5. Reconocimiento automático del habla

APLICACION DEL ALGORITMO CKY COMO PROCESADOR LINGÜISTICO DE UN SISTEMA DE RECONOCIMIENTO DEL HABLA BASADO EN LA BUSQUEDA DE PALABRAS

Antonio Bonafonte, Eduardo Lleida, José B. Mariño

Departamento de Teoría de la Señal y Comunicaciones
Universitat Politècnica de Catalunya
Antonio@tsc.upc.es

I. Introducción.

El objetivo final de un sistema de reconocimiento del habla es la compresión por parte de un sistema informático de un mensaje oral. Un primer enfoque en la resolución del problema se muestra en la figura I.

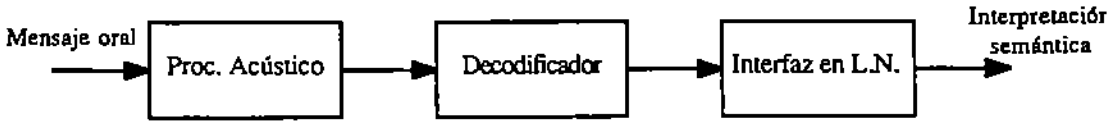


Fig.I Comprensión del mensaje oral. Un primer enfoque.

El procesador acústico (PA) realizaría lo que se conoce como reconocimiento del habla, es decir, transformar la señal a su entrada en una transcripción fonética del mensaje. Para ello, el procesador podría comparar unos patrones que representaran a las unidades fonéticas básicas (fonemas, palabras, sílabas...) con la señal de test. El decodificador debería representar las palabras según su pronunciación estándar, por ejemplo en su forma ortográfica, debiendo corregir errores que pudieran haberse cometido. La salida del decodificador podría activar la entrada de un interfaz en lenguaje natural (ILN).

La modularidad del esquema propuesto proporciona una gran simplicidad en el diseño de los distintos bloques. Sin embargo, debido al poco rendimiento de los sistemas de reconocimiento la propuesta no es viable. Una segunda posibilidad sería la de integrar el sistema de reconocimiento con el interfaz de lenguaje natural. Puede no tener sentido que el PA proporcione una salida que el ILN no considera válida. Existen sistemas que adoptan esta alternativa[1] pero debido a que todo proceso de integración resulta en un proceso mucho más complejo que los componentes y a la complejidad de los constituyentes que nos ocupan, es fácil comprender por qué estos sistemas utilizan simplificaciones importantes: no es frecuente el modelado de la coarticulación entre palabras, la utilización de la semántica es prácticamente inexistente, no se suelen considerar fenómenos de concordancia.

Otro enfoque propuesto por Schwartz[2] consiste en aplicar las distintas fuentes de conocimiento progresivamente, dando prioridad a aquellas que reducen más la entropía y que no supongan un incremento desmesurado del coste. Esta alternativa, es una situación intermedia entre las dos anteriores y presenta uno de los inconvenientes del primero: puede ocurrir que al

Este trabajo ha sido financiado por una ayuda PRONTIC, proyecto nº 105/88

aplicar las primeras fuentes de conocimiento obtengamos una propuesta que no es aceptada por las próximas fuentes a aplicar. La solución que adopta Schwartz es utilizar en el reconocimiento un algoritmo de reconocimiento que genere, las N mejores frases y, de entre ellas, elegir la primera que sea correcta de acuerdo a otros conocimientos no considerados. Para generar las hipótesis se suele utilizar además de la información acústico-fonética, información léxica (no se permite cualquier concatenación de unidades elementales) y para modelar el lenguaje un N-Gram con N usualmente 2 ó 3. El inconveniente de este esquema es su coste computacional. Incluso propuestas aproximadas[3] requieren un número de operaciones proporcional al número de hipótesis N. Además, algunas implementaciones requieren que N sea fijado a priori por lo que si ninguna hipótesis es aceptada por el post procesador deberían rehacerse los cálculos.

La alternativa que presentamos, consiste en liberar al procesador acústico de la tarea de generar frases y posponerlo al momento en que el procesador lingüístico realice el análisis sintáctico. El procesador acústico, utilizando información acústico-fonética y léxica, generará una celosía de palabras donde cada elemento de la celosía consta de un identificador de la palabra detectada, sus posiciones de comienzo y de fin y el coste de la detección, esto es, una medida de la semejanza entre la señal de test y los patrones de referencia. El número de elementos de esta celosía ha de ser suficiente para garantizar la existencia de la frase correcta al concatenar elementos de la celosía. El coste computacional de este buscador de palabras es pequeño al utilizar el algoritmo de un solo paso propuesto por Lleida[4].

El procesador lingüístico deberá tomar de entre todas las palabras aquellas que puedan formar la frase gramatical de mínimo coste. La versión que aquí presentamos tiene como base una generalización del algoritmo de Coocke-Younger-Kasami[5] (CKY) cuya entrada es la celosía de palabras descrita.

En el próximo apartado se revisará el funcionamiento del procesador acústico buscador de palabras. En el apartado III se describe la forma de la gramática de entrada y el modo de funcionamiento del procesador lingüístico. En IV se muestra un ejemplo de aplicación y en V se presentan los resultados obtenidos.

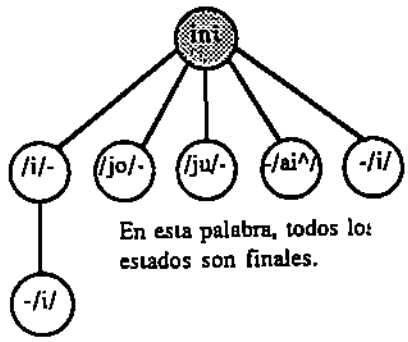
II. Procesador Acústico.

Procesado de señal. La señal proporcionada por un micrófono es filtrada en banda telefónica y muestreada a 8 KHz. La frase es aislada mediante un algoritmo de detección de final de frase. A continuación se realiza una parametrización de la señal: la señal es segmentada en tramas de 30 ms con una ventana de Hamming. El solape entre ventanas es de 15 ms. Para cada trama se estiman 12 coeficientes cepstra del espectro LPC de orden 8. La energía de la trama expresada en decibelios completa la parametrización. Antes de entrar en el algoritmo de reconocimiento el sistema evalúa la derivada temporal de los coeficientes cepstra y de la energía. El vector espectral, el vector de derivada espectral y el coeficiente de derivada de la energía se cuantifican independientemente mediante tres cuantificadores vectoriales con un número de centroides de 64, 64 y 32 respectivamente. De esta forma, cada trama de la señal de voz es representada mediante tres símbolos.

Unidad fonética. La unidad fonética que ha sido adoptada es la semisílaba. Esto es así pues el carácter silábico del castellano otorga a las sílabas una relativa independencia del contexto en que se hallan, independencia que es en gran parte trasladada a las semisílabas. Además, el número de semisílabas necesario para cubrir el castellano es relativamente pequeño, alrededor de 600. Para definir una semisílaba, se divide una sílaba por su vocal fuerte obteniendo una semisílaba inicial y otra final.

Modelado de las unidades fonéticas. Cada semisílaba se representa por medio de un Modelo Oculto de Markov (HMM). La estructura del modelo es la estándar: modelos de izquierda a derecha con la capacidad de saltar por encima de un estado. El número de estados se eligió de acuerdo a la duración media de cada semisílaba.

Conocimiento Léxico. La información léxica es presentada al procesador acústico de dos formas: mediante un autómata de estados finitos que dirige el algoritmo de reconocimiento y mediante un árbol léxico que identifica las palabras detectadas. Estas representaciones consideran tanto las distintas alternativas al pronunciar una palabra (ej: /tre´i^nta/, /tre´nta/, como el efecto de coarticulación entre palabras (ej /tre´i^nta i u´no/, /tre´i^nta j u´no/; al silabificar: /tre´i^n·ta·i·u´·no/, /tre´i^n·ta·ju´·no/). Así por ejemplo, el árbol léxico correspondiente a la palabra i será:



Algoritmo Detector de Palabras.[4] El procedimiento utilizado para localizar las palabras está basado en el algoritmo de Viterbi pero permite que en cualquier trama de señal pueda comenzar una palabra. Las transiciones entre unidades se hacen de acuerdo al autómata y cada vez que un camino alcanza un estado final del autómata, se utiliza el árbol léxico para saber qué palabra se ha detectado. El procesador acústico proporciona para cada trama las N mejores palabras que terminan en esa posición. Notar que al no permitir la formación de frases no se necesita utilizar un algoritmo de generación de múltiples hipótesis sino que a medida que le algoritmo de Viterbi recorre las tramas de la señal de test toma aquellas palabras que han recibido un coste de reconocimiento menor. Para no perder hipótesis no se ha procedido a una compactación del autómata, es decir, no se ha considerado el que varias palabras puedan compartir un mismo estado. El campo de búsqueda que recorre el algoritmo es el de un lenguaje de palabras aisladas.

III. Procesador Lingüístico.

El procesador lingüístico que presentamos está basado en el algoritmo CKY. Este algoritmo realiza un análisis sintáctico de la celosía de entrada de acuerdo con una gramática de libre contexto (CFG) cuyas reglas estén expresadas según las formas normales de Chomsky.

Normalización de las reglas. Una CFG, siguiendo la notación de Gonzalez y Thomason[6], se caracteriza por tener sus producciones de la forma $A \rightarrow a$, siendo A un símbolo no terminal y a en $V^+ = V^* - l$. V es la unión de los símbolos terminales y no terminales, V^* es el conjunto infinito de todas las frases que se pueden formar con elementos de V y l es la cadena vacía. Una CFG está representada en formas normales de Chomsky si todas las producciones son de la forma

$$A \rightarrow BC \quad \text{ó} \quad A \rightarrow a$$

siendo las letras mayúsculas símbolos no terminales y las minúsculas terminales. A pesar de que el algoritmo de análisis requiere que las normas estén en formas normales de Chomsky, en el diseño de la gramática es conveniente poder utilizar la potencialidad de una regla genérica en beneficio de la brevedad y de la claridad. También es conveniente permitir en el diseño la posibilidad de que un no terminal se reescriba como la cadena vacía l.. El conjunto de las transformaciones que hemos seguido al compilar la gramática ha sido:

1) Eliminar reglas $A \rightarrow \lambda$.

   Si A Æ l entonces añadir para toda regla del tipo $B \rightarrow \alpha A \beta$ la regla $B \rightarrow \alpha \beta$

2) Eliminar reglas $A \rightarrow \theta_1 \theta_2 \ldots \theta_v$ donde θ_i es un elemento de V.

   Seguimos las pautas dadas por Gonzalez-Thomason

3) Eliminar las reglas $A \rightarrow B$

   Dos son los procedimientos que se pueden seguir.

   a) Para toda regla $X \rightarrow \alpha A \beta$ añadir la regla $X \rightarrow \alpha B \beta$ .Notar que si A es *frase*, a la regla implícita de *Objetivo* $\rightarrow$ A ahora hay que añadir *Objetivo* $\rightarrow$ B, es decir, el análisis habrá tenido éxito si la frase observable se puede construir a partir de A o de B.

   b) Para toda regla $B \rightarrow \alpha$ añadir la regla $A \rightarrow \alpha$ Notar que si a es un terminal entonces estas reglas suelen localizarse en un diccionario de entrada que codifica la frase en preterminales. Es allí donde se debe duplicar la regla añadiendo una nueva categoría al terminal.

4) Finalmente, para reducir el tiempo de búsqueda, las reglas han sido ordenadas de acuerdo al símbolo de la derecha.

Hay que decir que en la aplicación de 1) y de 3) pueden aparecer reglas redundantes, es decir, reglas repetidas o reglas en que un no terminal se reescribe en si mismo. Conviene eliminarlas cada vez que se normaliza una regla.

Generalización del algoritmo CKY. Antes de exponer como se utiliza el algoritmo CKY como procesador lingüístico de la información entregada por la etapa anterior vamos a revisar los principios de este algoritmo. Adoptamos la notación utilizada por Aho[5].

Sea $w = a_1 a_2 \ldots a_n$ la cadena a analizar de acuerdo a una gramática G expresada mediante formas normales de Chomsky. El algoritmo consiste en la construcción de una tabla triangular de análisis cuyos elementos denominaremos t_{ij} , $1 \leq i \leq n$, $1 \leq j \leq n-i+1$. Cada elemento t_{ij} es una celda que contiene un subconjunto de los no terminales. Un no terminal A pertenece a t_{ij} si a partir de A, y aplicando las reglas de la gramática, se pueden obtener los j símbolos de la frase de entrada que se encuentran a partir de la posición i, es decir, $a_i a_{i+1} \ldots a_{i+j-1}$. La frase es correcta si el no terminal *frase* pertenece a t_{1n}. El procedimiento para llenar la tabla es:

1) Inicialización.

Desde i=1 hasta n, hacer:

Utilizando el diccionario añadir a t_{i1} las categorías asignadas a a_i.

2) Llenar celdas:

Desde j=2 hasta n hacer

Desde i=1 hasta n-j+1 hacer

/ Llenado de la casilla t_{ij} */*

Desde k=1 hasta j-1 hacer

Añadir a t_{ij} el no terminal A si existe alguna regla $A \rightarrow BC$ tal que B pertenezca a t_{ik} y C pertenezca a $t_{i+k,j-k}$

/ Es decir, para que A produzca los j símbolos que siguen a partir del i, buscamos alguna regla $A \rightarrow BC$ tal que B produzca los k primeros símbolos y C el resto */*

Fin del bucle en k

Fin del bucle en i

Fin del bucle en j

3)Verificar el éxito del análisis.

Buscar en t_{1n} la presencia del no terminal S (*frase*) y aplicar un algoritmo de "backtracking" para recuperar el árbol (o los árboles) de análisis.

En nuestro sistema, los datos que proporciona el procesador acústico es, como ya hemos visto, no una frase, sino una celosía de palabras. Consideraremos que un no terminal A pertenece a t_{ij} si a partir de A, y aplicando las reglas de la gramática, se pueden observar j tramas de señal a partir de la trama i. Además asociaremos a cada elemento de t_{ij} el coste asociado al reconocimiento de esa fracción de frase. Si el coste es excesivamente alto, esa posibilidad no se podrá formar a partir de elementos de la celosía. Para ilustrar el principio de funcionamiento considérese que en la celosía se encuentran un grupo de palabras $w_1, w_2, \ldots w_m$, que empiezan en las posiciones $i_1, i_2, \ldots i_m$, terminan en $f_1, f_2, \ldots f_m$, y han recibido los costes $q_1, q_2, \ldots q_m$. Si a partir de A, y aplicando las reglas de la gramática es posible derivar la cadena de terminales $w_1\ w_2 \ldots w_m$ y se cumple que $i_{l+1} = f_l + 1$ significa que A puede observar el segmento de señal que comprende desde la trama i_1 a la trama f_m y por tanto A pertenecerá a t_{i_1,f_m-i_1+1}, y el coste asociado será $q_1+q_2+ \ldots +q_m$

El algoritmo que resulta es:

1) Inicialización.

Mientras queden elementos en la celosía hacer:

Extraer una palabra de la celosía: palabra w; inicio i; final i+j-1; coste Q.

Utilizando el diccionario añadir a t_{ij} las categorías asignadas a w y con coste Q..

2) Llenar celdas:

Sea T el nº de tramas de la señal de entrada,

Desde j=2 hasta T hacer

Desde i=1 hasta T-j+1 hacer

/ Actualización de la casilla t_{ij} */*

Desde k=1 hasta j-1 hacer

Añadir a t_{ij} el no terminal A si existe alguna regla A→ BC tal que B pertenezca a t_{ik} y C pertenezca a $t_{i+k,j-k}$. Como coste asociado a A tomar la suma de los costes asociados a B y a C.

/ Es decir, para que A produzca las j tramas que siguen a partir de la i, buscamos alguna regla A→ BC tal que B produzca las k primeras tramas y C el resto */*

Fin del bucle en k

Si en la casilla t_{ij} aparece un no terminal varias veces y con distinto coste, eliminar los de mayor coste./* *Buscamos la frase gramatical más probable* */

Fin del bucle en i

Fin del bucle en j

3)Verificar el éxito del análisis.

Buscar en t_{1T} la presencia del no terminal S (*frase*) y aplicar un algoritmo de "backtracking" para recuperar el árbol (o los árboles) de análisis y la frase reconocida.(Una implementación posible es añadir a cada elemento de t_{ij} dos apuntadores: uno apuntará al elemento de t_{ik} y el otro, al elemento de $t_{i+k,j-k}$

Relajación de la condición de ajuste temporal. El algoritmo que se ha presentado exige de la frase reconocida que sea gramatical y que las distintas partes de la frase que se combinan mediante una regla tengan un perfecto ajuste temporal: el inicio de la subfrase de la derecha ha de estar a continuación del final de la subfrase de la izquierda. Además, la frase ha de durar todo el tiempo que nos ha indicado el detector de principio y fin. En una aplicación donde se quiera interpretar habla espontánea, puede interesar la interpretación de frases que no son puramente gramaticales. Un modo de realizarlo es "modelando la no gramaticalidad en el diseño de la gramática del lenguaje". Otra forma es adoptando técnicas de análisis flexible[7]. En cuanto a la segunda exigencia, el perfecto ajuste temporal, invalida nuestro sistema. Es bien sabida la dificultad de realizar una detección exacta y automática de los límites de la frase, tanto más cierto cuanto mayor es el nivel de ruido. Bastaría que el procesador acústico no propusiera la palabra

final como candidato en la última trama para que el sistema fracasara. Una solución que se adopta es relajar el detector de principio y fin y permitir que la frase pueda empezar y acabar dentro de unos márgenes. La incorporación de flexibilidad al procesador lingüístico es inmediata: buscar el símbolo *'frase'* no sólo en la casilla t_{iT} sino permitir que el inicio **i** varíe de 1 a 1+k, y la duración j, de T-i+1 hasta T-i+1-k. Se toma como candidato aquella frase que se halla en esta zona y que tiene un menor coste. Debido a la comparación de frases de distinta duración es necesaria una normalización del coste en función de la duración. En cuanto a la relajación de la continuidad entre dos subfrases, una posible solución es la de permitir la aplicación de una regla si las categorías constituyentes se encuentran en casillas próximas al lugar que les correspondería. Podría penalizarse la desviación de la posición de los constituyentes respecto el lugar óptimo. El precio que se pagaría es el de aumentar el tiempo de búsqueda para aplicar una regla en un factor multiplicativo. La solución que hemos adoptado ha sido la de realizar una cuantización de los inicios y duraciones de los elementos de la celosía. Así, si una palabra Q quedaba caracterizada por su inicio i, su duración d y su coste Q, la hemos transformado en otra palabra w' con el mismo identificador pero con atributos i', d' y Q', siendo

$$i' = \text{int}\,(i/h) \qquad d' = \text{int}\,(d/h)) \qquad \text{y} \qquad Q' = Q\ (d'/d)$$

y h el cuanto elegido en la medida de los inicios y duraciones prefijado en el diseño del sistema. La normalización de Q es necesaria pues en la misma casilla pueden coincidir símbolos que se refieren a palabras con una longitud ligeramente distinta con lo que, a falta de normalización, se favorecería al más corto.

A primera vista, esta solución no es es aceptable pues aunque pueden combinarse constituyentes que antes no coincidían de forma exacta, con lo que se logra la relajación en la condición de continuidad, también puede ocurrir que elementos que antes de normalizar ajustaban no lo hagan después de normalizar debido a que la aplicación

int ([ ]/h)) no es lineal.

No obstante, esta solución ha permitido aumentar sensiblemente el rendimiento reduciendo notablemente la carga computacional del procesador lingüístico. La característica anterior no ha producido ningún efecto considerable debido a que una misma palabra suele aparecer con distintos costes en una área de la tabla de análisis que queda transformada en varias casillas adyacentes.

IV. Descripción de la aplicación.

La aplicación que hemos utilizado para medir el grado de funcionamiento del sistema descrito ha sido el reconocimiento de números telefónicos de siete cifras. Se permite cualquier agrupación entre las cifras siempre que cada uno de los grupos esté entre 0 y 999. La razón para elegir esta tarea ha sido su interés práctico, la dificultad que presenta desde el punto de vista de reconocimiento debido al parecido entre palabras (a pesar de su tamaño limitado) y el disponer en nuestro laboratorio de grabaciones de este tipo de señales. La gramática que representa este lenguaje podría ser obviamente regular, sin embargo, nótese que el número de estados necesarios (en comparación a los autómatas para reconocer los números de uno, dos y tres dígitos) sería considerable. Es por esto que en los experimentos que se realizan en nuestro laboratorio con RAMSES[8,9], un sistema que integra los procesadores lingüístico y acústico, se suele utilizar como gramática el autómata que representa los números naturales del 0 al 999, permitiéndose cualquier número de grupos. El resultado es que en ocasiones la frase propuesta ¡no es un número telefónico!

Bases de datos. Para la realización de estos experimentos se han utilizado dos bases de datos:

a) Base de entrenamiento. Se ha utilizado para estimar los modelos ocultos de Markov de cada semisílaba. Consiste en cadenas de número enteros menores que un millón.

Ej: 21100/1/2/19
310001/99

Se dispone de un conjunto de cuarenta cadenas pronunciadas por diez locutores, cinco mujeres y cinco hombres. Las cadenas están agrupadas en dos series con cinco hablantes por serie.

b)Base de test.

Consiste en números telefónicos de siete cifras agrupadas en una, dos o tres cifras.

Ej: 223/97/54
3/65/3/64/1

Se dispone de diez señales de veinte locutores, diez mujeres y diez hombres, de los cuales diez son los que han participado en el entrenamiento. Ningún número ha sido repetido por dos locutores.

Gramática y diccionario. Debido a su brevedad, a continuación se muestran la gramática y el diccionario que se han utilizado. En el diccionario puede verse la definición de palabras que se ha utilizado. La gramática se muestra antes de la compilación.

{

DICCIONARIO

PARA EL RECONOCIMIENTO

DE LOS NUMEROS

TELEFONICOS

}

cero	:*CERO.
uno,dos,tres,cuatro,cinco	:*1-5.
seis,siete,ocho,nueve	:*6-9.
diez,once,doce,trece,catorce,quince	:*10-15.
dieci	:*DIECI.
treinta,cuarenta,cincuenta, sesenta,setenta,ochenta,noventa	:*P30-90.
veinte	:*VEINTE.
veinti	:*VEINTI.
cien	:*CIEN.
ciento	:*CIENTO.
doscientos, trescientos, cuatrocientos, quinientos, seiscientos, setecientos, ochocientos, novecientos	:*CENTENAS.
i	:*I.

{

GRAMATICA

PARA EL RECONOCIMIENTO

DE LOS NUMEROS

TELEFONICOS

}

{Agrupación arbitraria pero el mayor nº pronunciable es el 999 }

NºTelef:	1d 6D/2d 5D/3d 4D.
6D:	1d 5D/2d 4D/3d 3D.
5D:	1d 4D/2d 3D/3d 2D.
4D:	1d 3D/2d 2D/3d 1D.
3D:	1d 2D/2d 1D/3d.
2D:	1d 1D/2d.
1D:	1d.

{Relación de los signos 1d 2d y 3d con los preterminales del diccionario }

1d:	*CERO/1-9.
2d:	*10-15/*DIECI *6-9/*P30-90/*VEINTE/ DECENAS_I 1-9.
3d:	*CIEN/*CIENTO 2d
3d:	*CENTENAS 2D_o_NULL.

1-9:	*1-5/*6-9.
2D_o_NULL:	NULL/2d/1-9.
DECENAS_I:	*VEINTI/*P30-90 *I.

V. Resultados.

Se han realizado dos conjuntos de experimentos para evaluar la bondad del sistema: el primero en un entorno multilocutor, el segundo con independencia del locutor.

Los parámetros del procesador acústico y lingüísticos, además de los mencionados en la descripción del sistema fueron:

Nº de hipótesis por trama proporcionado por el procesador acústico: 5
Relajación del instante de inicio y fin de la frase: 5 %
Cuanto utilizado en la cuantificación temporal: 5 tramas

La elección del valor de estos parámetros ha sido, a falta de experimentación, arbitrario.

Antes de presentar los resultados, definamos algunos términos y veamos como se calcularon. Primeramente se determinó el número de palabras correctas (C), de inserciones (I), de sustituciones (S) y de omisiones (O). Para ello se realizó un alineamiento entre la frase correcta y la frase reconocida. A continuación se calcula

$$\% \text{ Error} = \frac{S+O+I}{L} * 100$$

$$\% \text{ Aciertos} = \frac{C}{L} * 100$$

siendo L el número de palabras de la frase correcta.

En cuanto al nivel de frase, una frase es correcta si no contiene ningún error (ni inserción, ni de omisión, ni sustitución)

Los resultados obtenidos se reflejan en la siguiente tabla:

Experimentos	% Error Pal.	%Acierto Pal.	%Acierto Frase
Dependiente	13,6	89,3	52,0
Independiente	11,7	91,2	49,5

El número de frases reconocidas fueron 125 en el experimento dependiente y 98 en el independiente. Sólo una de estas frase produjo una celosía que no fue aceptada por el procesador lingüístico.

VI. Conclusiones.

En esta comunicación se ha presentado la primera fase de desarrollo de un sistema automático de reconocimiento del habla en dos etapas: un procesador acústico basado en un detector de palabras y un procesador lingüístico regido por una generalización del algoritmo CKY. Un elemento que pareció ser crítico en algún experimento aislado fue el tamaño del cuanto utilizado en la cuantización de la duración de cada palabra presente en la celosía. Es por tanto de gran interés el estudiar métodos consistentes para relajar la condición de continuidad temporal entre palabras consecutivas. Actualmente estamos aplicando un método basado en unión y poda de agrupaciones de las hipótesis que proporciona el procesador acústico de una misma palabra. Este procedimiento ha proporcionado buenos resultados en un sistema similar a este pero cuya segunda etapa está regida por una gramática regular[10]. También estamos considerando la utilización de restricciones en la aplicación de las reglas para tratar el problema de la coarticulación. Tal y como se explica al describir el conocimiento léxico, una palabra está representada por todas las posibles cadenas de unidades fonéticas bajo las que la palabra puede aparecer. Sin embargo, algunas de estas cadenas sólo son posibles en determinados contextos. En este momento el sistema no utiliza esta información lo que puede ser una fuente de errores, especialmente para palabras cortas (en nuestro caso la palabra "i").

VII. Referencias.

[1] K. Kita, T. , H. Saito: "HMM Continuous Speech Recognition Using Predictive LR Parsing",Proc. ICASSP 89, pp. 703-706

[2] R. Schwartz, Y. Chow: "The N-Best Algorithm: An Efficient and Exact Procedure for finding the N-Most Likely Sentence Hypotheses", Proc ICASSP-90, pp. 81-84.

[3] José B. Mariño, Enric Monte, "Generation of Multiple Hipothesis in Connected Phonetic-Units Recognition by a modified One-Stage Dynamic Programming Algorithm," EUROSPEECH 89, pag 408-411

[4] L. Lleida, et al: "Demisyllable-Based HMM Spotting for Continuous Speech Recognition", Proc. ICASSP-91, pp. 709-712

[5] A.V. Aho, J.D. Ullman: *The Theory of Parsing, Translation and Compiling*, Prentince-Hall, Englewood Cliffs, N.J., 1972. pp. 314-320.

[6] R.C. Gonzalez, M.G. Thomason: *Syntactic Pattern Recognition. An Introduction*, Addison-Wesley , 1978, pp. 20-31,46-56.

[7] J.G. Carbonell, P.J. Hayes, "Recovery Strategies for Parsing Extragrammatical Language", Tech. report CMU-CS-84-107, Carniege-Mellon University Computer Science Technical Report, 1984

[8] J.B. Mariño et al.: "Recognition of Numbers by Using Demysillables and Hidden Markov Models", Proc. EUSIPCO-90, pp. 1363-1366.

[9] J.B.Mariño et al: "RAMSES, a Spanish Demisyllable Based Continuous Speech Recognition System," NATO-ASCI-90, Julio,1990

[10] E. Lleida et al: "Two Level Continuous Speech Recognition Using Demisyllable-Based HMM Word Spotting", Proc EUROSPEECH-91