconjunto de aquellas abordables por el algoritmo de Dos Pasos. Éste último permite, por ejemplo, mayor flexibilidad en la elección de criterios de normalización temporal, definición de la distancia segmento-palabra e imposición de restricciones de duración [18].

Tanto el algoritmo de Un Paso como el de Dos Pasos pueden ser modificados para proporcionar hipótesis de la secuencia de palabras pronunciada que observan restricciones sintácticas de tipo "regular". Estas restricciones permiten describir las sucesiones de palabras que se pueden dar en ciertas aplicaciones, pero distan mucho de modelar el lenguaje natural. En cualquier caso, su utilización permite restringir sustancialmente el espacio de búsqueda y conduce a mejorar notablemente el resultado.

Finalmente debemos indicar que no hemos hecho alusión al algoritmo Constructor de Niveles (debido

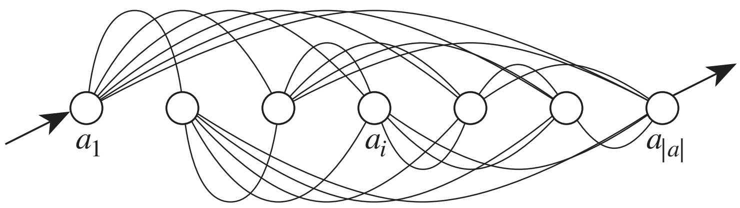

Figura 3: Grafo de búsqueda de la segunda etapa del algoritmo de Dos Pasos. Los nodos corresponden a las tramas de la pronunciación que queremos reconocer. Un arco entre dos nodos representa el "peso" de la menor distancia existente entre el propio segmento y toda palabra del diccionario.

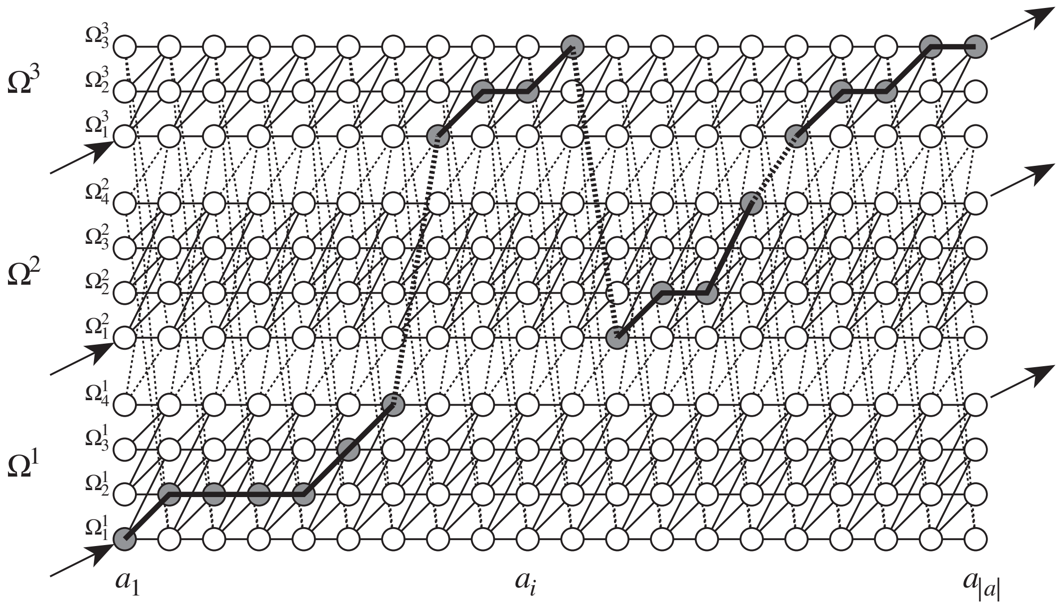

Figura 4: Grafo sobre el que se efectúa la búsqueda del mejor camino de alineamiento en el algoritmo de Un Paso. En el eje horizontal se disponen las tramas de la pronunciación que queremos reconocer. En el vertical, las tramas de todos los prototipos. Con trazado sólido se muestran las transiciones permitidas por el conjunto de producciones. Con trazado discontinuo, las transiciones entre modelos. En la figura se muestra, en trazado grueso, un ejemplo de alineamiento.

a Myers y Rabiner, en 1981 [19]) que gozó de cierta aceptación a principio de la década de los 80, pues, como varios autores han puesto de manifiesto [20] [18], no es más que una versión del algoritmo de Un Paso con restricciones sintácticas para acotar inferior y superiormente el número de palabras que pueden formar la secuencia de salida.

3 Métodos basados en el Conocimiento

3.1 Introducción

La metodología que subyace en los denominados *Sistemas Basados en el Conocimiento* (SBC) ("Knowledge Based Systems") está fuertemente entroncada en la disciplina de la Inteligencia Artificial (IA) [21], históricamente relacionada con el Reconocimiento Automático del Habla (RAH).

En el marco del RAH se han realizado importantes aportaciones a la Inteligencia Artificial. Ya desde el proyecto ARPA-SUR [6] [7] se propusieron arquitecturas ampliamente utilizadas en IA —como, por ejemplo, la arquitectura de "pizarra" ("blackboard") del sistema HERSAY-II [22]— y se realizaron importantes estudios sobre la optimalidad de las estrategias de búsqueda heurística (valga como ejemplo el sistema HWIM [23]). Por otra parte, en el RAH se han empleado gran cantidad de técnicas originadas en el campo de la IA como, por ejemplo, las redes semánticas del sistema EVAR [24], las redes de transición aumentada de los sistemas EVAR, HWIM y ARIAL-II [25], o casos particulares del algoritmo $A^{\star}$ como el "decodificador de pila" (*Stack Decoding*) utilizado en los sistemas IBM [9] [26]. Pero son los métodos basados en el conocimiento, y en especial la tecnología de los *Sistemas Expertos*, los que han tenido una influencia mayor en el RAH [27] [28] [29] [30] [31].

3.2 Sistemas Basados en el Conocimiento

El problema del RAH puede verse como el resultado de un complejo proceso de interpretación de la señal vocal para obtener una descripción simbólica relevante para cierta aplicación. Como ya se ha comentado, para abordar satisfactoriamente este problema es necesario hacer cooperar distintos niveles de comprensión.

Una aproximación basada en el conocimiento incorpora deductivamente al sistema todo el conocimiento que a priori se posee acerca de los diferentes aspectos del problema. Este conocimiento a priori debe contener no sólo información lingüística general (conocimiento acústico-fonético, fonológico, sintáctico y semántico) sino también incorporar conocimiento pragmático acerca del dominio de la tarea e información sobre las posibles estrategias de diálogo. Obviamente, esto exige suponer que dicho conocimiento está disponible en forma explícita entre los humanos, lo que no siempre es una suposición válida. Aunque este modo de transferencia de conocimientos ha sido tradicionalmente la forma esencial en que se han venido construyendo la práctica totalidad de los sistemas de IA, es con la formalización de los métodos basados en el conocimiento cuando este modo de incorporar el conocimiento al sistema se realiza de forma explícita y plenamente asumida.

El diseño de todo SBC implica abordar tres problemas fundamentales. El primer problema es el de la *adquisición del conocimiento* (o aprendizaje), donde los nuevos conocimientos que se incorporan pueden interaccionar con los ya existentes, modificándolos para evitar incoherencias. El conocimiento que deseamos capturar puede ser susceptible de formalizarse simultáneamente tanto en forma procedural como en forma declarativa. Como ejemplo ilustrativo se presenta en la Figura 5 una regla típica de la base de conocimientos del módulo de decodificación lingüística (categorías acústico-fonéticas generales) del sistema ROARS [32]. La regla, en forma declarativa, es una de las que permiten la detección de segmentos oclusivos sordos. La Figura 6 representa el significado físico de la misma.

En la comprensión del discurso hablado los humanos ponemos en juego una gran cantidad de recursos y conocimientos diversos que son de naturaleza implícita e indescifrables en su totalidad (entendemos pero no somos conscientes del proceso mental seguido). Debido a ello, y aunque se han realizado algunas tentativas

Regla Detección segmentos oclusivos sordos

    para cada EP > 0 **hacer**

        encontrar SP > 0 **en** $[\mathit{posición}(\mathrm{EP} > 0) - 5, \mathit{posición}(\mathrm{EP} > 0)]$

    si encontrado SP > 0 **entonces**

        Existe segmento oclusivo sordo.

        Principio segmento oclusivo:

            desde posición(SP > 0) **hacia atrás**

                encontrar EMd < 0 ó SP $= 0$

            Principio segmento := máximo$(\mathit{posición}(\mathrm{EMd} < 0), \mathit{posición}(\mathrm{SP} = 0))$

        Final segmento oclusivo:

            desde $\mathit{posición}(\mathrm{EP} > 0)$ **hacia delante**

                encontrar D > 0 **en** $[\mathit{posición}(\mathrm{EP} > 0), \mathit{posición}(\mathrm{EP} > 0) + 7]$

            si encontrado D > 0 **entonces**

                Final segmento := $\mathit{posición}(\mathrm{D} > 0)$

            si no

                Final segmento := $\mathit{posición}(\mathrm{EP} > 0)$

Figura 5: Ejemplo de regla declarativa para la detección de segmentos oclusivos sordos. EP, EM detectan cambios bruscos en la energía, SP determina la correspondencia con una forma espectral sorda y D marca el final de la explosión. Evidentemente, para estos parámetros existen procedimientos de cálculo específicos.

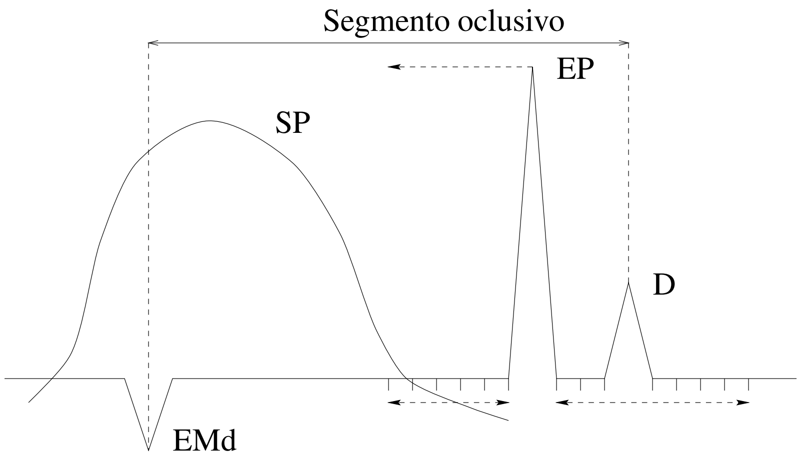

Figura 6: Ejemplo artificial de la localización de segmentos oclusivos sordos.

dependientes de la tarea [33], la adquisición y formalización del conocimiento es un problema abierto y para el cual no existen soluciones satisfactorias.

El segundo problema, obviamente muy ligado al primero, es la *representación del conocimiento*. La adecuada elección tanto del formalismo de representación como de la organización general de la base de conocimientos tiene una enorme influencia en los resultados finales de los SBC. En general, ésta debería satisfacer los siguientes requisitos [34]:

1. Integrar los diferentes aspectos del conocimiento.
2. Integrar los diferentes niveles de conocimiento.
3. Hacer explícito el conocimiento asociado a los diferentes niveles de comprensión.
4. Y permitir una construcción modular de la base de conocimientos del sistema.

Por último, el tercer problema se plantea en el *diseño de una estructura de control* que permita la manipulación del conocimiento de forma óptima en la resolución de un problema concreto. En general, las técnicas empleadas provienen del campo de la resolución de problemas en Inteligencia Artificial. Las soluciones basadas en el algoritmo $A^\star$ son las más utilizadas como *criterios y estrategias de análisis* [35].

A pesar de las importantes limitaciones que presentan los SBC, debidas principalmente a su carácter deductivo en la adquisición y a la transferencia de conocimientos al sistema, existe todavía un notable interés por esta metodología [34] [31] [36]. Los motivos son diversos, aunque quizás se podrían destacar tres de ellos. El primero reside en que se siguen presentando problemas acuciantes en el campo del RAH para los que aún no se ha encontrado solución (algorítmica) satisfactoria con las técnicas convencionales (como los Modelos Ocultos de Markov). El segundo motivo radica en la gran cantidad de resultados aprovechables, derivados de los trabajos que a lo largo de todos estos años se han venido realizando en los campos de la fonética acústica y la fonología para las diferentes lenguas, así como en los importantes logros en cuanto al conocimiento sintáctico-semántico y pragmático derivado de los estudios sobre Lenguaje Natural y Sistemas de Diálogo. Finalmente, hay que destacar que en los últimos tiempos se han conseguido notables avances en el problema de representación del conocimiento mediante el diseño de lenguajes específicos.

3.3 Conclusiones

Actualmente existe una cierta tendencia a considerar a los métodos basados en el conocimiento como una razonable aproximación para la manipulación del conocimiento deductivo que a priori se posee en los sistemas de RAH. Esto es así, sobre todo en la etapa de Decodificación Lingüistica [8] [32] y en los niveles superiores de Modelado del Lenguaje y/o Módulo de Diálogo [37] [34].

Sin embargo, esta metodología aún presenta algunos problemas para los cuales no existe una solución satisfactoria. Dos de los más importantes son: el problema de la adquisición del conocimiento y el problema de la posible integración de este conocimiento a priori deductivo en sistemas híbridos más generales.

4 Aproximación Estocástica

En esta aproximación al RAH, cada interacción entre un usuario y el sistema se ve como un experimento aleatorio consistente en que el usuario produce una señal vocal con la intención de que el sistema la interprete de una cierta manera. Bajo esta óptica, tanto la interpretación deseada por el usuario ("interpretación correcta") como la señal vocal producida por el mismo ("entrada al sistema") se consideran variables aleatorias que siguen una cierta distribución conjunta que, en principio, es desconocida. La misión del sistema será, habiendo observado únicamente el valor de su entrada, formular una hipótesis acerca de la interpretación correcta (esta hipótesis será la "interpretación del sistema", su "respuesta"). Obsérvese que, para poder llevar a cabo su misión con un cierto grado de acierto, el sistema deberá poseer algún tipo de información sobre la distribución conjunta de su entrada y la correspondiente interpretación correcta. Esta información se concretará, como veremos, en un cierto conjunto de modelos estocásticos.

Formalmente, sea E la variable aleatoria que representa la entrada al sistema de RAH, y que toma valores en un cierto conjunto $\mathbf{E}$ de entradas posibles. Para simplificar, consideraremos como entrada al sistema el resultado de la fase de preproceso (Tema 4, Capítulo 2). Esto nos permite ver $\mathbf{E}$ como un conjunto de secuencias, finitas y no nulas, de elementos de un cierto dominio $\mathbf{D}$ que se utiliza para representar fragmentos de la señal vocal correspondientes a intervalos de tiempo muy pequeños ("tramas"). Como ya sabemos, estos elementos pueden ser símbolos extraídos de un alfabeto finito o bien vectores de características.

Así mismo, sea I la variable aleatoria que representa la interpretación correcta, y que toma valores en un cierto conjunto $\mathbf{I}$ de interpretaciones posibles y, por tanto, de posibles respuestas del sistema.

Si el sistema de RAH pretende minimizar la probabilidad de producir interpretaciones erróneas[1], ante una entrada $e \in \mathbf{E}$ deberá producir la interpretación más probable en ese caso, es decir, $i^* = \arg\max \Pr(I = i \mid E = e)$ con i variando en $\mathbf{I}$. Resulta trivial demostrar que, entonces, $i^* = \arg\max \Pr(E = e \mid I = i) \cdot \Pr(I = i)$ con i variando en $\mathbf{I}$. A las probabilidades (o densidades de probabilidad)[2] del tipo $\Pr(E = e \mid I = i)$ las denominaremos "probabilidades acústicas", y a las del tipo $\Pr(I = i)$, "probabilidades a priori" o "de modelo de lenguaje".

Generalmente, el sistema desconocerá los valores de las probabilidades implicadas en la maximización anterior, y tendrá que limitarse a calcular ciertas estimaciones de las mismas, más o menos aproximadas, utilizando para ello un cierto conjunto de modelos estocásticos.

En tareas de reconocimiento de palabras aisladas (RPA), el conjunto $\mathbf{I}$ de interpretaciones posibles es igual a un cierto vocabulario finito $\mathbf{V}$, y una posibilidad consiste en dotar al sistema, para cada palabra v de $\mathbf{V}$, de una estimación P_v de su probabilidad a priori, y de un modelo M_v que le permita estimar sus correspondientes probabilidades acústicas.

En el caso de reconocimiento de palabras conectadas (RPC), las interpretaciones posibles son las secuencias, finitas y no nulas, de palabras de un cierto vocabulario finito $\mathbf{V}$, es decir, $\mathbf{I} = \mathbf{V}^+$. Las probabilidades acústicas pueden estimarse, de nuevo, utilizando un modelo diferente para cada palabra, suponiendo que sabemos construir el modelo acústico de cada secuencia de palabras a partir de los modelos acústicos de las palabras que la forman. Por otra parte, para estimar las probabilidades a priori necesitamos un modelo estocástico de $\mathbf{V}^+$, esto es, un modelo de lenguaje.

4.1 Modelos Ocultos de Markov

Se denomina modelo oculto de Markov (MOM) a una función probabilística de una cadena de Markov. Un MOM es, por tanto, un proceso doblemente estocástico: por una parte (que consideraremos oculta para cualquier observador externo del modelo), la cadena de Markov va evolucionando en el tiempo a través de una serie de estados según unas ciertas probabilidades de transición; por otra parte, cada estado tiene asociada una distribución de probabilidad de donde el modelo, cuando se encuentra en ese estado, extrae ("emite") un elemento (observable desde el exterior).

En principio, podemos formalizar el concepto de MOM como una tupla $M = (Q, \Sigma, \Pi, A, B)$ donde:

- Q es el conjunto, finito y no vacío, de estados del modelo. En cada instante discreto de tiempo, $t \geq 1$, M se encontrará en un estado de Q que denotaremos mediante la variable aleatoria X_t.

- Σ es el conjunto de elementos observables. En cada instante discreto de tiempo, $t \geq 1$, M emitirá un elemento de Σ que denotaremos mediante la variable aleatoria Y_t.

[1] Existe un planteamiento más general del problema, según el cual el objetivo es minimizar el valor esperado de un cierto coste que es función de la interpretación correcta y la producida por el sistema. En nuestro caso, consideramos que el coste es 0 si las interpretaciones coinciden y 1 en caso contrario.

[2] En adelante, utilizaremos el término probabilidad (y la notación Pr) para referirnos indistintamente a probabilidades o a densidades de probabilidad. El tratamiento matemático es análogo en ambos casos.

- Π es la distribución de probabilidad sobre los estados de Q que representa las probabilidades iniciales de los estados, de modo que $\Pi_q = \Pr(X_1 = q)$ para todo $q \in Q$.

- A_q es, para cada $q \in Q$, la distribución de probabilidad sobre los estados de Q que representa las probabilidades de transición desde q, de modo que $A_{qr} = \Pr(X_{t+1} = r \mid X_t = q)$ para todo $r \in Q$ y para todo $t \geq 1$.

- B_q es, para cada $q \in Q$, la distribución de probabilidad sobre los elementos de Σ que representa las probabilidades de emisión en q, de modo que $B_{qy} = \Pr(Y_t = y \mid X_t = q)$ para todo $y \in \Sigma$ y para todo $t \geq 1$.

Utilizaremos el término “componente estructural” o bien “topología” de M para hacer referencia a su conjunto de estados, Q, y su conjunto de transiciones con probabilidad no nula (“permitidas”), $\{(q,r) \in Q \times Q \mid A_{qr} > 0\}$. Por otra parte, con “la componente probabilística” o bien “las probabilidades” de M, haremos referencia conjunta a las distribuciones especificadas mediante Π, A y B.

Con lo dicho hasta ahora, el MOM M genera una secuencia infinita de elementos observables. Lo que nos interesa, sin embargo, son modelos que generen secuencias finitas y no nulas (de elementos de **D** en el caso de modelos acústicos; de palabras de **V** en el caso de modelos de lenguaje para RPC), de forma que podamos considerar que el modelo asigna a cada secuencia finita la probabilidad con la que la genera. Por este motivo, en RAH suelen utilizarse MOMs con un cierto estado especial que representa la parada del modelo. Formalmente, sea $\mathit{fin} \notin Q$ el estado de parada de M y consideremos que las distribuciones A_q, con $q \in Q$, son sobre $Q \cup \{\mathit{fin}\}$. En la Figura 7 se presenta una descripción algorítmica del funcionamiento del modelo M.

Para utilizar MOMs en RAH, es decir, para poder estimar con ellos probabilidades acústicas y de modelo de lenguaje, necesitamos tener resuelto el siguiente problema: dado un MOM $M = (Q, \mathit{fin}, \Sigma, \Pi, A, B)$ y una secuencia $y_{1\ldots T} \in \Sigma^+$, calcular la probabilidad de que M genere $y_{1\ldots T}$, es decir, que Y_t tome el valor y_t para todo $t \in \{1, \ldots, T\}$ y que $X_{T+1} = \mathit{fin}$. Si llamamos $M(y_{1\ldots T})$ a dicha probabilidad, podemos afirmar que

$$M(y_{1\ldots T}) = \sum_{q_{1\ldots T} \in Q^T} \Pi_{q_1} \cdot B_{q_1 y_1} \cdot \left(\prod_{t=2}^{T} A_{q_{t-1} q_t} \cdot B_{q_t y_t} \right) \cdot A_{q_T \mathit{fin}}.$$

Llevar a cabo el cálculo empleando directamente esta fórmula da lugar a un algoritmo cuyo coste temporal es prohibitivo, $O(T \cdot |Q|^T)$. No obstante, empleando la técnica de Programación Dinámica, se obtienen algoritmos cuyo coste temporal es $O(T \cdot |Q|^2)$ con un coste espacial $O(T \cdot |Q|)$.[3] Por ejemplo, el algoritmo “Forward” calcula $M(y_{1\ldots T})$ como $\sum_{q \in Q} \alpha(q,T) \cdot A_{q\mathit{fin}}$, donde “las alfas”, que se definen recursivamente como

$$\alpha(q,t) = \begin{cases} \Pi_q \cdot B_{q y_1}, & \text{si } (q \in Q) \wedge (t = 1); \\ \left(\sum\limits_{r \in Q} \alpha(r, t-1) \cdot A_{rq} \right) \cdot B_{q y_t}, & \text{si } (q \in Q) \wedge (2 \leq t \leq T); \end{cases}$$

se calculan haciendo uso de un bucle de inducción que evoluciona hacia adelante en el tiempo. Otra opción es el algoritmo “Backward”, similar al anterior y con los mismos costes computacionales, pero con el bucle de inducción hacia atrás [10].

Si disponemos de los MOMs adecuados, cualquiera de los dos algoritmos anteriormente mencionados nos permitiría estimar, ante cualquier entrada al sistema de RAH, las probabilidades acústicas y de modelo de lenguaje correspondientes a cada posible interpretación de la entrada. Se plantea, por tanto, el problema de obtener dichos MOMs. Este problema suele abordarse en dos etapas: deductiva e inductiva.

En la etapa deductiva, el diseñador del sistema de RAH, basándose en su conocimiento a priori del problema (posiblemente obtenido a partir de una experimentación previa), fija ciertas características de los

[3] En realidad, estos algoritmos pueden “implementarse” con un coste espacial $O(|Q|)$ Además, el factor $|Q|^2$ en el coste temporal puede rebajarse a la talla del conjunto de transiciones permitidas de M.

modelos. Debe decidir, por ejemplo, de qué tipo serán las distribuciones B_q. Si el conjunto de elementos observables Σ es un alfabeto finito, se utilizarán distribuciones "multinomiales" (MOMs "discretos" [10]), pero si los elementos son vectores de características reales habrá que elegir algún tipo de distribución continua multidimensional [38] [39]: normal (con matriz de covarianzas diagonal o no), combinación lineal convexa (de un cierto número) de normales o quizá algún otro. También habrá que considerar la posibilidad de que el modelo emita en las transiciones en vez de en los estados [40], y en caso de emitir en las transiciones, habrá que decidir si se permiten transiciones nulas, es decir, transiciones en las que no se emite ningún elemento observable [9]. Otro aspecto que hay que tener en cuenta en esta etapa es cómo se modela la duración de los estados, es decir, cómo determina el modelo cuántos instantes de tiempo consecutivos permanece en un estado una vez entra en él. Si se respeta el carácter markoviano de la cadena oculta (la formalización que hemos presentado), la probabilidad de que la duración de un cierto estado $q \in Q$ sea d es $(A_{qq})^{d-1} \cdot (1 - A_{qq})$ para todo $d \geq 1$; alternativamente, puede decidirse utilizar una cadena semi-markoviana, es decir, modelar explícitamente esa duración utilizando una distribución más adecuada, de modo que cuando el modelo transite al estado q, extraiga un valor d de esa distribución que determine que el modelo permanezca d instantes consecutivos emitiendo elementos observables según B_q antes de transitar a otro estado (necesariamente diferente). Los modelos semi-markovianos pueden producir, en RAH, tasas de reconocimiento mayores que los MOMs habituales, pero a costa de incrementar el orden de complejidad computacional de los algoritmos que los manejan [41] [42] [43] [44]. También es una posible decisión de diseño imponer ciertas restricciones sobre los parámetros de los modelos: por ejemplo, es una práctica frecuente "ligar" probabilidades de emisión, es decir, forzar a que las distribuciones de emisión de dos estados diferentes sean iguales.

Otro aspecto importante en la construcción de un MOM es la determinación de su topología. Aunque existen propuestas para aprenderla a partir de un conjunto de ejemplos de secuencias que debe generar el modelo [45] [46] [47], lo habitual es que se decida en la etapa deductiva. Generalmente (sobre todo en modelado acústico) se utilizan topologías de izquierda a derecha, que son aquellas en las que es posible ordenar los estados del modelo de modo que no esté permitida ninguna transición de un estado a otro anterior en dicha ordenación. Con ello se pretende reflejar en el modelo la evolución de la señal vocal en el tiempo.

Todos los elementos de los modelos que no han sido especificados en la etapa deductiva de su construcción, deben serlo en la etapa de aprendizaje inductivo. En esta etapa no es el diseñador quien aporta directamente su conocimiento al sistema, sino que es un algoritmo quien extrae conocimiento de un conjunto de ejemplos. Estos ejemplos son, generalmente, pares compuestos por una posible entrada al sistema y su correspondiente interpretación correcta. Dejando aparte interesantes propuestas heurísticas [48], esta etapa inductiva puede verse como un problema de optimización con restricciones donde el espacio de búsqueda ha quedado fijado en la etapa deductiva y la función objetivo depende del conjunto de ejemplos [49].

Para ilustrar las ideas expuestas anteriormente vamos a presentar un ejemplo sencillo. Supongamos una tarea de RPA cuyo vocabulario es $\mathbf{V} = \{Sí, No\}$. Supongamos que disponemos de un preproceso que da como resultado secuencias de símbolos del conjunto $\mathbf{D} = \{1, \ldots, 16\}$ y que disponemos también de 200 ejemplos de tales secuencias: 100 de ellas como resultado de preprocesar pronunciaciones de síes y las 100 restantes correspondientes a noes. Supongamos además que sabemos a priori que el sistema recibirá una entrada correspondiente a un sí aproximadamente el 70% de las veces, mientras que las restantes entradas corresponderán a noes. Decidimos entonces construir el siguiente sistema de RAH, basado en una aproximación estocástica: tendremos como modelos acústicos dos MOMs, $M_{Sí}$ y M_{No}, que queremos construir de modo que, para toda entrada posible al sistema, $y_{1\ldots T}$, la probabilidad con la que la genere el modelo $M_{Sí}$, $M_{Sí}(y_{1\ldots T})$, sea una buena estimación de la probabilidad condicionada de que un hablante produzca la entrada $y_{1\ldots T}$ al sistema, dado que pronuncia un sí (y análogamente para M_{No}). Una vez construidos los dos MOMs, los utilizaremos para reconocer de la siguiente forma: ante una pronunciación del usuario, el preproceso proporcionará como entrada a nuestro sistema una secuencia $y_{1\ldots T}$ de elementos de $\mathbf{D}$; entonces, utilizando el algoritmo "Forward" (o, alternativamente, el "Backward") calcularemos $M_{Sí}(y_{1\ldots T})$ y $M_{No}(y_{1\ldots T})$;

finalmente, la respuesta de nuestro sistema será *Sí* si $M_{Sí}(y_{1...T}) \cdot 0.7 > M_{No}(y_{1...T}) \cdot 0.3$, y *No* en caso contrario. Por tanto, lo único que nos queda por hacer es construir los dos modelos acústicos. En la etapa deductiva decidimos especificar lo siguiente de los modelos: para cada $v \in \mathbf{V}$, $M_v = (Q, fin, \Sigma, \Pi, A(v), B(v))$ donde $Q = \{1,2,3\}$, $\Sigma = \mathbf{D}$, $\Pi_1 = 1$, $\Pi_2 = \Pi_3 = 0$ y $A(v)_{mn} = 0$ para todo $m > n$. Como consecuencia, en la etapa inductiva deberemos aprender los valores de 114 paramétros (por cada uno de los 2 modelos, las 9 probabilidades de transición no especificadas previamente y las $3 \cdot 16$ probabilidades de emisión) sujetos a ciertas restricciones de carácter estocástico. En este espacio de búsqueda intentaremos optimizar una cierta función objetivo: elegimos, por ejemplo, la verosimilitud de nuestro conjunto de 200 ejemplos (es decir, el producto de las probabilidades con las que $M_{Sí}$ genere los 100 ejemplos de síes y M_{No} genere los 100 de noes). Este problema de optimización constituye un problema clásico de estimación de probabilidades en MOMs según el criterio de máxima verosimilitud para el que se conoce una solución eficiente: el algoritmo de Baum-Welch, también conocido como "Forward-Backward" [50] [51] [52] [53].

El algoritmo de Baum-Welch es un caso particular de algoritmo EM ("Expectation-Maximization") [54] [55]. Los algoritmos EM son algoritmos iterativos que se utilizan en estimación de parámetros por máxima verosimilitud en presencia de información incompleta[4]. A partir de una estimación inicial que debe proporcionarse al algoritmo, este va mejorando la estimación a lo largo de las iteraciones en el sentido de que la verosimilitud del conjunto de ejemplos nunca decrece (pese a que no se puede asegurar que el algoritmo converja siempre a un máximo de esta función).

Aunque el criterio de máxima verosimilitud es el clásico en la estimación de los parámetros de los MOMs en RAH, presenta la debilidad de que los modelos resultantes carecen, generalmente, de gran parte de la capacidad discriminativa que se les supone cuando son utilizados para el reconocimiento. Ello se debe a que la función objetivo de este criterio sólo tiene en cuenta la probabilidad con la que cada ejemplo es generado por el modelo de su interpretación correcta, y no aquellas probabilidades que le asignan los modelos de las interpretaciones que, en reconocimiento, competirían con ella. La estimación por máxima información mutua es una propuesta alternativa que pretende resolver el problema anteriormente expuesto [56] [57].

A continuación vamos a ocuparnos de los problemas adicionales que presentan las tareas de RPC frente a las de RPA, más simples. Recordemos que las interpretaciones posibles en una tarea de RPC son secuencias formadas por palabras extraídas de un vocabulario finito, fijado de antemano. Por lo tanto, el tamaño del conjunto de interpretaciones posibles es extremadamente grande (potencialmente infinito). De este hecho se derivan dos consecuencias fundamentales. Primeramente, un sistema de RPC no podrá tener un modelo acústico separado para cada interpretación posible, sino que deberá saber componerlos cuando los necesite, a partir de un conjunto finito de modelos acústicos de las palabras del vocabulario. Y en segundo lugar, ante una entrada al sistema, la búsqueda de la interpretación (que se estime) más probable no podrá llevarse a cabo de forma exhaustiva y será necesaria alguna estrategia más inteligente.

Componer el modelo acústico de una secuencia de palabras a partir de los de las palabras que la forman es sencillo con los MOMs que hemos presentado [58]. Intuitivamente, consiste en concatenar los MOMs de las palabras de la secuencia, de modo que las transiciones de un MOM al siguiente se permiten al identificar el estado de parada del primero con la distribución inicial del segundo. Téngase en cuenta que este tipo de composiciones suelen ser necesarias no sólo a la hora de reconocer, sino también en la etapa de aprendizaje inductivo, ya que en tareas de RPC generalmente dispondremos de ejemplos correspondientes a secuencias de palabras, no a palabras aisladas.

Respecto a la búsqueda de la interpretación más probable, esta suele llevarse a cabo utilizando el algoritmo "Stack Decoding" [12] [9], que puede verse como una aproximación al problema empleando la técnica de Ramificación y Poda (con criterio de selección "primero el mejor" y una función heurística de evaluación de los nodos).

[4]La estimación de parámetros en MOMs a partir de secuencias de elementos observables puede verse como un caso de estimación a partir de información incompleta si tenemos en cuenta que las secuencias de estados correspondientes permanecen "ocultas" y no quedan unívocamente determinadas por las secuencias observadas.

Algoritmo Funcionamiento de M

$t := 1$

/* Instante 1 */

Elegir X_1 según la distribución Π

Emitir Y_1 según la distribución B_{X_1}

repetir

$t := t+1$

/* Instante t */

Transitar a X_t según la distribución $A_{X_{t-1}}$

si $X_t \neq \mathit{fin}$ **entonces** emitir Y_t según la distribución B_{X_t} **fin si**

hasta $X_t = \mathit{fin}$

fin

Figura 7: Descripción algorítmica del funcionamiento del MOM $M = (Q, \mathit{fin}, \Sigma, \Pi, A, B)$.

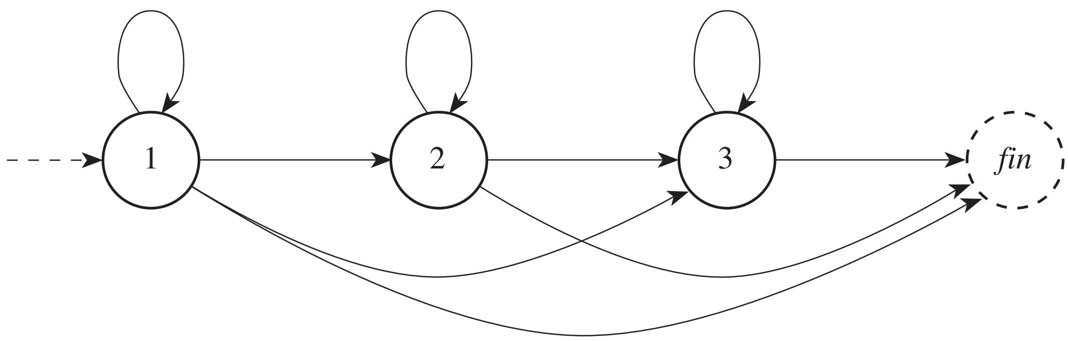

Figura 8: Topología elegida para los MOMs acústicos del ejemplo. Los estados se han representado mediante círculos; las transiciones permitidas, mediante flechas. La flecha en trazo discontinuo representa que el estado inicial es siempre el 1.

Vamos a concluir este apartado dedicado a los MOMs comentando lo que podríamos llamar “aproximación de Viterbi” al RAH. Se trata de una simplificación de la aproximación estocástica consistente en utilizar los modelos considerando que estos asignan a las secuencias que generan, no la probabilidad con las que las generan, sino la probabilidad conjunta de dicha secuencia y la secuencia de estados oculta más probable dada la secuencia generada (esta secuencia de estados suele denominarse “camino de Viterbi”, y la probabilidad conjunta correspondiente, “puntuación de Viterbi”) [59]. El artificio que supone considerar que los MOMs generan cada secuencia de elementos observables a lo largo de su camino de Viterbi, en vez de respetar la naturaleza aleatoria de la secuencia de estados, da lugar a algoritmos de menor coste computacional, tanto para el aprendizaje de los modelos (“Viterbi Extraction” [12]) como para su utilización en reconocimiento (algoritmo de Viterbi o “Viterbi Decoding” [60] [12]). Además, permite hacer uso del marco teórico del ATNL (identificando transiciones con producciones y logaritmos de probabilidades de emisión con distancias locales cambiadas de signo).

4.2 Gramáticas Estocásticas

Las gramáticas estocásticas (GEs) son otro tipo de modelo estocástico utilizado en RAH. El concepto de GE es una extensión probabilística del concepto de gramática propio de la teoría de lenguajes formales. Una gramática especifica un lenguaje (esto es, un conjunto de secuencias finitas de símbolos) mediante un conjunto finito de reglas que permiten generar todas las secuencias que pertenecen al lenguaje, y sólo esas. En una GE, cada regla lleva asociada una probabilidad de aplicación, lo que nos permitirá definir la probabilidad con la que la GE genera cada secuencia de símbolos.

Formalmente, podemos definir una GE como un par $G_P = (G, P)$ donde:

- G es una gramática, es decir, una tupla $G = (N, \Sigma, R, S)$ formada por: un conjunto finito y no vacío de no-terminales (N), de los cuales uno se distingue como símbolo inicial ($S \in N$); un conjunto finito y no vacío de terminales (Σ, disjunto con N); y un conjunto de reglas de reescritura (R). Como es habitual en la práctica, vamos a restringir nuestra discusión a gramáticas incontextuales, lo cual nos permite definir R como un subconjunto de $N \times (N \cup \Sigma)^+$. Denotaremos cada regla $r = (A, \alpha) \in R$ como $A \to \alpha$ y diremos que A es la parte izquierda de la regla r, y α es su parte derecha.

- P_A es, para cada $A \in N$, la distribución de probabilidad (sobre las reglas de R cuya parte izquierda es A) que representa las probabilidades de reescritura de A.

A la gramática G la denominaremos “gramática característica” de G_P, mientras que hablaremos de P como de “las probabilidades” del modelo.

Las anteriores definiciones nos permiten describir con cierta precisión cómo la GE incontextual (GEI) G_P genera secuencias finitas y no nulas de símbolos de Σ. El modelo parte, en $l = 1$, de una secuencia $\gamma_1 = S$ (su símbolo inicial) y va obteniendo, en etapas sucesivas $l \geq 2$, secuencias $\gamma_l \in (N \cup \Sigma)^+$ de la siguiente forma: primero, busca el no-terminal más a la izquierda en γ_{l-1} (sea A dicho no-terminal, de forma que $\gamma_{l-1} = \beta_1 A \beta_2$ con $\beta_1 \in \Sigma^*$ y $\beta_2 \in (N \cup \Sigma)^*$); después, extrae según la distribución P_A una regla $r_l = (A \to \alpha)$ de R; finalmente, la aplica para reescribir A de modo que $\gamma_l = \beta_1 \alpha \beta_2$. Si en una cierta etapa $L \geq 2$, γ_L no contiene ningún no-terminal, el modelo se detiene habiendo generado la secuencia $\gamma_L \in \Sigma^+$ (mediante la secuencia de reglas $r_{2 \ldots L}$, oculta para cualquier observador externo al modelo).

Dentro de la clase de las GEIs podemos señalar dos subclases interesantes, que se caracterizan por la forma de las reglas de R. Si $R \subseteq N \times (N^2 \cup \Sigma)$, entonces G (y, por tanto, G_P) está en forma normal de Chomsky (FNCh)[5]. Si $R \subseteq N \times ((\Sigma N) \cup \Sigma)$, entonces G es una gramática regular y G_P es una GE regular (GER).

[5]Para toda gramática incontextual, se puede construir una gramática en FNCh que genere el mismo lenguaje.

Análogamente a como sucedía con MOMs, necesitamos saber calcular la probabilidad de que una GEI $G_P = ((N, \Sigma, R, S), P)$ genere una cierta secuencia $y_{1\ldots T} \in \Sigma^+$. Llamando $G_P(y_{1\ldots T})$ a dicha probabilidad, podemos escribir

$$G_P(y_{1\ldots T}) = \sum_{r_{2\ldots L} \in R(y_{1\ldots T})} \left(\prod_{l=2}^{L} P_{r_l} \right),$$

donde con $R(y_{1\ldots T})$ denotamos el conjunto de todas las secuencias de reglas de R mediante las cuales una GEI puede generar $y_{1\ldots T}$ de la forma anteriormente descrita. Si G_P está en FNCh, el algoritmo "Inside" permite llevar a cabo el cálculo requerido con una complejidad temporal $O(T^3 \cdot |R|)$ [14], mientras que si se trata de un GER, entonces pueden utilizarse algoritmos análogos a los empleados en el caso de MOMs ("Forward" y "Backward") y que tienen una complejidad temporal $O(T \cdot |R|)$.

Diferentes técnicas de inferencia gramatical (Capítulo 1.5) pueden ser aplicadas para aprender inductivamente la gramática característica de una GEI. Respecto al aprendizaje inductivo de sus probabilidades, el algoritmo "Inside-Outside" es un algoritmo de reestimación aplicable si la GEI está en FNCh [61], mientras que las GERs admiten algoritmos análogos a los empleados con MOMs.

Podemos utilizar las GEs anteriormente presentadas, por ejemplo, como modelos de lenguaje en tareas de RPC (identificando el vocabulario de la tarea con el conjunto de terminales de la gramática característica) y, si el dominio **D** que utiliza el preproceso de nuestro sistema de RAH para representar fragmentos de la señal vocal es un conjunto finito de símbolos, también como modelos acústicos (identificando **D** con el conjunto de terminales). Si, por el contrario, nuestro preproceso produce secuencias de vectores de características (**D** es continuo), tendremos que complicar nuestro modelo asociando a cada terminal una distribución de probabilidad sobre **D** [62].

Finalmente, cabe comentar que es frecuente utilizar gramáticas no estocásticas como modelos de lenguaje [13]. Los modelos de este tipo suelen utilizarse en RAH para descartar aquellas hipótesis producidas por la etapa de decodificación lingüística que no se ajusten a las restricciones que imponen.

5 Aproximación Conexionista

En los últimos años, las *Redes Neuronales Artificiales* (RNAs) (o *Modelos Conexionistas*) se están convirtiendo en la base de potentes técnicas en reconocimiento de formas, especialmente en el campo del reconocimiento de imágenes y del habla. Las RNAs permiten construir modelos que pueden representar una función no lineal arbitraria; por otra parte, pueden modelar una tarea determinada adaptando los parámetros internos de la red mediante un proceso de *aprendizaje*[63]. Sin embargo, la propiedad de representación es muy diferente a la capacidad de aprender a partir de ejemplos, ya que, como en cualquier técnica de aproximación de funciones, la red se puede estancar en un mínimo local o, si la *topología* no es adecuada para la tarea que se desea abordar, puede generar un modelo inapropiado. Una introducción general a las RNAs se ha presentado en el Capítulo 6 del Tema 1, siendo el objetivo de este apartado mostrar una visión general de las RNAs aplicadas al Reconocimiento Automático del Habla.

5.1 Aplicación al Reconocimiento de Palabras Aisladas

Las distintas aproximaciones al Reconocimiento de Palabras Aisladas (RPA) se diferencian en la forma en que la red trata la variabilidad temporal del habla. En este punto, conviene distinguir dos categorías amplias de RNAs, atendiendo al patrón de interconexión de las unidades: *redes estáticas* y *redes dinámicas*. Las redes estáticas se caracterizan por ser sistemas "sin memoria", es decir, la salida de la red es función sólo de la entrada actual, no de pasadas o futuras entradas o salidas. Las entradas a este tipo de red son palabras completas. Las redes dinámicas, por su parte, son sistemas "con memoria" en las que se permite retroalimentación. En este caso, las neuronas de entrada a la red se deslizan por la palabra. Un ejemplo de cada una de estas dos arquitecturas se muestra en la figura 9.

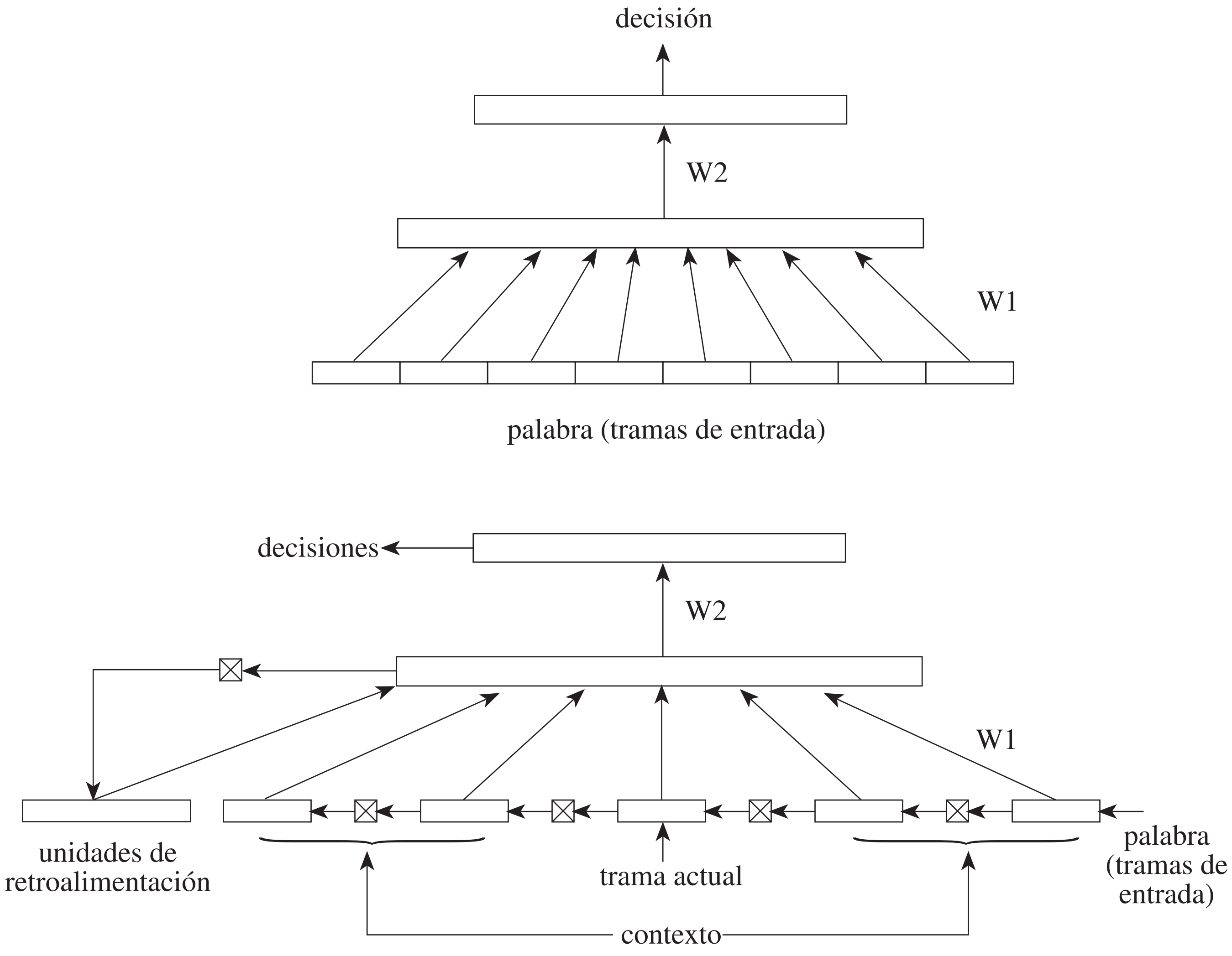

Figura 9: Un ejemplo de red estática y dinámica. La entrada de la red estática (Perceptrón Multicapa) está formada por la palabra completa. La red dinámica (red de Elman [64]) tiene como entrada una ventana contextual alrededor de la trama actual y retroalimentación de las activaciones de las unidades de la capa oculta.

Muchos modelos basados en RNAs estáticas se han utilizado para la clasificación de palabras aisladas. En este caso, la palabra de entrada se debe comprimir (o expandir) para obtener una longitud constante y, normalmente, hay una unidad de salida por palabra del léxico. Esta aproximación tiene el inconveniente de que es díficil realizar el alineamiento temporal necesario en la clasificación de señales acústicas, debido a que el patrón especial que representa a los elementos que componen la palabra son muy "elásticos" y un mismo elemento puede aparecer muy distorsionado aún en pronunciaciones de la misma palabra por un mismo locutor. Por lo tanto, para aprender que esos cambios en la posición no alteran la identidad del elemento se necesitan muchos datos de aprendizaje. Por otra parte, resulta complicado incorporar estas redes estáticas en reconocedores en tiempo real porque provocarían grandes retardos en el sistema, necesarios para aislar la palabra (delimitar su comienzo y su final) y almacenarla en un "buffer" de entrada a la red. Además, esta segmentación debería ser bastante exacta para que el sistema proporcionara buenas prestaciones.

En particular, el *Perceptrón Multicapa* ha sido quizás la arquitectura más utilizada para la clasificación de palabras aisladas [65] [66] [15] debido a que permite delimitar las regiones de decisión de un espacio de muestras arbitrariamente complejas. Normalmente, la capa de salida se define con un número de unidades igual al tamaño del léxico. El número de unidades en la capa de entrada debe ser suficiente, en la implementación más sencilla, para albergar el dato más largo (cuando aparece una pronunciación de menor longitud, las unidades de entrada no ocupadas se rellenan con *silencio*). Esta aproximación no lleva asociado un adecuado modelado de la estructura temporal del habla, y el coste computacional para el aprendizaje y el reconocimiento está sobredimensionado. Una alternativa para solucionar este inconveniente es escalar las pronunciaciones de la muestra de aprendizaje a una longitud dada, utilizando técnicas de preprocesado como Alineamiento Temporal Lineal o Segmentación de Traza [67]. Esta segunda alternativa tiene en cuenta la naturaleza no lineal de las variaciones espectrales de la señal acústica. Por último, se debe decidir el número de unidades de la(s) capa(s) oculta(s) y así, una vez fijada la topología de la red, se procede al entrenamiento (ajuste de los pesos) utilizando el algoritmo de aprendizaje supervisado denominado *Retropropagación del Error* ("Back Error Propagation"). En la fase de reconocimiento, la pronunciación se clasifica en la clase cuya salida correspondiente (unidad de salida asociada) tenga un valor máximo.

Para intentar superar los problemas derivados de los reconocedores estáticos, se han utilizado RNAs que incorporan retardos (como es el caso de la llamada red "NETtalk" [68] y la "Time Delay Neural Network" [69], integración temporal (sistemas híbridos) o conexiones recurrentes. La entrada a estas redes es secuencial, típicamente una trama de voz por cada unidad de tiempo. Estas redes, por lo tanto, se pueden integrar en aplicaciones en tiempo real más fácilmente que las redes estáticas, debido a que no se necesita segmentar la señal y sólo se utilizan retardos de poca duración.

Las redes recurrentes añaden conexiones que introducen activaciones anteriores en el cálculo de nuevos eventos [64] [70], lo que incorpora "memoria" en la red y, por tanto, la red es capaz de clasificar patrones dinámicos. Una dificultad de estas redes recurrentes es el desarrollo de algoritmos de aprendizaje eficientes, debido a que son inherentemente recursivos. La gran mayoría de algoritmos de aprendizaje son técnicas de descenso por gradiente similares al algoritmo de Retropropagación del Error, pero con un cálculo recursivo del gradiente, lo que los hace más complejos.

Finalmente, cabe mencionar que todos los modelos de RNAs que se han presentado en este apartado utilizan algoritmos de aprendizaje *supervisado*, es decir, la muestra de entrenamiento está formada por pares entrada/salida, donde la entrada es la muestra de voz (pronunciación) y la salida la clase a la que pertenece. Un modelo conexionista que también se ha utilizado para el RPA son las redes auto-organizativas, que utilizan algoritmos de aprendizaje *no-supervisados*, que permiten el agrupamiento de las muestras en clases similares. El ejemplo más conocido de estas redes son los mapas topológicos de Kohonen [71].

5.2 Aplicación al Reconocimiento de Palabras Conectadas

Para el reconocimiento de palabras conectadas con grandes diccionarios, donde no hay delimitadores de palabras y el tamaño del vocabulario no permite modelar palabras completas, se necesita construir modelos subléxicos. Desde este punto de vista, el problema consiste en reconocer secuencias de unidades subléxicas y ser capaz de concatenarlas para construir secuencias de palabras restringidas por una gramática. Las RNAs son capaces de realizar una clasificación subléxica (fonética) adecuada (clasificación a nivel de trama de voz) debido a su gran capacidad discriminativa, pero no existe ninguna arquitectura ni algoritmos que modelen adecuadamente la estructura temporal del habla. Se necesita un postproceso que interprete la secuencia de salida para generar una secuencia de palabras (o una secuencia de fonemas). Se han realizado varios intentos para interpretar la evolución temporal de las salidas de las RNAs utilizando técnicas de Programación Dinámica (por ejemplo,[72][73]). En una línea similar, los investigadores han intentado combinar el poder de clasificación de las RNAs con la capacidad de los Modelos Ocultos de Markov para modelar la estructura temporal del habla (por ejemplo, [74] [75]). Dentro de esta aproximación estadística, los métodos conexionistas se están empleando para mejorar las prestaciones de los actuales sistemas de RAH, utilizando Perceptrones Multicapa como estimadores de las probabilidades de emisión de los MOMs. Por otra parte, también sirven para modelar el contexto, usando como entrada a un Perceptrón Multicapa una ventana contextual o utilizando redes recurrentes.

Para concluir este apartado, cabe mencionar que se espera que esta metodología, todavía emergente, nos conduzca a nuevos paradigmas en el campo del RAH debido a su capacidad de aprendizaje, adaptibilidad y habilidad para procesar información ambigua. Sin embargo, el procesamiento de secuencias temporales sigue siendo un tema que deberá resolverse para la aplicación práctica de las RNAs en esta área. Se han propuesto algunas arquitecturas recurrentes que tratan de realizar alineamiento temporal, pero sólo se han experimentado en tareas pequeñas, debido a que los algoritmos de aprendizaje son extremadamente lentos, lo que los hace impracticables en tareas reales de RAH. En cambio, para el RPA, con pequeños diccionarios (difíciles), las RNAs han demostrado ser muy efectivas debido a su poder discriminativo, superando las prestaciones de ofrecen otras técnicas más tradicionales. El reconocimiento del habla continua, sin embargo, no se puede abordar actualmente sólo con métodos conexionistas, por lo que se deben combinar con un procedimiento de alineamiento, generalmente basado en técnicas de Programación Dinámica. Así, el alineamiento temporal se realiza *externamente* a la RNA [74] [75] o *internamente* [76]. Estos sistemas "híbridos" están suponiendo grandes mejoras en los actuales sistemas de reconocimiento, sobre todo, en cuanto a la integración "estadística" de los métodos conexionistas y los modelos estructurales (Modelos Ocultos de Markov).

Referencias

[1] H. Sakoe y S. Chiba. Dynamic Programming Algorithm Optimization for Spoken Words Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26:43–49, 1978.

[2] L. Rabiner y S. Levinson. Isolated and Connected Word Recognition—Theory and Selected Applications. *IEEE Transactions on Communications*, COM–29(5):621–659, 1981.

[3] W. L. Roch. Voice Recognition: Understanding the Master's Voice. *PC Magazine*, pp. 261–308, Octubre 1987.

[4] F. Casacuberta y E. Vidal. *Reconocimiento Automático del Habla*. Marcombo, 1987.

[5] R. D. Peacocke y D. H. Graf. An Introduction to Speech and Speaker Recognition. *Computer*, pp. 26–33, Agosto 1990.

[6] D. H. Klatt. Review of the ARPA Speech Understanding Project. *Journal of the Acoustics Society of America*, 62:1345–1366, 1977.

[7] W. A. Lea, editor. *Trends in Speech Recognition*. Prentice-Hall, 1980.

[8] J. P. Haton. Knowledge-Based Approaches in Acoustic-Phonetic Decoding of Speech. En Niemann, editor, *NATO-ASI*, volumen F46, pp. 51–69. Springer Verlag, 1988.

[9] L. R. Bahl, F. Jelinek, y R. L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190, 1983.

[10] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

[11] L. R. Rabiner, S. E. Levinson, y M. M. Sondhi. On the application of Vector Quantization and Hidden Markov Models to Speaker-Independent Isolated Word Recognition. *The Bell Systems Technical Journal*, 62(4):1075–1105, 1983.

[12] F. Jelinek. Continuous Speech Recognition by Statistical Methods. *Proceedings of the IEEE*, 64(4):532–556, 1976.

[13] S. E. Levinson. Structural Methods in Automatic Speech Recognition. *Proceedings of the IEEE*, 73(11):257–286, 1985.

[14] F. R. Jelinek, J. D. Lafferty, y R. L. Mercer. Basic Methods of Probabilistic Context-Free Grammars. En P. Laface y R. De Mori, editores, *Speech Recognition and Understanding*, pp. 345–360. Springer Verlag, 1992.

[15] R. P. Lippmann. Review of Neural Networks for Speech Recognition. *Neural Computation*, 1:1–38, 1989.

[16] H. Sakoe. Two-Level DP-Matching—A Dynamic Programming-Based pattern Matching Algorithm for Connected Word Reconition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(6):588–595, 1979.

[17] H. Ney. The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):263–271, 1984.

[18] P. Aibar, A. Marzal, E. Vidal, y F. Casacuberta. Problems and Algorithms in Optimal Linguistic Decoding: A Unified Formulation. En *Proceedings of the International Conference on Spoken Language Processing*, pp. 1625–1628, 1992.

[19] C. S. Myers y L. R. Rabiner. A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):284–297, 1981.

[20] C. Godin y P. Lockwood. DTW Schemes for Continuous Speech Recognition: A Unified View. *Computer Speech and Language*, 3, 1989.

[21] R. Davis y D. B. Lenat. *Knoledge-Based Systems in Artificial Intelligence*. McGraw-Hill, 1982.

[22] L. Erman, F. Hayes-Roth, V. Lesser, y D. Reddy. The HEARSAY-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainly. *Computing Surveys*, 12(2):213–253, 1980.

[23] J. Wolf y W. A. Wodds. The HWIM Speech Understanding System. En W. A. Lea, editor, *Trends in Speech Recognition*, pp. 316–339. Springer Verlag, 1980.

[24] H. Niemann et al. The Speech Understanding and Dialog System EVAR. En R. De Mori y C. Y. Suen, editores, *NATO-ASI Series*, pp. 271–302. Springer Verlag, 1985.

[25] G. Perennou. The ARIAL-II Speech Recognition System. En J. P. Haton, editor, *Automatic Speech Analysis and Recognition*. Reidel Pu. Co., 1982.

[26] A. Averbuch et al. Experiments with the Tangora 20.000 Word Speech Recognizer. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 701–704, 1987.

[27] R. De Mori et al. Integration of Acoustic, Phonetic, Prosodic and Lexical Knowledge in an Expert System for Speech Understanding. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1984.

[28] J. P. Haton. Knowledge-Based and Expert Systems in Automatic Speech Recognition. En R. De Mori y C. Y. Suen, editores, *NATO-ASI Series*. Springer Verlag, 1985.

[29] G. Mercier et al. A New Rule-Based Expert System for Speech Recognition. En R. De Mori y C. Y. Suen, editores, *NATO-ASI Series*, pp. 303–342. Springer Verlag, 1985.

[30] V. Zue y L. F. Lamel. An Expert Spectrogram Reader: A Knowledge-Based Approach to Speech Recognition. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 23–27, 1986.

[31] P. Alinat. Robust analitic speech recognition system. (esprit speech day). En P. VanHove, editor, *Proceedings of the EUROSPEECH*, 1991.

[32] J. M. Benedí, M. J. Castro, y J. A. Gómez. Acoustic-Phonetic Decoding in Spanish ROARS System. Informe técnico, ESPRIT-II ROARS, 1993.

[33] N. Carbonell et al. Acquisition et formalisation du raisonnement dans un systeme expert du lecture de spectrogrammes vocaux. En *Colloque ARC*, 1984.

[34] G. Sagerer y F. Kummert. Knowledge Based Systems for Speech Understanding. En Niemann, editor, *NATO-ASI*, volumen F46, pp. 421–458. Springer Verlag, 1988.

[35] W. A. Woods. Optimal Search Strategies for Speech Understanding Control. *Artificial Intelligence*, pp. 295–326, 1982.

[36] P. Gonzalez. *Technology Profile in Speech Recognition*. SRI International, 1992.

[37] P. Alinat *et al.* A Continous Speech Dialog System for Oral Control of a Sonar Console. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 368–371, 1987.

[38] L. R. Rabiner, B. H. Juang, S. E. Levinson, y M. M. Sondhi. Recognition of Isolated Digits using Hidden Markov Models with Continuous Mixture Densities. *AT&T Technical Journal*, 64:1211–1234, 1985.

[39] L. R. Bahl, P. F. Brown, P. V. de Souza, y R. L. Mercer. Speech recognition with continuous-parameter hidden Markov models. *Computer Speech and Language*, 2:219–234, 1987.

[40] K. F. Lee. *Automatic Speech Recognition: The Development of the SPHINX System*. Kluwer Academic, 1989.

[41] M. J. Russell y R. K. Moore. Explicit Modelling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 5–8, 1985.

[42] M. J. Russell y A. E. Cook. Experimental Evaluation of Duration Modelling Techniques for Automatic Speech Recognition. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 2376–2379, 1987.

[43] Y. Guédon y C. Cocozza-Thivent. Explicit State Occupancy Modelling by Hidden Semi-Markov Models: Application of Derin's Scheme. *Computer Speech and Language*, 4:167–192, 1990.

[44] Y. Guédon. Review of several stochastic speech unit models. *Computer Speech and Language*, 6:377–402, 1992.

[45] F. Casacuberta, E. Vidal, B. Mas, y H. Rulot. Learning the structure of HMM's through grammatical inference techniques. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 717–720, 1990.

[46] D. Jouvet, L. Mauuary, y J. Monné. Automatic adjustments of the structure of Markov models for speech recognition applications. En *Proceedings of the EUROSPEECH*, pp. 927–930, 1991.

[47] A. Stolcke y S. M. Omohundro. Best-First Model Merging for Hidden Markov Model Induction. Technical Report TR-94-003, International Computer Science Institute, California, 1994.

[48] L. R. Bahl, P. F. Brown, P. V. de Souza, y R. L. Mercer. Estimating hidden Markov model parameters so as to maximize speech recognition accuracy. *IEEE Transactions on Speech and Audio Processing*, 1(1):77–83, 1993.

[49] F. Prat y F. Casacuberta. On the estimation of the probabilistic-distribution functions of stochastic structural models by maximizing merit functions. En H. Bunke, editor, *Advances in structural and syntactic pattern recognition*, pp. 121–130. World Scientific, 1992.

[50] L. E. Baum. An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes. En Q. Shisha, editor, *Inequalities, III*, pp. 1–8. Academic Press, 1971.

[51] L. A. Liporace. Maximum Likelihood Estimation for Multivariate Observations of Markov Sources. *IEEE Transactions on Information Theory*, 28(5):729–734, 1982.

[52] B. H. Juang. Maximum-Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains. *AT&T Technical Journal*, 64(6):1235–1249, 1985.

[53] P. A. Devijver. Baum's Forward-Backward Algorithm Revisited. *Pattern Recognition Letters*, 3:369–373, 1985.

[54] A. P. Dempster, N. M. Laird, y D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39 (ser. B)(1):1–38, 1977.

[55] C. F. J. Wu. On the Convergence Properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

[56] L. R. Bahl, P. F. Brown, P. V. de Souza, y R. L. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 49–52, 1986.

[57] Y. Normandin y S. D. Morgera. An improved MMIE training algorithm for speaker-independent, small vocabulary, continuous speech recognition. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 537–540, 1991.

[58] F. Prat y F. Casacuberta. Adaptación de los modelos de Markov ocultos a la generación de secuencias finitas en reconocimiento automático del habla. En *Libro de Comunicaciones del V Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes*, pp. 204–211, 1992.

[59] N. Merhav y Y. Ephraim. Hidden Markov Modelling using a Dominat State Sequence with Application to Speech Recognition. *Computer Speech and Language*, 5:327–339, 1991.

[60] G. D. Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, 61(3):268–728, 1973.

[61] K. Lari y S. J. Young. The Estimation of Stochastic Context-Free Grammars using the Inside-Outside Algorithm. *Computer Speech and Language*, 4:35–56, 1990.

[62] H. Ney. Stochastic Grammars and Pattern Recognition. En *Speech Recognition and Understanding*, pp. 319–344. Springer Verlag, 1992.

[63] D. R. Hush y B. G. Horne. Progress in supervised neural networks. *IEEE Signal Processing Magazine*, pp. 8–38, 1993.

[64] J. L. Elman. Finding Structure in Time. Crl tech. report 8801, University of California, 1988.

[65] D. J. Burr. Experiments on Neural Net Recognition of Spoken and Written Text. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1162–1168, 1988.

[66] S. M. Peeling y R. K. Moore. Isolated Digit Recognition Experiments using the Multi-Layer Perceptron. *Speech Communication*, 7:403–409, 1988.

[67] M. J. Castro, F. Casacuberta, y C. Puchol. Trace Segmentation with Artificial Neural Networks. DSIC-II/13/91, Dept. Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, 1991.

[68] T. J. Sejnowski y C. R. Rosenberg. NETtalk: A parallel network that learns to read aloud. Technical Report JHU/EECS-86/01, Johns Hopkins University, Baltimore, 1986.

[69] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, y K. J. Lang. Phoneme Recognition Using Time-Delay Neural Networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339, 1989.

[70] M. Jordan. Serial Order: A Parallel Distributed Processing Approach. En J. L. Elman y D. E. Rumelhart, editores, *Advances in Connectionist Theory: Speech*. Hillsade, Erlbaum, 1989.

[71] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, segunda edición, 1988.

[72] T. Robinson y F. Fallside. Phoneme Recognition from the TIMIT Database using Recurrent Error Propagation Networks. Technical Report No. CUED/F-INFENG/TR 42, Engineering Dept., Cambridge University, 1990.

[73] P. Haffner, M. Franzini, y A. Waibel. Integrating time alignment and neural networks for high performance continuous speech recognition. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1991.

[74] M. Franzini, K. F. Lee, y A. Waibel. Connectionist Viterbi training: a new hibrid method for high performance continuous speech recognition. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1990.

[75] N. Morgan, H. Bourlard, S. Renals, M. Cohen, y H. Franco. Hybrid Neural Network/Hidden Markov Model Systems for Continuous Speech Recognition. *Internation Journal of Pattern Recognition and Artificial Intelligence*, 7(4):899–916, 1993.

[76] H. Sakoe, R. Isotani, K. Yoshida, K. Iso, y T. Watanabe. Speaker-Independent Word Recognition using Dynamic Programming Neural Networks. En *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 29–32, 1989.