Instituto Politécnico Nacional

Centro de Investigación en Computación

Secretaría de Investigación y Posgrado

ETIQUETADOR SEMIAUTOMÁTICO FONÉTICO DE UN CORPUS DE VOCES

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS EN INGENIERÍA DE CÓMPUTO CON OPCIÓN EN SISTEMAS DIGITALES

P R E S E N T A

ING. CRISTIAN REMINGTON JUÁREZ MURILLO

DIRECTOR DE TESIS:
DR. SERGIO SUÁREZ GUERRA

MÉXICO, D.F. JUNIO 2012

SIP-14

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 12:00 horas del día 27 del mes de abril de 2012 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Etiquetador semiautomático fonético de un corpus de voces"

Presentada por el alumno:

JUÁREZ	MURILLO	CRISTIAN REMINGTON
Apellido paterno	Apellido materno	Nombre(s)

Con registro: A 1 0 0 2 8 3

aspirante de: **MAESTRÍA EN CIENCIAS EN INGENIERÍA DE CÓMPUTO CON OPCIÓN EN SISTEMAS DIGITALES**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director de Tesis

Dr. Sergio Suárez Guerra

Dr. Oleksiy Pogrebnyak

Dr. Luis Pastor Sánchez Fernández

Dr. José Luis Oropeza Rodríguez

M. en C. Pablo Manrique Ramírez

Dr. Ricardo Barrón Fernández

PRESIDENTE DEL COLEGIO DE PROFESORES

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
DIRECCION

Dr. Luis Alfonso Villa Vargas

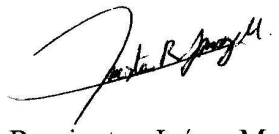
INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA DE CESIÓN DE DERECHOS

En la Ciudad de México el día 24 del mes mayo del año 2012, el que suscribe Cristian Remington Juárez Murillo alumno del Programa de Maestría en ciencias en ingeniería de cómputo con opción en sistemas digitales con número de registro A100283, adscrito al Centro de Investigación en Computación, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del Dr. Sergio Suárez Guerra y cede los derechos del trabajo titulado Etiquetador semiautomático fonético de un corpus de voces, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección cristianremingtonjm@gmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



Cristian Remington Juárez Murillo

Nombre y firma

RESUMEN

Se desarrolló un sistema para la segmentación y etiquetado semiautomático a nivel de fonemas para corpus de palabras implementado en Matlab. El sistema está basado en el algoritmo de alineación forzada, el cual utiliza modelos ocultos de Markov discretos, por lo cual se necesita comenzar con una fracción de archivos etiquetados y segmentados manualmente de manera previa. Dichos archivos deben incluir todos los fonemas presentes en el corpus para poder entrenar los modelos de Markov. También se realizaron experimentos para determinar si la segmentación acústica mejoraba la segmentación producida por alineación forzada; aunque los experimentos reportan que es mejor utilizar la alineación forzada sola.

Se utilizaron dos corpus. El primer corpus cuenta con todos los fonemas hablados en México; consiste de 150 tipos de palabras distintas repetidas 4 veces cada una por un solo hablante (600 archivos en total). El segundo corpus consiste de los dígitos del cero al nueve (diez tipos de palabras), repetidas 20 veces por cuatro hablantes, exceptuando las palabras cuatro y siete, las cuales fueron repetidas 19 veces (798 archivos en total). También se trabajó con los corpus de dígitos por separado, un hablante por corpus.

Se realizaron dos tipos de experimentos para evaluar la calidad de la segmentación automática: por comparación directa (por concordancia con respecto a las etiquetas manuales) y por comparación indirecta (por reconocimiento).

Se midió la concordancia entre los archivos de etiquetas manuales y automáticos a través del porcentaje de fronteras correctamente colocadas para cinco valores de tolerancia (10 ms, 20 ms, 30 ms, 40 ms y 50 ms), donde el valor más comúnmente utilizado por los investigadores del tema, es el de 20 ms. Para el corpus con todos los fonemas, se entrenó la alineación forzada con todos los archivos de etiquetas manuales, y la mejor concordancia lograda fue de 89.07% dentro de 20 ms; mientras que para el corpus de dígitos sólo se utilizó un cuarto de los archivos de etiquetas manuales para entrenar la alineación forzada, logrando una concordancia de 66.67% dentro de 20 ms. Los corpus de dígitos por separado alcanzaron respectivamente, 88.75%, 93.1179%, 82.2033% y 84.7222% de concordancia dentro de 20 ms.

Los archivos de etiquetas automáticos generados se utilizaron para entrenar palabras con el sistema de reconocimiento en HTK. Los porcentajes de reconocimiento obtenidos mediante sistemas entrenados con etiquetas automáticas se compararon con los porcentajes de reconocimiento obtenidos mediante etiquetas manuales. Para el corpus con todos los fonemas, el sistema entrenado con etiquetas manuales reportó un porcentaje de reconocimiento de 98.61%; mientras que el sistema entrenado con etiquetas automáticas reportó 98.50%. Para el corpus de dígitos, con todos los hablantes mezclados, el sistema entrenado con etiquetas manuales reportó un porcentaje de reconocimiento de 95.24%, mientras que el sistema entrenado con etiquetas automáticas reportó 96.74%. Por otro lado, los corpus de dígitos por separado (un hablante por corpus) reportaron respectivamente: 100% (manuales) contra 100% (automáticas); 92.95% (manuales) contra 100% (automáticas); 97.96% (manuales) contra 98.98% (automáticas); y 97.92% (manuales) contra 100% (automáticas). Estos valores nos indican que los sistemas de reconocimiento entrenados con etiquetas automáticas producen resultados mejores o similares que los sistemas entrenados con etiquetas manuales. Además, dado que el etiquetado manual siempre es inconsistente, intentar hacer que el etiquetado automático sea idéntico al etiquetado manual no tiene sentido, porque en ese caso el etiquetado automático sería inconsistente también.

El etiquetado manual de un corpus es una tarea tediosa, propensa a errores e inconsistente. El etiquetado automático ahorra una gran cantidad de tiempo y es consistente.

ABSTRACT

A system for semiautomatic labeling and segmentation of word corpora at phoneme level was implemented in Matlab. The system is based on the Forced Alignment algorithm which uses discrete Hidden Markov Models; therefore, it needs handmade label files to start. Those training files must include all the phonemes present in the corpus in order to train the Markov models. Also, experiments were performed to determine whether or not acoustic segmentation helped the forced alignment algorithm; however, experiments show the forced alignment on its own is better.

Two corpora were used. The first corpus has all the phonemes spoken in Mexico; it contains 150 different types of words, repeated 4 times each by a single speaker (which accounts for 600 files). The second corpus consists of digits from 0 to 9 (ten types of different words) repeated 20 times each by four speakers with the exception of words four and seven, which were repeated 19 times. All this accounts for 798 files. Also, each corpus was used separately, one speaker by corpus.

Two types of experiments were performed in order to assess the quality of the automatic segmentation: direct comparison (by measuring the agreement among the automatic and the handmade labels) and indirect comparison (by performing recognition experiments).

The agreement among automatic and handmade label files was measured using a percentage of the number of correctly placed edges within five values of tolerance (10 ms, 20 ms, 30 ms, 40 ms and 50 ms). The most commonly used tolerance value is 20 ms. For the experiments regarding the phonetically complete corpus, forced alignment was trained by using all the handmade label files and the best agreement achieved was 89.07% within a tolerance of 20 ms. For the experiments regarding the digits corpus, forced alignment was trained by using only a fourth of the handmade label files, which yielded an agreement of 66.67% within 20 ms. In addition, each digit corpus with a single speaker yielded 88.75%, 93.1179%, 82.2033% and 84.7222% within 20 ms, respectively.

The automatic label files generated were used to train speech recognition systems in HTK software. The recognition percentages obtained by systems trained with handmade label files were compared to those obtained by systems trained with automatic label files. The system trained with handmade labels for the phonetically complete corpus yielded a recognition percentage of 98.61%, while the system trained with automatic label files for that same corpus yielded 98.50%. The system trained with handmade label files for the digit corpus (with four speakers mixed) yielded a recognition percentage of 95.24%, while the system trained with automatic label files for that corpus yielded 96.74%. Also, each digit corpus with a single speaker yielded, respectively, 100% (handmade) vs. 100% (automatic); 92.95% (handmade) vs. 100% (automatic); 97.96% (handmade) vs. 98.98% (automatic); and 97.92% (handmade) vs. 100% (automatic).These values show that speech recognition systems trained with automatic label files yield a better or similar performance in comparison to those trained with handmade label files. In addition, since handmade labeling is always inconsistent, trying to make the automatic labeling to be identical to the handmade labeling makes no sense because automatic labeling would be inconsistent too.

Handmade labeling of a speech corpus is a tedious, error prone and inconsistent task. Automatic labeling saves a large amount of time and it’s also consistent.

AGRADECIMIENTOS

Les agradezco a mi madre Carmen Dolores Murillo Muñoz y a mi padre Jesús Juárez Flores por su apoyo incondicional, por creer siempre en mí, por darme educación, por procurar mi bienestar y por todo el cariño que me han dado. También le agradezco a mi hermano Roger Onasis Juárez Murillo por ser un gran amigo en quien siempre puedo confiar.

Le agradezco al Dr. Sergio Suárez Guerra por la atenta orientación que me brindó en la elaboración de este trabajo de tesis.

Agradezco a CONACYT.

Agradezco al Centro de Investigación en Computación y al Instituto Politécnico Nacional.

ÍNDICE GENERAL

ÍNDICE GENERAL I
ÍNDICE DE FIGURAS IV
ÍNDICE DE TABLAS VI
GLOSARIO VIII
CAPÍTULO 1. INTRODUCCIÓN 1
1.1 Antecedentes 1
1.2 Planteamiento del problema 2
1.3 Objetivos 3
1.4 Justificación 3
1.5 Hipótesis 3
1.6 Solución propuesta 3
1.7 Alcances del trabajo 4
1.8 Contribuciones 4
1.9 Organización del trabajo 4
CAPÍTULO 2. ESTADO DEL ARTE 6
2.1 Introducción 6
2.2 Métodos automáticos de segmentación 6
2.2.1 Segmentación fonética manual 7
2.2.2 Segmentación por dynamic time warping (DTW) 8
2.2.3 Segmentación por algoritmos evolutivos 8
2.2.4 Segmentación por técnicas de estudio de fractales 9
2.2.5 Segmentación por técnicas difusas 9
2.2.6 Segmentación por técnicas conexionistas 9
2.2.7 Segmentación por técnicas estocásticas 10
2.2.8 Segmentación por técnicas wavelet 11
2.3 Resumen 12
CAPÍTULO 3. MARCO TEÓRICO (MATERIALES Y MÉTODOS) 13
3.1 Introducción 13
3.2 El problema de segmentación-etiquetado 13
3.2.1 Evaluación de la segmentación 13
3.3 Fonética y fonología 15
3.3.1 Alfabetos fonéticos 15
3.3.2 Diccionarios de pronunciación 16
3.4 Escala de Bark 16
3.5 Transformada wavelet 17
3.5.1 Soporte de una función 18
3.5.2 El principio de incertidumbre de Heisenberg 18
3.5.3 Transformada wavelet continua 18
3.5.4 Noción de escala 19
3.5.5 Propiedades de las funciones wavelet 19
3.5.6 Wavelets Gaussianas moduladas 19
3.5.7 Obtención de los parámetros de escala y modulación 20
3.6 Algoritmo de segmentación wavelet de Galka-Ziolko 21
3.6.1 Pasos del algoritmo de Galka-Ziolko 22
3.7 Métrica y distorsión 23
3.8 Alineación forzada 24
3.8.1 Modelos ocultos de Markov 24
3.8.2 Algoritmo de Viterbi 25
3.9 Conclusiones 26
3.10 Resumen 26
CAPÍTULO 4. METODOLOGÍA Y DESARROLLO DE LA INVESTIGACIÓN 28
4.1 Introducción 28
4.2 Alfabeto fonético 28
4.3 Clases de fonemas y alófonos usados 28
4.4 Diccionario de pronunciaciones 28

4.5 Corpus 28
4.6 Procedimiento para el etiquetado manual 29
4.7 Opinión de un experto fonetista 31
4.8 Preprocesamiento 31
4.8.1 Preénfasis 32
4.8.2 Normalización 32
4.8.3 Detección de inicio y fin de palabra 32
4.9 Segmentación wavelet 32
4.9.1 Modificación al algoritmo de Galka y Ziolko 32
4.9.2 Cálculo de la CWT 33
4.9.3 Cálculo de la envolvente de los coeficientes wavelet 34
4.9.4 Suavizado de la envolvente de los coeficientes wavelet 34
4.9.5 Cálculo del mapa de importancia 34
4.9.6 La función de detección de eventos 34
4.9.7 Cálculo de los umbrales 35
4.10 Alineación forzada 35
4.10.1 Modelos ocultos de Markov discretos 35
4.10.2 Modelado con tres estados 35
4.10.3 Procedimiento para obtener secuencias de entrenamiento 37
4.10.4 Procedimiento para obtener el libro de código 37
4.10.5 HMM Toolbox 39
4.10.6 Procedimiento para entrenar el HMM 39
4.10.7 Algoritmo de Viterbi 41
4.11 Diseño del sistema propuesto 43
4.12 Programas realizados en Matlab 45
4.12.1 CalcularVectoresPorFonemas 45
4.12.2 ConcatenarVectoresV2 46
4.12.3 CrearLibroCodigo 47
4.12.4 EntrenarHMMFonemas 47
4.12.5 EntrenarHMMPalabrasV2 48
4.12.6 AlinearPalabrasV2 50
4.12.7 CompararArchivosEtiquetas 51
4.12.8 CompararArchivosEtiquetasV2 52
4.12.9 GuardarAnalisisMatrizFronteras 53
4.12.10 phn2lab 53
4.12.11 CompararArchivosEtiqAcusticas 54
4.13 Evaluación de los archivos de etiquetas 55
4.14 Resumen 55
CAPÍTULO 5. PRESENTACIÓN Y DISCUSIÓN DE RESULTADOS 56
5.1 Introducción 56
5.2 Segmentación acústica mediante wavelets 56
5.2.1 Obtención de los parámetros del algoritmo 56
5.2.2 Experimento 1 57
5.2.3 Experimento 2 62
5.2.4 Experimento 3 63
5.3 Experimentos con el corpus de 150 tipos de palabras 64
5.3.1 Experimento 1. Reconocimiento con etiquetas manuales 65
5.3.2 Experimento 2. Generación de etiquetas con alineación forzada 65
5.3.3. Experimento 3. Fronteras mejor detectadas mediante segmentación acústica 67
5.3.4. Experimento 4. Generación de etiquetas con alineación forzada y segmentación acústica 68
5.3.5. Experimento 5. Generación de etiquetas con alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras 69
5.4 Experimentos con el corpus de dígitos mezclados 70
5.4.1 Experimento 1. Reconocimiento con etiquetas manuales 70
5.4.2 Experimento 2. Generación de etiquetas con alineación forzada 71
5.4.3 Experimento 3. Generación de etiquetas con alineación forzada y segmentación acústica 72
5.5 Experimentos con cuatro corpus de dígitos de un solo hablante 74
5.5.1 Corpus1 francovoz. Etiquetas manuales. Parte baja del corpus 75
5.5.2 Corpus1 francovoz. Etiquetas automáticas. Parte baja del corpus 76
5.5.3 Corpus2 lisethvoz. Etiquetas manuales. Parte baja del corpus 77

5.5.4 Corpus2 lisethvoz. Etiquetas automáticas. Parte baja del corpus 78
5.5.5 Corpus2 lisethvoz. Etiquetas manuales. Parte alta del corpus 78
5.5.6 Corpus2 lisethvoz. Etiquetas automáticas. Parte alta del corpus 79
5.5.7 Corpus3 rodrigovoz. Etiquetas manuales. Parte baja del corpus 79
5.5.8 Corpus3 rodrigovoz. Etiquetas automáticas. Parte baja del corpus 80
5.5.9 Corpus3 rodrigovoz. Etiquetas manuales. Parte alta del corpus 80
5.5.10 Corpus3 rodrigovoz. Etiquetas automáticas. Parte alta del corpus 81
5.5.11 Corpus4 sergiovoz. Etiquetas manuales. Parte baja del corpus 81
5.5.12 Corpus4 sergiovoz. Etiquetas automáticas. Parte baja del corpus 82
5.5.13 Corpus4 sergiovoz. Etiquetas manuales. Parte alta del corpus 82
5.5.14 Corpus4 sergiovoz. Etiquetas automáticas. Parte alta del corpus 83
5.6 Comparación de los resultados 83
5.7 Resumen 85
CONCLUSIONES Y TRABAJO A FUTURO 87
Objetivos cumplidos 87
Ventajas 87
La propuesta de segmentación acústica 88
Resultados de reconocimiento para los corpus 88
Comparación con el estado del arte 89
Trabajo a futuro 89
Trabajos derivados 90
REFERENCIAS 91
ANEXO A. ALFABETOS FONÉTICOS 96
A.1 El alfabeto fonético del corpus TIMIT 96
A.2 IPA 96
A.3 Arpabet 97
A.4 Sampa 97
A.5. RFE 98
A.6 OGIbet 101
A.7 Worldbet 101
A.8. Mexbet 102
ANEXO B. DICCIONARIO DE PRONUNCIACIONES 103
B.1. Características del diccionario para el corpus con todos los fonemas 103
B.2. Diccionario de pronunciaciones para el corpus con todos los fonemas 104
B.3 Características del diccionario para el corpus de dígitos 106
B.4 Diccionario de pronunciaciones para el corpus de dígitos 106
ANEXO C. ARCHIVOS DE HTK 107
C.1 Introducción 107
C.2 Experimentos con el corpus con todos los fonemas 107
C.2.1 Experimento 1. Reconocimiento con etiquetas manuales 107
C.2.2 Experimento 2. Reconocimiento con etiquetas con alineación forzada 119
C.2.3 Experimento 4. Reconocimiento con etiquetas con alineación forzada y segmentación acústica 121
C.2.4 Experimento 5. Reconocimiento con etiquetas con alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras 123
C.3 Experimentos con el corpus de dígitos 126
C.3.1 Experimento 1. Reconocimiento con etiquetas manuales 126
C.3.2 Experimento 2. Reconocimiento con etiquetas con alineación forzada 136
C.3.3 Experimento 3. Reconocimiento con etiquetas con alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras 139

ÍNDICE DE FIGURAS

Figura 1.1. Diagrama de bloques de un sistema segmentador etiquetador de habla. 2
Figura 1.2. Segmentación y etiquetado utilizando reconocimiento de voz [27]. 2
Figura 1.3. Diseño del sistema propuesto. 4
Figura 2.1. Herramienta SpeechViewer. 7
Figura 2.2. Archivo de etiquetas. 8
Figura 3.1. Esquema de análisis con wavelets Gaussianas moduladas en la escala de Bark. 21
Figura 4.1.a. Fronteras sencillas entre /a/ y /f/. 30
Figura 4.1.b. Frontera sencilla entre /a/ y /n/. 30
Figura 4.2.a. Frontera difícil de marcar entre las vocales /i/ y /a/. 31
Figura 4.2.b. Frontera difícil de marcar entre /dʒ/ y /o/. 31
Figura 4.3. Modelo de Markov de tres estados. 36
Figura 4.4. Modelo de Markov de tipo Bakis con estados *dummy*. 36
Figura 4.5. Procedimiento para obtener secuencias de entrenamiento. 37
Figura 4.6. Árbol binario completo. 38
Figura 4.7. Procedimiento para crear un libro de código y almacenarlo en un archivo. 39
Figura 4.8. Procedimiento para entrenar los HMM a nivel de fonemas. 40
Figura 4.9. Procedimiento para entrenar los HMM a nivel de palabras. 41
Figura 4.10. Procedimiento para generar las etiquetas automáticas. 42
Figura 4.11. Alineación forzada con el algoritmo de Viterbi. 42
Figura 4.12. Diagrama de bloques del sistema propuesto. 43
Figura 4.13. Selección de fronteras. 45
Figura 5.1. Porcentaje de reconocimiento para el sistema entrenado con etiquetas manuales. 65
Figura 5.2. Porcentaje de reconocimiento para un sistema entrenado con etiquetas elaboradas con alineación forzada. 66
Figura 5.3. Porcentaje de reconocimiento para un sistema entrenado con etiquetas elaboradas con alineación forzada y segmentación acústica. 69
Figura 5.4. Porcentaje de reconocimiento para un sistema entrenado con etiquetas elaboradas con alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras. 70
Figura 5.5. Porcentaje de reconocimiento para un sistema entrenado con etiquetas manuales. 71
Figura 5.6. Porcentaje de reconocimiento para un sistema entrenado con etiquetas elaboradas por alineación forzada. 72
Figura 5.7. Porcentaje de reconocimiento para un sistema entrenado con etiquetas elaboradas por alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras. 73
Figura 5.8. Para cada corpus se realizaron cuatro experimentos de reconocimiento. 74
Figura 5.9. Resultados de reconocimiento para el corpus francovoz con etiquetas manuales en la parte baja del corpus. 75
Figura 5.10. Resultados de reconocimiento para el corpus francovoz con etiquetas automáticas en la parte baja del corpus. 76
Figura 5.11. Resultados de reconocimiento para el corpus francovoz con etiquetas manuales en la parte alta del corpus. 76
Figura 5.12. Resultados de reconocimiento para el corpus francovoz con etiquetas automáticas en la parte alta del corpus. 77
Figura 5.13. Resultados de reconocimiento para el corpus lisethvoz con etiquetas manuales en la parte baja del corpus. 77
Figura 5.14. Resultados de reconocimiento para el corpus lisethvoz con etiquetas automáticas en la parte baja del corpus. 78
Figura 5.15. Resultados de reconocimiento para el corpus lisethvoz utilizando etiquetas manuales en la parte alta del corpus. 78
Figura 5.16. Resultados de reconocimiento del corpus lisethvoz utilizando etiquetas automáticas en la parte alta del corpus. 79
Figura 5.17. Resultados de reconocimiento para el corpus rodrigovoz usando etiquetas manuales en la parte baja del corpus. 79
Figura 5.18. Resultados de reconocimiento para el corpus rodrigovoz usando etiquetas automáticas en la parte baja del corpus. 80
Figura 5.19. Resultados de reconocimiento para el corpus rodrigovoz con etiquetas manuales en la parte alta del corpus. 80

Figura 5.20. Resultados de reconocimiento para el corpus rodrigovoz con etiquetas automáticas en la parte alta del corpus. 81
Figura 5.21. Resultados de reconocimiento para el corpus sergiovoz con etiquetas manuales en la parte baja del corpus. 81
Figura 5.22. Resultados de reconocimiento para el corpus sergiovoz usando etiquetas automáticas en la parte baja del corpus. 82
Figura 5.23. Resultados de reconocimiento para el corpus sergiovoz con etiquetas manuales en la parte alta del corpus. 82
Figura 5.24. Resultados de reconocimiento para el corpus sergiovoz usando etiquetas automáticas en la parte alta del corpus. 83
Figura A.1.a. Primera parte del alfabeto fonético de la RFE [47]. 99
Figura A.1.b. Segunda parte del alfabeto fonético de la RFE [47]. 100

ÍNDICE DE TABLAS

Tabla 3.1. Frecuencia central y ancho de banda para cada banda crítica de acuerdo con la escala de Bark [32]……..…….....17
Tabla 3.2. Parámetros de escala y de frecuencia de modulación [32]…… 21
Tabla 5.1. Valores de ϵ_n para cuando N corresponde a 7 ms. Incrementar α disminuye este error…….....57
Tabla 5.2. Valores de ϵ_p' para cuando N corresponde a 7 ms. Disminuir α disminuye este error.....58
Tabla 5.3. Valores de Total_ϵ para cuando N corresponde a 7 ms. Algunos valores de α y N producen valores mínimos en el error……58
Tabla 5.4. Valores de ϵ_n para cuando N corresponde a 10 ms. Incrementar α disminuye este error…….....59
Tabla 5.5. Valores de ϵ_p' para cuando N corresponde a 10 ms. Disminuir α disminuye este error…… 59
Tabla 5.6. Valores de Total_ϵ para cuando N corresponde a 10 ms. Algunos valores de α y N producen valores mínimos en el error…….....60
Tabla 5.7. Valores de ϵ_n para cuando N corresponde a 15 ms. Incrementar α disminuye este error…….....60
Tabla 5.8. Valores de ϵ_p' para cuando N corresponde a 15 ms. Disminuir α disminuye este error…… 61
Tabla 5.9. Valores de Total_ϵ para cuando N corresponde a 15 ms. Algunos valores de α y N producen valores mínimos en el error…….....61
Tabla 5.10 Comparación de las segmentaciones acústicas…….....63
Tabla 5.11. Comparación de la segmentación acústica original y la segmentación acústica propuesta…… 64
Tabla 5.12. Etiquetas generadas por alineación forzada comparadas contra las etiquetas manuales……66
Tabla 5.13. Tipos de fronteras donde las propuestas acústicas están a menos de 10 ms respecto las fronteras manuales al menos el 70% de las veces acomodadas en orden alfabético…… 67
Tabla 5.14. Etiquetas generadas por alineación forzada y segmentación acústica, comparadas contra las etiquetas manuales…… 69
Tabla 5.15. Etiquetas generadas por alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras, comparadas contra las etiquetas manuales…….....69
Tabla 5.16. Etiquetas generadas por alineación forzada comparadas contra las etiquetas manuales……72
Tabla 5.17. Etiquetas generadas por alineación forzada y segmentación acústica comparadas contra las etiquetas manuales…… 73
Tabla 5.18. Concordancia del etiquetado automático respecto al etiquetado manual para cada corpus de dígitos……75
Tabla 5.19. Resultados de experimentos para el corpus con todos los fonemas…….. 83
Tabla 5.20. Resultados de experimentos para el corpus de dígitos……84
Tabla 5.21.a. Resultados de reconocimiento para el corpus franciscovoz……84
Tabla 5.21.b. Resultados de concordancia para el corpus franciscovoz…… 84
Tabla 5.22.a. Resultados de reconocimiento para el corpus lisethvoz…….. 84
Tabla 5.22.b. Resultados de concordancia para el corpus lisethvoz…….. 85
Tabla 5.23.a. Resultados de reconocimiento para el corpus rodrigovoz……85
Tabla 5.23.b. Resultados de concordancia para el corpus rodrigovoz…….. 85
Tabla 5.24.a. Resultados de reconocimiento para el corpus sergiovoz…… 85
Tabla 5.24.b. Resultados de concordancia para el corpus sergiovoz……85
Tabla A.1. Fonemas del español con símbolos IPA…….. 96
Tabla A.2. SAMPA para español…….. 97
Tabla A.3. Símbolos Worldbet para el español (tomada de [26])……..... 101

Tabla B.1. Fonemas en el corpus propuesto…….. 103
Tabla B.2. Diccionario de pronunciaciones para el corpus fonéticamente completo creado……104
Tabla B.3. Fonemas en el corpus de dígitos…….. 106
Tabla B.4. Transcripciones del corpus de dígitos en alfabeto worldbet……... 106

GLOSARIO

Alfabeto fonético

Los alfabetos fonéticos son alfabetos especializados para transcribir los sonidos del habla. Son conjuntos de caracteres que representan símbolos fonéticos. Entre los alfabetos fonéticos se tienen: el alfabeto de la revista de filología española (RFE), el alfabeto fonético internacional (IPA), TIMITBET, SAMPA (del inglés *Speech Assessment Methods Phonetic Alphabet*), SAMPA extendido, Worldbet, OGIbet, etc.

Algoritmos evolutivos

La programación evolutiva es un método para simular la evolución. Existen tres líneas de investigación en la computación evolutiva: algoritmos genéticos, estrategias de evolución, y programación evolutiva. Todos los métodos de computación evolutiva requieren una búsqueda iterativa basada en una población, con variación aleatoria y selección [19].

Alineación forzada

La alineación forzada (del inglés *forced alignment*) es el método más comúnmente usado para la alineación automática de habla. En dicho método se realiza un reconocimiento de habla restringiendo la búsqueda a la secuencia conocida de fonemas. La búsqueda produce la localización de los fonemas, así como su identidad. Estos sistemas son llamados de alineación forzada porque se obliga a que el resultado del reconocimiento sea la secuencia fonética propuesta, la cual se determina previamente utilizando un diccionario de pronunciaciones, reglas grafema a fonema o por un humano.

Alófono

Cada una de las variantes que se dan en la pronunciación de un mismo fonema, según la posición de este en la palabra o sílaba y según el carácter de los fonemas vecinos. Por ejemplo la b oclusiva en tumbo y la b fricativa en tubo son ambos alófonos del fonema /b/.

ANN

Una red neuronal artificial (del inglés, Artificial Neural Network) es un procesador distribuido paralelo que se compone de unidades de procesamiento simple, propenso a almacenar conocimiento experimental y hacerlo disponible para su uso. La red adquiere el conocimiento de su ambiente a través de un proceso de aprendizaje. Las conexiones entre neuronas, llamadas pesos sinápticos almacenan el conocimiento adquirido [25].

ASR

Del inglés *Automatic Speech Recognition*, o reconocimiento automático del habla. Es el proceso de extraer automáticamente y determinar información lingüística (también llamada información fonética) en una onda de habla, utilizando para ello computadoras o circuitos electrónicos [21].

Bark, escala de

La escala de Bark relaciona las frecuencias acústicas con frecuencias perceptuales. La escala de Bark Toma en cuenta el fenómeno del enmascaramiento, según el cual la percepción de ciertos sonidos es impedida por la presencia de otros sonidos. La escala de Bark se basa en bandas críticas; si hay dos sonidos en una banda crítica, el que tenga más energía enmascara al otro.

Conexionista, enfoque

El enfoque conexionista es una de las cuatro escuelas del pensamiento en el área del reconocimiento del habla. El enfoque conexionista hace uso de nodos simples cuyas conexiones se entrenan para reconocer habla (redes neuronales artificiales). Los modelos conexionistas dependen de tener una buena estrategia de entrenamiento. Los modelos conexionistas no necesitan hacer suposiciones acerca de las funciones de densidad de probabilidad [58].

Corpus

Corpus es una base de datos de habla, es una colección de habla grabada que cuenta con anotaciones y transcripciones necesarias. Hay tres categorías de corpus: diagnóstico analítico, genérico, y específico. Un corpus se puede construir leyendo fonemas aislados, palabras aisladas, frases aisladas, leyendo fragmentos de un texto, con habla semiespontánea, habla espontánea con un tema predeterminado y habla provocativa con el método de mago de Oz [4].

CSLU

Del inglés *Center for Spoken Language Understanding*, es un centro de investigación de la *Oregon Health and Science University*, que se enfoca en la investigación de algoritmos para la tecnología del habla y el lenguaje, imágenes y biología. El centro desarrolló el CSLU Toolkit, un conjunto de herramientas que habilitan la exploración, aprendizaje e investigación del habla e interacción hombre-computadora. El toolkit de CSLU cuenta con herramientas de etiquetado, visualización, audio, reconocimiento de habla, generación de habla y animación de rostros.

Diccionario de pronunciación

Es un conjunto de transcripciones fonéticas, que muestra las pronunciaciones consideradas para todas las palabras con las cuales trabajará el sistema ASR.

Difuso, modelo

Los modelos difusos son estructuras de modelado lingüísticas no numéricas con bloques funcionales bien definidos de interfaces de entradas y de salidas junto con un módulo de procesamiento [36].

Dynamic Time Warping (DTW)

Es una técnica basada en programación dinámica la cual expande o contrae el eje del tiempo de una locución de manera no lineal para que los mismos fonemas coincidan en sus respectivas posiciones tanto

en el habla de entrada como en las plantillas de referencia. Esto es necesario porque aunque el mismo hablante pronuncie dos veces la misma palabra la duración cambiará de forma no lineal con expansiones y contracciones [20].

Espectro

El término espectro hace referencia al contenido en frecuencia de una señal. El análisis en frecuencia, o análisis espectral es el proceso que permite obtener el espectro de una señal dada. El proceso de determinar el espectro de una señal en la práctica usando medidas reales, se llama estimación espectral [41].

Estocástico, enfoque

Es una de las escuelas del pensamiento en el área de reconocimiento de habla. El enfoque estocástico utiliza modelos probabilísticos para tratar información incierta o incompleta. El enfoque estocástico se adecúa bien al reconocimiento de habla, ya que en el habla existen diversas fuentes de variabilidad, como sonidos que se confunden fácilmente, variabilidad entre hablantes, efectos de contexto, palabras homófonas, etc [58].

Etiquetado

El etiquetado asocia un símbolo fonético a cada unidad segmental de una locución [46].

Fonema

Cada una de las unidades fonológicas mínimas que en el sistema de una lengua pueden oponerse a otras en contraste significativo, por ejemplo las consonantes iniciales de pozo y gozo. Dentro de cada fonema caben varios alófonos.

Fractal

Mandelbrot acuñó el término “fractal” para describir objetos demasiado irregulares para encajar en el conjunto geométrico tradicional. Un fractal es un conjunto que es autosimilar, tiene estructura fina (contiene detalles a un nivel arbitrario de escala), se obtiene de manera recursiva, el conjunto es incontablemente infinito [14].

Fricativas

Son los fonemas correspondientes a los sonidos que se producen al realizar un estrechamiento entre dos órganos articulatorios, lo cual genera una fricación. Los fonemas fricativos en español son /f/, /θ/, /s/, /j/, y /x/ [5].

Gaussiana modulada, wavelet

Janer [32] escogió para su tesis doctoral una wavelet Gaussiana modulada, a la cual le agregó un modulador para centrarla en las bandas críticas de la escala de Bark, tomando en cuenta al mismo tiempo el ancho de banda de la wavelet (ver escala de Bark).

HMM

Un modelo oculto de Markov (en inglés, *Hidden Markov Model*) es un autómata finito estocástico utilizado para modelar habla. La locución a modelar puede ser una palabra, una unidad más pequeña que la palabra, o inclusive una oración o un párrafo. El modelo está compuesto por las probabilidades de transiciones entre estados, las probabilidades de estados iniciales y las funciones de densidad de probabilidad de observaciones para cada estado [11].

HTK

Es un conjunto de herramientas para construir modelos ocultos de Markov (véase HMM). El núcleo de HTK es de propósito general pero está orientado principalmente a aplicaciones de reconocimiento de habla [61].

IPA, asociación

La asociación fonética internacional (del inglés, *International Phonetic Association*) es la organización más antigua y representativa para los fonetistas. El objetivo de la asociación es promover el estudio científico de la fonética y las aplicaciones prácticas de dicha ciencia. En fomento a dicho objetivo, la asociación IPA provee a la comunidad académica con una notación estándar para la representación fonética de todos los lenguajes [53].

IPA, alfabeto

Es el alfabeto establecido por la asociación fonética internacional para representar todos los sonidos de todas las lenguas del mundo [8].

Locución

Acto de hablar.

MLF

Master Label File. Es un tipo de archivos que utiliza HTK para ingresar transcripciones.

MMF

Master Macro File. Es un tipo de archivos que utiliza HTK para reunir varios modelos ocultos de Markov en un solo archivo.

Plosivo

También llamado fonema oclusivo. Es el tipo de fonemas que involucra el cierre u oclusión de los órganos fonadores durante un intervalo de tiempo, seguido de su apertura con la siguiente salida brusca de aire, esto es, una explosión [5].

Segmentación

El propósito de la segmentación es dividir una señal de habla continua en unidades discretas basándose en mediciones de similitud acústica [46].

Viterbi, búsqueda.

La búsqueda Viterbi resuelve de manera eficiente el problema de encontrar la secuencia óptima de estados en un modelo oculto de Markov que explique de mejor manera una secuencia de observaciones. La búsqueda Viterbi maximiza la probabilidad de la secuencia de estados dados el modelo y la secuencia de observaciones.

WAV

El formato WAVE (a veces acortado a "WAV") pertenece a la especificación RIFF (del inglés *Resource Interchange File Format*) de Microsoft para almacenar archivos multimedia [60].

Wavelet

Es una función cuyo promedio es cero, la cual es dilatada por un parámetro de escala, y es trasladada por un parámetro de retraso. La función wavelet es un átomo de tiempo-frecuencia.
La transformada wavelet de una función f, a una escala y un desplazamiento dados se realiza correlacionando la función f con el átomo wavelet (Mallat, S., 1999).

Worldbet

Es una codificación ASCII del alfabeto fonético internacional (véase IPA), la cual incluye símbolos adicionales para etiquetar bases de datos de habla para todos los lenguajes. Worldbet es un intento por crear un alfabeto fonético que cubra todos los lenguajes del mundo de manera sistemática [26].

CAPÍTULO 1. INTRODUCCIÓN

1.1 Antecedentes

El área de investigación en el reconocimiento de voz ha madurado durante más de 70 años, persiguiendo el santo grial de este campo: una máquina capaz de reconocer cualquier persona en cualquier situación [45].

El habla ha jugado un rol primordial en la comunicación humana. Los avances tecnológicos impactan el intercambio de información. Ahora el habla está incrementando su rol en la interacción humana con sistemas de información complejos [17]. De esta forma, las tecnologías del lenguaje nos ayudan a que utilicemos las computadoras sin renunciar a nuestro uso habitual del lenguaje [18].

El estudio de las tecnologías de lenguaje humano es una empresa multidisciplinaria que requiere habilidad en áreas de lingüística, psicología, ingeniería y ciencias de la computación. Crear máquinas que interactúen con personas de manera natural utilizando lenguaje requeire un profundo conocimiento de la estructura acústica y simbólica del lenguaje (dominio de la lingüística) y de los mecanismos y estrategias que la gente utiliza para comunicarse entre sí (dominio de la psicología). Dada la extraordinaria habilidad de la gente para conversar bajo situaciones adversas como reuniones sociales ruidosas, o canales de comunicación de banda limitada, los avances en procesamiento de señales son escenciales para producir sistemas robustos (dominio de la ingeniería eléctrica). Los avances en la ciencia de la computación son necesarios para crear arquitecturas y plataformas necesarias para representar y utilizar todo este conocimiento.
Los avances en la tecnología de lenguaje humano ofrecen la promesa de acceso universal a información en línea y a servicios. Estos sistemas combinarán entendimiento del lenguaje hablado y generación del mismo, para permitirle a la gente interactuar con las computadoras utilizando el habla para obtener información sobre cualquier tema [7].

En el proceso de investigación en las áreas de reconocimiento y síntesis de voz es necesario utilizar bases de datos de archivos de audio con distintas locuciones generadas por una o más personas, esto es, corpus de voces.

A los archivos de un corpus de voces se les colocan marcas de división ya sea a nivel de palabra, de sílaba o de fonema, es decir, son segmentados. A cada uno de los segmentos resultantes debe asociársele una etiqueta con su transcripción correspondiente que indique cuál palabra, sílaba o fonema está contenida en dicho segmento, es decir, el segmento es etiquetado.

En un principio las tareas de segmentación y etiquetado debían hacerse forzosamente a mano, lo cual implicaba un gran consumo de tiempo. La segmentación y el etiquetado manual de un corpus de voces son actividades laboriosas y tediosas. Al ser tareas realizadas de forma manual, dependen de un operador, volviéndolas propensas a errores y subjetivas.

A partir de la década de 1970 y hasta la fecha de elaboración de este trabajo se han publicado diversas técnicas (ver capítulo 2, Estado del arte) y sistemas para realizar la segmentación y el etiquetado de un corpus de voces, ya sea de forma semiautomática o automática. Lo anterior es un claro indicio de que este problema sigue siendo un reto atractivo para muchos investigadores y que su solución no se ha perfeccionado.

Un software segmentador y etiquetador confiable conlleva las ventajas de ahorro de tiempo en la investigación y desarrollo de aplicaciones de reconocimiento de voz, así como la eliminación de la subjetividad que implica segmentar y etiquetar manualmente un corpus.

Un sistema segmentador y etiquetador consta típicamente de: un extractor de parámetros de la voz, un segmentador, un etiquetador y un corrector basado en reglas fonológicas [46], como aparece en la figura 1.1.

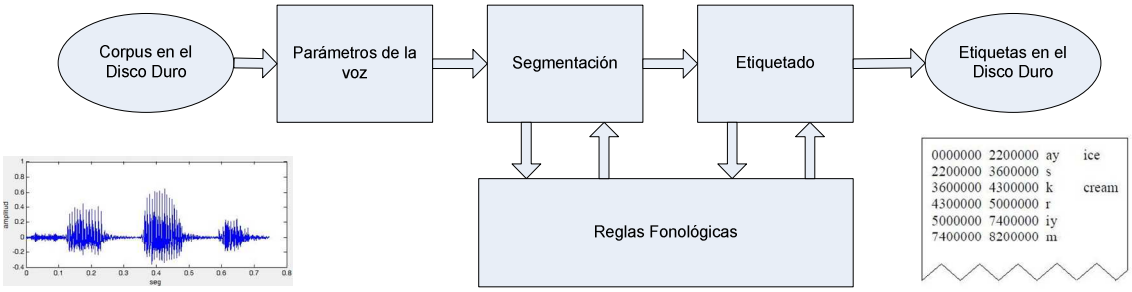


Figura 1.1. Diagrama de bloques de un sistema segmentador etiquetador de habla.

En [27] se mostró una arquitectura para realizar segmentación y etiquetado fonético utilizando un sistema de reconocimiento de voz, como se muestra en la figura 1.2.

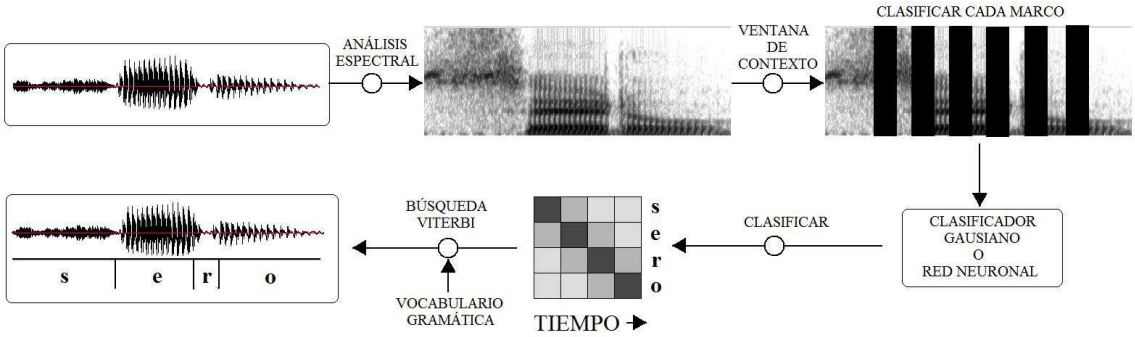


Figura 1.2. Segmentación y etiquetado utilizando reconocimiento de voz [27].

Conviene hablar ahora un poco acerca de las técnicas que se utilizaron en este trabajo.
La técnica de alineación forzada ha demostrado un rendimiento superior sobre las demás técnicas de segmentación y etiquetado automático. Aún así no es perfecta, ya que existen fronteras difusas muy difíciles de marcar. Otra manera de segmentar un corpus es a través de los eventos acústicos que se presentan en la señal de habla, por ejemplo por medio de técnicas wavelet, efectivas para localizar los cambios espectrales en la señal. La escala de Bark es una escala psicoacústica que toma en cuenta la no linealidad del oído humano a través del fenómeno de enmascaramiento. Como se mostrará en el capítulo 3, se ha utilizado la función Gaussiana modulada de acuerdo a la escala de Bark para realizar análisis wavelet. En este trabajo de tesis se realizó una combinación de estas técnicas para atacar el problema de la segmentación de un corpus.

1.2 Planteamiento del problema

Cuando la variación entre dos fonemas es continua, la frontera entre dichos fonemas es difusa y muy difícil de marcar debido al fenómeno de coarticulación. Los errores de concordancia de una segmentación automática son mayores debido a este tipo de fronteras difusas, aún utilizando alineación forzada.
Se propone realizar una combinación del algoritmo de alineación forzada con un algoritmo de segmentación acústica para probar si se mejora la segmentación fonética.

1.3 Objetivos

Objetivo general

Disponer de un sistema etiquetador semiautomático fonético para corpus de voces en español.

Objetivos específicos

- Utilizar las técnicas de segmentación y etiquetado semiautomático dentro de un corpus de voz en español.
- Realizar la extracción de parámetros wavelet utilizando Gaussianas moduladas en la escala de Bark.
- Emplear alineación forzada para la segmentación y etiquetado.
- Comparar el porcentaje de reconocimiento usando etiquetado automático contra el obtenido con etiquetado manual usando para ello HTK.

1.4 Justificación

Esta investigación es conveniente, porque ayudará a acelerar el desarrollo de aplicaciones y la investigación en el reconocimiento y síntesis de voz, además de que puede ser útil en las investigaciones de fonética acústica.
Las investigaciones y aplicaciones en reconocimiento de voz se beneficiarán de este trabajo, pues se tendrá una herramienta sobre la cual apoyarse para etiquetar corpus de voces de forma rápida y confiable.
Este trabajo es viable de ser realizado en un tiempo corto, debido a que los recursos necesarios son solamente el conocimiento de la señal de voz, una computadora y un entorno de programación. El sistema se puede implementar en cualquier lenguaje de programación disponible como C++, Java o Matlab.

1.5 Hipótesis

Se espera que la segmentación y etiquetado de un corpus de voz mejore utilizando una combinación de alineación forzada con segmentación acústica basada en wavelets Gaussianas moduladas en la escala de Bark, así como también se espera obtener una mayor efectividad en el reconocimiento de corpus de voces usando el etiquetado automático.

1.6 Solución propuesta

Como se mencionó en el planteamiento del problema, el algoritmo de alineación forzada es la solución más común para la segmentación y etiquetado automático de los corpus aunque suele fallar en fronteras difusas. Tomando en cuenta también la hipótesis planteada, se propone obtener las fronteras entre fonemas a partir del método de alineación forzada para luego refinar dichas fronteras con apoyo de un algoritmo basado en la transformada wavelet, el cual tome en cuenta aspectos de la percepción humana del sonido, como lo es la escala de Bark. En la figura 1.3 se muestra el diseño del sistema propuesto.
.

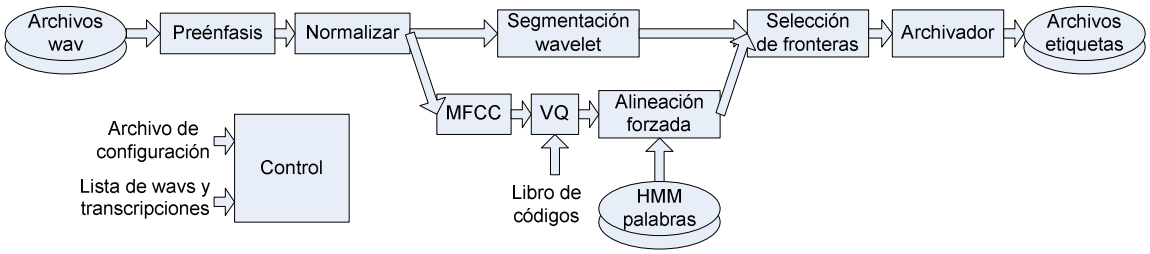


Figura 1.3. Diseño del sistema propuesto.

1.7 Alcances del trabajo

El tiempo y los recursos para el desarrollo del trabajo de tesis son limitados, por lo tanto es importante hacer explícitos los alcances del mismo.

- Partiremos de la suposición de que los archivos de audio entrantes al sistema tienen la pronunciación fonética adecuada para el español de México. También se supondrá que la pronunciación de cada archivo de audio concuerda con la transcripción de su archivo correspondiente.
- Este trabajo se enfoca en aplicar las técnicas de segmentación y etiquetado automáticos por medio de wavelets y alineación forzada, usando parámetros MFCC y modelos ocultos de Markov (HMM).
- El corpus utilizado contendrá todos los fonemas hablados en México, aunque no impondremos la condición de que el corpus sea balanceado. De acuerdo a [56] la elaboración de un corpus rico y balanceado involucra un enorme esfuerzo, ya que se deben seleccionar las frases pertinentes, verificar la distribución fonética de las mismas; y de ser necesario, eliminar algunas frases y agregar otras que mejoren la distribución. Cada remplazo afecta la distribución de otros fonemas, así que el proceso se repite tantas veces como sea necesario.
- Este trabajo no incluye el diseño ni desarrollo de un reconocedor de palabras. Se utiliza un reconocedor público, el sistema HTK, para obtener los resultados de reconocimiento.

1.8 Contribuciones

Se contribuye con una herramienta de segmentación fonética automática.
Se investiga si la combinación alineación forzada-segmentación acústica es mejor o no que la alineación forzada por sí misma.
Se investiga si el etiquetado automático mejora los resultados en los sistemas de reconocimiento automático de voz con respecto al uso del etiquetado manual.

1.9 Organización del trabajo

Capítulo 2. Estado del Arte.
Se presentan los avances en materia de segmentación y etiquetado de voz al momento de la elaboración de este trabajo de tesis.

Capítulo 3. Marco Teórico (Materiales y métodos).
Se exponen los temas que resultan básicos para la comprensión de este trabajo de tesis. Dichos temas se presentan en este orden: conceptos de segmentación y etiquetado, alfabetos fonéticos, diccionarios de

pronunciación, la escala de Bark, la transformada wavelet, el algoritmo de Galka-Ziolko, los modelos ocultos de Markov, y la alineación forzada.

Capítulo 4. Metodología y Desarrollo de la Investigación.
Este capítulo explica la justificación del alfabeto fonético utilizado, el pre-procesamiento digital (preénfasis, normalización), el algoritmo de segmentación acústica con transformada wavelet, y la implementación del algoritmo de alineación forzada.

Capítulo 5. Experimentos y Resultados.
El capítulo 5 explica el diseño de los experimentos, su realización, organiza e interpreta los resultados. El capítulo 5 le dedica una sección a cada conjunto de experimentos. La primera sección se dedica a la segmentación acústica; la segunda sección contiene los resultados de experimentos sobre un corpus con todos los fonemas, utilizando tanto alineación forzada como segmentación acústica; la tercera y última sección se dedica a los resultados de experimentos sobre un corpus de dígitos, tomando en cuenta tanto alineación forzada como segmentación acústica.

Conclusiones y Trabajo a Futuro.
Se comparará el trabajo realizado con los objetivos planteados al inicio. Se utilizan los resultados del capítulo 5 para elaborar conclusiones.

Anexo A.
Se presentan varios alfabetos fonéticos.

Anexo B.
Se describen los corpus utilizados en esta tesis, así como también sus diccionarios de pronunciaciones.

Anexo C.
Se describe de manera detallada cómo se utilizaron las herramientas de HTK para crear y evaluar sistemas de reconocimiento.

CAPÍTULO 2. ESTADO DEL ARTE

2.1 Introducción

Resolver el problema de segmentación y etiquetado es importante por las siguientes razones:
En la mayoría de los enfoques para reconocimiento del habla, las señales deben segmentarse antes de que se pueda tener un reconocimiento, ya que este etiquetado es el que posibilita el entrenamiento del sistema reconocedor para el conjunto de palabras a utilizar (diccionario). Se asume que las características de la señal en un segmento dado son constantes [62], aunque existen fonemas que no son acústicamente uniformes. La segmentación en palabras y fonemas es uno de los pasos fundamentales en los sistemas de reconocimiento automático de habla [15]. La segmentación del habla juega un rol importante en los sistemas de reconocimiento de habla porque reduce los requerimientos de memoria y minimiza la complejidad de cómputo de los sistemas de reconocimiento de habla continua [30].

La segmentación del habla tiene dos usos en particular: el primero es como una etapa crítica en los sistemas de reconocimiento de habla continua, el segundo es la segmentación de un corpus de voz. Si se está trabajando para el primer caso entonces es importante que la segmentación ocurra rápidamente, pero en el segundo caso basta con que la segmentación sea más rápida que la segmentación manual, la cual toma varios minutos por palabra (por ejemplo, el autor de este trabajo toma aproximadamente tres minutos para segmentar manualmente una palabra). En esta tesis se trabaja sobre la segunda opción, esto es, segmentación de un corpus.

Uno de los trabajos de investigación acerca de este tema más completos que se puede encontrar es el elaborado por John Paul Hosom en su tesis doctoral [27].

No existe una segmentación fonética que sea “correcta”. La naturaleza continua del habla impide la determinación exacta de las fronteras entre fonemas. Ni siquiera los fonetistas expertos pueden concordar en las fronteras de sus segmentos para una tolerancia arbitrariamente pequeña. Los investigadores de este tema han optado por evaluar la calidad de una segmentación comparando qué tan parecida es la segmentación automática a la segmentación manual generada por un experto. En este trabajo consideramos que no es suficiente comparar la segmentación automática con la manual, ya que no hay garantías de que la segmentación manual sea correcta, así que se realizaron mediciones indirectas con el uso de un reconocedor de habla, esto es, si el reconocimiento es alto entonces la segmentación puede considerarse que es de buena calidad. Una manera subjetiva de evaluar la calidad de una segmentación es utilizar el corpus segmentado para síntesis de voz, donde la voz sintetizada es juzgada por un humano; una segmentación de mejor calidad, generará habla, que suena más natural que una segmentación de más baja calidad.

2.2 Métodos automáticos de segmentación

El problema de la segmentación y etiquetado se ha afrontado utilizando diversas técnicas. Los primeros sistemas de segmentación y etiquetado trabajaban con una medición directa de la energía [46]. La energía sigue siendo una medición fundamental para resolver esta tarea.
Podemos clasificar las técnicas utilizadas en segmentación automática de la siguiente manera:

- Alineamiento temporal dinámico (del inglés Dynamic Time Warping o DTW, otras veces traducido al español como distorsión dinámica temporal). En adelante se hará referencia a esta técnica por sus siglas en inglés.

- Técnicas conexionistas.
- Técnicas estocásticas.
- Algoritmos evolutivos.
- Fractales.
- Técnicas difusas.
- Técnicas wavelet.
- Combinaciones de una o varias técnicas.

Destaca la técnica llamada alineación forzada (del inglés, forced alignment). La mayoría de los sistemas construidos para la segmentación y etiquetado automáticos están basados en alineación forzada, ya que ha demostrado un rendimiento superior. La alineación forzada puede llevarse a cabo mediante técnicas estocásticas o mediante una combinación de técnicas estocásticas y conexionistas. A continuación se explican los enfoques que se han seguido para resolver el problema.

2.2.1 Segmentación fonética manual

Tradicionalmente se ha considerado que la segmentación y etiquetados manuales son los de mejor calidad. Típicamente la segmentación manual la realiza un experto utilizando un software que despliega en pantalla la forma de onda de la voz, así como otras mediciones, por ejemplo el espectrograma de la señal. El software permite reproducir en audio los segmentos que el operador escoja. El operador elige los puntos que considera son las fronteras entre los fonemas. Ejemplos de este tipo de sistemas de operación manual son el HSLAB de HTK, y el SPEECHVIEWER del CSLU.

La herramienta SpeechViewer permite ver en una ventana una forma de onda de audio junto con su espectrograma. Esto resulta muy práctico para el segmentado y etiquetado manual. El espectrograma brinda una referencia visual sobre el contenido de frecuencias en diferentes intervalos de tiempo. Como se puede observar en la figura 2.1, SpeechViewer presenta una vista pequeña, que corresponde con la forma de onda completa; después, se presenta una vista más amplia. Para cada una de estas vistas existen sendos botones de reproducir. El botón de reproducción que está más a la izquierda corresponde a toda la señal, el segundo corresponde a la vista pequeña, el tercero a la vista ampliada.

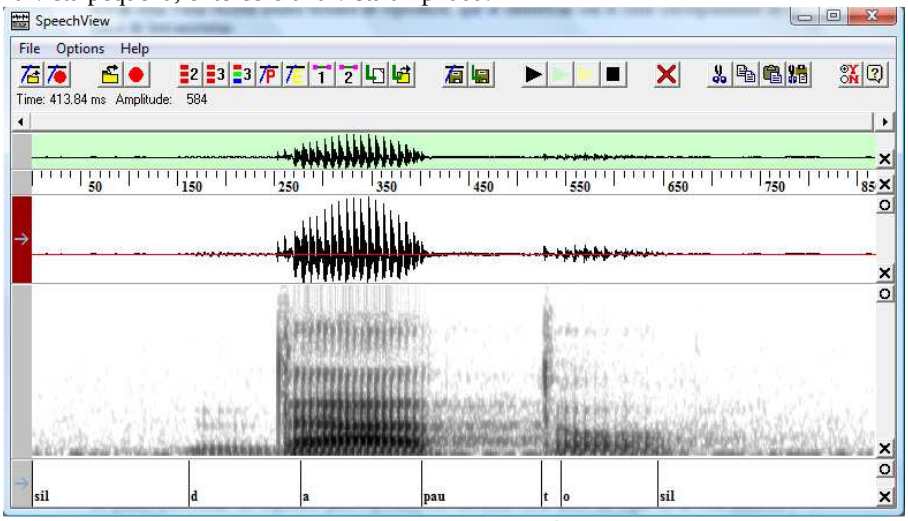


Figura 2.1. Herramienta SpeechViewer

Se puede seleccionar un segmento para reproducción, haciendo click sobre un lugar en la vista ampliada y arrastrando el mouse hacia otra parte de la forma de onda. Después se hace click en el botón de reproducción correspondiente. Esto reproducirá únicamente la sección seleccionada de audio.
En la parte inferior de la figura anterior se observan las etiquetas asociadas a cada segmento. Para etiquetar por primera vez un archivo, hay que hacer click en el botón "ADD WINDOW: New Label Window" que aparece en la barra de herramientas. El nombre del botón aparece pasando el apuntador por encima del botón. Luego, haciendo un click sencillo sobre el área de etiquetas hay que escribir la primera etiqueta y presionar la tecla ENTER. Cada vez que se presione dicha tecla se generará una nueva frontera entre segmentos. Dichos segmentos deben ubicarse en la posición correcta respecto al tiempo. De esta manera, en el segmento correspondiente a la /i/, debe escucharse solamente el sonido correspondiente a la /i/.

El archivo de etiquetas tiene extensión ".phn". En el archivo de etiquetas aparece la etiqueta asociada a cada segmento, así como el instante de inicio y fin (en milisegundos) para cada segmento, tal como se puede ver en la figura 2.2.

```
MillisecondsPerFrame: 1.0
END OF HEADER
0.000000 169.640621 s
169.640621 228.588253 j
228.588253 278.591945 e
278.591945 352.704525 pau
352.704525 381.278072 t
381.278072 590.222042 e
```

Figura 2.2. Archivo de etiquetas.

2.2.2 Segmentación por dynamic time warping (DTW)

En [50] se describe un sistema de etiquetado automático de habla. El sistema utiliza un reconocedor que genera una secuencia de fonemas tentativa. La secuencia generada se alinea con la secuencia correcta esperada usando programación dinámica. El proceso comienza con pocos datos etiquetados manualmente y va generando más etiquetas progresivamente. En dicho trabajo el método para evaluar la segmentación fue indirecto, utilizando un reconocedor, con el cual alcanzaron un 95% de reconocimiento de fonemas.

2.2.3 Segmentación por algoritmos evolutivos

En [37] se describe un sistema que utiliza el algoritmo llamado EASS (por el inglés *Evolutionary Algorithm for Speech Segmentation*). Se debe contar con el número correcto de segmentos. El material genético de cada individuo son las distancias de las fronteras tentativas respecto a las fronteras de una segmentación lineal inicial. Los individuos compiten por el método de torneo y se permite el elitismo. Una ventaja de este método es que no requiere entrenamiento previo.
El algoritmo no se probó a nivel de fonemas, sino a nivel de sílabas. Se reportó que el desempeño del algoritmo es similar al de un segmentador por modelos ocultos de Markov (HMM, por el inglés *Hidden Markov Model*). El HMM modeló el habla con cinco estados para los fonemas y los silencios, y utilizó tres estados para los silencios entre las palabras, con ventanas de Hamming de 25 ms y pasos de 10 ms. Una desventaja de este procedimiento es que a pesar de conocer previamente el número de fronteras correctas, el algoritmo puede arrojar errores de inserción y de borrado.
Lo criticable de su reporte es que no se muestran datos numéricos de los resultados.

2.2.4 Segmentación por técnicas de estudio de fractales

En [24] se propuso una medición nueva y útil para realizar segmentación de habla, pero la segmentación en sí no se llevó a cabo. La medición propuesta es la dimensión de la varianza de fractal. El algoritmo va calculando la dimensión de la varianza de fractal con una ventana que se va desplazando a través de la locución, lo cual genera una curva o trayectoria de dimensión. Grieder y Kinsner analizaron cualitativamente que la forma de la trayectoria de dimensión refleja las transiciones entre fonemas.
La desventaja del algoritmo es que acarrea los problemas asociados con la segmentación constante. Esto es, cuando el tamaño de la ventana es pequeño, los valores de la dimensión se vuelven erráticos; si por el contrario el tamaño de la ventana es grande, las características de distintos segmentos se fusionan. Así también, si el espaciamiento entre ventanas es pequeño habrá gran redundancia en los cálculos; si por el contario es muy grande habrá transiciones que no serán detectadas.
En dicho trabajo también es criticable el hecho de que no se presentaron resultados numéricos.

En [15] se describe un sistema que sí implementa la segmentación de habla basada en fractales. Al igual que el trabajo de Grieder y Kinsner, Fantinato et al. usaron la dimensión de fractal, pero esta vez calculada a partir de la transformada wavelet discreta (DWT, del inglés discrete wavelet transform). Se utilizaron las wavelets symlet, desde 12 hasta 24, pues experimentalmente fueron las que daban mejor resultado. El algoritmo contempla un preprocesamiento que obliga a que la dimensión obtenida esté numéricamente entre 1 y 2, esto porque no siempre se le puede encontrar similitud propia a la voz como si fuera un fractal. El algoritmo usa ventanas de 512 muestras con un traslape de 50%. Finalmente se analiza la trayectoria descrita por la dimensión. Si hay bordes en zonas de alta energía entonces eso se interpreta como una transición entre fonemas; bordes en zonas de baja energía se interpretan como transiciones entre palabras.
Es criticable que el trabajo no se comparó con otros sistemas ni tampoco se presentaron resultados numéricos.

2.2.5 Segmentación por técnicas difusas

En [9] el sistema aplica algoritmos difusos que asignan niveles de importancia a interpretaciones estructuradas de segmentos silábicos extraídos de la señal de una oración hablada. La fuente de conocimiento es una serie de reglas sintácticas cuyas categorías sintácticas son características fonéticas detectadas por una clasificación precategórica y categórica de los sonidos del habla. Este trabajo encaja en la categoría de sistemas de entendimiento del habla.
La evaluación se llevó a cabo mediante reconocimiento. Las consonantes sonoras se reconocieron con 91%, las no sonoras con 93%.

2.2.6 Segmentación por técnicas conexionistas

En [16] se presenta un sistema de segmentación automática que usa un perceptrón multicapa, con tres capas ocultas. El sistema utiliza un banco de filtros en la escala de Mel, con 19 canales. Los vectores de características se generan con ventanas de 50 ms, pero no menciona que haya usado traslape entre ellas. El perceptrón obtiene la probabilidad de que cada vector corresponda a cada fonema. Finster reportó que todas las fronteras se encontraron de manera correcta, lo cual es criticable porque no indica cuántos experimentos hizo, ni tampoco el intervalo de tolerancia para considerar que una frontera es “correcta”, así como tampoco describe las características del corpus usado.
Finster no reportó resultados numéricos con los cuales comparar su sistema.

En [49] se describe un sistema que combina árboles y redes neuronales en la estructura NTN (del inglés *Neural Tree Network*). Las palabras son modeladas con SWNTNs (del inglés *Subword Neural Tree Networks*), donde el número de SWNTNs corresponde al número de fonemas presentes en la palabra.
La evaluación de la segmentación fue directa, comparando la segmentación generada con la segmentación incluida en el corpus TIMIT. Los experimentos se efectuaron sobre cinco palabras, cada una con cuarenta repeticiones. Para una tolerancia de 25 ms obtuvieron 66.6% de fronteras correctas, y para una tolerancia de 20 ms obtuvieron 56.4%.

En [39] se presenta un sistema que implementa alineación forzada mediante una red neuronal. La precisión está dada en intervalos de 10 ms. Se utilizó un corpus numérico de veinte hablantes. La evaluación de la segmentación se hace de manera directa, comparando las fronteras generadas con las fronteras manuales. La mayoría de las fronteras se colocaron a una distancia dentro de una tolerancia de 23 ms respecto a las fronteras manuales. El mejor resultado obtenido fue un error de segmentación de 98.04 ms. En este trabajo se empleó una manera diferente de reportar los resultados. Se utilizó la ecuación 3.6.

2.2.7 Segmentación por técnicas estocásticas

Las técnicas estocásticas utilizan modelos probabilísticos para tratar con información incompleta o incierta. El enfoque estocástico más popular para el reconocimiento de voz son los modelos ocultos de Markov [57]. En [44] se presentan los algoritmos de los modelos ocultos de Markov, su derivación y su teoría subyacente. En el trabajo [3] se introduce el uso de los modelos ocultos de Markov al reconocimiento del habla; presentando también aplicaciones como localización de palabras (*word spotting*), segmentación, transcripción y rastreo del pitch.

En [2] se describe un sistema de segmentación y etiquetado basado en modelos ocultos de Markov. El sistema utilizaba un reconocedor fonético de patrones, al cual se le ingresa la señal sin filtrar, más cinco versiones filtradas de la señal utilizando filtros pasa banda. El reconocedor produce etiquetas la cual sólo se espera que sea una aproximación al fonema deseado. La secuencia de etiquetas producida es la que se utiliza como secuencia de observación para el modelo oculto de Markov.

En [27] se presenta una combinación de técnicas, pero la técnica base es HMM/ANN, la cual se escogió por su sólido marco matemático y su desempeño superior. La propuesta de Hosom fue integrar más información al modelo HMM/ANN, información que es usada por los humanos para reconocer el habla. Hosom propuso integrar tres tipos de características: características a nivel acústico, información para transiciones fonéticas y características fonéticas distintivas. Para las características a nivel acústico el sistema usa redes neuronales a las que ingresa características PLP junto con características de voicing, pitch, discriminación de intensidad, glotalización, e impulsos. Para las transiciones fonéticas Hosom modificó la definición de probabilidad de una secuencia de observaciones dada una secuencia de estados en un HMM, para que tome en cuenta para cada estado el estado anterior. Para las características fonéticas distintivas toma en cuenta manera, lugar y altura; utiliza una red neuronal con características en el dominio cepstral y características a nivel acústico, donde las salidas de la red neuronal son combinadas usando FLMP (del inglés *Fuzzy Logic Model of Perception*).
El sistema se probó sobre 14 corpora, y con diversas formas de medición. Respecto a la concordancia con la segmentación manual el mejor resultado fue de 92.57% de concordancia.

En [55] se describe un trabajo de segmentación y etiquetado automático fonéticos, utilizando HMM dependientes de contexto, logrando superar a los HMM independientes de contexto. Los resultados de Doroteo Torre Toledano son similares a los de Hosom, obteniendo un 92% de fronteras correctas.

En [54] se presenta una metodología que combina diversas técnicas pero se basa fundamentalmente en HMM. Se utilizan HMM de fonemas dependientes del contexto y se propone una forma de corregir errores sistemáticos. También se aplican técnicas de adaptación al hablante y finalmente se aplica un método para refinar las fronteras localmente, utilizando redes neuronales, lógica difusa y modelos de mixturas Gaussianas. La concordancia reportada dentro de una tolerancia de 20 ms es de 96.01%.

Nota. Como ya se ha mencionado, la alineación forzada es implementada mayormente con HMM, o con combinaciones de HMM con otras técnicas.

2.2.8 Segmentación por técnicas wavelet

En [59] se segmentó habla a nivel voz/no voz. El sistema usa la DWT para encontrar una curva del pitch. La wavelet que ofreció mejores resultados fue Haar. Sin embargo no se reportan en el artículo resultados numéricos.

En [1] se muestra un sistema basado en la DWT, en el cual se realiza una descomposición de seis niveles. Usa ventanas de Hamming de 256 muestras y un traslape de 25%. Para cada ventana obtiene coeficientes en la escala de Mel, y la energía en los seis niveles de descomposición wavelet. En el trabajo se define una función de distancia euclidiana entre cada par de ventanas adyacentes. Reportan que la exactitud fue de 91.7847%, y que los fonemas más difíciles de distinguir son aquellos que pertenecen a la misma clase. También reportan que el método fue mejor que usar solamente coeficientes en la escala de Mel.

En [31] el sistema usado utiliza wavelets Gaussianas moduladas en la escala de Bark. El analizador de Jáner descompone la señal en 17 bandas, pero en el artículo sólo se utilizaron las escalas 3 y 9. En dichas escalas se buscan los valores máximos de energía, mismos que deben sobrepasar un umbral. Otra variante propuesta es hacer lo mismo pero tomando el logaritmo de la señal previamente. La evaluación se realizó sobre el reconocimiento de los fonemas segmentados. Para el primer método obtuvieron un 92.05%, y con el método logarítmico obtuvieron un 84.5%. En ambos casos el reconocedor usó dos mixturas Gaussianas.

En [62] se usó la derivada de la energía en las subbandas de la DWT para obtener las fronteras tentativas. Se usa la wavelet de Meyer y la descomposición también es de seis niveles. Para cada subbanda se encuentra la envolvente de la energía, y después se obtiene la derivada de la misma utilizando un filtro. El algoritmo propone como fronteras aquellos puntos donde la energía y su derivada se aproximan lo suficiente. El algoritmo se probó sobre un corpus de 43 palabras. El error total fue de 4.3 (adimensional). En una palabra de ejemplo el error en el número de segmentos fue 0.125, y el error en las posiciones fue 2.29.

En [22] se usó también la DWT. La wavelet utilizada es la symlet12. Se realiza una descomposición de seis niveles, y para cada nivel el número de muestras se ajusta, de tal forma que todas las subbandas tengan el mismo número de muestras. Luego se obtiene la energía en cada banda, la cual se suaviza con un filtro. Las subbandas son ordenadas de mayor a menor según su contenido de energía, y esto se realiza para cada muestra de las subbandas, a dicho ordenamiento se le llama mapa de importancia. Los cambios significativos en el ordenamiento de las subbandas son llamados eventos. Los eventos que son suficientemente grandes en magnitud se proponen como fronteras entre fonemas. Se realizaron evaluaciones de manera directa e indirecta. El mejor resultado directo fue un error medio de 8 ms respecto a las fronteras manuales. El mejor resultado en cuanto a reconocimiento de fonemas fue de 50%.

2.3 Resumen

La segmentación automática puede evaluarse directamente, comparándola con una segmentación manual de referencia; o indirectamente utilizando un reconocedor y usando el porcentaje de reconocimiento generado.
Se han propuesto diversos enfoques para atacar el problema y en este trabajo se han clasificado en: DTW, técnicas conexionistas, estocásticas, evolutivas, difusas, fractales, wavelets. La mayoría de los sistemas analizados se apoyan en una combinación de técnicas.

CAPÍTULO 3. MARCO TEÓRICO (MATERIALES Y MÉTODOS)

3.1 Introducción

En este capítulo se exponen aquellos temas que resultan básicos para la comprensión de este trabajo de tesis. Se comienza con un repaso de los conceptos de segmentación y etiquetado. Después se pasa a la representación de los sonidos del habla, por lo cual se repasan conceptos de fonética, alfabetos fonéticos y diccionarios de pronunciación. Luego se revisa lo que es la escala de Bark, la cual intenta imitar la no linealidad del oído humano. Después nos adentramos al tema de la transformada wavelet, ya que es necesaria para el siguiente punto. Luego explicamos el algoritmo de Galka-Ziolko [22]. Por último damos un breve vistazo al algoritmo de alineación forzada y a los modelos ocultos de Markov.

3.2 El problema de segmentación-etiquetado

La segmentación es un proceso cuyo propósito es dividir una señal de habla continua en unidades discretas, basándose en algún tipo de medición de similitud acústica. Los esquemas de etiquetado asocian un símbolo fonético con cada unidad segmental, esto es, que a cada segmento se le asigna una etiqueta.

No existe un algoritmo sencillo para generar fronteras fonéticas. Sin embargo, si uno está satisfecho con fronteras acústicas (es decir, fronteras asociadas con cambios significativos en las características acústicas del habla), entonces es posible obtener algoritmos automáticos para segmentar el habla [46].

Para cada base de datos, el etiquetado y segmentación debe hacerse una vez. Este proceso realizado manualmente por un grupo de personas entrenadas es la opción principal. Aunque sea muy laborioso, este método provee un etiquetado de alta calidad que puede servir para evaluar los procedimientos automáticos [33].

La naturaleza continua del habla hace que no exista una verdadera segmentación. La segmentación fonética no es lo mismo que la segmentación acústica, ya que varios segmentos acústicos pueden representar un segmento fonético. Esto es, algunos fonemas no son acústicamente uniformes. Los métodos de análisis espectral permiten segmentación precisa desde el punto de vista acústico, pero para realizar segmentación fonética precisa se necesita información adicional, como la transcripción fonética [22].

En los enfoques clásicos, las señales de habla se segmentan de manera constante, por ejemplo, con bloques de 25 milisegundos de duración. En cada intervalo de tiempo se toma un bloque y se obtiene un vector de parámetros [32]. Sin embargo, este enfoque corre el peligro de perder información acerca de ciertos fonemas, ya que algunos sonidos pueden mezclarse en un bloque, o algunos fonemas pueden perderse [62], un ejemplo de esto ocurre con las plosivas [5]. Es por esta razón que surge la segmentación no constante. Además, la segmentación no constante es benéfica para los sistemas de reconocimiento de habla automáticos (ASR, del inglés *Automatic Speech Recognition*), ya que decrecen el espacio de búsqueda Viterbi, reduciendo el costo computacional [22].

3.2.1 Evaluación de la segmentación

La evaluación del desempeño de la segmentación se basa principalmente en tres conceptos:

- Exactitud en la colocación de fronteras.
- Número de inserciones.
- Número de borrados.

A continuación se mencionarán algunas de las medidas básicas que han sido propuestas para evaluar los resultados de la segmentación.

- Número de segmentos detectados automáticamente [62]. Se denota por n_a.
- Número de segmentos realizados manualmente [62]. Se denota por n_h o n_m.
- Número de inserciones [12]. Es el número de segmentos que se insertaron de más en la segmentación automática.
- Número de borrados [12]. Es el número de segmentos que faltaron en la segmentación automática.
- Tasa de inserción relativa. Se calcula con la ecuación 3.1 [22].

$$\lambda = \frac{\text{número de inserciones}}{n_a} \tag{3.1}$$

- Tasa de borrado relativa [22].

$$\mu = \frac{\text{número de borrados}}{n_h} \tag{3.2}$$

- Error en el número de segmentos [62].

$$\epsilon_{n\,(w)} = \frac{|n_a - n_h|}{n_h} \tag{3.3}$$

  Donde w indica que este valor depende de la palabra en cuestión.

- Error en la exactitud en la posición de los segmentos [62].

$$\epsilon_{p(w)} = \sum_j \min_i |p_j - q_i| \tag{3.4}$$

  Donde p_i es la posición de la i-ésima frontera en la segmentación automática, y q_j es es la posición de la j-ésima frontera en la segmentación manual.

- Error global. [62].

$$\epsilon(w) = \frac{1}{n_w} \sum_w 5\epsilon_n(w) + \epsilon_p \tag{3.5}$$

- Error de segmentación cuadrático medio. Este error se calcula sumando el cuadrado de las diferencias entre las fronteras de inicio de cada fonema, dividiendo esta suma entre el número total de etiquetas y obteniendo la raíz cuadrada del cociente [39].

$$\text{Error de segmentación} = \sqrt{\frac{\sum(\text{manual} - \text{automático})^2}{N}} \tag{3.6}$$

- Porcentaje de fronteras localizadas dentro de una distancia dada. Se denota por PB. Se suelen utilizar intervalos de tolerancia de entre 10 ms y 30 ms [23].
- Porcentaje de segmentos que concuerdan en ambas segmentaciones. Se denota por PF. [23].
- Porcentaje de fronteras que se encuentran separadas en más de 10 milisegundos [33].

En el caso de la segmentación fonética, la medida más utilizada es un porcentaje de concordancia respecto a las etiquetas manuales dentro de una tolerancia. Generalmente dicha tolerancia es de 20 ms; es decir, el porcentaje de fronteras automáticas que se encuentran dentro de 20 ms respecto a sus correspondientes fronteras manuales.

3.3 Fonética y fonología

Puesto que en este trabajo de tesis se encuentra relacionado con los sonidos del habla del español, está necesariamente relacionado con la fonética y la fonología, es por ello que se mencionan sus definiciones [5].

- Fonética. Es la disciplina que estudia los sonidos desde el punto de vista de su producción, transmisión y percepción, sin preocuparse del significado de los mismos.
- Fonología. Es la disciplina que estudia los sonidos dentro de una lengua, establece las normas para su ordenamiento.
- Fonema. Es la unidad fonológica más pequeña. Su número es reducido. No tiene significado por sí mismo, pero el significado de una palabra cambia si se intercambian dos fonemas, dando lugar al fenómeno de la oposición.

De acuerdo a [42], existen diversos tipos de fonética:

- La fonética auditiva se interesa en la percepción del sonido.
- La fonética articulatoria o genética fue la única utilizada durante mucho tiempo y es la que aún se utiliza en las descripciones de las lenguas. La fonética articulatoria se enfocaba en el análisis radiográfico y palatográfico de las articulaciones intencionadas, con filmes radiológicos donde los procesos de asimilación, anticipación articulatoria, etc. son bien patentes.
- La fonética acústica se ocupa de estudiar los componentes que conforman la onda sonora compleja de los sonidos articulados y de buscar cuál o cuáles de ellos son imprescindibles para su reconocimiento. Los datos proporcionados por la fonética acústica son más objetivos, adecuados y constantes que los de la fonética articulatoria para la descripción fónica y la comunicación humana.

3.3.1 Alfabetos fonéticos

Una transcripción fonética es traducir ya sea una forma de onda o una transcripción ortográfica en una cadena de símbolos que representan fonemas. Una transcripción fonética es cuando se incluyen marcas diacríticas para representar alófonos [10]. Para realizar estas transcripciones es que se necesitan alfabetos fonéticos.
En [8] se indica que los alfabetos fonéticos son alfabetos especializados para transcribir los sonidos del habla. Son conjuntos de caracteres que representan símbolos fonéticos.

En España el sistema fonológico del castellano está formado por 24 fonemas; en cambio, en México el sistema se ha reducido a 22 [40].

Un alfabeto fonético es definido por la Real Academia Española como una ortografía o un sistema de transcripción que trata de representar los sonidos con mayor exactitud que la ortografía convencional.

En este trabajo de tesis se utilizará el alfabeto fonético Worldbet, ya que ha demostrado ser útil para transcribir el español. Para ver una descripción más completa de los alfabetos fonéticos refiérase al anexo A.

3.3.2 Diccionarios de pronunciación

Un diccionario de pronunciaciones es un conjunto de transcripciones fonéticas. El diccionario muestra todas las pronunciaciones consideradas para cada palabra con la que trabajará el sistema ASR. Esto es, para cada palabra considerada, el diccionario contiene una o más secuencias de fonemas, de una manera muy similar a lo que uno encontraría en un diccionario.

3.4 Escala de Bark

Ciertos problemas de la ciencia requieren que el rango de frecuencias audibles se subdivida de una forma relacionada a la manera en que el oído parece realizar el proceso. En este tipo de problemas, las bandas críticas resultan muy útiles. Estas bandas han sido medidas en experimentos relativos al umbral para sonidos complejos, experimentos de enmascaramiento, y percepción de la fase. Las mediciones indican que las bandas críticas tienen un cierto ancho, pero su posición cambia continuamente. Parece ser que la subdivisión en bandas críticas corresponde de manera muy cercana a la mecánica de la cóclea, a la discriminación de frecuencias y a la escala Mel del pitch. La tabla 3.1 muestra los valores para frecuencias preferidas que definen los límites auditorios de las bandas críticas [63].

La escala de Bark se basa en el fenómeno del enmascaramiento. Existen dos tipos de enmascaramiento: en frecuencia y en tiempo. El enmascaramiento en frecuencia ocurre cuando dos sonidos tienen frecuencia similar, el sonido de menor frecuencia enmascara al de mayor frecuencia. En el enmascaramiento temporal, si se comienza con un solo tono, y después se escucha otro tono, debe pasar cierta cantidad de tiempo para poderlo escuchar. La escala de Bark está basada en bandas críticas. Si hay dos sonidos en una misma banda crítica, el que tenga más energía enmascarará al otro. La posición de una banda crítica no es fija, puede cambiar continuamente.

La escala de Bark relaciona frecuencias acústicas con frecuencias perceptuales. Para calcular los Barks b dada la frecuencia f se utiliza la ecuación 3.7.

$$\mathrm{b} = 13\tan^{-1}\left(0.76\frac{\mathrm{f}}{\mathrm{kHz}}\right) + 3.5\tan^{-1}\left(\frac{\mathrm{f}}{7.5\mathrm{kHz}}\right)^{2} \tag{3.7}$$

Cada banda crítica de la escala de Bark tiene asociada una frecuencia central y un ancho de banda, que se muestran en la tabla 3.1.

Tabla 3.1. Frecuencia central y ancho de banda para cada banda crítica de acuerdo con la escala de Bark [32].

Banda Crítica	Frecuencia de Corte	Ancho de Banda	Frecuencia Central
0	0	100	50
1	100	100	150
2	200	100	250
3	300	100	350
4	400	110	450
5	510	120	570
6	630	140	700
7	770	150	840
8	920	160	1000
9	1080	190	1170
10	1270	210	1370
11	1480	240	1600
12	1720	280	1850
13	2000	320	2150
14	2320	380	2500
15	2700	450	2900
16	3150	550	3400
17	3700	700	4000
18	4400	900	4800
19	5300	1100	5800
20	6400	1300	7000
21	7700	1800	8500
22	9500	2500	10500
23	12000	3500	13500
24	15500		

3.5 Transformada wavelet

La transformada wavelet es interesante para analizar señales no estacionarias. A diferencia de la STFT (Short Time Fourier Transform, STFT en inglés), la transformada wavelet utiliza ventanas cortas para analizar altas frecuencias y ventanas largas para analizar bajas frecuencias; esto es, realiza un análisis con un factor Q constante [48].

Las funciones wavelet son funciones base y se obtienen dilatando y trasladando un prototipo de una wavelet. Cada función base genera un filtro pasa banda de factor Q constante.

Ahora procedermos a hablar acerca del soporte de una función.

3.5.1 Soporte de una función

Se dice que una función $f \in L^2(\mathbb{R})$ tiene sopote en el interalo $[a,b] \subset \mathbb{R}$ si $f(t) = 0$ casi en cualquier parte fuera de $[a,b]$, esto es, si el conjunto de los puntos fuera de $[a,b]$ en los cuales $f(t) \neq 0$ tiene medida cero. Entonces escribimos $supp\ f\ \subset [a,b]$. El conjunto de funciones cuadráticamente integrables con soporte en $[a,b]$ se denota por $L^2([a,b])$. Dada una función arbitraria $f \in L^2(\mathbb{R})$ decimos que f tiene soporte compacto si $supp\ f\ \subset [a,b]$ para algún intervalo acotado $[a,b] \subset \mathbb{R}$ [34].

3.5.2 El principio de incertidumbre de Heisenberg

Tanto la transformada wavelet como la STFT deben atenerse al principio de incertidumbre de Heisenberg, el cual indica que sólo podemos sacrificar resolución en tiempo para obtener resolución en frecuencia o viceversa:

$$\Delta t \Delta f \geq \frac{1}{4\pi} \tag{3.8}$$

Esto significa que dos tonos sólo podrán distinguirse entre sí si están separados en frecuencia por más que Δf. Del mismo modo, dos pulsos en el tiempo sólo podrán distinguirse si están separados más que Δt. Un caso especial es la ventana Gaussiana, en el que la desigualdad 3.8 se vuelve igualdad.

3.5.3 Transformada wavelet continua

En el caso de la STFT, una vez que se escoge una ventana, se usará su resolución de manera constante en todo el plano tiempo-frecuencia. Esta es una limitación que supera la transformada wavelet. Aunque la transformada wavelet también se atiene a la desigualdad de Heisenberg, la resolución en tiempo se puede hacer arbitrariamente pequeña para análisis en altas frecuencias. De manera similar, la resolución en frecuencia se puede hacer arbitrariamente pequeña para el análisis en bajas frecuencias; pero no ambas al mismo tiempo.

La transformada wavelet se puede ver como un banco de filtros pasa banda con un ancho de banda relativo constante, esto es, con factor Q constante. La respuesta al impulso de cada uno de estos filtros es una versión escalada de un prototipo madre, esto es:

$$\psi_a = \frac{1}{\sqrt{a}} \psi\left(\frac{t}{a}\right) \tag{3.9}$$

donde *a* es el factor de escala, y $\frac{1}{\sqrt{a}}$ es para normalizar. Esto resulta en la definición de la transformada wavelet continua (en adelante llamada CWT):

$$CWT_x(\tau, a) = \langle x, \psi_{a,\tau} \rangle = \int x(t) \psi_{a,\tau}{}^*(t) dt = \frac{1}{\sqrt{a}} \int x(t)\, \psi^*\left(\frac{t-\tau}{a}\right) dt \tag{3.10}$$

La CWT es un producto interno que mide la similitud entre la señal $x(t)$ y las funciones base $\psi_{a,\tau}$ (wavelets). Cada coeficiente wavelet indica qué tan cercana es la señal a una función base en particular.

Ahora, conviene hablar un poco acerca del significado de escala y cómo está relacionado con la frecuencia.

3.5.4 Noción de escala

Cuando una función $f(t)$ es escalada: $f(t) \rightarrow f(at)$ donde $a > 0$, entonces la función se contrae si $a > 1$ y se expande si $a < 1$.

La interpretación de la ecuación de la CWT es que, conforme la escala se incrementa, la respuesta impulsiva $\psi\left(\frac{t-\tau}{a}\right)$ se extiende en el tiempo y sólo toma en cuenta el comportamiento a tiempo largo. Escalas muy grandes significan vistas globales, mientras que escalas pequeñas significan vistas detalladas.

En la transformada wavelet la escala es inversa a la frecuencia, y genera una representación bidimensional tiempo-escala.

La transformada wavelet es la opción cuando no se desea utilizar la STFT. Gabor adaptó la transformada de Fourier para mapear la señal a una representación bidimensional de tiempo-frecuencia (τ, f).

En seguda, conviene hablar acerca de las propiedades de las funciones wavelet.

3.5.5 Propiedades de las funciones wavelet

Las funciones $\psi(t)$ cuadráticamente integrables que satisfacen la condición de la ecuación 3.11 se pueden utilizar para analizar y reconstruir una señal sin pérdida de información:

$$\int_0^{+\infty} \frac{|\psi(\omega)|}{\omega} d\omega < +\infty \tag{3.11}$$

Esta condición se llama condición de admisibilidad e implica que la transformada de Fourier de $\psi(t)$ se desvanece en la frecuencia cero:

$$|\psi(\omega)|^2 = 0 \quad para\ \omega = 0 \tag{3.12}$$

Lo anterior significa que las wavelets deben tener un espectro de tipo pasa banda. Un cero en la frecuencia cero también significa que el promedio de la wavelet en el dominio del tiempo debe ser cero.

$$\int \psi(t) dt = 0 \tag{3.13}$$

3.5.6 Wavelets Gaussianas moduladas

En su tesis doctoral, Léonard Janer García [32] propuso crear familias de funciones wavelets que pudieran modularse en la escala de Bark, correspondiendo con las bandas críticas en las que el ser humano descompone la respuesta a tonos puros enmascarados. La función que escogió como wavelet fue la Gaussiana modulada, ya que tiene el mejor compromiso de resolución en tiempo y en frecuencia.

Janer no hizo síntesis a partir de las wavelets, y por lo tanto no impuso ortogonalidad entre ellas. Lo que él buscaba era poder identificar características de la señal en cada momento, y obtener un conjunto de parámetros para representar la señal de voz que no superaran la veintena de coeficientes.

Hay dos ventajas de haber escogido la Gaussiana modulada: por un lado, puede modularse y de esta forma, es posible centrar la Gaussiana modulada en la frecuencia central de una banda crítica de la escala de Bark. Por

otro lado, es posible manipular la desviación estándar de la Gaussiana, para de ese modo ajustar su ancho de banda al ancho de la banda crítica en la escala de Bark.

3.5.7 Obtención de los parámetros de escala y modulación

La función wavelet madre con la que Janer comenzó es:

$$\psi(t) = e^{-j\omega_0 t} e^{-\frac{1}{2}t^2} \tag{3.14}$$

donde su transformada de Fourier es:

$$\widehat{\psi}(\omega) = e^{-\frac{1}{2}(\omega+\omega_0)^2} \tag{3.15}$$

Para ajustar la Gaussiana modulada a las bandas de referencia:

$$\psi_a(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t}{a}\right) = \frac{1}{\sqrt{a}} e^{-j\omega_0 \frac{t}{a}} e^{-\frac{1}{2}\left(\frac{t}{a}\right)^2} \tag{3.16}$$

$$\widehat{\psi}_a(\omega) = \sqrt{a}\widehat{\psi}(a\omega) = \sqrt{a} e^{-\frac{1}{2}(a\omega+\omega_0)^2} \tag{3.17}$$

Para obtener la frecuencia central de la Gaussiana modulada, hay que observar la ecuación 3.17 y hacer que el exponente sea cero. De esa forma obtiene que la frecuencia central está dada por:

$$\omega_c = -\frac{\omega_0}{a} \tag{3.18}$$

Luego, para hacer que la Gaussiana modulada tenga el ancho de banda que tiene la banda crítica, se buscan las frecuencias de corte (recordar que en la frecuencia de corte, la magnitud de la transformada de Fourier es $1/\sqrt{2}$ veces el valor de la magnitud máxima).

$$\sqrt{a} e^{-\frac{1}{2}(a\omega_{-3}+\omega_0)} = \sqrt{\frac{a}{2}} \tag{3.19}$$

Al despejar 3.16:

$$a\omega_{-3} + \omega_0 = \sqrt{2\ln(\sqrt{2})} \tag{3.20}$$

Al combinar 3.15 y 3.17, se obtiene:

$$a = \frac{\sqrt{2\ln(\sqrt{2})}}{\omega_{-3} - \omega_c} \tag{3.21}$$

A partir de la ecuación anterior Janer obtuvo un conjunto de parámetros de escala y un conjunto de parámetros de frecuencia de modulación para ajustar las wavelets Gaussianas a la escala de Bark; para ello tomó los valroes de la tabla de la escala de Bark (tabla 3.1); como resultado generó la tabla 3.2.

Tabla 3.2. Parámetros de escala y de frecuencia de modulación (Tomado de [32]).

Índice i	Escala a_i	Modulación ω_i (rad/s)
0	0.0027	0.8482
1	0.0027	2.5447
2	0.0027	4.2412
3	0.0027	5.9376
4	0.0027	7.6341
5	0.0029	10.25
6	0.00233	10.25
7	0.001942	10.25
8	0.001631	10.25
9	0.001394	10.25
10	0.001191	10.25
11	0.001019	10.25
12	0.000882	10.25
13	0.000759	10.25
14	0.000653	10.25
15	0.000563	10.25
16	0.000479	10.25

En la figura 3.1 se muestra el esquema de análisis.

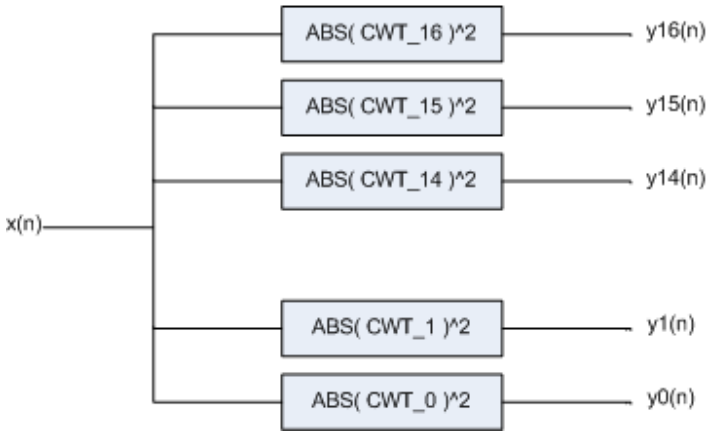


Figura 3.1. Esquema de análisis con wavelets Gaussianas moduladas en la escala de Bark.

3.6 Algoritmo de segmentación wavelet de Galka-Ziolko

Jakub Galka y Marius Ziolko desarrollaron un método de segmentación basado en wavelets [22].

Los métodos clásicos de segmentación se basan en rastrear los cambios temporales de las características de la señal. Galka y Ziolko propusieron un método basado en la transformada wavelet discreta (en adelante llamada DWT), utilizando la wavelet symlet12.
El método utiliza características acústicas, no fonéticas y por ello los resultados reportan una alta tasa de inserción, ya que algunos fonemas no son uniformes acústicamente.

El algoritmo se basa en detectar transiciones de energía entre distintas sub bandas wavelet. Una transición lo suficientemente significativa recibe el nombre de evento.

Una ventaja de la segmentación no uniforme es que le ayuda a los sistemas de reconocimiento de habla (ASR, por sus siglas en inglés) que se basan en modelos ocultos de Markov (HMM, por sus siglas en inglés), ya que la segmentación no uniforme reduce el número de segmentos que los elementos de más alto nivel deben procesar. Esto implica que el espacio de búsqueda del algoritmo de Viterbi decrece y con esto hay una mejora computacional.
Por ejemplo, diez segundos de voz resultan típicamente en 750 marcos traslapados de 20 milisegundos de duración. Utilizando segmentación no uniforme, este número se puede reducir a 100 marcos no uniformes.

3.6.1 Pasos del algoritmo de Galka-Ziolko

A continuación se mencionan los pasos que fueron propuestos en [22]:

1. Realizar una descomposición diádica de seis niveles utilizando la DWT y la wavelet symlet12. Cada nivel contiene aproximadamente la mitad de las muestras que el nivel anterior.

$$D = \left\{ \{d_{6,n}{}^2\}_n, \{d_{5,n}{}^2\}_n, \dots, \{d_{1,n}{}^2\}_n \right\} \tag{3.22}$$

2. Unificar la longitud de la discretización de tiempo. Puesto que cada nivel contiene menos muestras que el nivel anterior, cada nivel tiene sus muestras más separadas en tiempo que el nivel anterior. Esto es así porque se utiliza una transformada decimada. La unificación de la longitud de la discretización de tiempo se refiere a sumar el número adecuado de elementos de cada banda, de tal forma que al final, cada banda tenga igual número de elementos. De esta forma se obtiene :

$$B_{m,k} = \sum_{n=(k-1)2^{7-m}+1}^{k2^{7-m}} d_{m,n}{}^2 \tag{3.23}$$

3. Suavizar el espectro con un filtro FIR pasabajas de media móvil de 5 elementos.
4. Crear el mapa de importancia. Sea m_1 el primer nivel de descomposición, m_2 el segundo, etc. Para cada k-ésima muestra en el tiempo, se genera un vector, cuyas componentes son los niveles de descomposición wavelet ordenados según su contenido de energía en ese instante k. La colección de vectores columna que se generan con este procedimiento forma una matriz llamada el mapa de importancia. A continuación se muestra el k-ésimo vector columna en el mapa de importancia.

$$M_k = [m_1, m_2, m_3, m_4, m_5, m_6]^T \tag{3.24}$$

Ya que los elementos de cada vector columna de la matriz M_k están ordenados según su contenido de energía, tendremos que:

$$B_{m1,k} \geq B_{m2,k} \geq B_{m3,k} \geq B_{m4,k} \geq B_{m5,k} \geq B_{m6,k} \quad (3.25)$$

Los valores que puede tomar cada m corresponden al conjunto m $\in\{1,2,3,4,5,6\}$, ya que se hizo una descomposición de 6 niveles. En cada k-ésimo instante, la primera fila de la matriz M_k contiene las bandas que tuvieron la máxima energía, mientras que la última fila contiene las bandas que tuvieron la mínima energía.

5. Aplicar la función de detección de eventos. Un evento es un cambio de ordenamiento en las bandas, y es más fuerte si dicho cambio ocurre en las bandas con mayor energía. A continuación se muestra la función de detección de eventos:

$$f(k) = \sum_{m=1}^{6} \frac{\left|M_{m,k+1} - M_{m,k}\right|}{m} \quad (3.26)$$

6. Establecer los umbrales. Es posible que la función de eventos reporte más de una vez el mismo evento, para evitar este fenómeno se tendrán que elegir los picos más altos de f(k) que se encuentren dentro de un vecindario de t_min milisegundos y que sean de magnitud mayor que el umbral f_{tr}.
Existen dos formas de establecer el umbral. La primera forma es usar un umbral constante. La segunda forma es hacer un umbral adaptativo en un vecindario de 2N muestras:

$$f_{tr}(k) = \frac{\alpha \sum_{n=-N}^{N} f(k-n)}{2N} \quad (3.27)$$

Si un pico de f(k) es más alto que $f_{tr}(k)$ y además es el más alto de todos en el vecindario de 2N muestras, entonces cuenta como un evento. El vecindario de 2N muestras está relacionado con la duración mínima de un fonema, y en el trabajo de Galka y Ziolko se reporta que N corresponde a 100 milisegundos.

3.7 Métrica y distorsión

Como se menciona en [10], el espacio Cartesiano N-dimensional $\mathbb{R}^N$ es la colección de todos los vectores N-dimensionales de elementos reales. Una métrica $d(.,.)$ en $\mathbb{R}^N$ es una función de valores reales con tres propiedades: para todo $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^N$,

$$d(\boldsymbol{x}, \boldsymbol{y}) \geq 0 \quad (3.28.a)$$
$$d(\boldsymbol{x}, \boldsymbol{y}) = 0 \text{ si y solo si } \boldsymbol{x} = \boldsymbol{y} \quad (3.28.b)$$
$$d(\boldsymbol{x}, \boldsymbol{y}) \leq d(\boldsymbol{x}, \boldsymbol{z}) + d(\boldsymbol{z}, \boldsymbol{y}) \quad (3.28.c)$$

Cualquier función que cumpla estas propiedades es una métrica legítima en el espacio vectorial. Una medida de similitud que no se adhiere necesariamente a las propiedades formales de una distancia métrica se llama medida de distorsión.

En [20] se menciona, además que, para su uso efectivo en el reconocimiento de voz, una medida de distancia $d(\boldsymbol{x}, \boldsymbol{y})$ entre dos vectores $\boldsymbol{x}$ e $\boldsymbol{y}$ debe satisfacer preferentemente las ecuaciones 3.29:

$$d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x}) \quad (3.29.a)$$
$$d(\boldsymbol{x}, \boldsymbol{y}) > 0\,, \quad \boldsymbol{x} \neq \boldsymbol{y} \quad (3.29.b)$$
$$d(\boldsymbol{x}, \boldsymbol{y}) = 0, \quad \boldsymbol{x} = \boldsymbol{y} \quad (3.29.c)$$

Si $d(\boldsymbol{x}, \boldsymbol{y})$ es una distancia en el sentido matemático, debe satisfacer la desigualdad del triángulo.

3.8 Alineación forzada

Existen diversas soluciones al problema de la alineación fonética. La solución más común está basada en modelos ocultos de Markov (HMM, por sus siglas en inglés) o en la solución híbrida de modelos ocultos de Markov con redes neuronales artificiales (HMM/ANN, por sus siglas en inglés). Le siguen los sistemas basados en DTW. Por último quedan otros tipos de sistemas mucho menos utilizados en este ámbito (ver capítulo 2).

La alineación forzada se trata del uso de reconocedores de habla basados en HMM o HMM/ANN para obtener alineaciones fonéticas. Es por lo tanto, el método más común para alinear el habla. Realiza un reconocimiento de la señal de habla manteniendo una búsqueda restringida a la secuencia conocida de fonemas.

Puesto que la alineación se obtiene forzando el resultado del reconocimiento a que sea la secuencia fonética propuesta entonces el proceso recibe el nombre de alineación forzada. La secuencia fonética se determina a priori con un diccionario de pronunciaciones, con reglas de conversión de grafema a fonema, o por un humano. Dado que la alineación forzada está íntimamente relacionada con los sistemas HMM y con la búsqueda Viterbi, se presenta a continuación una explicación de los mismos.

3.8.1 Modelos ocultos de Markov

Los modelos ocultos de Markov reciben su nombre por el matemático ruso Andrei Andreevich Markov. Los modelos ocultos de Markov (HMM) son modelos estocásticos de señales; es decir, caracterizan las propiedades estadísticas de las señales.

Existen muchas situaciones en el reconocimiento automático del habla donde se necesitan tomar decisiones basadas en información incompleta o incierta. El modelado estocástico es un método general y flexible para manejar dichas situaciones. El modelado estocástico consiste en utilizar un modelo probabilístico para la incertidumbre o incompletitud de la información. La incertidumbre en el reconocimiento automático del habla surge por muchas razones; por ejemplo, la señal acústica es ambigua porque los eventos acústicos para los sonidos de habla individuales se reducen o se borran en el habla continua normal. Un modelo abstracto para estas situaciones de incertidumbre es que existen dos secuencias de variables aleatorias: Y(1), Y(2), ..Y(T) y X(1), X(2)…X(T). La secuencia de X's representa alguna secuencia que deseamos conocer, pero que no es directamente observable. La secuencia de Y's en cambio, sí es observable directamente y que está relacionada con la secuencia de X's. El modelado estocástico consiste en formular un modelo probabilístico para generar las secuencias de X's y para generar una secuencia de Y's basada en la secuencia de X's. Estos modelos se pueden utilizar para hacer inferencias en un sistema de reconocimiento de habla. Una clase específica de modelos estocásticos son los modelos basados en la teoría de un proceso de Markov [3].

La información de esta sección está basada en [44]. Los modelos de Markov son autómatas probabilísticos, y por lo tanto tienen un conjunto de estados. Los modelos observables de Markov son sencillos porque en ellos la salida en cada instante es el conjunto de estados, y cada estado tiene un significado físico. En cambio, en los modelos ocultos de Markov, la salida del sistema u observación es una función probabilística que depende del estado actual; en estos procesos existe un proceso que no es observable (de ahí la palabra "oculto").

Los elementos de un modelo oculto de Markov son:

- N, el número de estados en el modelo. Los estados individuales se identifican por $S = \{S_1, S_2, \dots, S_N\}$.
- M, el número de diferentes observaciones posibles en cada estado. Es el tamaño del alfabeto discreto. Estos símbolos son las salidas observables del sistema que se está modelando. En el caso del procesamiento de voz, estos símbolos representan vectores de características, que se denotan por $V = \{v_1, v_2, \dots, v_M\}$.
- A, la distribución de probabilidades de transición entre los distintos estados del modelo. Se representa con una matriz, $A = \{a_{ij}\}$, donde cada componente i,j representa la probabilidad de saltar hacia el estado j, dado que el modelo se encontraba en el estado i en el instante anterior (probabilidad condicional); es decir:

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i] \quad 1 \leq i, \leq N \tag{3.30}$$

- B, la distribución de probabilidad de los símbolos de observación en cada j-ésimo estado, $B = \{b_j(k)\}$:

$$b_j(k) = P[v_k en\ t | q_t = S_j] \quad 1 \leq j \leq N \quad 1 \leq k \leq M \tag{3.31}$$

- π, la distribución de probabilidad de estado inicial:

.

$$\pi_i = P[q_1 = S_i] \quad 1 \leq i \leq N \tag{3.32}$$

La notación resumida para representar un HMM es:

$$\lambda = (A, B, \pi) \tag{3.33}$$

Los tres problemas básicos de los HMMs son:

- Problema 1: dada una secuencia de observaciones $O = O_1, O_2, \dots, O_T$, y un modelo $\lambda = (A, B, \pi)$, ¿Cómo calcular $P(O|\lambda)$, es decir, la probabilidad de la secuencia de observaciones dado el modelo?
- Problema 2: dada la secuencia de observaciones $O = O_1, O_2, \dots, O_T$ y el modelo λ, ¿Cómo escoger una secuencia de estados $Q = q_1, q_2, \dots, q_T$ que sea óptima en algún sentido?
- Problema 3: ¿Cómo ajustar los parámetros del modelo λ para maximizar $P(O|\lambda)$?

El primer problema es un problema de evaluación. El tercer problema es necesario, porque en él se entrena al HMM. El segundo problema es de especial importancia en este trabajo, ya que conociendo la secuencia de estados y los instantes en que se presentaron, es posible hacer segmentación de habla.

3.8.2 Algoritmo de Viterbi

El algoritmo de Viterbi es una solución al problema 2 de los HMM. El criterio de maximización más utilizado es encontrar la secuencia de estados que maximice $P[Q|O,\lambda]$, que es equivalente a encontrar $P[Q, O|\lambda]$.

Se define la cantidad:

$$\delta_t(i) = max_{q_1, q_2, \dots, q_t - 1} P[q_1 q_2 \dots q_t = i, O_1, O_2, \dots, O_t | \lambda] \tag{3.34}$$

Dicha cantidad representa la máxima probabilidad siguiendo una secuencia de estados hasta el instante t, en el cual se han tenido t observaciones y la secuencia termina en el estado $q_t = S_i$. La cantidad $\delta_t(i)$ se calcula de manera recursiva:

$$\delta_{t+1}(j) = max_i[\delta_t(i)a_{ij}]b_j(O_{t+1}) \quad (3.35)$$

El algoritmo debe registrar los argumentos que maximizan la expresión 3.30, y al final debe dar marcha atrás para encontrar la secuencia de estados. El procedimiento completo del algoritmo de Viterbi es:

- Inicialización:

$$\delta_1(i) = \pi_i b_i(O_1) \quad 1 \leq i \leq N \quad (3.36a)$$
$$\psi_1(i) = 0 \quad (3.36b)$$

  Donde $\psi_1(i)$ es sólo un arreglo en el que se va registrando cada estado en la secuencia óptima. Este concepto no tiene nada que ver con la ψ con la que se representó a la función wavelet en secciones anteriores.

- Recursión:

$$\delta_t(j) = max_{1 \leq i \leq N}\left[\delta_{t-1}(i)a_{ij}\right]b_j(O_t) \qquad 2 \leq t \leq T, \ 1 \leq j \leq N \quad (3.37a)$$
$$\psi_t(j) = argmax_{1 \leq i \leq N}\left[\delta_{t-1}(i)a_{ij}\right] \qquad 2 \leq t \leq T, \ 1 \leq j \leq N \quad (3.37b)$$

- Terminación:

$$P^* = max_{1 \leq i \leq N}[\delta_T(i)] \quad (3.38a)$$
$$q_T{}^* = argmax_{1 \leq i \leq N}[\delta_T(i)] \quad (3.38b)$$

- Marcha atrás:

$$q_t{}^* = \psi_{t+1}(q_{t+1}{}^*) \qquad t = T-1, T-2, \ldots 1 \quad (3.39)$$

La alineación con el algoritmo de Viterbi es lo que se conoce como alineación forzada.

3.9 Conclusiones

El enfoque de este trabajo de tesis es implementar un mecanismo de segmentación para refinar las fronteras obtenidas mediante el método de alineación forzada. El mecanismo de segmentación acústica está basado en la transformada wavelet continua (CWT), donde las familias de wavelets están moduladas de acuerdo a las bandas críticas de la escala de Bark. En este trabajo de tesis se utilizan las wavelets Gaussianas moduladas en el algoritmo de Galka-Ziolko para encontrar los límites probables entre fonemas y con ellos refinar las fronteras obtenidas por alineación forzada.

3.10 Resumen

Cuando se evalúa el desempeño de una segmentación acústica normalmente se toman en cuenta la exactitud de la colocación de las fronteras, el número de inserciones y el número de borrados. Cuando la técnica asegura que el número de segmentos detectados es el correcto (como en la alineación forzada) normalmente se reporta el nivel de concordancia entre las etiquetas automáticas y las etiquetas manuales.
La escala de Bark relaciona frecuencias acústicas con frecuencias perceptuales y está basada en el fenómeno de enmascaramiento presente en el oído humano.
La transformada wavelet genera una representación escala-tiempo, similar a la frecuencia tiempo que genera la transformada de Fourier en tiempo corto (STFT), pero con mejoras, ya que la CWT permite análisis a diferentes resoluciones, mientras que la STFT mantiene la resolución fija durante todo el análisis.
La función wavelet Gaussiana modulada se modula en su frecuencia central, y se manipula su ancho de banda para que coincida con las bandas críticas de la escala de Bark. De esta manera se obtienen las wavelets

Gaussianas moduladas, que permiten un análisis wavelet tomando en cuenta las características perceptuales del oído humano.

El algoritmo de Galka-Ziolko segmenta la señal de habla de manera acústica, no fonética y se basa en la transformada wavelet discreta. Utiliza la wavelet symlet12 y realiza una descomposición diádica de seis niveles.

La alineación forzada se basa en un reconocedor del tipo HMM, y una búsqueda de programación dinámica Viterbi, en la cual el espacio de búsqueda está restringido a reconocer una cadena de fonemas conocida a priori.

CAPÍTULO 4. METODOLOGÍA Y DESARROLLO DE LA INVESTIGACIÓN

4.1 Introducción

Es importante aclarar la forma en la que se desarrolló este trabajo de tesis para que de esta forma alguien interesado pueda retomar el trabajo, reproducir los experimentos y darle continuidad al tema. Ese es precisamente el motivo de este capítulo en este trabajo de tesis.

Primero se presentan los aspectos de representación de los sonidos del habla, con la selección de un alfabeto fonético, además del diccionario de pronunciaciones. Posteriormente, se presenta el corpus de voces. Luego se presentan las operaciones de pre-procesamiento. Después pasamos al trabajo fuerte de la tesis, con la metodología para desarrollar el segmentador acústico con wavelet, e inmediatamente después, la obtención de la alineación forzada de fonemas.

4.2 Alfabeto fonético

El alfabeto fonético que se utilizó es Worldbet, ya que posee símbolos equivalentes a aquellos del IPA (Hiernonymus, J. L., 1993) necesarios para transcribir el español, y además existen varios ejemplos en la web de palabras en español etiquetadas con Worldbet. Para ver una descripción del alfabeto Worldbet para el español, así como otros alfabetos fonéticos, vea el anexo A.

4.3 Clases de fonemas y alófonos usados

Se utilizaron los fonemas hablados en México y solamente se distinguieron aquellos alófonos cuyas características fueran evidentemente diferentes respecto a los demás alófonos del mismo fonema. Por ejemplo, se distinguió entre /d/ y /D/, porque mientras /d/ es plosiva, /D/ es aproximante; por otro lado, no se distinguieron entre sí los distintos alófonos de la /n/, ya que las diferencias acústicas no se consideraron suficientemente evidentes como para distinguirlos.

Las clases utilizadas (expresadas en Worldbet) son: a, e, i, o, u, l, r(, r, j, m, n, w, n~, tS, dZ, p, t, k, b, d, g, V, D, G, f, s, x, sil, pau; donde pau se considera una pausa que ocurre antes de que ocurra la explosión en un fonema plosivo (lo que en inglés se conoce como *closure*). Estos eventos se suelen etiquetar con el diacrítico "_c", por ejemplo k_c, p_k, y t_c; pero dado que esencialmente son una pausa corta, es correcto etiquetarlos como pausa [35].

4.4 Diccionario de pronunciaciones

El diccionario de pronunciaciones representa todas las palabras del corpus utilizando el alfabeto Worldbet. Para ver una descripción detallada del diccionario de pronunciaciones utilizado en este trabajo de tesis vea el anexo B.

4.5 Corpus

Se desarrolló un corpus en español mexicano, el cual contiene todos los fonemas hablados en México. El corpus contiene 150 palabras repetidas 4 veces por un hablante. El formato utilizado es WAV, con 16 bits por

muestra y una frecuencia de muestreo de 16 KHz. Se utilizó este corpus como los archivos fuente de habla para los experimentos. El corpus es monoaural.
El sistema trabaja sobre corpus de palabras; es decir, cada archivo de audio contiene una palabra pronunciada, rodeada por un silencio al principio y un silencio al final.

También se utilizó un corpus de dígitos, donde las palabras son*: cero, uno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve*. El corpus incluye grabaciones de cuatro personas y tiene una frecuencia de muestreo de 11025 Hz, monoaural, con 16 bits. Estos mismos archivos también se utilizaron por separado; es decir, cuatro corpus de 200 palabras, de un solo hablante.

En el apéndice B se presenta el contenido fonético de cada uno de estos corpus.

4.6 Procedimiento para el etiquetado manual

Para segmentar manualmente una señal de voz lo más importante a tener en cuenta es identificar las regiones donde las características de la señal cambian, y las regiones donde las características se mantienen estables. Para ello nos podemos auxiliar tanto de las características en el dominio del tiempo como en el dominio de la frecuencia.

Como se indica en [42], el reconocimiento (por parte de un humano) de una consonante a través de su percepción depende esencialmente de la presencia de un cambio de frecuencias en sus elementos acústicos consitutivos, mientras que el de una vocal depende de la estabilidad de la frecuencia.

Para realizar el etiquetado manual se utilizó la herramienta SpeechViewer del CSLU toolkit. Básicamente se siguieron los siguientes pasos:

- Determinar la transcripción fonética.
- Insertar pausas antes de los fonemas plosivos y africados.
- Buscar fronteras entre fonemas usando el dominio del tiempo o el dominio de la frecuencia.
- Modificar las fronteras hasta que en cada segmento sólo se escuche su fonema correspondiente (en la medida de lo posible).

El primer paso para realizar el etiquetado manual es conocer la transcripción fonética de la locución; dicha transcripción se determina mediante reglas fonológicas, sin embargo, cuando se tenga la duda de cuál es la transcripción correcta para una palabra, existen en la web transcriptores fonéticos libres.

El segundo paso es insertar pausas en la transcripción. Antes de un fonema plosivo (*p, t,* y *k*) y antes del fonema africado *tS* se debe colocar una pausa *pau*; por ejemplo, la palabra *impacto* se transcribe como *i m pau p a pau k pau t o*. Otro ejemplo es la palabra *coche*, que se transcribe como *k o pau tS e*. Si el primer fonema de la palabra es *p, t, k,* o *tS*, entonces omitimos la pausa inicial, como en el ejemplo de la palabra *coche*.

El tercer paso es buscar las fronteras entre los fonemas más que los fonemas en sí; utilizando para ello las características en el dominio del tiempo o en el domino de la frecuencia. Algunas fronteras son muy fáciles de identificar ya sea por amplitud, por la aparición o desaparición de un formante, la presencia de una barra de explosión, etcétera, como se muestra en las figuras 4.1.a y 4.1.b. En este tipo de fronteras, donde la transición entre fonemas ocurre de manera abrupta, se marcaron las fronteras en los puntos donde ocurre un cambio evidente en las características de la señal.

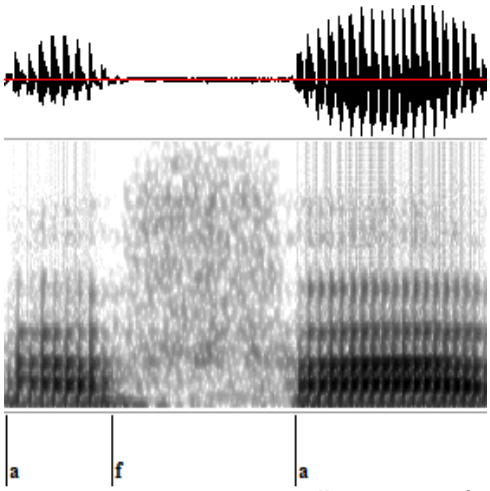


Figura 4.1.a. Fronteras sencillas entre /a/ y /f/.

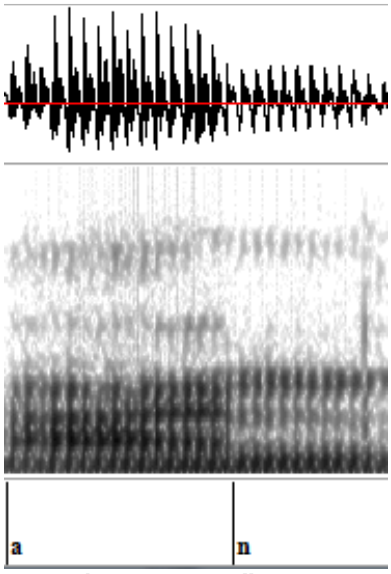


Figura 4.1.b. Frontera sencilla entre /a/ y /n/.

Una vez que se identifica una frontera se marca como tal, y se escuchan los segmentos que colindan con dicha frontera. Hay que mover la frontera hasta que se escuche un fonema sin escuchar al que está al lado. En las fronteras sencillas esto sí es posible, pero hay otros tipos de fronteras mucho más difíciles de marcar, en los cuales el sonido de los fonemas se traslapa; y donde existe un intervalo de transición continua entre los fonemas, como en las figuras 4.2.a y 4.2.b.

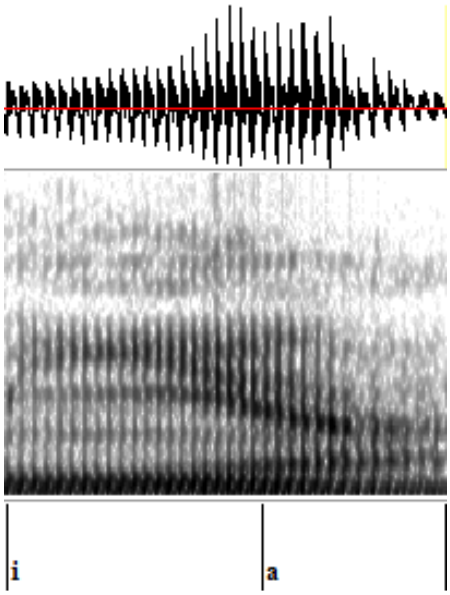


Figura 4.2.a. Frontera difícil de marcar entre las vocales /i/ y /a/.

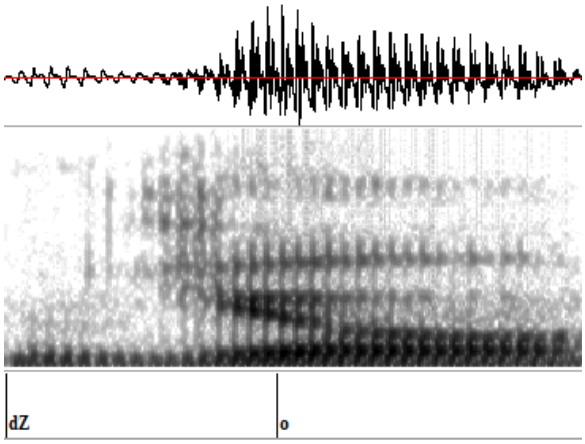


Figura 4.2.b. Frontera difícil de marcar entre /dʒ/ y /o/.

En el caso de fronteras difíciles de marcar, se decidió tomar aproximadamente el punto central del intervalo de transición entre los fonemas, como se puede apreciar en la figura anterior.

4.7 Opinión de un experto fonetista

Se estableció comunicación con el Mtro. Javier Cuétara Priede, experto fonetista y director del Centro de Enseñanza para Extranjeros (CEPE Taxco) de la UNAM, quien confirmó que los criterios utilizados en este trabajo de tesis para realizar la segmentación y etiquetado manuales son válidos.

4.8 Preprocesamiento

El preprocesamiento se compone del preénfasis, normalización y detección de inicio y fin de la palabra, como se presenta a continuación.

4.8.1 Preénfasis

El preénfasis es la técnica que comprime el rango dinámico de la señal aplanando la inclinación espectral de la misma. El efecto del preénfasis es aumentar la relación señal a ruido (SNR, del inglés Signal to Noise Ratio). El preénfasis se realiza enfatizando las componentes de alta frecuencia 6dB/octava. La ecuación 4.1 describe el preénfasis como un filtro digital:

$$H(z) = 1 - \alpha z^{-1} \tag{4.1}$$

Se escogió α = 0.97. Se implementó una función en Matlab para ejecutar el preénfasis de una señal.

4.8.2 Normalización

Los archivos de audio para el corpus deben contener señales cuya amplitud no sea demasiado pequeña, pero que tampoco sea tan grande como para saturarse. La normalización sirve para corregir esta insuficiencia [51]. En este caso, todas las muestras de la señal se escalan, para que la muestra cuyo valor absoluto es el mayor entre todas las muestras, después del escalamiento, tenga un valor absoluto de 1.

4.8.3 Detección de inicio y fin de palabra

Este es el problema de la localización de enpdoints (puntos de inicio o fin de locuciones). Se programaron dos funciones en Matlab que implementan el algoritmo de detección de endpoints propuesto en [43]. El algoritmo se basa en mediciones de energía y de cruces por cero para realizar la detección de los endpoints. El algoritmo requiere que los primeros 100 milisegundos en cada archivo de habla sean de silencio; de otra manera, el algoritmo fallará. En algunas ocasiones será deseable modelar los silencios, así que la etapa de detección de inicio y fin de palabra es opcional en el sistema.

4.9 Segmentación wavelet

En la etapa de segmentación wavelet se localizan fronteras probables entre fonemas a partir de cambios acústicos. En esta etapa del sistema se realiza una segmentación no constante. Dado que el método se basa en características acústicas reportará una tasa de inserción alta, lo cual no es un impedimento siempre y cuando entre esas fronteras propuestas por el algoritmo se encuentren también las fronteras reales entre los fonemas de la locución. Se busca que esta etapa encuentre los límites fonéticos, y por lo tanto los borrados son indeseables. Se deberá hacer un ajuste en los parámetros del algoritmo para evitar los borrados, haciendo un sacrificio con una tasa alta de inserciones.
En este trabajo se propone una combinación que no se había intentado, de técnicas para segmentación de habla tomando en cuenta aspectos perceptuales. Se proponen modificaciones a los algoritmos para que no sacrifiquen resolución en tiempo como en las versiones originales. Para cada vecindario, se encontró el valor del parámetro que minimiza el error, obteniéndose errores más pequeños que en la versión original, con la ventaja de que en esta propuesta no se toleran borrados de segmentos.

4.9.1 Modificación al algoritmo de Galka y Ziolko

El algoritmo de segmentación wavelet propuesto en este trabajo de tesis está basado en el trabajo de J. Galka y M. Ziolko, quienes realizaron segmentación acústica de un corpus en el idioma polaco [22].

En este trabajo se plantea modificar el algoritmo original de Galka-Ziolko de la siguiente manera:

- Se cambió el tipo de transformada wavelet. Originalmente se usaba la transformada discreta wavelet (DWT, del inglés *discrete wavelet transform*). Esta vez se usa una transformada wavelet continua (CWT).

- Se cambió el tipo de wavelet utilizada, de symlet-12 a wavelets Gaussianas moduladas en la escala de Bark. En lugar de realizar una descomposición diádica de 6 niveles, realizar una descomposición usando wavelets Gaussianas moduladas en escala de Bark con 17 niveles; esto es así porque en [32] el autor obtuvo los parámetros de escala y frecuencia de modulación para 17 niveles.
- Se elimina el paso llamado de "discretización de tiempo", ya la CWT es una transformada no decimada.
- La función de umbral se modifica (ver ecuación 3.24) para ser:

$$f_{tr}(k) = \alpha \sum_{n=-N}^{N} f(k-n) \tag{4.2}$$

Donde $f(k)$ es la función de detección de eventos, α es una constante de proporcionalidad y N es el rango de adaptación correspondiente al vecindario.

Las razones para implementar dichas modificaciones son:

- Las wavelets Gaussianas moduladas en la escala de Bark imitan las propiedades del sistema auditivo humano.
- La eliminación del paso de "discretización de tiempo" permite que no se sacrifique resolución temporal.
- La modificación de la ecuación 4.2 facilita la búsqueda del mejor valor del parámetro α, ya que de esa manera su valor tendrá que estar forzosamente entre 0 y 1.

El algoritmo modificado de Galka-Ziolko propuesto se describe en el siguiente pseudocódigo.

```
Segmentación acústica (señal de audio)
    1. Aplicar la CWT a la señal de audio utilizando wavelets Gaussianas
       moduladas en escala de Bark.
    2. Obtener el cuadrado del módulo de cada coeficiente wavelet.
    3. Para cada escala de análisis:
      3.1. Obtener la envolvente del cuadrado del módulo de los
           coeficientes wavelet.
      3.2. Suavizar la envolvente del cuadrado del módulo de los
           coeficientes wavelet.
    4. Crear el mapa de importancia ordenando de mayor a menor el
       cuadrado del módulo de cada coeficiente wavelet para cada
       instante.
    5. Aplicar la función de detección de eventos al mapa de importancia.
    6. Aplicar la función de umbral.
    7. Proponer como fronteras los instantes donde un evento supere al
    umbral.
```

4.9.2 Cálculo de la CWT

El Wavelet Toolbox® de Matlab permite realizar el cálculo de la CWT utilizando una interfaz común para todas las clases de wavelets. El toolbox permite agregar nuevas familias de wavelets, y utilizarlas para realizar transformadas de la misma manera en que se pueden usar las wavelets que ya vienen incluidas en el paquete. Los pasos para agregar una nueva familia wavelet son:

1. Escoger el nombre largo de la familia. En este caso se decidió por *Modulated Gaussian*.
2. Escoger el nombre corto de la familia. En este caso se escogió *mgau*.
3. Escoger el tipo de wavelet. Existen 5 tipos distintos:
   - Tipo 1: wavelets ortogonales con filtros FIR.
   - Tipo 2: wavelets biortogonales con filtros FIR.
   - Tipo 3: wavelets ortogonales sin filtro FIR, pero con función de escala.

- Tipo 4: wavelets sin filtro FIR y sin función de escala.
- Tipo 5: wavelets complejas sin filtro FIR y sin función escala.
  En este caso se escogió el tipo 5.

4. Definir los órdenes de las wavelets. Se escogió "0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16"; donde cada escalar se corresponde con una de las bandas de la escala de Bark. De esta manera, se pueden invocar a las wavelets mgau0, mgau1, mgau2,…, mgau16.
5. Construir la función que define a la wavelet en sí. Esta función se guarda como un archivo .m, y es llamada por Matlab cuando se calcula la CWT usando esta nueva wavelet.
6. Puesto que la función Gaussiana modulada no tiene soporte compacto, hay que escoger un soporte efectivo, esto es, un intervalo de valores, tales que para propósitos prácticos, la wavelet sea distinta de cero. Se escogió el intervalo [-1, 1].

4.9.3 Cálculo de la envolvente de los coeficientes wavelet

Los coeficientes de la CWT para una escala dada muestran variaciones muy rápidas que pueden degradar los resultados del algoritmo. El uso de envolvente y suavizado de la misma genera resultados mejores. Para calcular la envolvente se selecciona el valor máximo de los valores absolutos de una serie de coeficientes dentro de una ventana deslizante.

4.9.4 Suavizado de la envolvente de los coeficientes wavelet

El suavizado de los coeficientes se lleva a cabo con un filtro de medias móviles, especificado por

$$b = \frac{1}{5} ones(1,5) \tag{4.3}$$

4.9.5 Cálculo del mapa de importancia

Se acomodan los coeficientes wavelet en un renglón por cada escala, y luego cada renglón se concatena para formar una matriz con todos los coeficientes wavelets. El cálculo del mapa de importancia se realiza mediante una función que toma la matriz de coeficientes wavelets y genera otra matriz del mismo tamaño. Columna por columna, la función determina cuál subbanda fue la que tuvo más energía, y en la nueva matriz se acomodan los índices de cada subbanda de mayor a menor.

Sea el vector M_k, que contiene los niveles ordenados según el contenido de energía.

$$M_k = [m_1, m_2, m_3, \dots, m_{17}]^T \tag{4.4}$$

Es decir, la energía correspondiente a m_1 es mayor que la energía correspondiente a m_2, y así sucesivamente.

4.9.6 La función de detección de eventos

La detección de eventos asigna un valor a cada cambio en el ordenamiento de las subbandas. Si una columna y la columna siguiente en el mapa de importancia, son iguales, entonces la función de eventos asignaría un cero a ese evento; por otro lado, si el evento implica un cambio de ordenamiento en las subbandas, la función de detección de eventos asignará un valor positivo a dicho evento. Si el cambio ocurre en las bandas con mayor energía entonces se asignará un valor mayor, que si ocurre en las bandas de menor energía.

$$f(k) = \sum_{m=1}^{17} \left| \frac{M_{m,k+1} - M_{m,k}}{m} \right| \tag{4.5}$$

Donde M es la matriz del mapa de importancia.

4.9.7 Cálculo de los umbrales

Una vez que se ha calculado la función de eventos para toda la señal, hay que detectar aquellos eventos que sean lo suficientemente significativos. Un evento de magnitud grande corresponde con un cambio acústico grande; un evento de magnitud pequeña se corresponde con un cambio acústico pequeño. El propósito del cálculo de umbrales es seleccionar los eventos que hayan tenido magnitud grande dentro de un vecindario determinado. Puesto que los umbrales se calculan dependiendo del vecindario, el cálculo de umbrales se realiza de manera adaptativa. La ecuación para calcular los umbrales es la ecuación 4.2.

4.10 Alineación forzada

La alineación forzada es el método más comúnmente usado para la alineación automática de habla. En dicho método se realiza un reconocimiento de habla restringiendo la búsqueda a la secuencia conocida de fonemas. La búsqueda produce la localización de los fonemas, así como su identidad. Estos sistemas son llamados de alineación forzada porque se obliga a que el resultado del reconocimiento sea la secuencia fonética propuesta, la cual se determina previamente utilizando un diccionario de pronunciaciones, reglas grafema a fonema o por un humano.
De acuerdo a Hosom [28], la alineación forzada se puede implementar con modelos ocultos de Markov (HMM, por el inglés Hidden Markov Model). Utilizando los HMM, se aplica una búsqueda Viterbi restringida a únicamente la secuencia correcta de fonemas. El resultado de la búsqueda Viterbi contiene la alineación fonética, así como también el puntaje para la secuencia correcta de fonemas.
En este trabajo de tesis se escogió utilizar modelos ocultos de Markov discretos para implementar la alineación forzada. Los fonemas son modelados mediante tres estados. En esta etapa del sistema se realiza una segmentación constante.

4.10.1 Modelos ocultos de Markov discretos

Los modelos ocultos de Markov pueden ser discretos, continuos o semicontinuos. Se eligieron los modelos discretos por su sencillez. Se utilizó el HMM Toolkit de Kevin Murphy [38], el cual está escrito en Matlab.
Los vectores de parámetros a utilizar son los MFCC, con posibilidad a utilizar las primeras y segundas deltas. Se utilizó el toolkit Rastamat de Dan Ellis [13] para calcular los vectores MFCC.
El libro de código se calculó utilizando el algoritmo de KMeans con k = 2; esto para generar un árbol binario completo; por lo tanto el tamaño del libro de código sólo puede ser de 2^n vectores. En los experimentos se utilizó un libro de código de 256 vectores.
Los componentes de un modelo oculto de Markov son: el número de estados, el número de símbolos de observación (tamaño del alfabeto discreto), la distribución de probabilidad de transiciones A, la distribución de probabilidad de emisiones B y la distribución de estado inicial π. A continuación se explican las elecciones de los componentes.

4.10.2 Modelado con tres estados

Se escogió utilizar tres estados para modelar cada fonema, como aparece en [52]. Los modelos son de tres estados y son de tipo Bakis (de izquierda a derecha). Se permite una transición directa desde el primer estado hasta el último estado, como se muestra en la figura 4.3.

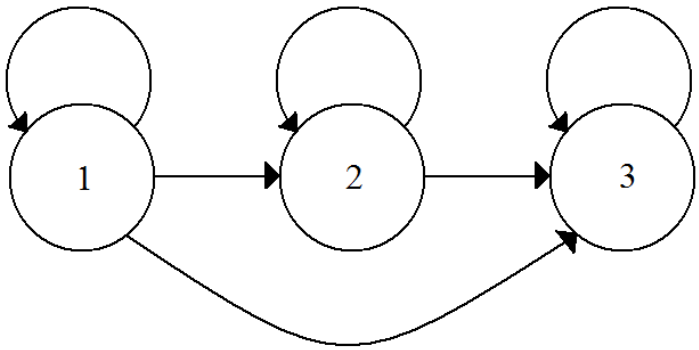


Figura 4.3. Modelo de Markov de tres estados.

La matriz de probabilidades de transición adquiere entonces la forma siguiente:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} \quad (4.6)$$

Donde a_{ij} es la probabilidad de que el estado anterior sea i y que el estado siguiente sea j. Se puede apreciar que el modelo no permite una transición desde un estado dado hacia un estado anterior.

Una vez que se han entrenado los modelos HMM a nivel de fonemas se procede a generar HMM a nivel de palabras. Para ello se necesita la transcripción fonética de la palabra; y basándonos en eso se concatenan los HMM de fonemas necesarios para modelar a la palabra. Una vez que se ha construido el modelo a nivel de palabra, se reestima utilizando las palabras disponibles en el corpus. Para concatenar dos o más HMM de nivel de fonemas es necesario agregarles estados *dummy* o estados de enlace [4]; uno de entrada y otro de salida. A través de los estados *dummy* se unen los modelos de Markov, como se muestra en la figura 4.4 y en la ecuación 4.7.

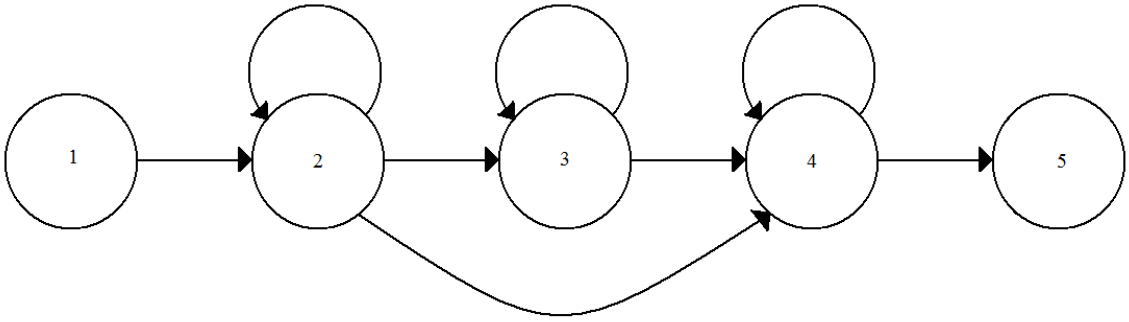


Figura 4.4. Modelo de Markov de tipo Bakis con estados *dummy*.

La matriz de probabilidades de transición se vuelve entonces:

$$A = \begin{bmatrix} 0 & a_{12} & 0 & 0 & 0 \\ 0 & a_{22} & a_{23} & a_{24} & 0 \\ 0 & 0 & a_{33} & a_{34} & 0 \\ 0 & 0 & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.7)$$

Un ejemplo de la matriz de probabilidad de transición de un modelo HMM con sus estados dummy es [61]:

$$A = \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0 & 0.0 \\ 0.0 & 0.6 & 0.4 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.6 & 0.4 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.7 & 0.3 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \quad (4.8)$$

4.10.3 Procedimiento para obtener secuencias de entrenamiento

Los pasos para obtener las secuencias de entrenamiento son:

- Generar un conjunto de etiquetas manuales. Hay que tener cuidado que las etiquetas manuales incluyan a todos los tipos de fonemas que se están tomando en cuenta. En el español hablado en México se utilizan 22 fonemas, pero se decidió utilizar 29 clases.

Los archivos de etiquetas tienen la extensión .phn.

- Generar vectores de características MFCC. Está la opción de incluir la energía, además de utilizar los coeficientes delta y los coeficientes de aceleración de los parámetros.

Se escribió el programa *CalcularVectoresPorFonemas* en Matlab, el cual lee cada archivo de etiquetas, determina los puntos de inicio y fin de cada fonema en cada palabra etiquetada, recorta esa sección de audio, le calcula los vectores MFCC y guarda esa secuencia de vectores como un archivo de texto. Cada clase de fonema debe tener un directorio específico donde se guardarán las secuencias de vectores MFCC (ver la figura 4.5).

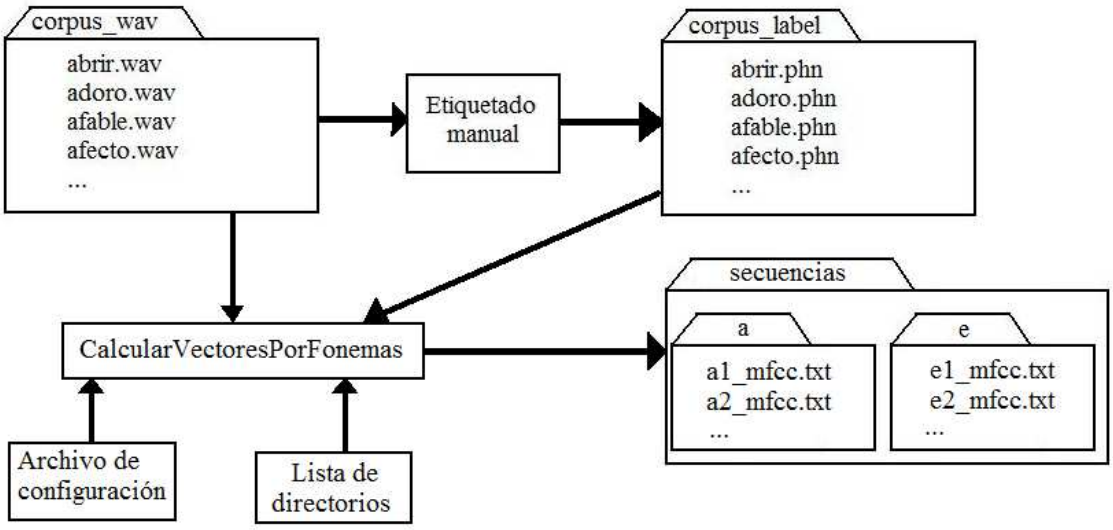


Figura 4.5. Procedimiento para obtener secuencias de entrenamiento.

Los vectores MFCC son calculados utilizando el toolkit *rastamat* de Dan Ellis [13]. El programa *CalcularVectoresPorFonemas* hace llamadas al toolkit *rastamat*.

4.10.4 Procedimiento para obtener el libro de código

Se implementó una clase en Matlab para construir árboles binarios. El árbol almacena en cada nodo un vector MFCC. El tipo de árboles que se pueden construir son árboles binarios completos, es decir, cada nodo tiene dos hijos excepto las hojas, además, cada nivel está completamente lleno, como en la figura 4.6.

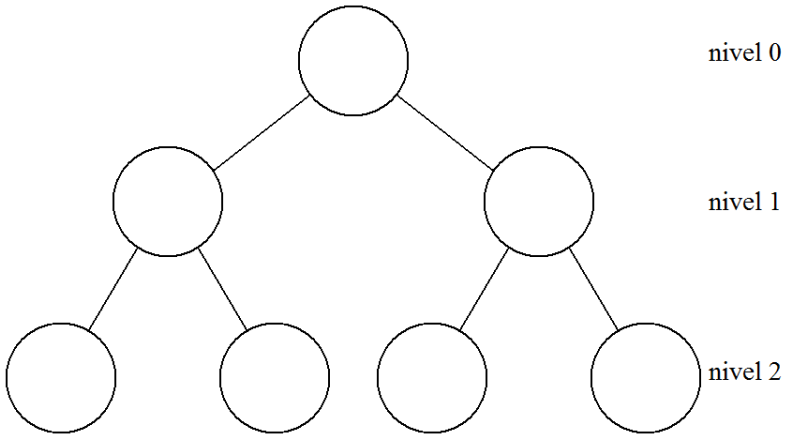


Figura 4.6. Árbol binario completo.

El árbol binario permite búsquedas muy rápidas, aunque el precio que hay que pagar es que hay que almacenar el doble de vectores de los que realmente se necesitan; por ejemplo, el árbol de la figura anterior puede almacenar un libro de código de 4 vectores (los cuales se almacenan en las hojas), pero necesita almacenar en memoria un total de 7 vectores.

Para encontrar los vectores del libro de código se siguió este procedimiento:

- Las secuencias de vectores MFCC almacenadas en directorios especiales, se mezclaron en un solo archivo.
- A partir del nuevo archivo generado, el cual almacena muchos vectores, hay que particionar esos vectores utilizando el algoritmo de KMeans, con k = 2. La partición se realiza de manera recursiva. Primero los vectores se particionan en dos grupos y se encuentra el centroide para cada uno de esos dos grupos; dichos centroides son vectores que se guardan en los primeros dos nodos hijos del árbol binario. Posteriormente, cada grupo recién creado se vuelve a particionar en dos grupos, y los centroides correspondientes se almacenan en nuevos nodos del árbol binario. El proceso continúa hasta que se hayan llenado todos los nodos del árbol. El tamaño del árbol debe establecerse previamente.
- Para almacenar el libro de código en un archivo, se hace un recorrido en el orden en que fue creado el árbol binario. Cada vez que se visita un nodo, el contenido de ese nodo se escribe en el archivo.
- Para cargar un libro de código a memoria a partir de un archivo, primero hay que crear el árbol binario con el tamaño adecuado. Luego, sabiendo que los vectores del archivo están acomodados en preorden, se realiza un recorrido en ese mismo orden en el nuevo árbol. Cada vez que se accede a un nodo, el vector correspondiente se carga a dicho nodo. El recorrido que se hace en el árbol es siempre en preorden.

Se escribieron dos programas en Matlab. El programa ConcatenarVectoresV2 se encarga de concatenar diversos vectores MFCC en un solo archivo. El programa CrearLibroCodigo lee el archivo con todos los vectores MFCC mezclados, y se encarga de particionarlos para generar el libro de código como se explicó arriba.

El procedimiento para crear un libro de código y almacenarlo en un archivo se ilustra en la figura 4.7.

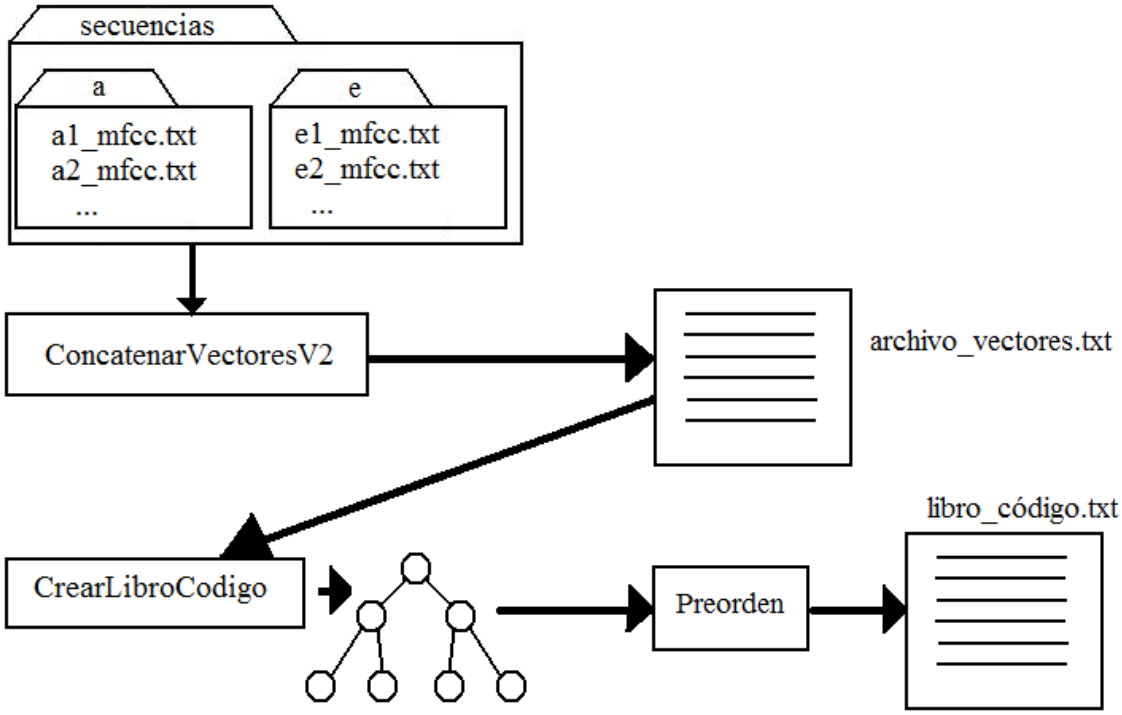


Figura 4.7. Procedimiento para crear un libro de código y almacenarlo en un archivo.

En los experimentos realizados siempre se utilizó la misma configuración para el árbol binario:

- Profundidad: 8.
- 256 hojas.
- Recorrido en preorden (primero raíz, luego hijo izquierdo y por último, hijo derecho).

4.10.5 HMM Toolbox

El HMM Toolbox de Kevin Murphy ofrece un conjunto de funciones para implementar modelos ocultos de Markov. En particular interesan las siguientes funciones:

- dhmm_em. Entrena un HMM utilizando un conjunto de secuencias de entrenamiento.
- viterbi_path. Calcula la secuencia de estados más probable, dado un HMM y una secuencia de observaciones.

4.10.6 Procedimiento para entrenar el HMM

Para entrenar un HMM discreto hacen falta secuencias de observaciones; traducir cada vector de cada secuencia de observaciones a un índice correspondiente en el libro de código; y también se necesita tener un estimado inicial para la matriz de probabilidad de transiciones, para la matriz de probabilidad de observaciones y para el vector de probabilidad de estado inicial.
Dado que se está trabajando sobre un modelo Bakis (de izquierda a derecha) el vector de probabilidad de estado inicial siempre es cero excepto en su primer estado:

$$\pi_i(1) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{4.9}$$

El entrenamiento de los HMM se realiza en dos etapas. En la primera etapa se crean los modelos a nivel de fonemas utilizando las secuencias de vectores MFCC producidas por el programa *CalcularVectoresPorFonemas*. En la segunda etapa los modelos a nivel de fonema se concatenan de acuerdo a una transcripción fonética, el modelo concatenado se reentrena pero esta vez no con secuencias de MFCCs correspondientes a fonemas, sino con secuencias de MFCCs correspondientes a palabras completas.

La primera etapa se implementó con el programa *EntrenarHMMFonemas*. La segunda etapa se implementó con el programa *EntrenarHMMPalabras*.

Para entrenar el HMM es necesario proporcionarle el libro de código y las secuencias de entrenamiento, como se muestra en las figuras 4.8 y 4.9.

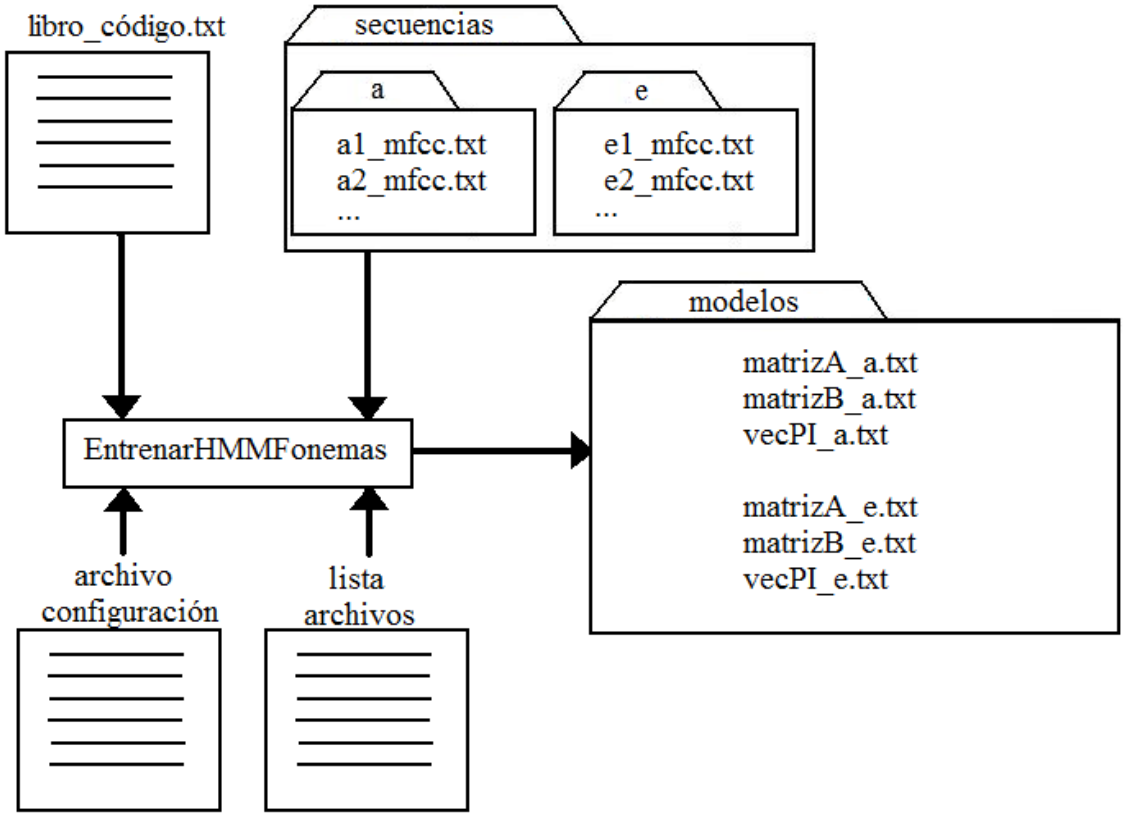


Figura 4.8. Procedimiento para entrenar los HMM a nivel de fonemas.

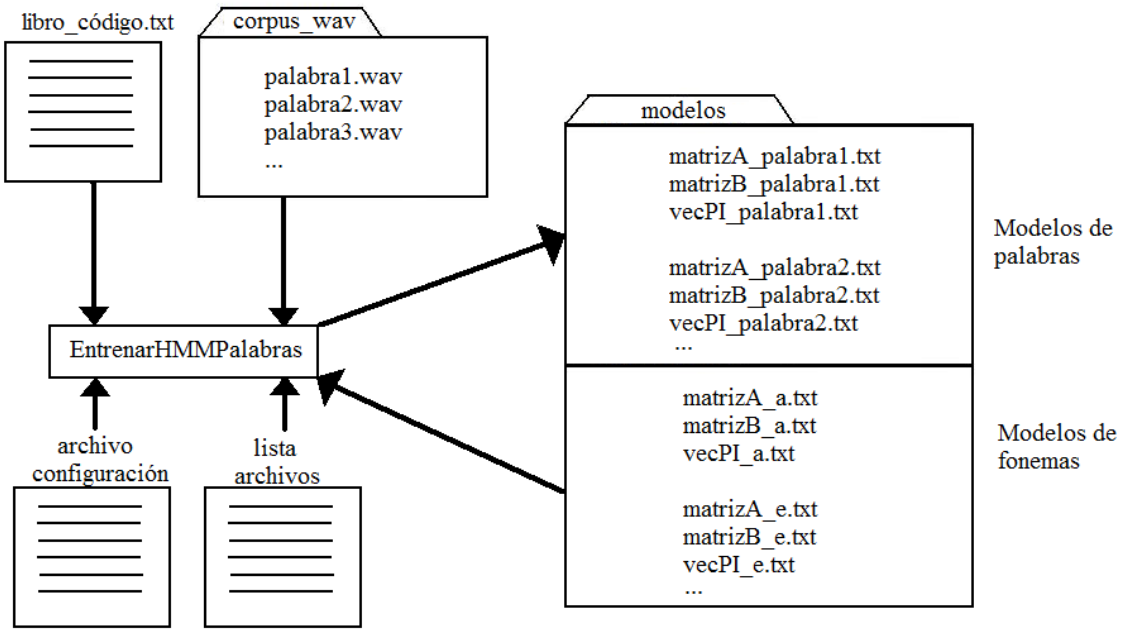


Figura 4.9. Procedimiento para entrenar los HMM a nivel de palabras.

4.10.7 Algoritmo de Viterbi

Dada una secuencia de observaciones y un modelo HMM entrenado, el algoritmo de Viterbi encuentra la secuencia de estados óptima que se ajuste a dicha secuencia de observaciones.

El algoritmo de Viterbi se obtiene utilizando la función viterbi_path del HMM toolbox.

El procedimiento para aplicar el algoritmo de Viterbi a una elocución es como sigue:

- Se carga el archivo wav de la palabra.
- Se carga el HMM de la palabra. En esta etapa hay que asegurarse de que la matriz de probabilidad de observaciones no contenga ningún cero; si se detecta un cero hay que sustituirlo por un valor pequeño. En este caso cuando se encuentra un cero se sustituye por 0.000001. Si se intenta hacer Viterbi cuando la matriz de probabilidad de observaciones tiene ceros, habrá errores en la ejecución porque muchos logaritmos de las probabilidades tenderán hacia infinito negativo.
- Se calcula la secuencia de vectores MFCC correspondiente al audio.
- Se aplica Viterbi sobre la secuencia de vectores MFCC, utilizando para ello el modelo (A, B, π) de la palabra. Viterbi entrega una secuencia de estados.
- Conociendo el número de estados por fonema, y la tasa de marco (lo que en inglés se conoce como *frame rate*), el algoritmo de Viterbi permite conocer los instantes en los que ocurrió cada fonema.

Se implementó el procedimiento anteriormente descrito en el programa *AlinearPalabrasV2*. Su uso se muestra en las figuras 4.10 y 4.11.

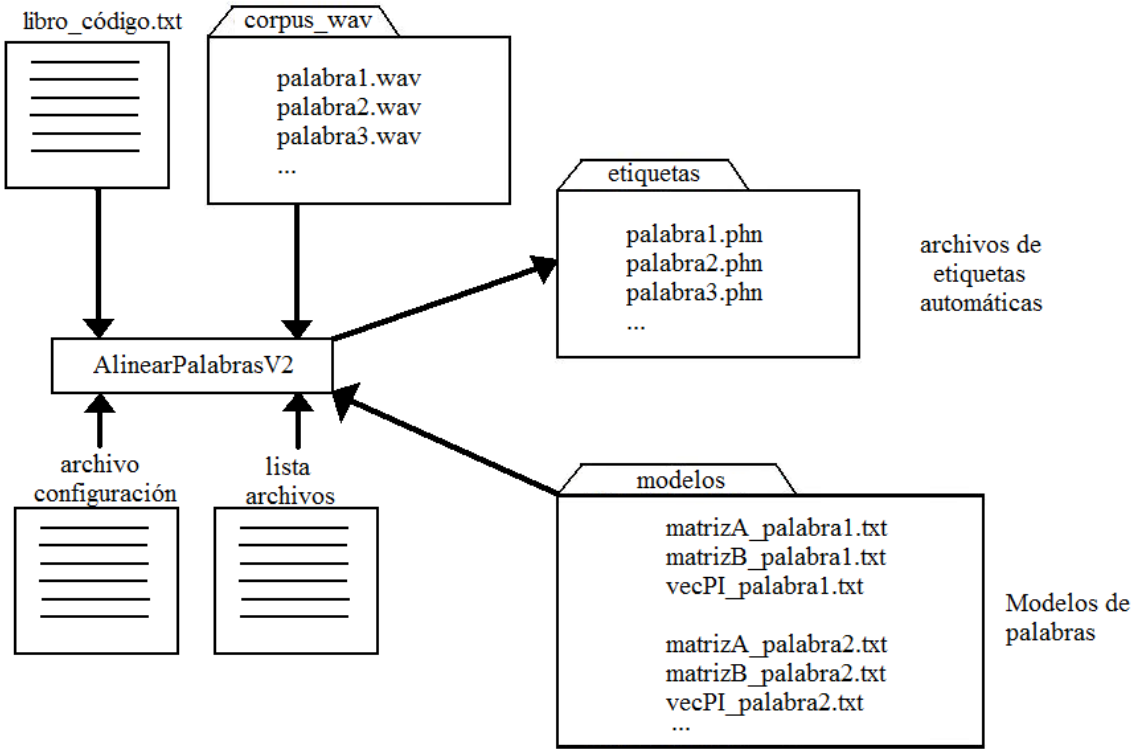


Figura 4.10. Procedimiento para generar las etiquetas automáticas.

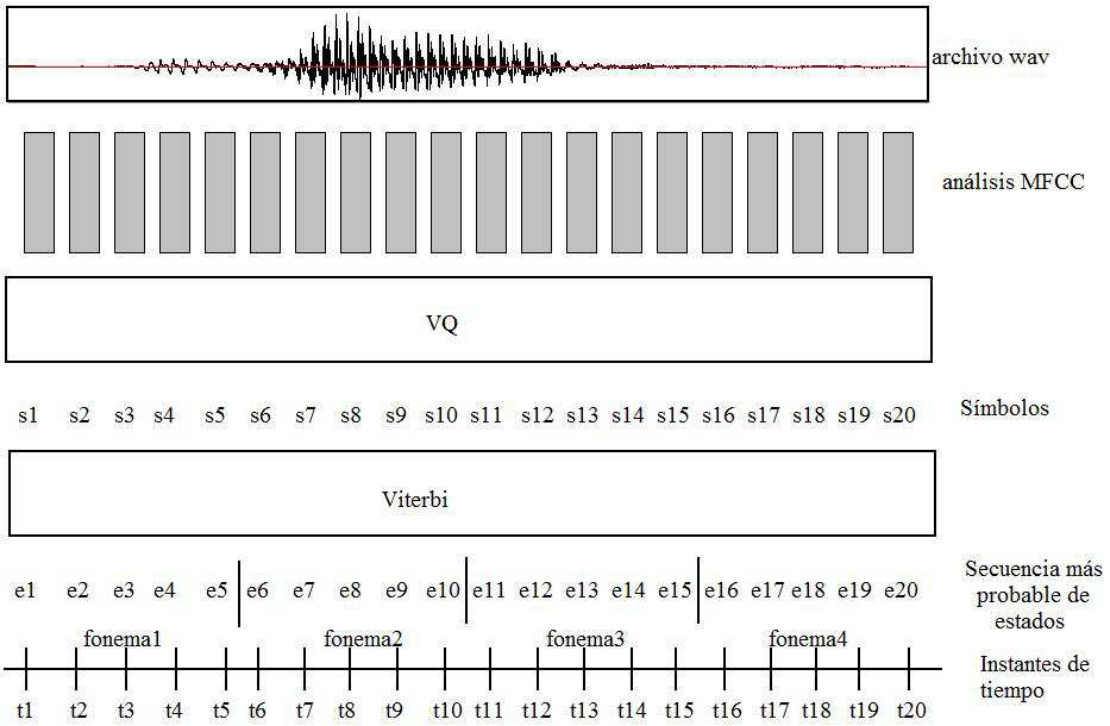


Figura 4.11. Alineación forzada con el algoritmo de Viterbi.

4.11 Diseño del sistema propuesto

El sistema propuesto contiene tres etapas principales: realizar la alineación forzada, realizar segmentación acústica wavelet, y seleccionar fronteras, como se muestra en la figura 4.12.

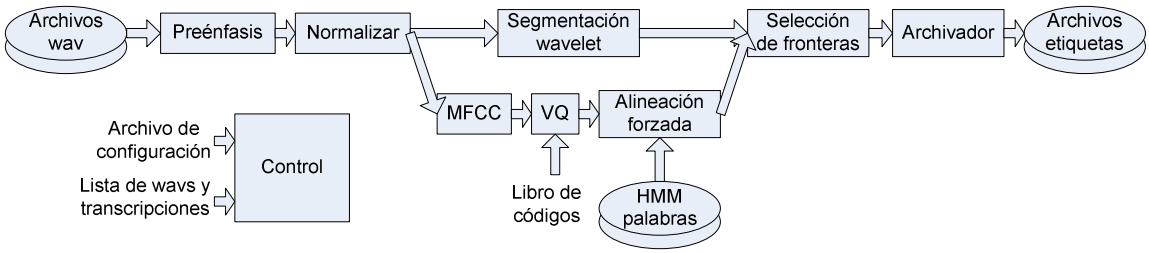


Figura 4.12. Diagrama de bloques del sistema propuesto.

El bloque de preénfasis enfatiza los componentes de alta frecuencia, lo cual es efectivo en aumentar la relación señal a ruido. El bloque de normalización disminuye la variación en la intensidad de las grabaciones. El bloque de segmentación wavelet realiza una segmentación acústica de la señal. De manera paralela al bloque de segmentación wavelet se encuentran los bloques para lograr segmentación por alineación forzada. El primero es el bloque de parametrización MFCC, que calcula dichos vectores para la señal de voz; le sigue el bloque de cuantificación vectorial, que sustituye la secuencia de vectores por una secuencia de símbolos; y por último tenemos el bloque de alineación forzada, donde se aplica el algoritmo de Viterbi a un modelo HMM de una palabra utilizando la secuencia de símbolos que se acaba de obtener; el algoritmo de Viterbi produce una secuencia de estados que se traduce en fronteras entre fonemas. El bloque de selección de fronteras toma la segmentación propuesta por alineación forzada y la refina usando la segmentación acústica, produciendo la segmentación definitiva. El bloque archivador guarda en un archivo de etiquetas las fronteras entre fonemas y la transcripción fonética correspondientes a la palabra. El bloque de control representa todas las demás partes de los programas, que coordinan la operación de los bloques antes descritos.

Conviene hablar un poco más acerca del bloque de selección de fronteras. Tanto la alineación forzada como la segmentación acústica proponen fronteras. Se le da preferencia a la alineación forzada, esto es, para cada frontera propuesta por la alineación forzada se busca la frontera acústica más próxima; lo anterior únicamente ocurre cuando la combinación fonema izquierdo – fonema derecho está permitida (tipo de frontera permitido); si el tipo de frontera no está permitido se utiliza la frontera de alineación forzada. Una vez que se encontró la frontera acústica más próxima, el sistema debe cerciorarse de que dicha frontera acústica en verdad corresponda a la frontera de alineación forzada que se está tratando en ese instante. Si la frontera acústica más cercana se encuentra a menos de 20 ms respecto a la frontera de alineación forzada actual, y además las fronteras verdaderamente se corresponden entre sí, entonces se toma la frontera acústica, desechando la frontera fonética propuesta por la alineación forzada; en cualquier otro caso se desecha la frontera acústica y se escoge la frontera fonética de alineación forzada.

El funcionamiento del sistema se ilustra como pseudocódigo en las siguientes líneas.

```
Inicio
Cargar el archivo wav.
Realizar preénfasis.
Realizar normalización.
Cargar la transcripción fonética.
Realizar alineación forzada:
  Calcular vectores MFCC.
  Utilizar cuantificador vectorial.
  Cargar HMM de la palabra.
  Aplicar algoritmo de Viterbi.
  Proponer fronteras basadas en alineación forzada.
Realizar segmentación acústica:
  Calcular CWT.
  Envolvente y suavizado de coeficientes wavelet.
  Calcular mapa de importancia.
  Calcular función de eventos.
  Aplicar umbrales.
  Proponer fronteras basadas en segmentación acústica.
Seleccionar fronteras:
  Para cada frontera propuesta por alineación forzada:
    Si el tipo de frontera es válido Entonces:
      Buscar la frontera acústica más cercana.
      Si la frontera acústica y la fonética se corresponden Entonces:
        Si distancia entre fronteras <= 20 ms Entonces:
          Conservar la frontera acústica.
        Si distancia entre fronteras > 20 ms Entonces:
          Conservar la frontera de alineación forzada.
        Fin Si
      Si las fronteras no se corresponden Entonces:
        Conservar la frontera de alineación forzada.
      Fin Si
    Si el tipo de frontera no es válido Entonces:
      Conservar la frontera de alineación forzada.
    Fin Si
Generar archivo de etiquetas usando las fronteras.
Fin
```

El procedimiento de selección de fronteras se muestra en el diagrama de flujo de la figura 4.13.

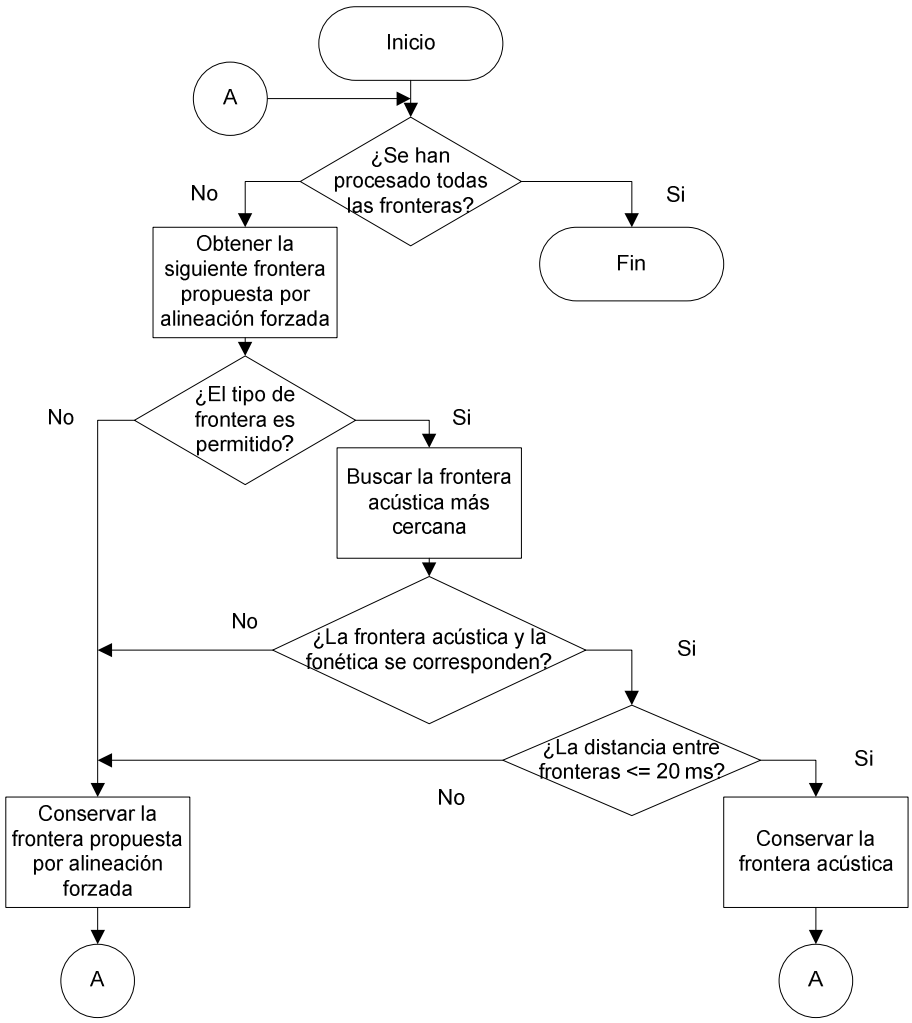


Figura 4.13. Selección de fronteras.

4.12 Programas realizados en Matlab

Se implementaron varios programas en Matlab, cada uno con un propósito específico. Estos programas permiten utilizar únicamente el alfabeto worldbet, con las clases de fonemas que se mencionaron en la sección 4.3. La independencia respecto al conjunto de clases de fonemas utilizados y respecto al alfabeto fonético se dejará como trabajo a futuro.

Para utilizar los programas es necesario cerciorarse que la variable path dentro de Matlab cuenta con las rutas hacia el HMM toolbox, y hacia el rastamat toolkit. A continuación se detalla el uso de los programas.

4.12.1 CalcularVectoresPorFonemas

Este programa calcula los vectores MFCC por fonema (sin considerar su contexto) a partir de archivos wav etiquetados. Esta función abre los archivos .wav que sí están etiquetados y los recorta fonema por fonema; a cada fonema recortado le obtiene su secuencia de vectores MFCC (con posibilidad de usar la energía, los coeficientes delta y los coeficientes de aceleración). Cada secuencia de vectores MFCC se guarda en un

archivo por separado. Existe un directorio especial para cada fonema en donde se irán depositando los archivos con los coeficientes MFCC.

La función es como sigue:

```
CalcularVectoresPorFonemas  (directorio_phn,
                            directorio_wav,
                            archivo_config,
                            lista_directorios)
```

Donde:

- directorio_phn: es el directorio donde se encuentran los archivos de etiquetas elaborados manualmente.
- directorio_wav: es el directorio donde se encuentran todos los archivos de audio en formato .wav.
- archivo_config: es el archivo de texto que indica cómo calcular los vectores MFCC.
- lista_directorios: es el archivo de texto que indica en qué directorio deben guardarse los archivos MFCC para cada fonema.

El archivo archivo_config tiene la estructura siguiente:

```
-tam_ventana_ms 20
-tam_traslape_ms 2
-min_frec_hz  0
-max_frec_hz  8000
-num_ceps     14
-num_bandas   24
-cep_lifter -22
-usar_preenfasis 1
-coef_preenfasis 0.97
-usar_deltas   1
-num_frames_deltas 2
-usar_accel  1
-num_frames_accel 1
-usar_energia 1
```

Básicamente dicha configuración indica: el tamaño de las ventanas para calcular los vectores será de 20 ms (-tam_ventana_ms 20), la tasa de marcos será de 2 ms (-tam_traslape_ms 2), la frecuencia mínima a tomar en cuenta en los filtros es de 0 Hz (-min_frec_hz 0), la frecuencia máxima a tomar en cuenta en los filtros es de 8000 Hz (-max_frec_hz 8000), el número de coeficientes cepstrales será de 14 (-num_ceps 14), el número de bandas usadas en los filtros MFCC será de 24 (-num_bandas 24), se usará un coeficiente liftrado de -22 (-cep_lifter -22, esto es así porque el toolkit rastamat le cambia de signo), sí se utilizará preénfasis (-usar_preenfasis 1), el coeficiente de preénfasis utilizado es de 0.97 (-coef_preenfsasis 0.97), sí se usarán los coeficientes detlta del vector MFCC (-usar_deltas 1), el número de marcos a tomar en cuenta para calcular las los coeficientes delta será 2 (-num_frames_deltas 2), sí se utilizarán los valores de los coeficientes de aceleración de los vectores MFCC (-usar_accel 1), el número de marcos a tomar en cuenta para calcular los coeficientes de aceleración de los vectores MFCC será de 1 (-num_frames_accel 1), sí se calculará la energía del marco (-usar_energia 1).

4.12.2 ConcatenarVectoresV2

Este programa simplemente concatena vectores producidos para diferentes fonemas en un solo archivo. El objetivo es tener en un solo archivo un conjunto de vectores MFCC que sirva para generar un libro de código. La función es la siguiente:

```
[total_vectores]    =ConcatenarVectoresV2  ( lista_archivos,
                                           archivo_destino )
```

Donde:

- total_vectores: es el número de vectores que se concatenaron y copiaron al archivo de salida.
- lista_archivos: es un archivo de texto que indica cuántos archivos se tomarán en cuenta de cada tipo de fonema para copiar su contenido.
- archivo_destino: es el archivo de salida a ser creado.

4.12.3 CrearLibroCodigo

Este programa recibe un conjunto de vectores MFCC en un solo archivo, y a partir de dichos vectores genera un libro de código de tamaño 2^n. La función es como sigue:

```
[libro_codigo]  =CrearLibroCodigo  (tam_libro_codigo,
                                    archivo_libro_codigo,
                                    archivo_vectores,
                                    tam_vectores,
                                    cantidad_vectores)
```

Donde:

- tam_libro_codigo: es el tamaño del libro de código, es decir, el número de vectores que lo componen. Este valor debe ser potencia de dos.
- archivo_libro_codigo: es el nombre del archivo donde se desea escribir el nuevo libro de código.
- archivo_vectores: es el archivo que contiene la colección de vectores MFCC mezclados. Este archivo es generado por el programa *ConcatenarVectoresV2*.
- tam_vectores: es el número de componentes que contiene cada vector MFCC. Todos los vectores deben ser de la misma longitud.
- cantidad_vectores: es el número de vectores contenidos en el archivo *archivo_vectores*. Este valor es generado por el programa *ConcatenarVectoresV2* cuando termina de ejecutarse.

4.12.4 EntrenarHMMFonemas

Este programa genera modelos HMM para cada fonema sin tomar en cuenta su contexto. Utiliza las mismas secuencias de vectores MFCC generadas por el programa *CalcularVectoresPorFonema*, además de que utiliza el libro de código generado por *CrearLibroCodigo*. Este programa utiliza el HMM Toolkit de Kevin Murphy [38]. El programa recibe una lista de archivos y abre uno a uno de ellos; para producir la matriz de probabilidad de transiciones A, la matriz de probabilidad de observaciones B, y el vector de probabilidad inicial π.
La función es la siguiente:

```
EntrenarHMMFonemas  (archivo_config,  lista_archivos )
```

Donde:

- archivo_config: es un archivo de texto que indica la ubicación y características del libro de código, el número de estados para modelar cada fonema (sin contar estados dummy), así como la ubicación para guardar los modelos.
- lista_archivos: es un archivo de texto que indica la ubicación y el número de archivos a tomar para realizar el entrenamiento.

El archivo de configuración tiene esta estructura:

```
-archivo_libro_codigo C:\Users\libros_codigo\libro_codigo_256.txt
-tam_libro_codigo     256
```

```
-tam_vectores         30

-num_estados_por_fonema 3
-max_iter_entrenamiento 200
```

La configuración básicamente indica lo siguiente: cargar el libro de código señalado (-archivo_libro_codigo C:\Users\libros_codigo\libro_codigo_256.txt), el libro de código consta de 256 vectores (-tam_libro_codigo 256), cada vector MFCC consta de 30 componentes (-tam_vectores 30), los modelos de Markov contendrán 3 estados (-num_estados_por_fonema 3), se utilizarán máximo 200 iteraciones para el entrenamiento (-max_iter_entrenamiento 200), se señala el directorio donde se guardarán los modelos (-directorio_modelos C:\Users\modelos_markov\). Además, la matriz de probabilidad de transiciones para el fonema /a/ se guardará bajo el nombre *matrizA_a.txt*, la matriz de probabilidad de observaciones para el fonema /a/ se guardará bajo el nombre *matrizB_a.txt*, el vector de probabilidad inicial para el fonema /a/ se guardará bajo el nombre *vecPI_a.txt*. Las demás líneas indican lo mismo para el resto de los fonemas.

El archivo lista_archivos tiene la siguiente estructura:

```
-directorio_fuente  C:\Users\vectores_fonemas\a\
-nombre_fonema a
-numero_archivos 560

-directorio_fuente  C:\Users\vectores_fonemas\e\
-nombre_fonema e
-numero_archivos 256

-directorio_fuente  C:\Users\vectores_fonemas\i\
-nombre_fonema i
-numero_archivos 108
…
```

Y así sucesivamente para todas las clases de fonemas.

El archivo indica lo mismo que el archivo *lista_archivos* usado por *ConcatenarVectoresV2*.

4.12.5 EntrenarHMMPalabrasV2

Este programa genera modelos HMM a nivel de palabra. El programa toma HMM a nivel de fonema y los concatena de acuerdo a una transcripción fonética proporcionada. El HMM de nivel de palabra es reentrenado utilizando archivos de audio de la palabra en cuestión.
La función es:

```
EntrenarHMMPalabrasV2  ( archivo_config, lista_archivos )
```

Donde:

- archivo_config: es un archivo de texto que indica qué parámetros utilizar para calcular los vectores MFCC, también indica las características del libro de código, la ubicación de los archivos de audio y de los modelos.
- lista_archivos: es un archivo de texto que indica el patrón en general que tienen los nombres de los archivos wav a ser abiertos, también indica cuántos de esos archivos abrir y la transcripción fonética que les corresponde.

La estructura del archivo de configuración es como sigue:

```
-recortar_silencios 0
-considerar_silencios 1

-tam_ventana_ms     20
```

```
-tam_traslape_ms    2
-min_frec_hz        0
-max_frec_hz        8000
-num_ceps           14
-num_bandas         24
-cep_lifter         -22
-usar_preenfasis    1
-coef_preenfasis    0.97
-usar_deltas        1
-num_frames_deltas  2
-usar_accel         0
-num_frames_accel   0
-usar_energia       1

-archivo_libro_codigo C:\Users\libros_codigo\libro_codigo_256.txt
-tam_libro_codigo     256
-tam_vectores         30

-num_estados_por_fonema 3
-max_iter_entrenamiento 200
```

El significado de estos parámetros es: no recortar silencios utilizando el algoritmo de Rabiner-Sambur (-recortar_silencios 0), agregar silencios automáticamente a las transcripciones (-considerar_silencios 1), el tamaño de los marcos para calcular los vectores MFCC es de 20 ms (-tam_ventana_ms 20), la tasa de los marcos es de 2 ms (-tam_traslape_ms 2), la frecuencia mínima a tomar en cuenta para calcular los vectores MFCC es de 0 Hz (-min_frec_hz 0), la frecuencia máxima a tomar en cuenta para calcular los vectores MFCC es de 8000 Hz (-max_frec_hz 8000), el número de coeficientes es 14 (-num_ceps 14), el número de bandas en los filtros es de 24 (-num_bandas 24), el coeficiente de liftrado es de -22 (-cep_lifter -22), utilizar preénfasis (-usar_preenfasis 1), el coeficiente de preénfasis es de 0.97 (-coef_preenfasis 0.97), calcular los coeficientes delta de los vectores MFCC (-usar_deltas 1), tomar en cuenta dos marcos hacia adelante y hacia atrás para calcular los coeficientes delta (-num_frames_deltas 2), no utilizar los coeficientes de aceleración de los vectores MFCC (-usar_accel 0), calcular la energía de cada marco (-usar_energia 1). También se indica cuál archivo es el libro de código (-archivo_libro_codigo C:\Users\libros_codigo\libro_codigo_256.txt), el tamaño del libro de código es de 256 vectores (-tam_libro_codigo 256), los vectores MFCC contienen 30 componentes (-tam_vectores 30), el número de estados por cada fonema sin tomar en cuenta estados dummy es 3 (-num_estados_por_fonema 3), el número máximo de iteraciones para entrenamiento es de 200 (-max_iter_entrenamiento 200). Se indica también el directorio donde se encuentran los archivos de audio (-directorio_wav C:\Users\corpora_tesis\cristian_wav\), se indica el directorio donde se encuentran los modelos (-directorio_modelos C:\Users\modelos_markov_sin_wavelets\). Las líneas como (-nombre_sencillo_matrizA_a matrizA_a.txt) indican el nombre de los archivos donde están almacenadas las matrices de probabilidad de transiciones, de observaciones, y el vector de probabilidad inicial para cada clase de fonema. Finalmente las líneas como (-usar_fonema_a 1) indican si se deben buscar modelos HMM para cada fonema en cuestión; estas opciones son útiles para corpus que no cuentan con todos los fonemas.

La estructura del archivo lista_archivos es la siguiente:

```
-nombre_archivos_wav abrir
-transcripcion       a V r( i r(
-numero_archivos     4

-nombre_archivos_wav adoro
-transcripcion       a D o r( o
-numero_archivos     4

-nombre_archivos_wav afable
-transcripcion       a f a V l e
-numero_archivos     4
```

…y así sucesivamente.

El significado de estos parámetros es: se abrirán archivos cuyo nombre comience con la cadena “abrir”, de esos archivos se abrirán cuatro instancias, abrir1.wav, abrir2.wav, abrir3.wav, abrir4.wav; además la transcripción de esos archivos es la cadena “a V r(i r(“. De manera similar para las palabras adoro y afable.

4.12.6 AlinearPalabrasV2

Este programa abre archivos de audio, obtiene sus correspondientes secuencias de vectores MFCC, carga el modelo HMM de la palabra en cuestión, aplica el algoritmo de Viterbi y obtiene las fronteras fonéticas.
La función es como sigue:

```
AlinearPalabrasV2   (archivo_config, lista_archivos )
```

Donde:

- archivo_config: es un archivo de texto que indica los parámetros para calcular los vectores MFCC, las características del libro de código, la ubicación de los modelos, y si se desea o no aplicar el procesamiento wavelet.
- lista_archivos: es un archivo de texto cuya estructura es igual que el archivo homólogo usado en *EntrenarHMMPalabrasV2*.

La estructura del archivo de configuración es la siguiente:

```
-recortar_silencios 0
-considerar_silencios 1

-tam_ventana_ms     20
-tam_traslape_ms    2
-min_frec_hz        0
-max_frec_hz        8000
-num_ceps           14
-num_bandas         24
-cep_lifter         -22
-usar_preenfasis    1
-coef_preenfasis    0.97
-usar_deltas        1
-num_frames_deltas  2
-usar_accel         0
-num_frames_accel   0
-usar_energia       1

-archivo_libro_codigo C:\Users\libros_codigo\libro_codigo_256.txt
-tam_libro_codigo     256
-tam_vectores         30

-directorio_wav C:\Users\corpora_tesis\cristian_wav\
-directorio_phn C:\Users\cristian_label_auto\
-directorio_modelos C:\Users\modelos_markov_sin_wavelets\

-num_estados_por_fonema_sin_dummies 3

-opcion_proceso_wavelet            0

-tam_ventana_envolvente_wavelet_ms 10
-num_coefs_filtroMM_wavelet        30
-ctte_alfa_wavelet                 0.05
-ctte_vecindario_wavelet_ms        10
-tolerancia_matrices_endpoints_ms  20
```

El significado de estos parámetros es: no recortar silencios utilizando el algoritmo de Rabiner-Sambur (-recortar_silencios 0), agregar silencios automáticamente a las transcripciones (-considerar_silencios 1), el tamaño de los marcos para calcular los vectores MFCC es de 20 ms (-tam_ventana_ms 20), la tasa de los marcos es de 2 ms (-tam_traslape_ms 2), la frecuencia mínima a tomar en cuenta para calcular los vectores MFCC es de 0 Hz (-min_frec_hz 0), la frecuencia máxima a tomar en cuenta para calcular los vectores MFCC es de 8000 Hz (-max_frec_hz 8000), el número de coeficientes es 14 (-num_ceps 14), el número de bandas en los filtros es de 24 (-num_bandas 24), el coeficiente de liftrado es de -22 (-cep_lifter -22), utilizar preénfasis (-usar_preenfasis 1), el coeficiente de preénfasis es de 0.97 (-coef_preenfasis 0.97), calcular los coeficientes delta de los vectores MFCC (-usar_deltas 1), tomar en cuenta dos marcos hacia adelante y hacia atrás para calcular los coeficientes delta (-num_frames_deltas 2), no utilizar los coeficientes de aceleración de los vectores MFCC (-usar_accel 0), calcular la energía de cada marco (-usar_energia 1). También se indica cuál archivo es el libro de código (-archivo_libro_codigo C:\Users\libros_codigo\libro_codigo_256.txt), el tamaño del libro de código es de 256 vectores (-tam_libro_codigo 256), los vectores MFCC contienen 30 componentes (-tam_vectores 30), el directorio donde se encuentran los archivos de audio (-directorio_wav C:\Users\corpora_tesis\cristian_wav\), el directorio donde se escribirán los archivos de etiquetas automáticas (-directorio_phn C:\Users\cristian_label_auto\), el directorio donde se encuentran los modelos HMM (-directorios_modelos C:\Users\modelos_markov_sin_wavelets\), el número de estados por fonema sin contar los estados dummy (-num_estados_por_fonema_sin_dummies 3), no usar el procesamiento wavelet (-opcion_proceso_wavelet 0).
Si se decidiera utilizar el procesamiento wavelet entonces: la ventana deslizante para obtener la envolvente tiene tamaño de 10 ms (-tam_ventana_envolvente_wavelet_ms 10), el número de coeficientes del filtro de media móvil para suavizar los coeficientes wavelet es de 30 (-num_coefs_filtroMM_wavelet 30), el coeficiente de proporcionalidad para calcular el umbral será de 0.05 (-ctta_alfa_wavelet 0.05), el tamaño del vecindario para calcular el umbral será de 10 ms (-ctte_vecindario_wavelet_ms 10).
Finalmente si se va a hacer uso del procesamiento wavelet, el parámetro –tolerancia_matrices_endpoinst_ms indica cuáles fronteras acústicas tomar; es decir, dada una frontera encontrada por alineación forzada, si se encuentra alguna frontera acústica dentro de esta tolerancia, entonces tomarla, si no se encuentra, hay que dejar la frontera propuesta por alineación forzada.

4.12.7 CompararArchivosEtiquetas

Este programa permite realizar estadísticas que indican qué tanta concordancia existe entre las etiquetas generadas manualmente y las etiquetas generadas por el sistema. Las estadísticas generadas por este programa toman en cuenta a todos los tipos de fronteras; es decir, no hay distinción si se trata de una frontera entre una /a/ y una /e/ u otro cualquier tipo de frontera.
Esta función abre uno a uno cada archivo de etiquetas automáticas, busca el archivo de etiquetas manuales equivalente y calcula lo siguiente:

- Error de segmentación. Se aplica la siguiente ecuación:

$$error = \sqrt{\frac{\sum_{k=1}^{N}(frontera\_manual_k - frontera\_manual_k)^2}{N}} \qquad (4.10)$$

- Porcentaje de fronteras correctamente colocadas dentro de 10 ms.
- Porcentaje de fronteras correctamente colocadas dentro de 20 ms.
- Porcentaje de fronteras correctamente colocadas dentro de 30 ms.
- Porcentaje de fronteras correctamente colocadas dentro de 40 ms.
- Porcentaje de fronteras correctamente colocadas dentro de 50 ms.

La función es como sigue:

```
[error_ms,
porcentaje_10ms,
porcentaje_20ms,
porcentaje_30ms,
        =CompararArchivosEtiquetas (directorio_manual,
                                    directorio_automatico)
```

```
porcentaje_40ms,
porcentaje_50ms]
```

Donde:

- directorio_manual: es el directorio donde se encuentran los archivos de etiquetas elaborados manualmente.
- directorio_automatico: es el directorio donde se encuentran los archivos de etiquetas elaborados por la herramienta.
- error_ms: Es el error de segmentación (expresado en milisegundos).
- porcentaje_10ms: es el porcentaje de fronteras correctamente colocadas dentro de 10 ms.
- porcentaje_20ms: es el porcentaje de fronteras correctamente colocadas dentro de 20 ms.
- porcentaje_30ms: es el porcentaje de fronteras correctamente colocadas dentro de 30 ms.
- porcentaje_40ms: es el porcentaje de fronteras correctamente colocadas dentro de 40 ms.
- porcentaje_50ms: es el porcentaje de fronteras correctamente colocadas dentro de 50 ms.

Algo importante es que este programa funciona sólo para archivos de etiquetas en el formato de CSLU; es decir, con una cabecera, y luego las etiquetas. Los tiempos de inicio y fin deben estar expresados en unidades de milisegundos.

4.12.8 CompararArchivosEtiquetasV2

Este programa permite realizar estadísticas que indican qué tanta concordancia existe entre las etiquetas generadas manualmente y las etiquetas generadas por el sistema. A diferencia del programa anterior, este programa realiza las estadísticas para cada tipo de frontera por separado. Dado que en este trabajo se utilizaron 29 clases, entonces habrá $29^2 = 841$ tipos de fronteras distintas; muchas de las cuales serán imposibles de que se presenten.

Esta función abre uno a uno cada archivo de etiquetas automáticas, busca el archivo de etiquetas manuales equivalente y calcula las mismas estadísticas que el programa anterior, con la diferencia de que esta vez; por ejemplo, en lugar de tener un solo valor de error de segmentación, se tienen 841 valores (colocados como una matriz de 29 x 29) de error de segmentación. Las fronteras que no se presenten simplemente tendrán un error de cero. Lo mismo pasa para las otras mediciones, el porcentaje de fronteras correctamente colocadas dentro de 10 ms será una matriz de 29 x 29, donde cada componente de la matriz indica el porcentaje buscado para un tipo de frontera en particular.

La función es la siguiente:

```
[error_ms,                    =CompararArchivosEtiquetasV2 (directorio_manual,
porcentaje_10ms,                                           directorio_automatico
porcentaje_20ms,                                           )
porcentaje_30ms,
porcentaje_40ms,
porcentaje_50ms,
matriz_error_ms,matriz_
porcentaje_10ms,
matriz_porcentaje_20ms,
matriz_porcentaje_30ms,
matriz_porcentaje_40ms,
matriz_porcentaje_50ms]
```

Donde cada argumento de entrada y de salida tiene el mismo significado que en el programa anterior; con la diferencia de que los argumentos de salida son matrices.

Las clases de fonemas se enumeraron en el siguiente orden:

1 /a/	11 /n/	21 /g/
2 /e/	12 /w/	22 /V/
3 /i/	13 /n~/	23 /D/
4 /o/	14 /tS/	24 /G/
5 /u/	15 /dZ/	25 /f/
6 /l/	16 /p/	26 /s/
7 /r(/	17 /t/	27 /x/
8 /r/	18 /k/	28 sil
9 /j/	19 /b/	29 pau
10 /m/	20 /d/	

El significado del componente (i,j) de cada matriz se determina tomando en cuenta la enumeración de las clases. Por ejemplo, error_ms(1,2) indica el error de segmentación del tipo de frontera que existe cuando a la izquierda está el fonema tipo 1 y a la derecha está el fonema tipo 2; esto es, se trata del tipo de frontera entre los fonemas /a/ y /e/ en ese orden. Si por el contrario nos interesa el tipo de frontera donde ocurre primero la /e/ y luego la /a/ entonces buscamos la componente error_ms(2,1).

Este programa simplemente crea las matrices de estadísticas, aunque es difícil saber viendo directamente las matrices cuáles fronteras son las que más fallan. Para hacer un ordenamiento de mayor a menor valor, se creó el siguiente programa *GuardarAnalisisMatrizFronteras*.

4.12.9 GuardarAnalisisMatrizFronteras

Este programa toma las matrices generadas por CompararArchivosEtiquetasV2, ordena de mayor a menor los valores de las matrices e indica también el tipo de frontera asociado con cada valor. Estos resultados se guardan a un archivo. La función es como sigue:

```
GuardarAnalisisMatrizFronteras  (nombre_archivo_destino,
                                 matriz_in)
```

Donde:

- nombre_archivo_destino: es el nombre del archivo de texto donde se guardarán los valores de la matriz ordenados de mayor a menor.
- matriz_in: es alguna matriz generada por *CompararArchivosEtiquetasV2*, de la cual se desea conocer cuáles tipos de fronteras reportan los valores más altos y qué valores reportan.

4.12.10 phn2lab

Este programa se realizó para convertir archivos de etiquetas en formato CSLU a formato HTK. El formato CSLU utiliza como unidades de tiempo el milisegundo, mientras que el formato de HTK se basa en unidades de 100 nanosegundos. Además, el formato de HTK no utiliza ninguna cabecera.
La función es la siguiente:

```
phn2lab    ( directorio_phn, directorio_lab, redondear )
```

Donde:

- directorio_phn: es el directorio donde se encuentran los archivos de etiquetas en formato CSLU que se desean convertir.
- drectorio_lab: es el directorio donde se depositarán los archivos en formato HTK equivalentes.

- redondear: si este parámetro vale 1 entonces se redondeará a la unidad de 100 ns más cercana, de otra forma se incluirán fracciones de unidades de 100 nanosegundos.

4.12.11 CompararArchivosEtiqAcusticas

Este programa evalúa la calidad de una segmentación acústica automática comparándola contra una segmentación fonética manual. Para realizar dicha comparación es necesario tomar en cuenta inserciones y borrados, ya que la segmentación acústica por lo general no tiene la misma cantidad de segmentos que una segmentación fonética; esto es así porque los fonemas no son acústicamente uniformes. Esta función realiza tanto estadísticas sin tomar en cuenta el tipo de fronteras así como también tomándolas en cuenta. La función es la siguiente:

```
[                          =CompararArchivosEtiqAcusticas  (directorio_manual,
porcentaje_inserciones,                                    directorio_automatico
porcentaje_borrados,                                       )
porcentaje_10ms,
porcentaje_20ms,
porcentaje_30ms,
porcentaje_40ms,
porcentaje_50ms,
matriz_porcentaje_borrados
,
matriz_error_ms,
matriz_porcentaje_10ms,
matriz_porcentaje_20ms,
matriz_porcentaje_30ms,
matriz_porcentaje_40ms,
matriz_porcentaje_50ms
]
```

Donde:

- porcentaje_inserciones: es la razón expresada en porcentaje del número total de inserciones respecto al número total de fronteras manuales.
- porcentaje_borrados: es la razón expresada en porcentaje del número total de borrados respecto al número total de fronteras manuales.
- porcentaje_10ms: similar a su valor homólogo en la sección 4.12.7.
- porcentaje_20ms: similar a su valor homólogo en la sección 4.12.7.
- porcentaje_30ms: similar a su valor homólogo en la sección 4.12.7.
- porcentaje_40ms: similar a su valor homólogo en la sección 4.12.7.
- porcentaje_50ms: similar a su valor homólogo en la sección 4.12.7.
- matriz_porcentaje_borrados: cada entrada (i,j) de esta matriz representa el porcentaje de borrados en específico para el tipo de frontera donde el fonema izquierdo es el i-ésimo y el fonema derecho es el j-ésimo (ver 4.12.8).
- matriz_error_ms: similar a su homólogo de la sección 4.12.8.
- matriz_porcentaje_10ms: similar a su valor homólogo de la sección 4.12.8.
- matriz_porcentaje_20ms: similar a su valor homólogo en la sección 4.12.8.
- matriz_porcentaje_30ms: similar a su valor homólogo en la sección 4.12.8.
- matriz_porcentaje_40ms: similar a su valor homólogo en la sección 4.12.8.
- matriz_porcentaje_50ms: similar a su valor homólogo en la sección 4.12.8.

4.13 Evaluación de los archivos de etiquetas

La evaluación de los archivos de etiquetas se realiza de dos formas: la primera forma es por comparación directa (concordancia o *agreement* en inglés) de los archivos de etiquetas generados por el sistema contra los archivos de etiquetas generados de forma manual; para ello se hace uso de los programas que se crearon en Matlab como *CompararArchivosEtiquetas*, *CompararArchivosEtiquetasV2* y *CompararArchivosEtiqAcusticas*. La segunda forma de evaluación es por reconocimiento. Se usa el mismo conjunto de archivos de audio y se entrena un reconocedor en HTK utilizando primero las etiquetas manuales, y después las etiquetas generadas por el sistema. La diferencia en el porcentaje de reconocimiento se deberá a la diferencia de calidad en los archivos de etiquetas.

4.14 Resumen

La metodología de este trabajo consiste en construir un sistema de línea base (solamente alineación forzada) que luego se compara con el sistema propuesto (alineación forzada y segmentación acústica) y también se compara contra el etiquetado manual.

Los parámetros de voz que se escogieron son los MFCC, con posibilidad a utilizar los coeficientes delta y los coeficientes de aceleración. Todas las clases de fonemas son modelados mediante HMM de tres estados, de izquierda a derecha, con posibilidad de saltar desde el primer al tercer estado. Cuando los modelos se concatenan se agregan dos estados *dummy* a cada modelo, es decir, cada fonema se vuelve de 5 estados (sus tres estados originales y dos estados *dummy*).

Los programas trabajan con el alfabeto fonético worldbet, y se utilizan 29 clases: 22 fonemas, 5 alófonos, pausa y silencio.

La evaluación de la calidad de los archivos de etiquetas se realiza mediante dos formas distintas: la primera es por comparación directa con las etiquetas generadas manualmente (concordancia); la segunda es mediante reconocimiento, entrenando reconocedores primero con los archivos de etiquetas manuales y luego con los archivos de etiquetas automáticos.

CAPÍTULO 5. PRESENTACIÓN Y DISCUSIÓN DE RESULTADOS

5.1 Introducción

En primer lugar se presentan resultados para el algoritmo de Galka-Ziolko modificado y la forma de evaluación de la segmentación acústica.

En segundo lugar se presentan los resultados de experimentos sobre un corpus con todos los fonemas hablados en México. En dichos experimentos se mide qué tanto se parecen las etiquetas automáticas con respecto a las etiquetas manuales; así como también se prueba el desempeño de un reconocedor de habla en HTK entrenado con diferentes archivos de etiquetas. También se presenta un estudio de qué tipos de fronteras son los que se detectan más fácilmente mediante segmentación acústica.

En tercer y último lugar se presentan los resultados de experimentos sobre un corpus de dígitos. Dicho corpus no contiene todos los fonemas hablados en México, pero contiene grabaciones de cuatro hablantes, así como 80 repeticiones de cada dígito para los dígitos del 0 al 9. En estos experimentos también se mide el parecido de las etiquetas automáticas y las etiquetas manuales y se prueba el desempeño de un reconocedor de habla en HTK entrenado con diferentes archivos de etiquetas.

En este trabajo se utilizó principalmente la medición de concordancia dentro de 10, 20, 30, 40 y 50 ms, así como también el error de segmentación cuadrático medio. Esto es así porque la alineación forzada siempre entrega la cantidad de segmentos correctos y un análisis de inserciones y borrados carece de sentido. En los experimentos con segmentación acústica, donde el número de segmentos difícilmente será igual al número de segmentos manuales, se utilizaron el error en el número de segmentos y el error en la posición de los segmentos; ya que lo que se buscaba era minimizar ambos errores simultáneamente. El haber realizado esos experimentos utilizando todas las medidas que evalúan las inserciones y los borrados hubiera sido impráctico.

5.2 Segmentación acústica mediante wavelets

El segmentador acústico wavelet debe localizar la mayoría de las probables fronteras entre fonemas. El número de segmentos propuestos difícilmente coincidirá con el número de segmentos que realmente existen en la locución del archivo .wav analizado. Cuando el número de segmentos detectados es mayor que el número de segmentos que se marcaron de manera manual se dice que hubo inserciones. Cuando el número de segmentos detectados es menor que el número de segmentos marcados de manera manual se dice que hubo borrados. Si la cantidad de borrados es excesiva nos encontramos en la situación de que casi no se encontraron fronteras. El caso contrario tampoco es bueno.
El algoritmo de Galka-Ziolko depende de dos parámetros (denotados por α y N) cuyo valor, cuasi óptimo se debe de encontrar. Se dice cuasi óptimo, dado que ambos valores llevan el compromiso de proporcionar un etiquetado acústico lo más cercano posible al etiquetado manual propuesto.

5.2.1 Obtención de los parámetros del algoritmo

De acuerdo a Reddy [46], los fonetistas coinciden aproximadamente un 90% de las veces en la posición de las fronteras entre segmentos colocados manualmente cuando se permite un rango de tolerancia de 20 ms. Debido a este fenómeno se escogieron valores de N para que sean equivalentes a 7.5 ms, 10 ms y 15 ms, los cuales implican vecindarios de 15 ms, 20 ms y 30 ms. Hay fonemas muy cortos del orden de 5 ms, nos referimos a los plosivos, por lo cual el segmento de 20 ms, a priori, resulta muy largo para etiquetar los mismos, sin embargo para fonemas sonoros, 20 ms es una buena opción.

Los experimentos realizados para encontrar valores aceptables para los parámetros α y N consistieron básicamente en seleccionar un conjunto de valores para cada parámetro, y luego probar cada combinación de valores sobre un corpus de prueba.

5.2.2 Experimento 1

En este experimento, el suavizado se realizó con un filtro de 30 coeficeintes (ver ecuación 4.3).Las combinaciones de valores que se probaron fueron:

- Para N que implica 7 ms, los valores de α son: 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10, 0.11, 0.12, 0.13 y 0.14.
- Para N que implica 10 ms, los valores de α son: 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10, 0.11, 0.12 y 0.13.
- Para N que implica 15 ms, los valores de α son: 0.04, 0.05, 0.06, 0.07, 0.08 y 0.09.

Valores más pequeños que los presentados simplemente producían demasiada sobresegmentación, y valores más grandes producían subsegmentación.

Se escogió un subconjunto de diez palabras a partir del corpus de 150 palabras. El subconjunto contiene los 22 fonemas hablados en México. Las diez palabras escogidas son: banco, cepillo, dentro, efectividad, enroscar, gama, jugar, leña, pecho y un_yugo.

Las mediciones utilizadas para evaluar la calidad de la segmentación fueron: el error en el número de segmentos ϵ_n (ver ecuación 3.3), el error en la posición de los segmentos ϵ_p', y un error total Total_ϵ, como se describe a continuación:

$$\epsilon_p'(w) = \frac{\epsilon_p(w)}{n_h} \tag{5.1}$$

Donde $\epsilon_p(w)$ está dado por la ecuación 3.4.

$$Total\_\epsilon(w) = \sqrt{\epsilon_n(w)^2 + \epsilon_p(w)'^2} \tag{5.2}$$

Los resultados del experimento se muestran en las tablas 5.1 a 5.9.

Tabla 5.1. Valores de ϵ_n para cuando N corresponde a 7 ms. Incrementar α disminuye este error.

ϵ_n											
N → 7 ms	alfa										
Palabra	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.11	0.12	0.13	0.14
"banco"	36.11	20.56	13.67	9.44	8.22	7.33	5.56	3.67	2.44	2.00	1.22
"cepillo"	46.33	32.44	22.89	16.56	12.44	9.00	6.00	4.22	3.00	2.33	1.22
"dentro"	41.11	23.67	16.33	11.44	8.67	8.00	7.33	5.89	4.78	4.11	2.56
"efectividad"	25.73	15.87	11.73	9.53	8.53	7.27	5.07	3.07	1.80	1.20	0.60
"enroscar"	38.55	23.09	16.55	13.27	11.00	9.36	7.73	5.55	3.91	2.55	1.36
"gama"	49.67	30.50	21.00	17.33	14.67	12.17	10.17	7.00	5.17	3.50	2.67
"jugar"	41.00	24.00	16.57	12.86	11.43	8.29	6.00	4.14	2.71	0.86	0.43
"leña"	47.83	28.00	21.00	18.67	16.67	13.83	11.17	7.00	5.50	4.00	2.33
"pecho"	27.86	18.29	15.14	13.86	12.29	11.14	9.43	6.29	4.57	3.43	1.57
"un_yugo"	36.38	25.00	20.25	17.50	15.00	12.75	10.63	6.88	5.38	3.00	2.00
Promedio	39.06	24.14	17.51	14.05	11.89	9.91	7.91	5.37	3.93	2.70	1.60

Tabla 5.2. Valores de ϵ_p' para cuando N corresponde a 7 ms. Disminuir α disminuye este error.

ϵ_p'											
N → 7 ms	alfa										
Palabra	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.11	0.12	0.13	0.14
"banco"	1.15	1.87	2.64	5.45	6.20	7.38	11.59	16.74	21.61	21.61	22.53
"cepillo"	1.10	1.10	1.10	3.24	3.56	5.11	12.91	23.29	35.20	44.54	44.54
"dentro"	1.94	2.47	2.88	3.33	6.72	6.72	6.72	9.05	9.05	9.17	26.83
"efectividad"	1.73	2.42	4.19	4.36	4.86	6.85	12.35	20.58	24.92	32.29	39.53
"enroscar"	1.74	2.87	3.33	4.50	4.64	6.66	7.84	10.05	15.01	17.06	22.84
"gama"	1.76	2.21	6.78	6.79	8.42	8.65	9.25	17.12	17.12	32.53	32.53
"jugar"	0.84	1.94	2.89	4.64	5.06	7.07	8.06	10.04	12.56	21.71	31.60
"leña"	1.52	2.56	2.56	2.56	2.56	6.94	6.94	15.18	21.40	26.84	48.54
"pecho"	1.00	2.12	2.16	3.71	3.72	4.90	5.35	10.08	10.23	20.11	22.18
"un_yugo"	1.88	1.88	2.30	2.99	3.57	3.66	5.27	7.82	14.10	26.23	26.23
Promedio	1.47	2.14	3.08	4.16	4.93	6.39	8.63	13.99	18.12	25.21	31.74

Tabla 5.3. Valores de Total_ϵ para cuando N corresponde a 7 ms. Algunos valores de α y N producen valores mínimos en el error.

Total_ϵ											
N → 7 ms	alfa										
Palabra	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.13	0.14
"banco"	36.13	20.64	13.92	10.90	10.30	10.40	12.86	17.14	21.75	21.71	22.57
"cepillo"	46.35	32.46	22.92	16.87	12.94	10.35	14.24	23.67	35.33	44.60	44.56
"dentro"	41.16	23.80	16.59	11.92	10.97	10.45	9.95	10.79	10.23	10.05	26.95
"efectividad"	25.79	16.05	12.46	10.48	9.82	9.99	13.35	20.80	24.99	32.31	39.53
"enroscar"	38.58	23.27	16.88	14.01	11.94	11.49	11.01	11.48	15.51	17.25	22.89
"gama"	49.70	30.58	22.07	18.61	16.91	14.93	13.75	18.50	17.88	32.72	32.64
"jugar"	41.01	24.08	16.82	13.67	12.50	10.89	10.04	10.86	12.85	21.72	31.61
"leña"	47.86	28.12	21.16	18.84	16.86	15.48	13.15	16.72	22.09	27.13	48.60
"pecho"	27.88	18.41	15.30	14.35	12.84	12.17	10.84	11.88	11.21	20.40	22.24
"un_yugo"	36.42	25.07	20.38	17.75	15.42	13.27	11.86	10.41	15.09	26.40	26.31
Promedio	39.09	24.25	17.85	14.74	13.05	11.94	12.10	15.23	18.69	25.43	31.79

Tabla 5.4. Valores de ϵ_n para cuando N corresponde a 10 ms. Incrementar α disminuye este error.

ϵ_n										
N → 10 ms	alfa									
Palabra	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.11	0.12	0.13
"banco"	13.33	9.00	8.11	6.78	5.00	3.56	1.78	0.67	0.33	0.22
"cepillo"	24.56	18.00	11.67	7.33	5.00	2.78	1.44	0.44	0.33	0.56
"dentro"	16.44	10.00	7.67	7.00	6.00	4.44	2.78	0.67	0.11	0.56
"efectividad"	12.47	9.93	8.20	6.80	5.07	2.60	1.07	0.53	0.20	0.20
"enroscar"	16.73	12.55	10.82	7.91	6.36	3.73	1.64	0.91	0.00	0.55
"gama"	20.33	16.50	14.67	12.67	9.00	5.33	3.33	2.33	0.50	0.17
"jugar"	16.43	12.00	10.29	7.71	5.57	2.43	1.00	0.29	0.43	0.86
"leña"	22.50	19.33	16.67	12.33	7.00	3.17	2.50	1.50	0.33	0.17
"pecho"	15.14	12.86	11.57	10.57	8.86	4.43	2.71	1.29	0.29	0.57
"un_yugo"	19.50	16.00	14.38	11.38	6.75	2.88	0.75	0.00	0.50	0.75
Promedio	17.74	13.62	11.40	9.05	6.46	3.53	1.90	0.86	0.30	0.46

Tabla 5.5. Valores de ϵ_p' para cuando N corresponde a 10 ms. Disminuir α disminuye este error.

ϵ_p'										
N → 10 ms	alfa									
Palabra	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.11	0.12	0.13
"banco"	3.44	5.81	8.79	7.17	12.24	16.75	19.66	44.82	46.36	87.87
"cepillo"	1.10	3.06	4.92	11.11	16.29	36.71	61.51	65.35	78.38	97.20
"dentro"	3.47	3.62	9.05	9.05	9.05	9.69	18.81	44.63	64.40	129.76
"efectividad"	3.69	4.69	4.86	8.26	11.17	25.41	35.73	47.43	78.08	90.18
"enroscar"	3.75	4.47	4.64	8.15	13.20	17.09	30.95	55.38	104.68	487.45
"gama"	6.79	8.64	8.64	8.65	14.24	17.71	36.12	36.12	60.17	125.43
"jugar"	2.77	4.64	6.49	6.50	10.04	12.11	19.89	61.09	77.07	168.48
"leña"	2.56	2.56	2.56	8.81	10.21	24.95	24.98	88.28	131.13	131.13
"pecho"	5.94	7.56	7.56	7.57	12.46	15.50	31.61	31.61	52.65	109.75
"un_yugo"	2.46	4.12	5.77	5.77	8.93	10.76	17.68	54.30	68.50	149.76
Promedio	3.60	4.92	6.33	8.10	11.78	18.67	29.70	52.90	76.14	157.70

Tabla 5.6. Valores de Total_ϵ para cuando N corresponde a 10 ms. Algunos valores de α y N producen valores mínimos en el error.

Total_ϵ										
N → 10 ms	alfa									
Palabra	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.13
"banco"	13.77	10.71	11.96	9.86	13.22	17.12	19.74	44.83	46.36	87.87
"cepillo"	24.58	18.26	12.66	13.31	17.04	36.81	61.53	65.35	78.38	97.20
"dentro"	16.81	10.64	11.86	11.44	10.86	10.66	19.01	44.63	64.40	129.76
"efectividad"	13.00	10.99	9.53	10.70	12.27	25.54	35.75	47.43	78.08	90.18
"enroscar"	17.14	13.32	11.77	11.36	14.65	17.49	30.99	55.39	104.68	487.45
"gama"	21.44	18.63	17.02	15.34	16.85	18.49	36.28	36.20	60.17	125.43
"jugar"	16.66	12.86	12.16	10.08	11.48	12.35	19.92	61.09	77.07	168.48
"leña"	22.65	19.50	16.86	15.15	12.38	25.15	25.11	88.30	131.13	131.13
"pecho"	16.27	14.92	13.82	13.00	15.29	16.12	31.72	31.63	52.65	109.75
"un_yugo"	19.65	16.52	15.49	12.76	11.19	11.14	17.70	54.30	68.51	149.76
Promedio	18.20	14.63	13.31	12.30	13.52	19.09	29.77	52.92	76.14	157.70

Tabla 5.7. Valores de ϵ_n para cuando N corresponde a 15 ms. Incrementar α disminuye este error.

ϵ_n						
N → 15 ms	alfa					
Palabra	0.04	0.05	0.06	0.07	0.08	0.09
"banco"	7.78	6.78	4.44	2.00	0.78	0.11
"cepillo"	13.33	7.67	3.89	1.67	0.22	0.67
"dentro"	8.44	7.00	5.67	2.89	0.56	0.11
"efectividad"	9.07	6.40	3.07	1.20	0.40	0.27
"enroscar"	11.45	7.91	5.27	1.73	0.18	0.82
"gama"	15.00	11.83	7.67	3.83	1.83	0.17
"jugar"	10.57	7.29	4.29	2.14	0.29	0.71
"leña"	17.00	12.00	5.50	2.33	0.50	0.50
"pecho"	12.43	9.71	5.14	2.43	0.14	0.29
"un_yugo"	14.25	11.25	5.00	0.38	0.63	0.88
Promedio	11.93	8.78	4.99	2.06	0.55	0.45

Tabla 5.8. Valores de ϵ_p' para cuando N corresponde a 15 ms. Disminuir α disminuye este error.

ϵ_p'						
N → 15 ms	alfa					
Palabra	0.04	0.05	0.06	0.07	0.08	0.09
"banco"	8.79	8.79	13.41	19.66	26.12	62.97
"cepillo"	3.07	10.53	23.41	47.06	75.10	102.05
"dentro"	5.67	9.05	9.09	29.95	55.41	62.90
"efectividad"	4.83	6.26	20.75	52.42	70.05	89.22
"enroscar"	4.64	10.57	17.38	47.47	104.22	433.09
"gama"	8.64	12.43	21.63	53.90	59.85	299.31
"jugar"	6.04	6.50	15.51	20.81	71.14	125.82
"leña"	3.79	4.43	28.10	38.85	124.52	154.36
"pecho"	4.21	9.88	19.90	33.49	69.99	74.52
"un_yugo"	3.19	4.47	9.55	45.99	102.54	211.50
Promedio	5.29	8.29	17.87	38.96	75.89	161.57

Tabla 5.9. Valores de Total_ϵ para cuando N corresponde a 15 ms. Algunos valores de α y N producen valores mínimos en el error.

Total_ϵ						
N → 15 ms	alfa					
Palabra	0.04	0.05	0.06	0.07	0.08	0.09
"banco"	11.74	11.10	14.13	19.76	26.13	62.97
"cepillo"	13.68	13.02	23.73	47.09	75.10	102.05
"dentro"	10.17	11.44	10.71	30.09	55.41	62.90
"efectividad"	10.27	8.96	20.98	52.43	70.05	89.22
"enroscar"	12.36	13.21	18.16	47.50	104.22	433.09
"gama"	17.31	17.16	22.95	54.04	59.88	299.31
"jugar"	12.18	9.76	16.09	20.92	71.14	125.82
"leña"	17.42	12.79	28.64	38.92	124.53	154.36
"pecho"	13.12	13.86	20.56	33.58	69.99	74.52
"un_yugo"	14.60	12.10	10.78	45.99	102.54	211.51
Promedio	13.29	12.34	18.67	39.03	75.90	161.58

Los valores escogidos para los parámetros son: $\alpha = 0.1$ y N → 7 ms = 112 muestras.

La selección de valores para los parámetros α y N involucra un compromiso entre el error en el número de segmentos y el error en la posición de las fronteras. En general, si se permite un porcentaje alto de inserciones, el error en la posición de las fronteras disminuirá; esto es, las fronteras serán mejor detectadas, pero pagando el precio de un error mayor en el número de segmentos. Por otro lado, si se disminuye el error en el número de segmentos aumentará el error en la posición de las fronteras, lo cual significa que las fronteras entre fonemas no serán bien detectadas. En estos experimentos se buscó disminuir al mismo tiempo ambos tipos de errores.

5.2.3 Experimento 2

Para el segundo experimento se agregó una restricción a la función de umbral (ecuación 4.2). La restricción consiste en que el evento seleccionado en cada vecindario debe ser el de mayor magnitud en dicho vecindario. Además también se probó con un suavizado con menos coeficientes (ver ecuación 4.3). Si el evento de mayor magnitud supera al valor del umbral en el vecindario dado, entonces se propone una frontera en ese punto.
Las combinaciones de valores que se probaron fueron:

- Configuración 1: Sin suavizado de coeficientes wavelet, $\alpha = 0$ y N→7.5ms.
- Configuración 2: Sin suavizado de coeficientes wavelet, $\alpha = 0$ y N→10ms.
- Configuración 3: Sin suavizado de coeficientes wavelet, $\alpha = 0$ y N→15 ms.
- Configuración 4: Con suavizado de coeficientes wavelet, filtro de 5 coeficientes, $\alpha = 0$ y N→7.5ms.
- Configuración 5: Con suavizado de coeficientes wavelet, filtro de 5 coeficientes, $\alpha = 0$ y N→10ms.
- Configuración 6: Con suavizado de coeficientes wavelet, filtro de 5 coeficientes, $\alpha = 0$ y N→15ms.
- Configuración 7: Con suavizado de coeficientes wavelet, filtro de 5 coeficientes, $\alpha = 0.08$ y N→7.5ms.
- Configuración 8: Con suavizado de coeficientes wavelet, filtro de 5 coeficientes, $\alpha = 0.08$ y N→10ms.
- Configuración 9: Con suavizado de coeficientes wavelet, filtro de 5 coeficientes, $\alpha = 0.08$ y N→15ms.
- Configuración 10: Con suavizado de coeficientes wavelet, filtro de 30 coeficientes, $\alpha = 0$ y N→7.5ms.
- Configuración 11: Con suavizado de coeficientes wavelet, filtro de 30 coeficientes, $\alpha = 0$ y N→10ms.
- Configuración 12: Con suavizado de coeficientes wavelet, filtro de 30 coeficientes, $\alpha = 0$ y N→15ms.
- Configuración 13: Con suavizado de coeficientes wavelet, filtro de 30 coeficientes, $\alpha = 0.08$ y N→7.5 ms.
- Configuración 14: Con suavizado de coeficientes wavelet, filtro de 30 coeficientes, $\alpha = 0.08$ y N→10ms.
- Configuración 15: Con suavizado de coeficientes wavelet, filtro de 30 coeficientes, $\alpha = 0.08$ y N→15ms.
- Configuración 16: Con suavizado de coeficientes wavelet, filtro de 30 coeficientes, $\alpha = 0.1$ y N→7ms (configuración de la sección 5.2.2).

Para cada una de las combinaciones anteriormente mencionadas se segmentó acústicamente todo el corpus de 150 tipos de palabras, generando archivos de etiquetas en los cuales no conocemos la identidad de ningún fonema, sólo nos interesa la colocación de fronteras. Después se compararon los archivos de etiquetas generados acústicamente con los archivos de etiquetas manuales.

Se compararon los resultados de porcentaje de inserciones, porcentaje de borrados y porcentaje de fronteras correctamente colocadas dentro de 10 ms, como se muestra en la tabla 5.10.

Tabla 5.10. Comparación de las segmentaciones acústicas.

Confi-gura-ción	%inser-ciones	%borra-dos	%10 ms	λ	μ	ϵ_n	ϵ_p	ϵ	error seg
Conf. 1	780.0634	0.7658	82.572	0.8714	0.0061	6.7337	21199.9896	21233.658	6.9342
Conf. 2	570.2667	1.6636	71.719	0.8334	0.0136	4.9227	28127.3434	28151.957	9.3204
Conf. 3	352.6274	4.5154	54.6607	0.7621	0.0380	3.0439	44205.8127	44221.032	14.7250
Conf. 4	1023.1	0.3697	91.8933	0.8985	0.0029	8.8317	16561.4997	16605.658	5.3410
Conf. 5	759.5458	0.8186	83.1793	0.8686	0.0072	6.5566	21244.7744	21277.557	6.9493
Conf. 6	525.7724	2.4558	70.1083	0.8233	0.0216	4.5386	29713.8218	29736.515	9.9723
Conf. 7	999.3927	0.4225	91.2596	0.8965	0.0036	8.6270	16878.8183	16921.953	5.4451
Conf. 8	729.522	1.2411	81.8854	0.8644	0.0107	6.2974	21973.3889	22004.876	7.1040
Conf. 9	440.1901	3.6969	62.1864	0.7985	0.0330	3.7998	37161.5926	37180.591	12.4332
Conf.10	1289	0.7394	84.8165	0.9180	0.0061	11.1269	21500.0915	21555.726	6.6743
Conf.11	877.6868	2.1125	68.5503	0.8853	0.0177	7.5764	30099.7602	30137.642	9.1942
Conf.12	513.5992	4.7531	48.3496	0.8231	0.0392	4.4335	44801.7556	44823.923	13.5750
Conf.13	949.2474	1.6636	72.1151	0.8927	0.0143	8.1942	28298.5205	28339.491	10.4285
Conf.14	585.8991	3.644	53.7893	0.84	0.0314	5.0576	41844.5875	41869.875	13.0275
Conf.15	226.7494	16.0549	29.9974	0.7085	0.1381	1.9573	86248.2953	86258.082	28.5342
Conf.16	587.8531	2.4557	69.3954	0.8387	0.0209	5.0745	34518.0010	34543.373	11.8494

Se escogió la combinación de usar un filtro de 5 coeficientes, $\alpha = 0.08$ y $N \rightarrow 7.5$ ms porque a pesar de tener mucha sobresegmentación tiene un bajo porcentaje de borrados y un alto porcentaje de fronteras correctamente colocadas dentro de 10 ms. También se pudo haber escogido la configuración 4, pero con una mayor sobresegmentación. Como ya se ha comentado, una segmentación acústica difícilmente producirá el mismo número de segmentos que el número de fonemas presentes en la locución. En general, es preferible que la segmentación acústica produzca inserciones (sobresegmentación) que borrados (subsegmentación), ya que si hay borrados perderemos también las fronteras entre fonemas.

5.2.4 Experimento 3

En este experimento se comparó el algoritmo original de segmentación acústica de Galka-Ziolko con el algoritmo de segmentación acústica propuesto en este trabajo.

Para comparar el algoritmo de segmentación acústica original propuesto por Galka y Ziolko, con el algoritmo modificado en este trabajo, se realizaron experimentos de segmentación acústica sobre el corpus de 150 tipos de palabras y 4 repeticiones, con el objetivo de maximizar el porcentaje de fronteras correctamente colocadas dentro de 10 ms y hacer una comparación.

Se realizaron varias pruebas con el algoritmo original, utilizando valores proporcionados en el artículo de Galka y Ziolko. La combinación de valores con la que se obtuvo un mayor porcentaje de fronteras correctamente colocadas a menos de 10 ms respecto a las fronteras manuales fue: $\alpha = 0.5$ $N \rightarrow 2.5$ ms. En este caso, se obtuvo:
Porcentaje de fronteras colocadas a menos de 10 ms respecto a las manuales: 80.1954%.
Porcentaje de inserciones: 550.8053%.
Porcentaje de borrados: 1.3995%.

Luego se utilizó el algoritmo modificado con los $\alpha = 0.08$ $N \rightarrow 7.5$ ms. Para este caso, se obtuvo:
Porcentaje de fronteras colocadas a menos de 10 ms respecto a las manuales: 91.2596%.
Porcentaje de inserciones: 999.3927%.
Porcentaje de borrados: 0.4225%.

En la tabla 5.11 se muestran las medidas de evaluación de las segmentaciones para comparar las configuraciones. La configuración 1 se refiere a la segmentación con el algoritmo original de Galka-Ziolko, la configuración 2 es la propuesta de este trabajo.

Tabla 5.11. Comparación de la segmentación acústica original y la segmentación acústica propuesta.

Configuración	%inserciones	%borrados	%10 ms	λ	μ	ϵ_n	ϵ_p	ϵ	error seg
Conf. 1	550.8053	1.3995	70.0026	0.8047	0.0120	4.7547	36402.6015	36426.375	11.4027
Conf. 2	999.3927	0.4225	91.2596	0.8965	0.0036	8.6270	16878.8183	16921.953	5.4451

El algoritmo propuesto en este trabajo produce un porcentaje de fronteras correctas mayor que el algoritmo original, pero pagando el precio de tener más inserciones. Si lo que se busca es disminuir las inserciones, es mejor utilizar el algoritmo original; sí lo que se busca es aumentar el número de fronteras colocadas correctamente respecto a las fronteras manuales (tal es este caso), es mejor utilizar el algoritmo propuesto.

5.3 Experimentos con el corpus de 150 tipos de palabras

Se realizaron cinco experimentos sobre el corpus que contiene todos los fonemas. El objetivo es comparar la calidad de los archivos de etiquetas generados mediante diferentes configuraciones.

El primer experimento consiste en utilizar todas las etiquetas manuales para entrenar un sistema de reconocimiento de habla en HTK. Todos los archivos del corpus se utilizan tanto en la fase de entrenamiento como en la fase de reconocimiento.

El segundo experimento consiste en generar etiquetas automáticas utilizando el algoritmo de alineación forzada. Los modelos HMM usados por la alineación forzada se entrenaron utilizando todas las etiquetas manuales; la razón de esto es que el corpus tiene sólo cuatro repeticiones por cada palabra, a diferencia del corpus de dígitos, el cual cuenta con veinte repeticiones por cada palabra. Luego de haber generado las etiquetas automáticas, éstas se usan para entrenar un sistema de reconocimiento en HTK. Al igual que en el experimento con etiquetas manuales, todos los archivos del corpus se utilizan tanto en la fase de entrenamiento como en la fase de reconocimiento.

El tercer experimento consiste en segmentar acústicamente todo el corpus, sin utilizar la alineación forzada. El objetivo de este experimento es determinar qué tipos de fronteras son las que se detectan mejor utilizando el procesamiento wavelet descrito en el capítulo 4. En esta ocasión no se entrena ningún sistema de reconocimiento, ya que la segmentación generada en este experimento no es a nivel de fonemas, sino sólo a nivel acústico.

El cuarto experimento consiste en generar etiquetas automáticas para todo el corpus, aplicando tanto alineación forzada como segmentación acústica. En este experimento no se toman en cuenta los tipos de fronteras; únicamente se verifica que la frontera acústica propuesta realmente corresponda con la frontera de alineación forzada, y que además, la distancia entre ambas fronteras sea menor o igual que 20 ms. Después de haber generado los archivos de etiquetas se entrena un sistema de reconocimiento en HTK. Todos los archivos del corpus se utilizan tanto en la fase de entrenamiento como en la de reconocimiento.

El quinto experimento consiste en generar etiquetas automáticas para todo el corpus, aplicando alineación forzada y segmentación acústica, pero esta vez, tomando en cuenta los tipos de fronteras. En este experimento primero se verifica el tipo de frontera en cuestión; si el tipo de frontera no es de los que se comportaron "mejor" en el experimento de segmentación acústica, entonces la frontera acústica propuesta es desechada y se conserva la frontera propuesta por alineación forzada. Por otro lado, si el tipo de frontera sí es de las que se comportaron mejor en el experimento de segmentación acústica, entonces se aplican los siguientes dos

criterios utilizados en el cuarto experimento (que las fronteras realmente se correspondan entre sí, y que la distancia entre las fronteras sea menor o igual a 20 ms).

El corpus utilizado contiene 150 palabras distintas repetidas 4 veces por un hablante (600 archivos) y contiene todos los fonemas hablados en México. La frecuencia de muestreo es de 16 KHz.

5.3.1 Experimento 1. Reconocimiento con etiquetas manuales

Se etiquetó manualmente el corpus. Se entrenó un sistema de reconocimiento en HTK utilizando todos los archivos del corpus. Luego se realizó una prueba de reconocimiento utilizando todos los archivos del corpus.

Dado que el corpus fue etiquetado utilizando la herramienta SpeechViewer del CSLU Toolkit, primero se cambió el formato a los archivos de etiquetas, desechando las cabeceras que agrega SpeechViewer y convirtiendo los puntos de inicio y fin de cada fonema, de milisegundos a unidades de 100 nanosegundos. Para ello se utilizó el programa phn2lab elaborado en Matlab.

Todas las veces que se utilizó HTK para crear y probar los sistemas de reconocimiento, se utilizó la misma metodología. Tanto la metodología como los archivos de configuración y los archivos generados por HTK se muestran en el anexo C; en este capítulo sólo presentaremos los resultados de los experimentos, ya que esto ayudará a hacer este capítulo más legible.

Los vectores de características utilizados consistieron en parámetros MFCC, sus primeras y segundas deltas; en ambas deltas se utilizaron dos marcos hacia adelante y dos marcos hacia atrás. Se utilizaron 14 cepstrales, más la energía. Se usaron 22 canales para los filtros y un coeficiente de liftrado de 22. Las frecuencias utilizadas para los filtros van desde 0 Hz hasta 8000 Hz (la frecuencia de muestreo de los archivos es de 16,000 Hz). El tamaño de cada marco usado fue de 20 ms, y la tasa de marco fue de 1 ms. El coeficiente de preénfasis utilizado fue 0.97.

A continuación se muestra el porcentaje de reconocimiento obtenido.

```
====================== HTK Results Analysis =======================
  Date: Tue Mar 06 15:10:56 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=95.83 [H=575, S=25, N=600]
WORD: %Corr=98.61, Acc=98.61 [H=1775, D=4, S=21, I=0, N=1800]
```

Figura 5.1. Porcentaje de reconocimiento para el sistema entrenado con etiquetas manuales.

Nos interesa el resultado de palabras (WORD), no de oraciones (SENT), por lo cual, el porcentaje de reconocimiento es de 98.61%.

5.3.2 Experimento 2. Generación de etiquetas con alineación forzada

En este experimento se utilizaron todas las etiquetas manuales para entrenar modelos HMM. Se generaron archivos de etiquetas para todo el corpus utilizando el algoritmo de alineación forzada.

En el proceso de alineación forzada se utilizaron parámetros MFCC, catorce cepstrales más la energía, además de sus primeras y sus segundas deltas, lo cual implica un tamaño de vectores de 45. Los marcos usados son de 20 ms, con una tasa de marco de 2 ms. Las frecuencias usadas para calcular los MFCC son desde 0 Hz hasta 8000 Hz. Se utilizaron 24 bandas y un coeficiente de liftrado de -22 (el toolkit rastamat cambia el signo del coeficiente de liftrado). El coeficiente de preénfasis utilizado es de 0.97. Las primeras y segundas deltas se calcularon tomando en cuenta dos marcos hacia adelante y dos marcos hacia atrás. La configuración usada para el programa *CalcularVectoresPorFonema* fue como sigue:

```
-tam_ventana_ms 20
-tam_traslape_ms 2
-min_frec_hz  0
-max_frec_hz  8000
-num_ceps     14
-num_bandas   24
-cep_lifter -22
-usar_preenfasis 1
-coef_preenfasis 0.97
-usar_deltas   1
-num_frames_deltas 2
-usar_accel  1
-num_frames_accel 2
-usar_energia 1
```

Se creó un libro de código de 256 vectores a partir de los vectores de entrenamiento.

Luego se procedió a entrenar los modelos HMM para cada fonema. Después se concatenaron los modelos HMM para formar modelos a nivel de palabra, y éstos fueron reentrenados utilizando las palabras del corpus. Finalmente se utilizó el algoritmo de alineación forzada, aplicando el algoritmo de Viterbi sobre cada palabra del corpus utilizando los modelos HMM a nivel de palabras reentrenados, compuestos por diversos HMM de nivel de fonemas.

Se generaron archivos de etiquetas para todos los archivos del corpus (600 archivos). La concordancia entre las etiquetas de alineación forzada y las etiquetas manuales se muestran en la tabla 5.12.

Tabla 5.12. Etiquetas generadas por alineación forzada comparadas contra las etiquetas manuales.

Concepto	Valor
Error medio cuadrático (ms)	16.2902
Porcentaje de fronteras correctas dentro de 10 ms	61.0259
Porcentaje de fronteras correctas dentro de 20 ms	89.0798
Porcentaje de fronteras correctas dentro de 30 ms	95.0555
Porcentaje de fronteras correctas dentro de 40 ms	97.3823
Porcentaje de fronteras correctas dentro de 50 ms	98.4135

Después se entrenó un sistema de reconocimiento en HTK. Tanto la fase de entrenamiento como de reconocimiento utilizaron todos los archivos del corpus. La metodología para usar HTK es la misma que la utilizada en el experimento 1 (sección 5.3.1), por lo cual no se volverá a explicar. Para ver una descripción más detallada vea el anexo C. Los resultados del reconocimiento se muestran en la figura 5.2.

```
====================== HTK Results Analysis =======================
  Date: Tue Mar 06 22:31:16 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=95.67 [H=574, S=26, N=600]
WORD: %Corr=98.50, Acc=98.50 [H=1773, D=8, S=19, I=0, N=1800]
------------------------ Confusion Matrix -------------------------
```

Figura 5.2. Porcentaje de reconocimiento para un sistema entrenado con etiquetas elaboradas con alineación forzada.

Al igual que en el experimento1, nos interesa el resultado de palabras (WORD), no de oraciones (SENT), por lo cual, el porcentaje de reconocimiento es de 98.50%.

5.3.3. Experimento 3. Fronteras mejor detectadas mediante segmentación acústica

En este experimento se segmentó acústicamente todo el corpus para determinar el tipo de fronteras que son más fácilmente detectables por medio de segmentación acústica. En este experimento no se utilizó la alineación forzada.

La configuración del segmentador acústico wavelet fue la siguiente: ventana deslizante de 10 ms para detección de envolvente, filtro de suavizado de 5 coeficientes, parámetro de proporcionalidad α=0.08, y parámetro de vecindario N => 7.5ms, que implica un vecindario de 15 ms.

Se midió la concordancia de las fronteras acústicas con respecto a las fronteras manuales, y se escogieron aquellos tipos de fronteras que hayan sido correctamente colocadas dentro de 10 ms respecto de las fronteras manuales al menos un 70% de las veces. Dichas fronteras se muestran en la tabla 5.13.

Tabla 5.13. Tipos de fronteras donde las propuestas acústicas están a menos de 10 ms respecto las fronteras manuales al menos el 70% de las veces acomodadas en orden alfabético.

Fonema izquierdo	Fonema derecho	Porcentaje 10 ms	Fonema izquierdo	Fonema derecho	Porcentaje 10 ms	Fonema izquierdo	Fonema derecho	Porcentaje
a	D	92.500000	j	o	88.888889	r(	sil	78.571429
a	e	100.000000	k	i	75.000000	r(	V	100.000000
a	f	95.000000	k	o	90.384615	s	a	83.333333
a	G	91.666667	k	pau	87.500000	s	e	100.000000
a	j	93.750000	k	s	75.000000	s	j	85.000000
a	l	100.000000	k	w	100.000000	s	l	100.000000
a	m	100.000000	l	a	85.000000	s	m	87.500000
a	n	97.222222	l	d	100.000000	s	o	91.666667
a	pau	93.750000	l	dZ	87.500000	s	r	75.000000
a	s	100.000000	l	e	83.333333	s	sil	89.743590
a	sil	78.409091	l	f	100.000000	sil	a	85.000000
a	u	100.000000	l	i	100.000000	sil	b	85.714286
a	V	80.555556	l	pau	70.000000	sil	d	78.125000
a	x	100.000000	l	s	100.000000	sil	g	91.666667
b	a	100.000000	m	a	97.727273	sil	i	83.333333
b	i	100.000000	m	b	87.500000	sil	k	88.095238
b	j	100.000000	m	e	87.500000	sil	l	100.000000
b	o	87.500000	m	i	75.000000	sil	m	84.375000
D	a	75.000000	m	o	87.500000	sil	n	75.000000
d	a	83.333333	n	a	90.000000	sil	n~	75.000000
D	e	100.000000	n	d	75.000000	sil	o	91.666667
d	e	87.500000	n	dZ	83.333333	sil	p	75.000000
D	j	100.000000	n	f	100.000000	sil	r	100.000000
d	o	100.000000	n	i	100.000000	sil	s	75.000000
D	o	94.444444	n	o	85.000000	sil	t	87.500000
D	sil	87.500000	n	pau	72.500000	sil	tS	81.250000
d	u	100.000000	n	r	75.000000	sil	w	75.000000
dZ	a	100.000000	n	s	100.000000	sil	x	71.428571
dZ	e	100.000000	n	sil	85.000000	t	a	91.666667
e	D	87.500000	n~	a	100.000000	t	e	95.833333
e	f	100.000000	n~	o	75.000000	t	i	100.000000
e	l	95.833333	n~	w	100.000000	t	o	89.583333
e	m	100.000000	o	l	100.000000	t	r(	100.000000
e	n	97.727273	o	m	100.000000	t	u	100.000000
e	n~	100.000000	o	n	97.222222	tS	a	81.250000
e	pau	75.000000	o	n~	75.000000	tS	e	75.000000
e	r	75.000000	o	pau	93.750000	tS	i	95.000000
e	s	93.750000	o	r(	75.000000	tS	o	87.500000
e	sil	81.666667	o	s	93.750000	tS	u	75.000000
e	x	75.000000	o	sil	80.084746	u	G	75.000000

f	a	85.714286	o	V	75.000000	u	l	75.000000
f	e	83.333333	p	a	97.500000	u	n	95.833333
f	i	83.333333	p	e	100.000000	u	pau	100.000000
f	o	87.500000	p	i	75.000000	u	r(	75.000000
f	u	100.000000	p	o	100.000000	u	s	87.500000
g	a	80.000000	p	u	75.000000	u	V	100.000000
g	e	100.000000	pau	k	93.181818	V	a	90.000000
g	i	100.000000	pau	p	95.000000	V	e	75.000000
G	i	75.000000	pau	t	94.565217	V	i	75.000000
g	o	100.000000	pau	tS	94.444444	V	j	75.000000
G	o	91.666667	r	a	95.000000	V	l	100.000000
i	a	100.000000	r	e	100.000000	V	o	100.000000
i	j	87.500000	r	o	100.000000	V	r(	100.000000
i	n	93.750000	r(	a	91.666667	V	u	100.000000
i	n~	75.000000	r(	D	75.000000	w	e	87.500000
i	pau	91.666667	r(	e	100.000000	x	a	100.000000
i	r(	100.000000	r(	i	75.000000	x	i	100.000000
i	s	85.714286	r(	m	100.000000	x	o	87.500000
i	V	100.000000	r(	o	83.333333	x	u	75.000000
j	a	100.000000	r(	pau	91.666667			
j	e	87.500000	r(	s	100.000000			

En este experimento no se realizó reconocimiento en HTK, ya que esta segmentación no es a nivel de fonema, sino a nivel acústico.

5.3.4. Experimento 4. Generación de etiquetas con alineación forzada y segmentación acústica

En este experimento se retomaron los modelos HMM generados en el experimento con alineación forzada y se crearon archivos de etiquetas para todo el corpus utilizando alineación forzada y segmentación acústica combinadas.

La alineación forzada genera fronteras probables entre fonemas, la segmentación acústica genera fronteras en los lugares donde se presente un cambio acústico (evento).

Para cada frontera generada por alineación forzada se busca la frontera acústica más cercana. Luego se verifica que dicha frontera acústica realmente corresponda a la frontera actual de alineación forzada. Si la frontera acústica realmente corresponde a otra frontera de alineación forzada, entonces se desecha la frontera acústica y nos quedamos con la frontera de alineación forzada. Si la frontera acústica sí se corresponde con la frontera de alineación forzada actual entonces se verifica si la distancia existente entre dichas fronteras es menor o igual que 20 ms. En el caso en que se cumplan las dos condiciones (que las fronteras se correspondan y que la distancia entre las dos fronteras sea menor o igual a 20 ms) la frontera de alineación forzada se desecha y nos quedamos con la frontera acústica. Si alguna de las dos condiciones no se cumple, se desecha la frontera acústica y nos quedamos con la frontera de alineación forzada. Se etiquetó todo el corpus. La concordancia de las etiquetas automáticas con respecto a las etiquetas manuales se muestra en la tabla 5.14.

Tabla 5.14. Etiquetas generadas por alineación forzada y segmentación acústica, comparadas contra las etiquetas manuales.

Concepto	Valor
Error medio cuadrático (ms)	17.9711
Porcentaje de fronteras correctas dentro de 10 ms	54.2305
Porcentaje de fronteras correctas dentro de 20 ms	83.6065
Porcentaje de fronteras correctas dentro de 30 ms	93.5219
Porcentaje de fronteras correctas dentro de 40 ms	97.0650
Porcentaje de fronteras correctas dentro de 50 ms	98.2813

Al igual que en el experimento 2 (sección 5.3.2) se entrenó y probó un sistema de reconocimiento en HTK. Tanto la fase de entrenamiento como la de reconocimiento utilizaron todos los archivos del corpus. Dado que la metodología es igual, no se volverá a explicar.

Los resultados del porcentaje de reconocimiento se muestran en la figura 5.3.

```
====================== HTK Results Analysis =======================
  Date: Wed Mar 07 01:43:25 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=95.00 [H=570, S=30, N=600]
WORD: %Corr=98.33, Acc=98.33 [H=1770, D=5, S=25, I=0, N=1800]
```

Figura 5.3. Porcentaje de reconocimiento para un sistema entrenado con etiquetas elaboradas con alineación forzada y segmentación acústica.

Como se puede observar, el porcentaje de reconocimiento es de 98.33%.

5.3.5. Experimento 5. Generación de etiquetas con alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras

Al igual que en el experimento 4 (sección 5.3.4), en este experimento se crearon archivos de etiquetas para todo el corpus con alineación forzada y con segmentación acústica; sin embargo en este experimento se tomaron en cuenta los tipos de fronteras que son detectadas de mejor manera por la segmentación acústica (ver sección 5.3.3).

En esta ocasión, para cada frontera se analiza qué tipo de frontera es. Si el tipo de frontera está contenida en la tabla 5.13 entonces se realiza el mismo procedimiento que en el experimento anterior: se encuentra la frontera acústica más cercana, se verifica que estas fronteras realmente se correspondan, y se verifica que la distancia entre ellas sea menor o igual que 20 ms. Sin embargo, en el caso de que el tipo de frontera no esté contenido en la tabla 5.13 entonces no se busca ninguna frontera acústica, y nos quedamos con la frontera propuesta por alineación forzada.

Se etiquetó todo el corpus. La concordancia de las etiquetas automáticas con respecto a las etiquetas manuales se muestra en la tabla 5.15.

Tabla 5.15. Etiquetas generadas por alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras, comparadas contra las etiquetas manuales.

Concepto	Valor
Error medio cuadrático (ms)	17.7177
Porcentaje de fronteras correctas dentro de 10 ms	55.4468
Porcentaje de fronteras correctas dentro de 20 ms	84.4262
Porcentaje de fronteras correctas dentro de 30 ms	93.6541
Porcentaje de fronteras correctas dentro de 40 ms	97.0914
Porcentaje de fronteras correctas dentro de 50 ms	98.4399

Se entrenó y probó un sistema de reconocimiento en HTK con la misma metodología. Los resultados del porcentaje de reconocimiento se muestran en la figura 5.4.

```
====================== HTK Results Analysis =======================
  Date: Wed Mar 07 11:56:50 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=95.33 [H=572, S=28, N=600]
WORD: %Corr=98.39, Acc=98.39 [H=1771, D=8, S=21, I=0, N=1800]
```

Figura 5.4. Porcentaje de reconocimiento para un sistema entrenado con etiquetas elaboradas con alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras.

Como se puede observar, el porcentaje de reconocimiento obtenido fue de 98.39%.

5.4 Experimentos con el corpus de dígitos mezclados

Se realizaron tres experimentos sobre un corpus de dígitos. El objetivo es comparar la calidad de los archivos de etiquetas generados automáticamente. Los archivos de etiquetas se generaron primero únicamente mediante alineación forzada; luego se generó otro conjunto de archivos de etiquetas pero utilizando la combinación de alineación forzada y segmentación acústica. El corpus también fue etiquetado manualmente.

El primer experimento consiste en utilizar las etiquetas manuales para entrenar un sistema de reconocimiento de habla en HTK. Todos los archivos del corpus se utilizan tanto en el entrenamiento como en el reconocimiento.

El segundo experimento consiste en tomar una cuarta parte de los archivos de etiquetas manuales; y a partir de éstos, entrenar modelos HMM para la alineación forzada. Una vez que se han entrenado modelos HMM, se utiliza el algoritmo de alineación forzada para generar etiquetas automáticas para todo el corpus. Luego se entrena un sistema de reconocimiento de habla en HTK. Al igual que en el experimento1, todos los archivos del corpus se utilizan tanto en la fase de entrenamiento, como en la fase de reconocimiento.

El tercer experimento toma los modelos HMM generados en el experimento 2, y con ellos genera archivos de etiquetas automáticas para todo el corpus; pero esta vez se utiliza alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras. Al igual que en el experimento 2, se entrena un sistema de reconocimiento en HTK y luego se prueba el porcentaje de reconocimiento. Todos los archivos del corpus se utilizan en el entrenamiento y en el reconocimiento.

El corpus de dígitos consiste en las palabras cero, *uno, dos, tres, cuatro, cinco, seis, siete, ocho y nueve*. Cada palabra fue repetida veinte veces por cuatro hablantes, lo cual nos da un total de 80 repeticiones para cada palabra (exceptuando las palabras *cuatro* y *siete*, de las cuales se cuenta sólo con 79 repeticiones). El corpus cuenta en total con 798 archivos de audio. Los archivos del corpus fueron grabados con una frecuencia de muestreo de 11025 Hz.

5.4.1 Experimento 1. Reconocimiento con etiquetas manuales

Se etiquetó manualmente el corpus de dígitos. Este corpus no cuenta con todos los fonemas hablados en México. Las transcripciones de este corpus se muestran en el anexo B.

Se entrenó un sistema de reconocimiento en HTK utilizando todos los archivos del corpus. Luego se realizó una prueba de reconocimiento utilizando todos los archivos del corpus.

Los vectores de características utilizados consistieron en parámetros MFCC, sus primeras y segundas deltas; en ambas deltas se utilizaron dos marcos hacia adelante y dos marcos hacia atrás. Se utilizaron 14 cepstrales, más la energía. Se usaron 22 canales para los filtros y un coeficiente de liftrado de 22. Las frecuencias utilizadas para los filtros van desde 0 Hz hasta 5512 Hz. El tamaño de cada marco usado fue de 20 ms, y la tasa de marco fue de 1 ms. El coeficiente de preénfasis utilizado fue 0.97.

Sólo se utilizó una mixtura en los modelos ya que el corpus es pequeño. Los resultados de reconocimiento se muestran en la figura 5.5:

```
====================== HTK Results Analysis =======================
  Date: Tue Mar 06 10:08:51 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=95.24 [H=760, S=38, N=798]
WORD: %Corr=95.24, Acc=95.24 [H=760, D=0, S=38, I=0, N=798]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e   Del [ %c / %e]
cero  80   0   0   0   0   0   0   0   0   0     0
 uno   0  79   0   0   0   0   0   0   1   0     0 [98.8/0.1]
 dos   0   0  79   0   0   0   0   0   1   0     0 [98.8/0.1]
tres   0   0   0  78   0   0   2   0   0   0     0 [97.5/0.3]
cuat   0   0   0   0  79   0   0   0   0   0     0
cinc   0   0   0   0   0  80   0   0   0   0     0
seis   0   0   0   7   0   6  55  12   0   0     0 [68.8/3.1]
siet   0   0   0   0   0   0   0  79   0   0     0
ocho   0   0   0   0   7   0   0   0  73   0     0 [91.3/0.9]
nuev   0   0   0   0   2   0   0   0   0  78     0 [97.5/0.3]
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.5. Porcentaje de reconocimiento para un sistema entrenado con etiquetas manuales.

5.4.2 Experimento 2. Generación de etiquetas con alineación forzada

En este experimento se entrenaron modelos HMM a partir de un subconjunto de etiquetas manuales. Dicho subconjunto fueron las primeras 20 repeticiones para cada palabra; esto es, se utilizó un cuarto de las etiquetas manuales para entrenar los modelos HMM.

Una vez que se entrenaron los modelos HMM, se utilizó el algoritmo de alineación forzada para generar archivos de etiquetas para todo el corpus (798 archivos). Esto es, a partir de 200 archivos de etiquetas manuales, se generaron 798 archivos de etiquetas automáticas.

La configuración para realizar la alineación forzada fue: utilizar 14 coeficientes (más la energía) de MFCC, además de sus primeras y segundas deltas, tomando en cuenta dos marcos hacia adelante y dos marcos hacia atrás. El tamaño de los marcos fue de 20 ms y la tasa de marcos fue de 2 ms. El rango de frecuencias usado en el cálculo de los MFCC fue desde 0 Hz hasta 5512 Hz. Se utilizaron 24 bandas, y un coeficiente de liftrado de -22 (el toolkit de rastamat le cambia el signo). El coeficiente de preénfasis utilizado fue de 0.97.

La configuración para el programa *CalcularVectoresFonemas* fue la siguiente:

```
-tam_ventana_ms 20
-tam_traslape_ms 2
-min_frec_hz  0
-max_frec_hz  5512
-num_ceps     14
-num_bandas   24
-cep_lifter -22
-usar_preenfasis 1
```

```
-coef_preenfasis 0.97
-usar_deltas   1
-num_frames_deltas 2
-usar_accel  1
-num_frames_accel 2
-usar_energia 1
```

Las etiquetas automáticas se compararon mediante concordancia con las etiquetas manuales. Los resultados se muestran en la tabla 5.16.

Tabla 5.16. Etiquetas generadas por alineación forzada comparadas contra las etiquetas manuales.

Concepto	Valor
Error medio cuadrático (ms)	49.4323
Porcentaje de fronteras correctas dentro de 10 ms	40.4322
Porcentaje de fronteras correctas dentro de 20 ms	66.6782
Porcentaje de fronteras correctas dentro de 30 ms	77.3440
Porcentaje de fronteras correctas dentro de 40 ms	82.9208
Porcentaje de fronteras correctas dentro de 50 ms	86.7201

Después, se entrenó un sistema de reconocimiento en HTK. Tanto la fase de entrenamiento, como la fase de reconocimiento utilizaron todos los archivos del corpus.

Las condiciones para entrenamiento y prueba en HTK fueron las mismas que en el experimento con etiquetas manuales, excepto que esta vez se utilizaron las etiquetas generadas por alineación forzada.

Los resultados del reconocimiento se muestran en la figura 5.6:

```
====================== HTK Results Analysis =======================
  Date: Tue Mar 06 10:36:32 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=96.74 [H=772, S=26, N=798]
WORD: %Corr=96.74, Acc=96.74 [H=772, D=0, S=26, I=0, N=798]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  80   0   0   0   0   0   0   0   0   0    0
 uno   0  79   0   0   0   0   0   0   1   0    0 [98.8/0.1]
 dos   0   0  80   0   0   0   0   0   0   0    0
tres   0   0   0  67   0   0  11   2   0   0    0 [83.8/1.6]
cuat   0   0   0   0  79   0   0   0   0   0    0
cinc   0   0   0   0   0  70   0  10   0   0    0 [87.5/1.3]
seis   0   0   0   0   0   0  80   0   0   0    0
siet   0   0   0   0   0   0   1  78   0   0    0 [98.7/0.1]
ocho   0   0   0   0   0   0   0   0  80   0    0
nuev   0   0   0   0   1   0   0   0   0  79    0 [98.8/0.1]
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.6. Porcentaje de reconocimiento para un sistema entrenado con etiquetas elaboradas por alineación forzada.

5.4.3 Experimento 3. Generación de etiquetas con alineación forzada y segmentación acústica

En este experimento se retomaron los modelos HMM generados en el experimento anterior. Esta vez se generaron archivos de etiquetas para todo el corpus mediante alineación forzada y segmentación acústica. Además, se tomaron en cuenta los tipos de fronteras de la sección 5.3.3.

La configuración para realizar la alineación forzada fue la misma que la utilizada en la sección 5.4.2. La configuración para realizar la segmentación acústica fue la misma que la utilizada en la sección 5.3.5.

Las etiquetas automáticas se compararon mediante concordancia contra las etiquetas manuales. Los resultados se muestran en la tabla 5.17:

Tabla 5.17. Etiquetas generadas por alineación forzada y segmentación acústica comparadas contra las etiquetas manuales.

Concepto	Valor
Error medio cuadrático (ms)	49.8570
Porcentaje de fronteras correctas dentro de 10 ms	37.7831
Porcentaje de fronteras correctas dentro de 20 ms	63.8550
Porcentaje de fronteras correctas dentro de 30 ms	76.2286
Porcentaje de fronteras correctas dentro de 40 ms	82.8163
Porcentaje de fronteras correctas dentro de 50 ms	86.5109

Después, se entrenó un sistema de reconocimiento en HTK. Tanto la fase de entrenamiento, como la fase de reconocimiento utilizaron todos los archivos del corpus.

Las condiciones para entrenamiento y prueba en HTK fueron las mismas que en el experimento con etiquetas manuales, excepto que esta vez se utilizaron las etiquetas generadas por alineación forzada y segmentación acústica.

Los resultados del reconocimiento se muestran en la figura 5.7.

```
====================== HTK Results Analysis =======================
  Date: Tue Mar 06 09:41:49 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=94.36 [H=753, S=45, N=798]
WORD: %Corr=94.36, Acc=94.36 [H=753, D=0, S=45, I=0, N=798]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  80   0   0   0   0   0   0   0   0   0    0
 uno   6  71   0   1   2   0   0   0   0   0    0 [88.8/1.1]
 dos   0   0  76   4   0   0   0   0   0   0    0 [95.0/0.5]
tres   0   0   0  68   0   0  12   0   0   0    0 [85.0/1.5]
cuat   0   0   0   0  79   0   0   0   0   0    0
cinc   0   0   0   0   0  80   0   0   0   0    0
seis   0   0   0   1   0   0  79   0   0   0    0 [98.8/0.1]
siet   0   0   0   0   0   0   0  79   0   0    0
ocho   0   0   0   0  16   0   0   0  64   0    0 [80.0/2.0]
nuev   0   0   0   0   3   0   0   0   0  77    0 [96.3/0.4]
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.7. Porcentaje de reconocimiento para un sistema entrenado con etiquetas elaboradas por alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras.

5.5 Experimentos con cuatro corpus de dígitos de un solo hablante

Se tomaron cuatro corpus de dígitos, cada uno con un solo hablante. Las palabras contenidas son los dígitos desde el cero hasta el nueve, repetidos 20 veces, lo cual produce un total de 200 archivos; exceptuando dos de los corpus, que sólo contienen 199 archivos. Los archivos faltantes se tuvieron que descartar ya que el nivel de voz era demasiado alto y había saturación.

La frecuencia de muestreo de los archivos utilizados es de 11025 Hz. Los corpus fueron etiquetados manualmente y luego fueron etiquetados automáticamente.

Para cada corpus se realizaron cuatro experimentos de reconocimiento como sigue:

- Lo primero fue dividir cada corpus por la mitad. La mitad que corresponde a las primeras repeticiones es la parte baja; la otra es la parte alta.
- Para cada corpus se entrenó el sistema de etiquetado automático utilizando únicamente etiquetas manuales de la parte baja. Luego, cada corpus se segmentó automáticamente, generando los archivos de etiquetas para todo el corpus.
- Para cada corpus se midió la concordancia entre las etiquetas manuales y las automáticas.
- Para cada corpus se entrenaron dos sistemas de reconocimiento en HTK: el primero se entrenó utilizando las etiquetas manuales de la parte baja; el segundo, utilizando las etiquetas automáticas de la parte baja.
- Cada sistema de reconocimiento se probó primero con los archivos wav de la parte baja del corpus, buscando obtener un 100% de reconocimiento.
- Luego, cada sistema de reconocimiento se probó con los archivos wav de la parte alta del corpus, los cuales no participaron en el entrenamiento.

En la figura 5.8 se muestra gráficamente la partición de cada corpus y por qué se hicieron cuatro experimentos de reconocimiento en cada corpus.

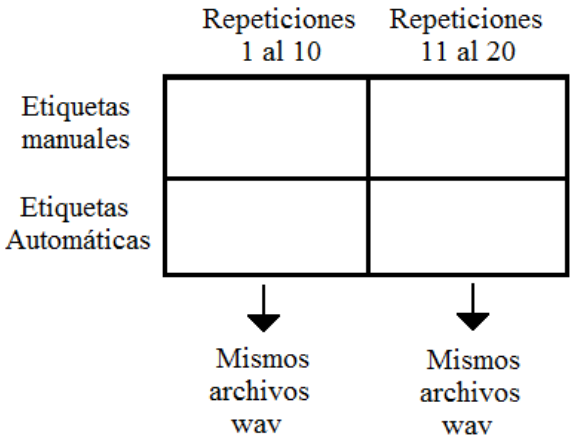


Figura 5.8. Para cada corpus se realizaron cuatro experimentos de reconocimiento.

En la tabla 5.18 se muestra la concordancia entre las etiquetas automáticas y las manuales para cada corpus.

Tabla 5.18. Concordancia del etiquetado automático respecto al etiquetado manual para cada corpus de dígitos.

	Corpus1 francovoz	Corpus2 lisethvoz	Corpus3 rodrigovoz	Corpus4 sergiovoz
Error medio cuadrático (ms)	20.7700	20.2340	24.1340	24.1696
Porcentaje de fronteras correctas dentro de 10 ms	65.6944	78.2303	55.8988	59.1666
Porcentaje de fronteras correctas dentro de 20 ms	88.7500	93.1179	82.2033	84.7222
Porcentaje de fronteras correctas dentro de 30 ms	93.4722	95.5056	91.1516	93.1944
Porcentaje de fronteras correctas dentro de 40 ms	94.5833	95.9269	92.8370	96.9444
Porcentaje de fronteras correctas dentro de 50 ms	95.1388	96.7696	94.8033	97.9166

5.5.1 Corpus1 francovoz. Etiquetas manuales. Parte baja del corpus

Se entrenó y probó un reconocedor en HTK utilizando los archivos de la parte baja del corpus con etiquetas manuales. Los resultados de reconocimiento se muestran en la figura 5.9.

```
====================== HTK Results Analysis =======================
  Date: Sun May 06 21:08:57 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=100, S=0, N=100]
WORD: %Corr=100.00, Acc=100.00 [H=100, D=0, S=0, I=0, N=100]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0    0
 uno   0  10   0   0   0   0   0   0   0   0    0
 dos   0   0  10   0   0   0   0   0   0   0    0
tres   0   0   0  10   0   0   0   0   0   0    0
cuat   0   0   0   0  10   0   0   0   0   0    0
cinc   0   0   0   0   0  10   0   0   0   0    0
seis   0   0   0   0   0   0  10   0   0   0    0
siet   0   0   0   0   0   0   0  10   0   0    0
ocho   0   0   0   0   0   0   0   0  10   0    0
nuev   0   0   0   0   0   0   0   0   0  10    0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.9. Resultados de reconocimiento para el corpus francovoz con etiquetas manuales en la parte baja del corpus.

5.5.2 Corpus1 francovoz. Etiquetas automáticas. Parte baja del corpus

Se entrenó y probó un reconocedor en HTK utilizando los archivos de la parte baja del corpus con etiquetas automáticas. Los resultados de reconocimiento se muestran en la figura 5.10.

```
====================== HTK Results Analysis =======================
  Date: Sun May 06 21:29:43 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=100, S=0, N=100]
WORD: %Corr=100.00, Acc=100.00 [H=100, D=0, S=0, I=0, N=100]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e   Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0     0
 uno   0  10   0   0   0   0   0   0   0   0     0
 dos   0   0  10   0   0   0   0   0   0   0     0
tres   0   0   0  10   0   0   0   0   0   0     0
cuat   0   0   0   0  10   0   0   0   0   0     0
cinc   0   0   0   0   0  10   0   0   0   0     0
seis   0   0   0   0   0   0  10   0   0   0     0
siet   0   0   0   0   0   0   0  10   0   0     0
ocho   0   0   0   0   0   0   0   0  10   0     0
nuev   0   0   0   0   0   0   0   0   0  10     0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.10. Resultados de reconocimiento para el corpus francovoz con etiquetas automáticas en la parte baja del corpus.

5.5.3 Corpus1 francovoz. Etiquetas manuales. Parte alta del corpus

Se tomó el reconocedor entrenado con etiquetas manuales de la parte baja y se probó con los archivos de la parte alta. Los resultados de reconocimiento se muestran en la figura 5.11.

```
====================== HTK Results Analysis =======================
  Date: Sun May 06 21:40:24 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=99, S=0, N=99]
WORD: %Corr=100.00, Acc=100.00 [H=99, D=0, S=0, I=0, N=99]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e   Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0     0
 uno   0  10   0   0   0   0   0   0   0   0     0
 dos   0   0  10   0   0   0   0   0   0   0     0
tres   0   0   0  10   0   0   0   0   0   0     0
cuat   0   0   0   0   9   0   0   0   0   0     0
cinc   0   0   0   0   0  10   0   0   0   0     0
seis   0   0   0   0   0   0  10   0   0   0     0
siet   0   0   0   0   0   0   0  10   0   0     0
ocho   0   0   0   0   0   0   0   0  10   0     0
nuev   0   0   0   0   0   0   0   0   0  10     0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.11. Resultados de reconocimiento para el corpus francovoz con etiquetas manuales en la parte alta del corpus.

5.5.4 Corpus1 francovoz. Etiquetas automáticas. Parte alta del corpus

Se tomó el reconocedor entrenado con etiquetas automáticas de la parte baja y se probó con los archivos de la parte alta. Los resultados de reconocimiento se muestran en la figura 5.12.

```
====================== HTK Results Analysis =======================
  Date: Sun May 06 21:54:13 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=99, S=0, N=99]
WORD: %Corr=100.00, Acc=100.00 [H=99, D=0, S=0, I=0, N=99]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0    0
 uno   0  10   0   0   0   0   0   0   0   0    0
 dos   0   0  10   0   0   0   0   0   0   0    0
tres   0   0   0  10   0   0   0   0   0   0    0
cuat   0   0   0   0   9   0   0   0   0   0    0
cinc   0   0   0   0   0  10   0   0   0   0    0
seis   0   0   0   0   0   0  10   0   0   0    0
siet   0   0   0   0   0   0   0  10   0   0    0
ocho   0   0   0   0   0   0   0   0  10   0    0
nuev   0   0   0   0   0   0   0   0   0  10    0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.12. Resultados de reconocimiento para el corpus francovoz con etiquetas automáticas en la parte alta del corpus.

5.5.3 Corpus2 lisethvoz. Etiquetas manuales. Parte baja del corpus

Se entrenó y probó un reconocedor en HTK utilizando los archivos de la parte baja del corpus con etiquetas manuales. Los resultados de reconocimiento se muestran en la figura 5.13.

```
====================== HTK Results Analysis =======================
  Date: Sun May 06 22:14:19 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=100, S=0, N=100]
WORD: %Corr=100.00, Acc=100.00 [H=100, D=0, S=0, I=0, N=100]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0    0
 uno   0  10   0   0   0   0   0   0   0   0    0
 dos   0   0  10   0   0   0   0   0   0   0    0
tres   0   0   0  10   0   0   0   0   0   0    0
cuat   0   0   0   0  10   0   0   0   0   0    0
cinc   0   0   0   0   0  10   0   0   0   0    0
seis   0   0   0   0   0   0  10   0   0   0    0
siet   0   0   0   0   0   0   0  10   0   0    0
ocho   0   0   0   0   0   0   0   0  10   0    0
nuev   0   0   0   0   0   0   0   0   0  10    0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.13. Resultados de reconocimiento para el corpus lisethvoz con etiquetas manuales en la parte baja del corpus.

5.5.4 Corpus2 lisethvoz. Etiquetas automáticas. Parte baja del corpus

Se entrenó y probó un reconocedor en HTK utilizando los archivos de la parte baja con etiquetas automáticas. Los resultados de reconocimiento se muestran en la figura 5.14.

```
====================== HTK Results Analysis =======================
  Date: Sun May 06 22:35:11 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=100, S=0, N=100]
WORD: %Corr=100.00, Acc=100.00 [H=100, D=0, S=0, I=0, N=100]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0    0
 uno   0  10   0   0   0   0   0   0   0   0    0
 dos   0   0  10   0   0   0   0   0   0   0    0
tres   0   0   0  10   0   0   0   0   0   0    0
cuat   0   0   0   0  10   0   0   0   0   0    0
cinc   0   0   0   0   0  10   0   0   0   0    0
seis   0   0   0   0   0   0  10   0   0   0    0
siet   0   0   0   0   0   0   0  10   0   0    0
ocho   0   0   0   0   0   0   0   0  10   0    0
nuev   0   0   0   0   0   0   0   0   0  10    0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.14. Resultados de reconocimiento para el corpus lisethvoz con etiquetas automáticas en la parte baja del corpus.

5.5.5 Corpus2 lisethvoz. Etiquetas manuales. Parte alta del corpus

Se tomó el reconocedor entrenado con etiquetas manuales de la parte baja y se probó con los archivos de la parte alta. Los resultados de reconocimiento se muestran en la figura 5.15.

```
====================== HTK Results Analysis =======================
  Date: Sun May 06 23:36:49 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=95.92 [H=94, S=4, N=98]
WORD: %Corr=95.92, Acc=95.92 [H=94, D=0, S=4, I=0, N=98]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0    0
 uno   0  10   0   0   0   0   0   0   0   0    0
 dos   0   0  10   0   0   0   0   0   0   0    0
tres   0   0   0  10   0   0   0   0   0   0    0
cuat   0   0   0   0  10   0   0   0   0   0    0
cinc   0   0   0   0   0  10   0   0   0   0    0
seis   0   0   0   1   0   2   5   1   0   0    0 [55.6/4.1]
siet   0   0   0   0   0   0   0   9   0   0    0
ocho   0   0   0   0   0   0   0   0  10   0    0
nuev   0   0   0   0   0   0   0   0   0  10    0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.15. Resultados de reconocimiento para el corpus lisethvoz utilizando etiquetas manuales en la parte alta del corpus.

5.5.6 Corpus2 lisethvoz. Etiquetas automáticas. Parte alta del corpus

Se tomó el reconocedor entrenado con etiquetas automáticas de la parte baja y se probó con los archivos de la parte alta. Los resultados de reconocimiento se muestran en la figura 5.16.

```
====================== HTK Results Analysis =======================
  Date: Sun May 06 23:35:21 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=98, S=0, N=98]
WORD: %Corr=100.00, Acc=100.00 [H=98, D=0, S=0, I=0, N=98]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0    0
 uno   0  10   0   0   0   0   0   0   0   0    0
 dos   0   0  10   0   0   0   0   0   0   0    0
tres   0   0   0  10   0   0   0   0   0   0    0
cuat   0   0   0   0  10   0   0   0   0   0    0
cinc   0   0   0   0   0  10   0   0   0   0    0
seis   0   0   0   0   0   0   9   0   0   0    0
siet   0   0   0   0   0   0   0   9   0   0    0
ocho   0   0   0   0   0   0   0   0  10   0    0
nuev   0   0   0   0   0   0   0   0   0  10    0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.16. Resultados de reconocimiento del corpus lisethvoz utilizando etiquetas automáticas en la parte alta del corpus.

5.5.7 Corpus3 rodrigovoz. Etiquetas manuales. Parte baja del corpus

Se entrenó y probó un reconocedor en HTK utilizando los archivos de la parte baja con etiquetas manuales. Los resultados de reconocimiento se muestran en la figura 5.17.

```
====================== HTK Results Analysis =======================
  Date: Mon May 07 01:39:13 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=100, S=0, N=100]
WORD: %Corr=100.00, Acc=100.00 [H=100, D=0, S=0, I=0, N=100]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0    0
 uno   0  10   0   0   0   0   0   0   0   0    0
 dos   0   0  10   0   0   0   0   0   0   0    0
tres   0   0   0  10   0   0   0   0   0   0    0
cuat   0   0   0   0  10   0   0   0   0   0    0
cinc   0   0   0   0   0  10   0   0   0   0    0
seis   0   0   0   0   0   0  10   0   0   0    0
siet   0   0   0   0   0   0   0  10   0   0    0
ocho   0   0   0   0   0   0   0   0  10   0    0
nuev   0   0   0   0   0   0   0   0   0  10    0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.17. Resultados de reconocimiento para el corpus rodrigovoz usando etiquetas manuales en la parte baja del corpus.

5.5.8 Corpus3 rodrigovoz. Etiquetas automáticas. Parte baja del corpus

Se entrenó y probó un reconocedor en HTK utilizando los archivos de la parte baja con etiquetas automáticas. Los resultados de reconocimiento se muestran en la figura 5.18.

```
====================== HTK Results Analysis =======================
  Date: Mon May 07 01:30:21 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=100, S=0, N=100]
WORD: %Corr=100.00, Acc=100.00 [H=100, D=0, S=0, I=0, N=100]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e   Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0     0
 uno   0  10   0   0   0   0   0   0   0   0     0
 dos   0   0  10   0   0   0   0   0   0   0     0
tres   0   0   0  10   0   0   0   0   0   0     0
cuat   0   0   0   0  10   0   0   0   0   0     0
cinc   0   0   0   0   0  10   0   0   0   0     0
seis   0   0   0   0   0   0  10   0   0   0     0
siet   0   0   0   0   0   0   0  10   0   0     0
ocho   0   0   0   0   0   0   0   0  10   0     0
nuev   0   0   0   0   0   0   0   0   0  10     0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.18. Resultados de reconocimiento para el corpus rodrigovoz usando etiquetas automáticas en la parte baja del corpus.

5.5.9 Corpus3 rodrigovoz. Etiquetas manuales. Parte alta del corpus

Se tomó el reconocedor entrenado con etiquetas manuales de la parte baja y se probó con los archivos de la parte alta. Los resultados de reconocimiento se muestran en la figura 5.19.

```
====================== HTK Results Analysis =======================
  Date: Mon May 07 02:00:31 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=97.96 [H=96, S=2, N=98]
WORD: %Corr=97.96, Acc=97.96 [H=96, D=0, S=2, I=0, N=98]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e   Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0     0
 uno   0   9   0   0   0   0   0   0   1   0     0 [90.0/1.0]
 dos   0   0  10   0   0   0   0   0   0   0     0
tres   0   0   0  10   0   0   0   0   0   0     0
cuat   0   0   0   0   9   0   0   0   0   0     0
cinc   0   0   0   0   0  10   0   0   0   0     0
seis   0   0   0   0   0   0  10   0   0   0     0
siet   0   0   0   0   0   0   0  10   0   0     0
ocho   0   0   0   0   0   0   0   0   9   0     0
nuev   0   0   0   1   0   0   0   0   0   9     0 [90.0/1.0]
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.19. Resultados de reconocimiento para el corpus rodrigovoz con etiquetas manuales en la parte alta del corpus.

5.5.10 Corpus3 rodrigovoz. Etiquetas automáticas. Parte alta del corpus

Se tomó el reconocedor entrenado con etiquetas automáticas de la parte baja y se probó con los archivos de la parte alta. Los resultados de reconocimiento se muestran en la figura 5.20.

```
====================== HTK Results Analysis =======================
  Date: Mon May 07 02:17:21 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=98.98 [H=97, S=1, N=98]
WORD: %Corr=98.98, Acc=98.98 [H=97, D=0, S=1, I=0, N=98]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0    0
 uno   0   9   0   0   0   0   0   0   1   0    0 [90.0/1.0]
 dos   0   0  10   0   0   0   0   0   0   0    0
tres   0   0   0  10   0   0   0   0   0   0    0
cuat   0   0   0   0   9   0   0   0   0   0    0
cinc   0   0   0   0   0  10   0   0   0   0    0
seis   0   0   0   0   0   0  10   0   0   0    0
siet   0   0   0   0   0   0   0  10   0   0    0
ocho   0   0   0   0   0   0   0   0   9   0    0
nuev   0   0   0   0   0   0   0   0   0  10    0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.20. Resultados de reconocimiento para el corpus rodrigovoz con etiquetas automáticas en la parte alta del corpus.

5.5.11 Corpus4 sergiovoz. Etiquetas manuales. Parte baja del corpus

Se entrenó y probó un reconocedor en HTK utilizando los archivos de la parte baja con etiquetas manuales. Los resultados de reconocimiento se muestran en la figura 5.21.

```
====================== HTK Results Analysis =======================
  Date: Mon May 07 02:44:20 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=100, S=0, N=100]
WORD: %Corr=100.00, Acc=100.00 [H=100, D=0, S=0, I=0, N=100]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0    0
 uno   0  10   0   0   0   0   0   0   0   0    0
 dos   0   0  10   0   0   0   0   0   0   0    0
tres   0   0   0  10   0   0   0   0   0   0    0
cuat   0   0   0   0  10   0   0   0   0   0    0
cinc   0   0   0   0   0  10   0   0   0   0    0
seis   0   0   0   0   0   0  10   0   0   0    0
siet   0   0   0   0   0   0   0  10   0   0    0
ocho   0   0   0   0   0   0   0   0  10   0    0
nuev   0   0   0   0   0   0   0   0   0  10    0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.21. Resultados de reconocimiento para el corpus sergiovoz con etiquetas manuales en la parte baja del corpus.

5.5.12 Corpus4 sergiovoz. Etiquetas automáticas. Parte baja del corpus

Se entrenó y probó un reconocedor en HTK utilizando los archivos de la parte baja con etiquetas automáticas. Los resultados de reconocimiento se muestran en la figura 5.22.

```
====================== HTK Results Analysis =======================
  Date: Mon May 07 03:00:27 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=100, S=0, N=100]
WORD: %Corr=100.00, Acc=100.00 [H=100, D=0, S=0, I=0, N=100]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  10   0   0   0   0   0   0   0   0   0    0
 uno   0  10   0   0   0   0   0   0   0   0    0
 dos   0   0  10   0   0   0   0   0   0   0    0
tres   0   0   0  10   0   0   0   0   0   0    0
cuat   0   0   0   0  10   0   0   0   0   0    0
cinc   0   0   0   0   0  10   0   0   0   0    0
seis   0   0   0   0   0   0  10   0   0   0    0
siet   0   0   0   0   0   0   0  10   0   0    0
ocho   0   0   0   0   0   0   0   0  10   0    0
nuev   0   0   0   0   0   0   0   0   0  10    0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.22. Resultados de reconocimiento para el corpus sergiovoz usando etiquetas automáticas en la parte baja del corpus.

5.5.13 Corpus4 sergiovoz. Etiquetas manuales. Parte alta del corpus

Se tomó el reconocedor entrenado con etiquetas manuales de la parte baja y se probó con los archivos de la parte alta. Los resultados de reconocimiento se muestran en la figura 5.23.

```
====================== HTK Results Analysis =======================
  Date: Mon May 07 03:18:39 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=97.92 [H=94, S=2, N=96]
WORD: %Corr=97.92, Acc=97.92 [H=94, D=0, S=2, I=0, N=96]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero   9   0   0   0   0   0   0   0   0   0    0
 uno   0  10   0   0   0   0   0   0   0   0    0
 dos   0   0  10   0   0   0   0   0   0   0    0
tres   0   0   0  10   0   0   0   0   0   0    0
cuat   0   0   0   0   9   0   0   0   0   0    0
cinc   0   0   0   0   0  10   0   0   0   0    0
seis   0   0   0   0   0   2   8   0   0   0    0 [80.0/2.1]
siet   0   0   0   0   0   0   0   9   0   0    0
ocho   0   0   0   0   0   0   0   0  10   0    0
nuev   0   0   0   0   0   0   0   0   0   9    0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.23. Resultados de reconocimiento para el corpus sergiovoz con etiquetas manuales en la parte alta del corpus.

5.5.14 Corpus4 sergiovoz. Etiquetas automáticas. Parte alta del corpus

Se tomó el reconocedor entrenado con etiquetas automáticas de la parte baja y se probó con los archivos de la parte alta. Los resultados de reconocimiento se muestran en la figura 5.24.

```
====================== HTK Results Analysis =======================
  Date: Mon May 07 03:26:56 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=96, S=0, N=96]
WORD: %Corr=100.00, Acc=100.00 [H=96, D=0, S=0, I=0, N=96]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e   Del [ %c / %e]
cero   9   0   0   0   0   0   0   0   0   0     0
 uno   0  10   0   0   0   0   0   0   0   0     0
 dos   0   0  10   0   0   0   0   0   0   0     0
tres   0   0   0  10   0   0   0   0   0   0     0
cuat   0   0   0   0   9   0   0   0   0   0     0
cinc   0   0   0   0   0  10   0   0   0   0     0
seis   0   0   0   0   0   0  10   0   0   0     0
siet   0   0   0   0   0   0   0   9   0   0     0
ocho   0   0   0   0   0   0   0   0  10   0     0
nuev   0   0   0   0   0   0   0   0   0   9     0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Figura 5.24. Resultados de reconocimiento para el corpus sergiovoz usando etiquetas automáticas en la parte alta del corpus.

5.6 Comparación de los resultados

Para analizar los resultados de los experimentos, definimos las siguientes abreviaturas:
M – Manual.
AF – Alineación forzada.
SA – Segmentación acústica.
AFSA – Alineación forzada con segmentación acústica.
AFSATF – Alineación forzada con segmentación acústica tomando en cuenta el tipo de fronteras.

Corpus de 150 tipos de palabras
En la tabla 5.19 se resumen los resultados obtenidos con el corpus que contiene todos los fonemas. En este conjunto de experimentos se utilizaron todos los archivos de etiquetas manuales para entrenar al sistema y generar todas las etiquetas automáticas.
Por otro lado, los sistemas de reconocimiento de HTK se entrenaron y probaron usando todos los archivos del corpus.

Tabla 5.19. Resultados de experimentos para el corpus con todos los fonemas.

Concepto	M	AF	AFSA	AFSATF
Error medio cuadrático (ms)	0	16.2902	17.9711	17.7177
Porcentaje de fronteras correctas dentro de 10 ms	100	61.0259	54.2305	55.4468
Porcentaje de fronteras correctas dentro de 20 ms	100	89.0798	83.6065	84.4262
Porcentaje de fronteras correctas dentro de 30 ms	100	95.0555	93.5219	93.6541
Porcentaje de fronteras correctas dentro de 40 ms	100	97.3823	97.0650	97.0914
Porcentaje de fronteras correctas dentro de 50 ms	100	98.4135	98.2813	98.4399
Porcentaje de reconocimiento	98.61	98.50	98.33	98.39

Corpus de dígitos de 4 hablantes
En la tabla 5.20 se resumen los resultados obtenidos para el corpus de dígitos. En este conjunto de experimentos se utilizó únicamente una cuarta parte de etiquetas manuales para entrenar al sistema de etiquetado automático y generar todas las etiquetas automáticas.
Por otro lado, los sistemas de reconocimiento de HTK se entrenaron y probaron usando todos los archivos del corpus.

Tabla 5.20. Resultados de experimentos para el corpus de dígitos.

Concepto	M	AF	AFSATF
Error medio cuadrático (ms)	0	49.4323	49.8570
Porcentaje de fronteras correctas dentro de 10 ms	100	40.4322	37.7831
Porcentaje de fronteras correctas dentro de 20 ms	100	66.6782	63.8550
Porcentaje de fronteras correctas dentro de 30 ms	100	77.3440	76.2286
Porcentaje de fronteras correctas dentro de 40 ms	100	82.9208	82.8163
Porcentaje de fronteras correctas dentro de 50 ms	100	86.7201	86.5109
Porcentaje de reconocimiento	95.24	96.74	94.36

Cuatro corpus de dígitos un solo hablante
A continuación se presenta un resumen de los resultados para los cuatro corpus de dígitos.
Cada uno de los cuatro corpus se dividió por la mitad; la mitad que corresponde a las primeras repeticiones es la parte baja; la otra mitad es la parte alta. El sistema de etiquetado automático se entrenó utilizando la parte baja de cada corpus para luego etiquetar la totalidad de los mismos.

Por otro lado, los sistemas de reconocimiento en HTK se entrenaron únicamente con la parte baja de cada corpus; primero con etiquetas manuales y luego con automáticas. Después, los sistemas se probaron utilizando tanto la parte baja como la parte alta de cada corpus.
La parte alta de cada corpus no participó en el entrenamiento, sólo en el reconocimiento. Adicionalmente, también se presenta la concordancia para cada corpus, como se puede observar en las tablas 5.21.a hasta 5.24.b.

Tabla 5.21.a. Resultados de reconocimiento para el corpus franciscovoz.

francovoz	Parte baja	Parte alta
Etiquetas manuales	100%	100%
Etiquetas automáticas	100%	100%

Tabla 5.21.b. Resultados de concordancia para el corpus franciscovoz.

francovoz	
Error medio cuadrático (ms)	20.7700
Porcentaje de fronteras correctas dentro de 10 ms	65.6944
Porcentaje de fronteras correctas dentro de 20 ms	88.7500
Porcentaje de fronteras correctas dentro de 30 ms	93.4722
Porcentaje de fronteras correctas dentro de 40 ms	94.5833
Porcentaje de fronteras correctas dentro de 50 ms	95.1388

Tabla 5.22.a. Resultados de reconocimiento para el corpus lisethvoz.

lisethvoz	Parte baja	Parte alta
Etiquetas manuales	100%	92.95%
Etiquetas automáticas	100%	100%

Tabla 5.22.b. Resultados de concordancia para el corpus lisethvoz.

lisethvoz	
Error medio cuadrático (ms)	20.2340
Porcentaje de fronteras correctas dentro de 10 ms	78.2303
Porcentaje de fronteras correctas dentro de 20 ms	93.1179
Porcentaje de fronteras correctas dentro de 30 ms	95.5056
Porcentaje de fronteras correctas dentro de 40 ms	95.9269
Porcentaje de fronteras correctas dentro de 50 ms	96.7696

Tabla 5.23.a. Resultados de reconocimiento para el corpus rodrigovoz.

rodrigovoz	Parte baja	Parte alta
Etiquetas manuales	100%	97.96%
Etiquetas automáticas	100%	98.98%

Tabla 5.23.b. Resultados de concordancia para el corpus rodrigovoz.

rodrigovoz	
Error medio cuadrático (ms)	24.1340
Porcentaje de fronteras correctas dentro de 10 ms	55.8988
Porcentaje de fronteras correctas dentro de 20 ms	82.2033
Porcentaje de fronteras correctas dentro de 30 ms	91.1516
Porcentaje de fronteras correctas dentro de 40 ms	92.8370
Porcentaje de fronteras correctas dentro de 50 ms	94.8033

Tabla 5.24.a. Resultados de reconocimiento para el corpus sergiovoz.

sergiovoz	Parte baja	Parte alta
Etiquetas manuales	100%	97.92%
Etiquetas automáticas	100%	100%

Tabla 5.24.b. Resultados de concordancia para el corpus sergiovoz.

francovoz	
Error medio cuadrático (ms)	24.1696
Porcentaje de fronteras correctas dentro de 10 ms	59.1666
Porcentaje de fronteras correctas dentro de 20 ms	84.7222
Porcentaje de fronteras correctas dentro de 30 ms	93.1944
Porcentaje de fronteras correctas dentro de 40 ms	96.9444
Porcentaje de fronteras correctas dentro de 50 ms	97.9166

5.7 Resumen

En este capítulo se presentaron diversos experimentos.
Primero se presentaron los experimentos relativos únicamente a segmentación acústica, donde se mostró el proceso para elegir los valores de los parámetros escogidos.

Se presentaron experimentos utilizando un corpus de un hablante con 150 tipos de palabras, y 4 repeticiones (600 archivos). En estos experimentos se determinó que la alineación forzada es mejor por sí misma que la combinación alineación forzada-segmentación acústica.

Se etiquetó manualmente y automáticamente todo el corpus. Se entrenó y probó un sistema de reconocimiento, primero con etiquetas manuales y luego con etiquetas automáticas usando todos los archivos del corpus. El sistema entrenado con etiquetas automáticas tuvo un porcentaje de reconocimiento menor, pero muy cercano al sistema entrenado con etiquetas manuales (98.50% contra 98.61%).
La concordancia obtenida fue de 89.0798% para 20 ms.

Luego se presentaron experimentos utilizando un corpus de dígitos de cuatro hablantes, diez tipos de palabras repetidas 20 veces.
El corpus fue etiquetado manualmente y automáticamente.
Se entrenó y probó un sistema de reconocimiento primero con etiquetas manuales y luego con etiquetas automáticas. El sistema entrenado con etiquetas automáticas tuvo un porcentaje de reconocimiento superior al entrenado con etiquetas manuales (96.74% contra 95.24%). La concordancia obtenida fue de 66.6782% para 20 ms.

Después se presentaron experimentos sobre cuatro corpus de dígitos de un sólo hablante, diez tipos de palabras repetidas 20 veces.
Los corpus fueron etiquetados manualmente y automáticamente utilizando alineación forzada.
Además, los corpus fueron partidos a la mitad: una mitad para entrenar un sistema de reconocimiento y la otra mitad para probar dicho sistema de reconocimiento.
En todos los casos, los sistemas entrenados con etiquetas automáticas mejoraron el porcentaje de reconocimiento alcanzado por sistemas entrenados con etiquetas manuales.
El resultado más notable es para el corpus *lisethvoz*, donde el sistema entrenado con etiquetas manuales sólo obtuvo 95.92% de reconocimiento, mientras que el sistema entrenado con etiquetas automáticas obtuvo 100% de reconocimiento. La mejor concordancia también ocurrió en el corpus *lisethvoz*, con 93.1179% para 20 ms.

CONCLUSIONES Y TRABAJO A FUTURO

Objetivos cumplidos

Se cumplieron los objetivos planteados en el capítulo 1:

- Se desarrolló un sistema segmentador y etiquetador semiautomático fonético para corpus de voces en español. El sistema fue implementado en un conjunto de programas elaborados en Matlab, los cuales se describieron en el capítulo 4.
- Se utilizaron las técnicas de segmentación y etiquetado dentro de corpus de voces en español. Se implementó el algoritmo de alineación forzada y se implementó un algoritmo de segmentación acústica.
- Se realizó la extracción de parámetros wavelet utilizando Gaussianas moduladas en la escala de Bark para la segmentación automática de fronteras acústicas. El sistema propuesto realiza un análisis wavelet en Gaussianas moduladas para segmentar acústicamente la señal, utilizando para ello la detección de eventos (cambios acústicos).
- Se realizó etiquetado automático utilizando alineación forzada y se determinó que la segmentación acústica con wavelets no contribuye a la mejora del desempeño de la alineación forzada por sí misma.
- Se comparó el porcentaje de reconocimiento de corpus de voces con etiquetado manual contra el obtenido con etiquetado automático en HTK y se observó que los resultados de las etiquetas automáticas son mejores o similares que los de etiquetas manuales.

La hipótesis planteada resultó no ser cierta parcialmente, ya que la alineación forzada por sí misma produce mejores resultados que la combinación de alineación forzada con segmentación acústica; tanto en concordancia de fronteras fonéticas como en el reconocimiento de palabras. No se encontró evidencia experimental de que la segmentación acústica mejore a la segmentación por alineación forzada.

Ventajas

Ahorro de tiempo.
El ahorro de tiempo es significativo. El tiempo utilizado para etiquetar manualmente los corpus usados fue de dos días para el corpus de dígitos, y tres días para el corpus fonéticamente completo. En cambio, el sistema puede tardarse sólo una hora en generar un libro de código (esta es la etapa más tardada), y tardarse aproximadamente 20 minutos en etiquetar cada uno de los corpus antes mencionados.

Consistencia.
Las etiquetas manuales no son consistentes, aún cuando sea una sola persona quien etiqueta. El sistema automático es consistente. Nuestros resultados muestran que, en el caso del corpus de dígitos, la concordancia con las etiquetas manuales fue baja (el porcentaje de fronteras correctas dentro de 20 ms fue de 66.6782%); sin embargo, al comparar los porcentajes de reconocimiento, las etiquetas automáticas fueron superiores a las manuales (las etiquetas automáticas obtuvieron un porcentaje de reconocimiento de 96.74%, mientras que las etiquetas manuales obtuvieron 95.24%).
Este mismo fenómeno se observó en los experimentos con los cuatro corpus de dígitos individuales, donde se observó que las etiquetas automáticas produjeron resultados mejores o iguales que las etiquetas manuales.

Esto nos lleva a la conclusión de que las etiquetas manuales no necesariamente son las mejores, ya que siempre serán inconsistentes; y por lo tanto no es necesario buscar que las etiquetas automáticas sean 100% iguales a las etiquetas manuales; en cuyo caso, las etiquetas automáticas serían inconsistentes también.

La propuesta de segmentación acústica

Se compararon el algoritmo de segmentación acústica original de Galka-Ziolko, con el algoritmo modificado en este trabajo, se realizaron experimentos de segmentación acústica sobre el corpus de 150 tipos de palabras y 4 repeticiones, con el objetivo de maximizar el porcentaje de fronteras correctamente colocadas dentro de 10 ms y hacer una comparación.

El algoritmo propuesto en este trabajo produce un porcentaje de fronteras correctas (91.2596%) mayor que el algoritmo original (80.1954%), pero pagando el precio de un mayor porcentaje de inserciones. Si lo que se busca es disminuir las inserciones, es mejor utilizar el algoritmo original; sí lo que se busca es aumentar el número de fronteras colocadas correctamente respecto a las fronteras manuales (como en este trabajo), es mejor utilizar el algoritmo propuesto.

Resultados de reconocimiento para los corpus

En el caso del corpus de 150 tipos de palabras solamente se hicieron cuatro repeticiones de cada palabra, así que para cada palabra se utilizaron los cuatro archivos de etiquetas manuales para entrenar al sistema de segmentación automática y después se segmentó automáticamente la totalidad del corpus.
La mejor concordancia alcanzada fue de 89.0798% dentro de 20 ms.
El porcentaje de reconocimiento con etiquetas manuales fue de 98.61%.
El mejor porcentaje de reconocimiento con etiquetas automáticas fue de 98.50%

En el caso del corpus de dígitos con los cuatro hablantes mezclados se contó con ochenta repeticiones por cada palabra. Solamente se utilizó una cuarta parte de los archivos etiquetados manualmente para entrenar al sistema de segmentación automática. Luego se segmentó automáticamente la totalidad del corpus.
La mejor concordancia fue de 66.6782% dentro de 20 ms.
El porcentaje de reconocimiento con etiquetas manuales fue de 95.24%.
El mejor porcentaje de reconocimiento con etiquetas automáticas fue de 96.74%

En el caso de los cuatro corpus de dígitos de un hablante por separado, el sistema de segmentación automática se entrenó solamente con las etiquetas manuales de la mitad de cada corpus; luego la totalidad de cada corpus fue segmentada automáticamente. Los experimentos de reconocimiento se efectuaron sobre el conjunto de archivos de entrenamiento y luego sobre el conjunto de archivos que no participaron en el entrenamiento.
La concordancia varió de uno a otro corpus.

El corpus *francovoz* tuvo una concordancia de 88.75% dentro de 20 ms.
El porcentaje de reconocimiento sobre archivos que no participaron en el entrenamiento, con etiquetas manuales fue de 100%.
El porcentaje de reconocimiento sobre archivos que no participaron en el entrenamiento, con etiquetas automáticas fue de 100%.

El corpus *lisethvoz* tuvo una concordancia de 93.1179% dentro de 20 ms.

El porcentaje de reconocimiento sobre archivos que no participaron en el entrenamiento, con etiquetas manuales fue de 92.95%.
El porcentaje de reconocimiento sobre archivos que no participaron en el entrenamiento, con etiquetas automáticas fue de 100%.

El corpus *rodrigovoz* tuvo una concordancia de 82.2033% dentro de 20 ms.
El porcentaje de reconocimiento sobre archivos que no participaron en el entrenamiento, con etiquetas manuales fue de 97.96%
El porcentaje de reconocimiento sobre archivos que no participaron en el entrenamiento, con etiquetas automáticas fue de 98.98%.

El corpus *sergiovoz* obtuvo una concordancia de 84.7222% dentro de 20 ms.
El porcentaje de reconocimiento sobre archivos que no participaron en el entrenamiento, con etiquetas manuales fue de 97.92%.
El porcentaje de reconocimiento sobre archivos que no participaron en el entrenamiento, con etiquetas automáticas fue de 100%.

En todos los experimentos, exceptuando al del corpus de 150 palabras con 4 repeticiones, las etiquetas automáticas produjeron mejores resultados de reconocimiento que las etiquetas manuales. En el caso mencionado, la diferencia en el porcentaje de reconocimiento es de sólo 0.11% y se considera irrelevante.

Comparación con el estado del arte

En el tema de segmentación y etiquetado automático los autores se comparan por el porcentaje de fronteras correctamente colocadas dentro de 20 ms. Esta es una medición de concordancia entre las etiquetas manuales y las etiquetas automáticas. En este trabajo también se realizó la comparación entre los porcentajes de reconocimiento obtenidos con sistemas entrenados con etiquetas automáticas y sistemas entrenados con etiquetas manuales.

Tomando en cuenta únicamente la concordancia, los trabajos más importantes son: [29] donde se reportó un porcentaje de fronteras correctas dentro de 20 ms de 92.57%; mientras que en [54] se reportó un porcentaje de fronteras correctas dentro de 20 ms de 96.01%.

Trabajo a futuro

- Realizar un corpus grande y fonéticamente balanceado de varios hablantes. Realizar el etiquetado manual por varios especialistas.
- Realizar pruebas tomando en cuenta el contexto de los fonemas (difonemas, trifonemas).
- Modificar los programas realizados para poder trabajar con cualquier alfabeto fonético.
- Segmentar los diptongos como una unidad y verificar su efecto en la concordancia alcanzada.

Trabajos derivados

Se publicó el trabajo titulado *Segmentación de habla a nivel de fonemas usando wavelets Gaussianas y análisis de energía en subbandas*., con los autores Cristian-Remington Juárez-Murillo, Sergio Suárez-Guerra, José-Luis Oropeza-Rodríguez en el Octavo Congreso Internacional de Cómputo en Optimización y Software. CICOS 2011. 22-25 Noviembre, UAEM México.

REFERENCIAS

[1] Alani, A., Deriche, M. (1999) A Novel Approach to Speech Segmentation Using the Wavelet Transform. Signal Processing and Its Applications, 1999. ISSPA'99. Proceedings of the Fifth International Symposium on, vol. 1, no.,pp. 127-130 vol.1, 1999. doi: 10.1109/ISSPA.1999.818129.

[2] Baker, J. K. (1972) Machine –Aided Labeling of Connected Speech. En Working Papers in Speech Recognition – II. Computer Science Department, Carnegie-Mellon University, 1972.

[3] Baker, J.K. (1979) Stochastic Modeling for Automatic Speech Understanding. En: Waibel, A., Lee, K.F. (eds) (1990) Readings in Speech Recognition. Morgan Kauffman Publishers, Inc. San Francisco California, USA, pp. 297 – 307.

[4] Becchetti, C., Ricotti, L.C. (1999) Speech Recognition. Theory and C++ Implementation. John Wiley & Sons, LTD, pp. 74 – 76.

[5] Bernal, J., Bobadilla, J., Gómez, P. (2000) Reconocimiento de voz y fonética acústica. Alfaomega, México, D.F., 2000, pp.42-83.

[6] Carnegie Mellon University (2002) The CMU Pronouncing Dictionary.
http://www.speech.cs.cmu.edu/cgi-bin/cmudict?in=C+M+U+Dictionary
Consultado el 30 de noviembre de 2011.

[7] Cole, R. A., Mariani, J., Uszkoreit, H., Varile, G., Zaenen, A., Zue, V., Zampolli, A. (1997) Survey of the State of the Art in Human Language Technology. Cambridge University Press and Giardini.

[8] Cuétara, J. (2004) Fonética de la Ciudad de México. Aportaciones desde las tecnologías del habla. Tesis para obtener el título de Maestro en Lingüística Hispánica. UNAM, pp. 47-51.

[9] De Mori, R., Laface, P. (1980) Use of Fuzzy Algorithms for Phonetic and Phonemic Labeling of Continuous Speech. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.PAMI-2, no.2, pp. 136-148, Marzo 1980. doi: 10.1109/TPAMI.1980.4766991.

[10] Deller, J.R., Hansen, J.H.L., Proakis, J.G. (2000a) Discrete-Time Processing of Speech Signals. IEEE Press. New York, pp 116-117.

[11] Deller, J.R., Hansen, J.H.L., Proakis, J.G. (2000b) Discrete-Time Processing of Speech Signals. IEEE Press. New York, pp 677-683.

[12] Demuynck, K., Laureys, T. (2002) A Comparison of Different Approaches to Automatic Speech Segmentation. Proceedings of the 5th International Conference on Text, Speech and Dialogue. Springer-Verlag, pp. 277-284.

[13] Ellis, D. (2005) PLP and RASTA (and MFCC, and inversion) in Matlab using melfcc.m and invmelfcc.m. http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/. Consultado el 27 de febrero de 2012.

[14] Falconer, K. (1990) Fractal Geometry. Mathematical Foundations and Applications. John Wiley & Sons, pp. xiii-xviii.

[15] Fantinato et al. (2008) A Fractal-Based Approach for Speech Segmentation. Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on, vol., no., pp.551-555, 15-17 Diciembre 2008. doi: 10.1109/ISM.2008.123.

[16] Finster, H. (1992) Automatic Speech Segmentation using Neural Network and Phonetic Transcription. Neural Networks, 1992. IJCNN., International Joint Conference on, vol.4, no.,pp.734-736 vol.4, 7-11 Junio 1992. doi: 10.1109/IJCNN.1992.227231.

[17] Flanagan, J.L., Benesti, J., Sondhi, M.M., Huang, Y. (2008) Springer Handbook of Speech Processing. Pp.V.

[18] Flores-Paulin, J.C. (2009) Técnicas para el Reconocimiento de Voz en Palabras Aisladas en la Lengua Náhuatl. Tesis de Maestría. Centro de Investigación en Computación. Instituto Politécnico Nacional, pp.20.

[19] Fogel, D. B. (1999) An Overview of Evolutionary Algorithms. Evolutionary Algorithms. The IMA Volumes in Mathematics and its Applications. Vol 1. pp. 89-109.

[20] Furui, S. (2001a) Digital Speech Processing, Synthesis, and Recognition. Marcel Dekker, Inc. New York, pp. 244.

[21] Furui, S. (2001b) Digital Speech Processing, Synthesis, and Recognition. Marcel Dekker, Inc. New York, pp 225.

[22] Galka, J., Ziolko, M. (2008) Wavelets in Speech Segmentation. Electrotechnical Conference, 2008. MELECON 2008. The 14th IEEE Mediterranean, vol., no., pp.876-879, 5-7 Mayo 2008.

[23] Gómez, J. A., Castro, M. J. (2002) Automatic Segmentation of Speech at the Phonetic Level. En: Structural, Syntactic, and Statistical Pattern Recognition. Lecture Notes in Computer Science, 2002, vol., 2396/2002, pp.883-921. doi:10.1007/3-540-70659-3_70.

[24] Grieder, W., Kinsner, W. (1994) Speech Segmentation by Variance Fractal Dimension. Electrical and Computer Engineering, 1994. Conference Proceedings. 1994 Canadian Conference on, vol., no., pp.481-485 vol.2, 25-28 Septiembre 1994. doi: 10.1109/CCECE.1994.405793.

[25] Haykin, S. (2005). Neural Networks. A Comprehensive Foundation. Second Edition, Pearson Prentice Hall, pp. 24.

[26] Hieronymus, J.L. (1993) ASCII Phonetic Symbols for the World's Languages: Worldbet. http://www.ling.ohio-state.edu/~edwards/WorldBet/worldbet.pdf. Consultado el 8 de marzo de 2012.

[27] Hosom, J.P. (2000a) Automatic Time Alignment of Phonemes Using Acoustic-Phonetic Information. Oregon Graduate Institute of Science and Technology. Ph.D. Thesis. pp.163-163. http://www.cslu.ogi.edu/people/hosom/hosom_thesis.pdf. Consultado el 8 de marzo de 2012.

[28] Hosom, J.P. (2000b) Automatic Time Alignment of Phonemes Using Acoustic-Phonetic Information. Oregon Graduate Institute of Science and Technology. Ph.D. Thesis. pp.3-4. http://www.cslu.ogi.edu/people/hosom/hosom_thesis.pdf. Consultado el 8 de marzo de 2012.

[29] Hosom, J.P. (2000c) Automatic Time Alignment of Phonemes Using Acoustic-Phonetic Information. Oregon Graduate Institute of Science and Technology. Ph.D. Thesis. pp.125-132. http://www.cslu.ogi.edu/people/hosom/hosom_thesis.pdf. Consultado el 8 de marzo de 2012.

[30] Hsieh, C.-T., Su, M.C., Chienn, S.C. (1995) Use of a Self-Learning Neuro-Fuzzy System for Syllabic Labeling of Continuous Speech. Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium., Proceedings of 1995 IEEE International Conference on, vol.4, no., pp.1127-1734 vol.4, 20-24 Marzo 1995. doi: 10.1109/FUZZY.1995.409915.

[31] Janer, L., Martí, J., Nadeu, C., Lleida-Solano, E. (1996) Wavelet Transforms for Non-Uniform Speech Recognition Systems. Spoken Language, 1996. ICSLP 96. Proceedings, Fourth International Conference on, vol.4, no., pp.2348-2351 vol.4, 3-6 Octubre 1996. doi: 10.1109/ICSLP.1996.607279.

[32] Janer, L. (1998) Transformada Wavelet Aplicada a la Extracción de Información en Señales de Voz. Tesis Doctoral. Universitat Politécnica de Catalunya. Barcelona, Mayo, 1998, pp. 63-73.

[33] Kacur, J., Cepko, J., Páleník, A. (2008) Automatic Labeling Schemes for Concatenative Speech Synthesis. ELMAR, 2008. 50th International Symposium, vol.2, no., p.639-642, 10-12 Septiembre 2008.

[34] Kaiser, G. (1994) A Friendly Guide to Wavelets. New York: Birkhauser.

[35] Lander, T., (1997) The CSLU Labeling Guide. Center of Spoken Language Understanding, Oregon Graduate Institute, pp.12-12. http://www.cslu.ogi.edu/corpora/docs/labeling.pdf. Consultado el 8 de marzo de 2012.

[36] Leondes, C.T. (1999) Fuzzy Theory Systems. Techniques and Applications. Vol. 1, Academic Press, pp. 3-4.

[37] Milone, D.H., Merelo, J.J., Rufiner, H.L. (2002) Evolutionary Algorithm for Speech Segmentation. Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on, vol.2, no., pp1115-1120, 2002. doi: 10.1109/CEC.2002.1004399

[38] Murphy, K. (1998), Hidden Markov Model (HMM) Toolbox for Matlab. http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html. Consultado el 27 de febrero de 2012.

[39] Olivier, A., Kirschning, I. (1999). Evaluación de métodos de determinación automática de una transcripción fonética. Segundo Encuentro Nacional de Computación 1999, ENC99, Pachuca, Hidalgo, México. Septiembre de 1999.

[40] Ortuño, M. (1980) Teoría y práctica de la lingüística moderna. Editorial Trillas.

[41] Proakis, J. G., Manolakis, D. G. (2007) Tratamiento digital de señales. Pearson Prentice Hall. pp. 204-205.

[42] Quilis, A. (1988) Fonética Acústica de la Lengua Española. Editorial Gredos. pp. 128-129.

[43] Rabiner, L. R., Sambur, M. R. (1974) Algorithm for Determining the Endpoints of Isolated Utterances. J. Acoust. Soc. Am. Vol. 56. Issue S1, pp S31-S36. doi: 10.1121/1.1914118

[44] Rabiner, L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, vol.77, no.2, pp.257-286, Febrero 1989. doi: 10.1109/5.18626.

[45] Rabiner, L., Juang, B.-H., Benesti, J., Sondhi, M.M., Huang, Y. (2008) Historical Perspective of the Field of ASR/NLU. En: Benesty, J., Sondhi, M.M., Huang, Y. (ed). Springer Handbook of Speech Processing. Springer. pp 583-583.

[46] Reddy, D. R. (1976) Speech Recognition by Machine: A Review. Proceedings of the IEEE, vol.64, no.4, pp.501-531, Abril 1976.

[47] Revista de filología española. (1915) University of Toronto Library. Alfabeto fonético. Tomo II. pp. 374-376. http://ia600307.us.archive.org/33/items/revistadefilolog02centuoft/revistadefilolog02centuoft.pdf. Visto el 30 de noviembre de 2011.

[48] Rioul, O., Vetterli, M. (1991) Wavelets and Signal Processing. Signal Processing Magazine, IEEE, vol.8, no.4, pp.14-38, Octubre 1991.

[49] Sharma, M., Mammone, R. (1995) Automatic Speech Segmentation Using Neural Tree Networks. Neural Networks for Signal Processing [1995] V. Proceedings of the 1995 IEEE Workshop, vol., no., pp.282-290, 31 Agosto – 2 Septiembre 1995. doi: 10.1109/NNSP.1995.514902.

[50] Spohrer, J.C., Brown, P.F., Roth, R. (1982) Automatic Labeling of Speech. Acoustics, Speech and Signal Processing, IEEE International Conference on ICASSP'82., vol.7, no., pp. 1641-1644, Mayo 1982. doi: 10.1109/ICASSP.1982.1171490.

[51] Suárez-Guerra, S. (2004) Una metodología para realizar trabajos de reconocimiento de voz. Serie Azul. No. 127. Instituto Politécnico Nacional. Centro de Investigación en Computación. 2004. pp. 1-4.

[52] Svendsen, T., Kvale, K. (1990) Automatic Alignment of Phonemic Labels with Continuous Speech. ICSLP-90, pp. 997-1000.

[53] The International Phonetic Association (2011) http://www.langsci.ucl.ac.uk/ipa/index.html. Consultado el 8 de marzo de 2012.

[54] Toledano, D. T., Gómez, L.A.H., Grande, L.V. (2003). Automatic Phonetic Segmentation. Speech and Audio Processing, IEEE Transactions on, vol.11, no.6, pp.617-625, Noviembre 2003. doi: 10.1109/TSA.2003.813579.

[55] Torre, D. (2001) Segmentación y Etiquetado Fonéticos Automáticos. Un enfoque Basado en Modelos Ocultos de Markov y Refinamiento Posterior de las Fronteras Fonéticas. Resumen de tesis doctoral. Universidad Politécnica de Madrid. http://coit.es/pub/ficheros/retevision_f665b978.pdf. Consultado el 8 de marzo de 2012.

[56] Villaseñor-Pineda, L., Montes-y-Gómez, M., Vaufreydaz, D., Serignat, J-F., (2003) Elaboración de un corpus balanceado para el cálculo de modelos acústicos usando la web. International Conference on Computing (CIC-2003), Ciudad de México, México, 2003.

[57] Waibel, A., Lee, K.F. (1990a) Readings in Speech Recognition. Morgan Kauffman Publishers, Inc. San Francisco California, USA, pp. 1-4.

[58] Waibel, A., Lee, K. F. (1990b) Readings in Speech Recognition. Morgan Kauffman Publishers, Inc. San Francisco California, USA, pp. 371-373.

[59] Wendt, C., Petropulu, A.P. (1996) Pitch Determination and Speech Segmentation Using the Discrete Wavelet Transform. Circuits and Systems, 1996. ISCAS'96., Connecting the World, 1996 IEEE International Symposium on, vol.2, no., pp.45-48 vol.2, 12-15 Mayo 1996.

[60] Wilson, S. (2003) WAVE PCM Soundfile Format. https://ccrma.stanford.edu/courses/422/projects/WaveFormat/. Consultado el 8 de marzo de 2012.

[61] Young, S. et al. (2005) The HTK Book. Cambridge University Engineering Department, pp. 2.

[62] Ziolko, B., Manandhar, S., Wilson, R.C. (2006) Phoneme Segmentation of Speech. Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, vol.4, no., pp.282-285, 0-0 0. doi: 10.1109/ICPR.2006.931.

[63] Zwicker, E. (1961) Subdivision of the Audible Frequency range into Critical Bands (Frequenzgruppen). The Journal of the Acoustical Society of America, vol.33, no.2, pp.248-248. doi: 10.1121/1.1908630.

ANEXO A. ALFABETOS FONÉTICOS

A.1 El alfabeto fonético del corpus TIMIT

El corpus TIMIT tuvo uno de los primeros alfabetos fonéticos. Los fonemas en ese alfabeto usaban el alfabeto inglés en minúsculas, a – z.

A.2 IPA

La asociación fonética internacional (del inglés, *International Phonetic Association*, IPA) es la principal organización representativa para fonetistas, así como la más antigua. El objetivo de IPA es promover el estudio científico de la fonética y las aplicaciones prácticas de dicha ciencia.
La IPA le provee a la comunidad académica de todo el mundo una notación estándar para la representación fonética de todos los lenguajes: el alfabeto fonético internacional (del inglés, *International Phonetic Alphabet*, también IPA). La última versión del alfabeto IPA se publicó en 2005.
En la tabla A.1 se muestran los fonemas del español utilizando los símbolos del IPA, así como las grafías, la cual fue tomada de [5].

Tabla A.1. Fonemas del español con símbolos IPA.

Fonema	Grafía	Ejemplos
/a/	a	c**a**v**a**
/b/	b,v	**v**aso, **b**ote, ca**v**a
/θ/	c,z	**c**ena, ca**z**a
/k/	c, qu, k	**c**asa, **qu**eso, **k**ilo
/ʧ/	ch	**ch**ico, mu**ch**acho
/d/	d	**d**ado
/e/	e	qu**e**so
/f/	f	**f**ama, ca**f**é
/g/	g, gu	**g**ama, **gu**isa, pa**g**a
/i/	i	gu**i**sa
/x/	j,g	pa**j**a, **g**itano
/l/	l	a**l**a, ma**l**
/ʎ/	ll	**ll**ave, ca**ll**e
/m/	m	**m**a**m**a
/n/	n	**n**a**n**a
/ɲ/	ñ	ca**ñ**a
/o/	o	b**o**te
/p/	p	**p**iedra, ca**p**a
/r/	r	pa**r**a, no**r**te
/r̄/	rr, r	pe**rr**o, **r**emo
/s/	s	**s**oy, do**s**
/t/	t	**t**apa, a**t**ar
/u/	u	d**u**lce
/ʝ/	y, hi	ma**y**o, la **hi**erba

A.3 Arpabet

Debido a que los símbolos del IPA no se podían escribir en una máquina de escribir o en un teclado convencional, la agencia de proyectos de investigación avanzada (ARPA, por sus siglas en inglés) auspició el desarrollo de un nuevo alfabeto fonético llamado ARPAbet. Este alfabeto cuenta con dos versiones, una que solamente usa símbolos de una sola letra, y la otra que usa solamente símbolos en mayúsculas, el cual necesita algunos símbolos de dos letras [10]. Un ejemplo del uso de ARPAbet es el diccionario de pronunciaciones del CMU [6].

A.4 Sampa

SAMPA es un alfabeto fonético que puede ser leído por máquinas. SAMPA es el acrónimo de Speech Assessment Methods Phonetic Alphabet, o alfabeto fonético para métodos de evaluación de habla. SAMPA fue desarrollado por un grupo internacional de fonetistas de 1987 a 1989 bajo el proyecto SPRITE 1541. Primero fue aplicado a lenguas europeas como el danés, el holandés, el inglés, francés, alemán e italiano (1989), luego se incorporaron el noruego y el sueco (1992), y subsecuentemente el griego, portugués y español (1993).

SAMPA y SAMPA extendido son una base internacional robusta para codificación estándar de notación fonética legible por máquinas. SAMPA consiste de un mapeo de símbolos del Alfabeto Fonético Internacional a código ASCII en el rango 33 a 127, los caracteres ASCII imprimibles de 7 bits. Existen guías para la transcripción de los lenguajes para los cuales se ha aplicado SAMPA que están asociados a la codificación (mapeo).

Existen otras propuestas para mapear el IPA a ASCII, pero SAMPA tiene la ventaja de que no es propuesto por un solo autor, sino resultado de la colaboración y consulta entre investigadores de habla en distintos países.

Una transcripción SAMPA está diseñada para analizarse sintácticamente de manera única. Al igual que ocurre con una cadena de IPA, una cadena SAMPA no necesita espacios entre símbolos sucesivos. En la tabla A.2 se muestra SAMPA para español.

Tabla A.2. SAMPA para español.

Plosivas		
Símbolo	Palabra	Transcripción
p	padre	“padre
b	vino	“bino
t	tomo	“tomo
d	donde	“donde
k	casa	“kasa
g	gata	“gata

Africadas		
Símbolo	Palabra	Transcripción
tS	mucho	“mutSo
jj	hielo	“jjelo

Fricativas		
Símbolo	Palabra	Transcripción
f	fácil	“faTil
B	cabra	“kaBra
T	cinco	“Tinko
D	nada	“naDa
s	sala	“sala
x	mujer	mu”xer
G	luego	“lweGo

Nasales		
Símbolo	Palabra	Transcripción
m	mismo	“mismo
n	nunca	“nunka
J	año	“aJo

Líquidas		
Símbolo	Palabra	Transcripción
l	lejos	“lexos
L	caballo	ka”baLo (o como jj)
r	puro	“puro
rr	torre	“torre

Semivocales		
Símbolo	Palabra	Transcripción
j	rei	rrej
	pie	pje
w	deuda	“dewDa
	muy	“mwi

Vocales		
Símbolo	Palabra	Transcripción
i	pico	“piko
e	pero	“pero
a	valle	“baLe
o	toro	“toro
u	duro	“duro

A.5. RFE

La revista de filología española (RFE) creó su propio alfabeto fonético en 1915. Fue diseñado específicamente para el español y por ello no puede compararse con otras lenguas. En la figura A.1 se reproduce el alfabeto fonético de la RFE [47].

Bilabiales.

b	esp. bondad.	boṇdáḍ
p	esp. padre..	páđre
m	esp. amar...	amár
m̥	and. mismo.	mim̥mo
b	esp. haba...	ába
b̥	and. las botas	la b̥ótah

Labiodentales.

m̥	esp. confuso.	kom̥fúso
f	esp. fácil....	fáθi̯l
v	esp. enf. vida	víđa

Interdentales.

ḍ	esp. enf. cruz / divina.....	krúz̧ ḍibína
ṭ	esp. hazte acá	áθṭe aká
ṇ	esp. onza....	óṇθa
z̧	esp. juzgar..	xu̯z̧gár
θ	esp. mozo...	móθo
đ	esp. rueda..	r̄wéđa
ɖ	esp. tomado.	tomáɖo
ḍ̥	esp. verdad.	bẹrđáḍ̥
l̥	esp. calzado.	kal̥θaɖo

Dentales.

d	esp. ducho..	dúĉo
t	esp. tomar..	tomáɹ
ṇ	esp. monte..	móṇte
z̧	esp. desde..	dézḍe
ṣ	esp. hasta...	áṣta
l̥	esp. falda...	fál̥da

Alveolares.

n	esp. mano..	mán̊o
n̥	and. asno...	án̥no
ŝ	ast. occid. / chobu(lobo)	ŝóbu
z	mex. los días	lo zíah
s	and. rosa...	r̄osa
ż	esp. rasgar..	r̄aż gáɹ
ṡ	esp. casa....	káṡa
l	esp. luna...	lúna
l̥	and. muslo..	mu̯l̥lo
r	esp. hora....	ǫ́ra
r̥	and. multitud	mu̯r̥titú
r̄	esp. carro...	kár̄o
ɹ	esp. color...	kolǫ́ɹ
i̯	mex. trigo..	ti̯ígo

Figura A.1.a. Primera parte del alfabeto fonético de la RFE [47].

Figura A.1.b. Segunda parte del alfabeto fonético de la RFE [47].

El alfabeto fonético de la RFE es ampliamente aceptado en el mundo hispánico y ha representado de manera eficiente los sonidos del español de México (Cuétara, 2004).

A.6 OGIbet

OGIbet está basado en el conjunto de etiquetas del TIMIT. Antes se usaba en el CSLU para realizar transcripciones fonéticas [35].

A.7 Worldbet

Worldbet es una codificación del alfabeto fonético internacional IPA a ASCII. Fue creado por James L. Hieronymus [26]. La mayoría de los símbolos se hicieron con apego al IPA para que su significado fuera obvio. Las variaciones alofónicas se pueden etiquetar en Worldbet usando diacríticos pegados al símbolo base. En la tabla A.3 se muestran los símbolos para el idioma español del alfabeto Worldbet.

Tabla A.3. Símbolos Worldbet para el español. (Tomada de [26]).

PLOSIVAS

IPA	Worldbet	Palabra	Palabra IPA	Palabra Worldbet
/p/	p	punto	/púnto/	p u n t o
/b/	b	baños	/báɲos/	b a n~ o s
/t/	t	tino	/tíno/	t i n o
/d/	d	donde	/dónde/	d o n d e
/k/	k	casa	/kása/	k a s a
/g/	g	ganga	/gáŋga/	g a N g a

FRICATIVAS

IPA	Worldbet	Palabra	Palabra IPA	Palabra Worldbet
/β/	V	haba	/áβa/	a V a
/f/	f	falda	/fálda/	f a l d a
/s/	s	casa	/kása/	k a s a
/z/	z	mismo	/mízmo/	m i z m o
/θ/	T	luces	/lúθes/	l u T e s
/ð/	D	dedo	/déðo/	d e D o
/x/	x	jamás	/xamás/	x a m a s
/ɣ/	G	lago	/láɣo/	l a G o

AFRICADAS

IPA	Worldbet	Palabra	Palabra IPA	Palabra Worldbet
/ʧ/	tS	chato	/ʧáto/	tS a t o
/dʒ/	dZ	un yugo	/undʒúɣo/	dZ u G o

NASALES

IPA	Worldbet	Palabra	Palabra IPA	Palabra Worldbet
/m/	m	mano	/máno/	m a n o
/n/	n	nada	/náða/	n a D a
/ɲ/	n~	baño	/báɲo/	b a n~ o
/ŋ/	N	banco	/báŋko/	b a N k o

SEMIVOCALES

IPA	Worldbet	Palabra	Palabra IPA	Palabra Worldbet
/l/	l	lado	/ládo/	l a d o
/ʎ/	L	pollo	/póʎo/	p o L o D
/ɾr/	r(	pero	péɾro	p e r(o
/r/	r	perro	/péro	p e r o
/j/	j	mayo	/májo/	m a j o
/w/	w	cuento	/kwénto/	k w e n t o

VOCALES

IPA	Worldbet	Palabra	Palabra IPA	Palabra Worldbet
/i/	i	piso	/píso/	p i s o
/e/	e	mesa	/mésa/	m e s a
/a/	a	caso	/káso/	k a s o
/o/	o	modo	/móðo/	m o D o
/u/	u	cura	/kúɾra/	k u r(a

A.8. Mexbet

Es un alfabeto fonético computacional creado para el español de México. Mexbet tomó como base a OGIbet y a Worldbet y fue creado por Esmeralda Uraga y Luis A. Pineda. Después fue mejorado por Cuétara [8], en el contexto del proyecto DIME de la UNAM.

ANEXO B. DICCIONARIO DE PRONUNCIACIONES

B.1. Características del diccionario para el corpus con todos los fonemas

Este diccionario está compuesto por 150 palabras que incluyen los 22 fonemas hablados en México. La pronunciación de dichas palabras es pronunciación mexicana y por lo tanto difiere de la pronunciación española. El diccionario contiene todos los fonemas hablados en México, esto significa que está completo, sin embargo no está balanceado.

El diccionario está escrito en el alfabeto Worldbet. Aunque Worldbet no tiene símbolos para todos los alófonos del español, eso no es importante en este trabajo de tesis puesto que este trabajo está a nivel de fonemas. Para saber más acerca de Worldbet consulte [26].

Se consideró a las pausas y los silencios como clases. Los silencios rodean las palabras; las pausas ocurren antes de un fonema plosivo.

En la tabla B.1 se muestra la cantidad de veces que aparece cada fonema en el corpus:

Tabla B.1. Fonemas en el corpus propuesto.

Fonema/alófono	cantidad	Fonema/alófono	cantidad	Fonema/alófono	cantidad
a	560	n	172	g	36
e	256	w	16	V	60
i	108	n~	28	D	77
o	397	tS	52	G	28
u	56	dZ	33	f	76
l	104	p	72	s	176
r(	132	t	116	x	36
r	36	k	128	sil	1196
j	68	b	35	pau	192
m	96	d	43		

B.2. Diccionario de pronunciaciones para el corpus con todos los fonemas

A continuación se muestra el diccionario de pronunciaciones del corpus con todos los fonemas.

Tabla B.2. Diccionario de pronunciaciones para el corpus fonéticamente completo creado.

Palabra	Transcripción	Palabra	Transcripción
abrir	a V r(i r(	feliz	f e l i s
adoro	a d o r(o	gama	g a m a
afable	a f a V l e	ganga	g a n g a
afecto	a f e pau k pau t o	garras	g a r a s
afinar	a f i n a r(	gato	g a pau t o
afortunado	a f o r(pau t u n a d o	general	x e n e r(a l
agarro	a g a r o	gente	x e n pau t e
ala	a l a	gitano	x i pau t a n o
alferez	a l f e r(e s	guerra	g e r a
alza	a l s a	guisa	g i s a
amigo	a m i g o	haba	a V a
anca	a n pau k a	hongo	o n g o
anfibio	a n f i V j o	hueso	w e s o
anfora	a n f o r(a	inyeccion	i n dZ e pau k s j o n
añoranza	a n~ o r(a n s a	isla	i s l a
arca	a r pau k a	israel	i s r a e l
atar	a pau t a r(	jamas	x a m a s
autobus	a u pau t o V u s	jamon	x a m o n
avaro	a V a r(o	jota	x o pau t a
bado	b a d o	jugar	x u g a r(
banco	b a n pau k o	la_hierba	l a j e r V a
baño	b a n~ o	lado	l a d o
billete	b i j e pau t e	lago	l a g o
bomba	b o m b a	lento	l e n pau t o
bote	b o pau t e	leña	l e n~ a
cafe	k a f e	llamar	dZ a m a r(
calle	k a j e	llave	dZ a V e
cambio	k a m b j o	mal	m a l
caña	k a n~ a	mama	m a m a
capa	k a pau p a	mano	m a n o
caray	k a r(a j	mayo	m a j o
casa	k a s a	mesa	m e s a
caso	k a s o	mismo	m i s m o
cava	k a V a	moda	m o d a
cepillo	s e pau p i j o	modo	m o d o
chato	tS a pau t o	nada	n a d a
chico	tS i pau k o	nana	n a n a
chinos	tS i n o s	niño	n i n~ o
chubasco	tS u V a s pau k o	norte	n o r(pau t e
coche	k o pau tS e	ñues	n~ w e s
codo	k o d o	oñate	o n~ a pau t e
colcha	k o l pau tS a	opcion	o p s j o n
colchon	k o l pau tS o n	paga	p a g a
concha	k o n pau tS a	paja	p a x a
consejo	k o n s e x o	papa	p a pau p a
coro	k o r(o	para	p a r(a
cosa	k o s a	pardo	p a r(d o

cuento	k w e n pau t o	pasa	p a s a
cuervo	k w e r(V o	pavo	p a V o
cura	k u r(a	pecho	p e pau tS o
dada	d a d a	pero	p e r(o
dato	d a pau t o	perro	p e r o
dedo	d e d o	piso	p i s o
dentro	d e n pau t r(o	pollo	p o j o
desde	d e s d e	punto	p u n pau t o
donde	d o n d e	quita	k i pau t a
dos	d o s	radio	r a d j o
dulce	d u l s e	raro	r a r(o
efectividad	e f e pau k pau t i V i D a D	remo	r e m o
eficiencia	e f i s j e n s j a	salchicha	s a l pau tS i pau tS a
efusion	e f u s j o n	seguido	s e g i d o
el_chico	e l pau tS i pau k o	sol	s o l
el_llama	e l dZ a m a	soy	s o j
el_llavero	e l dZ a V e r(o	tapa	t a pau p a
el_toro	e l pau t o r(o	tasa	t a s a
en_llamas	e n dZ a m a s	tengo	t e n g o
enfadarse	e n f a d a r(s e	tino	t i n o
enfermedad	e n f e r(m e d a d	toldo	t o l d o
enroscar	e n r o s pau k a r(	tres	t r(e s
es_mia	e s m i a	un_chico	u n pau tS i pau k o
estoy	e s pau t o j	un_farol	u n f a r(o l
falda	f a l d a	un_tomo	u n pau t o m o
fama	f a m a	un_yugo	u n dZ u g o
fase	f a s e	vaso	b a s o
favor	f a V o r(	yo	dZ o

B.3 Características del diccionario para el corpus de dígitos

En este corpus sólo se cuenta con 15 fonemas, más la clase de pau, que indica pausa. La tabla B.3 muestra la cantidad de veces que aparece cada fonema en el corpus.

Tabla B.3. Fonemas en el corpus de dígitos.

Fonema	Cantidad	Fonema	Cantidad
a	79	w	159
e	558	tS	80
i	80	t	238
o	558	k	159
u	80	d	80
r(	239	V	80
j	159	s	559
n	240	pau	318

B.4 Diccionario de pronunciaciones para el corpus de dígitos

A continuación se muestra el diccionario de pronunciaciones para el corpus de dígitos.

Tabla B.4. Transcripciones del corpus de dígitos en alfabeto worldbet.

Palabra	Transcripción
cero	s e r(o
uno	u n o
dos	d o s
tres	t r(e s
cuatro	k w a pau t r(o
cinco	s i n pau k o
seis	s e j s
siete	s j e pau t e
ocho	o pau tS o
nueve	n w e V e

ANEXO C. ARCHIVOS DE HTK

C.1 Introducción

En este anexo se proporcionan los archivos de configuración y los archivos de resultados importantes para los experimentos descritos en el capítulo 5. Note que no hay experimento 3 para el corpus con todos los fonemas, esto es así porque dicho experimento fue dedicado únicamente a la segmentación acústica y no se le dedicó un reconocimiento con HTK.

C.2 Experimentos con el corpus con todos los fonemas

Como se explicó en el capítulo 5 primero se realizaron experimentos con el corpus con todos los fonemas. Todos los experimentos relativos al corpus con todos los fonemas se realizaron con la misma configuración; lo único que cambia son los archivos de etiquetas utilizadas en el entrenamiento.

C.2.1 Experimento 1. Reconocimiento con etiquetas manuales

En este experimento se entrenó un reconocedor en HTK utilizando todos los archivos del corpus. El reconocimiento también se realizó sobre todos los archivos del corpus. Las etiquetas utilizadas fueron generadas manualmente.

Directorios de trabajo

Se crearon los directorios *data* y *model*. A su vez, el directorio data tiene los siguientes subdirectorios:

- *train/label_lab*: contiene los archivos de etiquetas de entrenamiento, los cuales tienen información acerca de las fronteras entre fonemas.
- *train/label_lab_palabras*: contiene las transcripciones ortográficas del corpus. Por ejemplo, en el corpus con todos los fonemas, el contenido del archivo *abrir1.lab* sería:
  sil
  abrir
  sil
- *train/revueltos*: contiene revueltos los archivos de etiquetas de entrenamiento y los archivos de vectores MFCC.

Bajo el directorio model se crearon los siguientes subdirectorios:

- hmm0: Contiene los modelos HMMs para cada fonema recién inicializados.
- hmm0flat: Contiene el archivo vFloors generado por HCompv.
- hmm1: Contiene los modelos después de la primera reestimación embebida con HERest.
- hmm2: Contiene los modelos después de la segunda reestimación embebida con HERest.
- hmm3, hmm4, hmm5, hmm6, hmm7, hmm8, hmm9, hmm10: Contienen respectivamente los modelos después de su respectiva iteración de reestimación embebida con HERest.
- proto: Contiene la configuración básica de cada HMM; tal como el número de estados y los saltos permitidos entre estados, el tamaño de los vectores, el número de mixturas, etc.

Configuración para HCopy

Archivo de configuración *analysis_conf.conf*:

```
SOURCEFORMAT = WAV
TARGETKIND = MFCC_0_D_A
WINDOWSIZE = 200000
TARGETRATE =  10000
DELTAWINDOW = 2
ACCELWINDOW = 2
SAVECOMPRESSED = F
SAVEWITHCRC = F
NUMCEPS = 14
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 24
CEPLIFTER = 22
ENORMALISE = F
```

El archivo lista_archivos.scp, indica a cuáles archivos wav se les calcularán los MFCC. El archivo consiste en dos columnas: la primera columna indica el archivo wav a abrir, la segunda columna indica el archivo .mfc que se fabricará.

A continuación se muestra un fragmento del contenido del archivo de lista:

```
C:\Users\remington\Documents\MATLAB\corpora_tesis\cristian_wav\abrir1.wav
C:\pruebas_htk14\data\train\revueltos\abrir1.mfc
C:\Users\remington\Documents\MATLAB\corpora_tesis\cristian_wav\abrir2.wav
C:\pruebas_htk14\data\train\revueltos\abrir2.mfc
C:\Users\remington\Documents\MATLAB\corpora_tesis\cristian_wav\abrir3.wav
C:\pruebas_htk14\data\train\revueltos\abrir3.mfc
C:\Users\remington\Documents\MATLAB\corpora_tesis\cristian_wav\abrir4.wav
C:\pruebas_htk14\data\train\revueltos\abrir4.mfc
C:\Users\remington\Documents\MATLAB\corpora_tesis\cristian_wav\adoro1.wav
C:\pruebas_htk14\data\train\revueltos\adoro1.mfc
```

Y así sucesivamente hasta incluir a todas las palabras del corpus.

La llamada a HCopy es como sigue:

```
HCopy -A -D -C analysis_conf.conf -S lista_archivos.scp
```

Donde la opción –A significa imprimir en pantalla los argumentos del comando, -D significa desplegar las opciones de configuración, –C significa que se proporcionará un archivo de configuración, y –S significa que se proporcionará un script.

Gramática

Como se mostró en el capítulo 5, la gramática utilizada para la tarea es la siguiente.

```
$grupo1 = abrir | adoro | afable | afecto | afinar | afortunado | agarro | ala |
alferez | alza;
$grupo2 = amigo | anca | anfibio | anfora | añoranza | arca | atar | autobus| avaro
| bado;
$grupo3 = banco | baño | billete | bomba | bote | cafe | calle | cambio | caña |
capa;
$grupo4 = caray | casa | caso | cava | cepillo | chato | chico | chinos | chubasco |
coche;
```

```
$grupo5 = codo | colcha | colchon | concha | consejo | coro | cosa | cuento | cuervo
| cura;
$grupo6 = dada | dato | dedo | dentro | desde | donde | dos | dulce | efectividad |
eficiencia;
$grupo7 = efusion | el_chico | el_llama | el_llavero | el_toro | en_llamas |
enfadarse | enfermedad | enroscar | es_mia;
$grupo8 = estoy | falda | fama | fase | favor | feliz | gama | ganga | garras |
gato;
$grupo9 = general | gente | gitano | guerra | guisa | haba | hongo | hueso |
inyeccion | isla;
$grupo10 = israel | jamas | jamon | jota | jugar | la_hierba | lado | lago | lento |
leña;
$grupo11 = llamar | llave | mal | mama | mano | mayo | mesa | mismo | moda | modo;
$grupo12 = nada | nana | niño | norte | ñues | oñate | opcion | paga | paja | papa;
$grupo13 = para | pardo | pasa | pavo | pecho | pero | perro | piso | pollo | punto;
$grupo14 = quita | radio | raro | remo | salchicha | seguido | sol | soy | tapa |
tasa;
$grupo15 = tengo | tino | toldo | tres | un_chico | un_farol | un_tomo | un_yugo |
vaso | yo;
([sil] (
($grupo1)|($grupo2)|($grupo3)|($grupo4)|($grupo5)|($grupo6)|($grupo7)|($grupo8)|($gr
upo9)|($grupo10)|($grupo11)|($grupo12)|($grupo13)|($grupo14)|($grupo15) )  [sil])
```

Donde las barras verticales denotan alternativas, los corchetes denotan ítems opcionales, los paréntesis contienen expresiones regulares y los símbolos de dólar denotan variables.
Como se puede observar, la gramática está hecha para detectar una palabra que posiblemente está rodeada por silencios. Dicha gramática se guardó en el archivo gram.txt.

Compilación de la gramática

La gramática se compiló a una red (con extensión SLF por el inglés Standard Lattice Format) utilizando la herramienta HParse. La red ya compilada se guardó en el archivo wdnet.slf.

```
HPars
e gram.txt wdnet.slf
```

Diccionario

El diccionario se guardó bajo el archivo *diccionario.txt*. El contenido del diccionario de pronunciaciones se muestra en el anexo B.

Transcripciones ortográficas

Se generó un MLF bajo el nombre *etiquetas_palabras.mlf* con las transcripciones ortográficas correspondientes a cada palabra en el corpus. A continuación se presentan las transcripciones ortográficas de las primeras palabras del corpus.

```
#!MLF!#
"C:\pruebas_htk14\data\train\label_lab\abrir1.lab"
abrir
.
"C:\pruebas_htk14\data\train\label_lab\abrir2.lab"
abrir
.
"C:\pruebas_htk14\data\train\label_lab\abrir3.lab"
```

```
abrir
.
"C:\pruebas_htk14\data\train\label_lab\abrir4.lab"
abrir
.
"C:\pruebas_htk14\data\train\label_lab\adoro1.lab"
adoro
.
```

Y así sucesivamente para todas las palabras del corpus.

Expansión de las transcripciones

Las transcripciones ortográficas se expandieron utilizando la herramienta HLEd. Expandir las transcripciones significa sustituir cada transcripción ortográfica por la secuencia de fonemas que le corresponde.

```
HLEd -d diccionario.txt -i etiquetas_fonemas.mlf mkfonemas0.led
etiquetas_palabras.mlf
```

Donde:
El archivo diccionario.txt es el diccionario de pronunciaciones. El archivo mkfonemas0.led es un archivo que contiene comandos para HLed; en este caso sólo se agregó el comando EX (para expandir). El archivo etiquetas_palabras.mlf es el archivo que contiene las transcripciones ortográficas de cada palabra. Finalmente, el archivo etiquetas_fonemas.mlf es el archivo que HLEd va a generar.

El contenido del archivo mkfonemas0.led es simplemente:

```
EX
```

Las primeras líneas del contenido del archivo etiquetas_fonemas.mlf son las siguientes:

```
#!MLF!#
"C:/pruebas_htk14/data/train/label_lab/abrir1.lab"
a
V
r(
i
r(
.
"C:/pruebas_htk14/data/train/label_lab/abrir2.lab"
a
V
r(
i
r(
.
"C:/pruebas_htk14/data/train/label_lab/abrir3.lab"
a
V
r(
i
r(
.
```

Y así sucesivamente para todas las palabras del corpus.

Creación de los protos

En el directorio model/proto se crearon protos para cada fonema. Los archivos proto indican la arquitectura de un HMM. Todos los protos tienen la misma configuración; por ejemplo, el archivo proto para la /a/ es el siguiente:

```
~o <VecSize> 45 <MFCC_0_D_A>
~h "a"
<BeginHMM>
 <NumStates> 5
 <State> 2
  <Mean> 45
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
  <Variance> 45
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
 <State> 3
  <Mean> 45
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
  <Variance> 45
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
 <State> 4
  <Mean> 45
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
  <Variance> 45
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
 <TransP> 5
   0.0 1.0 0.0 0.0 0.0
   0.0 0.3 0.3 0.4 0.0
   0.0 0.0 0.5 0.5 0.0
   0.0 0.0 0.0 0.5 0.5
   0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

Inicialización de los HMM

Este paso es muy importante, ya que aquí es donde se toma en cuenta la información de fronteras entre fonemas la cual está presente en los archivos de etiquetas de entrenamiento.

Cada modelo se inicializó utilizando HInit. Por ejemplo, el modelo de la /a/ se inicializó como sigue:

```
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H
model/proto/hmm_a.txt -l a -L data/train/label_lab a
```

Donde *train_list.txt* es un archivo que indica cuáles archivos MFCC se utilizarán para la inicialización; *model/proto/hmm_a.txt* es el archivo proto que se va a inicializar; la opción –l a indica que se deben de extraer los segmentos etiquetados con “a”; la opción –L *data/train/label_lab* indica que los archivos de

etiquetas se encuentran en dicha dirección; la última "a" representa el modelo que se va a inicializar; finalmente la opción –M *model/hmm0* indica que el modelo inicializado se colocará en dicho directorio.

El contenido del archivo train_list.txt es como sigue:

```
C:\pruebas_htk14\data\train\revueltos\abrir1.mfc
C:\pruebas_htk14\data\train\revueltos\abrir2.mfc
C:\pruebas_htk14\data\train\revueltos\abrir3.mfc
C:\pruebas_htk14\data\train\revueltos\abrir4.mfc
C:\pruebas_htk14\data\train\revueltos\adoro1.mfc
C:\pruebas_htk14\data\train\revueltos\adoro2.mfc
C:\pruebas_htk14\data\train\revueltos\adoro3.mfc
C:\pruebas_htk14\data\train\revueltos\adoro4.mfc
```

Y así sucesivamente para cada palabra en el corpus.

Todos los modelos de los fonemas como se muestra a continuación.

```
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_a.txt -l a -L
data/train/label_lab a
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_e.txt -l e -L
data/train/label_lab e
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_i.txt -l i -L
data/train/label_lab i
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_o.txt -l o -L
data/train/label_lab o
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_u.txt -l u -L
data/train/label_lab u
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_l.txt -l l -L
data/train/label_lab l
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_ere.txt -l r( -L
data/train/label_lab r(
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_r.txt -l r -L
data/train/label_lab r
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_j.txt -l j -L
data/train/label_lab j
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_m.txt -l m -L
data/train/label_lab m
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_n.txt -l n -L
data/train/label_lab n
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_w.txt -l w -L
data/train/label_lab w
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_enie.txt -l n~ -
L data/train/label_lab n~
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_tS.txt -l tS -L
data/train/label_lab tS
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_dZ.txt -l dZ -L
data/train/label_lab dZ
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_p.txt -l p -L
data/train/label_lab p
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_t.txt -l t -L
data/train/label_lab t
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_k.txt -l k -L
data/train/label_lab k
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_b.txt -l b -L
data/train/label_lab b
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_d.txt -l d -L
data/train/label_lab d
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_g.txt -l g -L
data/train/label_lab g
```

```
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_V.txt -l V -L
data/train/label_lab V
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_D_.txt -l D -L
data/train/label_lab D
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_G_.txt -l G -L
data/train/label_lab G
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_f.txt -l f -L
data/train/label_lab f
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_s.txt -l s -L
data/train/label_lab s
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_x.txt -l x -L
data/train/label_lab x
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_sil.txt -l sil -
L data/train/label_lab sil
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_pau.txt -l pau -
L data/train/label_lab pau
```

Obtención del archivo vFloors

El archivo vFloors se obtuvo de la siguiente manera:

```
HCompv -C configuracion2.conf -f 0.01 -m -S train_list.txt -M
model/hmm0flat model/proto/proto.txt
```

El contenido del archivo vFloors es:

```
~v varFloor1
<Variance> 45
 5.508769e-001 4.138514e-001 4.851382e-001 5.398993e-001 5.398020e-001 4.345673e-001
3.744746e-001 4.196871e-001 4.061993e-001 3.204137e-001 3.622048e-001 5.465696e-001
2.031601e-001 1.678533e-001 7.835667e-001 9.645393e-004 1.411108e-003 1.767580e-003
2.584259e-003 3.085497e-003 3.588757e-003 3.983074e-003 4.276903e-003 4.318450e-003
4.312104e-003 4.295133e-003 3.884995e-003 3.270525e-003 2.734666e-003 7.599809e-004
6.244714e-005 1.313806e-004 1.796301e-004 2.620457e-004 3.166813e-004 3.906144e-004
4.489699e-004 4.881570e-004 4.942003e-004 5.066756e-004 5.059033e-004 4.450319e-004
3.912364e-004 3.305101e-004 4.405646e-005
```

Este contenido se copia para construir el archivo *macros.hmm*.

El archivo macros.hmm

Este archivo se construye a partir del archivo *vFloors*; así como del tamaño y tipo de los vectores de características. El contenido del archivo *macros.hmm* es:

```
~o <VecSize> 45
   <MFCC_0_D_A>
~v varFloor1
<Variance> 45
 5.508769e-001 4.138514e-001 4.851382e-001 5.398993e-001 5.398020e-001 4.345673e-001
3.744746e-001 4.196871e-001 4.061993e-001 3.204137e-001 3.622048e-001 5.465696e-001
2.031601e-001 1.678533e-001 7.835667e-001 9.645393e-004 1.411108e-003 1.767580e-003
2.584259e-003 3.085497e-003 3.588757e-003 3.983074e-003 4.276903e-003 4.318450e-003
4.312104e-003 4.295133e-003 3.884995e-003 3.270525e-003 2.734666e-003 7.599809e-004
6.244714e-005 1.313806e-004 1.796301e-004 2.620457e-004 3.166813e-004 3.906144e-004
4.489699e-004 4.881570e-004 4.942003e-004 5.066756e-004 5.059033e-004 4.450319e-004
3.912364e-004 3.305101e-004 4.405646e-005
```

El archivo hmmdefs.mmf

Este archivo se construye copiando los modelos inicializados de cada fonema en un solo archivo.

Reestimación embebida

La llamada a HERest es como sigue:

```
HERest -C configuracion2.conf -I etiquetas_fonemas.mlf -t 250.0 150.0 1000.0 -S train_list.txt -H model/hmm0/macros.hmm -H model/hmm0/hmmdefs.mmf -M model/hmm1 hmm_list.txt
```

El contenido de configuracion2.conf es el siguiente:

```
#SOURCEFORMAT = WAV
TARGETKIND = MFCC_0_D_A
WINDOWSIZE = 200000
TARGETRATE =  10000
DELTAWINDOW = 2
ACCELWINDOW = 2
SAVECOMPRESSED = F
SAVEWITHCRC = F
NUMCEPS = 14
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 24
CEPLIFTER = 22
ENORMALISE = F
```

El contenido del archivo hmm_list.txt es simplemente una enumeración de los modelos HMM que se están utilizando:

```
a
e
i
o
u
l
r(
r
j
m
n
w
n~
tS
dZ
p
t
k
b
d
```

```
g
V
D
G
f
s
x
pau
sil
```

El contenido de etiquetas_fonemas.mlf ya se había mostrado; así como también el contenido de train_list.txt. A continuación se muestra parte del contenido del archivo hmmdefs.mmf para el fonema /a/.

```
~h "a"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 45
 1.020857e+000   -1.811225e+001   -4.782639e+000   -6.346516e+000   -3.340207e+000   -
7.887466e+000 1.602437e+000 6.427066e+000 5.747139e+000 -7.779238e+000 4.302804e+000
-1.688215e+001 4.207467e+000 -4.606221e-001 7.299561e+001 1.200328e-002 -1.746320e-
002  -3.593526e-002  5.374225e-003  -3.419835e-003  5.927755e-002  4.013520e-002  -
4.496258e-002 2.741523e-002 4.956562e-003 -1.490560e-002 -2.761397e-002 -4.059598e-
003   4.169252e-002   -5.627866e-003   -1.009803e-003   9.187628e-003   1.740936e-003
5.856110e-003 2.321525e-003 -1.106695e-003 2.125554e-003 -6.124188e-003 -3.065527e-
003  6.637390e-003  -2.705683e-003  8.042196e-003  -3.850854e-003  1.097723e-003  -
7.365100e-003
<VARIANCE> 45
 4.537814e+000 1.447706e+001 1.622793e+001 1.697465e+001 1.775537e+001 2.648637e+001
2.519330e+001 2.715384e+001 2.757578e+001 3.150501e+001 2.047670e+001 2.651976e+001
1.975646e+001 1.131631e+001 1.018461e+001 2.195455e-002 6.832687e-002 7.566674e-002
9.969140e-002 1.021450e-001 1.307779e-001 1.827914e-001 2.444821e-001 1.972301e-001
1.835336e-001 1.698828e-001 1.574582e-001 1.198166e-001 1.003602e-001 2.475741e-002
2.835614e-003 9.748815e-003 8.632273e-003 1.318413e-002 1.222356e-002 1.614392e-002
2.562832e-002 4.108019e-002 3.060767e-002 2.536621e-002 2.520193e-002 2.093872e-002
1.723841e-002 1.382523e-002 3.443014e-003
<GCONST> 2.842676e+001
<STATE> 3
<MEAN> 45
 2.096746e+000   -1.720603e+001   -8.498292e+000   -8.759493e+000   -1.157205e-001   -
1.617348e+000    4.385505e+000    -2.882290e+000    5.748392e+000    -7.713477e+000
7.285918e+000  -1.703294e+001  -1.097355e+000 4.699176e+000 7.216580e+001 5.915420e-
003  1.996144e-002  4.948774e-003  3.229174e-002  -2.917748e-002  -1.393829e-002  -
1.812068e-003 1.280568e-002 2.128827e-003 5.145522e-002 4.675608e-003 3.130546e-003
1.674370e-002 -1.409398e-002 -1.872918e-002 -3.381960e-004 6.722989e-003 2.633721e-
003  4.687508e-003  -1.133833e-003  -2.228116e-003  -5.404460e-005  1.593324e-003  -
2.826434e-003 4.866498e-003 -3.935270e-003 4.144058e-003 5.084737e-004 -5.242708e-
003 -6.174159e-003
<VARIANCE> 45
 5.038136e+000 1.154818e+001 1.149312e+001 3.088797e+001 1.917325e+001 2.553229e+001
3.019068e+001 2.472603e+001 3.156147e+001 2.133231e+001 2.709854e+001 2.496698e+001
2.804330e+001 1.335632e+001 1.197203e+001 2.393922e-002 6.687620e-002 7.075740e-002
1.021194e-001 1.204866e-001 1.381836e-001 1.994760e-001 2.019829e-001 1.723387e-001
1.860183e-001 1.785470e-001 1.673900e-001 1.185223e-001 1.266425e-001 2.728775e-002
3.028986e-003 9.350548e-003 8.584443e-003 1.336927e-002 1.491262e-002 1.847726e-002
2.770467e-002 2.977644e-002 2.478941e-002 2.601920e-002 2.513534e-002 2.335554e-002
1.593838e-002 1.974622e-002 3.751123e-003
<GCONST> 3.018373e+001
<STATE> 4
<MEAN> 45
```

```
 5.010641e-001   -1.098046e+001   -6.892002e+000   -5.419422e-001   1.419342e-001   -
4.416693e+000 6.986831e+000 7.770621e-001 4.838673e+000 -3.449009e+000 2.239552e+000
-1.046515e+001  1.518270e+000  1.817155e+000  6.391451e+001  -1.333716e-001  1.318690e-
001   6.847615e-002   1.428459e-001   1.216992e-001   3.240332e-002   4.448783e-002   -
4.876998e-002  -2.894922e-002  7.873203e-002  -9.164395e-002  1.403642e-001  1.767345e-
002  -1.542657e-002  -1.798100e-001  -8.633100e-003  -4.421419e-003  2.393473e-003  -
5.499103e-003 1.643308e-003 2.057158e-004 -8.826418e-003 -3.462042e-003 -6.226441e-
004   -4.486497e-003   4.880437e-003   6.245162e-003   3.425648e-003   1.598527e-003
1.705055e-003
<VARIANCE> 45
 9.063374e+000 1.208742e+001 2.028400e+001 3.065604e+001 3.096881e+001 3.450745e+001
3.191033e+001  3.137528e+001  3.091721e+001  2.733816e+001  2.496057e+001  2.839727e+001
2.275414e+001  1.395488e+001  1.468850e+001  9.380718e-002  1.686242e-001  1.844504e-001
2.998023e-001  3.226086e-001  3.812512e-001  4.655201e-001  5.192761e-001  4.418124e-001
4.301578e-001  4.094017e-001  3.768792e-001  3.055066e-001  2.619110e-001  5.814245e-002
6.425765e-003  1.532156e-002  1.894356e-002  2.904459e-002  3.205073e-002  4.166937e-002
5.576909e-002  6.369820e-002  5.433786e-002  5.081834e-002  4.800168e-002  4.409363e-002
3.672944e-002 3.390519e-002 4.827612e-003
<GCONST> 5.694909e+001
<TRANSP> 5
 0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
 0.000000e+000 9.720535e-001 1.620971e-002 1.173680e-002 0.000000e+000
 0.000000e+000 0.000000e+000 9.863797e-001 1.362020e-002 0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 9.681203e-001 3.187971e-002
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>
```

Cada fonema tiene su sección en el archivo hmmdefs.mmf, tal como la que se acaba de mostrar.

Reconocimiento

La llamada a HVite es como sigue:

```
HVite -o SWT -H model/hmm10/macros.hmm -H model/hmm10/hmmdefs.mmf -
S    testlist.txt   -i   recout.mlf   -w   wdnet.slf   -p   0.0   -s   5.0
diccionario.txt hmm_list.txt
```

Donde *model/hmm10/macros.hmm* es la última versión del archivo macros, producida en la última iteración de la reestimación embebida; *model/hmm10/hmmdefs.mmf* es la última versión de los modelos, ya reestimados; *testlist.txt* es simplemente una lista de archivos MFCC para probar el reconocedor; dado que los archivos de prueba son los mismos que los archivos de entrenamiento, el contenido del archivo *testlist.txt* es exactamente el mismo que el del archivo *train_list*.txt mostrado arriba. El archivo *wdnet.slf* es la gramática compilada; el *diccionario.txt* es el mismo que ya se había utilizado; *hmm_list.txt* es una lista de los modelos HMM utilizados. El archivo de salida *recout.mlf* contiene los resultados del reconocimiento.

El contenido del archivo de reconocimiento es:

```
#!MLF!#
"C:/pruebas_htk14/data/train/revueltos/abrir1.rec"
sil
abrir
sil
.
"C:/pruebas_htk14/data/train/revueltos/abrir2.rec"
sil
abrir
sil
.
"C:/pruebas_htk14/data/train/revueltos/abrir3.rec"
sil
afable
sil
.
"C:/pruebas_htk14/data/train/revueltos/abrir4.rec"
sil
abrir
sil
.
"C:/pruebas_htk14/data/train/revueltos/adoro1.rec"
sil
adoro
sil
.
```

Y así sucesivamente para todas las palabras reconocidas.

Evaluación

La llamada a HResults es como sigue:

```
HResults -p -A -D -T 1 -t -I etiquetas_palabras.mlf
label_list_palabras.txt recout.mlf > results.txt
```

El archivo de resultados es el siguiente:

```
No HTK Configuration Parameters Set
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/abrir3.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/abrir3.rec
 LAB: sil abrir  sil
 REC: sil afable sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/ala1.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/ala1.rec
 LAB: sil ala   sil
 REC: sil falda sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/ala3.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/ala3.rec
 LAB: sil ala   sil
 REC: sil falda sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/arca1.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/arca1.rec
 LAB: sil arca sil
 REC: sil anca sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/caray2.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/caray2.rec
 LAB: sil caray sil
 REC: sil atar  sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/caray3.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/caray3.rec
```

```
 LAB: sil caray sil
 REC: sil calle sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/caray4.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/caray4.rec
 LAB: sil caray sil
 REC: sil calle sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/chinos4.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/chinos4.rec
 LAB: sil chinos sil
 REC: sil chinos
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/haba1.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/haba1.rec
 LAB: sil haba sil
 REC: sil cava sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/haba4.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/haba4.rec
 LAB: sil haba sil
 REC: sil cava sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/hongo2.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/hongo2.rec
 LAB: sil hongo sil
 REC: sil punto sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/lado2.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/lado2.rec
 LAB: sil lado       sil
 REC: sil afortunado sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/llamar1.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/llamar1.rec
 LAB: sil llamar sil
 REC: sil llamar
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/mano1.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/mano1.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/mano2.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/mano2.rec
 LAB: sil mano  sil
 REC: sil mismo sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/mano3.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/mano3.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/mano4.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/mano4.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/para3.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/para3.rec
 LAB: sil para sil
 REC: sil para
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/perro2.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/perro2.rec
 LAB: sil perro sil
 REC: sil pardo sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/quita2.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/quita2.rec
 LAB: sil quita  sil
 REC: sil gitano sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/raro2.lab vs
C:/pruebas_htk14/data/train/label_lab_palabras/raro2.rec
 LAB: sil raro  sil
 REC: sil radio sil
```

```
Aligned  transcription:  C:/pruebas_htk14/data/train/label_lab_palabras/sol1.lab  vs
C:/pruebas_htk14/data/train/label_lab_palabras/sol1.rec
 LAB: sil sol sil
 REC: sil sol
Aligned  transcription:  C:/pruebas_htk14/data/train/label_lab_palabras/tino1.lab  vs
C:/pruebas_htk14/data/train/label_lab_palabras/tino1.rec
 LAB: sil tino  sil
 REC: sil mismo sil
Aligned transcription: C:/pruebas_htk14/data/train/label_lab_palabras/toldo4.lab  vs
C:/pruebas_htk14/data/train/label_lab_palabras/toldo4.rec
 LAB: sil toldo sil
 REC: sil codo  sil
Aligned  transcription:  C:/pruebas_htk14/data/train/label_lab_palabras/un_tomo1.lab
vs C:/pruebas_htk14/data/train/label_lab_palabras/un_tomo1.rec
 LAB: sil un_tomo   sil
 REC: sil inyeccion sil
====================== HTK Results Analysis =======================
  Date: Mon Mar 12 17:25:27 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=95.83 [H=575, S=25, N=600]
WORD: %Corr=98.61, Acc=98.61 [H=1775, D=4, S=21, I=0, N=1800]
```

En la primera sección se pueden observar aquellas palabras que fueron confundidas por el reconocedor. En la última sección se muestra el porcentaje de reconocimiento, que en este caso resultó ser de 98.61%.

C.2.2 Experimento 2. Reconocimiento con etiquetas con alineación forzada

El procedimiento fue exactamente igual que en la sección C.2.1, con la única diferencia que esta vez se utilizaron etiquetas generadas con alineación forzada en lugar de utilizar etiquetas manuales. Sólo mostraremos los resultados de la evaluación del reconocimiento.

El archivo producido por HResults es el siguiente:

```
No HTK Configuration Parameters Set
Aligned  transcription:  C:/pruebas_htk15/data/train/label_lab_palabras/ala4.lab  vs
C:/pruebas_htk15/data/train/label_lab_palabras/ala4.rec
 LAB: sil ala   sil
 REC: sil falda sil
Aligned  transcription:  C:/pruebas_htk15/data/train/label_lab_palabras/alza2.lab  vs
C:/pruebas_htk15/data/train/label_lab_palabras/alza2.rec
 LAB: sil alza   sil
 REC: sil anfora sil
Aligned  transcription:  C:/pruebas_htk15/data/train/label_lab_palabras/arca1.lab  vs
C:/pruebas_htk15/data/train/label_lab_palabras/arca1.rec
 LAB: sil arca sil
 REC: sil anca sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/caray3.lab  vs
C:/pruebas_htk15/data/train/label_lab_palabras/caray3.rec
 LAB: sil caray sil
 REC: sil calle sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/chinos4.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/chinos4.rec
 LAB: sil chinos sil
 REC: sil chinos
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/hongo2.lab  vs
C:/pruebas_htk15/data/train/label_lab_palabras/hongo2.rec
 LAB: sil hongo   sil
```

```
 REC: sil un_yugo sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/la_hierba4.lab
vs C:/pruebas_htk15/data/train/label_lab_palabras/la_hierba4.rec
 LAB: sil la_hierba sil
 REC: sil la_hierba
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/lado2.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/lado2.rec
 LAB: sil lado       sil
 REC: sil afortunado sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/llamar1.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/llamar1.rec
 LAB: sil llamar sil
 REC: sil llamar
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/mano1.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/mano1.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/mano2.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/mano2.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/mano3.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/mano3.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/mano4.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/mano4.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/nada2.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/nada2.rec
 LAB: sil nada sil
 REC:     moda sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/nana2.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/nana2.rec
 LAB: sil nana sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/para3.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/para3.rec
 LAB: sil para sil
 REC: sil para
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/pasa4.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/pasa4.rec
 LAB: sil pasa sil
 REC: sil pasa
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/pecho3.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/pecho3.rec
 LAB: sil pecho    sil
 REC: sil el_chico sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/piso4.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/piso4.rec
 LAB: sil piso  sil
 REC: sil mismo sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/raro2.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/raro2.rec
 LAB: sil raro  sil
 REC: sil radio sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/sol1.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/sol1.rec
 LAB: sil sol sil
 REC: sil sol
```

```
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/tapa3.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/tapa3.rec
 LAB: sil tapa sil
 REC: sil papa sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/tapa4.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/tapa4.rec
 LAB: sil tapa sil
 REC: sil papa sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/tengo2.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/tengo2.rec
 LAB: sil tengo sil
 REC: sil tengo
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/toldo4.lab vs
C:/pruebas_htk15/data/train/label_lab_palabras/toldo4.rec
 LAB: sil toldo sil
 REC: sil codo  sil
Aligned transcription: C:/pruebas_htk15/data/train/label_lab_palabras/un_tomo1.lab
vs C:/pruebas_htk15/data/train/label_lab_palabras/un_tomo1.rec
 LAB: sil un_tomo   sil
 REC: sil inyeccion sil
====================== HTK Results Analysis =======================
  Date: Tue Mar 06 22:31:16 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=95.67 [H=574, S=26, N=600]
WORD: %Corr=98.50, Acc=98.50 [H=1773, D=8, S=19, I=0, N=1800]
```

Como se puede observar, el porcentaje de reconocimiento es de 98.50%.

C.2.3 Experimento 4. Reconocimiento con etiquetas con alineación forzada y segmentación acústica

El procedimiento fue exactamente igual que en la sección C.2.1, con la única diferencia que esta vez se utilizaron etiquetas generadas con alineación forzada y segmentación acústica, sin tomar en cuenta los tipos de fronteras. Sólo mostraremos los resultados de la evaluación del reconocimiento.

El archivo producido por HResults es el siguiente:

```
HResults -p -A -D -T 1 -t -I etiquetas_palabras.mlf label_list_palabras.txt
recout.mlf
No HTK Configuration Parameters Set
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/ala2.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/ala2.rec
 LAB: sil ala   sil
 REC: sil falda sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/ala3.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/ala3.rec
 LAB: sil ala   sil
 REC: sil falda sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/ala4.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/ala4.rec
 LAB: sil ala   sil
 REC: sil falda sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/arca1.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/arca1.rec
 LAB: sil arca sil
 REC: sil anca sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/chico4.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/chico4.rec
 LAB: sil chico    sil
```

```
 REC: sil el_chico sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/chinos4.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/chinos4.rec
 LAB: sil chinos sil
 REC: sil chinos
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/haba1.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/haba1.rec
 LAB: sil haba sil
 REC: sil cava sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/haba4.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/haba4.rec
 LAB: sil haba sil
 REC: sil cava sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/hongo1.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/hongo1.rec
 LAB: sil hongo   sil
 REC: sil consejo sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/hongo2.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/hongo2.rec
 LAB: sil hongo sil
 REC: sil mano  sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/lado2.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/lado2.rec
 LAB: sil lado       sil
 REC: sil afortunado sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/llamar1.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/llamar1.rec
 LAB: sil llamar sil
 REC: sil llamar
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/mano1.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/mano1.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/mano2.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/mano2.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/mano3.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/mano3.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/mano4.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/mano4.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/para3.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/para3.rec
 LAB: sil para sil
 REC: sil para
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/raro2.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/raro2.rec
 LAB: sil raro  sil
 REC: sil radio sil
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/sol1.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/sol1.rec
 LAB: sil sol sil
 REC: sil sol
Aligned transcription: C:/pruebas_htk16/data/train/label_lab_palabras/tapa1.lab vs
C:/pruebas_htk16/data/train/label_lab_palabras/tapa1.rec
 LAB: sil tapa sil
 REC: sil papa sil
```

```
Aligned  transcription:  C:/pruebas_htk16/data/train/label_lab_palabras/tapa2.lab  vs
C:/pruebas_htk16/data/train/label_lab_palabras/tapa2.rec
 LAB: sil tapa sil
 REC: sil papa sil
Aligned  transcription:  C:/pruebas_htk16/data/train/label_lab_palabras/tapa3.lab  vs
C:/pruebas_htk16/data/train/label_lab_palabras/tapa3.rec
 LAB: sil tapa sil
 REC: sil papa sil
Aligned  transcription:  C:/pruebas_htk16/data/train/label_lab_palabras/tapa4.lab  vs
C:/pruebas_htk16/data/train/label_lab_palabras/tapa4.rec
 LAB: sil tapa sil
 REC: sil papa sil
Aligned  transcription:  C:/pruebas_htk16/data/train/label_lab_palabras/tasa1.lab  vs
C:/pruebas_htk16/data/train/label_lab_palabras/tasa1.rec
 LAB: sil tasa sil
 REC: sil pasa sil
Aligned  transcription:  C:/pruebas_htk16/data/train/label_lab_palabras/tasa3.lab  vs
C:/pruebas_htk16/data/train/label_lab_palabras/tasa3.rec
 LAB: sil tasa sil
 REC: sil pasa sil
Aligned  transcription:  C:/pruebas_htk16/data/train/label_lab_palabras/tasa4.lab  vs
C:/pruebas_htk16/data/train/label_lab_palabras/tasa4.rec
 LAB: sil tasa sil
 REC: sil pasa sil
Aligned  transcription:  C:/pruebas_htk16/data/train/label_lab_palabras/tengo2.lab  vs
C:/pruebas_htk16/data/train/label_lab_palabras/tengo2.rec
 LAB: sil tengo sil
 REC: sil tengo
Aligned  transcription:  C:/pruebas_htk16/data/train/label_lab_palabras/toldo4.lab  vs
C:/pruebas_htk16/data/train/label_lab_palabras/toldo4.rec
 LAB: sil toldo sil
 REC: sil codo  sil
Aligned  transcription:  C:/pruebas_htk16/data/train/label_lab_palabras/un_tomo1.lab
vs C:/pruebas_htk16/data/train/label_lab_palabras/un_tomo1.rec
 LAB: sil un_tomo   sil
 REC: sil inyeccion sil
Aligned  transcription:  C:/pruebas_htk16/data/train/label_lab_palabras/yo2.lab  vs
C:/pruebas_htk16/data/train/label_lab_palabras/yo2.rec
 LAB: sil yo     sil
 REC: sil cambio sil
====================== HTK Results Analysis =======================
  Date: Wed Mar 07 01:43:25 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=95.00 [H=570, S=30, N=600]
WORD: %Corr=98.33, Acc=98.33 [H=1770, D=5, S=25, I=0, N=1800]
```

Como se puede observar, el porcentaje de reconocimiento obtenido es de 98.33%.

C.2.4 Experimento 5. Reconocimiento con etiquetas con alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras

El procedimiento fue exactamente igual que en la sección C.2.1, pero esta vez se utilizaron etiquetas generadas con alineación forzada y segmentación acústica pero tomando en cuenta los tipos de fronteras. Sólo mostraremos los resultados de la evaluación del reconocimiento.

El archivo producido por HResults es el siguiente:

```
HResults -p -A -D -T 1 -t -I etiquetas_palabras.mlf label_list_palabras.txt
recout.mlf
No HTK Configuration Parameters Set
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/ala2.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/ala2.rec
 LAB: sil ala   sil
 REC: sil falda sil
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/ala4.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/ala4.rec
 LAB: sil ala   sil
 REC: sil falda sil
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/arca1.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/arca1.rec
 LAB: sil arca sil
 REC: sil anca sil
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/caray3.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/caray3.rec
 LAB: sil caray sil
 REC: sil calle sil
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/chinos4.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/chinos4.rec
 LAB: sil chinos sil
 REC: sil chinos
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/hongo2.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/hongo2.rec
 LAB: sil hongo sil
 REC: sil mano  sil
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/la_hierba4.lab
vs C:/pruebas_htk17/data/train/label_lab_palabras/la_hierba4.rec
 LAB: sil la_hierba sil
 REC: sil la_hierba
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/lado2.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/lado2.rec
 LAB: sil lado       sil
 REC: sil afortunado sil
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/llamar1.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/llamar1.rec
 LAB: sil llamar sil
 REC: sil llamar
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/mano1.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/mano1.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/mano2.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/mano2.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/mano3.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/mano3.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/mano4.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/mano4.rec
 LAB: sil mano sil
 REC: sil mama sil
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/nada2.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/nada2.rec
 LAB: sil nada sil
 REC:     moda sil
Aligned transcription: C:/pruebas_htk17/data/train/label_lab_palabras/para3.lab vs
C:/pruebas_htk17/data/train/label_lab_palabras/para3.rec
```

```
 LAB: sil para sil
 REC: sil para
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/pasa4.lab  vs
C:/pruebas_htk17/data/train/label_lab_palabras/pasa4.rec
 LAB: sil pasa sil
 REC: sil pasa
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/raro2.lab  vs
C:/pruebas_htk17/data/train/label_lab_palabras/raro2.rec
 LAB: sil raro  sil
 REC: sil radio sil
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/sol1.lab  vs
C:/pruebas_htk17/data/train/label_lab_palabras/sol1.rec
 LAB: sil sol sil
 REC: sil sol
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/tapa1.lab  vs
C:/pruebas_htk17/data/train/label_lab_palabras/tapa1.rec
 LAB: sil tapa sil
 REC: sil papa sil
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/tapa2.lab  vs
C:/pruebas_htk17/data/train/label_lab_palabras/tapa2.rec
 LAB: sil tapa sil
 REC: sil papa sil
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/tapa3.lab  vs
C:/pruebas_htk17/data/train/label_lab_palabras/tapa3.rec
 LAB: sil tapa sil
 REC: sil papa sil
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/tapa4.lab  vs
C:/pruebas_htk17/data/train/label_lab_palabras/tapa4.rec
 LAB: sil tapa sil
 REC: sil papa sil
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/tasa1.lab  vs
C:/pruebas_htk17/data/train/label_lab_palabras/tasa1.rec
 LAB: sil tasa sil
 REC: sil pasa sil
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/tasa3.lab  vs
C:/pruebas_htk17/data/train/label_lab_palabras/tasa3.rec
 LAB: sil tasa sil
 REC: sil pasa sil
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/tasa4.lab  vs
C:/pruebas_htk17/data/train/label_lab_palabras/tasa4.rec
 LAB: sil tasa sil
 REC: sil pasa sil
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/tengo2.lab  vs
C:/pruebas_htk17/data/train/label_lab_palabras/tengo2.rec
 LAB: sil tengo sil
 REC: sil tengo
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/un_tomo1.lab
vs C:/pruebas_htk17/data/train/label_lab_palabras/un_tomo1.rec
 LAB: sil un_tomo   sil
 REC: sil inyeccion sil
Aligned  transcription:  C:/pruebas_htk17/data/train/label_lab_palabras/yo2.lab  vs
C:/pruebas_htk17/data/train/label_lab_palabras/yo2.rec
 LAB: sil yo     sil
 REC: sil cambio sil
====================== HTK Results Analysis =======================
  Date: Wed Mar 07 11:56:50 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=95.33 [H=572, S=28, N=600]
WORD: %Corr=98.39, Acc=98.39 [H=1771, D=8, S=21, I=0, N=1800]
```

C.3 Experimentos con el corpus de dígitos

En esta sección se presentan los archivos de configuración y archivos de resultados importantes para los experimentos realizados con el corpus de dígitos. Todos los experimentos hechos para el corpus de dígitos siguieron la misma metodología, lo único que cambia entre ellos es el conjunto de archivos de etiquetas de entrenamiento.

C.3.1 Experimento 1. Reconocimiento con etiquetas manuales

En este experimento se entrenó un reconocedor en HTK utilizando todos los archivos del corpus tanto para entrenar como para reconocer. Los archivos de etiquetas utilizados fueron generados manualmente.

Configuración para HCopy

La llamada a HCopy se realizó de la siguiente manera:

```
HCopy -A -D -C analysis_conf.conf -S lista_archivos.scp
```

El archivo de configuración utilizado tiene los siguientes valores:

```
SOURCEFORMAT = WAV
TARGETKIND = MFCC_0_D_A
WINDOWSIZE = 200000
TARGETRATE =  10000
DELTAWINDOW = 2
ACCELWINDOW = 2
SAVECOMPRESSED = F
SAVEWITHCRC = F
NUMCEPS = 14
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 24
CEPLIFTER = 22
ENORMALISE = F
```

A continuación se muestra un fragmento de la lista de archivos *lista_archivos.scp*:

```
C:\Users\remington\Documents\corpusCIC_sin_silencios\wavs\cero1.wav
C:\pruebas_htk11\data\train\revueltos\cero1.mfc
C:\Users\remington\Documents\corpusCIC_sin_silencios\wavs\cero2.wav
C:\pruebas_htk11\data\train\revueltos\cero2.mfc
C:\Users\remington\Documents\corpusCIC_sin_silencios\wavs\cero3.wav
C:\pruebas_htk11\data\train\revueltos\cero3.mfc
C:\Users\remington\Documents\corpusCIC_sin_silencios\wavs\cero4.wav
C:\pruebas_htk11\data\train\revueltos\cero4.mfc
C:\Users\remington\Documents\corpusCIC_sin_silencios\wavs\cero5.wav
C:\pruebas_htk11\data\train\revueltos\cero5.mfc
C:\Users\remington\Documents\corpusCIC_sin_silencios\wavs\cero6.wav
C:\pruebas_htk11\data\train\revueltos\cero6.mfc
```

Y así sucesivamente hasta incluir todos los archivos del corpus.

Gramática

La gramática empleada se guardó en el archivo gram.txt y es la siguiente:

```
$palabra = cero | uno | dos | tres | cuatro | cinco | seis | siete | ocho | nueve;
(($palabra))
```

Como se puede observar, basta con identificar uno de los diez dígitos. No hay silencios que rodeen las palabras.

Compilación de la Gramática

La gramática se compiló con HParse:

```
HParse gram.txt wdnet.slf
```

Diccionario

El diccionario se guardó en el archivo *diccionario.txt* y su contenido es el siguiente:

```
a           a
e           e
i           i
o           o
u           u
r(          r(
j           j
n           n
w           w
tS          tS
t           t
k           k
d           d
V           V
s           s
pau         pau
cero        s e r( o
uno         u n o
dos         d o s
tres        t r( e s
cuatro      k w a pau t r( o
cinco       s i n pau k o
seis        s e j s
siete       s j e pau t e
ocho        o pau tS o
nueve       n w e V e
```

Transcripciones ortográficas

Se generó un archivo MLF con las transcripciones ortográficas para cada palabra del corpus. El archivo se llamó *etiquetas_palabras.mlf*. A continuación se muestra la primera parte del archivo:

```
#!MLF!#
"C:\pruebas_htk11\data\train\label_lab_palabras\cero1.lab"
cero
```

```
.
"C:\pruebas_htk11\data\train\label_lab_palabras\cero2.lab"
cero
.
"C:\pruebas_htk11\data\train\label_lab_palabras\cero3.lab"
cero
.
"C:\pruebas_htk11\data\train\label_lab_palabras\cero4.lab"
cero
.
"C:\pruebas_htk11\data\train\label_lab_palabras\cero5.lab"
cero
.
"C:\pruebas_htk11\data\train\label_lab_palabras\cero6.lab"
cero
.
"C:\pruebas_htk11\data\train\label_lab_palabras\cero7.lab"
cero
.
"C:\pruebas_htk11\data\train\label_lab_palabras\cero8.lab"
cero
.
```

Y así sucesivamente hasta cubrir todas las palabras del corpus.

Expansión de las transcripciones

Se expandieron las transcripciones usando HLEd. A continuación se muestra la llamada a HLEd.

```
HLEd -d diccionario.txt -i etiquetas_fonemas.mlf  mkfonemas0.led etiquetas_palabras.mlf
```

El archivo *mkfonemas0.led* únicamente contiene el comando EX para expandir la transcripción. El archivo *etiquetas_fonemas.mlf* es el archivo de salida generado. Parte del contenido de dicho archivo es el siguiente:

```
#!MLF!#
"C:/pruebas_htk11/data/train/label_lab_palabras/cero1.lab"
s
e
r(
o
.
"C:/pruebas_htk11/data/train/label_lab_palabras/cero2.lab"
s
e
r(
o
.
"C:/pruebas_htk11/data/train/label_lab_palabras/cero3.lab"
s
e
r(
o
.
"C:/pruebas_htk11/data/train/label_lab_palabras/cero4.lab"
s
e
r(
o
.
```

Y así sucesivamente para todas las palabras del corpus.

Creación de los protos

Los protos tienen la misma forma que en la sección C.2.1, sólo que esta vez se necesitan menos fonemas y por lo tanto se usan menos protos. Todos los protos tienen la misma forma que en la sección C.2.1.

Inicialización de los HMM

Cada modelo se inicializó utilizando HInit. Las llamadas a HInit realizadas son:

```
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_a.txt -l a -L
data/train/label_lab a
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_e.txt -l e -L
data/train/label_lab e
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_i.txt -l i -L
data/train/label_lab i
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_o.txt -l o -L
data/train/label_lab o
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_u.txt -l u -L
data/train/label_lab u
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_ere.txt -l r( -L
data/train/label_lab r(
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_j.txt -l j -L
data/train/label_lab j
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_n.txt -l n -L
data/train/label_lab n
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_w.txt -l w -L
data/train/label_lab w
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_tS.txt -l tS -L
data/train/label_lab tS
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_t.txt -l t -L
data/train/label_lab t
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_k.txt -l k -L
data/train/label_lab k
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_d.txt -l d -L
data/train/label_lab d
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_V.txt -l V -L
data/train/label_lab V
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_s.txt -l s -L
data/train/label_lab s
HInit -A -D -T 1 -S train_list.txt -M model/hmm0 -H model/proto/hmm_pau.txt -l pau -
L data/train/label_lab pau
```

Parte del contenido del archivo *train_list.txt* es el siguiente:

```
C:\pruebas_htk11\data\train\revueltos\cero1.mfc
C:\pruebas_htk11\data\train\revueltos\cero2.mfc
C:\pruebas_htk11\data\train\revueltos\cero3.mfc
C:\pruebas_htk11\data\train\revueltos\cero4.mfc
C:\pruebas_htk11\data\train\revueltos\cero5.mfc
C:\pruebas_htk11\data\train\revueltos\cero6.mfc
C:\pruebas_htk11\data\train\revueltos\cero7.mfc
C:\pruebas_htk11\data\train\revueltos\cero8.mfc
C:\pruebas_htk11\data\train\revueltos\cero9.mfc
```

Y así sucesivamente hasta incluir a todos los archivos del corpus.

Obtención del archivo vFloors

El archivo vFloors se obtuvo con la siguiente llamada a HCompv:

```
HCompv -C configuracion2.conf -f 0.01 -m -S train_list.txt -M model/hmm0flat
model/proto/proto.txt
```

El contenido del archivo vFloors es:

```
~v varFloor1
<Variance> 45
 6.554497e-001 4.948741e-001 8.126443e-001 1.032979e+000 6.625013e-001 5.156850e-001
6.564798e-001 4.955027e-001 8.301526e-001 7.251499e-001 3.133180e-001 4.211241e-001
3.226240e-001 2.350539e-001 8.956401e-001 9.280490e-004 1.489328e-003 2.240592e-003
3.262572e-003 3.661258e-003 4.042482e-003 4.516434e-003 4.648505e-003 5.291539e-003
5.203812e-003 4.574812e-003 3.912786e-003 3.468940e-003 2.894746e-003 9.657675e-004
6.787670e-005 1.378357e-004 2.090368e-004 3.241712e-004 4.022469e-004 4.568678e-004
5.040662e-004 5.413693e-004 6.206735e-004 6.260727e-004 5.322237e-004 4.735736e-004
4.203585e-004 3.534619e-004 6.897919e-005
```

El archivo macros.hmm

Este archivo se construyó a partir de vFloors. Su contenido es:

```
~o <VecSize> 45
   <MFCC_0_D_A>
~v varFloor1
<Variance> 45
 6.554497e-001 4.948741e-001 8.126443e-001 1.032979e+000 6.625013e-001 5.156850e-001
6.564798e-001 4.955027e-001 8.301526e-001 7.251499e-001 3.133180e-001 4.211241e-001
3.226240e-001 2.350539e-001 8.956401e-001 9.280490e-004 1.489328e-003 2.240592e-003
3.262572e-003 3.661258e-003 4.042482e-003 4.516434e-003 4.648505e-003 5.291539e-003
5.203812e-003 4.574812e-003 3.912786e-003 3.468940e-003 2.894746e-003 9.657675e-004
6.787670e-005 1.378357e-004 2.090368e-004 3.241712e-004 4.022469e-004 4.568678e-004
5.040662e-004 5.413693e-004 6.206735e-004 6.260727e-004 5.322237e-004 4.735736e-004
4.203585e-004 3.534619e-004 6.897919e-005
```

El archivo hmmdefs.mmf

Este archivo se construye copiando y pegando los modelos inicializados de cada fonema.

Reestimación embebida

Las llamadas a HERest se efectuaron de la siguiente manera:

```
HERest -C configuracion2.conf -I etiquetas_fonemas.mlf -t 250.0 150.0 1000.0 -S
train_list.txt -H model/hmm0/macros.hmm -H model/hmm0/hmmdefs.mmf -M model/hmm1
hmm_list.txt
HERest -C configuracion2.conf -I etiquetas_fonemas.mlf -t 250.0 150.0 1000.0 -S
train_list.txt -H model/hmm1/macros.hmm -H model/hmm1/hmmdefs.mmf -M model/hmm2
hmm_list.txt
HERest -C configuracion2.conf -I etiquetas_fonemas.mlf -t 250.0 150.0 1000.0 -S
train_list.txt -H model/hmm2/macros.hmm -H model/hmm2/hmmdefs.mmf -M model/hmm3
hmm_list.txt
HERest -C configuracion2.conf -I etiquetas_fonemas.mlf -t 250.0 150.0 1000.0 -S
train_list.txt -H model/hmm3/macros.hmm -H model/hmm3/hmmdefs.mmf -M model/hmm4
hmm_list.txt
HERest -C configuracion2.conf -I etiquetas_fonemas.mlf -t 250.0 150.0 1000.0 -S
train_list.txt -H model/hmm4/macros.hmm -H model/hmm4/hmmdefs.mmf -M model/hmm5
hmm_list.txt
HERest -C configuracion2.conf -I etiquetas_fonemas.mlf -t 250.0 150.0 1000.0 -S
train_list.txt -H model/hmm5/macros.hmm -H model/hmm5/hmmdefs.mmf -M model/hmm6
hmm_list.txt
```

```
HERest -C configuracion2.conf -I etiquetas_fonemas.mlf -t 250.0 150.0 1000.0 -S
train_list.txt -H model/hmm6/macros.hmm -H model/hmm6/hmmdefs.mmf -M model/hmm7
hmm_list.txt
HERest -C configuracion2.conf -I etiquetas_fonemas.mlf -t 250.0 150.0 1000.0 -S
train_list.txt -H model/hmm7/macros.hmm -H model/hmm7/hmmdefs.mmf -M model/hmm8
hmm_list.txt
HERest -C configuracion2.conf -I etiquetas_fonemas.mlf -t 250.0 150.0 1000.0 -S
train_list.txt -H model/hmm8/macros.hmm -H model/hmm8/hmmdefs.mmf -M model/hmm9
hmm_list.txt
HERest -C configuracion2.conf -I etiquetas_fonemas.mlf -t 250.0 150.0 1000.0 -S
train_list.txt -H model/hmm9/macros.hmm -H model/hmm9/hmmdefs.mmf -M model/hmm10
hmm_list.txt
```

HERest debe llamarse una vez por cada iteración de reestimación embebida.
El contenido del archivo *configuracion2.conf* es:

```
#SOURCEFORMAT = WAV
TARGETKIND = MFCC_0_D_A
WINDOWSIZE = 200000
TARGETRATE =  10000
DELTAWINDOW = 2
ACCELWINDOW = 2
SAVECOMPRESSED = F
SAVEWITHCRC = F
NUMCEPS = 14
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 24
CEPLIFTER = 22
ENORMALISE = F
```

El contenido del archivo *hmm_list.txt* es la lista de fonemas utilizados. Como el corpus de dígitos no contiene todos los fonemas del español, la lista será más pequeña que en el caso del corpus completo.

```
a
e
i
o
u
r(
j
n
w
tS
t
k
d
V
s
pau
```

Después de la última iteración (se realizaron 10 iteraciones), el modelo para la /a/, contenido en hmmdefs es el siguiente:

```
~h "a"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 45
 6.391406e+000   -8.609447e+000   -4.435300e+000   -1.254970e+001   8.244609e+000   -
2.081951e+000   -4.737635e+000   -2.302032e+000   -9.772567e+000   -9.637329e+000   -
```

```
9.810239e+000 5.639935e+000 -1.539207e-001 1.246108e+000 6.394442e+001 -2.789234e-
002 8.252249e-003 3.607908e-002 3.208783e-002 -3.698376e-002 -6.520212e-002
1.262681e-002 7.191704e-002 4.066378e-002 -1.238049e-001 7.439204e-002 3.411115e-002
-4.805028e-002 2.413726e-002 -1.653407e-002 1.235721e-003 3.798286e-003 8.984376e-
004 9.755729e-004 -2.344604e-003 -9.191213e-004 -3.545546e-003 -2.512057e-003
1.010085e-002 1.157937e-003 1.586094e-003 -6.638773e-003 -2.816813e-003 1.989674e-
003 -4.801860e-003
<VARIANCE> 45
 4.020576e+000 4.910030e+000 1.072857e+001 9.708337e+000 1.177548e+001 1.648092e+001
1.623426e+001 1.700168e+001 1.592197e+001 2.371626e+001 1.461760e+001 1.774174e+001
1.111272e+001 6.830937e+000 4.329995e+000 1.366729e-002 2.962012e-002 4.321791e-002
7.238989e-002 9.506541e-002 9.546868e-002 8.295696e-002 6.991546e-002 1.263835e-001
1.286487e-001 1.254600e-001 1.242559e-001 6.310038e-002 5.538759e-002 1.333626e-002
1.336861e-003 3.139935e-003 7.014467e-003 9.337248e-003 1.479417e-002 1.448580e-002
9.405031e-003 9.377636e-003 1.258493e-002 2.377348e-002 2.661693e-002 1.416606e-002
9.820807e-003 7.876933e-003 2.146712e-003
<GCONST> 5.416597e+000
<STATE> 3
<MEAN> 45
 -1.010576e+000 -1.041593e+001 -1.594342e+001 -8.117215e+000 -1.123568e+001 -
9.940168e+000 1.636332e+000 -1.723570e+001 -2.711855e+000 -2.391150e+001 -
3.418496e+000 -2.872347e+000 -5.731386e+000 6.779318e-001 7.400513e+001 -3.583869e-
002 -5.724829e-002 2.097183e-001 4.729469e-002 -1.126668e-001 -1.309594e-001
1.545226e-001 7.801795e-002 -1.235836e-001 -6.683277e-002 4.505718e-002 7.309340e-
002 4.166768e-002 -2.268128e-002 -1.388486e-002 5.324709e-003 1.019972e-002
5.651727e-003 2.823130e-004 -8.826769e-005 -1.013211e-003 -2.335917e-003 9.273727e-
003 -6.551283e-004 4.150386e-003 1.043994e-003 2.721319e-003 -1.835801e-003 -
6.034776e-003 -8.516188e-003
<VARIANCE> 45
 6.553956e+000 2.374557e+001 4.711252e+001 1.024522e+001 2.732317e+001 3.819331e+001
7.786598e+001 5.344527e+001 7.734258e+001 3.648202e+001 6.276641e+001 7.388886e+001
2.321540e+001 1.127369e+001 2.317275e+000 3.313739e-002 9.491936e-002 6.128638e-002
1.171191e-001 1.347995e-001 1.662367e-001 2.048796e-001 1.963443e-001 2.387581e-001
2.495745e-001 2.133580e-001 2.845865e-001 1.695629e-001 1.034273e-001 2.276112e-002
1.742153e-003 4.242215e-003 6.242143e-003 1.103302e-002 1.322940e-002 1.519576e-002
1.755069e-002 1.689141e-002 1.924255e-002 1.836876e-002 1.697579e-002 1.532702e-002
1.137539e-002 9.250238e-003 7.402440e-004
<GCONST> 3.024797e+001
<STATE> 4
<MEAN> 45
 1.227498e+000 1.583425e+000 -4.018637e+000 -3.696594e+000 -4.202891e+000 -
4.084752e+000 -1.001325e+000 -1.496550e+000 -2.549658e+000 -1.063210e+001 -
5.019620e+000 -1.732813e+000 -2.841651e+000 1.627566e+000 5.839483e+001 -1.315235e-
001 1.588333e-001 1.174203e-001 1.799958e-001 4.964380e-002 9.066524e-002 1.186448e-
002 -4.058070e-002 -7.880648e-002 5.610518e-002 -8.721112e-002 -3.904101e-002
1.480468e-001 -5.632386e-002 -4.155269e-001 -9.790575e-003 -9.188893e-003 -
2.863641e-003 -1.843276e-003 1.164139e-002 1.676523e-002 9.389578e-003 -8.955052e-
003 -2.319442e-002 -3.999378e-003 -6.009669e-003 2.430704e-003 9.155988e-003
4.973494e-003 1.744844e-003
<VARIANCE> 45
 1.354646e+001 2.043565e+001 1.265054e+001 2.094287e+001 3.325015e+001 3.373713e+001
5.688851e+001 3.057398e+001 1.099887e+002 7.506293e+001 2.785057e+001 2.216446e+001
3.106676e+001 2.279114e+001 4.401134e+001 1.107526e-001 2.682639e-001 2.546653e-001
4.357856e-001 5.753331e-001 5.737264e-001 6.801877e-001 7.163800e-001 9.398773e-001
7.121850e-001 6.816790e-001 4.980584e-001 5.607042e-001 3.533373e-001 1.419018e-001
1.035150e-002 1.583330e-002 2.362101e-002 3.831621e-002 4.961523e-002 6.045118e-002
6.848656e-002 7.780135e-002 8.082085e-002 7.249722e-002 7.045815e-002 5.913753e-002
6.471695e-002 4.290119e-002 5.777458e-003
<GCONST> 7.315857e+001
<TRANSP> 5
 0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
 0.000000e+000 9.663058e-001 9.874674e-003 2.381952e-002 0.000000e+000
```

```
 0.000000e+000 0.000000e+000 9.882358e-001 1.176423e-002 0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 9.758534e-001 2.414659e-002
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>
```

El archivo hmmdefs contiene una definición similar para cada fonema utilizado.

Reconocimiento

La llamada a HVite se ejecutó de la siguiente manera:

```
HVite -o SWT -H model/hmm10/macros.hmm -H model/hmm10/hmmdefs.mmf -S  testlist.txt -
i recout.mlf -w wdnet.slf -p 0.0 -s 5.0 diccionario.txt hmm_list.txt
```

Dado que tanto el entrenamiento como el reconocimiento se está haciendo sobre todo el corpus, *testlist.txt* tiene el mismo contenido que *train_list.txt*. El archivo producto del reconocimiento es *recout.mlf*; parte de su contenido se muestra a continuación.

```
#!MLF!#
"C:/pruebas_htk11/data/train/label_lab_palabras/cero1.rec"
cero
.
"C:/pruebas_htk11/data/train/label_lab_palabras/cero2.rec"
cero
.
"C:/pruebas_htk11/data/train/label_lab_palabras/cero3.rec"
cero
.
"C:/pruebas_htk11/data/train/label_lab_palabras/cero4.rec"
cero
.
"C:/pruebas_htk11/data/train/label_lab_palabras/cero5.rec"
cero
.
"C:/pruebas_htk11/data/train/label_lab_palabras/cero6.rec"
cero
.
```

Y así sucesivamente hasta abarcar la totalidad del corpus.

Evaluación

La llamada a HResults se realizó de la siguiente manera:

```
HResults  -p  -A  -D  -T  1  -t  -I  etiquetas_palabras.mlf  label_list_palabras.txt
recout.mlf
No HTK Configuration Parameters Set
Aligned  transcription:  C:/pruebas_htk11/data/train/label_lab_palabras/uno79.lab  vs
C:/pruebas_htk11/data/train/label_lab_palabras/uno79.rec
 LAB: uno
 REC: ocho
Aligned  transcription:  C:/pruebas_htk11/data/train/label_lab_palabras/dos2.lab  vs
C:/pruebas_htk11/data/train/label_lab_palabras/dos2.rec
 LAB: dos
 REC: ocho
Aligned  transcription:  C:/pruebas_htk11/data/train/label_lab_palabras/tres64.lab  vs
C:/pruebas_htk11/data/train/label_lab_palabras/tres64.rec
 LAB: tres
```

```
 REC: seis
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/tres76.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/tres76.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis2.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis2.rec
 LAB: seis
 REC: siete
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis3.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis3.rec
 LAB: seis
 REC: tres
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis6.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis6.rec
 LAB: seis
 REC: cinco
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis7.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis7.rec
 LAB: seis
 REC: cinco
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis10.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis10.rec
 LAB: seis
 REC: siete
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis14.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis14.rec
 LAB: seis
 REC: siete
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis18.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis18.rec
 LAB: seis
 REC: siete
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis19.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis19.rec
 LAB: seis
 REC: tres
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis22.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis22.rec
 LAB: seis
 REC: cinco
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis23.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis23.rec
 LAB: seis
 REC: cinco
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis26.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis26.rec
 LAB: seis
 REC: siete
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis30.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis30.rec
 LAB: seis
 REC: siete
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis34.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis34.rec
 LAB: seis
 REC: siete
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis38.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis38.rec
 LAB: seis
 REC: siete
```

```
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis46.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis46.rec
 LAB: seis
 REC: tres
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis54.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis54.rec
 LAB: seis
 REC: siete
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis59.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis59.rec
 LAB: seis
 REC: tres
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis63.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis63.rec
 LAB: seis
 REC: cinco
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis67.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis67.rec
 LAB: seis
 REC: cinco
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis70.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis70.rec
 LAB: seis
 REC: siete
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis71.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis71.rec
 LAB: seis
 REC: tres
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis74.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis74.rec
 LAB: seis
 REC: siete
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis75.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis75.rec
 LAB: seis
 REC: tres
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis78.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis78.rec
 LAB: seis
 REC: siete
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/seis79.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/seis79.rec
 LAB: seis
 REC: tres
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/ocho1.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/ocho1.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/ocho5.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/ocho5.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/ocho15.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/ocho15.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/ocho19.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/ocho19.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/ocho21.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/ocho21.rec
```

```
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/ocho23.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/ocho23.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/ocho54.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/ocho54.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/nueve3.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/nueve3.rec
 LAB: nueve
 REC: cuatro
Aligned transcription: C:/pruebas_htk11/data/train/label_lab_palabras/nueve7.lab vs
C:/pruebas_htk11/data/train/label_lab_palabras/nueve7.rec
 LAB: nueve
 REC: cuatro
====================== HTK Results Analysis =======================
  Date: Tue Mar 06 10:08:51 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=95.24 [H=760, S=38, N=798]
WORD: %Corr=95.24, Acc=95.24 [H=760, D=0, S=38, I=0, N=798]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  80   0   0   0   0   0   0   0   0   0    0
 uno   0  79   0   0   0   0   0   0   1   0    0 [98.8/0.1]
 dos   0   0  79   0   0   0   0   0   1   0    0 [98.8/0.1]
tres   0   0   0  78   0   0   2   0   0   0    0 [97.5/0.3]
cuat   0   0   0   0  79   0   0   0   0   0    0
cinc   0   0   0   0   0  80   0   0   0   0    0
seis   0   0   0   7   0   6  55  12   0   0    0 [68.8/3.1]
siet   0   0   0   0   0   0   0  79   0   0    0
ocho   0   0   0   0   7   0   0   0  73   0    0 [91.3/0.9]
nuev   0   0   0   0   2   0   0   0   0  78    0 [97.5/0.3]
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

Como en esta ocasión el corpus es más sencillo, se vuelve práctico también mostrar la matriz de confusión. El porcentaje de reconocimiento es de 95.24%.

C.3.2 Experimento 2. Reconocimiento con etiquetas con alineación forzada

El procedimiento fue exactamente igual que en la sección C.3.1, con la única diferencia de que esta vez se utilizaron etiquetas generadas con alineación forzada en lugar de etiquetas manuales. Sólo se mostrarán los resultados de la evaluación del reconocimiento.

```
HResults -p -A -D -T 1 -t -I etiquetas_palabras.mlf label_list_palabras.txt
recout.mlf
No HTK Configuration Parameters Set
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/uno79.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/uno79.rec
 LAB: uno
 REC: ocho
```

```
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres3.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres3.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres7.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres7.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres11.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres11.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres19.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres19.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres26.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres26.rec
 LAB: tres
 REC: siete
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres27.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres27.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres31.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres31.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres35.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres35.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres38.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres38.rec
 LAB: tres
 REC: siete
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres39.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres39.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres67.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres67.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres71.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres71.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/tres75.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/tres75.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/cinco10.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/cinco10.rec
 LAB: cinco
 REC: siete
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/cinco14.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/cinco14.rec
 LAB: cinco
 REC: siete
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/cinco34.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/cinco34.rec
```

```
 LAB: cinco
 REC: siete
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/cinco38.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/cinco38.rec
 LAB: cinco
 REC: siete
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/cinco42.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/cinco42.rec
 LAB: cinco
 REC: siete
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/cinco50.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/cinco50.rec
 LAB: cinco
 REC: siete
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/cinco62.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/cinco62.rec
 LAB: cinco
 REC: siete
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/cinco66.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/cinco66.rec
 LAB: cinco
 REC: siete
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/cinco74.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/cinco74.rec
 LAB: cinco
 REC: siete
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/cinco78.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/cinco78.rec
 LAB: cinco
 REC: siete
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/siete3.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/siete3.rec
 LAB: siete
 REC: seis
Aligned transcription: C:/pruebas_htk12/data/train/label_lab_palabras/nueve1.lab vs
C:/pruebas_htk12/data/train/label_lab_palabras/nueve1.rec
 LAB: nueve
 REC: cuatro
====================== HTK Results Analysis =======================
  Date: Tue Mar 06 10:36:32 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=96.74 [H=772, S=26, N=798]
WORD: %Corr=96.74, Acc=96.74 [H=772, D=0, S=26, I=0, N=798]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  80   0   0   0   0   0   0   0   0   0    0
 uno   0  79   0   0   0   0   0   0   1   0    0 [98.8/0.1]
 dos   0   0  80   0   0   0   0   0   0   0    0
tres   0   0   0  67   0   0  11   2   0   0    0 [83.8/1.6]
cuat   0   0   0   0  79   0   0   0   0   0    0
cinc   0   0   0   0   0  70   0  10   0   0    0 [87.5/1.3]
seis   0   0   0   0   0   0  80   0   0   0    0
siet   0   0   0   0   0   0   1  78   0   0    0 [98.7/0.1]
ocho   0   0   0   0   0   0   0   0  80   0    0
nuev   0   0   0   0   1   0   0   0   0  79    0 [98.8/0.1]
Ins    0   0   0   0   0   0   0   0   0   0
```

```
      ===================================================================

      No HTK Configuration Parameters Set
```

El porcentaje de reconocimiento obtenido es de 96.74%.

C.3.3 Experimento 3. Reconocimiento con etiquetas con alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras

El procedimiento es exactamente igual que el que se siguió en la sección C.3.1; con la única diferencia de que esta vez las etiquetas usadas se generaron con alineación forzada y segmentación acústica tomando en cuenta los tipos de fronteras. Sólo se mostrarán los resultados de la evaluación del reconocimiento.

```
      HResults -p -A -D -T 1 -t -I etiquetas_palabras.mlf label_list_palabras.txt
      recout.mlf
      No HTK Configuration Parameters Set
      Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/uno3.lab vs
      C:/pruebas_htk13/data/train/label_lab_palabras/uno3.rec
       LAB: uno
       REC: cuatro
      Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/uno31.lab vs
      C:/pruebas_htk13/data/train/label_lab_palabras/uno31.rec
       LAB: uno
       REC: cuatro
      Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/uno43.lab vs
      C:/pruebas_htk13/data/train/label_lab_palabras/uno43.rec
       LAB: uno
       REC: cero
      Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/uno47.lab vs
      C:/pruebas_htk13/data/train/label_lab_palabras/uno47.rec
       LAB: uno
       REC: cero
      Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/uno55.lab vs
      C:/pruebas_htk13/data/train/label_lab_palabras/uno55.rec
       LAB: uno
       REC: cero
      Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/uno59.lab vs
      C:/pruebas_htk13/data/train/label_lab_palabras/uno59.rec
       LAB: uno
       REC: cero
      Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/uno63.lab vs
      C:/pruebas_htk13/data/train/label_lab_palabras/uno63.rec
       LAB: uno
       REC: tres
      Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/uno67.lab vs
      C:/pruebas_htk13/data/train/label_lab_palabras/uno67.rec
       LAB: uno
       REC: cero
      Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/uno79.lab vs
      C:/pruebas_htk13/data/train/label_lab_palabras/uno79.rec
       LAB: uno
       REC: cero
      Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/dos3.lab vs
      C:/pruebas_htk13/data/train/label_lab_palabras/dos3.rec
       LAB: dos
       REC: tres
      Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/dos11.lab vs
      C:/pruebas_htk13/data/train/label_lab_palabras/dos11.rec
       LAB: dos
       REC: tres
```

```
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/dos19.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/dos19.rec
 LAB: dos
 REC: tres
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/dos59.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/dos59.rec
 LAB: dos
 REC: tres
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/tres12.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/tres12.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/tres19.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/tres19.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/tres24.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/tres24.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/tres28.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/tres28.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/tres44.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/tres44.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/tres48.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/tres48.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/tres52.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/tres52.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/tres56.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/tres56.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/tres64.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/tres64.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/tres68.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/tres68.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/tres76.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/tres76.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/tres80.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/tres80.rec
 LAB: tres
 REC: seis
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/seis73.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/seis73.rec
 LAB: seis
 REC: tres
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho5.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho5.rec
```

```
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho7.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho7.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho11.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho11.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho15.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho15.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho19.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho19.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho23.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho23.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho27.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho27.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho35.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho35.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho39.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho39.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho43.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho43.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho47.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho47.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho55.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho55.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho59.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho59.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho67.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho67.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho71.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho71.rec
 LAB: ocho
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/ocho79.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/ocho79.rec
 LAB: ocho
 REC: cuatro
```

```
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/nueve1.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/nueve1.rec
 LAB: nueve
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/nueve5.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/nueve5.rec
 LAB: nueve
 REC: cuatro
Aligned transcription: C:/pruebas_htk13/data/train/label_lab_palabras/nueve33.lab vs
C:/pruebas_htk13/data/train/label_lab_palabras/nueve33.rec
 LAB: nueve
 REC: cuatro
====================== HTK Results Analysis =======================
  Date: Tue Mar 06 09:41:49 2012
  Ref : etiquetas_palabras.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=94.36 [H=753, S=45, N=798]
WORD: %Corr=94.36, Acc=94.36 [H=753, D=0, S=45, I=0, N=798]
------------------------ Confusion Matrix -------------------------
       c   u   d   t   c   c   s   s   o   n
       e   n   o   r   u   i   e   i   c   u
       r   o   s   e   a   n   i   e   h   e
       o           s   t   c   s   t   o   v
                       r   o       e       e  Del [ %c / %e]
cero  80   0   0   0   0   0   0   0   0   0    0
 uno   6  71   0   1   2   0   0   0   0   0    0 [88.8/1.1]
 dos   0   0  76   4   0   0   0   0   0   0    0 [95.0/0.5]
tres   0   0   0  68   0   0  12   0   0   0    0 [85.0/1.5]
cuat   0   0   0   0  79   0   0   0   0   0    0
cinc   0   0   0   0   0  80   0   0   0   0    0
seis   0   0   0   1   0   0  79   0   0   0    0 [98.8/0.1]
siet   0   0   0   0   0   0   0  79   0   0    0
ocho   0   0   0   0  16   0   0   0  64   0    0 [80.0/2.0]
nuev   0   0   0   0   3   0   0   0   0  77    0 [96.3/0.4]
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================

No HTK Configuration Parameters Set
```

El porcentaje de reconocimiento obtenido es de 94.36%.