UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Construcción de un reconocedor de voz utilizando Sphinx y el corpus DIMEx100

T E S I S

QUE PARA OBTENER EL TÍTULO
DE:INGENIERA EN COMPUTACIÓN

P R E S E N T A:

ELIA PATRICIA PÉREZ PAVÓN

Asesor: Dr. Luis Alberto Pineda Cortés

México, D.F. 2006

Todo comenzó en una clase, después un comentario me llevó a la curiosidad. La curiosidad mató al gato pero como no soy gato, finalmente surgió esta tesis que costó esfuerzo y dedicación pero ahora que puedo leer este trabajo puedo decir que valió la pena porque a través de él pude introducirme en esa área de la Universidad que me era desconocida hasta entonces: la investigación.

AGRADECIMIENTOS

A mi asesor el Dr. Luis Alberto Pineda por la oportunidad que me dio de formar parte del grupo DIME en el IIMAS en donde desarrollé esta tesis, encontré buenos amigos y conocí personas muy interesantes. También quiero agradecer su apoyo constante durante la realización de este trabajo y todo el conocimiento que nos transmitió durante las sesiones de seminario a las que asistí desde que empecé a trabajar con él. Mil gracias.

Al grupo DIME (Haydé, Ivonne, Fernanda, Isabel, Varinia, Iván, Sergio, Javier Cuétara, Laura, Arturo) por su valiosa colaboración en la etiquetación del corpus, etiquetación sin la cual no habría podido desarrollar mi trabajo. Gracias a Cesar por todo el apoyo técnico y que me prestara su laptop cada vez que necesitaba hacer pruebas.

Al Dr. Luis Villaseñor (INAOE) por permitirme conocer Sphinx y sus módulos.

Al Dr. Lucian Galescu (IHMC) por su valiosa ayuda en el proceso de construcción del sistema y su extenso conocimiento de la herramienta Sphinx.

A mis padres por todo su apoyo a lo largo de estos 23 años de mi vida y por impulsarme a terminar lo que había comenzado, los quiero mucho.

A Andrés por su paciencia y comprensión a lo largo de estos meses de trabajo en los cuales hubo malos ratos pero que ya pasamos y seguimos más juntos que antes. También porque algunas de las imágenes de esta tesis las hizo él.

A mis amigos por alentarme en todo momento y por darme la fuerza para continuar cuando pensaba que ya no había camino. Gracias Dan por tus consejos de redacción.

Al Proyecto NSF/CONACYT 39380-U, DIME-II: Diálogos Inteligentes Multimodales en Español, bajo la responsabilidad del Dr. Luis Pineda Cortés.

INDICE

Página

INDICE DE FIGURAS

INDICE DE TABLAS

CAPÍTULO 1

INTRODUCCIÓN

Entender el lenguaje hablado es una tarea difícil; el objetivo de la investigación en ASR (*Automatic Speech Recognition* – Reconocimiento automático de voz) es atacar el problema computacionalmente mediante la construcción de sistemas que mapeen una señal acústica a una cadena de palabras. En particular, un reconocedor de habla es un dispositivo que automáticamente transforma una señal acústica de entrada a texto.

El problema general de la trascripción automática del lenguaje, para cualquier hablante en cualquier ambiente, esta aun lejos de resolverse; sin embargo, en años recientes se ha visto una maduración en la tecnología del ASR hasta el punto donde esto es viable si el dominio es limitado.

El reconocimiento de voz es fundamentalmente una tarea de clasificación de patrones. Se toma un patrón de entrada, que en este caso es la señal de voz, y se clasifica dentro del conjunto de realizaciones fonéticas o alofónicas establecido. Sin embargo, la principal dificultad del reconocimiento del habla es que las señales de entrada son significativamente diferentes debido a la gran variedad de hablantes (hombres, mujeres, niños, etc.), la velocidad y la forma con la que se expresan, la región de la que vienen e incluso su estado de ánimo.

Una gran área de aplicación para el reconocimiento de voz es la interacción humano-computadora [12]. Muchas tareas se resuelven bastante bien mediante interfaces visuales, pero el habla tiene el potencial de ser una mejor interfaz que el teclado para tareas donde es útil la comunicación en lenguaje natural o para las cuales el teclado no es apropiado. Esto incluye aplicaciones, por ejemplo, donde el usuario tenga objetos que manipular o equipo

que controlar. Otra área importante de aplicación es la telefonía, donde el reconocimiento de voz ya es usado para introducir dígitos o para reconocer palabras para obtener un servicio. En algunas aplicaciones una interfaz multimodal combinando habla y el uso del teclado puede ser más eficiente que una interfaz gráfica de usuario sin habla.

Esta tesis de licenciatura se planteó debido a que poco trabajo se ha realizado en lo que se refiere al reconocimiento de voz para el español y es necesario contar con esta tecnología que cada vez toma más fuerza y tiene mayores aplicaciones.

Esta tesis, además, servirá para completar uno de los módulos del proyecto DIME II (Diálogos Inteligentes Multimodales en Español) que se desarrolla en el IIMAS (Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas) y pretende la construcción de un sistema multimodal dentro del dominio de diseño de cocinas.

Un reconocedor de voz para el español de México es un recurso muy valioso que puede servir para desarrollar proyectos mayores y lograr considerables avances que impacten en la tecnología de nuestro país. Mucho se ha avanzado para otros idiomas como el inglés y mucho han hecho otros países para incursionar en el mercado latino; sin embargo, se considera la posibilidad de desarrollar la tecnología desde adentro y, aunque tomará tiempo, se pondrá una piedra más en la construcción de tecnología útil para la investigación en México. El corpus DIMEx100, del que se hablará en el siguiente capítulo, es una muestra de ello y es un recurso indispensable en la elaboración de esta tesis.

Es así que en este capítulo, además de dar una explicación del porque del tema de la tesis, se menciona parte de la teoría matemática involucrada en la construcción de un reconocedor, la intuición del canal ruidoso, el estado del arte de estas tecnologías del habla, así como algunos antecedentes de importancia.

1.1 Formulación Matemática

De acuerdo a Jelinek [7], para hablar del problema de diseño de un reconocedor de voz se necesita una formulación matemática; para este efecto se propone, tanto en modelado de pronunciación para ASR, así como para la corrección de ortografía en OCR (*Optical Character Recognition* – Reconocimiento óptico de caracteres), mapear de una cadena de símbolos a otra. En el reconocimiento de voz, dada una cadena de símbolos que representa la pronunciación de una palabra en un contexto, se busca la correspondiente cadena de símbolos en el diccionario de pronunciación. De esta manera se puede hablar de modelos probabilísticos de variación de pronunciación y ortografía, en particular, del modelo de inferencia de Bayes o canal ruidoso (figura 1). Jelinek introdujo la metáfora del canal ruidoso en una aplicación del modelo para reconocimiento de voz en 1976. Esta idea consiste en tratar a la entrada (ya sea una mala pronunciación o una palabra mal escrita) como una instancia de la forma léxica (la pronunciación léxica o la correcta ortografía) que ha pasado a través de un canal de comunicación ruidoso. Este canal introduce "ruido" por lo que es difícil reconocer la palabra o elocución original. El objetivo es entonces la construcción de un modelo de canal ruidoso que permita imaginar como se modificó la elocución original y así poder recuperarla. Este ruido, tratándose de reconocimiento de voz, puede ser causado por variaciones en la pronunciación, variaciones en la realización del fonema, variación acústica debido al canal (micrófono, teléfono, la red), etc.

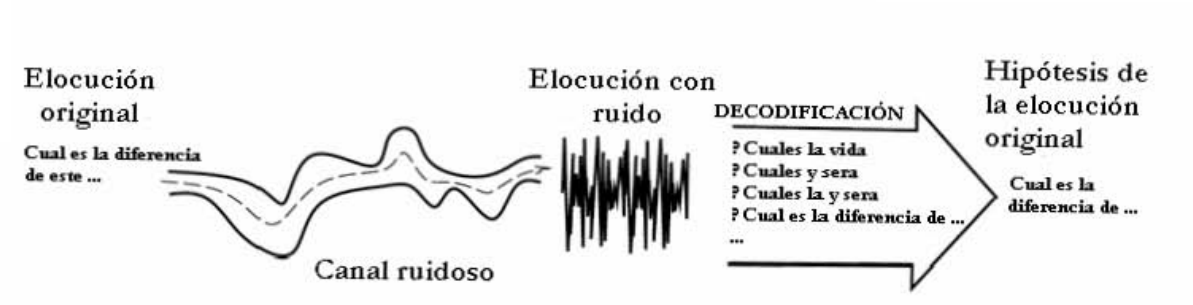


Figura 1. Canal Ruidoso

Para la tarea de reconocimiento de voz, en general, se asume que teniendo una entrada acústica O, se puede tratar como una secuencia de símbolos u observaciones individuales (por ejemplo, partiendo la señal cada 10 ms y representando cada segmento mediante valores en punto flotante de la energía y la frecuencia de ese segmento). Cada índice

entonces representa un intervalo de tiempo, y sucesivos o_i representan segmentos consecutivos temporales de la entrada:

$$O = o_1, o_2, o_3, \ldots, o_t$$

Además, O es una secuencia de símbolos tomados de un alfabeto cualquiera *O*.

De manera similar, se trata a una elocución como si estuviera compuesta simplemente por una cadena de palabras que pertenecen a un diccionario *W*. El mensaje se representa como sigue:

$$W = w_1, w_2, w_3, \ldots, w_t$$

La implementación probabilística de la intuición anterior, se expresa de la siguiente forma:

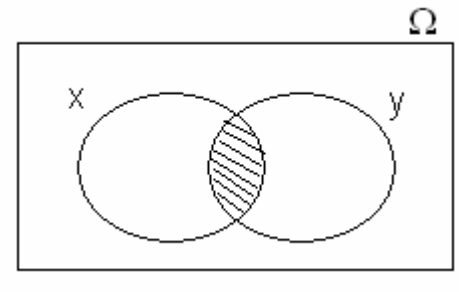
$$\hat{W} = \arg\max_{W \in \mathcal{L}} P(W|O) \quad \ldots \quad (1)$$

La función probabilística en (1) garantiza dar la elocución óptima *W*, pero no es operacional puesto que dada una elocución W y una secuencia acústica O se necesita calcular $P(W|O)$ lo que significa obtener la probabilidad de una elocución dadas todas las posibles secuencias de observación en el mundo de acuerdo a un alfabeto cualquiera *O*. $P(W|O)$ resulta imposible de obtener, sin embargo, es posible calcular este valor a través del teorema de Bayes que nos dice que dada una probabilidad $P(x|y)$:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \quad \ldots (2)$$

A $P(x|y)$ se le conoce como probabilidad condicional y es la probabilidad de que un segundo evento (x) se presente, si un primer evento (y) ya ha sucedido.

A continuación se muestra una prueba gráfica del teorema:



$$P(x|y) = \frac{P(x \cap y)}{P(y)} \quad \ldots (3)$$

$$P(y|x) = \frac{P(x \cap y)}{P(x)} \quad \ldots (4)$$

Despejando (4) y sustituyendo en (3) finalmente se obtiene la expresión en (2):

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

Q.E.D.

De esta manera, la expresión (1) modificada con el teorema de Bayes queda como sigue:

$$\hat{W} = \arg\max_{W \in L} \frac{P(O|W)P(W)}{P(O)} \quad \ldots \quad (5)$$

Las probabilidades de la parte derecha de la expresión en (5) son, en su mayoría, más fáciles de computar que la probabilidad de la expresión en (1) [$P(W|O)$]; sin embargo, $P(O)$, la probabilidad de la secuencia de observación acústica aún resulta difícil de estimar porque no se sabe cual es la secuencia de observación de todas las posibles dentro del dominio, pero se sabe que se está examinando la misma secuencia de observación O para cada elocución en potencia y, por lo tanto, se debe tener la misma $P(O)$ para todas las elocuciones posibles. Como el denominador es el mismo para cada elocución candidata W se puede ignorar y la expresión queda de la siguiente manera:

$$W = \arg\max_{W \in L} \frac{P(O|W)P(W)}{P(O)} = \arg\max_{W \in L} P(O|W)P(W) \ldots (6)$$

Así, la elocución W más probable dada una secuencia de observación O se calcula mediante el producto de dos probabilidades $P(O|W)$ y $P(W)$. Para el caso del

reconocimiento de voz $P(O|W)$, la probabilidad de observación, se calcula a través **del modelo acústico** y $P(W)$, la probabilidad a priori, se calcula a través **del modelo del lenguaje** como se ve en (7).

$$W = \arg\max_{W \in L} \underbrace{P(O|W)}_{\text{modelo acustico}} \underbrace{P(W)}_{\text{modelo del lenguaje}} \quad \ldots (7)$$

1.2 Funcionamiento de un Sistema Reconocedor de voz

El funcionamiento de un sistema reconocedor de voz comprende dos etapas: una de entrenamiento y una de reconocimiento [9]. Durante la etapa de entrenamiento, se proporciona al sistema cierta cantidad de pronunciaciones que se desea que éste tenga o "memorice"; lo que el sistema almacena son las propiedades de un conjunto (fonemas, alófonos, etc.) y no las pronunciaciones en sí. Durante la etapa de reconocimiento, el sistema identifica una pronunciación que con mayor probabilidad se parece a las pronunciaciones que están en la memoria del sistema.

EL esquema de reconocimiento se muestra en la figura 2:

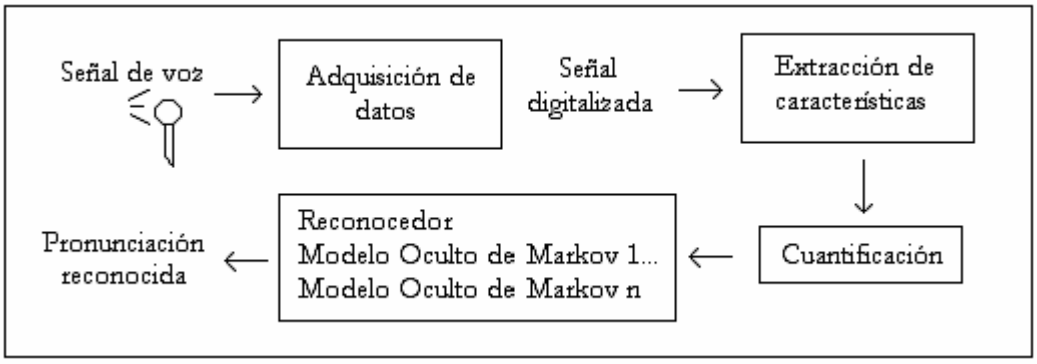


Figura 2. Esquema de reconocimiento

El módulo de adquisición de datos convierte la señal sonora a eléctrica y después a una secuencia de valores numéricos, es decir, hace la conversión analógica a digital.

El módulo de extracción de propiedades o características obtiene datos de la señal como son energía espectral, tono, formantes, etc., correspondientes a una pronunciación. El proceso consiste en dividir la secuencia de valores, obtenida en el módulo anterior, en segmentos correspondientes a una duración de entre 10 y 35 milisegundos debido a que se ha determinado que la duración de todos los sonidos del habla está en ese rango. La salida de este módulo consiste en una secuencia de vectores de características de los segmentos.

El módulo de cuantificación identifica los distintos sonidos que están presentes en la pronunciación. La salida de este módulo es una secuencia de valores, donde cada valor representa el sonido con el que está asociado un vector de características.

El módulo de reconocimiento es el que finalmente identifica a una pronunciación dada y la clasifica dentro de sus modelos como un sonido conocido, parecido a uno conocido o bien, desconocido. La complejidad de este módulo dependerá del tipo de identificación que se requiera. Por ejemplo, un reconocedor de gramáticas será más complejo que uno de palabras y un reconocedor de palabras será más complejo que uno de letras o de fonemas.

1.3 Modelos y algoritmos

Existen diferentes niveles de procesamiento del lenguaje:

- Fonética y fonología.
- Morfología
- Sintaxis
- Semántica
- Pragmática
- Discurso

A lo largo de 50 años de investigación en el área de Procesamiento del Lenguaje, se han madurado una serie de modelos formales o teorías que resuelven algunos problemas en alguno de los niveles descritos anteriormente y que se encuentran en varias herramientas computacionales [8]. Contar con estos *toolkits* facilita el desarrollo de sistemas y permite contar con nuevas técnicas de procesamiento. El hablar de reconocimiento de voz es procesar el lenguaje en un nivel fonético y fonológico.

Entre los principales modelos que componen algunas de las herramientas computacionales conocidas se encuentran: las máquinas de estados, los sistemas de reglas formales y la lógica; la teoría probabilística se incluye en cada unos de esos modelos. Estos elementos, a su vez, contienen algoritmos conocidos de "paradigmas computacionales". Entre los algoritmos más importantes se tienen los de búsqueda en el espacio de estados y los de programación dinámica.

Para describir más a fondo los modelos utilizados en el procesamiento del lenguaje y del habla, se presenta primero a las máquinas de estado. Éstas se definen, de manera muy simple, como modelos formales que tienen estados, transiciones entre los estados y cuyo comportamiento queda completamente determinado por la secuencia de datos de entrada que recibe. Variaciones de éstos modelos son los autómatas de estados finitos determinísticos y no determinísticos, transductores de estados finitos, autómatas de pesos, modelos de Markov y Modelos Ocultos de Markov; estos últimos tienen un componente probabilístico.

Un autómata de pesos o modelo de Markov simple es especificado por un conjunto de estados**,** un conjunto de **probabilidades de transición**, un **estado inicial**, uno o más **estados finales** y un conjunto de **probabilidades de observación**.

Un Modelo Oculto de Markov (HMM) formalmente difiere de un modelo de Markov simple pues agrega dos requerimientos. Primero, un HMM tiene un conjunto separado de símbolos de observación los cuales no salen del mismo alfabeto como el conjunto de estados que los componen. Segundo, la función de probabilidad de observación no está

limitada a valores discretos (por ejemplo 0 y 1), en un HMM las probabilidades puede tomar cualquier valor real en el intervalo [0,1].

Otro modelo muy relacionado con las máquinas de estado y que también se utiliza mucho cuando se trabaja con aspectos morfológicos, fonológicos y sintácticos, son los llamados sistemas de reglas formales. De los más importantes que se conocen son las gramáticas regulares, las gramáticas libres de contexto así como sus variaciones probabilísticas.

Los algoritmos asociados con estos modelos presentados requieren buscar una solución en un espacio de estados. Los más frecuentemente usados para estas tareas, conocidos como algoritmos de grafos, son primero el mejor, A* o búsqueda en profundidad.

Se conocen, principalmente, dos algoritmos diferentes los cuales calculan simultáneamente la probabilidad de una secuencia de observación dada, y arrojan la elocución más probable de acuerdo al modelo del canal ruidoso. Estos son **el algoritmo de Viterbi** y el **A***.

El tercer modelo es la lógica. Aquí se tiene el cálculo de predicados, redes semánticas y dependencias conceptuales. Estas representaciones lógicas son usadas frecuentemente cuando se trabaja con aspectos de semántica, pragmática y discurso.

1.4 Estado del arte en el Procesamiento del Lenguaje

Muchos investigadores piensan que estamos en un buen momento para el perfeccionamiento del procesamiento del lenguaje[1].

La primera máquina reconocedora de dígitos se desarrolló en 1952 en los laboratorios Bell y fue un sistema estadístico que podía reconocer cualquiera de los 10 dígitos (0-9) dichos por un hablante (Davis et al., 1952). Con este sistema se alcanzaron resultados de hasta

[1] Jurafsky & Martin, *Speech and Language Processing*, página 9. Prentice Hall, 2000

99% de precisión escogiendo el patrón que tuviera el mayor coeficiente de correlación relativa con la entrada. A principios de 1970, Lenny Baum, de la Universidad de Princeton, desarrolla el enfoque del Modelo Oculto de Markov hacia el reconocimiento de voz y lo comparte con varios contratistas de ARPA (Advanced Research Projects Agency), incluyendo IBM. En 1971, DARPA (Defense Advanced Research Projects Agency) establece el programa SUR (Speech Understanding Research) para desarrollar un sistema computacional que pudiera entender el habla continua. Lawrence Roberts, quien inició el programa, gastó tres millones de dólares por año de fondos del gobierno durante cinco años. Los principales grupos del proyecto SUR se establecieron en CMU, SRI, el Laboratorio Lincoln de MIT, SDC (Systems Development Corporation) y BBN (Bolt, Beranek and Newman).

Durante los 70´s se desarrollaron los sistemas de conversión de texto en habla, con los que es posible "escuchar" un texto almacenado en una computadora. Al mismo tiempo surgen sistemas que reconocen palabras aisladas, sistemas que verifican la identidad de la persona que habla y nuevas técnicas de codificación de la voz para mejorar su procesamiento. En cuanto a la década de los 80´s, los adelantos en el campo del reconocimiento de voz continua permitieron eliminar el tener que introducir pausas entre cada palabra para que la computadora reconozca enunciados. En 1982, Jim y Janet Barker, pioneros de la industria de habla, fundan Dragon Systems.

En 1989, Steve Young desarrolla la primera versión de HTK (Hidden Markov Model Toolkit) dentro del grupo Speech Vision and Robotics de la Universidad de Cambridge.

Actualmente se puede acceder mediante el habla a la información almacenada en un sistema informático utilizando el teléfono, o bien, prescindir del teclado cuando se trata de una agenda muy sofisticada o si contamos con un software especializado que nos permite, haciendo uso de un micrófono, dictarle a un programa de procesamiento de texto. También ya contamos con sistemas que permiten generar mensajes verbales que incluso se pueden introducir en algunos electrodomésticos, automóviles o juguetes con el fin de emitir un número limitado de mensajes.

Existen un gran número de empresas que desarrollan productos para comercializar. Tal es el caso de Scansoft en conjunto con Nuance, que domina el campo de aplicaciones para telefonía, IBM Via Voice de IBM y el sistema de dictado de la compañía Philips.

1.5 Notas Históricas

El procesamiento del lenguaje y habla ha sido tratado, desde sus inicios, por diferentes departamentos en sus áreas de investigación: la lingüística computacional en lingüística, el procesamiento del lenguaje natural en ciencias de la computación, el reconocimiento del habla en ingeniería eléctrica y la psicolingüistica computacional en psicología. El campo del procesamiento del lenguaje natural surgió después de la Segunda Guerra Mundial al tiempo que la computación despegaba.

Durante los años 40´s y 50´s surgieron los autómatas y las teorías probabilísticas. Los autómatas surgieron como consecuencia de la máquina de Turing (1936), que más tarde, en su forma electrónica, sería la computadora digital. McColloch y Pitts (1943) desarrollaron el modelo de la neurona; Kleene (1951 y 1956) trabajo en autómatas finitos y expresiones regulares; Claude Shannon (1948) aplicó modelos probabilísticas de procesos discretos de Markov. En base a esto Noam Chomsky (1956) consideró a las máquinas de estado finitos como una forma de caracterizar las gramáticas y definió un lenguaje de estado finitos como un lenguaje generado por una gramática de estados finitos. De estas ideas surgió el campo de la teoría de los lenguajes formales.

Un aspecto importante de este período fue el desarrollo de algoritmos probabilísticos para el procesamiento del lenguaje. Además se tuvo una gran aportación de Shannon en lo que se refiere a la teoría de la información con “Teoría Matemática de la Comunicación” donde se muestra que todas las fuentes de información se pueden medir y que los canales de comunicación tienen una unidad de medida similar. En esta teoría sentó las bases para la corrección de errores, supresión de ruidos y redundancia. El concepto de la entropía es una

característica importante de su teoría que explica que existe un cierto grado de incertidumbre de que el mensaje llegué completo.

A principios de los 60´s el procesamiento del lenguaje se dividió en dos paradigmas: los simbólicos y los estocásticos.

El paradigma simbólico surgió principalmente del trabajo de Chomsky en lo que se refiere a la teoría de los lenguajes formales, así como al trabajo de muchos lingüistas y computólogos realizando algoritmos de parseo. En el verano de 1956, importantes investigadores en el campo decidieron que la nueva línea de investigación que surgía se llamaría Inteligencia Artificial.

El paradigma estocástico surgió en los departamentos de estadística y de ingeniería electrónica. Para finales de los 50´s, el método de Bayes se empezó a usar para resolver el problema del reconocimiento óptico de caracteres. En 1959, Bledsoe y Browning construyeron un sistema bayesiano para reconocimiento de texto basado en una red neuronal.

Durante los siguientes años y hasta comienzos de los 80´s, el paradigma estocástico jugó un papel principal en el desarrollo de algoritmos para reconocimiento de voz, particularmente el uso de Modelos Ocultos de Markov. Trabajaron bajo este paradigma Shannon, después Jelinek, Bahl, Mercer y otros colegas de Centro de Investigación Thomas J. Watson en IBM y Baker en la Universidad de Carnegie Mellon, quien fue influenciado por el trabajo de Baum y sus colegas en Princeton.

El paradigma basado en la lógica fructificó en el lenguaje PROLOG así como las gramáticas funcionales y de unificación.

En lo que se refiere al entendimiento del lenguaje natural, Terry Winograd desempeñó un papel importante con su sistema SHRDLU en 1972, simulando un robot dentro de un

mundo de bloques de juguete y que era capaz de aceptar comandos de texto en lenguaje natural.

Los modelos de estados finitos reaparecieron entre 1983 y 1993. Durante este período hubo algo conocido como “el regreso del empirismo” en donde se les dio un mayor uso a los modelos probabilísticos en el procesamiento del habla y el lenguaje. También se vio gran auge en lo que a generación del lenguaje natural se refiere.

Finalmente, en los últimos años del siglo XX, se estandarizó el uso de la probabilidad en las herramientas para el procesamiento del lenguaje y del habla. Muchos algoritmos de parseo y otros métodos empezaron a incorporar probabilidades y se mejoraron considerablemente las técnicas.

Dentro de ese contexto esta tesis presenta, en el primer capítulo, una justificación del tema escogido dando la importancia que se merece a este tipo de tecnologías del habla. Se muestra parte de la teoría matemática involucrada en la construcción de los sistemas reconocedores de habla y su funcionamiento en general. Además, se menciona el estado del arte de las tecnologías del habla y se dan un conjunto de notas históricas sobre los inicios y el desarrollo que ha tenido esta tecnología a lo largo de 50 años de investigaciones.

En el capítulo dos, se presenta el corpus Dimex100: un nuevo corpus de habla y fonético para el español de México [13], su recopilación, características y estadísticas; que, como se verá, es un valioso recurso para la construcción de reconocedores. De igual forma, se muestra la importancia de los conocimientos lingüísticos involucrados en la construcción de este tipo de sistemas.

En el capítulo tres, se trata la construcción del reconocedor de voz utilizando Sphinx el software de la Universidad de Carnegie Mellon, la importancia de los datos que se requieren para su construcción como son los modelos acústicos, el diccionario de pronunciación y el modelo de lenguaje, así como las diferentes técnicas empleadas para crearlos.

En el capítulo cuatro, se presentan los experimentos realizados, basados en la cantidad de datos que se tienen, mostrando las dificultades encontradas durante el desarrollo de las pruebas y comparando los resultados entre los diferentes experimentos.

Finalmente, en el capítulo cinco, se encuentran las conclusiones, comentarios y consideraciones surgidas a lo largo del desarrollo de este trabajo.

CAPÍTULO 2

DIMEx100

A pesar de los recientes progresos en reconocimiento de voz, la disponibilidad de un corpus fonético para la creación de modelos acústicos en español es todavía muy limitada[1]. La creación de este tipo de recursos es necesaria por varias razones, por ejemplo, que los TTSs o sistemas de texto a voz necesitan ser enfocados a comunidades lingüísticas específicas, y hay que considerar los modelos acústicos para los alófonos más comunes del dialecto de manera que se pueda incrementar el reconocimiento. Un conjunto de alófonos motivado lingüística y empíricamente es también importante para la creación de diccionarios de pronunciación. El español de México, por ejemplo, es un lenguaje con 22 fonemas (17 consonantes y 5 vocales) como se indican en las tablas 1 y 2. En la primera columna (izquierda a derecha) aparecen los modos de articulación y en la primera fila los puntos de articulación.

Consonantes	Labiales	Labiodentales	Dentales	Alveolares	Palatales	Velares
Oclusivas sordas	p		t			k
Oclusivas sonoras	b		d			g
Africada sorda		f			tS	
Fricativa sorda				s		x
Fricativa sonora					Z	
Nasales	m			n	n~	
Vibrantes				r(/ r		
Laterales				l		

Tabla 1. Fonemas del español de México (consonantes)

[1] Entre los pocos corpus disponibles podemos mencionar el Latino-40 (http://www.ldc.upenn.edu), el ALBAYZIN corpus para el español de castilla y el Spanish Speech Corpus 1 (http://www.elda.fr).

Vocales	Anteriores	Media	Posteriores
Cerradas	i		u
Medias	e		o
Abiertas		a	

Tabla 2. Fonemas del español de México (vocales)

El modo de articulación se refiere a la postura que adoptan los órganos que producen los sonidos. Si los órganos cierran total y momentáneamente la salida del aire los sonidos son oclusivos; cuando la salida de aire provoca fricción al atravesar el paso o estrechamiento formado por los órganos de la cavidad bucal los sonidos son fricativos; si hay un primer momento de cierre seguido de otro de fricción o roce los sonidos son africados; cuando el aire sale tanto por las fosas nasales como por la boca y además es de cierre (como los oclusivos) el sonido es nasal; un sonido vibrante se caracteriza porque la lengua toca una o repetidas veces los alvéolos; finalmente si el aire se escapa por los lados de la lengua entonces tenemos un sonido lateral . En el caso de las vocales, se refiere a la abertura de la boca al pronunciarlas.

El punto de articulación es el lugar donde toman contacto los órganos que intervienen en la producción del sonido. Si para producir un sonido entran en contacto los labios se crean sonidos labiales; si se produce un contacto del labio inferior con los incisivos superiores entonces son labiodentales; si se utilizan los dientes superiores y la lengua son dentales; si interviene la lengua y los alvéolos los sonidos son alveolares; si se producen en la zona del paladar duro los sonidos son palatales; finalmente si interviene la lengua y el velo del paladar el sonido es velar. En el caso de las vocales es la parte de la boca donde se articulan.

La sonoridad se obtiene de las cuerdas vocales, si las cuerdas vocales no vibran los sonidos se llaman sordos, de lo contrario son sonoros [2].

Como resultado del trabajo empírico realizado para el centro del país [3], se identificaron 37 alófonos (26 sonidos consonánticos y 11 vocales y semi-consonantes) que son los que se

encuentran en cantidad suficiente para la creación de modelos acústicos [13]. Este conjunto se muestra en las tablas 3 y 4.

Consonantes	Labiales	Labiodentales	Dentales	Alveolares	Palatales	Velares
Oclusivas sordas	p		t			k
Oclusivas sonoras	b		d			g
Africada sorda		f			tS	
Africada sonora					dZ	
Fricativa sorda			s_[	s		x
Fricativa sonora	V		D	z	Z	G
Nasales	m		n_[	n	n~	N
Vibrantes				r(/ r		
Laterales				l		

Tabla 3. Consonantes del Mexbet (Cuétara, 2004)

Vocales	Anteriores	Media	Posteriores
	j		w
Cerradas	i		u
Medias	e		o
	E		O
Abiertas	a_j	a	a_2

Tabla 4. Vocales del Mexbet (Cuetara, 2004)

Otros corpus para el español de México, por ejemplo el Tlatoa (Kirshning, 2001 y Gamboa, 2002), solo consideran los principales fonemas del lenguaje por lo que tienen conflictos de criterio para la trascripción de algunos sonidos consonánticos y semi-consonánticos. Para una discusión adicional ver [3], [13].

2.1 Características del Corpus

Un nuevo corpus de habla y fonético para el español de México

Con el objetivo de contar con un recurso útil en el desarrollo de tecnologías de habla en México, se creó el Corpus DIMEx100. Este corpus está formado por un conjunto de 5010 frases seleccionadas del Corpus230 [13], que cuenta con 344 619 frases, 235 891 unidades léxicas y 15 millones de palabras, que a su vez fue seleccionado de Internet y cuyas frases estaban ordenadas de menor a mayor valor de perplejidad. La perplejidad es, intuitivamente, una medida del número de unidades lingüísticas que pueden seguir a una unidad de referencia. En relación al corpus, por ejemplo, la más baja perplejidad de una palabra es el menor número de palabras diferentes que es probable que la sigan en una frase, por lo tanto, los enunciados con una baja perplejidad están formados por palabras con un alto poder discriminatorio o un alto contenido de información y son precisamente esas frases las que constituyen en su mayoría el corpus DIMEx100. Además, el corpus fue editado manualmente eliminándose palabras extranjeras y abreviaciones de modo que las frases fueran de fácil lectura. Estas frases fueron grabadas por 100 hablantes; 50 diferentes frases para cada uno más 10 comunes que fueron grabadas por todos para dar en total 6000 frases repartidas en 100 carpetas. Para medir lo apropiado del corpus aplicado a futuros trabajos, se controlaron las características de los hablantes; por otro lado, también se midió la cantidad y distribución de muestras por cada unidad fonética.

2.2 Niveles de etiquetación

En un principio los 37 alófonos del trabajo mencionado anteriormente [3] fueron considerados para las transcripciones y diccionarios fonéticos del corpus DIMEx100. Con esto sería posible la construcción de los modelos acústicos para el reconocedor de habla.

Se definieron entonces los siguientes niveles de etiquetación:

- Un nivel fonético: alófonos

- Un nivel fonológico: fonemas
- Un nivel ortográfico: palabras

La herramienta que se escogió para el proceso de etiquetación es SpeechView, este software permite ver el espectrograma, el oscilograma y el pitch de una señal acústica (figura 4). SpeechView forma parte del toolkit CSLU (Center for Spoken Language and Understanding) de la Universidad de Oregon.

Se utilizó, además, la herramienta *TranscríbEMex* (Cuétara y Villaseñor, 2004) para obtener un etiquetado automático con alineación default y de esa manera ayudar a los etiquetadores durante el proceso de etiquetación. *TranscríbEMex* genera, a partir de una palabra o frase, la correspondiente trascripción alofónica, fonética y de sílabas.

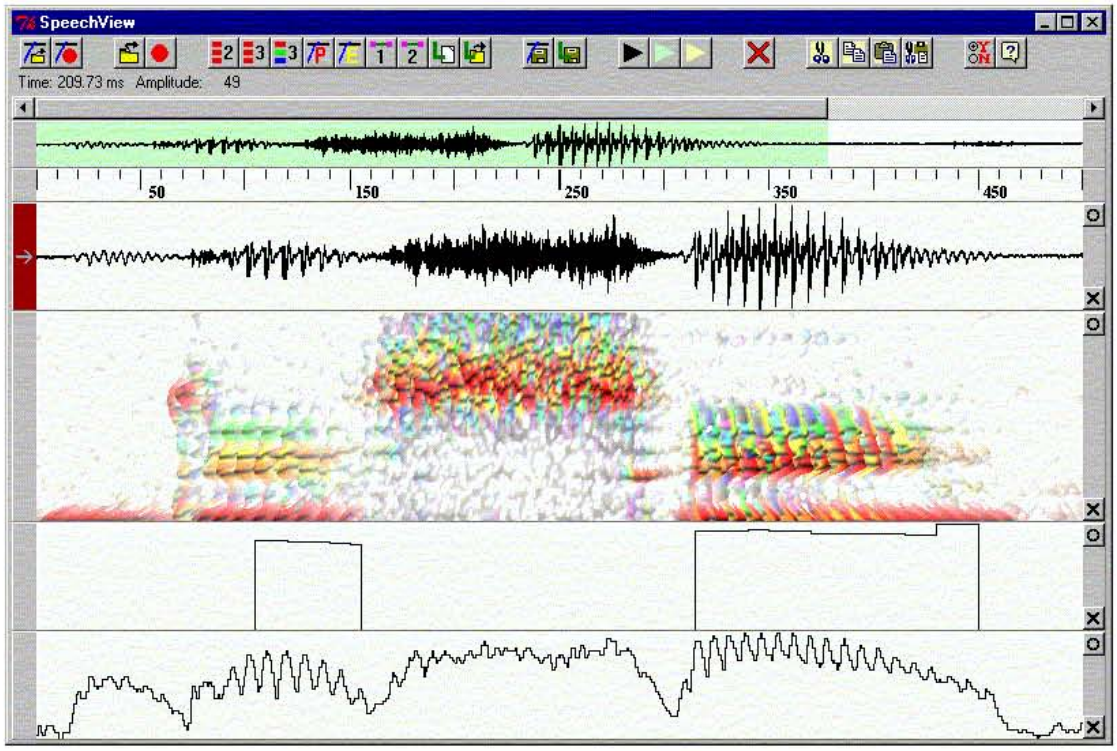


Figura 3. SpeechView

Después de un tiempo de estar trabajando el corpus, las personas involucradas plantearon un problema con el nivel ortográfico: ¿a cuál de los otros dos niveles se tenían que alinear las palabras? Entonces se discutió qué representaba cada nivel. En el nivel fonético se tiene la realización que realmente produjo el hablante de la cadena hablada. En el nivel fonológico se tiene la abstracción del lenguaje, es decir, las unidades abstractas cuya alteración produce un cambio en el significado. Tomando en consideración las dos representaciones, se llegó a la conclusión de que el nivel fonológico no era útil para la construcción de diccionarios de pronunciación por no ser coherente con lo que el hablante produce en realidad (se consideró que la trascripción debería permitir la construcción automática de los diccionarios de pronunciación), y entonces se decidió reestructurar el proceso de etiquetación. Mas tarde, surgió otra situación que tuvo como consecuencia un nuevo cambio en los niveles de etiquetación: la inclusión de acentos y cierres.

Como resultado de estas reflexiones, y con la finalidad de estudiar el impacto en el reconocimiento debido a diferentes niveles de granularidad, se definió la etiquetación del corpus de la siguiente manera[2]:

- Una trascripción fina (T54)
- Una trascripción media (T44)
- Una trascripción básica, T22)
- Un nivel ortográfico (palabras, Tp)

En el nivel T54 se observa una segmentación lo más fina posible a fin de consignar una mayor cantidad de datos acústicos. Se tienen los 37 alófonos del Mexbet, más 8 cierres de oclusivas y africadas (p_c, t_c, k_c, b_c, d_c, g_c, tS_c, dZ_c), más 9 vocales que pueden recibir acentos (i_7, e_7, E_7, a_j_7, a_7, a_2_7, O_7, o_7 y u_7). Se consideran fenómenos como acentuación y asimilación de sonidos (en un sentido restringido, "asimilación" consiste en la conversión de un fonema en otro por la influencia del que le sigue o del que precede).

[2] Niveles de representación segmental en el Proyecto DIME-II, elaborado por el Maestro Javier Cuétara Priede (2005).

A éste nivel de etiquetación se alinean los niveles restantes. El alfabeto fonético para esta trascripción se encuentra en las tablas 5, 6 y 7.

Consonantes	Labiales	Labiodentales	Dentales	Alveolares	Palatales	Velares
Oclusivas sordas	p/p_c		t/t_c			k/k_c
Oclusivas sonoras	b/b_c		d/d_c			g/g_c
Africada sorda		f			tS/tS_c	
Africada sonora					dZ/dZ_c	
Fricativa sorda			s_[	s		x
Fricativa sonora	V		D	z	Z	G
Nasales	m		n_[	n	n~	N
Vibrantes				r(/ r		
Laterales				l		

Tabla 5. Consonantes del nivel T54

Vocales tónicas	Anteriores	Media	Posteriores
Cerradas	i_7		u_7
Medias	e_7		O_7
Medias abiertas	E_7		O_7
Abiertas	a_j_7	a_7	a_2_7

Tablas 6. Vocales tónicas del nivel T54

Vocales átonas	Anteriores	Media	Posteriores
Paravocales	j		w
Cerradas	i		u
Medias	e		o
Medias abietas	E		O
Abiertas	a_j	a	a_2

Tablas 7. Vocales átonas del nivel T54

En el nivel T44 se considera una trascripción media, donde se conservan los 22 fonemas del español de México, los cierres de las oclusivas y la africada sorda (desaparece la africada

sonora), los alófonos aproximantes de las oclusivas sonoras (V, D, G), las 9 vocales que pueden recibir acento, las paravocales (j, w) y 5 símbolos para ciertas parejas de consonantes en posición final de sílaba o coda silábica (-B, -D, -G, -N, -R).

Para éste nivel se realiza una etiquetación semi-automática que se obtiene del nivel T54 (ver apéndice 1). Manualmente se marcan los símbolos de las codas silábicas. El alfabeto fonético para esta trascripción se observa en las tablas 8, 9, 10 y 11.

Consonantes	Labiales	Labiodentales	Dentales	Alveolares	Palatales	Velares
Oclusivas sordas	p/p_c		t/t_c			k/k_c
Oclusivas sonoras	b/b_c		d/d_c			g/g_c
Africada sorda		f			tS/tS_c	
Africada sonora						
Fricativa sorda				s		x
Fricativa sonora	V		D		Z	G
Nasales	m			n	n~	
Vibrantes				r(/ r		
Laterales				l		

Tablas 8. Consonantes del nivel T44

Vocales tónicas	Anteriores	Media	Posteriores
Cerradas	i_7		u_7
Medias	e_7		o_7
Abiertas		a_7	

Tablas 9. Vocales tónicas del nivel T44

Vocales átonas	Anteriores	Media	Posteriores
Paravocales	j		w
Cerradas	i		u
Medias	e		o
Abiertas		a	

Tablas 10. Vocales átonas del nivel T44

	Coda silábica
Labiales p/b	-B
Dentales t/d	-D
Velares k/g	-G
Nasales n/m	-N
Vibrantes r/r(	-R

Tablas 11. Codas silábicas del nivel T44

En el nivel T22 se segmentan única y exclusivamente los 22 fonemas del español de México (Tabla 1). Esta etiquetación se realiza de forma automática a partir del nivel T54 (ver apéndice 1).

El nivel Tp se refiere a la trascripción ortográfica de palabras. Su objetivo es la creación de diccionarios de pronunciación de manera automática (ver apéndice 3).

2.3 Características técnicas

Las grabaciones se realizaron en el Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET-UNAM), en una cabina insonorizada, con el software *Wave Lab,* una tarjeta de sonido *Sound Blaster Audigy Platinum ex* (24 bit/96KHz/100db SNR) y un micrófono de condensación con diafragma sencillo.

Cada frase fue grabada en un formato mono estereo a 16 bits y con un periodo de muestreo de 44.1 Khz.

2.4 Características socio-lingüísticas de los hablantes

Siguiendo a Perissinotto (1975), los hablantes fueron seleccionados de acuerdo a edad (16 a 36 años de edad), nivel de educación (estudios mayores a secundaria) y lugar de origen (Ciudad de México). Un gran porcentaje de los hablantes fueron de la comunidad de ciudad universitaria: investigadores, estudiantes, maestros y trabajadores. El promedio de edad fue

de 23 años, la mayoría de los hablantes era de nivel licenciatura (87%) y el resto graduados y también en su mayoría los hablantes nacieron y viven en la ciudad de México (82%). Sólo 18 personas de otros lugares residiendo en la ciudad de México participaron en las grabaciones. El corpus resultó balanceado en género 49 fueron hombres y 51 mujeres. Aún cuando el español de México tiene varios dialectos, el dialecto de la ciudad de México es el más hablado en la población del país.

2.5 Características Lingüísticas del corpus

En adición a definir el conjunto de símbolos para cada nivel de etiquetación (sección 2.2), se identificó el contexto en el cual ocurren todos ellos (Tabla 12). Estos contextos han sido caracterizados a través de reglas fonotácticas, por ejemplo, la vocal /a/ puede ser realizada como velar en frente de otro sonido velar y en frente de /l/ en coda silábica: alto; la palatal /a/ es realizada en frente de sonidos palatales, y la central /a/ en cualquier otro lugar. Esto muestra que la variación alofónica del español puede ser modelada con reglas fonotácticas como lo apoyan Moreno y Mariño [3].

	Fonemas	***Alófonos***	***Contexto***
Bilabial oclusiva sorda	/p/	p_c p	En todos lo casos
Dental oclusiva sorda	/t/	t_c t	En todos los casos
Velar oclusiva sorda	/k/	k_c k	En todos los casos
	/k/	k_c k_j	_{e, i, j}
Bilabial oclusiva sonora	/b/	b_c b	///_
	/b/	b_c b	{m, n}_
	/b/	V	En todos los demás casos
Dental oclusiva sonora	/d/	d_c d	///_
	/d/	d_c d	{m, n}_
	/d/	D	En todos los casos
Velar oclusiva sonora	/g/	g_c g	///_
	/g/	g_c g	{m, n}_
	/g/	G	En todos los demás casos
Palatal africada sorda	/tS/	tS_c t	En todos los casos
Labiodental fricativa	/f/	f	En todos los casos

Alveolar fricativa sorda	/s/	z	v_v
	/s/	z	_{b,d,g,Z,m,n,n~,l,r,r(}
	/s/	s_[	_{t}
	/s/	s	En todos los demás casos
Velar fricativa sorda	/x/	x	En todos los casos
Palatal fricativa sonora	/z/	dZ_c dZ	///_
	/z/	dZ_c dZ	{m, n}_
	/z/	Z	En todos los demás casos
Nasal bilabial	/m/	m	En todos los casos
Nasal alveolar	/n/	n_[	_{t, d}
	/n/	N	_{k, g}
	/n/	n	En todos los demás casos
Nasal palatal	/n~/	n~	En todos los casos
Lateral alveolar	/l/	l	En todos los casos
Vibrante simple	/r(/	r(	En todos los casos
Vibrante múltiple	/r/	r	En todos los casos
Vocal alta palatal	/i/	j	_{a, e, o , u}
	/i/	j	{a, e, o, u}_
	/i/	i	En todos los demás casos
Vocal media palatal	/e/	E	_{r}
	/e/	E	{r}_
	/e/	E	_{p, t, k, b, d, g, tS, f, x, Z, l, r(}$
	/e/	e	En todos los demás casos
Vocal abierta	/a/	a_2	_{u, x}
	/a/	a_2	_{l}$
	/a/	a_j	_{tS, n~,Z,j}
	/a/	a	En todos los demás casos
Vocal media velar	/o/	O	_{r}
	/o/	O	{r}_
	/o/	O	_{consonante}$
	/o/	o	En todos los demás casos
Vocal alta velar	/u/	w	_{a, e, o ,i}
	/u/	w	{a, e, o, i}_
	/u/	u	En todos los demás casos

Tabla 12. Fonemas y alófonos para el español de México y contextos en los que se observan Error! Reference source not found..

En la tabla 12 se muestran varios símbolos en la columna de contexto (lado derecho), estos símbolos representan:

_	Posición
///_	Inicio absoluto de palabra
_ { }	Posición anterior
{ }_	Posición posterior
v_v	intervocálico
$	Final de sílaba

2.6 Estadísticas del corpus

2.6.1 Estadísticas de todo el corpus DIMEx100

Dentro de sus 6000 frases, el corpus DIMEx100 tiene 59812 palabras de las cuales 8715 son diferentes y entre ellas, las más representadas son: *de, la, el,* y, *en*. (Figura 5).

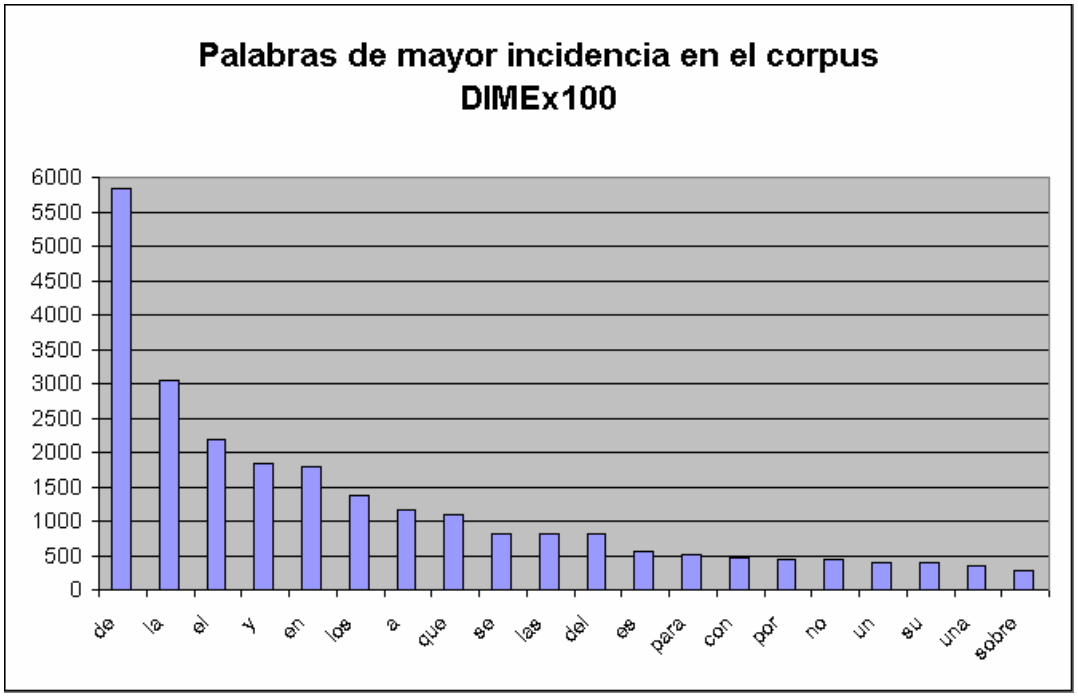


Figura 4. Palabras de mayor incidencia en al corpus DIMEx100

2.6.2 Estadísticas de los datos que se ocuparon para el desarrollo del reconocedor de habla.

Para realizar el reconocedor de habla, ya se contaba con 35% del corpus DIMEx100 etiquetado. De éste 35%, se ocupó 30% para entrenar y el resto para realizar las pruebas correspondientes.

Dentro de las 30 carpetas que se ocuparon para el entrenamiento tenemos 1500 frases con un total de 15226 palabras de las cuales 3546 son diferentes.

Así como lo muestran las estadísticas de todo el corpus, las palabras más representadas en esta muestra son: *de, la, el, en*, *y*. (Figura 6).

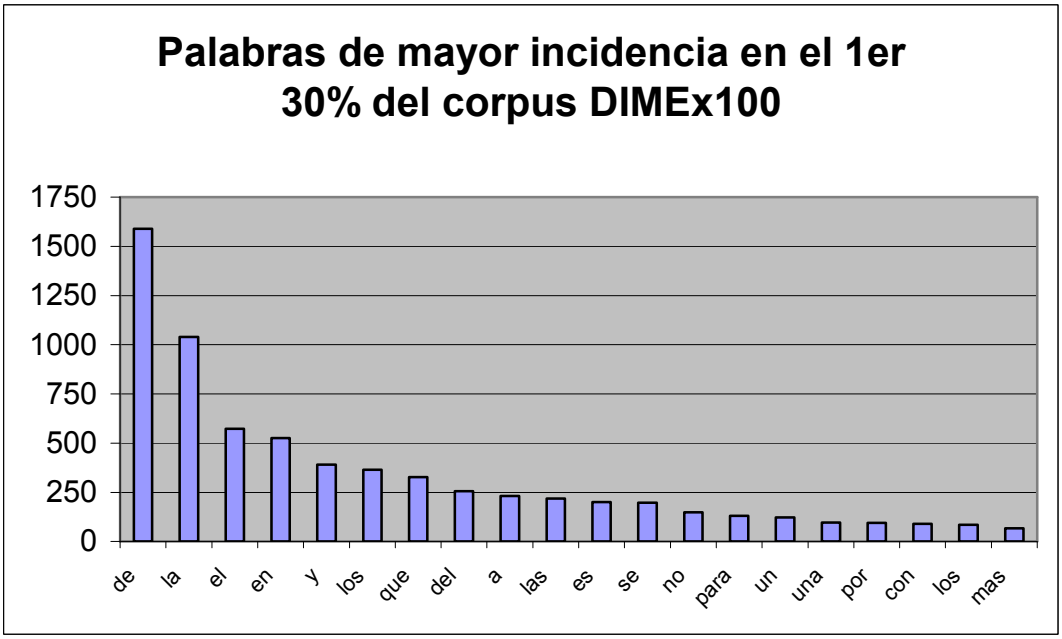


Figura 5. Palabras de mayor incidencia en los datos de entrenamiento

También se obtuvieron las diferentes distribuciones fonéticas de acuerdo a cada nivel de etiquetación. Las gráficas de estas distribuciones se muestran en las figuras 7, 8, 9, 10 y 11.

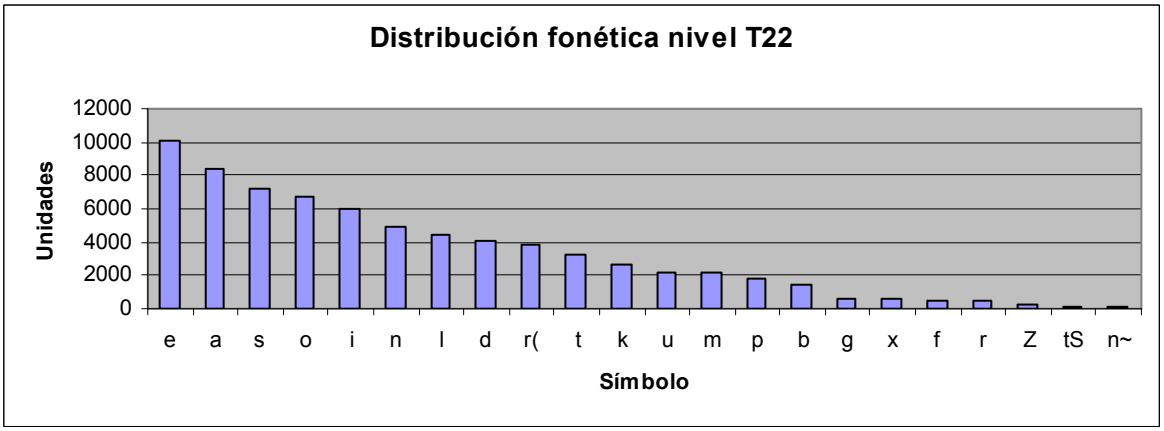


Figura 6. Distribución fonética del nivel T22 en los datos de entrenamiento

En la gráfica del nivel T22 se observa que todos los símbolos tienen más de 50 muestras por lo que se considera cantidad suficiente para construir los modelos acústicos a este nivel.

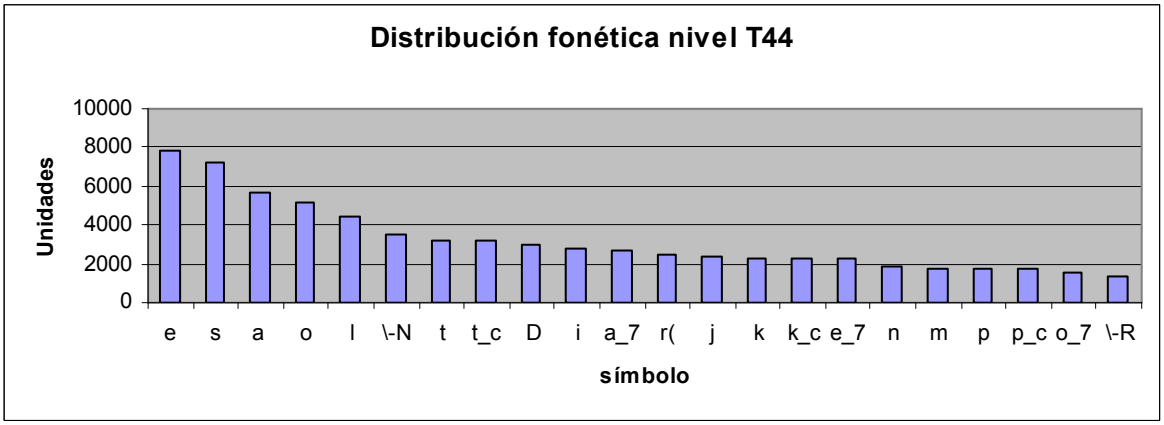


Figura 7. Distribución fonética del nivel T44 en los datos de entrenamiento

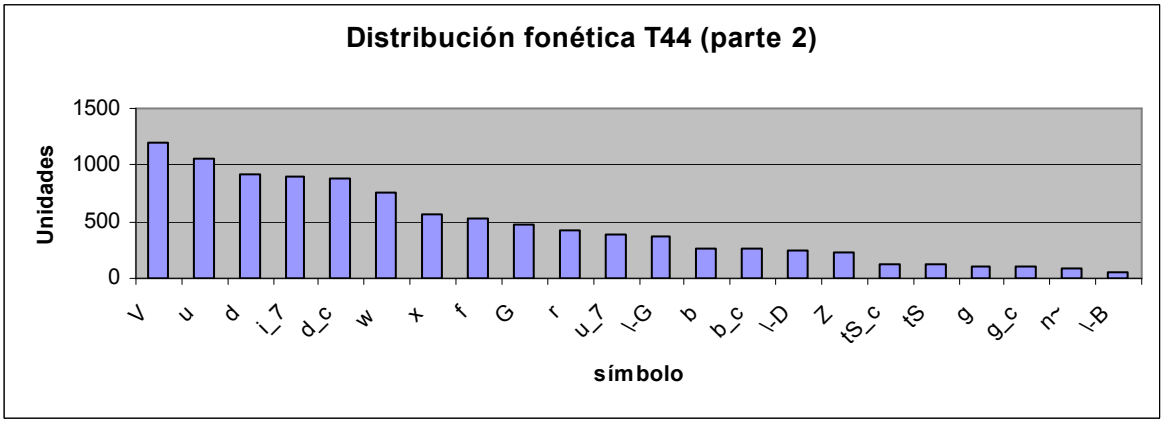


Figura 8. Distribución fonética del nivel T44 en los datos de entrenamiento (parte 2)

En las gráficas del nivel T44 el símbolo con menor número de unidades es –B que tiene 57; sin embargo, también se podrían construir modelos acústicos usando el nivel T44.

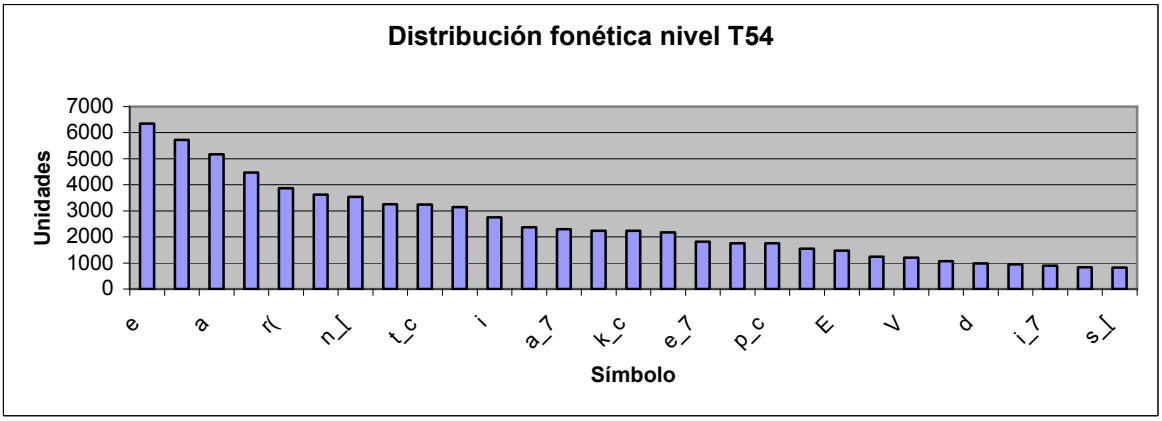


Figura 9. Distribución fonética del nivel T54 en los datos de entrenamiento

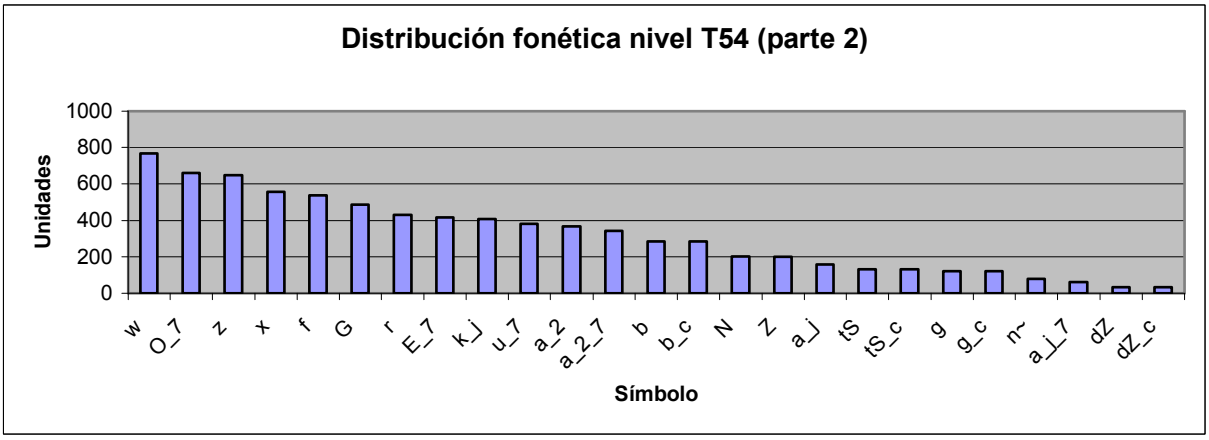
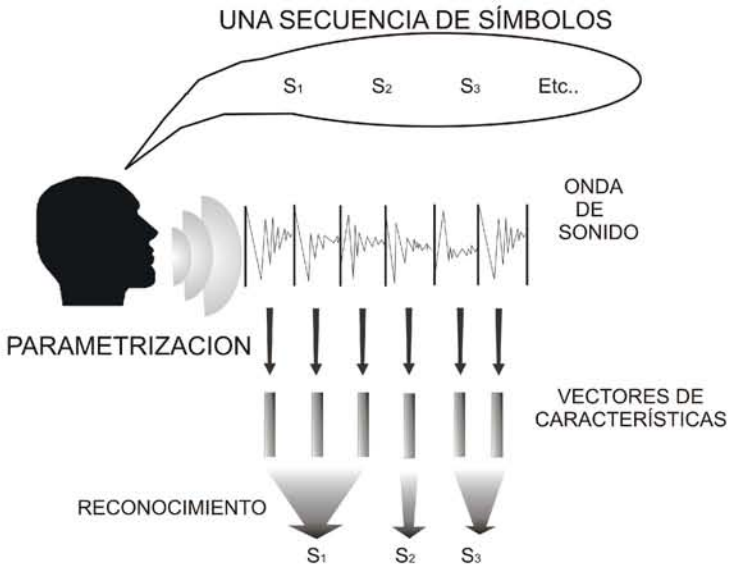


Figura 10. Distribución fonética del nivel T54 en los datos de entrenamiento (parte 2)

Finalmente en las gráficas del nivel T54 se observa que hay dos símbolos (dZ y dZ_c) que solo alcanzan 31 muestras cada uno por lo que no es recomendable construir modelos acústicos a este nivel de granularidad con únicamente 30% del corpus.

CAPÍTULO 3

Construcción de un reconocedor de voz



3.1 Elementos básicos de un reconocedor

En el primer capítulo de esta tesis, se mostró que la intuición matemática de un sistema reconocedor de voz esta dada por la ecuación:

$$W = \arg\max_{W \in L} \underbrace{P(O|W)}_{\text{modelo acustico}} \quad \underbrace{P(W)}_{\text{modelo del lenguaje}}$$

En esta ecuación podemos identificar que los elementos básicos de un sistema reconocedor de voz son:

1. El modelo acústico.
2. El modelo de lenguaje.
3. El diccionario de pronunciación.

A continuación se presenta un esquema de cómo se relacionan estos elementos.

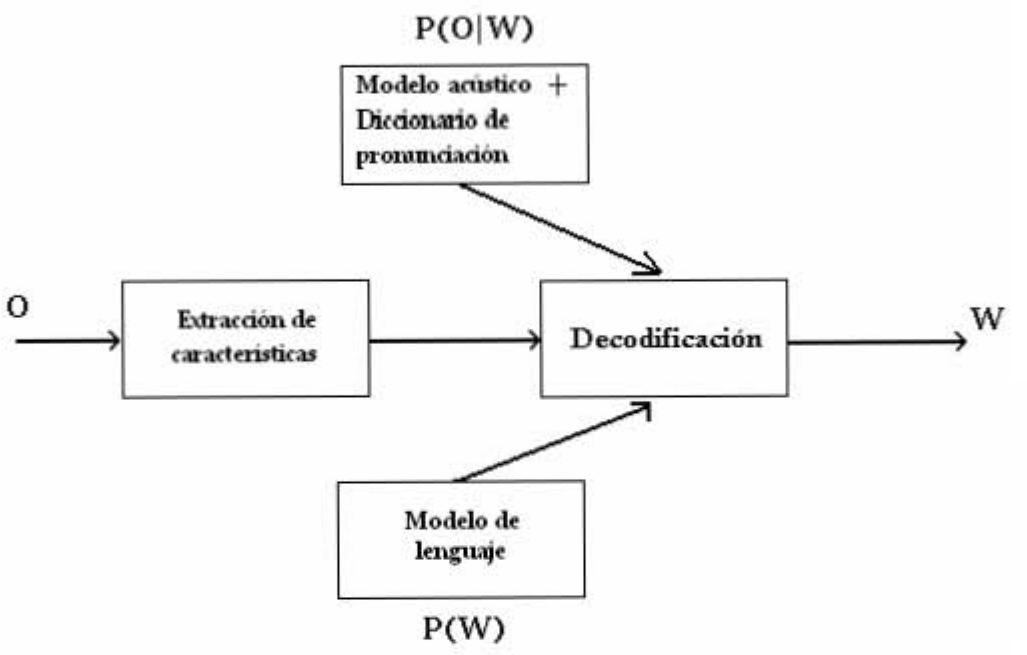


Figura 11. Estructura de reconocimiento de acuerdo a la teoría del canal ruidoso

3.1.1 Modelos acústicos

Los modelos acústicos se pueden ver como filtros que contienen la variabilidad acústica de una lengua. Dependiendo de las expectativas de los modelos, y dada una señal de entrada se obtiene la hipótesis de lo que el hablante ha dicho.

El modelo acústico captura las propiedades acústicas de la señal de entrada, obtiene un conjunto de vectores de características que después compara con un conjunto de patrones que representan símbolos de un alfabeto fonético y arroja los símbolos que más se parecen. Esta es la intuición del proceso matemático probabilístico llamado modelo oculto de Markov; este tipo de modelo es el que se utilizó para la creación de los modelos acústicos en el presente trabajo.

El modelo acústico incluye:

1. El análisis acústico, en el cuál se caracteriza a la señal de entrada en una secuencia de vectores acústicos.
2. Los modelos acústicos para las unidades que forman las palabras (por ejemplo, fonemas, que usualmente son modelados dependientes del contexto).
3. El diccionario de pronunciación, en el cuál se define la descomposición de las palabras en unidades más pequeñas que corresponden a las unidades dentro del alfabeto fonético definido.

El análisis acústico (figura 12) se puede realizar mediante diferentes técnicas de filtrado, entre ellas se encuentran las técnicas LPC (*Linear Predictive Coding*) o MFCC (*Mel Frequency ceptral Coefficients*).

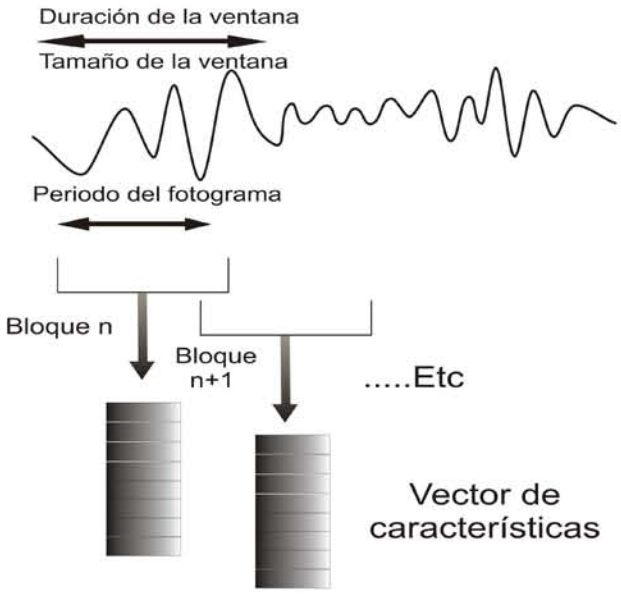


Figura 12. Extracción de características.

LPC

Una aproximación común para estimar las resonancias en variación de tiempo del tracto vocal, en un archivo de audio, se hace con **predicción lineal**. En este análisis, las propiedades de la composición espectral de la radiación, tracto vocal y excitación glotal son representadas mediante un filtro digital de variación de tiempo.

Lo que se asume en esta técnica es que la señal de habla es producida por un *buzzer* al final de un tubo. La glotis (el espacio entre las cuerdas vocales) produce el *buzz* el cual es caracterizado por su intensidad (*loudness*) y frecuencia (*pitch*). El tracto vocal (la garganta y la boca) forma el tubo el cual es caracterizado por su resonancia y a eso se le llama los formantes.

LPC analiza la señal mediante la estimación de los formantes, removiendo su efecto de la señal de habla y obteniendo, finalmente, la intensidad y frecuencia del *buzz* remanente.

MFCC´s

Para calcular estos coeficientes o características espectrales, en resumen, la técnica consiste en los siguientes pasos de acuerdo a L. Rabiner y Juang (1993):

1. Se divide la señal en fotogramas de cierto número de milisegundos.
1. Por cada fotograma se obtiene el espectro de amplitud.
2. Se calcula el logaritmo del espectro.
3. Se convierte a espectro de Mel.
4. Se calcula la transformada de coseno discreto.

Existen diversas técnicas para construir modelos acústicos [14], como son:

- Técnicas topológicas: *Dynamic Time Warping* (DTW), basado en la teoría de clusters y comparación de distancias entre los mismos.
- Técnicas probabilísticas: Modelos ocultos de Markov (HMM), que son modelos generativos de las palabras del vocabulario.
- Redes neuronales.

TECNICAS TOPOLÓGICAS

Las técnicas topológicas, para el caso del reconocimiento de voz, consisten fundamentalmente en un análisis de la señal acústica a fin de obtener un conjunto o un vector de parámetros (que puede ser acústico, coeficientes espectrales, etc.), que se ven

como un punto en un espacio n-dimensional. El conjunto de puntos en el espacio con las mismas características forma un cluster. Varios clusters nos representan el conjunto de modelos acústicos para el sistema reconocedor de voz.

Dada una señal de entrada, ésta se parametriza y el sistema reconocedor la compara con cada cluster que tiene de referencia calculando distancias en el espacio n-dimensional de los clusters creados; el símbolo de salida es el que corresponde al cluster al que más se aproxime.

El DTW es la primera técnica que ha permitido sacar al mercado productos de reconocimiento de voz al con resultados favorables.

TECNICAS PROBABILÍSTICAS

Un enfoque alternativo al de medir distancias entre clusters es el uso de Modelos Ocultos de Markov (HMM) (Baum *et al*, 1966-72).

Un proceso estocástico se define como un conjunto de variables aleatorias X_t cuya distribución varía con respecto a un parámetro, generalmente el tiempo. La variable t toma valores de un subconjunto de números enteros o reales no negativos. Las variables aleatorias X_t toman valores en un conjunto que se denomina *espacio de estados*.

Un HMM es un proceso estocástico donde el cambio de estado sólo depende del estado actual y no de los anteriores; es una máquina de estados finitos en la cuál el siguiente estado depende únicamente del estado actual, y asociado a cada transición entre estados se produce un vector de observaciones o parámetros (correspondiente a un punto del espacio n-dimensional como se vio en el caso de las técnicas topológicas). Se llama HMM debido a que lleva asociados dos procesos: uno oculto (no observable directamente) correspondiente a las transiciones entre estados, y otro observable (y directamente relacionado con el primero), cuyas realizaciones son los vectores de parámetros que se producen desde cada estado y que forman el patrón a identificar.

Para aplicar la teoría de los HMM´s en reconocimiento de voz, se representa cada palabra del vocabulario del reconocedor como un modelo generativo (que se calculara en la fase de entrenamiento) y posteriormente, se calcula la probabilidad de que la palabra a reconocer haya sido producida por algunos de los modelos de la base de datos del reconocedor. Para ello, se asume que durante la pronunciación de una palabra el aparato fonador puede adoptar sólo un número finito de configuraciones articulatorias o estados, y que desde cada uno de esos estados se producen uno o varios vectores de observación cuyas características espectrales dependerán (probabilísticamente) del estado en el que se hayan generado. Vista la generación de la palabra, las características espectrales de cada fragmento de señal dependen del estado activo en cada instante, y las del espectro de la señal durante la pronunciación de una palabra dependen de la función de transición entre estados.

Los tres problemas básicos con los que debe lidiar un HMM [15] son:

1. Evaluación: ¿Cómo puede calcular eficientemente $P(O|W)$?
2. Decodificación: ¿Cómo escoger la secuencia de estados que sea optima en algún sentido?
3. Aprendizaje: ¿Cómo ajustar los parámetros del modelo para maximizar $P(O|W)$?

Dada una secuencia de observación O = o_1, o_2,…, o_T y creados los modelos acústicos, $P(O|W)$ se calcularía obteniendo todas las posibles secuencias de estados de longitud T. En cada tiempo t = 1, 2,..., T se tienen N posibles estados alcanzables, por lo tanto N^T operaciones lo que es un número exponencial. El algoritmo de avance (*forward algorithm*) es una manera más eficiente para calcular estas probabilidades ya que calcula la probabilidad de la secuencia de observación parcial o_1 ,o_2..., o_t en el estado i hasta el tiempo t, dado el modelo de markov; de esta manera el número de operaciones se reduce a N^2T.

El problema de decodificación se resuelve mediante el algoritmo de Viterbi (usado por primera vez para reconocimiento de voz por Vintsyuk [8]) que es una simplificación del algoritmo de avance en donde sólo se consideran las probabilidades acumuladas en los

caminos optimos, es decir, sólo se considera la secuencia de estados que resulta de maximizar las expresiones del algoritmo de avance.

Finalmente, el problema del aprendizaje se resuelve con el algoritmo Baum-Welch (Baum, 1972), que es un caso especial del algoritmo de maximización de la expectativa (*Expectation Maximization*, *Dempster, Laird, and Rubin*). El algoritmo Baum-Welch permite entrenar las probabilidades de transición entre los estados y las probabilidades que emite cada estado de manera iterativa usando los algoritmos de avance y retroceso (*forward and backward algorithms*)

El HMM que es el actual responsable de las más exitosa tecnología en muchos reconocedores de habla (HTK y SPHINX son muestra de ello) y otras aplicaciones de procesamiento de voz, es el modelo de izquierda a derecha (figura 14). Su éxito se debe a su rigurosa formulación matemática para determinar de manera óptima los parámetros de un modelo dado y a su conformidad con el principio de optimalidad (Bellman, 1957: “dada una secuencia óptima de decisiones, toda subsecuencia de ella es, a su vez, óptima”) aplicado al reconocimiento de patrones de manera estadística.

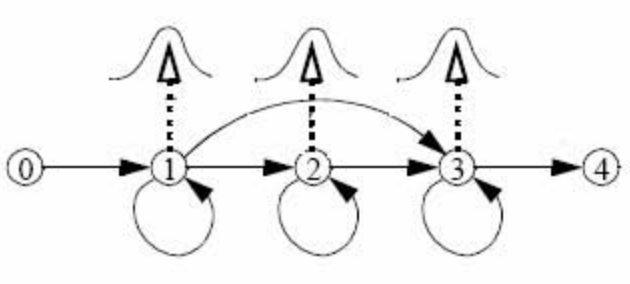


Figura 13. Modelo Oculto de Markov (*left to rigt)*

Actualmente, la mayoría de los reconocedores en funcionamiento se basan en técnicas estadísticas pues son de las que requieren menos memoria física y tienen un mejor tiempo de respuesta.; sin embargo, necesitan de una fase de entrenamiento que es mucho más lenta y costosa pero que se realiza una sola vez, por lo que el precio parece valer la pena.

REDES NEURONALES

Una red neuronal es una forma de sistema computacional multiprocesador que contiene:

- Elementos de procesamiento simple.
- Un alto grado de interconexión.
- Mensajes escalares simples.
- Capacidad de interacción entre elementos.

Los sistemas de reconocimiento basados en redes neuronales pretenden, interconectando un conjunto de unidades de proceso (o neuronas) en paralelo (de forma similar que en la mente humana), obtener funciones de reconocimiento similares a las humanas, tanto en tiempo de respuesta como en tasa de error. Esa forma de interconexión de las unidades de proceso es especialmente útil en aplicaciones que requieren una gran potencia de cálculo para evaluar varias hipótesis en paralelo, como sucede en los problemas de reconocimiento de voz.

Las unidades de proceso pueden ser de varios tipos: las más simples (y utilizadas) disponen de varias entradas y la salida es el resultado de aplicar alguna transformación no lineal a la combinación lineal de todas las entradas; otras, un poco más elaboradas, se caracterizan por disponer de memoria; en ellas la salida en cada momento depende de entradas anteriores en el tiempo.

Igual que se dijo para las técnicas anteriores, una red neuronal debe ser entrenada para resolver un tipo determinado de problemas. El algoritmo particular de entrenamiento dependerá de la estructura interna de las neuronas pero, en cualquier caso, el entrenamiento se llevará a cabo a partir de una base de datos etiquetada, como sucedía con los modelos de Markov, y será un proceso iterativo en el que se modifican los parámetros de la red para que ante un conjunto determinado de estímulos (patrones), produzca una respuesta determinada: la secuencia de símbolos reconocida.

Además del habla existe una gran diversidad de tipos de ruido que afectan el ambiente acústico de un sistema reconocedor de voz por lo que siempre es una ventaja tener un conocimiento *a priori* de la naturaleza del ruido. Es importante crear, además de los

modelos de las unidades acústicas, uno o más modelos de relleno entre los cuales el más significativo es el silencio; sin embargo, para evitar confusión entre los sonidos sordos (principalmente los fricativos) y segmentos sin habla contaminados con ruido, solo segmentos de más de 10 milisegundos son usados para entrenar el modelo del silencio (así lo hacen algunos *toolkits* como Sphinx). Así, dado que los sonidos fricativos u otros fonemas sordos no duran más de 10 milisegundos, el sistema rara vez los confundirá con intervalos de ruido puro.

3.1.2 Modelo de lenguaje

De acuerdo a Jurafski [8], el modelo del lenguaje contiene las propiedades lingüísticas del lenguaje y nos da la probabilidad a priori de una secuencia de palabras.

Adivinar la siguiente palabra, o lo que se llama *predicción de la palabra,* es una subtarea esencial en el reconocimiento de voz, reconocimiento de escritura, comunicación aumentativa para discapacitados o detección de errores de ortografía. En esas tareas, la identificación de una palabra es difícil debido a que la entrada es ruidosa o ambigua y es por eso que mirando la palabra o palabras previas, puede dar una pista importante sobre cuales pueden ser las siguientes.

Para la comunicación aumentativa la habilidad de predecir la palabra se vuelve esencial debido a que, como el sistema permite que personas discapacitadas hablen mediante la selección de palabras de un menú, no se puede tener todas las posibles palabras del idioma en una pantalla pero conociendo cuales palabras son las que posiblemente quiere el usuario, solo esas se le pondrían en el menú.

N-gramas

El problema de adivinar la siguiente palabra esta relacionado con el problema de calcular la probabilidad de una secuencia de palabras. Algoritmos que asignan probabilidades a un

enunciado pueden ser usados para asignar la probabilidad a la siguiente palabra en un enunciado incompleto y viceversa.

Este modelo de predicción de la palabra es conocido como N-grama. Un modelo de N-grama usa las N-1 palabras previas para predecir la siguiente, en otras palabras, es la probabilidad de una palabra dadas las N-1 palabras previas. En reconocimiento de voz es tradicional usar el término Modelo del Lenguaje para éstos modelos estadísticos de secuencias de palabras.

El modelo más simple posible de secuencia de palabras permitiría a cualquier palabra del lenguaje seguir a otra. En la versión probabilística de esta teoría, cada palabra tendría la misma probabilidad de seguir a otra palabra. Por ejemplo, si el idioma tuviera 100 000 palabras, la probabilidad de cualquier palabra de seguir a otra sería de 1/100 000, o bien, 0.00001.

La suposición de que la probabilidad de una palabra depende sólo de la palabra previa es llamada *suposición de Markov*. Es por eso que los modelos de Markov son la clase de modelos probabilísticas que suponen que se puede predecir la probabilidad de unidades futuras sin ver demasiado hacia atrás.

Las probabilidades de N-gramas, en determinado corpus, pueden ser calculadas mediante un simple conteo de N-gramas y después normalizando, lo que se conoce como la estimación de la máxima probabilidad [4]. La ventaja de los N-gramas es que ellos aprovechan en gran medida el conocimiento léxico. Una desventaja, en algunos casos, es que los N-gramas son muy dependientes del corpus con el que son entrenados, por lo tanto, dado que los corpus de entrenamiento son finitos es aceptable que no todas las combinaciones de palabras aparezcan y es seguro que se tenga un gran número de casos en donde la probabilidad sea cero y entonces se obtengan resultados muy malos. La tarea de reevaluar algunas de las probabilidades de los N-gramas con valores de cero o un número muy pequeño y asignarles valores diferentes de cero es llamada *smoothing o suavizado*.

Smoothing algorithms o **algoritmos de suavizado** [8], proporcionan una mejor forma de estimar la probabilidad de N-gramas que nunca ocurren pues les agregan un poco de la masa de probabilidad (*probability mass*). Los algoritmos de suavizado más comunes incluyen *backoff* o *deleted interpolation*, ya sea con Witten-Bell discounting o Good-Turing discounting.

El algoritmo ***Witten-Bell Discounting*** esta basado en una simple pero muy inteligente intuición acerca de eventos de frecuencia cero. Primero se piensa en una palabra o N-grama de frecuencia cero como un evento que no ha pasado aun; cuando éste ocurra, tendrá que ser la primera vez que vemos este N-grama. De esta manera, la probabilidad de ver un N-grama de frecuencia cero puede ser modelada mediante la probabilidad de ver un N-grama por primera vez. Éste es un concepto recurrente en procesamiento estadístico del lenguaje.

El calculo de la probabilidad de ver un N-grama por primera vez se realiza contando las veces que vemos un N-grama por primera vez en un corpus de entrenamiento. Esto es muy simple de producir debido a que la cuenta de los N-gramas que se ven por primera vez es solo el número de tipos de N-gramas en los datos de entrenamiento (se debe ver cada tipo de N-grama exactamente una vez).

El algoritmo de ***Good-Turing Discounting*** proporciona una forma un poco más compleja de realizar el suavizado. Fue descrito por primera vez por Good (1953), pero éste le dio el crédito a Turing por la idea original. La idea básica del algoritmo es re-estimar la cantidad de masa de probabilidad para asignarla a los N-gramas con valor cero o muy bajo, observando el número de N-gramas con valores altos.

Los algoritmos ***backoff*** y ***deleted interpolation*** proporcionan dos maneras de redistribuir la masa de probabilidad obtenida de los algoritmos de descuento (Good turing and Witten Bell dicounting) [5].

Los N-gramas se obtienen de los enormes textos de entrenamiento que comparten las mismas características del lenguaje que las señales de entrada esperadas. Los modelos de

lenguaje basados en corpus como los N-gramas, son evaluados separando el corpus en un conjunto de entrenamiento y un conjunto de prueba, entrenando el modelo en el conjunto de entrenamiento y evaluando sobre el conjunto de prueba. Estos conjuntos pueden ir variando de acuerdo a la cantidad de datos que se tengan.

Gramáticas restringidas

En los sistemas de reconocimiento de voz sencillos [17], los enunciados de entrada esperados generalmente son modelados con gramáticas estrictas, en otras palabras, al usuario sólo se le permite decir aquellos enunciados que están explícitamente cubiertos por la gramática. Estas gramáticas son convenientes cuando se modelan interfaces de control telefónico por voz. En estos casos, se utilizan expresiones regulares en la definición de la gramática.

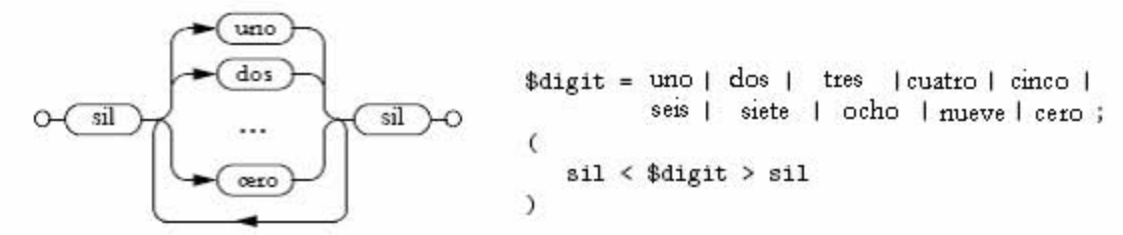


Figura 14. Ejemplo de una red de reconocimiento de dígitos

Esto, por supuesto, no funciona en sistemas de reconocimiento robustos donde la entrada es libre como en el caso del dictado.

Cuando se trabaja con sistemas de reconocimiento de voz que utilizan vocabularios enormes, es común utilizar modelos de lenguaje basados en N-gramas. Los N-gramas, a diferencia de las gramáticas, ven la probabilidad por secuencia de palabras, por ejemplo, un par de palabras en el caso de bigramas, o tres palabras para los trigramas, con lo que el modelo se vuelve menos restrictivo.

3.1.3 Diccionario de pronunciación

Una parte muy importante en el reconocimiento es el diccionario de pronunciación. En él se especifica la adecuada secuencia de sonidos (representados mediante un conjunto de símbolos) que componen una palabra. Los símbolos pueden ser definidos específicamente para la tarea de reconocimiento, o bien, obtenidos de un alfabeto fonético. Entre los alfabetos fonéticos más conocidos encontramos el Alfabeto Fonético Internacional (AFI), de la *Internacional Phonetic Association* (IPA), del que se derivan el Diccionario Electrónico de Formas Simples Flexivas del Español (DEFSFE), el *Speech Assessment Methods Phonetic Alphabet* (SAMPA) que tiene versiones en diferentes idiomas, el Worldbet, etc. Para esta tesis se tienen los niveles T54, T44 y T22 definidos en el corpus DIMEx100 y que se muestran en el capítulo 2.

Los diccionarios de pronunciación se construyen a partir de grandes corpus. Puede ser el mismo corpus usado durante el entrenamiento de los modelos acústicos o puede ser otro aun mayor. Mientras más palabras se tengan en el diccionario y en el modelo de lenguaje, menos restrictivo se vuelve el reconocedor de voz.

El corpus DIMEx100 cuenta con etiquetación realizada de forma manual en tres diferentes niveles de segmentación además de un nivel de palabras. El contar con etiquetación manual en un corpus oral, permite generar diccionarios de pronunciación mucho más amplios debido a que una misma palabra puede ser hablada de diferente manera por diferentes hablantes o incluso por la misma persona, y así, no solo se tiene la forma canónica de la palabra sino sus posibles variantes debido a ciertos fenómenos que ocurren en el lenguaje o, simplemente, porque el sujeto que grabó para el corpus tenía prisa, estaba cansado, etc.

Un ejemplo de varias formas para una misma palabra dentro del corpus DIMEx100, lo encontramos con WEB:

WEB	g u e
WEB (2)	g u e b
WEB (3)	u e b

3.2 Factores que influyen en el reconocimiento

Los principales factores que influyen en la construcción de un reconocedor de voz [11] son:

1. **El ruido**. Como ya se había mencionado en el modelo acústico, el ruido en el canal de comunicación puede afectar en gran medida el reconocimiento de un sonido e incluso hacer que el sistema no funcione.

2. **Ambigüedad en la pronunciación de las palabras**. Si aparecen problemas de claridad en la expresión o tenemos palabras similares el porcentaje de error en el reconocimiento puede ser elevado. Un diccionario muy amplio es causa de que surjan complicaciones en el sistema debido a que, implícitamente, hay palabras susceptibles de confusión. Por ejemplo, en un diccionario de más de 20,000 palabras en inglés puede darse el caso de que una de cada dos palabras se diferencien de otra solo por un fonema, además, mientras más grande sea el diccionario, mayor dificultad para recordarla.

3. **El tamaño del corpus**. Con un diccionario de menos de 50 palabras el sistema funciona muy bien, pues la variedad de opciones es baja y la tasa de error será baja también. Si se trabaja con un diccionario más amplio pero una gramática sencilla, entonces tampoco se presentan grandes complicaciones; sin embargo, con un diccionario muy amplio se presenta la ambigüedad y además la cantidad de datos de audio también debe ser considerablemente grande

4. **La variabilidad de acento en los hablantes**. A pesar de que teniendo un corpus lo suficientemente grande podemos tener independencia del hablante, es importante considerar el acento, puesto que un sistema creado para el español de México puede ser ineficiente tratándose de un argentino, un costarricense u otra persona cuyo acento sea muy marcado. Lo mismo sucedería en un sistema para el idioma inglés si es un latino el que lo utiliza.

3.3 Las herramientas

HTK (*Hidden Markov Model Toolkit*)

HTK es una herramienta para construir Modelos Ocultos de Markov creada por el departamento de Ingeniería de la Universidad de Cambridge; sin embargo, fue diseñada principalmente para construir modelos basados en el procesamiento de señales de habla.

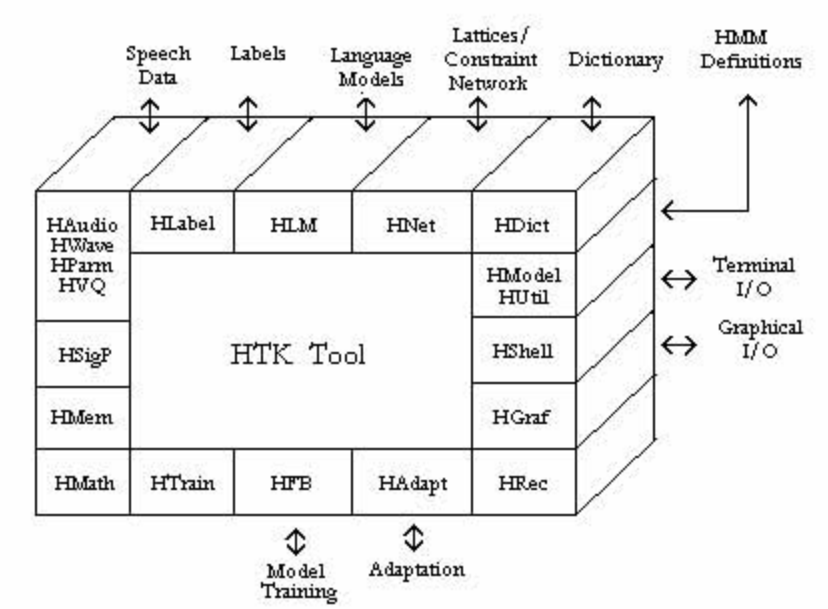


Figura 15. Arquitectura del software HTK

Contiene diferentes algoritmos de estimación de parámetros como el algoritmo Baum-Welch o el algoritmo Viterbi.

SPHINX

SPHINX es uno de los mejores sistemas de reconocimiento que existen en el mundo. Fue desarrollado en la Universidad de Carnegie Mellon [18] y al igual que HTK, se basa en la construcción de Modelos Ocultos de Markov. Los componentes de SPHINX son el *Sphinxtrain*, para entrenamiento de los modelos, y el SPHINX *decoder* para reconocimiento.

El *Sphinxtrain* esta compuesto por un conjunto de programas que han sido compilados para dos tipos de sistemas: Linux y alpha. Genera modelos acústicos discretos, semicontinuos o continuos (HMM con topología left to rigth) que pueden tener desde 1 hasta 24 gaussianas en cada estado.

El SPHINX *decoder* contiene algoritmos de programación dinámica como Baum-welch o Viterbi. La versión del *decoder* utilizada en este trabajo (s3.4) esta limitada a usar modelos de lenguaje de bigramas o trigramas y sólo trabaja con modelos acústicos continuos de 3 o 5 estados.

Procesamiento de las señales de audio con Sphinx.

Una señal de audio de 16 kHz es segmentada en fotogramas (*frames*). Cada fotograma es de 20 milisegundos o 320 muestras de habla y es multiplicado por una función de ventana de *Hamming* que se aplica cada 10 milisegundos. Consecutivos fotogramas se traslapan cada 10 milisegundos o 160 muestras de habla. De éstas muestras, se calculan los coeficientes LPC para finalmente obtener 12 coeficientes cepstrales de derivada LPC.

Módulos de Sphinx *Decoder*

Los ejecutables que Sphinx *decoder* contiene son:

1. **`decode`**: El decoder de Sphinx-3 s3.2/s3.3/s3.X para procesar archivos cepstrales.
2. **`gausubvq`**: Para construir subvectores de los clusters de los modelos acústicos.
3. **`livedecode`**: Ejecutable para las pruebas en vivo.
4. **`livepretend`**: Ejecutable para las pruebas en modo batch.
5. **`align`**: Ejecutable para realizar un alineamiento.
6. **`allphone`**: El reconocedor de fonemas de Sphinx-3.
7. **`astar`**: El generador del N-mejor en Sphinx-3.
8. **`dag`**: La aplicación para realizar la búsqueda del mejor patrón en Sphinx-3.

CAPÍTULO 4

Desarrollo y Experimentación

Para la construcción del reconocedor de voz se decidió usar Sphinx (*Sphinxtrain y Sphinx decoder 3.4*) debido a su simplicidad y disponibilidad. Se trabajó sobre plataforma linux.

Para el reconocedor se pretendía usar la totalidad del corpus DIMEx100; sin embargo, debido a la serie de cambios que se presentaron en los niveles de etiquetación y que se mencionan en el capítulo 2 de esta tesis, hasta el momento del desarrollo de este trabajo se tenía etiquetado y revisado sólo el 35% del mismo. Utilizando las primeras 30 carpetas del corpus, que corresponden a un 30% del total, se realizó el entrenamiento de los modelos acústicos, el 5% restante se reservó para las pruebas.

Con las transcripciones de esas 30 carpetas en sus niveles T22, T44 y T54 combinadas con el nivel de palabras, se crearon varios diccionarios de pronunciación (ver apéndice 3).

En esta tesis se empleó el nivel T22 para la construcción del sistema reconocedor de voz como un primer experimento para la validación del corpus. Gracias a que se realizó una etiquetación manual del corpus, el diccionario cuenta con varias pronunciaciones para una sola palabra con lo que tenemos mayor diversidad acústica para el reconocedor y mayor probabilidad de reducir la tasa de error.

La creación de modelos acústicos es el proceso más largo en la construcción de un reconocedor; éste proceso puede tomar mucho tiempo dependiendo de la cantidad de datos de entrenamiento con que se cuente.

Antes de comenzar con el entrenamiento los archivos de audio fueron modificados de diferentes maneras. Primero, se realizó una conversión en donde los archivos pasaron de 44.1 kHz, que es como se grabaron originalmente, a 16 kHz, que es como los necesita Sphinx. Después, debido a un error llamado "error de cuantización" que ocurre cuando el número de bits es reducido de la señal original, se realizó un proceso llamado *dithering*. Aplicar *dither* a una señal significa mezclarle "ruido" de manera controlada lo que aporta grandes beneficios a la señal; por ejemplo, si se tiene una onda limpia con curvas perfectas, el reducir el número de bits en la señal provoca que la curva se vea escalonada; debido a esto la señal pierde resolución y añadir entonces un poco de ruido hace más suave su forma escalonada creando un sonido más natural.

Utilizando la herramienta *Sphinxtrain*, se crearon modelos acústicos con las siguientes características:

- Modelo Oculto de Markov de 3 estados, continuo.
- 8 gaussianas por estado de cada modelo.
- Se crearon trifonemas (trifonemas significa que son fonemas dependientes del contexto, es decir, en donde cada fonema depende de los fonemas vecinos que lo acompañan).

Los Modelos Ocultos de Markov semicontinuos, al igual que los continuos, emplean funciones de densidad de probabilidad continuas para representar el espacio de vectores que contienen las características de la señal. La principal diferencia entre estos dos tipos de modelos es que, en el caso de los modelos semicontinuos, las funciones de densidad de probabilidad se comparten entre todos los estados de todos los modelos; por otro lado, en el caso continuo, cada estado de cada modelo tiene su propia función de densidad de probabilidad.

Podemos decir que los modelos semicontinuos presentan más ventajas porque el número de funciones de densidad de probabilidad no depende del número de modelos y, por lo tanto, el coste computacional se reduce en cierta medida; sin embargo, para nuestro caso se

generaron modelos acústicos continuos pensando que se tiene un buen corpus de entrenamiento, de tamaño considerable, y contamos con máquinas con un buen nivel de procesamiento (la maquina donde se realizaron las primeras pruebas es una HP con procesador centrino a 1.4 Gz, 256 MB de RAM y 40 GB de disco duro).

En total se crearon 22 modelos acústicos, correspondientes a las unidades fonéticas definidas en el nivel T22 del corpus DIMEx100 (tabla 13).

Consonantes	Labiales	Labiodentales	Dentales	Alveolares	Palatales	Velares
Oclusivas sordas	p		t			k
Oclusivas sonoras	b		d			g
Africada sorda		f			tS	
Fricativa sorda				s		x
Fricativa sonora					Z	
Nasales	m			n	n~	
Vibrantes				r(/ r		
Laterales				l		

Tabla 13. Nivel T22

El reconocimiento de voz debe lidiar, esencialmente, con la discriminación entre conjuntos discretos de unidades lingüísticas; por ejemplo enunciados, palabras o unidades fonéticas.

Posteriormente, con la trascripción de los enunciados se crearon dos modelos de lenguaje de 3-gramas. El modelo de lenguaje junto con el diccionario de pronunciación y los modelos acústicos, son los elementos esenciales del reconocedor.

Los modelos del lenguaje se crearon a partir de las 35 carpetas que se encontraban etiquetadas. El primero sólo abarcó las primeras 30 carpetas que se ocuparon durante el entrenamiento y el segundo se construyó con las 35 en total.

4. 1 Experimentos

Se le llama *entrenamiento* al proceso de aprender de las unidades de sonido.

El proceso de usar el conocimiento adquirido para deducir la secuencia de unidades acústicas más probable es llamado *decodificación* o simplemente *reconocimiento.*

Durante el periodo de experimentación el reconocedor se probó de dos diferentes maneras. Una fue en modo batch, es decir, reuniendo varios archivos (audio pregrabado) y procesándolos juntos; la otra manera fue realizando pruebas en vivo utilizando un micrófono para introducir la señal al sistema. En el caso de las pruebas en modo batch, se utilizó el audio que ya se tenía (puede ser audio grabado posteriormente y que no pertenezca al corpus) y se procesó junto con sus transcripciones, diccionario de pronunciación y modelo del lenguaje. En el caso de las pruebas en vivo, sólo fue necesario contar con el diccionario de pronunciación y el modelo de lenguaje. En ambos casos se utilizaron los mismos modelos acústicos que se obtuvieron durante el entrenamiento.

Experimento 1

El primer experimento consistió en utilizar las 30 carpetas que se ocuparon durante el entrenamiento para introducirlas al decoder en modo batch y así medir la calidad de los modelos acústicos.

Lo que el decoder arroja al final de la ejecución es un archivo que contiene todo el proceso de reconocimiento, desde que entra la señal de audio, después, como se generan los valores de las probabilidades hasta formar la primera palabra y finalmente, como la va uniendo con la siguiente que reconoce hasta que se forma toda la frase y continua con la siguiente elocución.

A continuación se presenta un ejemplo con la elocución s00101 donde se observa como se va formando la hipótesis parcial hasta obtener la elocución completa:

```
PARTIAL HYP: <sil>
PARTIAL HYP: <sil> EN
PARTIAL HYP: <sil> ENE
PARTIAL HYP: <sil> INDIQUE
PARTIAL HYP: <sil> EN EL CASO
PARTIAL HYP: <sil> EN EL CASO DEL
PARTIAL HYP: <sil> EN EL CASO DE LAS
PARTIAL HYP: <sil> EN EL CASO DE LAS QUE
PARTIAL HYP: <sil> EN EL CASO DE LAS CON
PARTIAL HYP: <sil> EN EL CASO DE LAS COLORES
PARTIAL HYP: <sil> EN EL CASO DE LA PSICOLOGIA
PARTIAL HYP: <sil> EN EL CASO DE LA PSICOLOGIA


Backtrace(s00101)
 LatID  SFrm  EFrm        AScr     LScr Type
     3     0     4      -21827   -74100   -1 <sil>
    73     5    20     -248521  -130150    0 EN
   269    21    30     -192166   -62092    0 EL
   828    31    57     -348897  -128079    0 CASO
   900    58    65     -127141   -45087    0 DE
   992    66    74     -160208   -90896    0 LA
  2023    75   154    -1014908  -207195    0 PSICOLOGIA
  2033   155   155           0   -23133    0 </s>
           0   155    -2113668  -760732 (Total)

EN EL CASO DE LA PSICOLOGIA (s00101)
```

Experimento 2

El segundo experimento consistió en utilizar las cinco carpetas restantes del corpus que ya estaban etiquetadas, y que se separaron precisamente para realizar una prueba con datos diferentes a los del entrenamiento. Primero se modificó el diccionario de pronunciación que ya se tenía, esto se logró agregando las palabras de las cinco carpetas que no se encontraban en él; posteriormente, se construyó un nuevo modelo de lenguaje en 3-gramas usando la trascripción de las 35 carpetas. Nuevamente se ejecutó el decoder en modo batch.

Experimento 3

El tercer experimento consistió en hacer pruebas en vivo (hablarle directamente al sistema con un micrófono). En este tipo de experimentos se debe tener cuidado en cuanto al diccionario de pronunciación ya que el sistema sólo puede reconocer palabras representadas

explícitamente en el diccionario. Si se dicen palabras que no aparecen en el diccionario, el decoder encuentra la mejor aproximación que muchas veces resulta en una frase carente de sentido.

Para la prueba se escogieron 10 personas para que leyeran 10 frases del corpus DIMEx100 cada una; estas frases fueron seleccionadas de las 35 carpetas usadas en la construcción del reconocedor de voz. No se utilizó ningún dispositivo especial ni lugar especial; tampoco se especificaron características de lectura a los hablantes. La intención era realizar una prueba en condiciones normales.

4.2 Resultados

Para medir los porcentajes de error en el reconocimiento para todos los experimentos, se hizo uso de una herramienta llamada **align** [16]. Esta herramienta compara al texto de referencia con el texto obtenido en el reconocedor. De esa manera se obtiene una evaluación a nivel de elocución y a nivel de palabra.

La evaluación, además de mostrar las palabras que fueron correctamente reconocidas del grupo de prueba, muestra las palabras que se hipotetizaron erróneamente (o se eliminaron), las palabras que se insertaron, conocidas también como palabras falsas y las sustituciones. El resultado a nivel de elocución es una medida muy importante en tareas en las que es necesario un reconocimiento absolutamente correcto, como es el caso del reconocimiento de números de tarjetas de crédito.

Otra medida ampliamente aceptada en reconocimiento de voz es la precisión de palabra (*WA: Word Accuracy*) [19] la cual se obtiene como porcentaje con la siguiente fórmula:

$$WA = 100\left(1 - \frac{W_S + W_I + W_D}{W}\right)\% \quad \ldots (8)$$

donde W es el total de palabras en la trascripción de referencia y W_S, W_I, W_D son el numero de palabras sustituidas, insertadas y eliminadas respectivamente.

Un ejemplo sencillo del cálculo de la precisión de palabra se presenta a continuación:

```
REF(1/1): el derecho de la union europea
HYP(1/1):    derecho de la reunion europea
```

La palabra “el” fue borrada y la palabra “unión” fue sustituida por “reunión” en el texto de la hipótesis. Sustituyendo estos valores en la fórmula (8) se obtiene:

$$WA = 100\left(1 - \frac{2}{6}\right) = 66.7\%$$

4.2.1 Evaluación del experimento 1

El resultado de comparar las trascripciones reales con las hipótesis arrojadas por el decoder para el experimento 1 fue:

```
REF(1/1): en el caso de la psicologia
HYP(1/1): en el caso de la psicologia

SENTENCE 1  (s00101)
Correct          = 100.0%    6   (     6)
Errors           =   0.0%    0   (     0)

================================================================================

REF(1/1): de la ciudad de mexico para el mundo
HYP(1/1): de la ciudad de mexico para el mundo

SENTENCE 2  (s00102)
Correct          = 100.0%    8   (    14)
Errors           =   0.0%    0   (     0)

================================================================================

REF(1/1): y sin embargo no deja de ser una cuestion muy IMPORTANTE
HYP(1/1): y sin embargo no deja de ser una cuestion muy IMPORTANTES

SENTENCE 3  (s00103)
Correct          =  90.9%   10   (    24)
Errors           =   9.1%    1   (     1)
```

```
================================================================================

REF(1/1): el derecho de la union europea
HYP(1/1): el derecho de la union europea

SENTENCE 4  (s00104)
Correct          = 100.0%    6   (    30)
Errors           =   0.0%    0   (     1)

================================================================================

REF(1/1): mantenimiento de alfombras en la ciudad de mexico
HYP(1/1): mantenimiento de alfombras en la ciudad de mexico

SENTENCE 5  (s00105)
Correct          = 100.0%    8   (    38)
Errors           =   0.0%    0   (     1)

================================================================================

REF(1/1): certificados de idiomas en caso de que los posea
HYP(1/1): certificados de idiomas en caso de que los posea

SENTENCE 6  (s00106)
Correct          = 100.0%    9   (    47)
Errors           =   0.0%    0   (     1)

================================================================================

REF(1/1): eso es lo que se refleja
HYP(1/1): eso es lo que se refleja

SENTENCE 7  (s00107)
Correct          = 100.0%    6   (    53)
Errors           =   0.0%    0   (     1)

================================================================================

REF(1/1): en la mayoria de los casos se emplea metaforicamente
HYP(1/1): en la mayoria de los casos se emplea metaforicamente

SENTENCE 8  (s00108)
Correct          = 100.0%    9   (    62)
Errors           =   0.0%    0   (     1)

================================================================================

REF(1/1): fondo de las naciones unidas para la infancia unicef
HYP(1/1): fondo de las naciones unidas para la infancia unicef

SENTENCE 9  (s00109)
Correct          = 100.0%    9   (    71)
Errors           =   0.0%    0   (     1)

================================================================================
```

```
REF(1/1): instituto nacional de estadistica geografia E  informatica
inegi
HYP(1/1): instituto nacional de estadistica geografia DE informatica
inegi

SENTENCE 10  (s00110)
Correct          =  87.5%    7   (    78)
Errors           =  12.5%    1   (     2)

==========================================================================
```

Se mostró solamente el análisis de las primeras 10 elocuciones de una carpeta, el resumen del análisis se presenta a continuación:

```
---------- SUMMARY ----------
Reference file:  all.corpus
Hypothesis file: all.hyp
Some extra alignment was attempted.


SENTENCE RECOGNITION PERFORMANCE:
sentences                          1500
  correct                   85.4% (1281)
  with error(s)             14.6% ( 219)
    with substitution(s)    10.0% ( 150)
    with insertion(s)        4.3% (  65)
    with deletion(s)         4.3% (  65)


WORD RECOGNITION PERFORMANCE:
Correct          =  98.2% ( 15039)
Substitutions    =   1.2% (   189)
Deletions        =   0.5% (    79)
Insertions       =   0.5% (    78)
Errors           =   2.3% (   346)

Ref. words       =  15307
Hyp. words       =  15306
Aligned words    =  15385

WORD ACCURACY =  98.2% ( 15039)  97.7%
```

CONFUSION PAIRS (155):			
6	DE DEL	1	ECONOMICA ECONOMICO
4	IMPORTANTE IMPORTANTES	1	DEMAS TEMAS
4	LAS LA	1	CENTRO CENTRA
4	HACER HACE	1	DECIR ES
3	TAMBIEN CAMBIE	1	TRATADOS DOS
2	EL DE	1	INTERNACIONALES INTERNACIONAL
2	OBJETIVOS OBJETIVO	1	INTENET INTERNET

2	EXISTEN	EXISTE	1	ESTES	ES
2	DE	EN	1	QUILMES	QUIENES
2	ACTIVIDADES	ACTIVIDAD	1	EJECUCION	OAXACA
2	PRINCIPIO	PRINCIPIOS	1	SISTEMAS	TEMAS
2	PROGRAMAS	PROGRAMA	1	APOYO	APOYA
2	ESPECIALIZADO	ESPECIAL	1	ESTO	ESTAS
2	W	U	1	COOPERATIVISMO	COOPERATIVOS
2	SECCION	ACCION	1	FIGURA	FIGURAS
2	LA	LAS	1	VERSION	VERSOS
1	E	DE	1	ENTRE	ENTRAMOS
1	EN	DE	1	HASTA	A
1	DEBE	DE	1	EL	ESTE
1	CIUDAD	SIDO	1	PUEDEN	PUEDE
1	DE	DIA	1	NUEVO	NO
1	ADVIENTO	BIEN	1	ESTE	ES
1	LA	DE	1	MUY	UN
1	ESTO	ESTABA	1	ESTO	ES
1	ESTA	ESTADO	1	HAY	CALLE
1	MAL	MAS	1	DE	A
1	DECIR	RESIDE	1	LA	ESTAR
1	FACULTAD	FACULTADES	1	HISTORIA	A
1	UNA	UN	1	TEORIA	TEORIAS
1	LA	LEGAL	1	HA	ASI
1	EVALUACION	EN	1	SIDO	A
1	DE	LOS	1	DECIR	DE
1	PSICOLOGIA	GUIA	1	LAS	DOS
1	HUSO	CURSO	1	LA	LOS
1	SIDO	SIDA	1	L	LA
1	DE	DESDE	1	ESTA	HASTA
1	RIVEROS	VER	1	DISPUESTO	RESPUESTA
1	CIENCIA	CIENCIAS	1	MUY	VOY
1	HEY	HAY	1	ESPERA	PERO
1	SE	ASI	1	M	MISMO
1	MUY	NO	1	UNA	ZONA
1	LLEGAR	LA	1	TE	TAPAS
1	UNA	NO	1	APASIONA	Y
1	NUEVA	NO	1	PRUEBAS	CENTRO
1	W	DOBLE	1	NO	LE
1	ENCUENTRA	ENCUENTRO	1	IBEROAMERICANA	UNICA
1	PARA	TODO	1	DE	NEGO
1	POCOS	POCO	1	GOBIERNO	EN
1	ACTIVIDADES	ARTE	1	LABOR	LABORAL
1	Y	ASI	1	EL	QUE
1	BOP	VOTO	1	CAMBIO	UN
1	NUESTRA	ES	1	AREA	REA
1	EXPERIENCIA	TEORIAS	1	HAGA	A
1	TRIGESIMO	DECIMO	1	SIDO	CITAN
1	INSTITUCIONES	ISTITUCIONES	1	RESULTADOS	RESULTADO
1	UN	NO	1	LUGAR	LUGARES
1	VICISITUDES	TODA	1	AUDITORIA	TEORIA
1	DE	TELE	1	BUENA	BUEN
1	ARTIFICIAL	TI	1	LAS	GAS
1	SE	DE	1	TIENE	TIENEN
1	ELEVADORES	LLEVADO	1	ESTA	ESTES
1	ES	HAS	1	O	POSEE
1	GARCHA	DERECHA	1	SEDE	DE
1	DECIR	ASI	1	PRIMERA	PRI

1 EFECTOS DEFECTO 1 INTERNACIONAL TE 1 DE NACIONAL 1 MAS MASAS 1 PODRA TODA	1 TAMBIEN BIEN 1 CONSEJOS CONSEJO ... ------ 189

En las siguientes tablas podemos apreciar la evaluación del experimento 1:

Reconocimiento a nivel de elocución

elocuciones	1500	--
Reconocidas correctamente	1281	85.4%
Con error	219	14.6%
Con sustitución	150	10.0%
Con inserción	65	4.3%
Con eliminación	65	4.3%

Tabla 14. Resultados del experimento uno a nivel de elocución

Reconocimiento a nivel de palabra

Palabras correctas	15039	98.2%
Con errores	346	2.3%
sustituciones	189	1.2%
eliminaciones	79	0.5%
inserciones	78	0.5%

Tabla 15. Resultados del experimento uno a nivel de palabra

```
Precisión de palabra =  98.2% ( 15039)  97.7%
```

El porcentaje de error a nivel de palabra fue de 2.3%. En la tabla que muestra los pares de palabras que causaron confusión se pueden ver ejemplos como “versión” y “versos” o “auditoria” y “teoría” cuyas pronunciaciones son similares en alguna sílaba.

El porcentaje de error a nivel de enunciado fue de 14.6%; se hace notar que el porcentaje de error se reparte entre los errores por sustituciones, eliminaciones e inserciones debido a que en una misma elocución pueden presentarse varios de ellos. En este primer experimento se puede decir que se obtuvieron buenos modelos acústicos.

4.2.2 Evaluación del experimento 2

Los resultados del segundo experimento fueron:

```
REF(1/1): colegio de *** ** BACHILLERES del estado de michoacan
HYP(1/1): colegio de VER SI QUIERES     del estado de michoacan

SENTENCE 1  (s03101)
Correct          =  85.7%    6   (     6)
Errors           =  42.9%    3   (     3)
   SC   BACHILLERES ==> SI QUIERES

**************************************************************************

REF(1/1): anatomia microscopica del canal raquideo y de la medula espinal
HYP(1/1): anatomia microscopica del canal raquideo y de la medula espinal

SENTENCE 2  (s03102)
Correct          = 100.0%   10   (    16)
Errors           =   0.0%    0   (     3)

**************************************************************************

REF(1/1): A     VER si vamos revisando el foro
HYP(1/1): HABER *** si vamos revisando el foro

SENTENCE 3  (s03103)
Correct          =  71.4%    5   (    21)
Errors           =  28.6%    2   (     5)
   MC   A VER ==> HABER

**************************************************************************

REF(1/1): bueno si quieres aqui tenemos un plano
HYP(1/1): bueno si quieres aqui tenemos un plano

SENTENCE 4  (s03104)
Correct          = 100.0%    7   (    28)
Errors           =   0.0%    0   (     5)

**************************************************************************

REF(1/1): los autobuses esta estacionados en frente de la puerta
HYP(1/1): los autobuses esta estacionados en frente de la puerta
```

```
SENTENCE 5  (s03105)
Correct          = 100.0%    9   (    37)
Errors           =   0.0%    0   (     5)

==========================================================================

REF(1/1): que es eso a lo que llaman software gratuito
HYP(1/1): que es eso a lo que llaman software gratuito

SENTENCE 6  (s03106)
Correct          = 100.0%    9   (    46)
Errors           =   0.0%    0   (     5)

==========================================================================

REF(1/1): por un lado los que recien comienzan hallaran T_C toda la
explicaciones necesarias
HYP(1/1): por un lado los que recien comienzan hallaran *** toda la
explicaciones necesarias

SENTENCE 7  (s03107)
Correct          =  92.3%   12   (    58)
Errors           =   7.7%    1   (     6)

==========================================================================

REF(1/1): ambito de aplicacion de la ley de condiciones generales de la
contratacion
HYP(1/1): ambito de aplicacion de la ley de condiciones generales de la
contratacion

SENTENCE 8  (s03108)
Correct          = 100.0%   12   (    70)
Errors           =   0.0%    0   (     6)

==========================================================================

REF(1/1): la sensacion que habia tenido por la man~ana se hizo aun mayor
HYP(1/1): la sensacion que habia tenido por la man~ana se hizo aun mayor

SENTENCE 9  (s03109)
Correct          = 100.0%   12   (    82)
Errors           =   0.0%    0   (     6)

==========================================================================

REF(1/1): ** ELABORACIO de la solicitud y el diagnostico
HYP(1/1): EL REGULACION de la solicitud y el diagnostico

SENTENCE 10  (s03110)
Correct          =  85.7%    6   (    88)
Errors           =  28.6%    2   (     8)
   SC   ELABORACIO ==> EL REGULACION

==========================================================================
```

```
---------- SUMMARY ----------
Reference file:  test_corpus
Hypothesis file: test_hyp
Some extra alignment was attempted.


SENTENCE RECOGNITION PERFORMANCE:
sentences                            250
  correct                  78.8% ( 197)
  with error(s)            21.2% (  53)
    with substitution(s)   13.2% (  33)
    with insertion(s)       9.6% (  24)
    with deletion(s)        6.0% (  15)


WORD RECOGNITION PERFORMANCE:
Correct          =  97.6% (  2527)
Substitutions    =   1.8% (    46)
Deletions        =   0.6% (    16)
Insertions       =   1.0% (    27)
Errors           =   3.4% (    89)

Ref. words       =   2589
Hyp. words       =   2600
Aligned words    =   2616

WORD ACCURACY =  97.6% (  2527)  96.6%
```

```
CONFUSION PAIRS (45):

     2          EN EL
     1 BACHILLERES QUIERES
     1           A HABER
     1  ELABORACION REGULACION
     1         LOS EN
     1      AYUDAS ATLAS
     1     ADMIRAR MIRA
     1        SOLO MOZO
     1          EL DEL
     1        SERA SE
     1      CANADA CAMARA
     1     DIARIOS SERIAS
     1          DE DEL
     1          DE DESEO
     1    SOFTWARE FUERE
     1      COSTEO COSTERO
     1         LOS LAS
     1        DADO LADO
     1 MENCIONADAS ES
     1     PENSION ATENCION
     1   SEGURIDAD DE
     1         ESE ES
     1       DEBEN DE
     1          EL LA
     1          LA HONORIS
     1         LOS OTRO
     1    DERECHOS DERECHO
     1         DEL DE
     1        LAGO LA
     1      BOVEDA PUBLICA
     1   EMOCIONES NACIONES
     1   ACORDAMOS DAMOS
     1          LA PARA
     1         QUE TIENE
     1         ESA ESTE
     1   CAMPEONES CAMBIO
     1        CUAL CUAN
     1       OESTE ESTEN
     1          DE SE
     1          SU SE
     1       PADRE PARA
     1      BREVES DE
...

------
    46
```

Las siguientes tablas muestran los principales datos de la evaluación del experimento 2:

Reconocimiento a nivel de elocución

elocuciones	250	--
Reconocidas correctamente	197	78.8%
Con error	53	21.2%
Con sustitución	33	13.2%
Con inserción	24	9.6%
Con eliminación	15	6.0%

Tabla 16. Resultados del experimento dos a nivel de elocución

Reconocimiento a nivel de palabra

Palabras correctas	2527	97.6%
Con errores	89	3.4%
sustituciones	46	1.8%
eliminaciones	16	0.6%
inserciones	27	1.0%

Tabla 17. Resultados del experimento dos a nivel de palabra

```
Precisión de palabra =  97.6% ( 2527)  96.6%
```

En este experimento, en comparación con el anterior, podemos observar que el error aumento a nivel de palabra en 1.1% y a nivel de elocución en 6.6%. Estas cifras no muestran una diferencia considerable. La precisión de palabra se conserva alrededor de 97%; no obstante, debido a que los datos que se introdujeron para obtener las hipótesis no fueron utilizados durante el entrenamiento, nos da una mayor certeza de lo preciso de los modelos acústicos.

4.2.3 Evaluación del experimento 3

Los resultados del experimento en vivo fueron obtenidos:

```
---------- SUMMARY ----------
Reference file:  frases.txt
Hypothesis file: exp3.txt
Some extra alignment was attempted.


SENTENCE RECOGNITION PERFORMANCE:
sentences                            100
  correct                    2.0% (   2)
  with error(s)             98.0% (  98)
    with substitution(s)    98.0% (  98)
    with insertion(s)       76.2% (  77)
    with deletion(s)        30.7% (  31)


WORD RECOGNITION PERFORMANCE:
Correct          =  41.2% (   339)
Substitutions    =  52.1% (   428)
Deletions        =   6.7% (    55)
Insertions       =  35.2% (   289)
Errors           =  93.9% (   772)

Ref. words       =    822
Hyp. words       =   1056
Aligned words    =   1111


WORD ACCURACY =  41.2% (   339)   6.1%
```

Reconocimiento a nivel de elocución

elocuciones	100	--
Reconocidas correctamente	2	2.0%
Con error	98	98.0%
Con sustitución	98	98.0%
Con inserción	86	86.0%
Con eliminación	23	23.0%

Tabla 18. Resultados del experimento tres a nivel de elocución

Reconocimiento a nivel de palabra

Palabras correctas	403	48.1%
Con errores	731	87.3%
sustituciones	400	47.8%
eliminaciones	34	4.1%
inserciones	297	35.5%

Tabla 19. Resultados del experimento tres a nivel de palabra

```
Precisión de palabra =  48.1% ( 403)  12.7%
```

Como podemos ver en los resultados, lo que sucedió en la prueba en vivo difiere mucho de lo obtenido en las pruebas en modo batch con los audios bien cuidados del corpus DIMEx100. Este resultado tiene mucho que ver con el ruido en el ambiente, el micrófono y la tarjeta de audio de la máquina donde se instaló el sistema. Analizando los porcentajes se observa que el número de sustituciones, eliminaciones e inserciones es muy alto y se intuye que no hay problema con los modelos acústicos o el modelo del lenguaje de acuerdo a los resultados de las pruebas anteriores.

Debido a estos resultados, se tomó la decisión de cambiar al sistema de equipo y volver a realizar la prueba 3. El equipo utilizado en la segunda ocasión tiene, principalmente, las siguientes características: procesador intel pentium 4 a 3 GHz, 1 GB de memoria, tarjeta de audio AC´97 integrada en una placa intel 915G/P/GV.

Estos son los resultados obtenidos con el nuevo equipo:

```
---------- SUMMARY ----------
Reference file:  frases.txt
Hypothesis file: exp3_2.txt
Some extra alignment was attempted.


SENTENCE RECOGNITION PERFORMANCE:
sentences                          100
  correct                  45.0% (  45)
  with error(s)            55.0% (  55)
    with substitution(s)   50.0% (  50)
    with insertion(s)      27.0% (  27)
```

```
    with deletion(s)        15.0% (  15)

WORD RECOGNITION PERFORMANCE:
Correct          =  85.1% (   712)
Substitutions    =  12.8% (   107)
Deletions        =   2.2% (    18)
Insertions       =   4.7% (    39)
Errors           =  19.6% (   164)

Ref. words       =   837
Hyp. words       =   858
Aligned words    =   876

WORD ACCURACY =  85.1% (  712)  80.4%
```

Reconocimiento a nivel de elocución

elocuciones	100	--
Reconocidas correctamente	45	45.0%
Con error	55	55.0%
Con sustitución	50	50.0%
Con inserción	27	27.0%
Con eliminación	15	15.0%

Tabla 20. Resultados del experimento tres (con otro equipo) a nivel de elocución

Reconocimiento a nivel de palabra

Palabras correctas	712	85.1%
Con errores	164	19.6%
sustituciones	107	12.8%
eliminaciones	18	2.2%
inserciones	39	4.7%

Tabla 21. Resultados del experimento tres (con otro equipo) a nivel de palabra

```
Precisión de palabra =  85.1% ( 712)  80.4%
```

En este segundo intento se puede observar una mejora considerable en el reconocimiento. Se puede decir que, efectivamente, los modelos acústicos y el modelo del lenguaje son de

calidad muy aceptable y el cambio de equipo mejoró notablemente la captura de la señal. Adicionalmente a estos experimentos se decidió hacer una comparación entre el software *Dragon Naturally Speaking 8* de la empresa *ScanSoft,* especializado en dictado, contra el sistema construido en esta tesis.

Puntos de comparación:

	Dragon Naturally Speaking	**Sistema DIMEx100**
Número de usuarios	Ilimitado (pero cada usuario debe realizar un proceso de entrenamiento)	Ilimitado
Modelo del lenguaje	No disponible	Disponible y con posibilidad de modificaciones
Diccionario fonético	No disponible	Disponible y con posibilidad de modificaciones

Tabla 22. Comparación entre Dragon Naturally Speaking y el sistema construido en este trabajo

No podemos hacer comparaciones en cuanto al entorno de operación debido a que *Dragon Naturally Speaking* trabaja sobre Windows y realiza una configuración de audio para cada usuario utilizando herramientas del sistema operativo mejorando mucho el reconocimiento de voz.

Para este experimento se tomaron a las mismas 10 personas del experimento 3 y se les pidió que leyeran sus elocuciones pero ahora utilizando el producto comercial.

Estos fueron los resultados:

```
---------- SUMMARY ----------
Reference file:  frases.txt
Hypothesis file: dragon.txt
Some extra alignment was attempted.
```

```
SENTENCE RECOGNITION PERFORMANCE:
sentences                          100
  correct                   71.0% (  71)
  with error(s)             29.0% (  29)
    with substitution(s)    27.0% (  27)
    with insertion(s)        5.0% (   5)
    with deletion(s)        13.0% (  13)


WORD RECOGNITION PERFORMANCE:
Correct          =  91.6% (   767)
Substitutions    =   5.5% (    46)
Deletions        =   2.9% (    24)
Insertions       =   0.7% (     6)
Errors           =   9.1% (    76)

Ref. words       =    837
Hyp. words       =    819
Aligned words    =    843


WORD ACCURACY =  91.6% (   767)  90.9%
```

Reconocimiento a nivel de elocución

elocuciones	100	--
Reconocidas correctamente	71	71.0%
Con error	29	29.0%
Con sustitución	27	27.0%
Con inserción	5	5.0%
Con eliminación	13	13.0%

Tabla 23. Resultados del experimento tres con Dragon Naturally Speaking a nivel de elocución

Reconocimiento a nivel de palabra

Palabras correctas	767	91.6%
Con errores	76	9.1%
sustituciones	46	5.5%
eliminaciones	24	2.9%
inserciones	6	0.7%

Tabla 24. Resultados del experimento tres con Dragon Naturally Speaking a nivel de palabra

```
Precisión de palabra =  91.6% ( 767)  90.9%
```

Como se puede ver, el producto comercial sobrepasa los porcentajes obtenidos en el experimento tres con el sistema montado en el equipo HP. Al cambiar el equipo y repetir el experimento, se observa una mejora que muestra resultados más satisfactorios y más similares a los del producto comercial. Los resultados de Dragon Naturally Speaking son muy buenos debido a que es un producto en el que participó mucha gente especializada y se distribuye mundialmente.

Finalmente se agregó ruido aleatorio, mediante el programa Matlab, en algunos archivos de audio para observar el comportamiento del sistema reconocedor de voz en otras condiciones; sin embargo, se comprobó que ese tipo de ruido aleatorio no es conveniente al sistema ya que los audios modificados no fueron reconocidos en absoluto.

CAPÍTULO 5

Conclusiones

Después de la preparación de los datos para entrenamiento y pruebas y habiendo aprendido a usar el paquete Sphinx se crearon diccionarios de pronunciación en diferentes niveles (aunque para el reconocedor solo se usó el nivel T22), se crearon modelos de lenguaje en 3-gramas y finalmente los modelos acústicos para después llevar a cabo las pruebas de reconocimiento.

El sistema reconocedor de voz que se obtuvo en este trabajo es para habla continua, es independiente del hablante y ya se mencionó que su número de usuarios es ilimitado.

Se obtuvieron resultados satisfactorios en los primeros experimentos de prueba en modo batch. En el primer experimento se tuvo un porcentaje de error menor a 5% a nivel de palabra; éste es un porcentaje muy bueno considerando que para el entrenamiento sólo se usó un 30% del corpus. Durante el segundo experimento, aun cuando se aumentó el modelo de lenguaje y se hizo la prueba con datos que no se ocuparon durante el entrenamiento, los niveles de reconocimiento se mantuvieron en un rango aceptable. Por el contrario, en el tercer experimento (el de la prueba en vivo), los porcentajes de error se elevaron a más del 50% tanto a nivel de palabra como a nivel de elocución lo que causó sorpresa y confusión debido a los resultados de las primeras dos pruebas. Finalmente, se pudo cambiar al sistema reconocedor de voz a un equipo con mejores características y se observó un cambio considerable en los niveles de reconocimiento en la prueba en vivo, y aunque no fueron porcentajes iguales a los obtenidos en los experimentos en modo bach, si estuvieron más cercanos.

En la prueba adicional al experimento tres con un producto comercial de dictado (Dragon Naturally Speaking 8.0) se obtuvieron niveles de reconocimiento de más del 70% a nivel de elocución y poco más de 90% a nivel de palabra. Aun nos falta trabajo para llegar a los niveles de reconocimiento de un producto comercial pero estamos en el proceso.

El reconocedor de voz esta limitado a un modelo de lenguaje que solo incluye palabras pertenecientes al 35% del corpus. Esto restringe el reconocimiento pero sirve para probar el desempeño de los modelos acústicos; sin embargo, es posible cambiar el modelo de lenguaje y el diccionario de pronunciación por unos mucho más amplios para futuros experimentos.

Los sistemas de reconocimiento de voz no toman en cuenta el contexto cultural, no conocen el significado de lo que el hablante dice y sólo se limitan a clasificar las frases desde un punto de vista acústico; sin embargo, como se pudo observar en este trabajo, no se trata de un problema trivial.

Desarrollar un software para construir este tipo de sistemas requiere de una enorme cantidad de conocimientos matemáticos y mucha programación. Podemos mencionar, entre otros, conocimientos de estadística y probabilidad, reconocimiento de patrones, lenguajes formales y autómatas, y eso, sin mencionar las habilidades de programación. El grupo de personas que han creado paquetes como Sphinx y HTK para generar Modelos Ocultos de Markov ha hecho una labor de años de investigación en el área y es por eso que esos paquetes son considerados de lo mejor que existe en el mercado.

El objetivo de la tesis no es mostrar el software para construcción de sistemas de reconocimiento de voz, sino mostrar el proceso de construcción del reconocedor, la validación del corpus DIMEx100 como un nuevo recurso para la construcción de tecnologías del habla y los resultados obtenidos de la experimentación; sin embargo, el conocer lo complejo de las herramientas que se utilizan para la construcción de este tipo de sistemas nos da una idea de los avances que se han hecho hasta el momento en el campo del procesamiento del lenguaje natural y justifica el hecho de usarlas.

Durante el proceso de construcción de un sistema reconocedor de voz el tener el recurso del corpus no es suficiente para lograr resultados; se deben conocer las herramientas que se tienen a la mano (en este caso Sphinx), se debe poder desarrollar material para manipulación de la información (scripts para automatizar tareas) y además contar con un buen equipo donde el sistema pueda ser probado.

Como se pudo comprobar en esta tesis el impacto de contar con un corpus bien cuidado, como en este caso del corpus DIMEx100, resultó en unos modelos acústicos bastante buenos. No obstante, el nivel de reconocimiento en las pruebas en vivo mostró que aun con buenos modelos acústicos existen otros factores que influyen en el resultado y es por eso que en la construcción de un reconocedor de voz hay que considerar tanto las cuestiones técnicas como el grupo de personas al que va dirigido. El corpus DIMEx100 esta formado por personas mayores de edad y en su mayoría, residentes del Distrito Federal y zona Metropolitana, debido a esto, se buscó que las personas que participaran en el último experimento tuvieran características similares a los 100 hablantes del corpus DIMEx100.

Un problema que se vio con frecuencia al momento de estar haciendo las pruebas en vivo con el reconocedor de voz, es que si algo funcionaba mal, era difícil saber si se trataba porque la persona debiera hablar más fuerte o más claro y había que ajustar el volumen de audio. En estos casos, también hay que tener en cuenta el tipo de micrófono, la calidad del canal de transmisión, el eco, el ruido, la distancia a la que la persona sostiene el micrófono, etc. además de que se comprobó como es que influyen las características del equipo en el que se encuentra el propio sistema. Debido a factores como los mencionados, aun se tienen muchas dificultades en la operación de los sistemas reconocedores de voz. Es importante que en una aplicación que se vaya a utilizar en cualquier circunstancia, se tenga independencia del hablante y del canal de comunicación considerando todos los aspectos que podrían hacer que el sistema fallara.

Esta tesis, como se mencionó en el primer capítulo, se desarrolló dentro del contexto del proyecto DIME coordinado por el IIMAS y del que tuve la fortuna de formar parte. En este

trabajo se concentran los esfuerzos de muchas personas que han colaborado en el proyecto DIME desde que éste dio inicio. Desde hace dos años que me integré al proyecto, he visto la enorme cantidad de trabajo que se tuvo que hacer para que mi tesis se pudieran realizar, empezando por formar y entrenar al grupo de etiquetadotes del corpus DIMEx100 (Haydé Castellanos, Ivonne López, Fernanda López, Varinia Estrada, Iván Moreno, Isabel López), en el cual colaboré, donde se pasó por un proceso de cambio de las unidades fonéticas hasta llegar a los niveles T54, T44 y T22 que se mencionan en el capítulo dos y con los que actualmente se continua el proceso de etiquetación del corpus por personas que realizan su servicio social dentro del mismo proyecto o personas que se acaban de integrar al grupo. Pero el trabajo dentro del proyecto no se limita a la etiquetación y validación del corpus DIMEx100, se tiene otra tarea más dentro del contexto como el etiquetado del corpus DIME, que es un corpus de habla espontánea en donde no sólo se tienen archivos de audio sino también video proporcionando el medio para otro tipo de estudios. En adición a todo lo mencionado, se han realizado seminarios entre el mismo grupo y con colaboradores nacionales y extranjeros lo que ha enriqueciendo aun más el proyecto.

Dentro del mismo contexto del proyecto DIME, se tuvo la oportunidad de realizar estancias en el INAOE en Puebla (Dr. Luis Villaseñor) y en el IHMC (Institute for Human and Machina Cognition) en Pensacola, Florida (Dr. Lucian Galescu) en donde se pudo aprender a utilizar la herramienta Sphinx para desarrollar este trabajo. Proyectos como DIME dan la oportunidad a los alumnos de vivir experiencias únicas colaborando en investigación abriendo así sus opciones.

Expectativas a largo plazo

En el futuro y conforme se avance en la etiquetación del Corpus DIMEx100 se podrán mejorar los modelos acústicos aumentando la cantidad de datos y también aumentando el diccionario de pronunciación. Con más datos será posible entrenar los modelos acústicos en el nivel T54 para observar si podemos obtener mejoras significativas en los resultados (en el nivel T44 ya se contaba con unidades en cantidad suficiente como para reproducir estos

experimentos). Tampoco hay que olvidar el modelo de lenguaje que tiene un peso significativo en el reconocedor.

Encontrar la manera de introducir al sistema una buena señal de audio, ya sea con una mejor tarjeta de sonido o tratando de disminuir el ruido en el ambiente, impactará enormemente el reconocimiento como lo demostró la prueba con el producto comercial que, sin utilizar equipo especial (salvo el micrófono que se incluye con el software), tuvo buenos resultados.

Otra posible mejora en el reconocimiento de voz consiste en producir de manera automática un diccionario de pronunciación con diferentes realizaciones de acuerdo al trabajo empírico que se tiene hasta ahora (las 35 carpetas etiquetadas y los diccionarios obtenidos de las mismas).

Otra característica del sistema es su portabilidad a otras aplicaciones como, por ejemplo, "Golem" el robot que se encuentra en el departamento de Ciencias de la Computación en el IIMAS; sin embargo para este efecto habría que adecuar el diccionario de pronunciación y el modelo de lenguaje a las necesidades del proyecto.

Un reconocedor de voz para el español de México utilizando la totalidad del corpus DIMEx100, debería proporcionar resultados con una tasa de error verdaderamente baja y por eso es importante resaltar la gran utilidad de un recurso de este tipo para la construcción de tecnologías de habla.

Tener un reconocedor de voz para el español de México que sea confiable permitiría construir sistemas de dialogo para acceder a una gran cantidad de información a través de la voz. Esta manera tan natural de comunicarnos facilitaría el prestar servicios interactivos a través de un teléfono, una televisión o una computadora. También en el caso de las personas con alguna discapacidad, por ejemplo, para una persona que no puede escuchar, sería muy útil un teléfono que mostrara lo que la persona al otro lado de la línea está diciendo.

APÉNDICES

1. Proceso de etiquetación automática y semiautomática de los niveles T22 y T44 respectivamente.

2. Proceso de creación de los modelos acústicos.

3. El diccionario de pronunciación.

4. Modelos de lenguaje.

5. Proceso de decodificación o reconocimiento.

6. Proceso de evaluación.

APÉNDICE 1

Proceso de etiquetación automática y semiautomática de los niveles T22 y T44 respectivamente

Aquí se presentan los scripts que sirven para producir los archivos de etiquetas de los niveles T44 y T22 para el corpus DIMEx100.

Se utiliza un script en perl para copiar y modificar los archivos del nivel T54, que es el nivel que se etiqueta de forma manual y del que podemos obtener los otros dos, y un shell script para preparar los archivos necesarios para el script de perl así como los directorios donde se guardarán los archivos una vez generados.

Para crear el etiquetado semi-automático de l nivel T44 se utilizan los siguientes scripts:

SHELL SCRIPT

```
echo "Procesando... "
mkdir T44
mkdir T44/comunes
mkdir T44/individuales
rm archivos.txt
ls comunes/*.phn > archivos.txt
ls individuales/*.phn >> archivos.txt
./paraT44.pl
```

SCRIPT EN PERL

```
#!/usr/bin/perl

  open (ENTRADA1, "<archivos.txt") || die "ERROR: No se pudo abrir
    archivos.txt";
  while(<ENTRADA1>)
  {
   open (ENTRADA, "<$_") || die "ERROR: No se pudieron abrir los phn";
   $a="";
   while($b=<ENTRADA>)
   {
   $f=0;
     chop($b);
     if(($b =~ s/\s(s_[\[\]]|z)\s/ s/g))
     {
       $f=1;
     }
     if(($b =~ s/\s(n_\[|n\[|N)\s/ n/g))
     {
       $f=1;
```

```
      }
      if(($b =~ s/\sE\s/ e/g))
      {
        $f=1;
      }
      if(($b =~ s/\sa_[2|j]\s/ a/g))
      {
        $f=1;
      }
      if(($b =~ s/\sO_7\s/ o_7/g))
      {
        $f=1;
      }
      if(($b =~ s/\sE_7\s/ e_7/g))
      {
        $f=1;
      }
      if(($b =~ s/\s(a_2_7|a_j_7)\s/ a_7/g))
      {
        $f=1;
      }
      if(($b =~ s/\sO\s/ o/g))
      {
        $f=1;
      }
      if(($b =~ s/\sdZ\s/ Z/g))
      {
        $f=1;
      }
      if(($b =~ s/\sk_j\s/ k/g))
      {
        $f=1;
      }
      if($f==0)
      {
        chop($b);
        $a.=$b."\n";
      }
      if($f==1)
      {
      $a.=$b."\n";
      }
    }
    close (ENTRADA);
    open (SALIDA, ">T44/$_") || die "ERROR: No se pudo abrir para
guardar";
    print SALIDA $a;
    close (SALIDA);
  }
  close(ENTRADA1);
```

- **archivos.txt** contiene el nombre y ruta de los archivos .phn del nivel T54 que serán utilizados para crear el nivel T44, al que después se le agregarán, manualmente, las codas silábicas.

Para crear los archivos correspondientes al nivel T22 se utilizan los siguientes scripts:

SHELL SCRIPT

```
echo "Procesando... "
mkdir T22
mkdir T22/comunes
mkdir T22/individuales
rm archivos.txt
ls comunes/*.phn > archivos.txt
ls individuales/*.phn >> archivos.txt
./paraT22.pl
```

SCRIPT EN PERL

```
#!/usr/bin/perl

  open (ENTRADA1, "<archivos.txt") || die "ERROR: No se pudo abrir
archivos.txt";
  while(<ENTRADA1>)
  {
   open (ENTRADA, "<$_") || die "ERROR: No se pudieron abrir los phn";
   $a="";
     print $_;
   while($b=<ENTRADA>)
   {
   $f=0;
     chop($b);
     if(($b =~ s/\s(s_[\[\]]|z)\s/ s/g))
     {
       $f=1;
     }
     if(($b =~ s/\s(e_7|E_7)\s/ e/g))
     {
       $f=1;
     }
     if(($b =~ s/\s(i_7|j_7|j)\s/ i/g))
     {
       $f=1;
     }
     if(($b =~ s/\s(w_7|u_7|w)\s/ u/g))
     {
       $f=1;
     }
     if(($b =~ s/\s(o_7|O_7|O)\s/ o/g))
     {
       $f=1;
     }
     if(($b =~ s/\s(V|v)\s/ b/g))
     {
       $f=1;
     }
     if(($b =~ s/\sD\s/ d/g))
     {
```

```
      $f=1;
    }
    if(($b =~ s/\sG\s/ g/g))
    {
      $f=1;
    }
    if(($b =~ s/\s(n_\[|n\[|N)\s/ n/g))
    {
      $f=1;
    }
    if(($b =~ s/\sE\s/ e/g))
    {
      $f=1;
    }
    if(($b =~ s/\s(a_[2|j]|a_[2|i|j]_7|a_7)\s/ a/g))
    {
      $f=1;
    }
    if(($b =~ s/\sdZ\s/ Z/g))
    {
      $f=1;
    }
    if(($b =~ s/\s(p_c|b_c|t_c|k_c|dZ_c|d_c|g_c|tS_c)\s/ c/g))
    {
      @cadena=split(' ',$b);
      $f=2;
    }
    if(($b =~ /\s(p|b|t|k|dZ|d|g|tS)\s/))
    {
      @cadena2=split(' ',$b);
      $sal=$cadena[0]." ".$cadena2[1]." ".$cadena2[2];
      $b=$sal;
      $f=1;
    }
    if(($b =~ s/\sk_j\s/ k/g))
    {
      @cadena2=split(' ',$b);
      $sal=$cadena[0]." ".$cadena2[1]." ".$cadena2[2];
      $b=$sal;
      $f=1;
    }
    if($f==0)
    {
      chop($b);
      $a.=$b."\n";
    }
    if($f==1)
    {
    $a.=$b."\n";
    }
  }
  close (ENTRADA);
  open (SALIDA, ">T22/$_") || die "ERROR: No se pudo abrir para
guardar";
  print SALIDA $a;
  close (SALIDA);
 }
```

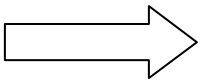
```
close(ENTRADA1);
```

- **archivos.txt** contiene el nombre y ruta de los archivos .phn del nivel T54 que serán utilizados para crear el nivel T22.

Ejemplo de conversión de T54 a T44 y a T22:

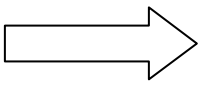
Nivel T54

```
56.149734 88.903740 k_c
88.903740 117.647057 k
117.647057 146.402084 w
146.402084 185.160431 a_2_7
185.160431 248.509628 l
248.509628 285.319092 e
285.319092 322.250793 z
322.250793 365.473999 l
365.473999 423.770111 a
423.770111 455.178345 D
455.178345 495.744049 i
495.744049 577.214355 f
577.214355 628.564514 e
628.564514 662.164063 r(
662.164063 761.060730 e_7
761.060730 802.901672 n
802.901672 852.983948 s
852.983948 877.074158 j
877.074158 912.892517 a
```



Nivel T44

```
56.149734 88.903740 k_c
88.903740 117.647057 k
117.647057 146.402084 w
146.402084 185.160431 a_7
185.160431 248.509628 l
248.509628 285.319092 e
285.319092 322.250793 s
322.250793 365.473999 l
365.473999 423.770111 a
423.770111 455.178345 D
455.178345 495.744049 i
495.744049 577.214355 f
577.214355 628.564514 e
628.564514 662.164063 r(
662.164063 761.060730 e_7
761.060730 802.901672 -N
802.901672 852.983948 s
852.983948 877.074158 j
877.074158 912.892517 a
```



Nivel T22

```
56.149734 117.647057 k
117.647057 146.402084 u
146.402084 185.160431 a
185.160431 248.509628 l
248.509628 285.319092 e
285.319092 322.250793 s
322.250793 365.473999 l
365.473999 423.770111 a
423.770111 455.178345 d
455.178345 495.744049 i
495.744049 577.214355 f
577.214355 628.564514 e
628.564514 662.164063 r(
662.164063 761.060730 e
761.060730 802.901672 n
802.901672 852.983948 s
852.983948 877.074158 i
877.074158 912.892517 a
```

APÉNDICE 2

Proceso de creación de los modelos acústicos

Para la creación del reconocedor se empleó la herramienta llamada Sphinx decoder versión s3.4 y para la construcción de los modelos acústicos se empleó la herramienta Sphinxtrain, ambas pertenecientes al grupo CMU SPHINX.

Se trabajó en ambiente linux Mandrake donde se instalaron todas las herramientas correspondientes:

- Sphinx3 Decoder
- Sphinxtrain
- Sox (para conversión de los audios)
- Perl (se crearon algunos scripts en este lenguaje)

Codificación y preparación de los datos

El procedimiento de creación de los modelos acústicos se realiza en dos fases, se tiene una fase de preparación del ambiente y datos y la fase de entrenamiento.

El ambiente es creado automáticamente mediante un script de Sphinx (**setup_SphinTrain.pl**) que genera una serie de directorios. En unos directorios se copian archivos de configuración y scripts de la herramienta (etc, scripts_pl), en otros se depositan todos los datos necesarios para la creación de los modelos (etc, wav) y en otros se guardan los archivos de salida, tanto los de errores como archivos temporales y los modelos (bwaccumdir, fear, logdir, model_arquitectura, model_parameters).

En total se crean diez directorios más *trees* que aparece cuando se ejecuta el modulo 5 del entrenamiento:

1. **bin**. Es el directorio donde se copian algunos scripts para manipulación de datos y se crean ligas hacia varios archivos ejecutables de la herramienta.

2. **bwaccumdir.** Es el directorio donde se guardan archivos temporales que arroja el algoritmo Baum-Welch durante el entrenamiento. Son archivos de acumulación de vectores para computar medias o varianzas.
3. **etc**. En este directorio se copia un archivo de configuración de Sphinxtrain y además es donde se guardan los diccionarios, el archivo que contiene la trascripción del corpus, el archivo de control (que contiene la lista de nombres de los archivos wav pero sin la extensión), y la lista de los modelos (en nuestro caso los fonemas del nivel T22).
4. **feat**. Es el directorio donde se depositan los archivos de características una vez procesados de los archivos de audio.
5. **gifs**. Es el directorio en donde se copian dos imágenes que indican error o acierto después de ejecutado un script durante la creación de los modelos.
6. **logdir**. En este directorio se guardan todos los archivos log después de ejecutado un script.
7. **model_architecture**. Es el directorio donde se guardan los archivos de definición de los modelos, tanto para los modelos dependientes del contexto como los independientes, y también al archivo que contiene la topología de los modelos. Todos estos archivos definen la estructura de los HMM.
8. **model_parameters**. En este directorio es donde se guardan todos los modelos acústicos.
9. **scripts_pl**. En este directorio se copian los scripts que corren cada uno de los nueve módulos que contiene Sphinxtrain.
10. **trees**. En este directorio se guardan archivos creados durante el módulo cinco del entrenamiento.
11. **wav**. Éste es el directorio donde se copian nuestros archivos wav para el entrenamiento.

Una vez creada la estructura de directorios, el siguiente paso es procesar los archivos wav para extraer los vectores de características.

Se llevó a cabo la conversión de archivos wav de 44.1 kHz a 16 kHz con la herramienta sox y se realizó la extracción de características desde Sphinxtrain mediante el script

make_feats.pl en el cual también se especificó la aplicación de *dither* a todos los archivos de audio.

Creación de los Modelos Acústicos utilizando Sphinxtrain

Lo que SPHINX necesita para realizar el entrenamiento de los modelos es:

1. Los archivos de audio convertidos a MFCC o algún otro formato que acepte SPHINX.
2. El correspondiente archivo de trascripción del corpus.
3. La lista con las unidades acústicas para los modelos que se quieren entrenar.
4. El archivo de control que contiene la lista de nombres del conjunto de archivos de características (por ejemplo MFCC) sin extensión.
5. El diccionario de pronunciación.
6. El diccionario con sonidos de relleno (al menos debe contener etiquetas para silencio al principio, en medio y al final de un enunciado, extras puede ser ruido o sonidos que no formen una palabra).

Dentro del directorio "etc" creado por **setup_SphinTrain.pl** se encuentra un archivo de configuración, que contiene las rutas donde encontrar los archivos necesarios para el entrenamiento.

Dentro del archivo de configuración podemos encontrar variables para especificar, entre otras cosas:

- **FEATFILE_EXTENSION.** El tipo de los archivos a procesar.
- **MAX_ITERATIONS, MIN_ITERATIONS.** Número máx. y mín. de iteraciones del algoritmo Baum-Welch.
- **STATESPERHMM.** Número de estados del Modelo Oculto de Markov.
- **HMM_TYPE.** Tipo de modelo a generar ya sea discreto, continuo o semicontinuo.

- **N_TIED_STATES.** Debe ser un valor entre 500 y 2500 que nos especifica el número total de distribuciones de estado compartidas del grupo final de HMM entrenados (los modelos acústicos).
- **CONVERGENCE_RATIO.** Es un número que puede ir de 0.1 a 0.001 y especifica la proporción entre la diferencia en probabilidad de la iteración actual y la anterior de Baum-Welch, y la probabilidad total de la iteración anterior. Cuantas más iteraciones de Baum-Welch se ejecuten, mejor se aprenderán las distribuciones de sus datos.
- **GAUSSIANSPERSTATE.** Especifica el número de gaussianas por estado que debe ser un número entre 4 y 32, pero en el caso de tener pocos datos, es recomendable que el número de gaussianas no sea mayor a 8.

El entrenamiento se realiza mediante el script **run_all.pl** que contiene a su vez otros scripts que ejecutan cada uno de los nueve módulos que tiene Sphinxtrain para generar los modelos acústicos.

- MODULO 00. Es el módulo de verificación de los archivos de entrenamiento.
    - Se asegura de que los diccionarios (tanto de pronunciación como de sonidos de relleno) sean congruentes con la lista de las unidades acústicas.
    - Observa que no haya entradas repetidas en el diccionario.
    - Verifica que existan todos los archivos que se listan en el archivo de control.
    - Observa que el número de líneas en el archivo de trascripción sea el mismo que en el archivo de control.
    - Por último, determina si la cantidad de datos de entrada es suficiente o razonable para empezar el entrenamiento y que todas las palabras en la trascripción aparezcan en el diccionario.

  El script que corre este módulo es: **scripts_pl/00.Verify/verify_all.pl**

- MODULO 01. Es el módulo donde se realiza la cuantificación de vectores que es una técnica de codificación que ha sido aplicada con éxito tanto a compresión de habla como de imágenes.

- Consiste en agregar los vectores de características a un solo archivo para después, realizar un cómputo de los centroides en el espacio de vectores.
- No aplica en el caso de modelos continuos.

El script que corre este módulo es: **scripts_pl/01.Vector_quantize/slave.VQ.pl**

- MODULO 02. Es el módulo de entrenamiento de los modelos independientes del contexto.
  - Aquí se realizan iteraciones con el algoritmo Baum-Welch.

  El script que corre este módulo es: **scripts_pl/02.ci_shmm/slave_convg.pl**

- MODULO 03. Es el módulo donde se atan los modelos y se crean los llamados trifonemas.

  El script que corre este módulo es:
  scripts_pl/03.makeuntiedmdef/make_untied_mdef.pl

- MODULO 04. En este módulo se vuelve a iterar con el algoritmo Baum-Welch y se crean modelos dependientes del contexto.

  El script que corre este módulo es: **scripts_pl/04.cd_shmm/slave_convg.pl**

- MODULO 05a. En este módulo se construyen árboles de decisión.

  El script que corre este módulo es: **scripts_pl/05.buildtrees/make_questions.pl**

- MODULO 05b. En este módulo se construyen árboles de decisión para cada estado del HMM.

  El script que corre este módulo es: **scripts_pl/05.buildtrees/slave_treebuilder.pl**

- MODULO 06. En este módulo se podan árboles anteriores.

  El script que corre este módulo es: **scripts_pl/06.prunetree/slave_state_tie_er.pl**

- MODULO 07. En éste módulo, se reestrenan los modelos dependientes del contexto hasta determinado número de gaussianas.

El script que corre este módulo es: **scripts_pl/07.cd_shmm/slave_convg.pl**

- MODULO 08. Es el módulo donde se realiza el borrado de interpolaciones.
    - No aplica en el caso de modelos continuos

    El script que corre este módulo es:

    scripts_pl/08.deleted_interpolation/deleted_interpolationl.pl

- MODULO 09. En éste módulo se realiza la conversión de los modelos al formato de SPHINX 2.
    - No es necesario en el caso de modelos para SPHINX 3

    El script que corre este módulo es: **scripts_pl/s2_models/make_s2_models.pl**

APÉNDICE 3

El diccionario de pronunciación

Para construir los diccionarios de pronunciación, usados durante el entrenamiento de los modelos acústicos, se utilizaron directamente las etiquetas del corpus en sus niveles de palabras, T22, T44 y T54. Con el uso de un script en perl (crea_diccionario.pl), se unieron, mediante la alineación en tiempo, la etiqueta de la palabra a su correspondiente representación en los niveles T22, T44 y T54.

Lo que hace el script crea_diccionario.pl es:

- Abre el archivo de etiquetas Tp y su correspondiente archivo de etiquetas T22 (por decir uno de los niveles).
- De Tp se obtiene un arreglo con el tiempo inicial, tiempo final y la palabra. La palabra se guarda en una variable.
- Posteriormente se obtienen los tiempos de cada símbolo en T22 y mientras el tiempo de los símbolos esté dentro del intervalo de tiempo de la palabra, se agregan los símbolos a la variable en donde previamente se había guardado la palabra. Repitiendo este proceso hasta acabar con todos los datos se va creando un diccionario con las palabras del corpus y sus correspondientes realizaciones de acuerdo al nivel alineado.

crea_diccionario.pl

```
#!/usr/bin/perl

 if(@ARGV[0] eq "")
 {
   print " Escribe la ruta donde se guardará el diccionario creado \n";
   exit(0);
 }

 open (ENTRADA1, "<alofonos.txt") || die "ERROR: No se pudo abrir palabras.txt";
 open (ENTRADA2, "<palabras.txt") || die "ERROR: No se pudo abrir alofonos.txt";
```

```
$a = "";
$lis = 0;

while( ($b= <ENTRADA1>) && ($c =<ENTRADA2>))
{

 open (PALABRAS, "<$b") || die "ERROR: No se pudo abrir ".$b;
 open (ALOFON, "<$c") || die "ERROR: No se pudo abrir ".$c;
 $corta = <PALABRAS>;
 $corta = <PALABRAS>;
 $corta = <ALOFON>;
 $corta = <ALOFON>;

 while ($c=<PALABRAS>)
 {

  @cadena1 = split(' ',$c);
  $lim = $cadena1[1];
  $band = 1;
  $alos="";

       while ($band==1)
       {

          if($d=<ALOFON>)
          {
            @cadena2 = split(' ',$d);
            if($cadena2[1] <= $lim)
            {
               if($lis == 1)
               {
                 $alos.=$guar;
                 $lis = 0;
               }
               $alos.=$cadena2[2]." ";
            }else{
                  $band = 0;
                  $guar = $cadena2[2]." ";
                  $lis = 1;}
          }else {$band = 0;}
       }
        $a.=$cadena1[2]."     ".$alos."\n";
 }
}
 close (ENTRADA);
 open (SALIDA, ">@ARGV[0]/diccionario.txt") || die "ERROR: No se pudo crear el diccionario";
```

```
print SALIDA $a;

close (SALIDA);
close(ENTRADA1);
close(ENTRADA2);
```

A continuación se presenta un ejemplo en donde se muestra como se va formando el diccionario de pronunciación:

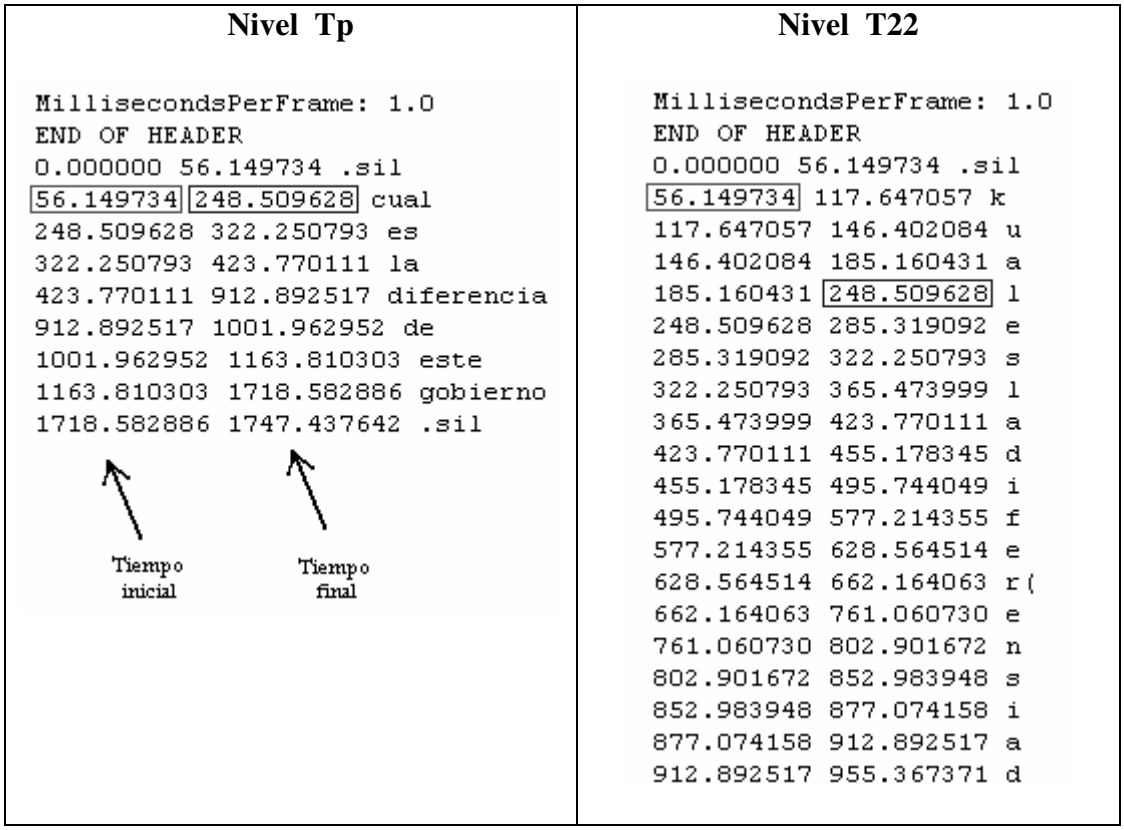


Figura 16. Ejemplo de alineación del nivel T22 al nivel de palabra

Lo que se obtiene finalmente es:

cual k u a l
es e s
la l a
diferencia d i f e r(e n s i a
de d e
este e s t e
gobierno g o b i e r(n o

APÉNDICE 4

Modelos de lenguaje

Para el propósito de la tesis se construyeron modelos de lenguaje de 3-gramas a partir del corpus DIMEx100 y la herramienta CMU Statistical Language Model Toolkit.

Una vez instalada la herramienta, se procede a preparar los datos necesarios para crear el modelo del lenguaje. Lo primero es tener las trascripciones del corpus, o bien, trascripciones del tipo de enunciados esperados durante el reconocimiento. En el caso del primer experimento se realizó un modelo de lenguaje a partir de las 30 carpetas del corpus que se ocuparon durante el entrenamiento de los modelos acústicos. Para el segundo y tercer experimento, se ocuparon las trascripciones de las 35 carpetas con que se contaba al momento.

El formato que deben tener las trascripciones es el siguiente:

<s> CUAL ES LA DIFERENCIA DE ESTE GOBIERNO </s>
<s> AVANCEMOS CON EL RESTO DE LAS OPCIONES </s>

Debe haber un inicio (<s>) y un fin (</s>) para cada enunciado.

Supongamos que el archivo con las trascripciones se llama *corpus.trn* para facilidad con los comandos.

1. El primer comando que se corre es **text2wfreq** de la siguiente manera:
   cat *corpus.trn* | **text2wfreq** | **sort** –rn –k 2 > *corpus.wfreq*

2. A continuación usamos **wfreq2vocab**:
   cat *corpus.wfreq* | **wfreq2vocab** –gt 0 > *corpus.vocab*

El parámetro –gt nos permite especificar el número de veces que debe ocurrir una palabra para ser incluida en el vocabulario.

3. El siguiente comando es **text2wngram** (aunque podría ir antes de wfreq2vocab):
   cat *corpus.trn* | text2wngram –n 3 –temp /tmp > *corpus.w3gram*

El parámetro –n indica el orden del modelo de lenguaje, aunque por default es 3. El parámetro –temp nos permite especificar el directorio donde guardar los archivos temporales que se crean al ejecutar el comando.

4. Ahora usamos el comando **wngram2idngram**:
   cat *corpus.w3gram* | **wngram2idngram** –n 3 -vocab *corpus.vocab* –temp /tmp > *corpus.id3gram*

5. Finalmente usamos el comando **idngram2lm**:
   idngram2lm -idngram *corpus.id3gram* –vocab *corpus.vocab* -context *con.ccs* -witten_bell –n 3 –vocab_type 0 –arpa *corpus3g.lm*

El parámetro –vocab_type permite especificar si se trata de un vocabulario cerrado (0) o si se trata de un vocabulario abierto (1).

Además podemos especificar la estrategia que se usará para el suavizado, es decir, la técnica en la cual se cambian las probabilidades para eventos con valor 0 o muy pequeño o eventos con valor mayor a 1. Estas pueden ser: Good Turing, Witten-Bell, descuento absoluto o lineal. Para este trabajo se escogió **Witten-Bell**.

APÉNDICE 5

Proceso de Decodificación o reconocimiento

Lo que SPHINX necesita para realizar el reconocimiento es:

1. El diccionario de pronunciación.
2. El diccionario con sonidos de relleno.
3. Los modelos acústicos.
4. El modelo de lenguaje.
5. Los datos de prueba.

El comando ocupado para realizar la prueba en modo batch es **livepretend**, que pertenece a sphinx 3 también llamado fast decoder package. Los argumentos necesarios para correr este comando son:

$**>livepretend** *ctlfile audiodir argfile*

- *ctlfile* es un archivo que contiene la lista de los nombres de todos los audios que se van a usar durante la prueba.
- *audiodir* es el directorio donde se encuentran todas las grabaciones para la prueba que deben ser convertidas a formato "raw".
- *argfile* es el archivo que contiene las banderas necesarias para que se efectúe el reconocimiento. El que se usó para las pruebas contiene lo siguiente:

```
-mdef
/Users/patricia/acoustic_models/dimex2/models/dimex2.1000.mdef
-mean         /Users/patricia/acoustic_models/dimex2/models/means
-var          /Users/patricia/acoustic_models/dimex2/models/variances
-mixw
/Users/patricia/acoustic_models/dimex2/models/mixture_weights
-tmat
/Users/patricia/acoustic_models/dimex2/models/transition_matrices
-subvq        /Users/patricia/acoustic_models/dimex2/models/subvq2
-fdict        /Users/patricia/acoustic_models/dimex2/models/dimex2.filler
-feat           1s_c_d_dd
-upperf         6855.49756
-lowerf         133.33334
-nfilt          40
-nfft       512
-samprate       16000
-dict
/Users/patricia/acoustic_models/dimex2/models/diccionario.new
```

```
-lm
/Users/patricia/acoustic_models/dimex2/models/corpus.lm.DMP
-agc             max
-varnorm         no
-cmn             current
-subvqbeam       1e-02
-epl             4
-fillprob        0.02
-lw              9.5
-maxwpf          10
-beam            1e-60
-wbeam           1e-35
-reportpron      0
-reportfill      0
      -outrawdir /dev/null
```

Para facilitar las pruebas usando livepretend se utilizó el siguiente script:

```
#!/bin/sh

S3BATCH=/Users/patricia/src/Sphinx3/src/programs/livepretend
TASK=/Users/patricia/acoustic_models/dimex/raw
CTLFILE=/Users/patricia/acoustic_models/dimex/ctl3
ARGS=/Users/patricia/acoustic_models/dimex/dimex.args

echo " "
echo "sphinx3-test"
echo "Run CMU Sphinx-3 in Batch mode to decode an example utterance."
echo " "

timestamp=`date +%Y%m%d%H%M%S`

mkdir ${timestamp}

$S3BATCH ${CTLFILE} ${TASK} ${ARGS}  2> ${timestamp}/log

echo ""
grep FWDVIT ${timestamp}/log
```

El comando para correr las pruebas en vivo es **livedecode**:

$**>livedecode** *argfile*

- *argfile* es el archivo que contiene las banderas necesarias para que se efectúe el reconocimiento. Se puede usar el mismo archivo utilizado con livepretend.

APÉNDICE 6

Proceso de evaluación.

El proceso de evaluación se realizó con la herramienta align. Para correr align se necesita el archivo que encuentra las hipótesis a partir del archivo que se obtiene de la ejecución del decoder y el archivo que contiene las transcripciones de referencia; la herramienta se corre de la siguiente manera:

$> **align** –hyp *hypfile* –def *transfile* > *results*

Donde *hypfile* es el archivo con las hipótesis, *transfile* es el archivo con las trascripciones originales y por último *results* es el nombre que tendrá el archivo con la evaluación.

En el archivo de evaluación lo que se observa son la elocución de referencia y la elocución de hipótesis para todos los datos de prueba, además se muestran las palabras eliminadas, insertadas o sustituidas en la elocución de hipótesis y se obtienen porcentajes de error por cada elocución, posteriormente, se da un resumen global con porcentajes a nivel de elocución y a nivel de palabra

A continuación se muestra una parte del archivo de evaluación:

```
REF(1/1): colegio de *** ** BACHILLERES del estado de michoacan
HYP(1/1): colegio de VER SI QUIERES     del estado de michoacan

SENTENCE 1  (s03101)
Correct          =  85.7%    6   (     6)
Errors           =  42.9%    3   (     3)
   SC   BACHILLERES ==> SI QUIERES
```

Bibliografía

[1] Alcaraz, Enrique.- Martínez Linares, Maria A. *Diccionario de Lingüística Moderna*. Barcelona, 1997.

[2] Constitución de las palabras: sonidos, fonemas y letras. http://roble.cnice.mecd.es/~msanto1/lengua/1sofolet.htm

[3] Cuétara Priede, Javier. (2004) *Fonética de la ciudad de México. Aportaciones desde las tecnologías del habla, tesis de maestría inédita*, México: UNAM.

[4] Fisher, R.A. (1922) *On the mathematical foundation of theoretical statistics. Transaction of the Royal Society of London*.

[5] Goldsmith, John. (2005) *Ngram models and the Sparsity problem*.

[6] Gordon, E. Pelton. *Voice Processing*. McGraw-Hill, Inc., 1993.

[7] Jelinek, F. *Statistical Methods for Speech Recognition*. MIT Press, 1997.

[8] Jurafsky Daniel & Martin James. *Speech and Language Processing – An introduction to Natural Language Processing, Conputational Linguistic and Speech Recognition*, Prentice- Hall Inc. New Jersey 2000.

[9] La estadística como herramienta para el desarrollo de sistemas automáticos reconocedores de habla. iies.faces.ula.ve/Revista/Articulos/Revista_14/rev14maldonado.htm

[10] Li Deng, Douglas O' Shaughnessy. *Speech Processing. A Dynamic and Optimization-Oriented Approach*. Marcel Dekker, Inc., 2003.

[11] Los sistemas integrales del habla, del lenguaje y la interfaz humana. http://www.imim.es/quark/21/021095.htm

[12] Martí Antonín, María A.- Alonso Martín, Juan Alberto et. al. *Tecnologías del habla*, UOC Aragón, Barcelona 2003.

[13] Pineda, Luis A. – Villaseñor, Luis et. al. *DIMEx100: A new phonetic and speech corpus for Mexican Spanish*, Iberamia, 2004.

[14] Poza Lara, M.J.- Villarrubia Grande, L.- Siles Sánchez, J.A. (1991) *Teoría y aplicaciones del reconocimiento automático del habla*, *Comunicaciones de Telefónica I+D 3*.

http://www.tid.es/presencia/publicaciones/comsid/esp/articulos/vol23/habla/habla.html

[15] Rabiner, L. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of The IEEE*, Vol. 77, NO. 2, Febrero 1989.

[16] Stan Janet, *Align -- String Alignment and Scoring Program*, Carnegie Mellon University, July 1987.

[17] Stochastic Language Models (N-Gram) specification. http://www.w3.org/TR/ngram-spec/

[18] The CMU Sphinx Group Open Source Speech Recognition Engines http://cmusphinx.sourceforge.net/html/cmusphinx.php

[19] *Towards understanding spontaneous speech: Word Accuracy vs. Concept Accuracy*. (1996). http://arxiv.org/abs/cmp-lg/9605028