6 - RECONOCIMIENTO Y SÍNTESIS DEL HABLA

ENTRENAMIENTO DISCRIMINATIVO PARA HMM UTILIZANDO REDES NEURONALES RECURRENTES

J.E. Díaz Verdejo
J.C. Segura Luna
A. Peinado Herreros
A. Rubio Ayuso
J.A. Gálvez López

Dept. Electrónica y Tecnología de Computadores, Univ. Granada
18071 Granada (Spain)
Tlf. 34-58-243283 Fax. 34-58-243230

Resumen

En el presente articulo se presentan los resultados obtenidos a partir de una estructura en red para los Modelos Ocultos de Markov, aplicados al reconocimiento del habla. La topología de la red es la de una Red Neuronal Recurrente, en la que cada iteracion temporal es identificada con una capa. El entrenamiento de dicha red se realiza mediante técnicas de retropropagación. Dos tipos de medidas de error se utilizan para el entrenamiento: máxima semejanza y entrenamiento discriminativo. La aplicación de las técnicas de retropropagación para la reestimación de los HMM-RNN en el caso de entrenamiento por máxima semejanza proporciona las mismas ecuaciones de reestimación que el algoritmo de Baum-Welch utilizado para entrenar los HMM. El entrenamiento discriminativo se basa en la probabilidad de clasificación correcta de las secuencias a partir de la medida de máxima semejanza. Los resultados obtenidos han demostrado que el mejor procedimiento para entrenar los RNN-HMM consiste en realizar una primera estimación mediante la medida de máxima semejanza para, posteriormente, reentrenarlos mediante el algoritmo de entrenamiento discriminativo.

1 Introducción

Las redes neuronales están siendo aplicadas al reconocimiento automático del habla en los últimos años. Sin embargo, aunque constituyen un potente marco de trabajo, adolecen de un importante problema para modelar patrones secuenciales.

La restricción impuesta por las redes neuronales sobre la estaticidad de los patrones origina la necesidad de fijar el número de elementos de cada uno de los mismos. En el caso del reconocimiento de voz, el número fijo de elementos se traduce en una igual duración para cada uno de los patrones.

Se plantean, por tanto, dos importantes problemas para aplicar las redes neuronales al reconocimiento automático del habla: el modelado de la secuencialidad, y la fijación de la duración de las unidades de reconocimiento.

Algunas de las soluciones propuestas fijan el número de entradas de la red, con lo que es necesario realizar algún tipo de alineamiento temporal, usualmente no lineal, con el fin de fijar la duración de las secuencias. En este contexto se sitúan los Perceptrones Multicapa [2] y las Redes de Programación Dinámica [3].

Se han propuesto también soluciones intermedias que, a pesar de utilizar redes con un número fijo de entradas, intentan modelar la secuencialidad de la señal de voz mediante el uso de unidades de descripción de tamaño relativamente pequeño y el establecimiento de elementos capaces de aportar información sobre el contexto. Tal es el caso de las Redes de

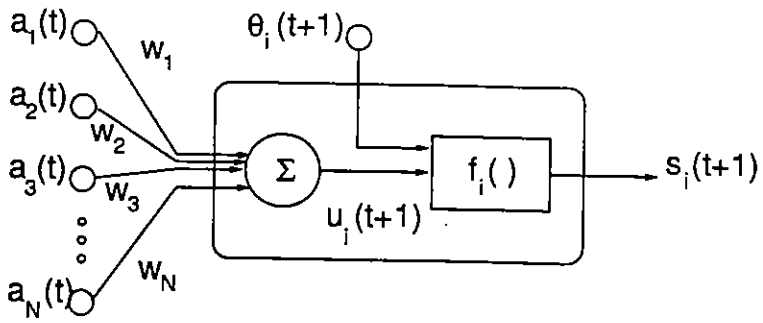

Figura 1: Esquema de una neurona

Retardo Temporal [1]. Sin embargo, aunque estas redes procesan de forma adecuada las propiedades dinámicas locales, no pueden hacerlo con secuencias.

La aproximación propuesta en el presente trabajo utiliza un tipo de red denominado Red Neuronal Recurrente, caracterizado por realizar un procesamiento cíclico, pudiéndose introducir datos de entrada en cada uno de los ciclos. De esa forma se resuelven los problemas de alineamiento temporal, que ya no es necesario, y se modela correctamente la secuencialidad, desapareciendo, como consecuencia, las restricciones sobre el tipo de unidades.

La aproximación que proporciona mejores resultados en la actualidad para el reconocimiento de voz se basa en modelos estocásticos, que presentan una gran potencia y unas bases perfectamente establecidas, y que son capaces de procesar la secuencialidad de la voz. Estos sistemas son los denominados Modelos Ocultos de Markov. En ellos cada patrón de voz es tratado como si fuese la salida de un sistema estocástico con estados internos cuya evolución se rige por leyes probabilísticas. La clasificación de una secuencia se realiza atendiendo a una medida de la similitud entre dicha secuencia y los modelos de secuencia previamente entrenados.

Dados los resultados obtenidos mediante modelos ocultos de Markov cabe cuestionarse acerca de la conveniencia de la utilización de redes neuronales para el reconocimiento del habla. El interés de dichas redes reside en la naturaleza discriminativa de los métodos de entrenamiento, la posibilidad de uso de estructuras no lineales más generales que las implementables según el modelo estocástico, y el uso de un formalismo más intuitivo [4]. Sin embargo, existe un argumento más interesante: los modelos ocultos de Markov presentan una fuerte relación con las redes neuronales recurrentes, pudiendo considerarse como un caso particular de dichas redes.

En los últimos años, algunos esquemas de entrenamiento de tipo discriminativo han sido implementados a partir de las técnicas utilizadas en los modelos ocultos de Markov, basados en nuevas medidas como son el MMI y el MDI [6]. Sin embargo, es en el contexto de redes neuronales donde dichos algoritmos pueden ser desarrollados de forma natural y sencilla. En el presente trabajo se utilizará una función de error, propuesta por Bridle [4] equivalente al MMI, para la que se derivará un algoritmo de entrenamiento utilizando técnicas de retropropagación.

2 Redes Neuronales Recurrentes

Una Red Neuronal consta de una numero grande de unidades elementales llamadas *neuronas*. Estas neuronas se encuentran conectadas entre sí a través de conexiones denominadas *conexiones sinápticas*, cada una de las cuales tiene asociado un *peso*.

La neurona es la unidad elemental de proceso, realizando una suma pesada de las entradas (figura 1), y obteniendo una señal denominada *entrada de la red*. La función de activación f determina la salida de la neurona, llamada *valor de activación*, a, a partir de la entrada de la red y de un *estímulo externo*, θ. Las neuronas se asocian en *capas*. Las entradas de una neurona dada

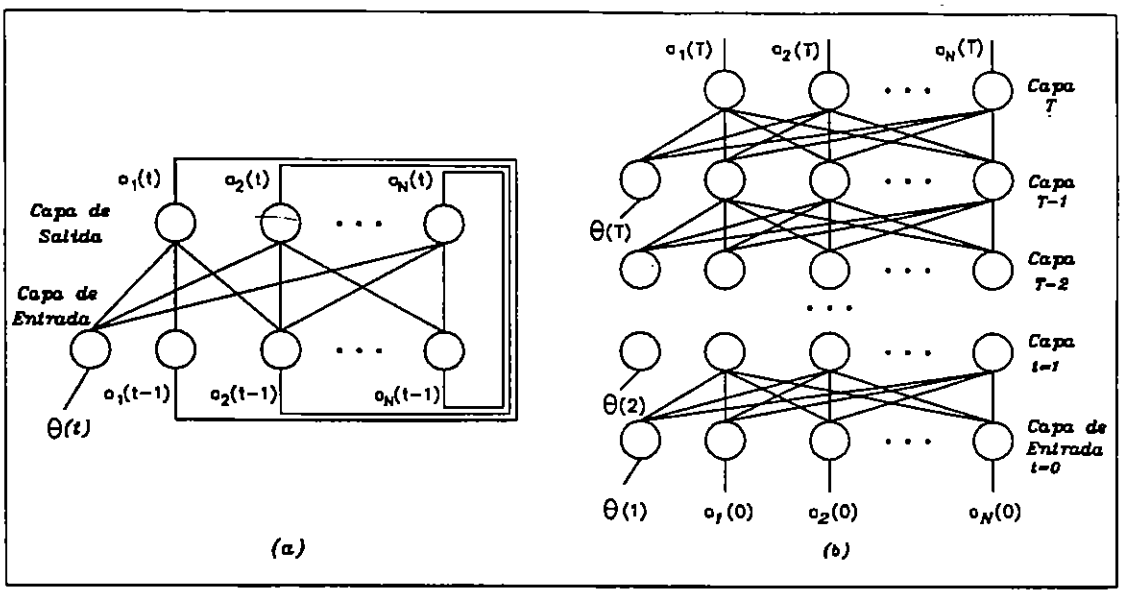

Figura 2: a) Esquema de una RNN. b) Red equivalente para T estímulos externos

provendrán de las capas inferiores, mientras que las salidas de todas las neuronas de una capa constituirán las entradas de las capas siguientes.

En la capa $t+1$, la neurona i realizará las operaciones

$$\begin{aligned} u_i(t+1) &= \sum_{j=1}^{N} w_{ij}(t+1) \cdot a_j(t) \\ a_i(t+1) &= f_i(\theta_i(t+1), u_i(t+1)) \end{aligned} \tag{1}$$

siendo N_t el número de neuronas de la capa t que están conectadas con la neurona considerada y w_{ij} el peso asociado a dicha conexión entre las neuronas i y j.

Una Red Neuronal Recurrente (RNN) es una Red Neuronal de una única capa con las salidas conectadas a las entradas (figura 2). Si se presenta una secuencia de T estímulos externos θ a una RNN, ésta puede ser considerada como una Red Neuronal compuesta por la replicación T veces de la única capa que constituye la RNN. La RNN presenta dos tipos de entradas: los estímulos iniciales $a_i(0)$ y los estímulos externos $\theta_i(t)$. Los estímulos externos constituyen la secuencia a clasificar, mientras que los estímulos iniciales inicializan la dinámica de la red.

La reestimación de los parámetros de la red se puede realizar mediante las técnicas de retropropagación, a partir de las respuestas obtenidas para las diferentes secuencias que constituyen el conjunto de entrenamiento. El algoritmo de retropropagación permite el cálculo del error cometido sobre la estimación de uno de los parámetros a partir de los errores cometidos en la capa superior.

La señal de retropropagación $\delta_i(t)$ se define [4]

$$\delta_i(t) \equiv \frac{\partial \epsilon}{\partial a_i(t)} \tag{2}$$

La actualización de los parámetros de la red se puede realizar recursivamente a partir de la señal de retropropagación

$$p'_{ij}(t) = p_{ij} \oplus \eta_j \sum_{t=1}^{T} \delta_i(t) \frac{\partial a_i(t)}{\partial w_{ij}(t)} \tag{3}$$

donde $\oplus$ representa cualquiera de las cuatro operaciones aritméticas básicas, y p es cualquier parámetro de la red.

3 RNN y HMM

En el presente trabajo sólo se considera un HMM de *izquiera-a-derecha*, por lo que las probabilidades de los estados iniciales se eligen de forma que el primer estado del modelo sea el estado inicial.

Dada una secuencia de entrada $O_1, O_2, ..., O_N$, la fase de evaluación de un HMM consistirá en la obtención de la probabilidad de que dicha secuencia haya sido producida por el modelo considerado, λ. Esta probabilidad puede ser obtenida de forma recursiva definiendo la *probabilidad hacia adelante* [8]

$$a_i(1) = \pi_i \cdot b_i(O_1) \tag{4}$$

$$a_i(t) = \left[\sum_{j=1}^{N} a_j(t-1) \cdot w_{ij}\right] \cdot b_i(O_t) \tag{5}$$

Estas ecuaciones son equivalentes a las obtenidas para las RNN si se identifica π_i, las probabilidades de los estados iniciales, con las entradas iniciales, y se elige como función de activación f el producto de la entrada de la red, $u_i(t)$ con el estímulo externo ($b_i(O_t)$, las pdf's):

$$u_i(t) = \sum_{j=1}^{N} w_{ij} \cdot a_j(t-1) \tag{6}$$

$$a_i(t) = u_i(t) \cdot b_i(O_t) \tag{7}$$

Se puede utilizar un criterio de máxima semejanza para entrenar los modelos. Las ecuaciones de reestimación obtenidas a partir del algoritmo de retropropagación son idénticas a las ecuaciones de Baum-Welch [8] si se selecciona una actualización multiplicativa.

4 Entrenamiento discriminativo

El algoritmo de entrenamiento de Baum-Welch no es un entrenamiento discriminativo. Unicamente intenta maximizar la probabilidad del modelo correcto durante la fase de entrenamiento. Por lo tanto, durante la fase de evaluación, una secuencia será clasificada como correspondiente a la clase del modelo que proporcione una probabilidad de generación $\mathcal{L}(O|\lambda)$ mayor.

Debido a la naturaleza discriminativa de las redes neuronales, pueden implementarse fácilmente nuevas estimaciones que sigan un esquema discriminativo.

A efectos de clasificación, puede definirse una nueva probabilidad normalizando las probabilidades proporcionadas por los modelos para una secuencia:

$$P(O|\lambda) \equiv \frac{\mathcal{L}(O|\lambda)}{\sum_{\lambda'} \mathcal{L}(O|\lambda')} \tag{8}$$

Esta probabilidad corresponde a la distribución de probabilidad de los modelos para una secuencia dada.

La medida propuesta [7] es:

$$\mathcal{J} \equiv -\log(P(O|\lambda_c)) \tag{9}$$

siendo λ_c el modelo correcto asociado a la secuencia O. De esta forma, $P(O|\lambda_c)$ es la probabilidad de acertar la clase correcta para la secuencia eligiéndola a partir de la distribución de probabilidad definida.

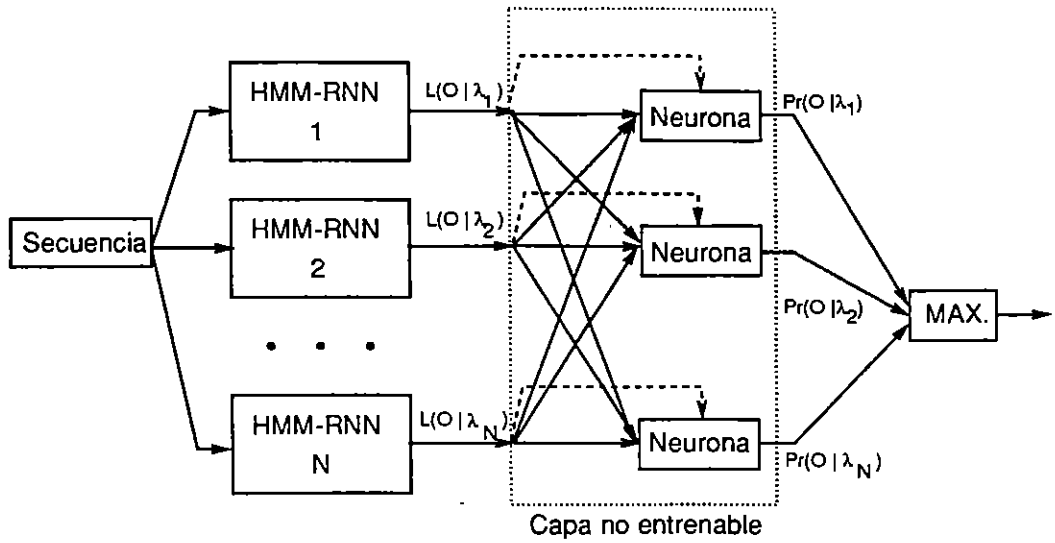

Figura 3: Esquema de la red con la capa añadida para el entrenamiento discriminativo

La única modificación de la topología de la red es la adición de una capa que calcule las probabilidades (figura 3). Esta capa no es entrenable, ya que sus pesos tienen valor fijo 1. La función de activación es la división del estímulo externo por la entrada de la red.

Las ecuaciones obtenidas para la actualización de los parámetros de la red, según una implementación aditiva son:

$$w'^{k}_{ij} = w^{k}_{ij} + \eta^{k} \frac{\partial \mathcal{J}}{\partial w^{k}_{ij}} = w^{k}_{ij} + \eta^{k} \frac{P(O|\lambda_c) - \delta_{\lambda,\lambda_c}}{\mathcal{L}(O|\lambda_c)} \sum_{t=1}^{T-1} \delta^{k}_{i}(t+1) \cdot b^{k}_{i}(O_{t+1}) \cdot a^{k}_{j}(t+1) \quad (10)$$

$$b'^{k}_{i}(v_l) = b^{k}_{i}(v_l) + \eta^{k} \frac{\partial \mathcal{J}}{\partial b^{k}_{i}(l)} = b^{k}_{i}(v_l) + \frac{\eta^{k}}{b^{k}_{i}(v_l)} \cdot \frac{P(O|\lambda_c) - \delta_{\lambda,\lambda_c}}{\mathcal{L}(O|\lambda_c)} \sum_{t=1}^{T} \delta^{k}_{i}(t) \cdot a^{k}_{i}(t) \cdot \delta_{kr}(O_t, v_l) \quad (11)$$

donde el superíndice k corresponde al modelo siendo actualizado.

Es importante reseñar en este punto algunos comentarios acerca de la elección de una implementación aditiva para las ecuaciones de reestimación. Las devidadas $\frac{\partial \mathcal{J}}{w_{ij}}$ y $\frac{\partial \mathcal{J}}{b_i(v_l)}$ pueden ser negativas, y, de hecho, lo son cuando el modelo que está siendo actualizado es el correcto. Estos valores negativos pueden originar problemas de convergencia en el caso seleccionar una reestimación multiplicativa.

Para las simulaciones mediante *software* de las RNN-HMM es necesario considerar 3 cuestiones: el entrenamiento para múltiples secuencias, el escalado para evitar problemas de desbordamiento, y la posibilidad de asignación de valores nulos a probabilidades debido a que el conjunto de entrenamiento es finito.

5 Resultados experimentales

Los datos considerados han sido muestreados a 8 kHz, y preenfatizados con un factor de preénfasis $\mu = 0.95$. A la señal obtenida se le aplicaron ventanas de Hamming de 32 ms, con un desplazamiento de 8 ms para cada una de ellas. Cada segmento así obtenido se caracterizó mediante 16 coeficientes cepstrales y 16 coeficientes de la derivada del cepstrum obtenidos mediante regresión lineal sobre los 7 segmentos más próximos. Finalmente, los vectores de

	In. Lin.		MS		ED		MS + ED	
N	Ent.	Test	Ent.	Test	Ent.	Test	Ent.	Test
6	8.91	11.72	2.50	5.31	1.95	8.44	0.70	4.84
7	9.84	11.72	2.11	4.84	2.58	8.59	0.47	4.22
8	9.92	12.34	1.80	4.22	2.19	9.06	0.70	4.06
9	9.53	12.97	1.56	4.53	2.42	9.06	0.55	4.06
10	9.77	14.22	1.09	3.91	2.66	9.69	0.39	3.12

Tabla 1: Tasas de error sobre las secuencias de entrenamiento (Ent.) y evaluación (Test) para los modelos iniciales (In.Lin.) y los tres métodos probados.

parámetros obtenidos se promediaron para cada dos segmentos consecutivos y se decimaron a 16 mseg.

Las palabras utilizadas fueron codificadas mediante un diccionario de 64 centros en todas los experimentos considerados, utilizando una distancia pesada [9]. Se utilizó una red para cada palabra, con un número de neuronas (estados) entre 2 y 12.

El vocabulario seleccionado consta de los dígitos castellanos y las 6 palabras CUERPO, HOMBRO, CODO, MUÑECA, MANO, DEDOS pensadas para el control de los 6 micromotores de un robot.

La base de datos consta de 40 locutores (20 masculinos y 20 femeninos), cada uno de los cuales pronunció 3 repeticiones de cada una de las palabras (1920 palabras en total). La SNR de las grabaciones es de 24 dB.

Los experimentos considerados se realizaron en modo multilocutor, utilizando 2 de las repeticiones de cada una de las palabras para el entrenamiento del sistema, y la tercera para la evaluación del mismo.

Se realizaron experimentos preliminares para seleccionar el valor mínimo a asignar a las probabilidades de producción de símbolos $b_i(v_l)$ y el número de neuronas por modelo. Para estos experimentos sólo se utilizó el entrenamiento por máxima semejanza.

El valor seleccionado para el mínimo de la probabilidad de producción de símbolos es 10^{-4}. El número de neuronas por modelo utilizado en los experimentos discriminativos se eligió entre 6 y 10, debido al gran incremento en la complejidad computacional con el número de estados considerado.

Se desarrollaron tres experimentos: estimación por máxima semejanza (MS), entrenamiento discriminativo (ED) y entrenamiento por máxima semejanza más entrenamiento discriminativo (MS+ED).

Los resultados obtenidos se muestran en la tabla 1. La inicialización de los modelos se ha realizado mediante segmentación lineal de las secuencias de entrenamiento en el número de estados considerado.

El número de iteraciones consideradas para el entrenamiento discriminativo fue 20, mientras que en el caso de entrenamiento por máxima semejanza y discriminativo se consideraron 15 iteraciones.

Puede observarse que el procedimiento más efectivo de entrenamiento es la conjunción de ambos de forma que en primer lugar se realiza una estimación de los parámetros de la RNN mediante el algoritmo de máxima semejanza. Posteriormente, se reestiman dichos parámetros mediante un número reducido de iteraciones utilizando el algortimo discriminativo. De esta forma se puede reducir en gran medida el costo computacional del entrenamiento, dado que el algoritmo MS presenta una convergencia más rápida que el algoritmo EC.

La tabla 1 muestra una gran reducción en las tasas de error en entrenamiento cuando se utiliza un esquema de entrenamiento discriminativo. Esto es debido al hecho de que este tipo de entrenamiento intenta separar las clases correspondientes a cada modelo. De todas formas, las diferencias para el experimento MS y el experimento MS+EC son pequeñas para las secuencias de evaluación del sistema. La figura 4 muestra la evolución de las tasas de error en función del número de iteraciones para los experimentos MS y MS+EC. En ambos

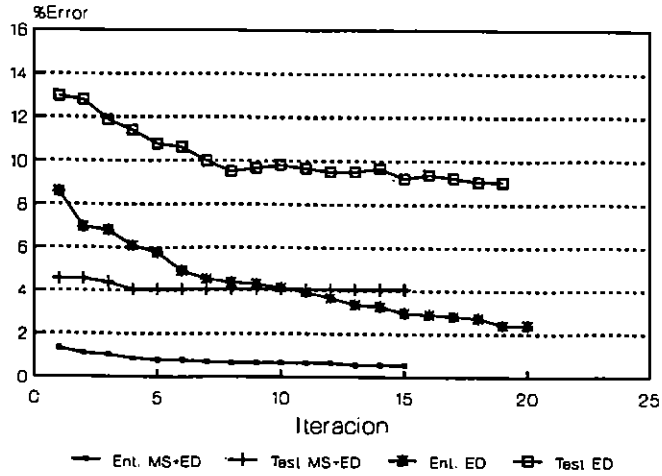

Figura 4: Evolución de las tasas de error para los experimentos MS y MS+EC

casos se puede observar que las tasas de error sobre el conjunto de entrenamiento disminuye monótonamente, mientras que las tasas de error sobre el conjunto de evaluación no cambian o disminuyen muy lentamente tras un número pequeño de iteraciones. Se han detectado algunas limitaciones debido al valor mínimo impuesto para las probabilidades de producción de símbolos. El algoritmo de entrenamiento discriminativo intenta ajustar dichos valores de forma que disminuya la probabilidad total, pero cuando éstos son próximos al valor mínimo, no pueden modificarse significativamente, produciendo oscilaciones en la convergencia.

6 Conclusiones

Se ha mostrado que los HMM pueden considerarse un caso particular de RNN. En este contexto, se ha desarrollado un algoritmo de entrenamiento utilizando las técnicas de retropropagación. Este algoritmo es discriminativo y proporciona mejores resultados que el algoritmo de entrenamiento por máxima semejanza. Los mejores resultados se obtienen para el caso en que los modelos son inicializados mediante el algoritmo de Baum-Welch, y entrenados mediante el algoritmo discriminativo.

La principal ventaja de esta topología y algoritmo de entrenamiento para el reconocimiento del habla es evitar el alineamiento temporal y la reducción en el costo computacional frente a otros tipos de redes. Sin embargo, no existe garantía acerca de la convergencia del algoritmo de entrenamiento discriminativo, y el valor del factor de aprendizaje η se ha mostrado crítico para evitar problemas de convergencia.

Por otra parte, el incremento en complejidad computacional del esquema de entrenamiento competitivo frente al entrenamiento por máxima semejanza es importante en el caso de simulaciones mediante *software* debido a que todas las secuencais deben entrenar a todos los modelos. Sin embargo, en la implementación *hardware* del sistema este incremento es despreciable dada la naturaleza paralela de la red compuesta al unir todos los modelos.

7 Bibliografía

[1] H. Bourlard; C.J. Wellekens. **Speech Dynamics and Recurrent Neural Networks.** *Proc. IEEE ICASSP-89*, vol. 1 pp. 33-36. Glasgow, 1989.

[2] P. Demichellis; L. Fisore; P. Laface; G. Micca; E. Piccolo. **On the Use of Neural Networks for Speaker Independent Isolated Word Recognition.** *Proc. IEEE ICASSP-89*, vol. 1, pp 314-317. Glasgow, 1989.

[3] H. Sakoe; R: Isotani; K. Yoshida; K. Iso; T. Watanabe.**Speaker-Independent Word Recognition using Dynamic Programming Neural Networks.***Proc. IEEE ICASSP-89*, vol. 1, pp. 29-32. Glasgow, 1989.

[4] J. Hwang; J.A. Vlontzos; S. Kung.**A Systolic Neural Network Architecture for HMM.** *IEEE Trans. ASSP,* vol. 37, n. 12, pp. 1967-1979. Diciembre, 1989.

[5] Kung, S; Hwang, J.. **A Unifying Algorithm/Architecture for Artificial Neural Networks.***Proc. ICASSP-89.* pp. 2505-2508

[6] Y. Ephrain; L. Rabiner.**On the Relation Between Modeling Approaches for Speech Recognition.** *IEEE Trans. Information Theory,* vol. 36, n. 2, pp. 372-379. Marzo, 1990.

[7] J.S. Bridle.**Alphanet: A recurrent neural network architecture with a HMM interpretation.***Speech Communication,* vol. 9, n. 1, pp. 83-92. Febrero, 1990.

[8] L.R. Rabiner. **A tutorial on HMM and Select Applications in Speech Recognition.** *Proc. of the IEEE,* vol. 77, n. 2, pp. 257-285. Febrero, 1989.

[9] J.C. Segura Luna. **Modelos de Markov con Cuantización Dependiente para Reconocimiento de Voz.***Tesis Doctoral.* Universidad de Granada, Noviembre 1991.

Para conseguir que un sistema automático pudiera seguir esta misma estrategia, además de trabajar con modelos de prosodia de una gran riqueza y complejidad, sería necesario que ese sistema comprendiese el texto que debe leer. Aunque ya hay sistemas capaces de realizar análisis semánticos del texto y representar ese conocimiento, normalmente se trata de aplicaciones bastante restringidas, y con un gran consumo de recursos. Estos dos inconvenientes hacen que no resulte práctico incorporar este tipo de sistemas a un conversor texto-voz, al menos para el tipo de aplicaciones en que estamos interesados.

Puesto que no resulta práctico intentar realizar un análisis semántico y pragmático del texto que se va a leer, otra opción es la de hacer un análisis sintáctico del mismo. La estructura sintáctica refleja una parte de la estructura semántica del mensaje, y puede proporcionar información suficiente para generar unos modelos de prosodia mucho más ricos y naturales que los empleados por otros sistemas sin este análisis.

En este artículo se va a presentar el analizador sintáctico desarrollado para el conversor texto-voz AMIGO, en Telefónica I+D. En el apartado II se hace una somera descripción del conversor AMIGO (para una descripción más detallada, [2]), centrándose en la información empleada hasta ahora para la generación de la prosodia, y en cómo se generaba esta información. En el apartado III se justifica la utilidad de un análisis sintáctico para conversión texto-voz. En los apartados IV y V se describe la solución elegida para abordar este problema , y aspectos concretos de la realización de la misma. Finalmente se presentan los resultados conseguidos.

2 DESCRIPCIÓN DEL CONVERSOR AMIGO. ANÁLISIS DE CATEGORÍAS

El conversor AMIGO mantiene básicamente la estructura de todos los conversores texto-voz para texto sin restricciones. El primer módulo se encarga de normalizar la escritura del texto de entrada, y seleccionar de éste una unidad de trabajo que será procesada por el resto del conversor. Como unidad de trabajo hemos elegido la frase, pues unidades inferiores pueden mantener fuertes dependencias entre sí.

Sobre esa frase primero se realiza un preproceso destinado a unificar y convertir en texto ciertos tipos de elementos que pueden aparecer: números, abreviaturas, acrónimos, etc. En todo momento se intenta aprovechar toda la información disponible sobre la palabra, para decidir, si es posible, la categoría gramatical de la misma. También se realiza en este módulo la silabicación y acentuación fonética.

A continuación se realiza una categorización gramatical de las palabras. Esta información se venía usando hasta ahora para decidir sobre la inserción de nuevas pausas, no marcadas ortográficamente, de acuerdo a una serie de reglas heurísticas. Para el desarrollo de este módulo, junto con el pausador, partimos de un trabajo del Departamento de Ingeniería Electrónica de la UPM, sobre el que realizamos numerosos cambios, conservando lo esencial de su filosofía.

A la salida de este módulo toda palabra tiene una categoría *gramatical*, de entre un conjunto de 36 posibles. Sin embargo, el conjunto de categorías que utiliza el módulo es de 64. Además de esta categoría, también se recoge otra información útil para el resto de los

módulos, como las formas no personales de los verbos, o si se trata de verbos auxiliares.

Sobre esta información, el pausador determinaba, cuando la longitud del texto entre pausas superaba cierto umbral, y teniendo en cuenta el contexto cercano a la palabra sobre la que había que decidir, si se insertaba una pausa adicional o no. Además, esta pausa quedaba caracterizada por la categoría de la palabra que la había inducido.

A continuación se realiza la conversión de grafemas a alófonos, de acuerdo a unas reglas que recogen hasta 5 posibles factores. En esta conversión se genera un alfabeto de alófonos (el propuesto por Antonio Quilis [3], con 45 unidades) mucho más completo que el que posteriormente se va a utilizar en la síntesis (de 26 unidades).

La prosodia se controla mediante dos parámetros, la duración de cada sonido y la entonación de la frase. Para las duraciones se emplea un modelo totalmente multiplicativo [4], que recoge el efecto del tipo de alófono, alófonos adyacentes, acento, y longitud del grupo fónico y posición del alófono dentro del mismo. El modelo de entonación asigna valores de f0 a determinadas sílabas del grupo fónico, dependiendo de la longitud del mismo y del tipo de pausa. Estos valores se interpolan linealmente, para generar un contorno de picos y valles [5].

Finalmente la información de prosodia y de la secuencia de alófonos se pasa al sintetizador. Éste es el desarrollado por AT&T para su conversor en inglés, y funciona por concatenación de unidades codificadas en LPC Multipulso. La longitud de las unidades es arbitraria, y actualmente nuestro inventario cuenta con trozos de alófonos, demialófonos, trialófonos y tetraalófonos.

Como se ha descrito, el análisis de categorías se utiliza para decidir sobre la inserción de nuevas pausas, e indirectamente para controlar la prosodia generada (también se utiliza para otros pequeños retoques, como determinar la correcta acentuación de palabras que pueden ser átonas o no, y para realizar la concordancia de género cuando se expanden los números). Este análisis de categorías no responde a unas categorías gramaticales muy académicas, sino que se encuentra en un compromiso entre algo correcto, algo útil y algo sencillo y realizable. Este tipo de análisis es muy similar al del sistema de Olivetti y al del CSTR [6], [7].

El primer problema que presenta esta estrategia es que para generar la prosodia sólo se utiliza la información disponible en el caso en que se haya decidido introducir una pausa, y si no, la estructura que hubiera podido recoger el análisis de pausado, se pierde.

Otro problema es que tanto el análisis de categorías como la inserción de pausas se hace con muy poco contexto (3 ó 4 palabras como mucho a cada lado de la que se está considerando), y por tanto no se captura la estructura básica de la frase. No se pueden descubrir relaciones entre elementos de nivel superior.

La primera solución que se consideró fue la de hacer que las reglas que estaba manejando el pausador se aplicasen a toda la frase, y se conservasen sus resultados para el módulo de prosodia, independientemente de que se decidiera insertar una pausa o no. Estas reglas buscan la formación de agrupaciones de palabras dentro de las cuales no se pueden realizar pausas. Posteriormente se pensó en una jerarquización de este proceso para conseguir algo más que conjuntos de palabras en una estructura plana [7].

3 UTILIDAD DE UN MÓDULO DE ANÁLISIS SINTÁCTICO

Como ya se ha comentado en la introducción, el conocimiento de la estructura sintáctica del texto que se pretende sintetizar puede ser uno de los medios necesarios para conseguir una voz sintética de calidad mucho más próxima a la natural.

Aunque el análisis sintáctico por sí sólo no mejora en ningún aspecto la calidad de la síntesis, proporciona información adicional que permite el desarrollo de modelos de prosodia mucho más completos.

La realización de pausas se hace habitualmente en los puntos marcados mediante signos ortográficos, asignando la misma caracterización de la pausa a todos los signos que son iguales. Algunos sistemas más complejos y orientados hacia la conversión de texto sin restricciones pueden realizar pausas adicionales, basándose en la longitud de la frase y las categorías de las palabras [6], [8].

Se puede utilizar la información generada por el analizador sintáctico para conseguir un mejor pausado, tanto en la inserción de nuevas pausas como en la caracterización de todas ellas (por ejemplo, hacer pausas de distinta duración ante una coma que introduce una proposición subordinada de relativo, explicativa, y una coma correspondiente a una enumeración).

También se puede utilizar esta información para la generación de lo que habitualmente se considera prosodia. Hay numerosas referencias que indican la existencia de esta dependencia de la prosodia con la estructura sintáctica, tanto para determinar la duración de los sonidos ([9], [10]) como la entonación ([11],[12]).

4 SOLUCIÓN PROPUESTA

Aunque el análisis sintáctico en sentido estricto del texto es un problema abordable, el planteamiento para este trabajo fue bastante más modesto, por tres razones: resultaba para nosotros muy difícil el intentar definir una gramática para el español capaz de tratar cualquier tipo de texto, el módulo no podía ser demasiado complejo, y tampoco sabíamos qué nivel de precisión y de exactitud era necesario en el módulo para el tipo de información que pretendíamos conseguir de él.

Por tanto nuestra primera idea fue la de hacer agrupaciones de palabras que formasen unidades sencillas, y luego intentar hacer que estas unidades se englobasen en un segundo nivel, y así hasta llegar a agrupar todo en el nivel superior de la frase. De hecho, desde el primer momento nos referimos a este módulo como *estructurador* o *agrupador*, pues no creíamos, y seguimos sin estar muy seguros, que lo que haga se pueda denominar análisis sintáctico formal.

5 REALIZACIÓN

Se decidió emplear un sistema por reglas, pues se pensó que sería factible diseñar reglas capaces de realizar un análisis con un rigor limitado. No se disponía ni de conocimientos suficientes para intentar formalizar esas relaciones gramaticales en un sistema experto o

similar, ni de suficiente texto analizado para desarrollar y entrenar un sistema con autoaprendizaje, o un sistema estocástico.

Una vez decidido que se iban a ir formando agrupaciones de palabras, se planteó la cuestión de si la agrupación debía realizarse de arriba a abajo (decidiendo las unidades de nivel inferior que constituyen cada unidad, partiendo desde la frase), o de abajo a arriba (integrando grupos en grupos de un nivel superior, partiendo desde las palabras). Aun cuando el segundo enfoque parecía más sencillo, llegaba un momento en que no se adivinaba la estrategia para continuar la agrupación, una vez pasados los niveles inferiores. Por otra parte, cualquiera de las dos estrategias era incapaz de resolver correctamente determinadas situaciones, como el decidir si una conjunción copulativa forma parte de una enumeración, o está coordinando dos proposiciones, y a qué nivel en este caso.

Finalmente se decidió emplear una estrategia mixta, en la que las primeras tareas se realizan de abajo a arriba, y luego se intenta descubrir la estructura principal de la frase, de arriba a bajo. Esta tarea es el núcleo del análisis, pues las primeras apenas aportan nada al mismo, sólo simplifican las labores posteriores.

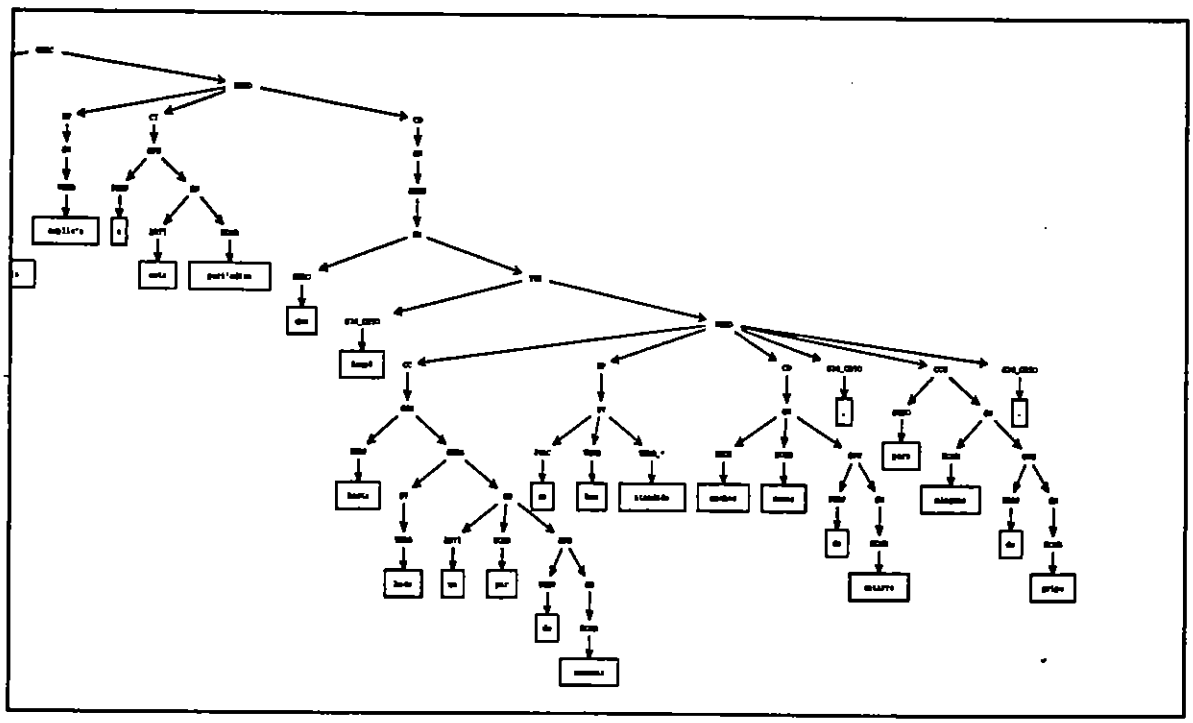

FIGURA 1: Fragmento del resultado del análisis de la frase: **Un médico de atención primaria explicó a este periódico que *"hasta hace un par de semanas se han atendido muchos casos de catarro, pero ninguno de gripe"***. La frase entre comillas aparece bajo la etiqueta YUX, con una estructura propia, y a su vez integrada dentro de la frase portadora. Todas las representaciones gráficas de los árboles sintácticos generados por el conversor texto-voz han sido dibujadas con la herramienta **bonsai**, desarrollada por J. I. Helfman en AT&T Bell Labs.

Para el diseño de las reglas se analizó por los autores (sin un profundo conocimiento

lingüístico) un reducido corpus de texto, unas 2000 palabras, procedentes de artículos de la prensa diaria (el tipo de texto que maneja el conversor en su principal aplicación hasta ahora). Se pergeñaron el tipo y la secuencia de reglas que se debían aplicar, y se depuraron y modificaron analizando otra cantidad similar de texto nuevo.

En este proceso se pusieron de manifiesto ciertos fenómenos que hasta ese momento no habíamos considerado, como la presencia de texto entre comillas que puede formar frases completas, proposiciones subordinadas, sintagmas, o frases sin una completa estructura gramatical.

A continuación se describe a grandes rasgos el conjunto de reglas resultante, así como su realización en el prototipo de alto nivel del conversor texto-voz AMIGO.

El análisis sintáctico está profundamente ligado a un análisis de categorías, y con un enfoque conjunto, los errores de categorizado que aparecen al tratar formas ambiguas se podrían evitar, mediante el análisis de las estructuras resultantes para las distintas hipótesis. Sin embargo, el que el estructurador tenga que manejar un conjunto de categorías por palabra, y generar estructuras para las distintas combinaciones posibles, complicaba muchísimo su diseño, sobre todo con el enfoque que pretendíamos seguir. Por otra parte, los resultados del módulo categorizador son lo bastante buenos como para confiar en ellos, y desarrollar el estructurador sobre los mismos, sin realimentación para corregirlos.

Ante todo fue necesario realizar algunas modificaciones al módulo categorizador, como la de guardar información sobre el número (singular o plural) de las formas verbales, información que luego iba a necesitar el estructurador.

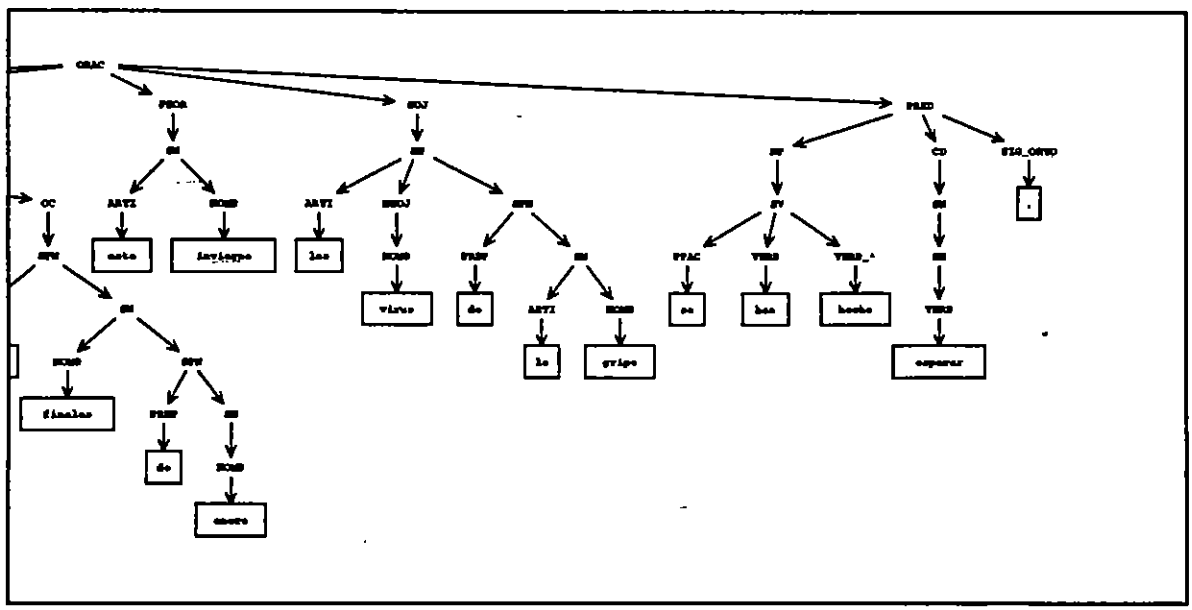

FIGURA 2: Fragmento del resultado del análisis de la frase: **Mientras que el año pasado afectó a un millón de madrileños desde mediados de diciembre hasta finales de enero, *este invierno los virus de la gripe se han hecho esperar.***

De las 36 categorías *gramaticales* que podía decidir el categorizador a su salida, sólo un conjunto de 23 era relevante para el estructurador. La primera tarea es la de realizar una correspondencia suprayectiva entre estos dos conjuntos.

Se aislan y procesan primeramente las proposiciones entre comillas, guiones y paréntesis. Estas estructuras formarán grupos cuyo contenido quedará oculto para el análisis de la oración portadora. Los posibles anidamientos se resuelven mediante el uso de rutinas recursivas. Cada uno de estos trozos se procesan empleando las mismas reglas que para la oración principal.

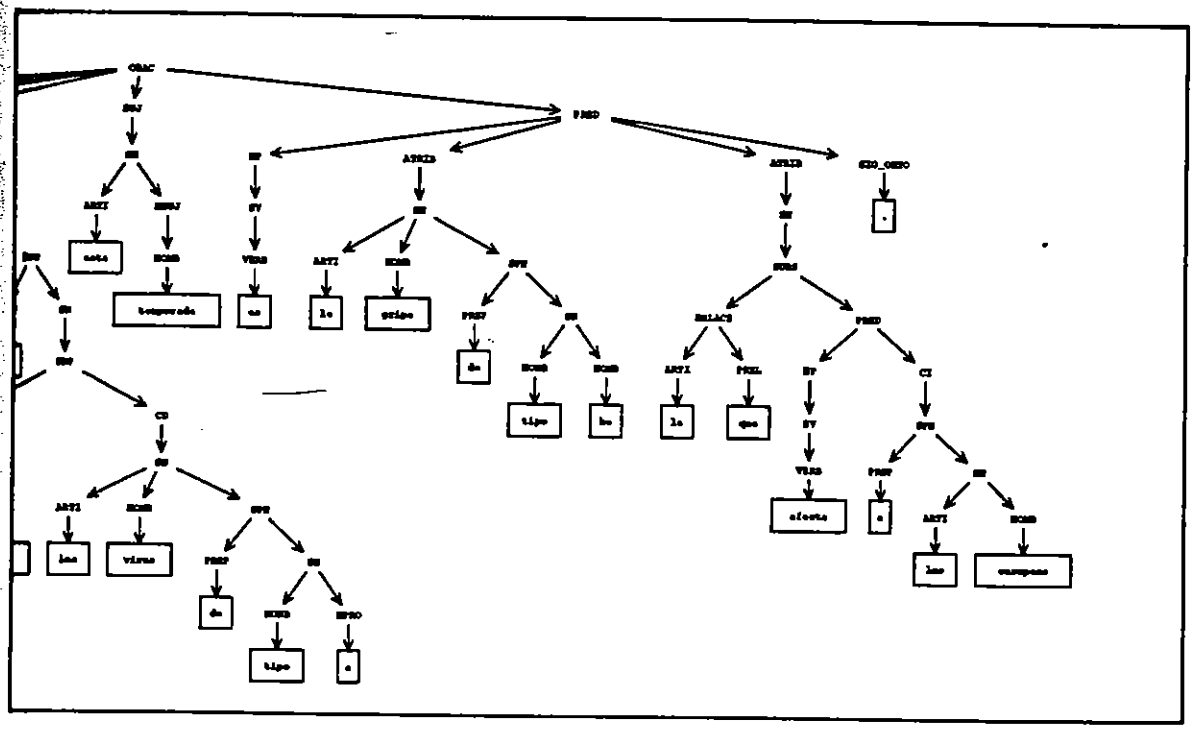

FIGURA 3: Fragmento del resultado del análisis de la frase: **Después de varios años de presentarse los virus de tipo A,** ***esta temporada es la gripe de tipo B la que afecta a los europeos.*** En la figura 2 se identifica correctamente el sujeto de la frase, y el complemento de tiempo queda bajo la etiqueta de Frase ExtraORacional, pues no podemos representar el predicado a ambos lados del sujeto. En esta figura, sin embargo, el sujeto viene detrás del verbo, y entonces toma incorrectamente como sujeto el complemento de tiempo *esta temporada.*

Se van formando elementos nominales y agrupándolos entre sí. Un elemento nominal es una secuencia de nombres (en sentido amplio), posiblemente precedida de un determinante, y quizás de una preposición. En este último caso el tipo de grupo resultante se distingue de los anteriores, pues va a desarrollar funciones totalmente distintas. Se intenta descubrir si un adverbio pertenece al sintagma nominal, o forma parte del predicado. También se tratan las aposiciones y algunos tipos de enumeraciones.

Se identifican y localizan algunos elementos relevantes de la oración: verbos, partículas subordinantes, partículas coordinantes. En este punto se da el salto al segundo enfoque, y se va a intentar descubrir la estructura principal de la oración.

Se trata de ver qué partículas van a generar de verdad proposiciones subordinadas y coordinadas, y cuáles no (por ejemplo, enumeraciones). Básicamente, se define un área de influencia de la partícula, y se busca un verbo en ese área de influencia.

Finalmente se analizan cada una de las proposiciones. Entre los sintagmas nominales a la izquierda del verbo, si alguno concuerda en número con éste, se etiqueta como sujeto. Por tanto, no es capaz de encontrarlo en el caso en que se haya producido una inversión, y el sujeto quede tras el verbo. Los complementos circunstanciales de tiempo que no van precedidos de preposición, se confunden con el complemento directo. Además, con nuestra manera de representar la información en nuestras estructuras de datos, no se permite que un sintagma quede dentro de los límites de otro, si no está incluido en él. Sería necesario reordenar la frase, pero esto no es admisible en una aplicación como ésta.

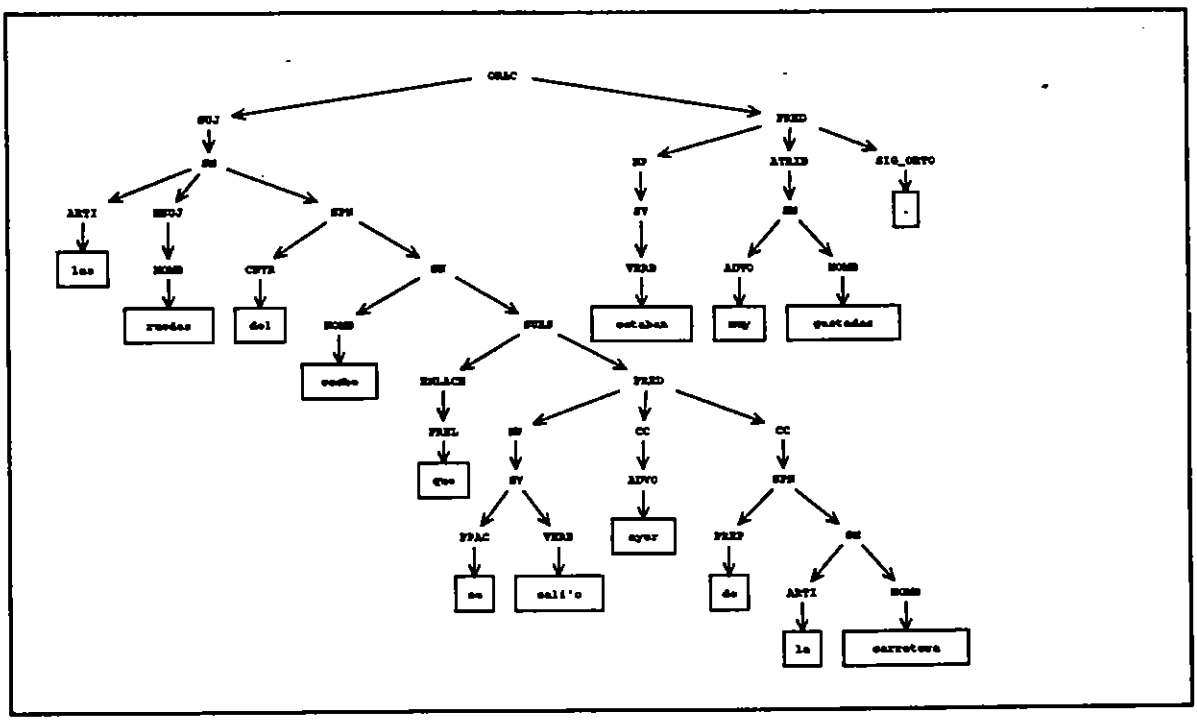

FIGURA 4: Ejemplo del resultado del análisis de una frase con proposición subordinada de relativo: ***Las ruedas del coche que se salió ayer de la carretera estaban muy gastadas.***

Si es necesario, se engloban las proposiciones subordinadas dentro de los sintagmas de los que dependan. Por ejemplo, en el caso de las proposiciones subordinadas de relativo, en determinados casos se busca el sintagma nominal que concuerda en número con el verbo de la subordinada, y se engloba con éste en el nivel correspondiente.

Para la realización de este analizador, además de programar las reglas descritas, ha sido necesario diseñar unas estructuras de datos capaces de representar este tipo de información, y al mismo tiempo que no complicasen demasiado el acceso a la misma. El problema es que hay que mantener la información del análisis en todos sus niveles; no es suficiente con el nivel superior. Optamos por dos tipos, una estructura de datos *de trabajo*, sobre la que es fácil la aplicación de las reglas, y otra estructura *definitiva*, sobre la trabajan el resto de los

módulos.

La estructura *de trabajo* está formada por listas doblemente encadenadas de estructuras que representan los límites del sintagma, una lista por cada palabra. Sólo se guarda información sobre el límite del sintagma, y el tipo de sintagma.

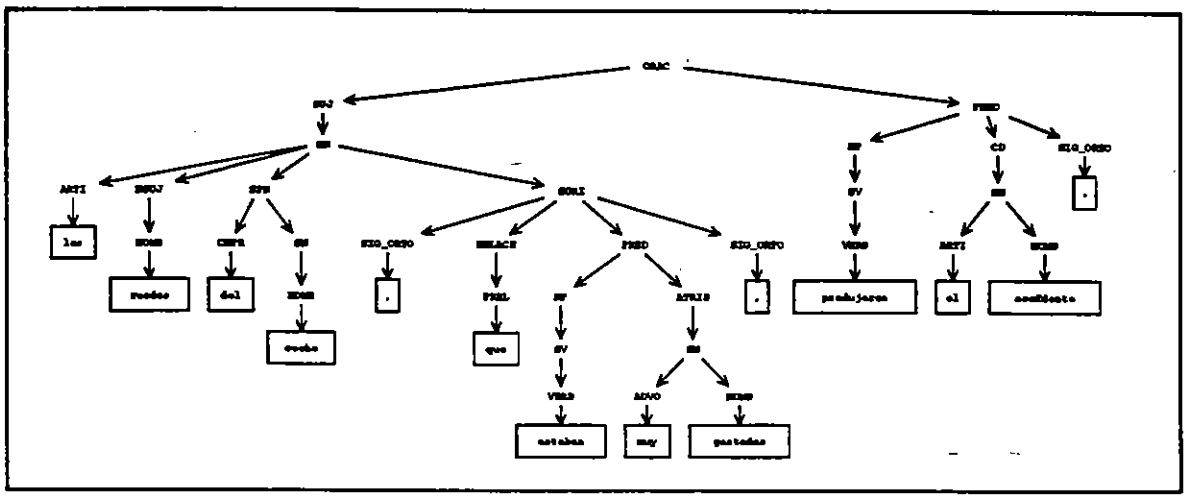

FIGURA 5: Otro ejemplo del resultado del análisis de frases con proposiciones subordinadas de relativo: ***Las ruedas del coche, que estaban gastadas, produjeron el accidente.*** Tanto en la figura anterior como en ésta, la proposición subordinada queda englobada en el nivel correcto.

La estructura *definitiva* recoge la información de la anterior y la representa en forma de árbol, en el que cada nodo representa un sintagma . En el nodo se guarda información sobre el nodo *padre* y los nodos *hijos*, y otra información adicional útil para el resto de los módulos, como el número de sílabas y sílabas tónicas que componen cada sintagma.

6 RESULTADOS

Este analizador no pretende ser un analizador sintáctico formal. Es útil ajustarse a los preceptos de una gramática establecida, pues eso proporciona mayor solidez y un cierto soporte teórico a nuestro trabajo , y permitirá aprovechar resultados de otros grupos sobre representación prosódica de la información sintáctica, pero no hemos tenido esto como objetivo por sí mismo. Nuestro principal objetivo ha sido desarrollar un analizador capaz de tratar cualquier tipo de frase, incluso las mal construidas, y formar agrupaciones de sintagmas y niveles dentro de la estructura que se aproximasen a la estructura real de la frase; nos importa más cuál es el límite entre dos sintagmas de un cierto nivel que las etiquetas de cada uno de esos sintagmas. También intentamos evitar el tener que representar información semántica.

Esto, junto con el esfuerzo que nos suponía analizar a mano una gran cantidad de texto para obtener una valoración significativa, ha hecho que nos limitásemos a comprobar los resultados con unas 6000 palabras, incluyendo las 4000 que ya utilizamos en el desarrollo. Además comprobamos con unas 26000 palabras que el sistema era capaz de analizar una gran variedad

de frases, sin preocuparnos por la calidad del análisis.

Los resultados han sido muy satisfactorios, superando ampliamente nuestras ambiciones y espectativas al principio del desarrollo. Salvo la falta de rigor en las etiquetas de los sintagmas, el sistema suele tratar correctamente las frases sencillas y las relaciones de subordinación. Los principales problemas provienen de las relaciones de coordinación, de los sintagmas nominales encabezados por preposición y englobados dentro de otro sintagma nominal, y los complementos directos y circunstanciales que aparecen por delante del verbo, y que puede llegar a confundirlos con el sujeto.

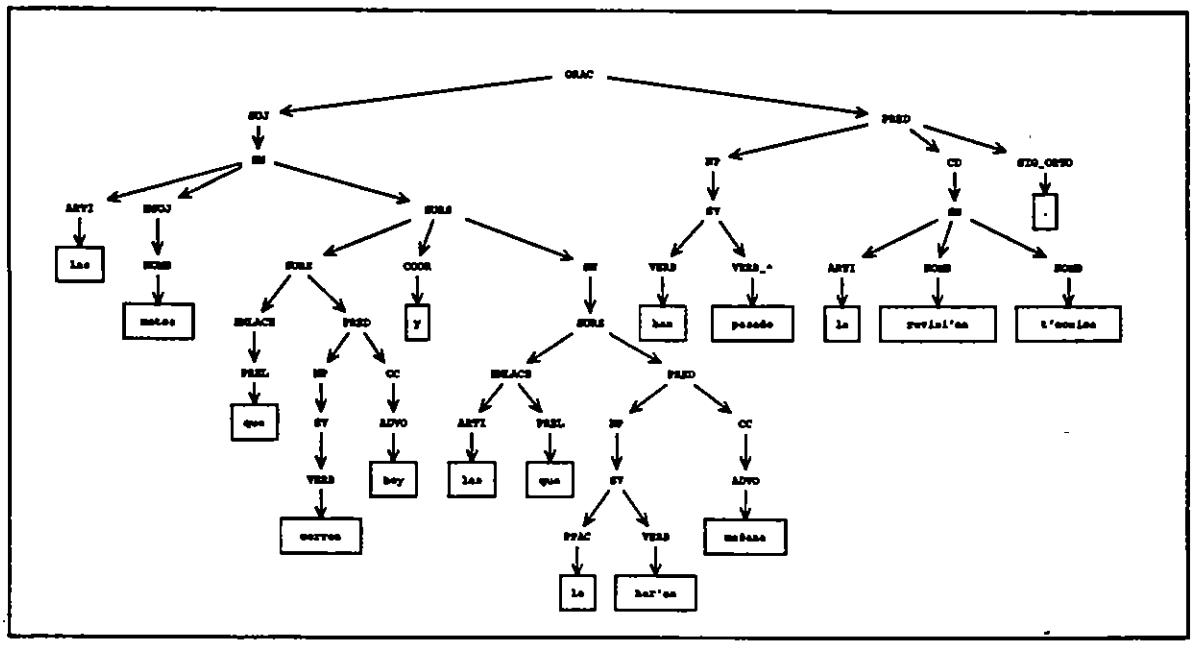

FIGURA 6: Resultado del análisis de la frase: ***Las motos que corren hoy y las que lo harán mañana han pasado la revisión técnica***. El analizador descubre correctamente la enumeración de proposiciones de relativo.

A partir de ahora la tarea que queda por realizar es demostrar realmente que esta información puede mejorar la calidad de la voz, desarrollando un nuevo módulo de pausado, con mejores criterios para la inserción de pausas, y más flexible, y un nuevo modelo de prosodia, particularmente de la entonación. La mayor parte de los estudios sobre entonación se han centrado hasta ahora en la modelización de los grupos terminales. Para aprovechar los resultados de este analizador, habrá que caracterizar los grupos no terminales, relacionando los contornos de F0 con la estructura sintáctica de la frase.

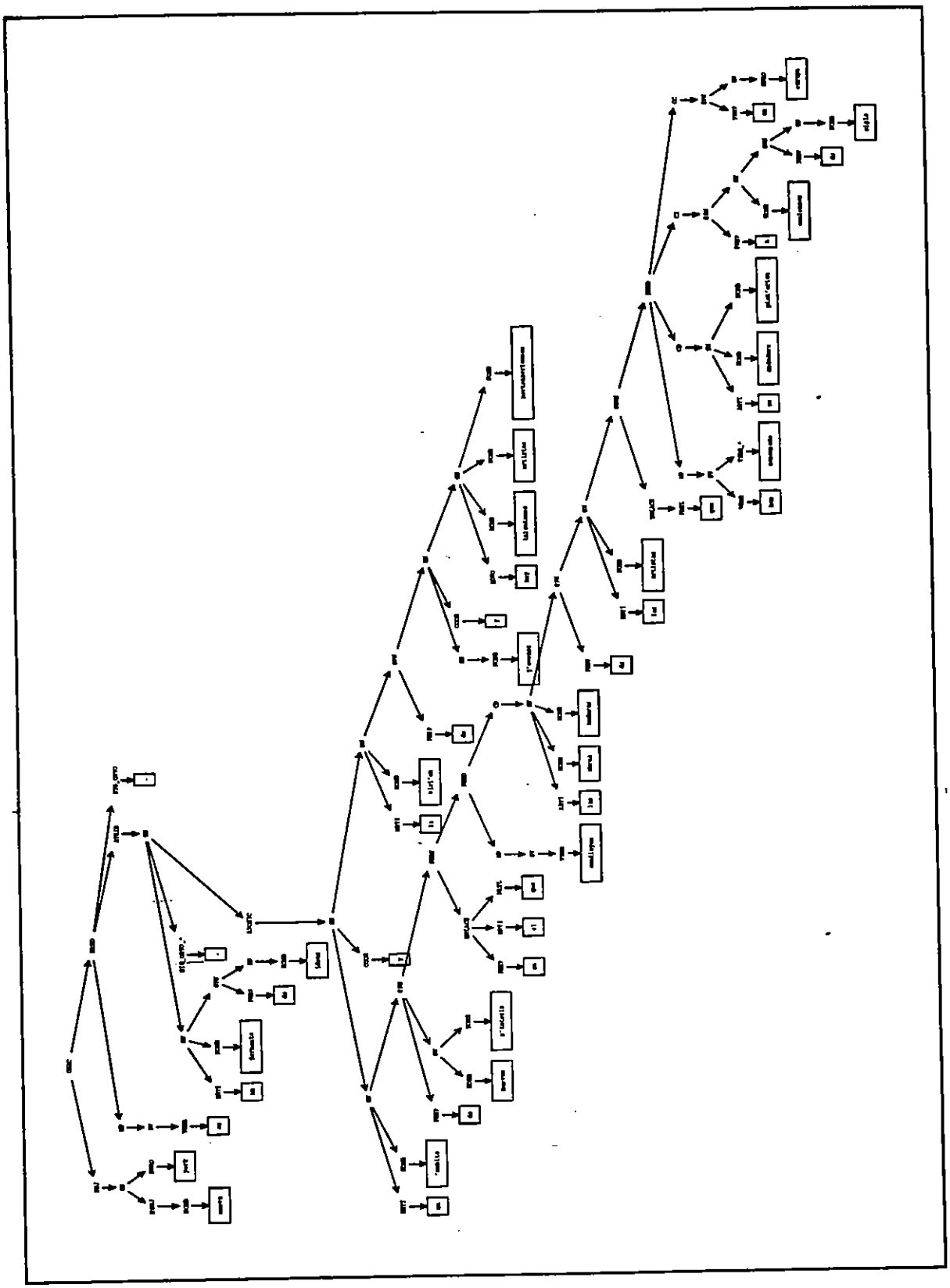

FIGURA 7: Resultado del análisis de la frase: ***Nueva York es un fermento de ideas, un ámbito de nuevas síntesis en el que confluyen las obras maduras de los artistas que han comenzado su andadura pictórica a comienzos de siglo en Europa y la visión de jóvenes***

y muy talentosos artistas norteamericanos. En este caso, la primera partícula coordinante no queda en el nivel correcto.

7 CONCLUSIÓN

En este artículo se ha presentado un analizador sintáctico para un sistema de conversión texto-voz. En esta aplicación la información generada por un análisis sintáctico puede ser muy útil para el objetivo de lograr voz sintética de muy alta calidad.

Hemos demostrado que si se admite una cierta relajación del rigor formal de los resultados, es factible incorporar este módulo a un sistema de conversión texto-voz que debe funcionar en tiempo real, con muy poca carga adicional de tiempo de proceso y de ocupación de memoria.

8 REFERENCIAS

[1] J.A. Siles, *An Audiotex system with speech recognition and synthesis to be used over the Spanish telephone network*, Secondes Rencontres Europeennes de L'Audiotex, (1991)

[2] M.A. Rodríguez, *AMIGO: Un conversor texto-voz para español*, VIII Congreso SEPLN, (1992)

[3] A. Quilis, *El comentario fonológico y fonético de texto*, ed ARCOS/LIBROS, (1988)

[4] A. Macarrón, *Design of Duration Rules for a Spanish Text-to-Speech Synthesizer*, Proc. Eurospeech '91, (1991)

[5] P.J. Moreno, *Improving Naturalness in a Text to Speech System with a New Fundamental frequency Algorithm*, Proc. Eurospeech '89, (1989)

[6] S. Quazza, *Contextual Syntactic Analysis for text-to-Speech Conversion*, Proc. Eusipco '87, (1987)

[7] A.I.C. Monaghan, *A Multi-Phase Parsing Strategy for Unrestricted Text*, ESCA Workshop on Speech Synthesis, (1990)

[8] A. Santos, *Diseño y evaluación de reglas de duración en la conversión texto a voz*, Revista de la SEPLN, bol. 6, (1988)

[9] D.H. Klatt, *Vowel lengthening is syntactically determined in a connected discourse*, Journal of Phonetics, 3, (1975)

[10] T.H. Crystal, *Segmental durations in connected-speech signals: Current results*, JASA 83 (4), (1988)

[11] J. 't Hart, *Integrating different levels of intonation analysis*, Journal of Phonetics, 3, (1975)

[12] G. Bailly, *Integration of rhythmic and syntactic constraints in a model of generation of French prosody*, Speech Communication, 8, (1989)