UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

APLICACIÓN DE TECNOLOGÍAS DE ROBUSTEZ EN RECONOCIMIENTO DE VOZ A LA ENSEÑANZA DE SEGUNDO IDIOMA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELECTRICISTA

PABLO ANDRÉS RAVEST CATALÁN

PROFESOR GUÍA
NÉSTOR BECERRA YOMA

MIEMBROS DE LA COMISIÓN
CARLOS MOLINA SÁNCHEZ
JORGE WUTH SEPÚLVEDA

SANTIAGO DE CHILE
ABRIL DE 2009

Resumen de la Memoria para optar al
Título de Ingeniero Civil Electricista
Por: Pablo Andrés Ravest Catalán
Prof. Guía: Dr. Néstor Becerra Yoma
Santiago, Abril de 2009

"Aplicación de tecnologías de robustez en reconocimiento de voz a la enseñanza de segundo idioma"

El objetivo principal de esta memoria es mejorar el rendimiento de un sistema de evaluación de pronunciación automático basado en ASR (*Automatic Speech Recognition*) frente a cambios de locutor. Para lograr esto se propone la implementación de dos técnicas de robustez existentes en la literatura especializada: MLLR (*Maximum Likelihood Linear Regression*), que realiza una transformación lineal de los parámetros del modelo acústico para adaptarlo a un locutor específico; y VTLN (*Vocal Tract Length Normalization*), que normaliza el banco de filtros de Mel utilizado en la parametrización de las señales para compensar por diferencias en el tracto vocal de los locutores. Estos métodos se aplican de forma no supervisada y considerando una cantidad de información de adaptación limitada, debido a las exigencias que presentan los sistemas de CAPT (*Computer Aided Pronunciation Training*).

Este documento presenta experimentos con estas técnicas en ASR y CAPT considerando señales de locutores con distinto manejo del inglés y bajo variadas condiciones de ruido. En ASR se obtienen disminuciones del WER (*Word Error Rate*) de hasta un $30,56\,\%$ con MLLR de 25 señales y $16,23\,\%$ con VTLN de 1 señal. Los métodos muestran ser eficaces incluso al considerar pocas señales de adaptación, obteniéndose mejoras promedio del WER de $19,4\,\%$ y $6,34\,\%$ en MLLR con 5 señales y VTLN con 1 señal respectivamente. En evaluación de pronunciación, VTLN produce mejoras promedio del coeficiente de correlación entre los resultados entregados por el sistema y la evaluación esperada de 3,1 % y 5,01 % para dos bases de datos probadas. MLLR fue incapaz de aumentar la correlación debido a problemas con el modelo competitivo del CAPT y al modo de aplicación no supervisado.

Agradecimientos

En primer lugar me gustaría expresar mi agradecimiento a mis padres Gonzalo y Virginia, a mi hermano Gonzalo y a mi hermana Catalina. Gracias por el apoyo y cariño que me han brindado durante toda mi vida.

Quisiera agradecer a todos mis amigos, profesores y compañeros de los múltiples colegios y ciudades en que he estado: Colegio Javiera Carrera y Instituto Regional de Educación en Rancagua; Complejo Educacional Esperanza y Liceo Camilo Ortúzar Montt en Santiago; y Juan Pablo II en Calama. Si bien hemos seguido caminos distintos, los buenos recuerdos se mantienen todavía.

También me gustaría agradecer a todas las personas con las que he convivido durante mi vida universitaria. Gracias a mis compañeros de sección en plan común y compañeros eléctricos por ayudarme cuando lo he necesitado y honrarme con su amistad. Finalmente, agradezco al profesor Néstor Becerra y los miembros del LPTV por todo el conocimiento y ayuda que me han entregado y que me permitieron terminar exitosamente este trabajo.

Índice general

Índice de figuras

Índice de Tablas

Capítulo 1

Introducción

1.1. Reconocimiento de voz y aplicación a aprendizaje de un segundo idioma

La utilización de tecnologías computacionales para el apoyo en el aprendizaje de idiomas es cada vez más común estos días. Los sistemas típicos contienen ejercicios en los que se acepta y evalúa texto ingresado por el alumno, y que en algunos casos consideran reproducción de voz. Entre las ventajas de estas herramientas es posible mencionar que proporcionan un grado de interactividad mayor al alumno, el que participa activamente de los ejercicios, en una relación uno a uno con el programa computacional.

Actualmente, un área de investigación de gran interés consiste en desarrollar sistemas basados en reconocimiento de voz que puedan complementar las enseñanzas de compresión de lectura y de escucha, con actividades más activas que involucren conversación y diálogo por parte del alumno. Estos sistemas podrían plantear problemas seleccionados por profesores

calificados, analizar la respuesta del alumno, y luego presentar al alumno los errores que puede haber cometido, de forma automática.

Los sistemas de CAPT (*Computer Aided Pronunciation Training*) buscan mejorar uno de los aspectos fundamentales del habla: la pronunciación. Estos sistemas actúan como tutores virtuales, invitando a los estudiantes a repetir determinadas palabras con el fin de practicar y mejorar la calidad del lenguaje hablado, específicamente en lo que respecta a la producción de sonidos (características segmentales).

A pesar de los grandes avances en las herramientas CAPT, aún persisten ciertos problemas que impiden su masificación. En esta memoria se aborda uno de los principales problemas de los sistemas de evaluación de pronunciación y del reconocimiento de voz en general: la degradación del rendimiento de los sistemas frente a cambios de locutor.

1.2. Motivación

Los avances en la investigación de reconocimiento de voz han permitido un alto rendimiento de los sistemas de reconocimiento independientes de locutor con HMMS de densidad continua. Sin embargo, aún es el caso de que entrenar un sistema para un locutor específico resultará en mejor rendimiento para ese locutor que el utilizar un reconocedor de voz independiente de locutor. El incoveniente de los sistemas dependientes de locutor es que se debe recolectar una gran cantidad de información (típicamente horas) para poder lograr un modelo lo suficientemente preciso. Esto resulta impracticable si se piensa utilizar el sistema de forma masiva, como es el caso de las tecnologías de evaluación de pronunciación basadas en ASR (*Automatic Speech Recognition*). Esto provee una motivación clara al uso de técnicas que permitan adaptar los modelos independientes de locutor a un nuevo locutor utilizando

una cantidad pequeña de datos adaptación.

En esta memoria se propone la implementación de dos técnicas existentes en la literatura para tratar el problema de la variabilidad inter-locutor en ASR y CAPT: *Maximum Likelihood Linear Regression* (MLLR) y *Vocal Tract Length Normalization* (VTLN). MLLR es una técnica que modifica el modelo acústico del reconocedor mediante una transformación lineal de las medias de las componentes Gaussianas (Leggetter & Woodland, 1994). Por otra parte, VTLN realiza una normalización del banco de filtros de Mel utilizado para el procesamiento de las señales, con el fin de reducir la degradación del rendimiento del sistema debido a diferencias en la longitud del tracto vocal de los locutores (Lee *et al.*, 1998; Panchapagesan & Alwan, 2008). Estos métodos se aplican de manera no supervisada, considerando las transcripciones obtenidas en el reconocimiento y no las reales.

1.3. Objetivos

El objetivo principal de esta memoria es mejorar el rendimiento de un sistema de evaluación de pronunciación automático basado en ASR frente a cambios de locutor. Los objetivos específicos de este trabajo son los siguientes:

- Analizar el problema de robustez frente a cambios de locutor en ASR y su posible aplicación a CAPT.
- Implementar MLLR y VTLN en un sistema de reconocimiento de voz, de manera no supervisada.
- Probar el desempeño de estas técnicas en ASR, considerando locutores con diferentes características y en distintos ambientes de ruido.

- Implementar MLLR y VTLN en un sistema de evaluación de pronunciación basado en ASR, de manera no supervisada.

- Probar el rendimiento de estas técnicas en CAPT basado en reconocimiento de voz, considerando locutores con distinto dominio del inglés y pocas señales de adaptación.

1.4. Estructura de la Memoria

El capítulo 2 tiene como objetivo el introducir al lector a los temas de reconocimiento de voz, robustez de locutor en ASR y los sistemas de evaluación de pronunciación, que serán tratados en esta memoria. Se busca con esto brindar una base teórica que permita comprender los métodos propuestos y los análisis realizados, incluso para personas que no posean conocimientos avanzados en el reconocimiento de patrones y biometría.

En el capítulo 3 se describe la implementación propuesta de MLLR y VTLN en un sistema de reconocimiento de voz. Se detallan las consideraciones realizadas para aplicar cada una de estas técnicas, en un esquema no supervisado. Además se presentan las condiciones de evaluación utilizadas y los resultados obtenidos con ellas, para distintas configuraciones.

En el capítulo 4 se presenta la implementación realizada de MLLR y VTLN en un sistema de evaluación de pronunciación basado en ASR. Se describe como estas técnicas afectan el proceso de entrenamiento de los clasificadores y el proceso de evaluación objetiva de la pronunciación. Al final de este capítulo se detallan las pruebas realizadas para evaluar el desempeño del sistema y se brinda un análisis de los resultados.

Finalmente, en el capítulo 5 se entregan las conclusiones y análisis finales del trabajo realizado y se proponen trabajos futuros al respecto.

Capítulo 2

Reconocimiento automático de voz aplicado a la enseñanza de segundo idioma

2.1. Introducción

Este capítulo tiene como objetivo principal interiorizar al lector en la tecnología de reconocimiento de voz, los problemas de robustez ante cambios de locutor que éstos presentan y su aplicación a evaluación de pronunciación del idioma inglés. Se busca entregar una base teórica suficiente para comprender los análisis y técnicas presentes en esta memoria.

En primer lugar se describen las etapas y metodologías de un sistema de reconocimiento de voz basado en modelos ocultos de Markov (HMM, *Hidden Markov Modelos*). Se detalla la formulación del problema, las distintas etapas y cómo se evalúa el rendimiento del ASR

(*Automatic Speech Recognition*). A continuación se plantea el problema de robustez frente a cambios de locutor que presentan los sistemas de reconocimiento de voz y se entrega una revisión de los principales métodos de normalización y adaptación aplicados a ASR. Además, se presentan con un mayor grado de detalle las técnicas *Maximum Likelihood Linear Regression* y *Vocal Tract Length Normalization*, las que son utilizadas en esta memoria. En la siguiente sección se analizan las ventajas de CAPT (*Computer Aided Pronunciation Training*) como herramienta de aprendizaje de segundo idioma y el estado del arte de la aplicación de estas técnicas basado en reconocimiento de voz. Finalmente, se analiza el método de calificación objetivo utilizado por el sistema de evaluación de pronunciación desarrollado por el Laboratorio de Procesamiento y Transmisión de Voz (LPTV) de la Universidad de Chile, debido a que sobre éste se realizan las pruebas de las técnicas de robustez.

2.2. Reconocimiento automático de voz

2.2.1. Introducción

La función de un ASR (*Automatic Speech Recognition*) consiste básicamente en convertir una señal acústica, capturada con un micrófono u otro medio, en una secuencia de palabras. Las palabras reconocidas pueden ser el resultado final para aplicaciones tales como comando y control, preparación de documentos, entrada de datos, etc. También pueden servir como entradas a un procesamiento linguístico más avanzado con el fin de lograr comprensión del habla (Cole *et al.*, 1996).

Las características del ASR dependen fuertemente de la aplicación donde se utiliza. En la Tabla 2.1 se pueden apreciar algunos de los parámetros más relevantes. A modo de ejemplo, el reconocedor utilizado en esta memoria para la evaluación de pronunciación del inglés co-

rresponde a uno de palabra aislada, independiente de locutor y con un vocabulario pequeño.

Parámetros	**Rango**
Tipo de Habla	Palabras aisladas a diálogo continuo
Estilo de Habla	Diálogo leído a diálogo espontáneo
Enrolamiento	Dependiente de locutor a independiente de locutor
Vocabulario	Pequeño ($<$ 20 palabras) a grande ($>$ 20,000 palabras)
Modelo de Lenguaje	Estado finito a Sensible al contexto
SNR	Alto ($> 30dB$) a bajo ($< 10dB$)
Transductor	Micrófono con cancelación de ruido a teléfono

Tabla 2.1: *Parámetros típicos utilizados para caracterizar la capacidad de un ASR.*

La Figura 2.1 muestra los componentes principales de un ASR típico. La señal de voz digitalizada es transformada en primer lugar a un conjunto útil de características a una tasa fija. Estas medidas son luego utilizadas para buscar la secuencia de palabras más probable, haciendo uso de restricciones impuestas por los modelos acústicos y de lenguaje. Los parámetros que definen estos componentes del sistema son calculados previamente, utilizando las señales de entrenamiento.

En la sección siguiente se presenta la fundamentación matemática del proceso de reconocimiento de voz. Este desarrollo definirá de forma natural los modelos principales que aparecen en la Figura 2.1.

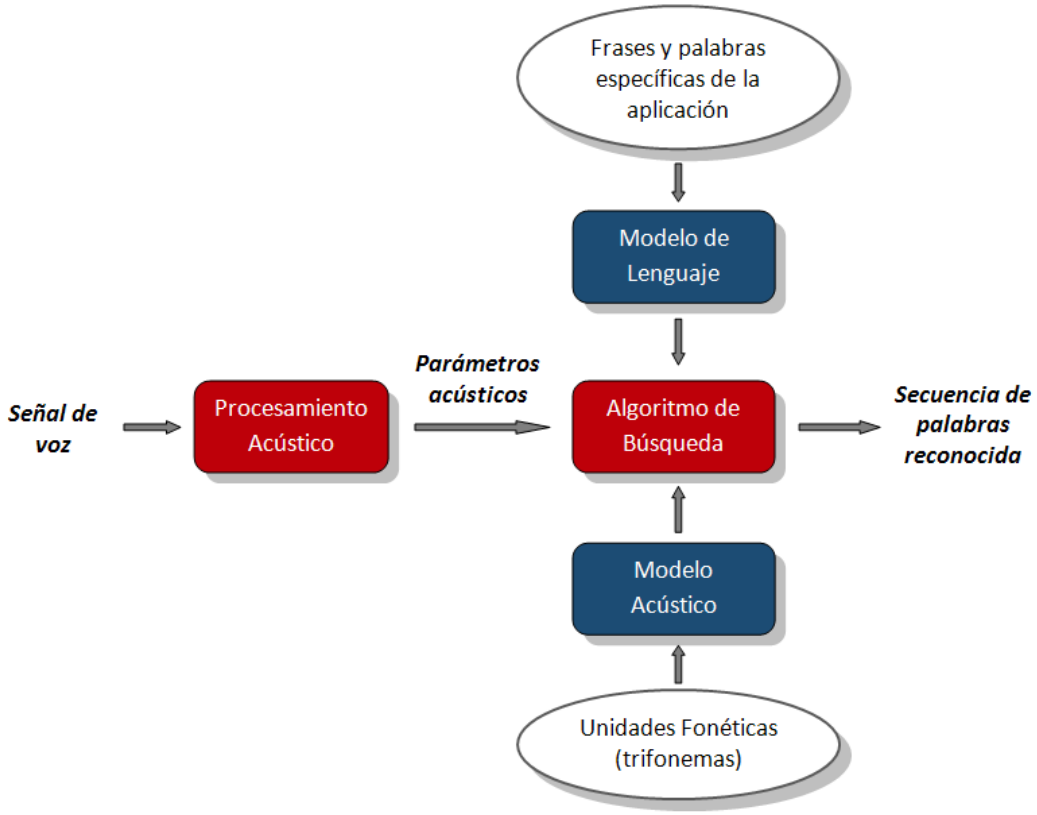


Figura 2.1: *Diagrama de bloques que describe el proceso de un ASR.*

2.2.2. Formulación matemática del problema

La tarea de reconocimiento de voz puede formularse matemáticamente utilizando un enfoque estadístico (Jelinek, 1998). Esta definición precisa permite una descomposición del problema en unidades más sencillas de manejar.

Sea $O = \{o_1, o_2, ..., o_T\}$ una secuencia de vectores de parámetros obtenidos a partir de la evidencia acústica observada. Sobre estos datos el reconocedor tomará una decisión sobre qué palabras fueron pronunciadas. Sea además $W = \{w_1, w_2, ..., w_J\}$ una secuencia de palabras, cada una de las cuales pertenece a un vocabulario fijo y conocido que dependerá de la aplicación específica en la que se usará el ASR.

Si $P(W/O)$ corresponde a la probabilidad de que la secuencia de palabras W haya sido pronunciada, dado que se observó la secuencia de vectores de parámetros O, entonces la búsqueda realizada por el ASR entregará la secuencia de palabras $\hat{W}$ que satisface la ecuación (2.1). $\hat{W}$ corresponderá entonces al candidato con la mayor probabilidad a posteriori (*MAP*).

$$\hat{W} = \arg\max_{W} P(W/O) \tag{2.1}$$

En la ecuación (2.1) se asume de forma implícita que todas las palabras del mensaje son igualmente importantes para el usuario. Esto quiere decir que los errores en el reconocimiento no son penalizados de manera distinta dependiendo de qué palabra no fue reconocida.

El teorema de Bayes permite reescribir el término $P(W/O)$ como:

$$P(W/O) = \frac{P(W)P(O/W)}{P(O)} \tag{2.2}$$

El término $P(W)$ corresponde a la probabilidad de que la secuencia de palabras W sean pronunciadas, $P(O/W)$ es la probabilidad de que la secuencia de vectores de parámetros O sea observada cuando el usuario utera W, y $P(O)$ es la probabilidad promedio de que O sea observado.

Dado que la maximización en (2.1) se realiza considerando una única secuencia de observación O, es posible reescribirla usando (2.2):

$$\hat{W} = \arg\max_{W} P(W)P(O/W) \tag{2.3}$$

La ecuación (2.3) determina los procesos que son de importancia para el desarrollo de

un reconocedor de voz: $P(O/W)$, que recibe el nombre de modelo acústico; y $P(W)$ que corresponde al modelo de lenguaje. Los parámetros que determinan estas probabilidades se determinan de manera de maximizar las verosimilitudes del conjunto de señales acústicas que se utiliza para entrenar. En las secciones siguientes se describirán detalladamente estos modelos.

2.2.3. Parametrización acústica

La primera etapa de un ASR corresponde a la extracción de información acústica de las señales de voz que se desea reconocer. Por lo tanto, en primer lugar es necesario determinar el tipo y la cantidad de parámetros que se considerarán.

Una señal de voz presenta dos características de suma importancia que deben ser consideradas si se desea realizar una parametrización adecuada: la señal de voz es un proceso estocástico no estacionario; y existen variaciones temporales entre señales que contienen la misma información fonética.

La variabilidad temporal en las señales de voz puede deberse a distintos factores. La variabilidad intra-locutor es definida como la información acústico fonética que se extrae de la señal de voz que varía entre elocuciones de un mismo individuo. De forma análoga, se desprende el concepto de variabilidad inter-locutor, el cual corresponde a las variaciones entre elocuciones pertenecientes a un grupo amplio de locutores. En (Yang *et al.*, 1996) se muestra que estas variaciones dependen tanto de las características fisiológicas estáticas (por ejemplo, la dimensión y morfología del tracto vocal) como de las características fisiológicas dinámicas (tales como diferencias en la velocidad de movimiento de labios o de lengua). En la Figura 2.2 se muestra un ejemplo de dichas variabilidades. Además, existen otras fuentes de variabilidad como el ruido ambiental (y su variación en el tiempo) y la dependencia de la

fuente (generalmente un micrófono) por la cual fue generada la señal.

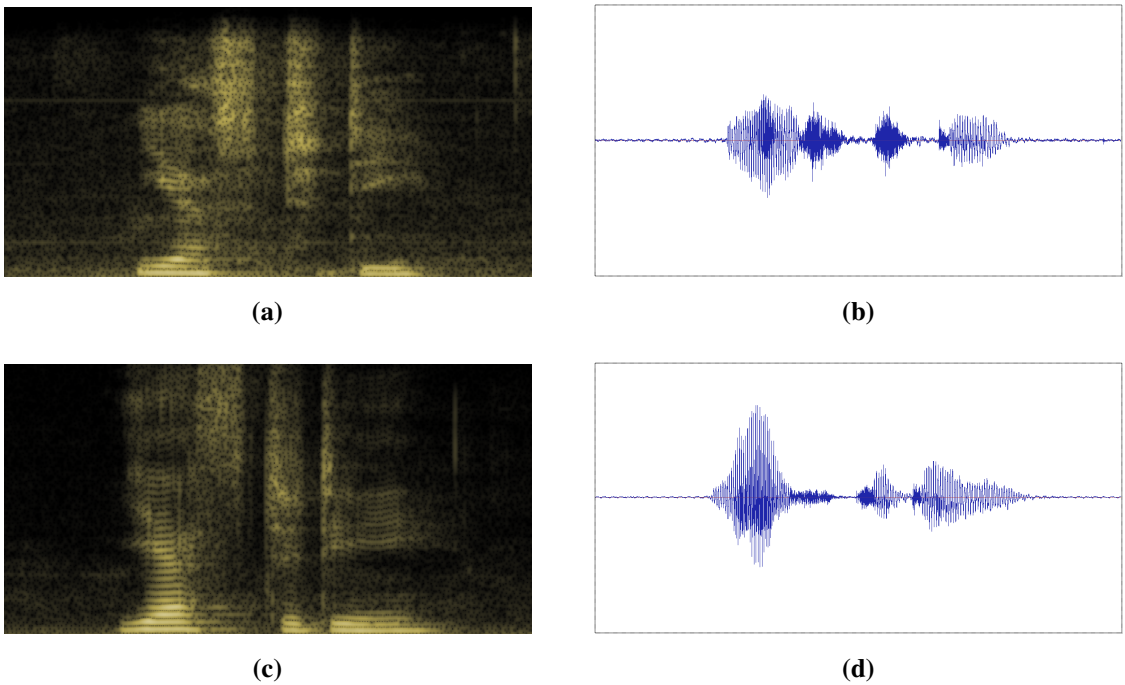


Figura 2.2: *Diferencias en el dominio espectral (Figuras (a) y (c)) y temporal (Figuras (b) y (d)) de dos señales de distintos locutores pronunciando la palabra "yesterday". El tiempo se representa en el eje horizontal en número de muestras (de 0 a 19700). En los espectrogramas el eje vertical corresponde a la frecuencia, de 0 y 8000 Hz, y la energía se representa en colores (de negro, menos intenso, a blanco, más intenso). En los gráficos en el dominio del tiempo el eje vertical denota la amplitud de la señal cuyo rango se sitúa entre -3000 y 3000.*

Todas estas características de la señal de voz dificultan el proceso de modelación de la voz, por lo que normalmente se realiza un preprocesamiento de la señal antes de la etapa de extracción de parámetros. Esta etapa tiene por objeto realzar la información de voz por sobre otro tipo de información que pueda contener la señal y de esta forma dejar todas las señales a analizar en condiciones similares para su caracterización. Esto se puede lograr mediante las siguientes tareas: detección del inicio y fin de la información de voz; supresión de segmentos de silencio; y, compensación de ruido aditivo y/o convolucional.

La primera etapa del pre-procesamiento es la conversión análogo-digital de la señal de voz. Esta tarea es realizada por el *hardware* de captura o por interfaces telefónicas. Luego la señal es procesada por un filtro inicio-fin el que elimina la información irrelevante que está antes y después del primer y último pulso de voz detectados (Lamel *et al.*, 1981; Savoji, 1989).

El siguiente paso es dividir la señal en segmentos que pueden ser considerados estadísticamente estacionarios, los que se denominan ventanas o *frames*. Con esto se busca lograr una caracterización de la señal ventana a ventana. Para esta segmentación generalmente se toman intervalos de 10 a 30 [ms], los que pueden tener un traslape de hasta 50 % entre ventanas consecutivas. Para evitar las distorsiones en el análisis espectral que pueden generar las discontinuidades en los límites de cada ventana, se utiliza la técnica de enventanado de Hamming (Picone *et al.*, 1993).

A continuación se realiza un análisis espectral por cada ventana, el que consta de un análisis por DFT (*Discrete Fourier Transform*) y de la aplicación de bancos de filtros por bandas. La utilización de estos filtros se debe a que la percepción auditiva humana no es capaz de distinguir frecuencias individuales, sino que capta franjas de frecuencias. Además la respuesta del sistema auditivo humano en el espectro de frecuencias no es lineal, lo que lleva a utilizar una escala en que la concentración de las frecuencias producto del filtrado simule la capacidad discriminativa del oído humano (en un rango de frecuencias aproximado de entre 300 y 3400 [Hz]). Una de las escalas más utilizada para estos efectos es la escala Mel. En la ecuación (2.4) se describe la transformación asociada a esta escala, para una frecuencia f:

$$MEL(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.4}$$

El banco de filtros se compone de un conjunto de funciones triangulares con ganancia

unitaria para la frecuencia central, con superposición de 50 % y un ancho de banda constante en escala Mel (ver Figura 2.3). Este es el último paso de la etapa de pre-procesamiento.

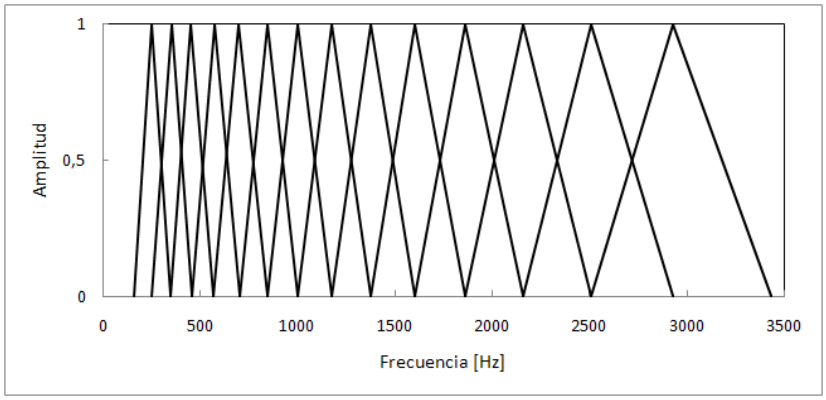


Figura 2.3: *Banco de filtros de Mel con 14 filtros triangulares.*

El método usado para la parametrización de señales acústicas de voz es el basado en la extracción de coeficientes cepstrales. Analizar una señal de voz en el dominio cepstral o *cepstrum* contribuye a realzar las componentes asociadas a los formantes del tracto vocal, incluso en señales con ruido. Los parámetros basados en el cepstrum se han convertido en uno de los métodos más usados en clasificación de patrones acústicos y ya se ha transformado en un estándar dentro del área de procesamiento de voz.

El cálculo de MFCC (*Mel Frequency Cepstral Coefficient*) se realiza a partir de la energía contenida en cada filtro y mediante una DCT (*Discrete Cosine Transform*). En procesamiento de voz, se obtiene un vector de parámetros MFCC para cada *frame* a analizar, es decir, una señal de voz es caracterizada como una secuencia de vectores de observación en el dominio MFCC.

2.2.4. Modelación acústica utilizando modelos ocultos de Markov

Las cadenas de Markov consisten en una secuencia finita de estados conectados entre sí por probabilidades de transición. Cada unidad temporal, que en el caso de procesamiento de voz corresponde a la ventana o *frame*, debe evaluar la posibilidad de mantenerse en el estado actual o avanzar al siguiente, lo que está determinado por las funciones de distribución de probabilidad de cada estado.

Los HMM (*Hidden Markov Model*) han sido ampliamente utilizados en los sistemas de reconocimiento de voz, en especial los modelos de primer orden (Jelinek, 1998; Rabiner, 1989). En estos modelos, el estado actual de una señal depende solamente de la señal que la precede. La salida de una secuencia de estados no es la secuencia misma, ésta permanece oculta en el proceso. Sin embargo, lo que se conoce es que esa secuencia produjo un conjunto de parámetros acústicos de la señal.

El uso de modelos ocultos de Markov en el campo del reconocimiento corresponde típicamente a aquellos que sólo permiten transiciones al siguiente o al mismo estado. Una topología de este tipo se denomina ”de izquierda a derecha”(left-to-right). En la Figura 2.4 es posible observar un ejemplo de una estructura de este tipo, considerando 5 estados y sin salto entre estados.

Un HMM queda definido por: las probabilidades de transición de estados; por la f.d.p (función de distribución de probabilidad); y por las probabilidades iniciales (Rabiner, 1989). Las probabilidades de transición para un HMM con M estados debe cumplir con la siguiente restricción:

$$\sum_{j=1}^{M} a_{i,j} = 1 \qquad \forall i = 1, \ldots, M \tag{2.5}$$

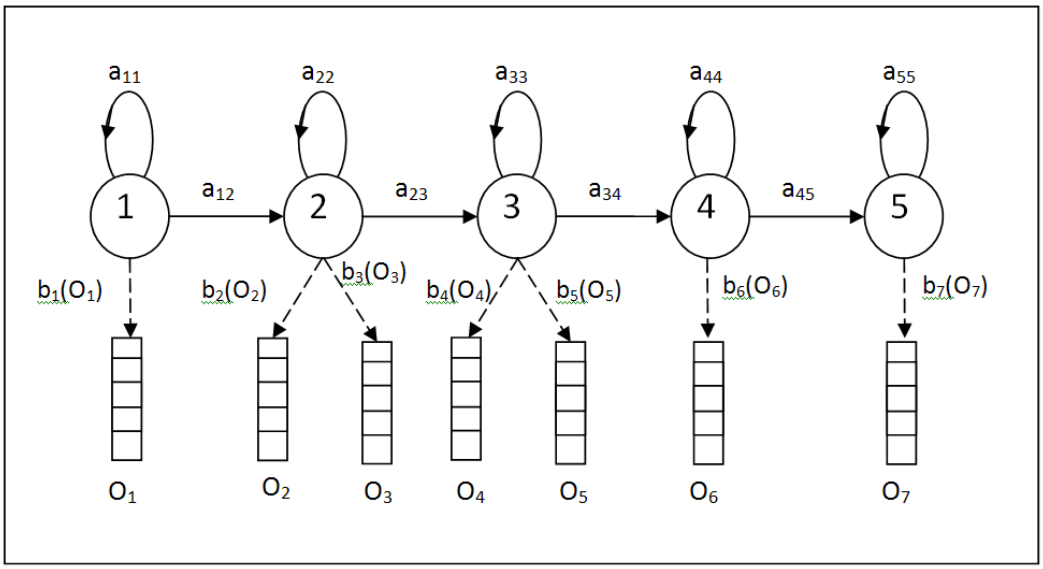


Figura 2.4: *Estructura de un HMM con topología de izquierda-a-derecha compuesta por 5 estados sin salto de estado.*

donde $a_{i,j}$ corresponde a la probabilidad de estar en el estado j dado que el anterior estado fue i.

La distribución de probabilidad de que una observación haya sido generada por el estado j se representa por la ecuación (2.6). Cabe mencionar que en la tarea de reconocimiento de voz es usual utilizar poblaciones para modelar las f.d.p. de estados. En este caso suponemos una población de G distribuciones normales independientes, cada una con un peso de probabilidad asignado, y restringido por la ecuación (2.7).

$$b_j(O_t) = \sum_{g=1}^{G} \left\{ p_g \cdot \prod_{n=1}^{N} \left[(2\pi)^{-0,5} \cdot (Var_{j,g,n})^{-0,5} \cdot e^{\left(-\frac{1}{2} \frac{(O^o_{t,n} - E_{j,g,n})}{(Var_{j,g,n})} \right)} \right] \right\} \tag{2.6}$$

$$\sum_{g=1}^{G} p_g = 1 \tag{2.7}$$

donde j, g, n son los índices para el estado, la componente Gaussiana y el coeficiente del vector de observación, respectivamente; p_g corresponde al peso de probabilidad de la población g-esima; $O_t = [O^o_{t,1}, O^o_{t,2}, \ldots, O^o_{t,N}]$ es el vector de observación de la señal acústica de dimensión N en el instante t; $E_{j,g,n}$ y $Var_{j,g,n}$ son la media y varianza para un determinado modelo en el estado j, componente Gaussiana g y coeficiente cepstral n. Cabe mencionar que la matriz de covarianza de las Gaussianas es supuesta diagonal, es por esta razón que en el párrafo se hace mención a la varianza.

Los HMMs representan unidades fonéticas. En este caso se utilizan los denominados trifonemas. Éstos se componen de una unidad fonética (fonema) central más dos segmentos de fonemas que preceden y suceden a la unidad central (Schwartz *et al.*, 1985).

Volviendo a la ecuación (2.3), la probabilidad de que un vector de parámetros acústicos O haya sido generado por el HMM de la secuencia de palabras W queda dada por (Jelinek, 1998; Rabiner, 1989):

$$P(O/W) = \sum_{S \in \Lambda} P(O,S/W) = \sum_{S \in \Lambda} P(S/W) \cdot P(O/S) \tag{2.8}$$

donde $S = [s_1, s_2, \ldots, s_T]$ representa cualquier secuencia de estado dentro del conjunto Λ; el conjunto Λ son todas las posibles secuencias de estados que son capaces de generar la secuencia de vectores de parámetros acústicos O. Si ahora descomponemos la ecuación (2.8) según la descripción de los HMM se obtiene la siguiente ecuación:

$$P(O/W) = \left(A(O,S_1) \cdot \prod_t A(S_t, S_{t+1}) \right) \cdot \left(\prod_t b_s(O_t) \right) \tag{2.9}$$

Al reemplazar la ecuación (2.9) en (2.3), finalmente se obtiene:

$$\begin{aligned}\hat{W} &= \arg\max_{W,S} P(W)\cdot P(O/W) \\ &= \arg\max_{W,S} P(W)\cdot\left(A(O,S_1)\cdot\prod_t A(S_t,S_{t+1})\right)\cdot\left(\prod_t b_s^W(O_t)\right)\end{aligned} \tag{2.10}$$

2.2.5. Modelo de lenguaje

El modelo de lenguaje entrega información *a priori* en la tarea de reconocimiento de la voz (definido por $P(W)$ en la ecuación (2.3)). Los métodos para estimarlo pueden variar desde ser un algoritmo de reglas gramaticales, hasta ser netamente una representación estadística del lenguaje utilizado. Los más usados son los modelos estocásticos de tipo M-grama. Esto considera que la ocurrencia de una palabra dentro de una sucesión de ellas está condicionada a la probabilidad de las $M-1$ palabras anteriores. La ecuación (2.11) refleja lo que se obtiene de un modelo M-grama.

$$P(W_1,W_2,W_3,\ldots,W_N) = \prod_{i=1}^{N} P(W_i \mid W_{i-M+1},\ldots,W_{i-2},W_{i-1}) \tag{2.11}$$

El criterio para la estimación de los parámetros que determinan el modelo de lenguaje es el estimador de máxima verosimilitud. En él se maximiza la probabilidad de observar las secuencias de algún conjunto de entrenamiento.

Uno de los problemas de los modelos estocásticos es que no considera probabilidad para las secuencias de palabras que no se encuentran en el conjunto de entrenamiento. Según la definición, estas probabilidades quedan en cero para aquellos casos en que no existe ocurrencia. El problema de generalización del modelo de lenguaje es tratado con diversas técnicas. Por ejemplo, existe el modelo de lenguaje a nivel de clases, depuración de parámetros o

modelos de lenguaje por palabras (Becchetti & Ricotti, 1999; Laurila *et al.*, 1998).

Lo que aún falta por resolver es cómo encontrar la secuencia de estados óptima que genera un vector de parámetros acústicos. Una secuencia de estados (S) determina inmediatamente una secuencia de HMMs, estos a su vez determinan la secuencia de palabras reconocidas (W). Para resolver este problema existe el algoritmo de Viterbi.

2.2.6. Algoritmo de Viterbi

Para realizar la búsqueda de la ecuación fundamental se necesita evaluar todas las posibles secuencias de estado para cada instante de tiempo en la señal de voz. Como es de suponer, esto es impracticable en un sistema computacional. Para minimizar la carga existe el algoritmo de Viterbi. El método consiste en ir optimizando a nivel local las secuencias de estado. Con ello, en forma inductiva, se resuelve el problema de optimización global (Jelinek, 1998). El algoritmo de Viterbi al ir optimizando a nivel local va descartando secuencias. Con ello logra reducir el campo de búsqueda y generar un algoritmo más viable desde el punto de vista computacional.

Definamos a $\delta_t(i)$ como la probabilidad de observar la secuencia de parámetros acústicos O hasta el tiempo t junto con la secuencia de estados más verosímil hasta t y que además, el estado s en t sea i. Es decir:

$$\delta_t(i) = \max_{s_1,s_2,\ldots,s_{t-1}} P(s_1, s_2, \ldots, s_{t-1}, s_t = i, o_1, o_2, \ldots, o_t/\lambda_w) \tag{2.12}$$

Suponiendo la recursividad del algoritmo esto se traduce en:

$$\begin{aligned}\delta_t(i) &= \max_{s_1,s_2,\ldots,s_{t-1}} P(o_t, s_t = i/s_1,\ldots,s_{t-1},o_1,\ldots,o_{t-1},\lambda_w)\cdot P(s_1,\ldots,s_{t-1},o_1,\ldots,o_{t-1},\lambda_w) \\ &= b_i(o_t)\cdot \max_{s_1,s_2,\ldots,s_{t-1}} P(s_1,\ldots,s_{t-1},o_1,\ldots,o_{t-1},\lambda_w) \\ &= b_i(o_t)\cdot \max_{s\in\Gamma}(a(s,i)\cdot\delta_{t-1}(s)) \end{aligned} \tag{2.13}$$

Para llegar al cálculo de $\delta_t(i)$ se debe evaluar todos los posibles caminos para llegar a $s_t = i$. Estos posibles caminos están agrupados en el espacio Γ , por lo tanto Γ es un conjunto de secuencias de t estados, es decir $\Gamma \in \Re^t$. δ_w es el modelo de la secuencia de palabra W hasta el instante t. El término $a(s,i)$ determina la probabilidad de transición del último estado en la secuencia S al estado dado en t que es i. Asumiendo la recursividad del algoritmo, si buscamos la secuencia de estados más verosímil para llegar a s_t, la secuencia anterior debe ser $\delta_{t-1}(s)$ donde s pertenece al conjunto Γ . Con esto, en forma recursiva, se llega a que $\delta_t(i)$ es la secuencia más probables de estados para llegar al tiempo t con el estado i (Jelinek, 1998).

Luego, para obtener la información del estado en el cual se está en el tiempo t se define la función $\psi_t(i)$, que a medida que se avanza en el algoritmo guardará la información del estado óptimo. Finalmente el algoritmo se define según la siguiente secuencia:

1. **Inicialización**

$$\begin{aligned}\delta_1(i) &= \Pi_i \cdot b_1(o_1) \qquad i\in\Gamma \\ \psi_1(i) &= 0 \end{aligned} \tag{2.14}$$

2. **Recursión**

$$\delta_t(i) = b_i(o_t) \cdot \max_{s \in \Gamma} (a(s,i) \cdot \delta_{t-1}(s)) \qquad i \in \Gamma \quad 2 \leq t \leq k$$
$$\psi_t(i) = \arg\max_{s \in \Gamma} (a(s,i) \cdot \delta_{t-1}(s)) \tag{2.15}$$

3. **Terminación**

   Se determina la probabilidad de la secuencia de estados más verosímil y el último estado de dicha secuencia.

$$P_{max} = \max_{i \in \Gamma} (\delta_k(i))$$
$$\hat{s} = \arg\max_{i \in \Gamma} (\delta_k(i)) \tag{2.16}$$

4. **Alineamiento**

   Finalmente se reconstruye la secuencia de estados más verosímil. Recordar que la función fue creada especialmente para esta etapa.

$$s_t = \psi_{t+1}(s_{t+1}) \qquad t = 1, \ldots, k-1 \tag{2.17}$$

Se han presentado en esta sección los elementos más importantes del proceso de reconocimiento de voz. A continuación se presentará una métrica para cuantificar el rendimiento de un ASR.

2.2.7. Evaluación del rendimiento de un ASR

Una métrica del rendimiento utilizada comúnmente en la evaluación de un ASR es el WER (*Word Error Rate*). El WER se deriva de la distancia Levenshtein, trabajando al nivel de palabras en vez de al nivel de fonemas.

El WER se define como:

$$WER = \frac{S+D+I}{N} \tag{2.18}$$

donde S es el número de palabras sustituidas, D es el número de palabras eliminadas, I es el número de palabras insertadas y N es el número de palabras total que efectivamente fueron pronunciadas.

En esta memoria se considerará además la reducción porcentual del WER con respecto al obtenido en el *baseline* (caso base del experimento, sin aplicar las técnicas de adaptación). Este valor puede obtenerse a partir de la siguiente ecuación:

$$\textit{Reducción Porcentual del WER} = \frac{WER_{baseline} - WER_{exp}}{WER_{baseline}} * 100 \tag{2.19}$$

donde WER_{exp} corresponde al experimento de reconocimiento considerando una determinada configuración de la técnica de adaptación que se desea comparar con la configuración del *baseline*.

A pesar de los grandes avances logrados en los sistemas ASR, aún persisten varios problemas que han impedido su masificación. A continuación se discutirá un poco más de estos problemas y cuáles son los avances al respecto que se encuentran en la literatura.

2.3. Robustez en ASR

2.3.1. Motivación

Los avances en la investigación de reconocimiento de voz han permitido un alto rendimiento de los sistemas de reconocimiento independientes de locutor con HMMS de densidad continua. Estos sistemas funcionan de manera adecuada debido a que utilizan una gran cantidad de datos de entrenamiento para alcanzar un modelamiento detallado de los patrones de voz.

Sin embargo, aún es el caso de que entrenar un sistema para una persona en específico resultará en mejor rendimiento para ese individuo que el utilizar un reconocedor de voz independiente de locutor. Típicamente, un sistema independiente de locutor presenta una tasa de error de 2 a 3 veces mayor que a obtenida con un reconocedor dependiente de locutor (Huang & Lee, 1993). Esto se debe a que en un sistema independiente de locutor (Hamaker, 1999) no se utiliza la capacidad de los modelos de describir las peculiaridades de cada locutor (largo y forma del tracto vocal, acento, edad, sexo, etc.).

El incoveniente de los sistemas dependientes de locutor es que se debe recolectar una gran cantidad de información (típicamente horas) para poder lograr un modelo lo suficientemente preciso. Esto resulta impracticable si se piensa utilizar el sistema en un universo amplio de locutores, como es el caso de gran parte de las aplicaciones de reconocimiento de voz. Además, existe una gran cantidad de datos disponibles para el entrenamiento de sistemas independientes de locutor, tales como el Wall Street Journal CSR Corpus (*Continuous Speech Recognition*) (Paul & Baker, 1992). Esto provee una motivación clara al uso de técnicas que nos permitan adaptar los modelos independientes de locutor a un nuevo locutor utilizando una cantidad pequeña de datos de adaptación.

Es generalmente aceptado que una de las principales causas de variabilidad inter-locutor corresponde a la diferencia en tamaño y/o forma del tracto vocal. Las posiciones de los *peaks* de las formantes en el espectro de una uteración son inversamente proporcionales al largo del tracto vocal. En los adultos el tracto vocal puede variar de aproximadamente unos 13 [cm] en las mujeres hasta unos 18 [cm] en los hombres, lo que produce unas diferencias en los centros de las frecuencias formantes que pueden llegar a un 25 % entre locutores (Lee *et al.*, 1998). Estas diferencias pueden llegar a ser mayores en el caso de los niños, lo que explica en parte el bajo rendimiento de los ASR entrenados con bases de datos de adultos y probadas en niños (Potamianos & Narayanan, 2003).

Es por esta razón que parte del trabajo de investigación en el tema de robustez de locutor se ha centrado en el desarrollo de una técnica de normalización de la longitud del tracto vocal o VTLN (*Vocal Tract Length Normalization*). Existen muchas formas distintas de implementar VTLN en la literatura, pero todas realizan una distorsión (*warping*) del eje de las frecuencias del espectro de las señales o del banco de filtros (Mel o Bark) utilizado en la parametrización. Uno de los problemas de VTLN es que normalmente se realiza una búsqueda del factor de normalización en un conjunto de posibilidades, por lo que se necesita calcular las nuevas caracteristicas normalizadas de la señal para cada factor. Las técnicas más recientes se basan en el hecho de que el *warping* de frecuencias es equivalente a una transformada lineal en el espacio cepstral (Pitz *et al.*, 2001; Panchapagesan & Alwan, 2008). Esto permite realizar el proceso de VTLN de forma menos costosa computacionalmente, dado que la transformada puede ser aplicada directamente sobre las características obtenidas sin normalización.

Otra de las formas típicas de brindar robustez a los ASR es mediante la adaptación de los modelos acústicos. La idea básica de estas técnicas puede observarse gráficamente en la Figura 2.5. Esencialmente, se desea modificar el sistema de reconocimiento de voz para que se adapte de mejor forma al ambiente de operación (Lee, 2008). Considerando un reconocedor independiente de locutor bien entrenado y utilizando una pequeña cantidad de información

de un nuevo usuario, el sistema puede ser adaptado para mejorar el modelamiento de ese usuario.

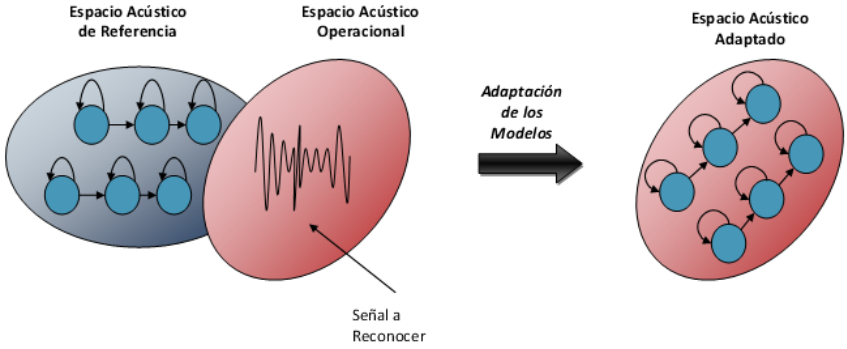


Figura 2.5: *Representación de alto nivel del proceso de adaptación de locutor.*

Los métodos de adaptación de locutor pueden clasificarse de acuerdo al modo de entrenamiento (Hamaker, 1999) en *aprendizaje supervisado* y *aprendizaje no supervisado*. En *aprendizaje supervisado* se utiliza la transcripción correcta de las señales de entrenamiento y el sistema de reconocimiento sólo debe alinear el diálogo del usuario a esa transcripción. En cambio, en *aprendizaje no supervisado* el reconocedor se "alimenta" a sí mismo, utilizando la transcripción obtenida al reconocer las señales de entrenamiento utilizando los modelos originales no adaptados. De esta forma, la transcripción utilizada puede contener errores de reconocimiento.

Además, es posible clasificar los métodos por modo de adaptación en *incremental* o *batch*. En modo *incremental* los modelos son adaptados continuamente a medida que los datos de adaptación están disponibles. Por lo tanto, los modelos cambian repetidamente y los modelos adaptados son utilizados para producir las hipótesis (transcripción reconocida) de la siguiente adaptación. Por otra parte, en modo *batch* los modelos son adaptados considerando toda la información de adaptación y sus transcripciones en un sólo bloque, por ejemplo, después de una sesión de enrolamiento. En este trabajo se utilizará el esquema de la Figura 2.6, que

corresponde a un método de adaptación batch no supervisado.

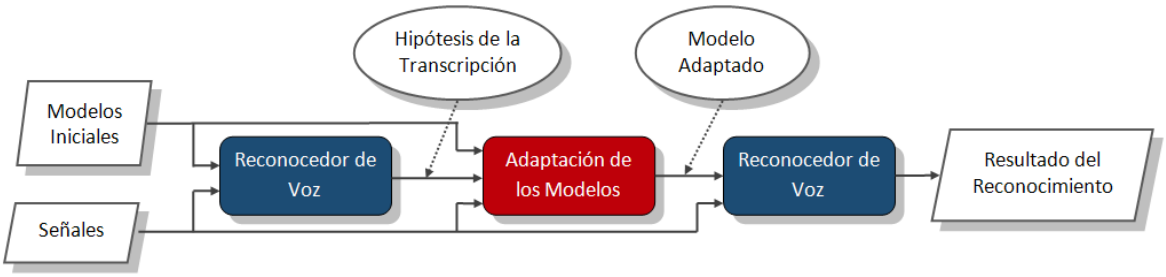


Figura 2.6: *Esquema de aprendizaje no supervisado en modo batch.*

Finalmente, los métodos de adaptación pueden clasificarse[1] de acuerdo a dos familias principales (Lee, 2008): *adaptación directa de los modelos* y *adaptación indirecta de los modelos*. A continuación se describen estos dos tipos de técnicas de adaptación.

2.3.2. Adaptación directa

En las técnicas de adaptación directa los parámetros del modelo son reestimados basándose en un criterio de optimización, utilizando aprendizaje bayesiano. Una caso representativo de esta familia y que ha sido ampliamente investigado corresponde a MAP (*Maximum a Posteriori*) (Gauvain & Lee, 1994). La formulación de MAP para estimar los parámetros es:

$$\Lambda_{MAP} = \arg\max_{\Lambda} g(\Lambda/X) \tag{2.20}$$

donde $g(\Lambda/X)$ es la función densidad de probabilidad (p.d.f) *a posteriori* de los parámetros del modelo Λ dada una señal acústica X. MAP asume que los parámetros del modelo Λ son vectores aleatorios descritos por su p.d.f $g(\Lambda)$, llamada distribución previa (*prior*). Utilizando las reglas Bayesianas, la ecuación (2.20) puede reescribirse como:

[1]Esta no es una clasificación estricta, ya que hay métodos que pueden pertenecer a ambas clases

$$\Lambda_{MAP} = \arg\max_{\Lambda} f(X/\Lambda) \cdot g(\Lambda) \tag{2.21}$$

donde $f(X/\Lambda)$ es la verosimilitud de los datos de adaptación X. La ecuación (2.21) indica que MAP combina información conocida de antemano (modelos de entrenamiento) con valores estimados de los datos de adaptación. Por lo tanto, resulta de vital importancia el tener una buena estimación de las distribuciones previas, para lo cual existen varios métodos en la literatura (Lee, 2008).

En (Gauvain & Lee, 1994) se reportan mejoras de entre un 35 % a un 46 % en el WER dependiendo de la cantidad de datos de adaptación al utilizar MAP. Cabe destacar que dado el diseño de MAP, la reestimación de los modelos se realiza sólo en las unidades acústicas que se encuentren en los datos de adaptación. Aquellos modelos que no hayan sido "vistos" no serán adaptados. Debido a esto, la mayor parte de los algoritmos que buscan mejorar MAP se enfocan en adaptar estas unidades. Entre estos métodos es posible mencionar a MAP basado en correlación (Huo & Lee, 1997) donde se asume una correlación entre las unidades acústicas y SMAP (*Structural Maximum a Posteriori*) (Shinoda & Lee, 1997) donde los parámetros del HMM están organizados en una estructura de tipo árbol. SMAP en particular presenta un desempeño muy superior a MAP cuando se utilizan pocas señales de adaptación, pero converge a un rendimiento similar a MAP cuando la cantidad de datos de adaptación es grande.

2.3.3. Adaptación indirecta

En los métodos de adaptación indirecta se calcula una transformada matemática la cual será aplicada a los parámetros del modelo. Un caso representativo de esta familia es MLLR (*Maximum Likelihood Linear Regression*) (Hamaker, 1999; Leggetter & Woodland, 1994),

en el cual las transformadas son obtenidas resolviendo un problema de maximización utilizando la técnica de EM (*Expectation Maximization*). La matriz de transformación utilizada para entregar un nuevo estimado de la media adaptada está dada por:

$$\hat{\mu} = W\xi \tag{2.22}$$

donde W es una matriz de transformación de $n \times (n+1)$ (donde n es la dimensión de los datos) y ξ es el vector extendido de las medias, el que corresponde a:

$$\xi = [w, \mu_1, \mu_2, \ldots, \mu_n]^T \tag{2.23}$$

donde w representa la constante de *offset* cuyo valor normalmente está fijo en 1. Cabe destacar que MLLR, a diferencia de MAP, realiza una adaptación de todo el modelo acústico gracias al uso de un árbol de regresión para relacionar las componentes Gaussianas. Este árbol se crea utilizando el modelo inicial del sistema (independiente de locutor).

En (Gauvain & Lee, 1994) se reportan mejoras de entre un 35 % a un 46 % en el WER dependiendo de la cantidad de datos de adaptación con esta técnica. La relativa simplicidad de implementación del método y el hecho de que funciona bien al considerar una pequeña cantidad de señales (decenas de segundos a minutos) han favorecido su uso masivo. Esto ha producido que se hayan desarrollado múltiples mejoras al método, entre las cuales es posible destacar MLLR restringido (*Constrained MLLR*) (Afify & Siohan, 2000) en el cual en vez de utilizar una estimación de máxima verosimilitud (ML) se utiliza un criterio de máximo a posteriori y SMAPLR (*Structural Maximum a Posteriori Linear Regression*) (Myrvoll, 2002). Estos métodos buscan algunos de los problemas de MLLR tales como la sensibilidad del rendimiento frente al número de transformadas que se utilizan, reduciendo así el peligro de sobreajuste que puede conllevar un aumento del WER.

A continuación se presentan de forma más detallada dos métodos de robustez en ASR que son utilizados en esta memoria: *Maximum Likelihood Linear Regression* (MLLR); y *Vocal Tract Lenght Normalization* (VTLN).

2.3.4. Maximum Likelihood Linear Regression (MLLR)

MLLR es una técnica de adaptación que estima un conjunto de transformadas lineales para la media[2] de un modelo acústico. El efecto de estas transformaciones es trasladar las medias de los componentes del sistema inicial para que cada estado del conjunto de HMM tenga una mayor probabilidad de generar los datos de adaptación.

Este método de adaptación puede ser aplicado de forma muy flexible, dependiendo de la cantidad de datos de adaptación que estén disponibles. Si sólo está disponible una pequeña cantidad de información entonces es posible generar una transformada *global* de adaptación, la cual es aplicada a cada Gaussiana que compone el conjunto de modelos. A medida que se cuenta con una mayor cantidad de datos de adaptación se puede realizar una adaptación mejorada aumentando el número de transformadas. Cada transformada es ahora más específica y puede ser aplicada a ciertas agrupaciones de componentes Gaussianas. Notar que MLLR adapta el conjunto completo de unidades acústicas aún cuando existan pocas señales de adaptación.

La agrupación de Gaussianas se realiza utilizando un árbol de regresión de clases. Este árbol se construye de forma tal que las componentes agrupadas estén cercanas en el espacio acústico, de forma tal que se puedan transformar de manera similar. La construcción puede realizarse de dos formas principales: un enfoque basado en conocimiento fonético (Leggetter & Woodland, 1994); y un enfoque basado en análisis de los datos acústicos.

[2]También es posible obtener transformadas para las varianzas, pero eso no fue utilizado en esta memoria

En el primer enforque todos los componentes Gaussianos están asignados inicialmente a una única clase global. Al ser necesarias más clases, debido a la disponibilidad de más datos, la clase global se divide de acuerdo a definiciones fonéticas amplias. Cada trifonema es mapeado en su fonema central para decidir su categoría fonética amplia y luego es asignado a una clase. Por ejemplo, la división en dos clases utiliza una clase para las vocales y otra clase para el resto de los otros fonémas. La división en 47 clases ocurre cuando cada fonema central es asignado a su propia clase, separado de los demás. Este método asegura que todas las Gaussianas de todos los estados de cada modelo están asignadas a la misma clase de regresión. Si bien en (Leggetter & Woodland, 1994) no se realizó una adaptación de los modelos de silencio, es posible incluir estos modelos también (Lee, 2008). En este caso, la primera división se realiza separando los modelos de silencio del resto en dos clases distintas.

Por otra parte, en el enfoque basado en análisis de los datos acústicos se construye el árbol agrupando los componentes que se encuentren cercanos en el espacio acústico. La agrupación se realiza utilizando algún algoritmo de *clustering* de los datos, sin utilizar conocimiento fonético de los componentes. En (Young *et al.*, 2001) el árbol se construyó utilizando un algoritmo de separación de centroides, ocupando distancia Euclidiana para cuantificar la cercanía de los modelos en el espacio acústico.

El uso de un árbol[3] de regresión de clases permite que la construcción de transformadas ocurra de manera dinámica y robusta. Como primer paso de la adaptación, el método dinámico determina si hay suficientes datos para la estimación de una transformada en cada una de las clases de regresión base (las hojas). Si hay suficiente información en uno de estos nodos, se determina una transformada específica para esa clase de regresión. En caso contrario,

[3]En una estructura de árbol binario, cada nodo puede estar conectado a cero, uno o dos nodos "hijos", los que se encuentran en un nivel inferior del árbol. Si está conectado a un nodo inferior se dice que es "padre" de ese hijo. El árbol se recorre comenzando por la "raíz" (nodo superior) hasta llegar a las "hojas", que son los nodos sin hijos.

se recurre a nodos superiores los cuales acumulan información de los "hijos" y se repite el proceso. Este proceso se repite hasta que cada una de las Gaussianas esté asociada a una transformada. En la Figura 2.7 se puede observar un diagrama de este proceso.

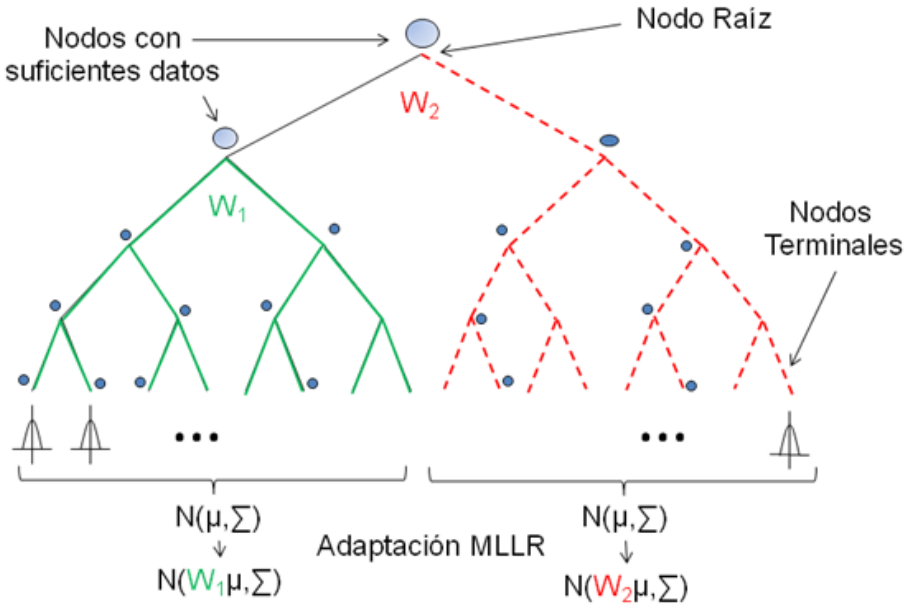


Figura 2.7: *Diagrama de un árbol de regresión de clases. En este ejemplo sólo 2 nodos tienen suficiente información de adaptación que permita determinar una transformada, por lo que se calculan dos transformadas, W_1 y W_2. Notar que todos los nodos base, donde están agrupadas las Gaussianas, están asociados a una de estas transformadas.*

El desarrollo matemático de MLLR con una matriz de transformación diagonal y considerando sólo adaptación de las medias de las Gaussianas se muestra a continuación (Lee, 2008). Sea $\{(\mu_j, \Sigma_j)\}$ el conjunto de todas las densidades Gaussianas con media μ_j y covarianza Σ_j de dimensión n. Sea además ν_j el vector extendido de medias definido como $\nu_j = [1, \mu_{j,1}, \ldots, \mu_{j,n}]^T$ y W_j una matriz de transformación de $n \times (n+1)$. Por definición, el vector de medias adaptado se obtiene como:

$$\hat{\mu}_j = W_j \nu_j \tag{2.24}$$

Usando la densidad adaptada la verosimilitud en un estado j está dada por:

$$p_j(o) = \frac{1}{(2\pi)^{n/2} \mid \Sigma_j \mid^{1/2}} e^{\{-\frac{1}{2}(o-W_j\nu_j)^T \Sigma_j^{-1}(o-W_j\nu_j)\}} \tag{2.25}$$

Las matrices W_j son compartidas por varias Gaussianas, lo que está determinado por el árbol de regresión. La estimación de éstas se realiza de forma tal que la verosimilitud de los datos de adaptación, dadas las densidades adaptadas, sea máxima. Este criterio de estimación resulta consistente con el criterio de ML (*Maximum Likelihood*) utilizado en el entrenamiento del modelo ácustico.

Sea $O_t = \{o_1, \ldots, o_T\}$ un conjunto de vectores de observación (datos de adaptación), $S = \{s_1, \ldots, s_T\}$ una secuencia de estados de HMM (el conjunto de todas las secuencias se denotará por S_T) y λ los parámetros del HMM. La verosimilitud completa de los datos de adaptación, dado el modelo adaptado está dada por:

$$\begin{aligned} p(O/\lambda, W) &= \sum_{S \in S_T} p(O, S/\lambda, W) \\ p(O, S/\lambda, W) &= \prod_{t=1}^{T} a_{s_{t-1}s_t} p_{s_t}(o_t/\lambda, W) \end{aligned} \tag{2.26}$$

donde $a_{s_{t-1}s_t}$ corresponde a la probabilidad de transición de estado entre los estados s_{t-1} y s_t y $p_{s_t}(o_t/\lambda, W)$ es la verosimilitud de un modelo adaptado específico.

Las transformadas W son estimadas utilizando el criterio de máxima verosimilitud dado por:

$$\hat{W} = \arg\max_{W} p(O/\lambda, W) \tag{2.27}$$

Este criterio es aplicado usando EM (*Expectation Maximization*), al maximizar la siguiente función auxiliar:

$$Q(W,\hat{W}) = \prod_{S \in S_T} p(O,S/\lambda,W) \log p(O,S/\lambda,\hat{W}) \tag{2.28}$$

Para una transformada en particular W_m, $\{m_1, m_2, \ldots, m_R\}$ corresponden a las R componentes Gaussianas que están agrupadas como queda determinado en el árbol de regresión de clases. Al maximizar la función auxiliar con respecto a la media transformada y considerando sólo estos componentes agrupados, es posible obtener lo siguiente:

$$\sum_{t=1}^{T} \sum_{r=1}^{R} L_{m_r}(t) \Sigma_{m_r}^{-1} o(t) \nu_{m_r}^{T} = \sum_{t=1}^{T} \sum_{r=1}^{R} L_{m_r}(t) \Sigma_{m_r}^{-1} W_m \nu_{m_r} \nu_{m_r}^{T} \tag{2.29}$$

y $L_{m_r}(t)$, la probabilidad de ocupación (también conocida como *state occupation count* o número de ocupación del estado) obtenida a partir de un proceso *forward-backward* (Young *et al.*, 2001), se define como:

$$L_{m_r}(t) = P(q_{m_r}(t)/M, O_T) \tag{2.30}$$

donde $q_{m_r}(t)$ representa la componente Gaussiana m_r en el instante t, $O_t = \{o(1), \ldots, o(T)\}$ corresponde a los datos de adaptación y M corresponde al conjunto de modelos.

Para obtener W_m, se definen dos nuevos términos: Z y G. El lado izquiero de la ecuación (2.29), designado comúnmente como Z, no depende de la matriz de transformación y corresponde a:

$$Z = \sum_{t=1}^{T} \sum_{r=1}^{R} L_{m_r}(t) \Sigma_{m_r}^{-1} o(t) \xi_{m_r}^{T} \tag{2.31}$$

Una nueva variable $G^{(i)}$ es definida con los elementos

$$g_{jq}^{(i)} = \sum_{r=1}^{R} v_{ii}^{(r)} d_{jq}^{(r)} \tag{2.32}$$

donde

$$V^{(r)} = \sum_{t=1}^{T} L_{m_r}(t) \Sigma_{m_r}^{-1} \tag{2.33}$$

$$D^{(r)} = \xi_{m_r} \xi_{m_r}^{T} \tag{2.34}$$

Puede observarse que a partir de estos dos términos W_m puede ser calculado como

$$w_i^T = G_i^{-1} z_i^T \tag{2.35}$$

donde w_i es el i-ésimo vector de W_m y z_i es el i-ésimo vector de Z. Estas ecuaciones pueden resolverse utilizando eliminación Gaussiana o descomposición LU para calcular W_m fila por fila.

Cabe recordar que el árbol de generación de clases es utilizado de forma dinámica para generar las clases, dado que la cantidad de información a adaptar puede variar. Esto implica que no se conoce *a priori* que clases de regresión serán utilizadas para estimar las transformadas. Esto no presenta un problema, ya que $G^{(i)}$ y Z para la clase de regresión escogida pueden ser obtenidas a partir de sus clases hijas (como están definidas por el árbol). Si el nodo padre R tiene como hijos las clases $\{R_1, \ldots, R_C\}$ entonces:

$$Z = \sum_{c=1}^{C} Z^{(R_c)} \tag{2.36}$$

$$G^{(i)} = \sum_{c=1}^{C} G^{(iR_c)} \tag{2.37}$$

Por lo tanto resulta claro que sólo es necesario calcular $G^{(i)}$ y Z solamente para las clases de regresión más específicas, esto es, las clases base.

2.3.5. Vocal Tract Length Normalization (VTLN)

Al implementar VTLN se presentan generalmente dos problemas (Zhan & Waibel, 1997) que pueden ser resueltos de distintas maneras: dado un conjunto de señales de voz de un locutor, cómo obtener el factor de normalización; y una vez obtenido un factor de normalización, cómo realizar la normalización. El primer problema generalmente se resuelve de dos formas: obtener el factor estimando el largo del tracto vocal (VTL) del locutor o realizando una búsqueda del factor entre un rango de posibilidades predefinido (*line search*). Para estimar el VTL se considera típicamente la relación existente de éste con las posiciones de las formantes. Sin embargo, este método tiene la desventaja de que la frecuencia de las formantes y su relación con VTL son altamente dependientes del contexto (podrían variar considerablemente en otros contextos para un mismo locutor) y es necesario separar y seleccionar los segmentes sonoros (*voiced*) de las señales de voz, ya que no tiene sentido calcular la frecuencia de las formantes utilizando segmentos no sonoros (consonantes y ruidos). Estos problemas dificultan la utilización de este método.

Debido a la razón mencionada arriba, han aparecido métodos que utilizan *line search* para obtener el factor de normalización (Lee *et al.*, 1998). Sea $\hat{\alpha}_i$ el factor de normalización

asociado a un locutor i. Sea λ un conjunto de HMMs, X_i^{α} corresponde a un conjunto de coeficientes cepstrales de un conjunto de uteraciones normalizados por el factor α, y W_i denota las transcripciones de las uteraciones. Entonces el factor de normalización óptimo de un locutor i se define como:

$$\hat{\alpha}_i = \arg\max_{\alpha} P(X_i^{\alpha}/\lambda, W_i) \tag{2.38}$$

En general resulta difícil encontrar una solución exacta de la ecuación (2.38) por lo que típicamente se realiza una búsqueda considerando un conjunto finito de factores de normalización (de 0,88 a 1,12 en (Lee *et al.*, 1998) donde 1 equivale al caso sin normalización).

La distorsión del eje de frecuencias del espectro se realiza normalmente considerando una función de *warping*. Una de las funciones más usadas corresponde a la función lineal por tramos (Pitz *et al.*, 2001), que se muestra en la Figura 2.8.

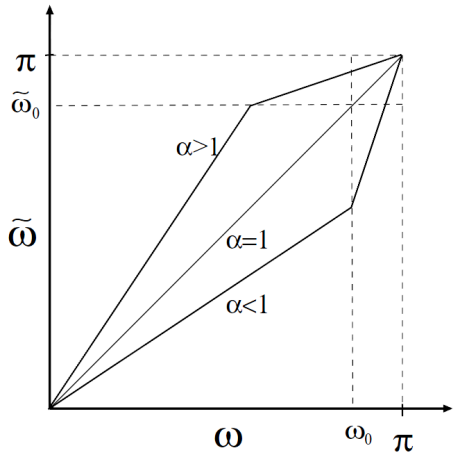


Figura 2.8: *Función lineal por tramos para distintos valores de* α*. Notar que* $\alpha \leq 1$ *corresponde a comprimir el espectro,* $\alpha = 1$ *es el caso sin distorsión y* $\alpha > 1$ *corresponde a estirar el espectro. Dado que el VTL de las mujeres es menor que en los hombres, resulta necesario estirar el espectro y, por ende, típicamente* α *es mayor que uno. En hombres ocurre el efecto opuesto.*

Esta función se define como:

$$g_{(\alpha,\omega_0)} : \omega \longrightarrow \tilde{\omega} = \begin{cases} \alpha\omega & \text{si } \omega \leq \omega_0 \\ \alpha\omega_0 + \frac{\pi-\alpha\omega_0}{\pi-\omega_0}(\omega-\omega_0) & \text{si } \omega > \omega_0 \end{cases} \tag{2.39}$$

donde ω_0 es el punto de inflexión donde cambia la pendiente de la función, el que se define como:

$$\omega_0 = \begin{cases} \frac{7}{8}\pi & \text{si } \alpha \leq 1 \\ \frac{7}{8\cdot\alpha}\pi & \text{si } \alpha > 1 \end{cases} \tag{2.40}$$

En esta memoria se considerará una modificación al método anterior, donde el *warping* se realizará sobre los centros de las frecuencias de los filtros en escala Mel (Lee *et al.*, 1998; Panchapagesan & Alwan, 2008). Para estimar la energía asociada a cada filtro se realizará una interpolación considerando las energías de los filtros sin distorsión (Molina *et al.*, 2009).

2.4. Rol de las tecnologías de voz en el aprendizaje de idiomas

2.4.1. Motivación

La utilización de tecnologías computacionales para el apoyo en el aprendizaje de idiomas es cada vez más común estos días. Los sistemas típicos contienen ejercicios en los que se acepta y evalúa texto ingresado por el alumno, y que en algunos casos consideran reproducción de voz.

Entre las ventajas de estas herramientas es posible mencionar que proporciona un grado de interactividad mayor al alumno, el que participa activamente de los ejercicios, en una relación uno a uno con el programa computacional. Esto difiere del formato clásico del aprendizaje de un idioma, donde un profesor debe enseñarle a varios alumnos, y por lo tanto debe dividir su atención. Esto produce además que las clases sean mayormente expositivas, sin que el alumno pueda poner en práctica sus conocimientos para hablar el segundo idioma de forma continua, ya que no es posible para el profesor el evaluar a todos al mismo tiempo.

Actualmente, un área de investigación de gran interés consiste en desarrollar sistemas basados en reconocimiento de voz que puedan complementar las enseñanzas de compresión de lectura y de escucha, con actividades más activas que involucren conversación y diálogo por parte del alumno. Estos sistemas podrían plantear problemas seleccionados por profesores calificados, analizar la respuesta del alumno, realizar una evaluación de su pronunciación, por ejemplo, y luego presentar al alumno los errores que puede haber cometido, de forma automática.

En un estudio realizado por el Dr. Ray Clifford, del Defense Language Institute (DLI), se concluye que alumnos que han sido instruidos sólo en áreas de lectura, escritura y audición de un nuevo lenguaje, obtienen peores resultados, tanto en comprensión de lectura como en audición, que aquellos estudiantes que además fueron entrenados para hablar la segunda lengua (Price & Rypa, 1998).

La posibilidad de recibir habla humana por parte de aplicaciones de instrucción de lenguaje basadas en computador, permite complementar las actividades de lectura y audición, con actividades de producción de lenguaje y entregar respuestas tal como lo haría un instructor humano, como calificar la pronunciación realizada o identificar los errores cometidos (Franco *et al.*, 1997).

Se menciona reiteradamente que es necesario diferenciar dos tipos de público objetivo: Aquellos que están comenzando a aprender el idioma, y aquellos que ya tienen conocimientos, pero que necesitan reforzarlos o mantenerlos en el tiempo. Para el primer grupo, resulta necesaria la existencia de material explícito y de dificultad progresiva, en cambio para el segundo grupo es necesaria la existencia de material tanto que sea adecuado para su nivel, como que trate tópicos de interés (Price & Rypa, 1998).

En el sistema VILTS (*Voice Interactive Language Training System*, Sistema Interactivo de Voz para el Entrenamiento del Lenguaje) desarrollado por SRI Labs. En (Price & Rypa, 1998) se proponen métodos para hacerse cargo de todas las instancias comprometidas en el aprendizaje de un idioma, considerando tres tipos de aprendizaje: Aprender a Escuchar; Aprender a Leer; y Aprender a Hablar, que es el que se relaciona mayormente con el tema de esta memoria.

Según el sistema anterior, el habla es la habilidad principal en el estudio de cualquier lenguaje. Se ha demostrado que la habilidad en el habla mejora la habilidad de escuchar, y que ambos en conjunto forman el propósito principal de un lenguaje: La comunicación interactiva. Se distinguen dos áreas para la evaluación del habla: qué se dijo, y cómo se dijo. En el diseño de una aplicación interactiva es importante intentar entender qué es lo que se dijo, independiente de qué tan mal se haya dicho. En cambio, para responder a la pregunta de cómo se dijo, es importante que la aplicación pueda entregar una calificación correlacionada con aquella que entregaría un nativo experto en el idioma.

2.4.2. Estado del arte de la aplicación de técnicas de voz al aprendizaje de un segundo idioma

La pronunciación de una elocución y su posterior calificación corresponde a uno de los problemas principales que se ha intentado resolver, en el marco de la aplicación al aprendizaje de un segundo idioma y del reconocimiento de voz en general. Los CAPT (*Computer Aided Pronunciation Training*, Entrenamiento de Pronunciación Asistido por Computador) corresponden una posibilidad interesante para el uso masivo de tecnologías de voz. Estos sistemas actúan como tutores virtuales, invitando a los estudiantes a repetir determinadas palabras con el fin de practicar y mejorar la calidad del lenguaje hablado, específicamente en lo que respecta a la producción de sonidos (características segmentales).

Una de las primeras formas que se mencionan en la literatura especializada de entregar una evaluación numérica a la pronunciación de un usuario, es mediante la verosimilitud entregada por el algoritmo forzado de Viterbi. Este algoritmo obtiene el logaritmo de la probabilidad de una secuencia de observaciones en un modelo basado en HMM. Entre las mejoras que se han implementado a esta metodología, se encuentra el normalizar la verosimilitud por la duración de cada estado. Aún así los resultados que se obtienen no son suficientemente correlacionados con las calificaciones de un humano experto (Franco *et al.*, 1997).

Una técnica que ha presentado buenos resultados es el método de la probabilidad a posteriori (Franco *et al.*, 1997), el que ya fue descrito con anterioridad en este capítulo. Con este método, se ha podido obtener correlaciones similares a las que se obtienen al correlacionar dos calificadores humanos expertos. Resulta importante destacar que la correlación entre las evaluaciones de dos calificadores expertos (profesores de lingüística por ejemplo) es del orden de 0,76-0,87 (Franco *et al.*, 1997), por lo que esto introduce inmediatamente una restricción a la correlación máxima posible de obtener para un sistema automático de

reconocimiento de voz, ya que éstos deben ser entrenados considerando calificaciones de distintos evaluadores que en ocasiones difieren en sus puntuaciones para una misma señal.

La utilización de información acústica a priori de individuos parlantes no nativos, puede resultar ser una fuente importante de información para discriminar entre pronunciaciones correctas o incorrectas. Un ejemplo de esto es el caso de un chileno que comience a aprender inglés y pronuncie la palabra "tool". Resulta muy probable que esta pronunciación considere una t dental, que es la pronunciación típica de esta consonante de un hablante nativo hispano, en vez de una t alveolar, que es la correcta pronunciación de un nativo del idioma inglés. La naturaleza de estos errores típicos depende del idioma nativo del individuo y pueden ser utilizadas para mejorar los modelos utilizados por un sistema CAPT. Sin embargo, la modelación de estos errores de pronunciación exige la grabación de bases de datos sofisticadas que incorporen varios tipos de errores en cada lenguaje.

Una variante al método tradicional de evaluación de pronunciación mediante el algoritmo de Viterbi es sencillamente no forzar la búsqueda a sólo una palabra, sino que realizarla sobre palabras competidoras. Por ejemplo en (Indrayanti *et al.*, 2006) se evaluó el rendimiento de la tecnología de ASR en evaluación de pronunciación para un set finito de errores. En este caso la tarea del ASR sería entregar la palabra que más se asemeja a la pronunciada por un usuario. La dificultad estaría centrada en generar las palabras competidoras para cada evaluación y establecer escalas graduales. En (Molina *et al.*, 2008) se desarrolló un método automático para la generación de estas palabras competidoras, basado en la distancia K-L (Kullback-Leibler) entre los modelos.

A continuación se describen aspectos básicos del sistema de evaluación de pronunciación basado en ASR desarrollado por el Laboratorio de Procesamiento y Transmisión de Voz (LPTV) de la Universidad de Chile ((Molina *et al.*, 2008)), el cual se utilizó en el capítulo 4 de esta memoria.

2.4.3. Sistema de evaluación de pronunciación del LPTV

El problema de evaluación de pronunciación radica en cómo mapear las medidas objetivas que entrega el ASR a medidas subjetivas que imitan la evaluación que entregaría un profesor humano. Sea M la cantidad de niveles de cuantización de las notas (*scores*) subjetivos (en este trabajo se utilizará $M = 2$ y $M = 5$). Cada métrica de confiabilidad puede asumirse como una nota entregada por un clasificador y cada nivel de nota subjetivo sería una clase. Sea O la secuencia de vectores de observación correspondientes a la frase pronunciada por un estudiante. Usando la regla de Bayes, el nivel de de nota subjetivo puede estimarse de una métrica individual WF_j como:

$$d_{WF_j}(O) = \arg\max_{C_m} P(C_m/WF_j(O)) = \arg\max_{C_m} \left\{ \frac{P(WF_j(O)/C_m \cdot P(C_m)}{P(WF_j(O))} \right\} \tag{2.41}$$

donde: WF_j corresponde a las características de las palabras (métricas de confiabilidad) extraidas del ASR; d_{WF_j} es la nota elegida por el clasificador asociado a WF_j; y C_m es una clase asociada al nivel de nota subjetivo m, donde $1 < m < M$. $P(C_m)$ se considera distribuida uniformemente e igual a $1/M$. De esta forma, sólo resulta necesario estimar las f.d.p *a priori* $P(WF_j(O)/C_m$ y $P(WF_j(O))$, lo que se realiza utilizando una base de señales y su evaluación subjetiva. En la Figura 2.9 se muestra un diagrama de bloques que describe este proceso.

En este trabajo se utilizó dos métricas de confiabilidad ((Molina *et al.*, 2008)) para la evaluación de la pronunciación: WD (*Word Density*); y POS.

Para una palabra objetivo w_t WD se define (Kwan *et al.*, 2002) en la ecuación (2.42).

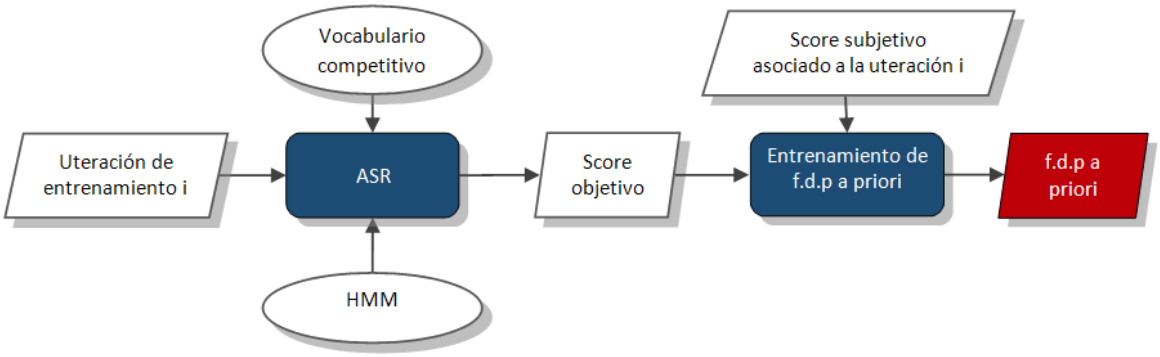


Figura 2.9: *Diagrama de bloques que describe el proceso de entrenamiento de las curvas (o f.d.p) a priori.*

$$WDCM_t = \frac{\sum_{r \in E(w_t,H)} Q(h_r)}{\sum_{l=1}^{N} Q(h_l)} \tag{2.42}$$

$$Q(h_r) = P(h_r)^{\gamma} P(O/h_r)$$

donde h_r es la r-ésima hipótesis en la lista de las N-mejores (*N-Best*) de Viterbi; $Q(h_r)$ es la calificación de verosimilitud que entrega la búsqueda de Viterbi; $P(h_r)$ es la probabilidad del modelo de lenguaje de h_r; $P(O/h_r)$ es la probabilidad de observación de h_r; γ es factor de escalamiento del modelo acústico; $E(w_t,H)$ corresponde a los índices de las hipótesis que contienen la palabra w_t; y finalmente, H representa todos los alineamientos de las N-mejores hipótesis obtenidas de la decodificación de Viterbi.

Por otro lado, POS corresponde al índice de la hipótesis más probable donde la palabra objetivo, w_t fue reconocida (ecuación (2.43)).

$$POS_t = \arg\max_r \{[Q(h_r)] \mid r \in E(w_t,H)\} \tag{2.43}$$

2.5. Conclusiones

En este capítulo se ha entregado una base teórica suficiente para comprender las técnicas y análisis presentados en los siguientes capítulos de esta memoria. En base a estos antecedentes, es posible afirmar que el problema de robustez frente a cambios de locutor de los sistemas de reconocimiento de voz es complejo y se encuentra lejos de estar completamente resuelto. Si bien en la actualidad existe una gran variedad de técnicas de adaptación y normalización que buscan reducir los efectos de la variabilidad inter-locutor que sufren los sistemas dependientes de locutor, estas técnicas presentan ventajas y desventajas entre sí (ya sea en desempeño, funcionamiento con muy poca información (una señal), facilidad de implementación, etc.) que dificultan su elección para una aplicación específica.

Por tales motivos, uno de los objetivos de este trabajo es realizar una comparación entre dos de los métodos más utilizados actualmente (VTLN y MLLR) basada en el desempeño obtenido tanto en reconocimiento de voz como en evaluación de pronunciación basada en ASR. Los resultados obtenidos son contrastados con los presentes en la literatura (Panchapagesan & Alwan, 2008; Wang *et al.*, 2007; Lee, 2008) para su validación.

Capítulo 3

Implementación de técnicas de robustez en ASR

3.1. Introducción

En este capítulo se describe la implementación de *Maximum Likelihood Linear Regression* (MLLR) y *Vocal Tract Length Normalization* (VTLN) en un reconocedor automático de voz (ASR, *Automatic Speech Recognition*). Estas técnicas buscan otorgar robustez al sistema frente a cambios de locutor. MLLR realiza una transformación lineal de los modelos acústicos con el fin de que se adapten a las condiciones de prueba. Por otra parte, VTLN produce una distorsión (*warping*) del eje de frecuencias del banco de filtros utilizado en la parametrización de las señales. El objetivo es reducir el efecto de las diferencias en la longitud del tracto vocal entre los locutores, que es una de las principales causas de variabilidad inter-locutor.

Estos métodos se implementan de manera no supervisada, siguiendo el esquema de la Figura 2.6, es decir, ocupando las transcripciones obtenidas directamente del reconocimiento. Al final de este capítulo se muestra una serie de experimentos realizados con el fin de evaluar el desempeño del sistema para distintas configuraciones.

3.2. Implementación de MLLR en ASR

La implementación de MLLR (*Maximum Likelihood Linear Regression*) se realiza utilizando el *toolkit* HTK (Young *et al.*, 2001), haciendo uso principalmente de las herramientas HEA_{DAPT} y HHE_D. HTK es un conjunto de herramientas para construir HMMs y que contiene además funciones que permiten el reconocimiento y la posterior adaptación de estos modelos acústicos. En esta memoria, HTK es usado para la construcción de los HMMs y la adaptación de los modelos mediante MLLR.

Es posible identificar tres etapas fundamentales en la adaptación con MLLR: construcción del árbol de regresión de clases; cálculo de las matrices de transformación; y reconocimiento final utilizando los modelos adaptados. A continuación se explica en detalle la implementación de cada uno de estos procesos.

3.2.1. Construcción del árbol de regresión de clases

El árbol de regresión de clases (ver Figura 2.7) permite que la construcción de transformadas de MLLR ocurra de manera dinámica y robusta. Esto se logra mediante la agrupación de densidades Gaussianas cercanas en el espacio acústico, lo que permite el cálculo de matrices de transformación para un conjunto de Gaussianas incluso cuando la información disponible

sea pequeña. El proceso general de construcción del árbol se muestra en la Figura 3.1.

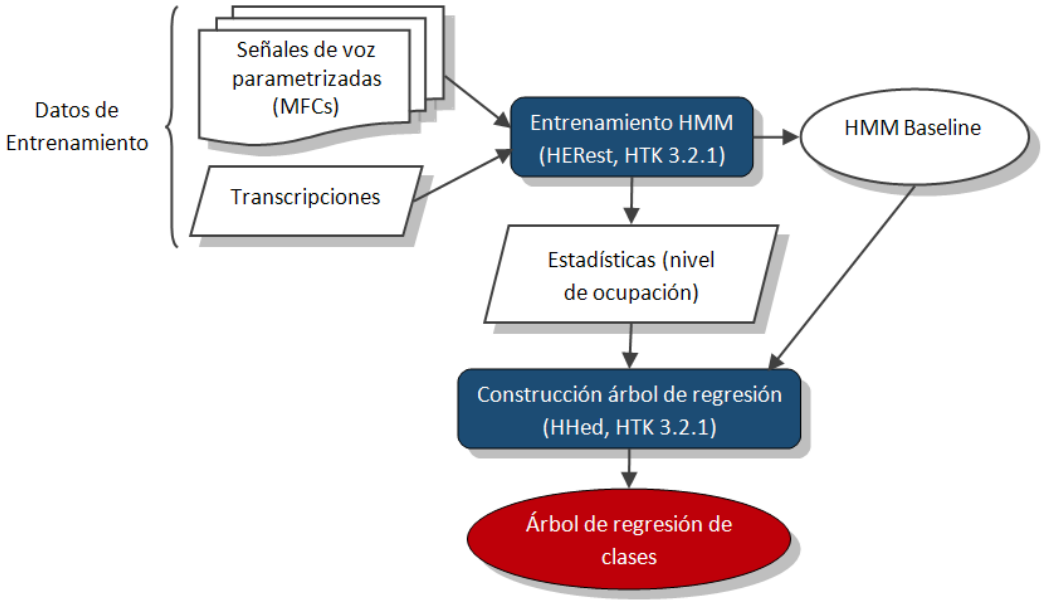


Figura 3.1: *Diagrama de bloques que describe el proceso de construcción del árbol de regresión de clases para MLLR.*

La primera etapa del proceso ocurre durante la etapa final del entrenamiento del modelo acústico, el cual se realiza utilizando la función HER_{EST} de HTK. HER_{EST} es un programa que realiza una reestimación de los parámetros de un conjuto de HMMs mediante el algoritmo de Baum-Welch (Young *et al.*, 2001). La aplicación recibe como entradas: un archivo de configuración; el listado de las señales de entrenamiento parametrizadas en el dominio MFCC (*Mel Frequency Cepstral Coefficients*); las transcripciones correctas de estas señales; el listado de trifonemas (cada uno de los cuales tiene asociado un modelo acústico en el HMM); y, los HMM de la etapa anterior de entrenamiento. El proceso de entrenamiento considera un HMM inicial (hmm0) que es utilizado en la primera etapa de entrenamiento del modelo acústico. Este HMM contiene modelos de los monofonemas considerando una única Gaussiana, donde cada media y varianza es la misma.

Durante la última reestimación del conjunto de HMMs resulta necesario agregar la opción "-s" de HER_{EST} para habilitar la creación de un archivo de estadísticas del modelo acústico. Este archivo contiene el número de ocupación (*occupation count*) de todos los estados del conjunto de HMM que está siendo entrenado y es utilizado para realizar el proceso de *clustering* de los estados del HMM durante la construcción del árbol de regresión de clases. El término número de ocupación corresponde a la cantidad de *frames* asociados a un estado en particular y puede ser utilizado como medida de la cantidad de datos de entrenamiento que hay disponibles para la estimación de los parámetros de ese estado.

Una vez que se ha obtenido el archivo de estadísticas, es posible crear el árbol de regresión usando el programa HHE_D. Esta herramienta de HTK crea el árbol y lo almacena como parte del HMM que recibe como entrada. El proceso de clusterización de los estados se realiza mediante un algoritmo de separación de centroides, utilizando como métrica la distancia euclidiana. El programa recibe como entrada: los HMMs del sistema independiente de locutor; la carpeta donde se guardará el HMM con el árbol; un *script* con instrucciones para HHE_D; y un listado de los trifonemas entrenados. Las instrucciones que se entregan a HHE_D en este caso corresponden a renombrar el conjunto de HMMs, cargar el archivo de estadísticas obtenido anteriormente y especificar la cantidad de nodos terminales (nodos base del árbol que no tienen asociados hijos) que debe tener el árbol. A continuación se muestra un extracto del árbol de regresión que se obtiene como resultado de este proceso:

```
    r "tree_32"
<REGTREE> 32
<NODE> 1 2 3
<NODE> 2 4 5
<TNODE> 28 4104
<TNODE> 30 3040
...
```

El árbol de regresión queda entonces descrito por nodos no terminales ("<NODE>") y los nodos terminales ("<TNODE>"). Cada nodo contiene su índice (primer número) seguido de los índices de sus "hijos" (si se trata de un nodo no terminal) o del número de densidades que componen el nodo terminal. Además de almacenar el árbol de regresión en el HMM, HHE_D agrega a cada modelo la palabra clave "<RCLASS>" seguida del identificador del nodo terminal al cual está asociado. De esta forma, tanto el árbol y el modelo acústico sin adaptar se encuentran completamente definidos en un único archivo.

3.2.2. Cálculo de las matrices de transformación

El siguiente paso en la aplicación de MLLR corresponde a calcular las matrices de transformación, dado un conjunto de test. Estas matrices intentarán adaptar el espacio acústico de forma específica para cada locutor, con el fin de disminuir el WER del sistema. En la Figura 3.2 se puede observar un diagrama de bloques de este proceso.

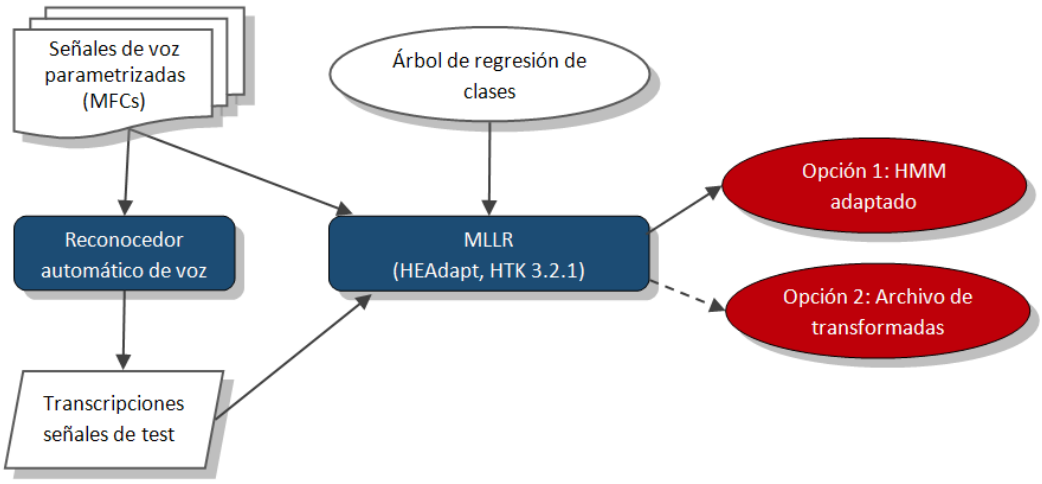


Figura 3.2: *Diagrama de bloques que describe el proceso de cálculo de las matrices de transformación de MLLR.*

En primer lugar es necesario obtener las transcripciones reconocidas de las señales de test usando el sistema independiente de locutor (*baseline*). Nótese que esto implica que estas transcripciones pueden presentar errores ya que el reconocedor no es perfecto.

El reconocedor entrega un archivo con un identificador de cada señal y el listado de palabras reconocidas. Este formato difiere del usado por HTK para la adaptación, el cual necesita recibir como entrada las palabras descompuestas en trifonemas. Para ejecutar esta conversión se diseñó un programa[1] el cual realiza varios llamados a la función HLE_D de HTK. Este programa recibe como entrada la transcripción y un diccionario con las palabras incluidas en el modelo de lenguaje con su descomposición en monofonemas.

Las señales de la base de test utilizadas para la adaptación con MLLR deben ser parametrizadas primero al dominio MFC. El proceso completo utilizado es el siguiente: primero se aplica a la señal acústica un detector de inicio-fin para eliminar los silencios antes y después de los pulsos de voz; después se segmenta la señal en *frames*, usando la ventana de Hamming; los MFCC son calculados; se aplica CMN (*Cepstral Mean Normalization*), que es una técnica de cancelación de ruido; y, finalmente, se convierten los MFCC al formato requerido por HTK. Cada señal acústica utilizada es parametrizada en 33 coeficientes por *frame*. Los primeros once corresponden a diez coeficientes cesptrales y la energía del *frame*. Los siguientes once corresponden a los deltas (derivadas temporales de estos coeficientes) y los subsiguientes a los deltas-deltas (doble derivada temporal).

La adaptación de los CDHMMs con MLLR se llevó a cabo utilizando la herramienta HEA_{DAPT} de HTK. Este programa funciona de la siguiente forma. Al comenzar, HEA_{DAPT} carga un conjunto completo de definiciones de HMM, incluyendo el árbol de regresión de clases y el índice de la clase terminal de cada componente Gaussiana.

[1] Toda la programación se realizó en lenguaje c/c++ utilizando Borland C++ Builder 5.

A continuación HEA_{DAPT} comienza a procesar las señales. Después de cargar una señal en memoria, HEA_{DAPT} utiliza la transcripción asociada para construir un HMM compuesto que representa a toda la uteración. Este HMM compuesto se obtiene concatenando los HMM de trifonemas correspondientes a la transcripción, considerando el modelo acústico independiente de locutor (*baseline*). El algoritmo de *Forward-Backward* se aplica entonces para obtener el alineamiento entre los estados (del HMM) y los frames (de la señal acústica), y la información necesaria para formar la función auxiliar estándar (ecuación (2.28)) es acumulada para cada componente Gaussiana. Cuando todas las señales han sido procesadas, se calculan las estadísticas de las clases base de regresión, utilizando las estadísticas por componente. Posteriormente se recorre el árbol de regresión y se realiza el cálculo de transformadas para aquellas clases de regresión que presenten suficiente información (mayor que el número de ocupación elegido como límite inferior), como se muestra en la figura 2.7.

Una vez finalizado el proceso anterior, HEA_{DAPT} genera el conjunto adaptado de HMMs o un archivo con las matrices de transformación. Este archivo puede ser utilizado para transformar el HMM independiente de locutor inicial a un nuevo locutor, basado en los datos de adaptación considerados. El resultado de esta transformación es el mismo modelo acústico que se puede obtener como salida directa de HEA_{DAPT}.

Al realizar la implementación de MLLR utilizando HTK se identificaron cuatro parámetros que afectan significativamente el proceso de adaptación. Estos son: cantidad de nodos terminales del árbol de regresión de clases; cantidad de uteraciones (ventana) utilizadas; límite inferior del número de ocupación; y tipo de matriz de adaptación. A continuación se presenta en detalle cada una de estas variables.

El primer parámetro de importancia que debe especificarse antes de aplicar MLLR es la cantidad de nodos terminales que forman el árbol de regresión. Esto determina la cantidad máxima de transformadas que se pueden calcular y cómo se asociarán los estados de los

trifonemas (HMMs) a cada nodo terminal. El número de nodos sugerido por HTK es de 32.

La cantidad de uteraciones por locutor que se desean utilizar en la adaptación también puede variar, dependiendo de las restricciones de la base de datos utilizada. Un aumento de la ventana implica un aumento de la cantidad de información disponible. Por ende, MLLR calculará un mayor número de matrices de transformación, debido a que la cantidad de nodos terminales que superan la restricción impuesta por el límite inferior del número de ocupación aumenta.

En la función HEA_{DAPT} no es posible configurar la cantidad de transformadas que se desea obtener de forma directa. Sí es posible definir un valor límite de ocupación mínimo (*threshold*) para determinar si se obtiene o no una transformada asociada a ese nodo terminal. De esta forma, un límite del número de ocupación bajo implica que se calculará una mayor cantidad de transformadas (más nodos terminales superan el límite) para una ventana dada. El valor por *default* de este número en HTK es de 700.0, con el cual no se calculan transformadas al considerar una ventana de 1 uteración.

Finalmente, en HTK es posible especificar el tipo de matriz de adaptación que se desea utilizar, lo que determina la cantidad de pesos que se calculan para la adaptación de las medias. Se distinguen tres tipos de matrices principales: matriz por tramos; matriz diagonal; y matriz completa. En el caso de matriz completa, se calculan los pesos asociados a todas las combinaciones de coeficientes para la transformación, obteniéndose una matriz de transformación de 33×33 donde ningún valor es igual a cero. Es decir, se considera que existe una correlación entre todos los parámetros. En el caso de una matriz por tramos, se considera que no existe una correlación entre los coeficientes estáticos (los primeros once), los delta y los delta-delta, por lo que se utiliza una matriz definida por bloques (ver ecuación (3.1)). Finalmente, es posible asumir que no existen correlaciones entre los parámetros, con lo cual sólo es necesario calcular una matriz de transformación de las medias diagonal. HEA_{DAPT}

considera por defecto una matriz de transformación por tramos.

$$\mathbf{A} = \begin{pmatrix} A_s & 0 & 0 \\ 0 & A_\Delta & 0 \\ 0 & 0 & A_{\Delta^2} \end{pmatrix} \tag{3.1}$$

La primera implementación realizada de MLLR se basó en la generación de los archivos de transformación y, directamente desde el ASR, cargar las matrices de transformación para realizar la adaptación. Este proceso se implementó completamente. El código requerido para recorrer el árbol, identificar los modelos y las transformadas asociadas fue generado sin encontrar diferencias con respecto al HMM ya adaptado que entrega HTK. Sin embargo no fue posible realizar el reconocimiento con los modelos adaptados de forma correcta, por complicaciones con el proceso de Viterbi que no fue posible resolver. Por lo tanto, fue necesario desarrollar nuevamente la implementación considerando ahora los HMMs ya adaptados.

Para la generación de los HMMs se diseñó un programa que recibe los nombres de las carpetas donde se encuentran los archivos necesarios y los parámetros de MLLR que se usarán. Este programa genera los comandos de HEA_{DAPT} para la obtención de los HMMs asociados a cada una de las ventanas con la configuración especificada. A continuación se muestran los campos que presenta este archivo de configuración y qué recibe cada uno de ellos:

Parámetros	Descripción
[PATH_MFCS_ADAPTACION]	Carpeta de las señales parametrizadas en MFCC
[NUM_INICIO_MFC]	Primera señal que se utilizará
[NUM_FIN_MFC]	Última señal que se utilizará
[PATH_HMM_SALIDA]	Carpeta donde se guardarán los HMMs adaptados
[TRANSCRIPCION_MFCS]	Archivo con las transcripciones de las señales
[LISTADO_TRIFONEMAS]	Listado de trifonemas del modelo acústico
[PATH_HMM]	Carpeta con el HMM *baseline*
[TAM_VENTANA]	Tamaño de la ventana
[TIPO_MATRIZ_TMF]	Tipo de matriz de transformación
[NUM_OCUPACION]	Límite inferior al número de ocupación
[ETIQUETA]	Sufijo identificador (adicional) de la carpeta salida
[NUM_NODOS]	Número de nodos del árbol de regresión

El ASR utiliza una versión binarizada del reconocedor con el fin de reducir el espacio requerido para su almacenamiento. Además, se realiza una reducción de este modelo, eliminando los trifonemas que no se encuentren en el modelo de lenguaje y que por lo tanto no serán utilizados en el proceso de reconocimiento. Para este fin normalmente se utiliza un programa de binarización programado por miembros del LPTV (Laboratorio de Procesamiento y Transmisión de Voz de la Universidad de Chile). El proceso de binarización utilizado en los modelos adaptados obtenidos con HTK presenta algunas diferencias que dificultan el uso directo de este programa. HEA_{DAPT} incorpora líneas al HMM con información nueva (árbol de regresión de clases e identificador de la clase de los modelos) que no es de utilidad después de la adaptación y que no es "interpretada correctamente" por el programa de binarización (debido a que cambia el contenido del archivo). Por otra parte, la utilización de ventanas para generar las transformadas implica que se trabajará con una gran cantidad de HMMs,

cada uno de los cuales debe ser binarizado. Debido a estas razones, se creó un programa que recibe un listado de HMMs (en un archivo .txt) y realiza su binarización, eliminando primero las líneas extras agregadas por HEA_{DAPT}.

3.2.3. Reconocimiento final utilizando los modelos adaptados

Una vez obtenidos los HMMs de la forma descrita anteriormente es posible realizar el reconocimiento de las señales de test utilizando los modelos adaptados (dependientes de locutor). Este proceso se ilustra en la Figura 3.3.

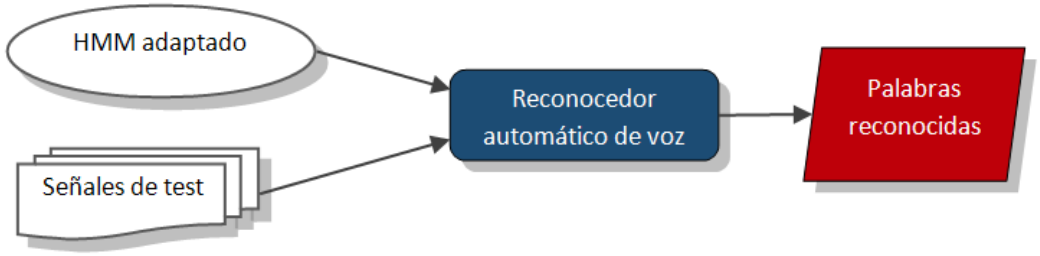


Figura 3.3: *Diagrama de bloques que describe el proceso de reconocimiento final utilizando los modelos adaptados con MLLR.*

Las pruebas de reconocimiento se realizaron utilizando el ASR con arquitectura cliente-servidor desarrollado por el LPTV. El servidor actúa básicamente como reconocedor, cargando los modelos de lenguaje y acústicos en memoria, y preparándose para la recepción de señales enviadas por el cliente. Este último envía las señales de audio y espera la recepción de las transcripciones de las señales una vez finalizado el reconocimiento.

El servidor utiliza un archivo de configuración en el cual se especifica la carpeta donde están los modelos de lenguaje que deben ser guardados en memoria al iniciar la ejecución del programa. Dado a que se están utilizando HMMs adaptados entrenados para ventanas de señales específicas, es necesario cerrar el servidor, cambiar los modelos correspondientes

y volver a comenzar la ejecución del servidor cada vez que se desee cambiar el modelo acústico. Más aún, el cliente también debe enviar las señales correspondientes a ese HMM solamente. Para realizar estas acciones de forma automática se creó un programa que genera un archivo .bat con todas las instrucciones que deben ejecutarse.

El programa anterior recibe las siguientes entradas:

Parámetros	Descripción
PATH_CLIENTE	Carpeta donde se encuentra el cliente
PATH_SERVER	Carpeta donde se encuentra el servidor
PATH_MODELOS	Carpeta donde se guardarán los modelos que cargará el ASR
PATH_WAVS	Carpeta con las señales de test
PATH_BIN	Carpeta con los HMMs adaptados
VENTANA	Tamaño de la ventana
NUM_SENALES	Cantidad de señales de la base de test

El resultado final de la ejecución del programa es un archivo con las transcripciones obtenidas con los modelos adaptados utilizando MLLR.

3.3. Implementación de VTLN en ASR

Para realizar las pruebas con VTLN se utilizó la implementación descrita en (Molina *et al.*, 2009). En dicho trabajo, la transformación o *warping* se realiza sobre el banco de filtros Mel que se aplica sobre el espectro, en vez de aplicarlos directamente sobre éste último. Se pueden distinguir tres rutinas principales que interactúan en esta implementación: un

proceso principal de búsqueda del α[2] óptimo; un subproceso de transformación del banco de filtros dado un cierto α, el que se realiza en cada iteración del proceso principal; y la etapa de reconocimiento final utilizando el factor de normalización óptimo. A continuación se explicará con mayor detalle la ejecución de VTLN en sus distintas etapas.

3.3.1. Transformación (*Warping*) del banco de filtros

El proceso de parametrización cepstral del *frame* de una señal de voz y cómo se ve afectado por el factor de normalización de VTLN se muestra en la Figura 3.4.

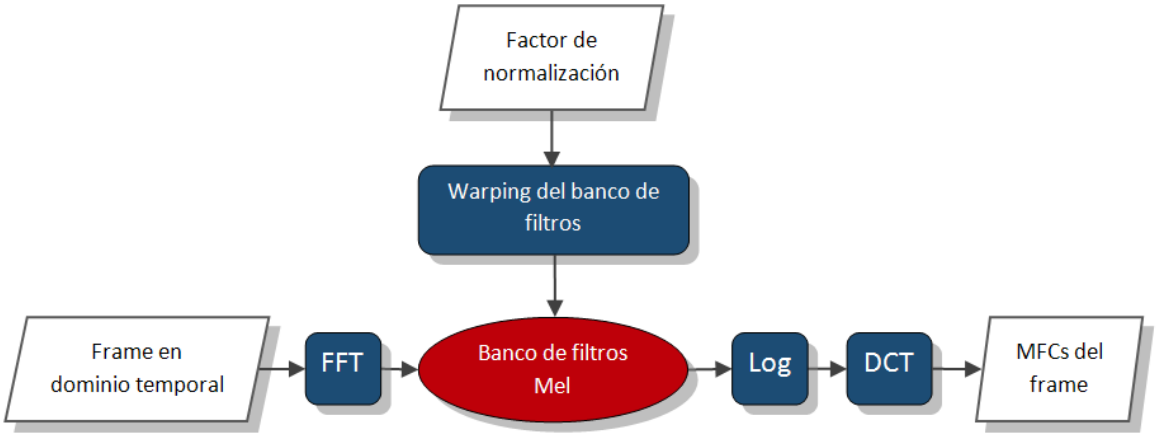


Figura 3.4: *Diagrama de bloques que describe el proceso de parametrización cepstral del frame de una señal de voz y cómo se ve afectado por VTLN.*

El *warping* del banco de filtros se realiza en una función implementada dentro del reconocedor. Esta función recibe como entradas el factor de normalización con el que se debe realizar el *warping* y una estructura llamada latino que contiene los parámetros de la señal que se obtienen al aplicar el banco de filtros original. El programa cuenta además con la información de las frecuencias centrales de cada uno de los 14 filtros de Mel utilizados en la

[2]El factor de normalización se designa normalmente con la letra griega α

parametrización. Esta rutina se ejecuta en cada una de las iteraciones de búsqueda del factor de normalización óptimo y una vez más en el reconocimiento final, para obtener las palabras reconocidas al aplicar VTLN.

El primer paso del proceso corresponde a la creación de la función de *warping* lineal por tramos. La forma de esta función es similar a la que se muestra en la ecuación (2.39) pero modificada para ser aplicada a las frecuencias centrales de los filtros (en vez de considerar frecuencias entre 0 y π, se define la función y el punto de inflexión considerando los valores de las frecuencias centrales). Una vez calculada la función, se procede a determinar los nuevos valores en el eje de frecuencias de estos filtros.

Una vez finalizado lo anterior se debe calcular el espectro de cada uno de los *frames* de la señal considerando el banco de filtros modificado por el factor de normalización. Realizar este proceso directamente resulta computacionalmente costoso, ya que se debe realizar para cada una de los α que componen el barrido. Por lo tanto, en (Molina *et al.*, 2009) se realizó una interpolación lineal de los valores filtrados considerando los valores iniciales que están almacenados en latino y las nuevas frecuencias centrales de los filtros de Mel. Si bien esto resta precisión al cálculo, se estimó que la reducción del tiempo de procesamiento era razón suficiente para justicar el uso de la interpolación.

Por último, se recalculan los coeficientes cepstrales de la señal y se almacenan estos valores en la estructura latino. Estos coeficientes cepstrales recalculados son entregados al proceso principal de VTLN, el que se describe a continuación.

3.3.2. Búsqueda del factor de normalización óptimo

En la Figura 3.5 se muestra un diagrama de bloques del proceso de búsqueda del factor de normalización óptimo.

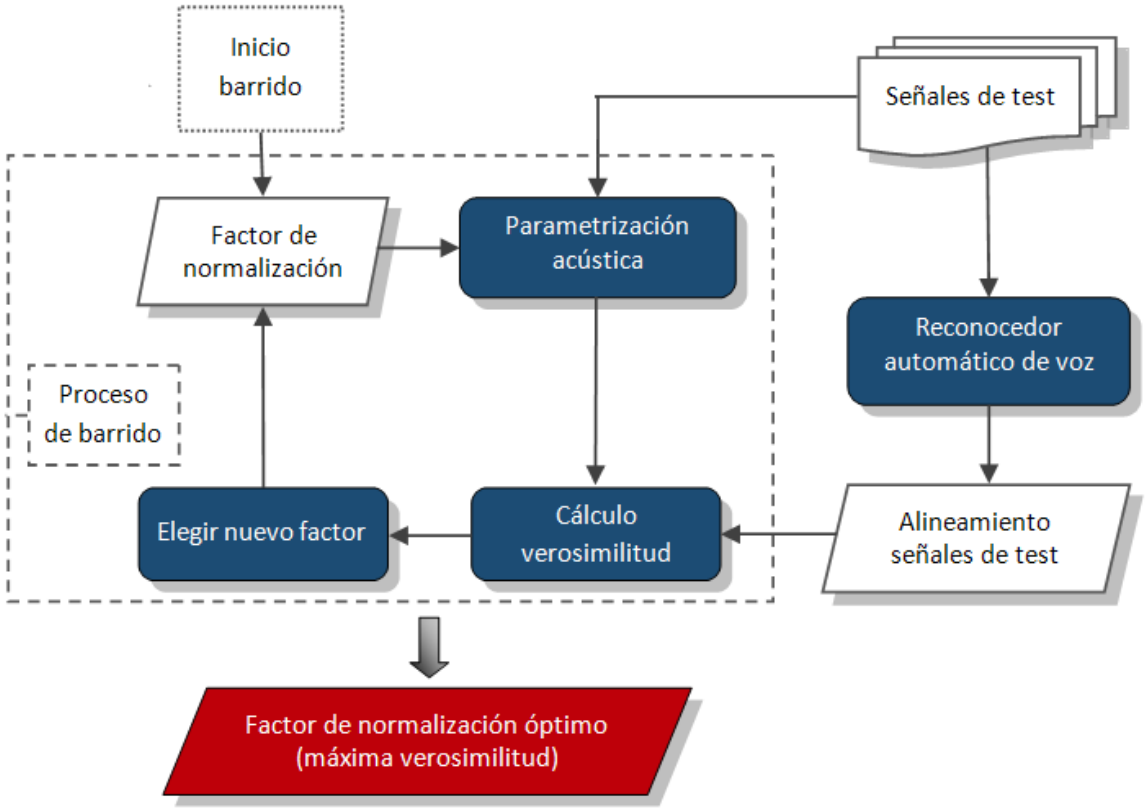


Figura 3.5: *Diagrama de bloques que describe el proceso de búsqueda del factor de normalización óptimo, el que se realiza iterativamente considerando factores de normalización equiespaciados entre 0,85 y 1,15.*

El criterio utilizado para elegir el factor de normalización óptimo es el de máxima verosimilitud (ecuación (2.38). Para calcular esta verosimilitud es necesario conocer el alineamiento obtenido con el ASR sin VTLN, por lo que es necesario realizar el reconocimiento una vez antes de comenzar la búsqueda. También se deben conocer los parámetros del modelo acústico, pero éstos están definidos globalmente en el reconocedor y es posible acceder a

ellos en la rutina.

El proceso de búsqueda se realiza de forma iterativa. Se consideran 30 factores de normalización equiespaciados entre 0,85 y 1,15. En cada iteración ocurre lo siguiente: se elige un α; se realiza el *warping* del banco de filtros, explicado en la sección anterior; luego se obtienen los nuevos MFCC; y finalmente se calcula la verosimilitud asociada a este factor usando la ecuación (2.38). Este proceso se repite hasta que se hayan obtenido las verosimilitudes asociadas a cada uno de los 30 factores. El α óptimo corresponde entonces a aquel que produce una mayor verosimilitud.

3.3.3. Reconocimiento utilizando el factor de normalización óptimo

En la Figura 3.6 se observa la última etapa de la implementación de VTLN en ASR, que corresponde al reconocimiento final considerando el factor de normalización óptimo. El resultado final de esta etapa corresponde a las transcripciones reconocidas con VTLN.

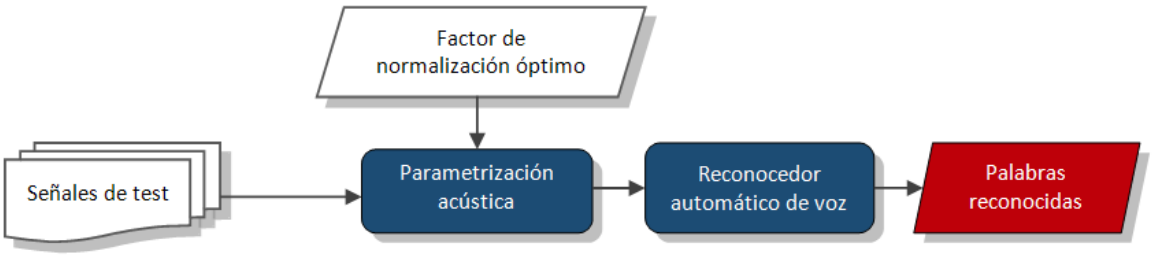


Figura 3.6: *Diagrama de bloques que describe el proceso de reconocimiento final aplicando VTLN.*

3.4. Condiciones de evaluación

Las pruebas de rendimiento de las técnicas de robustez descritas en este capítulo fueron realizadas utilizando un reconocedor automático de voz implementado en el LPTV. Este ASR tiene una arquitectura de cliente-servidor. Esta arquitectura permite que el proceso de reconocimiento (y evaluación de pronunciación también como se verá más adelante) pueda realizarse de forma remota y, por ende, permite un despliegue masivo del sistema a través de Internet. Por comodidad, los experimentos descritos en esta memoria se realizaron de manera local, con el cliente y el servidor funcionando en el mismo computador.

El reconocedor considera un modelo acústico entrenado utilizando HTK y un modelo de lenguaje de trifonemas. Cada trifonema fue modelado con una topología de izquierda-a-derecha (*left-to-right*) de tres estados sin de salto de estados, con ocho densidades Gaussianas por estado y con matrices de covarianza diagonales. En este sistema, la frase reconocida corresponde a la primera hipótesis (la más probable) dentro de la lista de las N-mejores obtenida por la decodificación de Viterbi. Se considera un máximo de 10 hipótesis en la lista de Viterbi.

Las señales utilizadas en los experimentos se encuentran muestreadas a 8000 muestras por segundo. La división por *frames* se realiza cada 25 [ms] con 50 % de superposición. Cada *frame* es procesado por una ventana de Hamming y un banco de 14 filtros Mel con frecuencias entre 300 [kHz] y 3000 [kHz]. Se utilizan 10 coeficientes cepstrales más uno de energía, las derivadas de estos coeficientes (deltas) y las doble derivada (delta-delta), obteniéndose así 33 coeficientes.

Las pruebas de MLLR y VTLN en ASR se realizaron sobre 3 diferentes configuraciones de bases de entrenamiento y test. Se consideran tres tipos de tareas: una tarea de vocabulario pequeño grabada por vía telefónica; una tarea de vocabulario medio grabado utilizando como

transductor un teléfono; y una de vocabulario medio en un ambiente libre de ruido.

3.4.1. Experimento en ambiente limpio LATINO-40 (EL40)

Se realizaron experimentos utilizando la base de datos de LATINO-40 del LDC (*Linguistic Data Consortium*). Esta base de datos está compuesta de diálogo continuo de 40 locutores nativos latinoamericanos (20 hombres y 20 mujeres), con 125 uteraciones por locutor. Las frases fueron extraídas de periódicos en español y no están agrupadas en párrafos o historias. La base de entrenamiento contiene 4501 uteraciones provistas por 36 locutores. Se incorporan además 8 señales (no pertenecientes a la base LATINO-40 del LDC) de un locutor distinto, con el objetivo de incluir en el modelo acústico todos los trifonemas que serán utilizados durante las pruebas. Esta condición resulta necesaria para la aplicación de MLLR utilizando HTK. El vocabulario está compuesto de unas 6000 palabras.

La base de test contiene 500 uteraciones (unas 4000 palabras) provistas por 4 locutores (2 hombres y 2 mujeres) los cuales no están presentes en el conjunto de locutores de entrenamiento. Estas 500 uteraciones se dividieron en grupos de 1, 5, 25 y 125 señales por locutor para realizar la adaptación con MLLR, sin superposición de señales entre grupos. En VTLN se consideró sólo una señal para realizar la normalización.

3.4.2. Experimento de consulta de cine telefónica (ECCT)

Una base de datos en español grabada por vía telefónica fue utilizada para probar las implementaciones descritas en este capítulo. En ECCT, los usuarios respondían a un sistema de información de cine basado en ASR implementado con Galaxy II (Seneff *et al.*, 1998) en el Laboratorio de Procesamiento y Transmisión de Voz de la Universidad de Chile. El

vocabulario está compuesto por 221 palabras. La base de entrenamiento está compuesta por 12494 uteraciones grabadas aproximadamente por 140 locutores. Toda la información de entrenamiento fue utilizada para entrenar los CDHMMs (*Continuouos Density Hidden Markov Models*).

La base de test del experimento ECCT corresponde a 3261 uteraciones que en total consideran 4566 palabras. Esta base no se encuentra etiquetada por locutor, por lo que no es posible utilizar ventanas mayores que una señal para realizar la adaptación. Por lo tanto las pruebas con MLLR y VTLN consideraron sólo una ventana de una señal.

3.4.3. Experimento LATINO telefónico (ELT)

ELT corresponde a un experimento vía telefónica considerando un vocabulario medio. En primer lugar se seleccionaron 1403 transcripciones de señales presentes en la base LATINO-40, las que fueron leídas por locutores nativos del español por vía telefónica. Además, se consideró la base de entrenamiento de ECCT. Por lo tanto, la base de entrenamiento de ELT contiene: 1403 señales de LATINO-40 grabadas por vía telefónica; las señales de 12494 señales grabadas del sistema de información de cine (de ECCT); y, 57 señales adicionales grabadas por 2 locutores nativos del español por vía telefónica, las que fueron incorporadas para completar el modelo acústico con trifonemas faltantes y de esta forma permitir la utilización de MLLR. En total, la base de entrenamiento considera 13954 señales.

La base de test utilizada considera las mismas transcripciones que la base de test de EL40, las que fueron pronunciadas por 20 locutores (10 hombres y 10 mujeres, 25 señales por locutor) por vía telefónica. Dado que se tienen 25 uteraciones por locutor, las pruebas de MLLR se realizaron con 1, 5 y 25 señales (el máximo posible) por locutor. El experimento con VTLN consideró una señal para la normalización.

3.5. Resultados y discusión

Siguiendo el procedimiento descrito en este capítulo para la implementación de MLLR y VTLN en ASR, se evaluó el rendimiento del sistema para cada una de las configuraciones experimentales: EL40, ECCT y ELT. La efectividad del ASR se midió a través del WER (*Word Error Rate*) definido en la ecuación (2.18). Más especificamente, se consideró la variación del WER obtenido al aplicar una técnica de robustez con respecto al caso base con modelo acústico independiente de locutor (*baseline*). Además se presentan los resultados de tres pruebas realizadas para observar el efecto de los parámetros de MLLR presentados en este capítulo en el proceso de adaptación.

La adaptación con MLLR fue realizada con los siguientes ajustes: un árbol de regresión con 32 nodos terminales; número de ocupación de 700; y matriz de transformación por tramos (ecuación (3.1)). Ésta es la configuración *default* de MLLR utilizando HTK (Young *et al.*, 2001).

3.5.1. Evaluación del rendimiento del ASR con MLLR y VTLN

Los resultados de las pruebas del experimento EL40 se pueden ver en la Figura 3.7, donde se muestra la reducción porcentual del WER con respecto al *baseline* versus la cantidad de uteraciones utilizada para la adaptación con MLLR. Se incluye además el rendimiento del sistema obtenido con VTLN. Como se puede observar en la Figura 3.7, la adaptación con MLLR produce mejoras de hasta un 22,73 % para el caso de 25 señales. Esto supone una reducción adicional de un 6,5 % del WER del caso base, en comparación al resultado obtenido con VTLN. Sin embargo, se puede ver que al considerar una señal de adaptación VTLN es claramente superior a MLLR, donde se obtiene el mismo WER del caso base. Esto

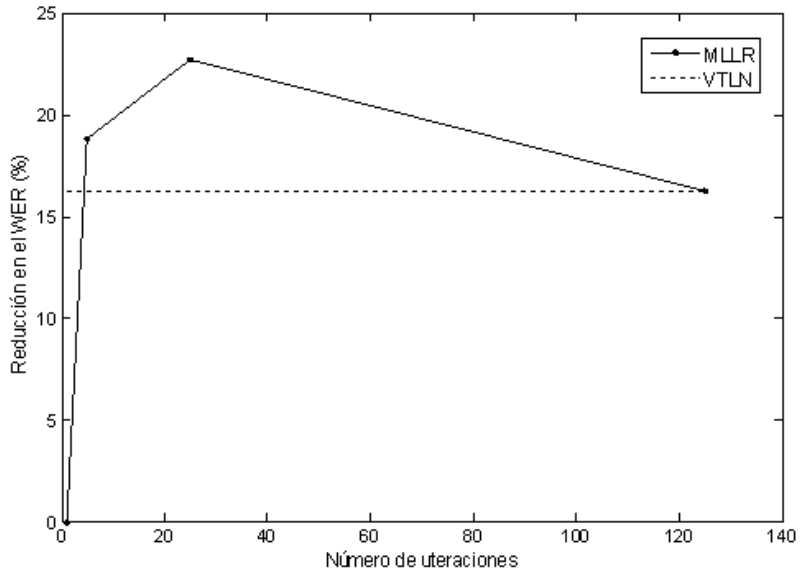


Figura 3.7: *Variación del WER en la base EL40 al usar VTLN y MLLR. VTLN considera una ventana de 1 señal. MLLR consideró ventanas de 1, 5, 25 y 125. El WER del baseline es de 3,08.*

se debe a que la información entregada por una sola uteración no es suficiente para superar el límite inferior al número de ocupación utilizado (700). Por ende, los HMMs del *baseline* no son modificados en ese caso.

Otro resultado interesante de la Figura 3.7 es la disminución del rendimiento al considerar 125 señales de adaptación en comparación con el caso de 25 señales. Este resultado sugiere que usar una mayor cantidad de uteraciones para una configuración dada no implica un aumento en el desempeño de MLLR como técnica de robustez. Este comportamiento es consistente con los observados en (Lee, 2008; Leggetter & Woodland, 1994) y está asociado a un sobre-ajuste (*overfitting*) de los modelos. Dada la gran cantidad de señales utilizadas para la adaptación, el número de matrices de transformación es alto. Por ende, el espacio acústico es modificado de forma excesiva, lo que conlleva una degradación del rendimiento.

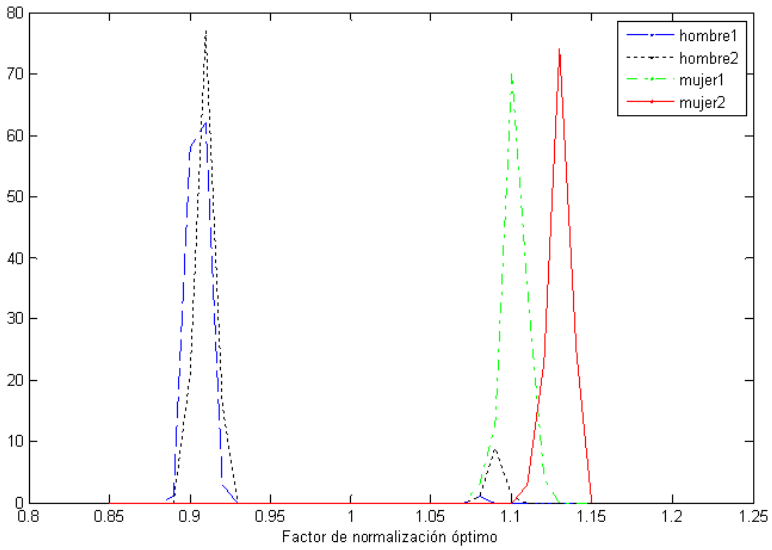


Figura 3.8: *Histograma de los factores de normalización obtenidos en la base EL40.*

En la Figura 3.8 se muestra un histograma de los factores de normalización obtenidos en EL40 para cada uno de los usuarios incluidos en la base de test. Se observa que para los dos locutores el valor de los factores de normalización se concentra en torno a 0,9, lo que corresponde a una compresión del espectro. Esto resulta consistente con los resultados presentes en (Zhan & Waibel, 1997; Pylkkonen, n.d.). La razón de este comportamiento es que el largo del tracto vocal de los hombres es mayor que el de las mujeres en promedio. Dado que el modelo acústico fue entrenado tanto con hombres como mujeres, resulta necesario realizar una compresión del espectro al utilizarse en hombres para compensar por estas diferencias de VTL. En el caso de las locutoras de la base de test se observa el comportamiento opuesto, distingüiéndose un estiramiento del espectro. Nótese que efectivamente se realiza una compensación por distancia del tracto vocal, aún cuando el factor de normalización se elige con el criterio de máxima verosimilitud y no por un método de estimación directo del VTL.

En la Tabla 3.1 se muestran los resultados obtenidos en la base ECCT con VTLN y MLLR. El rendimiento de VTLN es superior a MLLR en este caso, el que nuevamente no realiza

Configuración	**WER %**	**Reducción del WER con respecto al Baseline %**
Baseline	13,06	0
MLLR 1 señal	13,06	0
VTLN 1 señal	12,26	6,13

Tabla 3.1: *Variación del WER en la base ECCT al aplicar VTLN y MLLR.*

ningún tipo de adaptación a los modelos acústicos. Este resultado junto con el obtenido en VTLN indican un comportamiento robusto de VTLN ante la presencia de muy pocas señales de adaptación (1 en este caso), en comparación con MLLR.

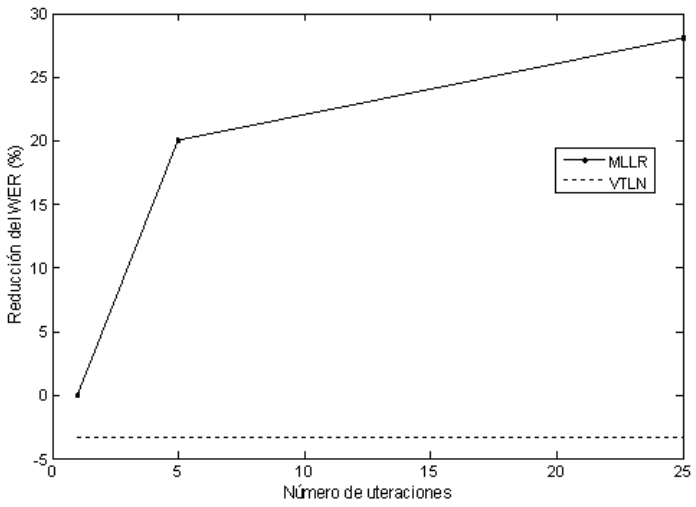


Figura 3.9: *Variación del WER en la base ELT al usar VTLN y MLLR. VTLN considera una ventana de 1 señal. MLLR consideró ventanas de 1, 5 y 25. El WER del baseline es de 14,33.*

La Figura 3.9 muestra la reducción porcentual del WER al aplicar VTLN y MLLR en el experimento ELT. En este experimento se lograron resultados similares a los obtenidos en EL40 al aplicar MLLR, obteniéndose reducciones del WER de hasta un 28 % con respecto al *baseline*. Sin embargo, con VTLN se observa una leve disminución en el desempeño, lo

que no resulta consistente con los resultados obtenidos anteriormente. Este resultado resulta sorprendente de cierta forma, debido a que las diferencias (*mismatch*) entre las condiciones de entrenamiento y prueba es mayor en esta base que en EL40. Por ende, era razonable el esperar un mejor desempeño en VTLN, similar al obtenido con MLLR en esta base.

3.5.2. Pruebas de configuración de MLLR en ELT

Como se ha visto en este capítulo, MLLR es una técnica que posee muchos parámetros ajustables. Por ende, resulta interesante analizar el efecto que tiene sobre el rendimiento del sistema el modificar estas variables. Este motivo llevó a la realización de tres pruebas para estudiar el impacto que tiene el cambiar los siguientes parámetros: cantidad de nodos terminales del árbol de regresión; tipo de matriz de transformación; y límite inferior al número de ocupación. Las pruebas se realizaron considerando la configuración *default* de MLLR utilizada para las pruebas de rendimiento de EL40, ELT y ECCT, y variando solamente el parámetro que se desea estudiar.

En la Figura 3.10 se muestra el efecto que tiene el variar del límite inferior del número de ocupación en el rendimiento del sistema, considerando ventanas de 1, 5 y 25 señales. Es posible observar que la variación de este parámetro produce un efecto similar en los casos de 1 y 5 señales: un punto de rendimiento máximo, con degradaciones al reducir o aumentar el valor del número de ocupación. Este comportamiento es válido también para la ventana de 125 uteraciones, donde pruebas adicionales revelaron una disminución del rendimiento del sistema en comparación con el mejor caso, la que comienza a observarse al considerar un número de ocupación de 1500 (donde la reducción del WER es un 2,5 % menor que al considerar un valor de 1000 del número de ocupación). Recordar que el número de ocupación está ligado directamente a la cantidad de matrices de transformación que se calculan. De esta forma, es posible observar que, para una configuración dada, existe una cantidad de matrices

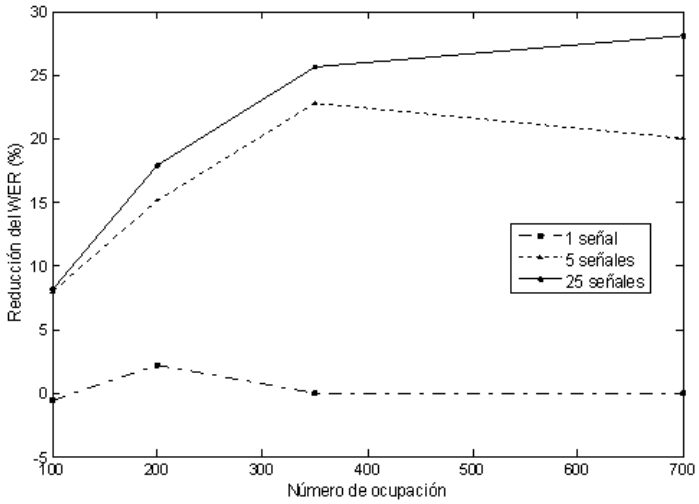


Figura 3.10: *Variación del WER en la base ELT al cambiar el número de ocupación y el tamaño de la ventana de MLLR. El WER del baseline es de 14,33.*

de adaptación calculadas óptima, con la cual se obtiene un máximo de rendimiento.

Resulta interesante el hecho de que el fijar el límite inferior al número de ocupación en 700 produjo las mayores reducciones del WER observadas (28,05 % con la ventana de 25 señales). Más aún, el resultado obtenido con la ventana de 5 señales resulta similar al máximo obtenido (una diferencia de sólo un 2,8 %). Estos resultados indican que el valor por *default* en HTK de este parámetro está bien estimado y corresponde a un buen punto de partida, sobre el cual se pueden realizar mayores ajustes en caso de ser necesarios.

En la Tabla 3.2 se muestra el WER y la reducción del WER porcentual del sistema obtenidas con MLLR al variar el tipo de matriz de transformación utilizada. Al observar esta tabla resulta evidente que el mejor caso corresponde al de matriz por tramos, donde se observa un aumento de casi un 10 % de reducción del WER en comparación con el caso base. Nótese que este caso es menos costoso en procesamiento que el caso de matriz completa, ya que se calcula una cantidad de coeficientes menor. Este resultado sugiere que el considerar

Tipo de Matriz MLLR	**WER** %	**Reducción del WER con respecto al Baseline** %
Matriz diagonal	12,33	13,96
Matriz por tramos	10,31	28,05
Matriz completa	11,63	18,84

Tabla 3.2: *Variación del WER en la base ELT al cambiar el tipo de matriz de transformación de MLLR. El WER del baseline es de 14,33. La ventana es de 25 señales.*

independencia entre los coeficientes cepstrales (y la energía), los delta y los delta-delta es una buena suposición.

Número de Nodos Terminales Árbol de Regresión	**WER** %	**Reducción del WER con respecto al Baseline** %
16 nodos	9,95	30,56
32 nodos	10,31	28,05
64 nodos	10,31	28,05

Tabla 3.3: *Variación del WER en la base ELT al cambiar la cantidad de nodos terminales del árbol de regresión de clases. El WER del baseline es de 14,33. La ventana es de 25 señales.*

Finalmente, en la Tabla 3.3 se presentan los resultados de las pruebas realizadas modificando el número de nodos terminales del árbol de regresión. Se observa que no existe una gran diferencia en el rendimiento, observándose un ligero aumento al considerar un número menor de nodos terminales. Notar que el aumentar la cantidad de nodos terminales no produce ningún cambio en el desempeño del sistema. Esto se explica por el método de generación del árbol de regresión usado por HTK, en el cual agregar una mayor cantidad de nodos simplemente realiza divisiones de nuevas clases a partir de los nodos terminales del árbol inicial. Dado que la información de adaptación y el límite del número de ocupación no

cambian, se mantiene la cantidad de transformadas y las componentes Gaussianas que serán afectadas por cada matriz de transformación.

3.6. Conclusiones

En este capítulo se presentaron dos técnicas de robustez frente a cambios de locutor aplicadas a ASR: *Maximum Likelihood Linear Regression* (MLLR) y *Vocal Tract Length Normalization*. MLLR modifica el conjunto de HMMs del reconocedor mediante una transformación lineal de las medias de las componentes Gaussianas, las que se encuentran agrupadas en una estructura de árbol. VTLN por otra parte realiza una normalización del banco de filtros de Mel utilizado para el procesamiento de las señales. Este método busca reducir la degradación del rendimiento del sistema debido a diferencias en la longitud del tracto vocal de los locutores, que corresponde a una de las principales causas de variabilidad inter-locutor. Ambos métodos se implementan de forma no supervisada, es decir, utilizando las transcripciones obtenidas directamente del reconocedor.

Los resultados experimentales obtenidos indican que ambas técnicas producen mejoras significativas al rendimiento del sistema en la mayoría de los casos. Se lograron reducciones del WER de hasta un $30,56\,\%$ al utilizar MLLR con una ventana de 25 señales y de hasta un $16,23\,\%$ con VTLN de una señal. En general MLLR presentó un mejor desempeño que VTLN en las pruebas realizadas. Sin embargo, al considerar una cantidad de información de adaptación muy limitada (1 señal) VTLN genera una disminución considerable del WER del sistema, a diferencia de MLLR el cual sólo fue capaz de lograr una reducción máxima de $2,23\,\%$. Este comportamiento es consistente al observado en (Panchapagesan & Alwan, 2008), en el que para el caso de 1 señal MLLR presenta un rendimiento muy inferior a VTLN, obteniéndose incluso un desempeño peor que el *baseline*. En (Wang *et al.*, 2007) se

obtuvieron resultados similares, en los que MLLR no produce mejoras para ventanas muy pequeñas.

Las pruebas realizadas para analizar el efecto de cada uno de los parámetros ajustables de MLLR produjeron resultados similares a los observados en (Young *et al.*, 2001). El principal factor que afecta el desempeño de la técnica es la cantidad de matrices de transformación calculadas con una cierta cantidad de señales, la que se puede modificar de manera indirecta en HTK mediante el límite del número de ocupación. Variaciones de este parámetro produjeron diferencias de hasta casi un 20 % en la reducción del WER con respecto al sistema base. El tipo de matriz produjo también un impacto importante, observándose una degradación de mas de 9 % al no utilizar una matriz por tramos. Los resultados obtenidos indican la necesidad de realizar una etapa de ajuste con los parámetros de MLLR para obtener un desempeño óptimo, lo que ya ha sido planteado en (Young *et al.*, 2001) y (Leggetter & Woodland, 1994).

Capítulo 4

Implementación de técnicas de robustez en CAPT

4.1. Introducción

En este capítulo se detallan los aspectos principales de la implementación de MLLR y VTLN en un sistema de evaluación automático de pronunciación basado en ASR (*Automatic Speech Recognition*). La aplicación de estas técnicas de robustez comparte varios aspectos en común con la descrita en el capítulo 3, por lo que sólo se explica en detalle las etapas nuevas de la implementación en el CAPT (*Computer Aided Pronunciation Training*). Estas técnicas son evaluadas considerando tres bases de datos compuestas por los siguientes tipos de locutores: expertos del idioma inglés (lingüistas); adultos no expertos del inglés; y niños no hablantes nativos del idioma inglés.

El CAPT descrito en este capítulo está implementado en una arquitectura cliente-servidor,

lo que permite el acceso remoto a esta herramienta. Debido a esto, no se posee información previa de los usuarios del sistema que pueda ser utilizada para aplicar las técnicas de robustez. Esto restringe la cantidad de señales de adaptación disponibles a unas pocas uteraciones (una de ser posible). Por este motivo la evaluación del rendimiento de estas técnicas presentada en este capítulo se realiza considerando una cantidad muy limitada de información (1-5 uteraciones).

Resulta importante destacar que tanto MLLR como VTLN utilizan el criterio de máxima verosimilitud en el proceso de adaptación, el cual se ha comprobado que produce un aumento de la tasa de reconocimiento y, por ende, del desempeño de un ASR. En cambio, en un sistema de evaluación de pronunciación el desempeño se mide normalmente por la correlación entre los scores objetivos y subjetivos para una determinada base de test. Esta métrica de rendimiento no está relacionada directamente a la tasa de reconocimiento. Por ende, se desconoce *a priori* el efecto que tiene sobre el rendimiento del CAPT. Sin embargo, dado que el CAPT utiliza las estadísticas (lista de las N-mejores hipótesis) obtenidas con el ASR, se espera que una mejora en el reconocimiento utilizando VTLN y MLLR produzca también una mejora en la correlación.

4.2. Implementación de MLLR en CAPT

El proceso de entrenamiento y prueba del sistema de evaluación de pronunciación considera tres etapas de importancia para este trabajo: entrenamiento del modelo acústico; entrenamiento de las curvas *a priori* (o f.d.p. *a priori*) de las medidas de confiabilidad; y generación de scores subjetivos de la base de test.

El entrenamiento del modelo acústico y la etapa de generación del árbol de regresión se

realiza de manera similar a la descrita en el capítulo 3. Al entrenar el conjunto de HMMs utilizando HTK se obtiene el archivo de estadísticas con los números de ocupación de los modelos y luego se determina el árbol de regresión. Este árbol de regresión es utilizado en todos los cálculos de matrices de transformación que se realizan posteriormente.

A continuación se describen con un mayor detalle las otras dos etapas del proceso que no forman parte de la implementación de MLLR en ASR explicada anteriormente en este trabajo.

4.2.1. Entrenamiento de curvas a priori

El entrenamiento de las curvas *a priori* (ecuación (2.41)) sigue el esquema que se muestra en la Figura 4.1. En esta etapa se aplica MLLR a la base de entrenamiento utilizada para entrenar las curvas *a priori*. El proceso de cálculo de las transformadas se realiza de forma análoga al caso descrito anteriormente con ASR. HEA_{DAPT} es utilizado múltiples veces para calcular los HMMs adaptados a partir de la base de datos, considerando los parámetros de configuración. Este programa recibe como entradas: las transcripciones reconocidas por el ASR utilizando el HMM *baseline*, las cuales son convertidas a formato HTK con un programa similar al descrito en el capítulo 3 ; el modelo acústico; y el árbol de regresión obtenido a partir de los datos de entrenamiento del HMM.

Los HMMs adaptados obtenidos son utilizados para realizar nuevamente el reconocimiento de las uteraciones de la base de entrenamiento de las curvas *a priori*. Este procedimiento es similar al realizado en la etapa de reconocimiento final de la implementación de VTLN en ASR. Primero se generan los archivos .bat con las instrucciones necesarias para manejar los modelos y realizar el reconocimiento para cada una de las ventanas de señales consideradas en la adaptación. El formato de los resultados de salida del reconocedor utilizado

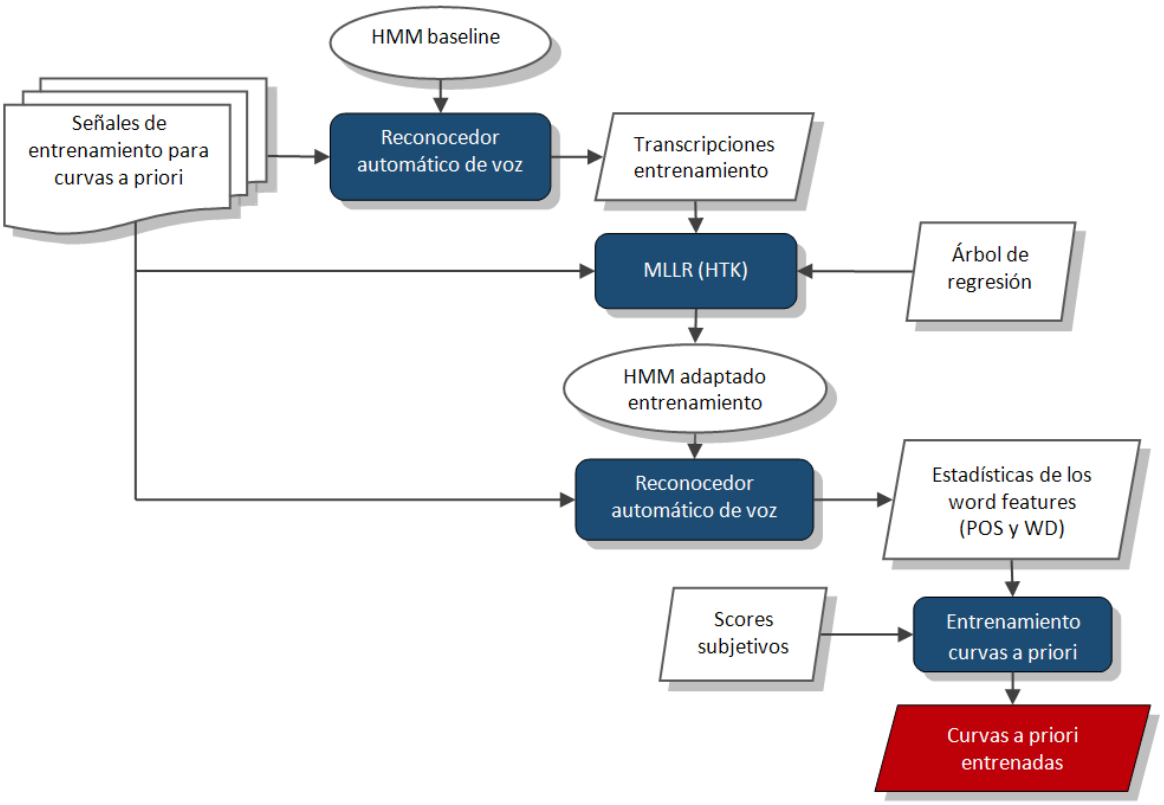


Figura 4.1: *Diagrama de bloques que describe el proceso de entrenamiento de las curvas de Bayes al aplicar MLLR.*

para evaluación de pronunciación es diferente al de ASR. Además de contener las palabras reconocidas para cada una de las señales, este archivo almacena el valor de cada una de las métricas de confiabilidad (POS y WD) y el score asociado a cada clasificador. Esta nota objetiva es obtenida utilizando las curvas entrenadas, las que son cargadas por el servidor al momento de iniciar su ejecución. En esta parte del entrenamiento sólo se utiliza los valores de WD y POS de cada una de las señales, los que son utilizados para entrenar las curvas *a priori*.

4.2.2. Generación de scores objetivos de la base de test

En la Figura 4.2 se muestra un diagrama de bloques de la etapa de generación de scores subjetivos de la base de test al utilizar MLLR. Notar que el árbol de regresión de clases utilizado corresponde al obtenido a partir del modelo acústico, el que ya fue usado en el entrenamiento de las curvas *a priori*.

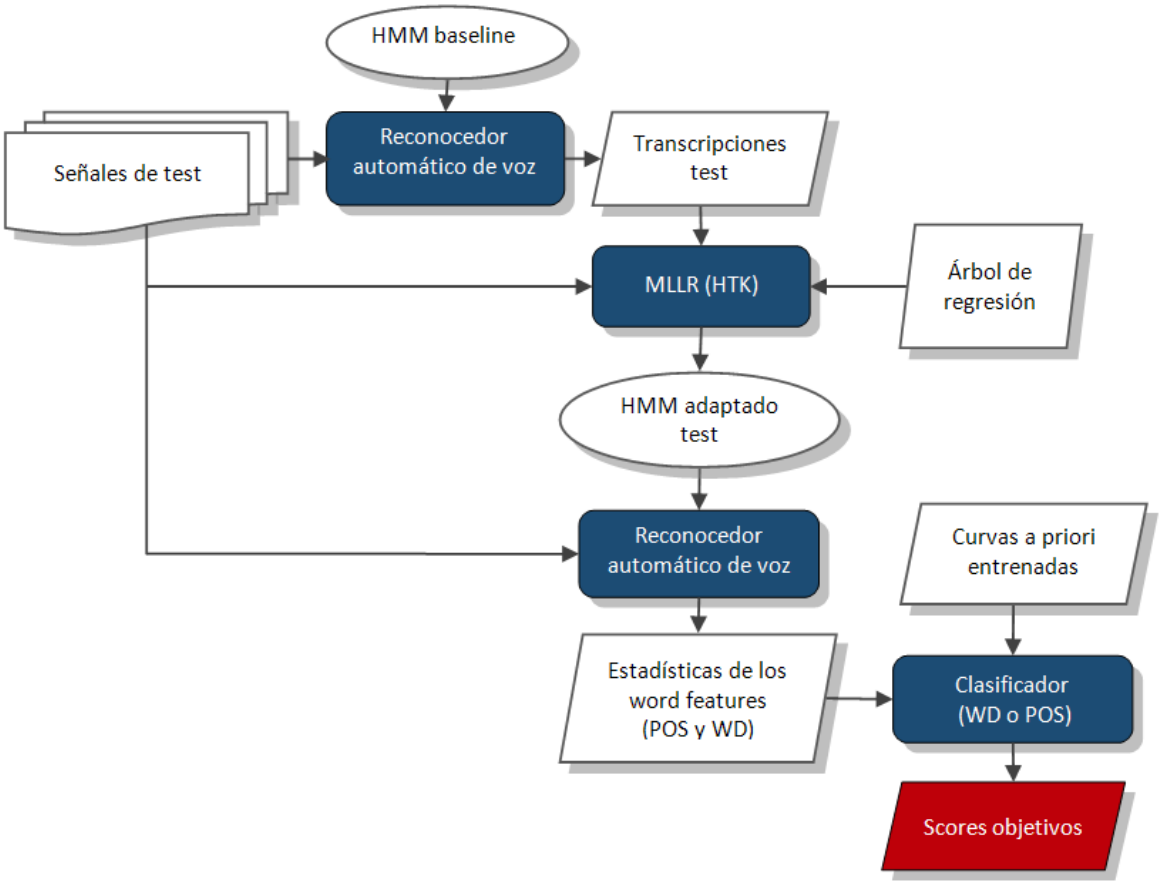


Figura 4.2: *Diagrama de bloques que describe el proceso de test al aplicar MLLR.*

En primer lugar, la base de test es utilizada para realizar el reconocimiento con los modelos originales. Con la transcripción resultante se obtienen los HMMs adaptados, los que son usados para un nuevo reconocimiento de las señales. Finalmente, se obtiene como sali-

da del sistema de evaluación de pronunciación la nota objetiva asociada a cada uno de los clasificadores (WD o POS).

MLLR considera distintos parámetros ajustables que afectan el proceso de adaptación. Al realizar la implementación de MLLR en este sistema, se consideró la misma configuración en las dos etapas descritas anteriormente. Las variables que se mantuvieron iguales son las siguientes: árbol de regresión de clases; número de ocupación; tamaño de la ventana de adaptación (cantidad de uteraciones por locutor); y tipo de matriz de transformación. El valor de estos parámetros utilizado fue el recomendado por HTK[1], con excepción del tamaño de la ventana el cual se varió al realizar las pruebas.

4.3. Implementación de VTLN en CAPT

La implementación de VTLN en el sistema de evaluación de pronunciación considera dos etapas distintivas (al igual que en MLLR): entrenamiento de las curvas *a priori* y generación de scores objetivos de la base de test. El proceso de entrenamiento del modelo acústico no se ve afectado por VTLN.

4.3.1. Entrenamiento de curvas de Bayes

El entrenamiento de las curvas *a priori* sigue el esquema que se muestra en la Figura 4.3. Este comienza con el reconocimiento de las señales de la base de entrenamiento de las curvas *a priori*. Utilizando esta transcripción y las uteraciones, es posible realizar la

[1]Configuración *default* de HTK: 32 nodos terminales del árbol de regresión; limíte de número de ocupación igual a 700; y matriz de transformación por tramos.

búsqueda iterativa del factor de normalización óptimo. Éste es elegido usando el criterio de máxima verosimilitud.

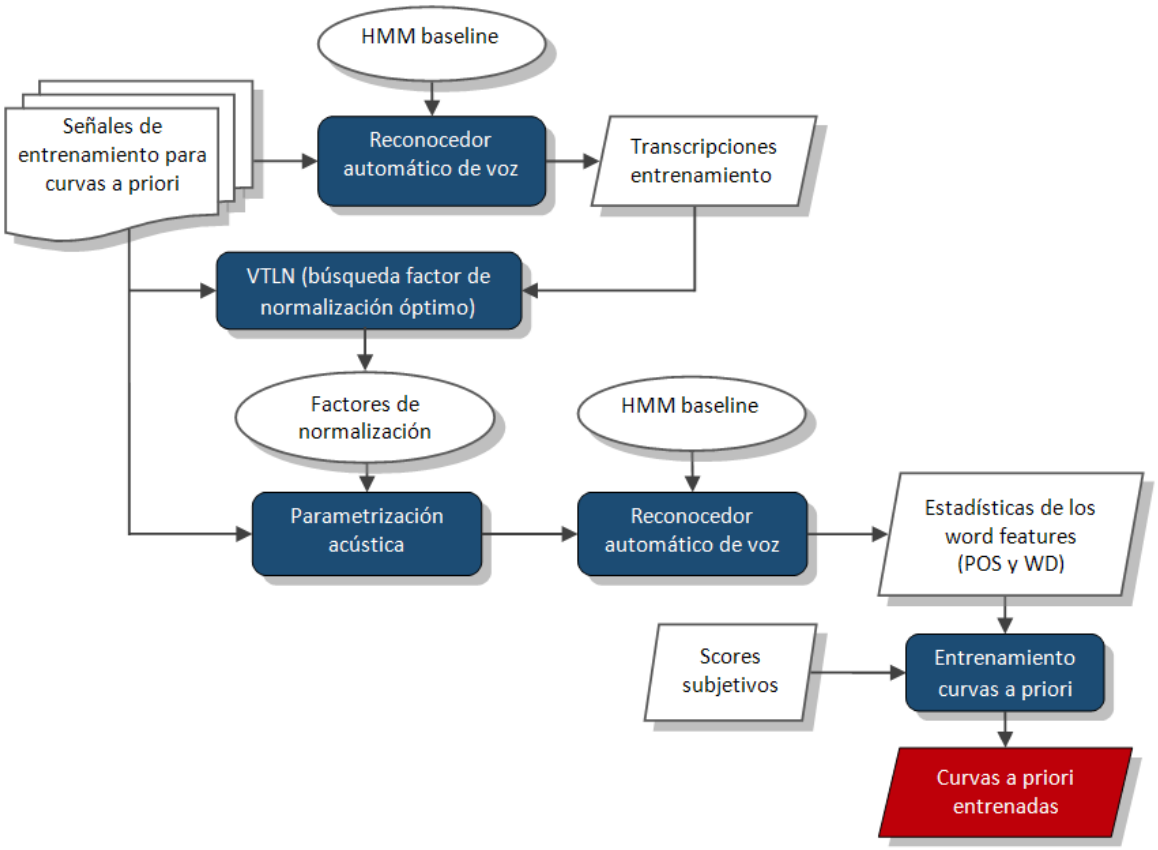


Figura 4.3: *Diagrama de bloques que describe el proceso de entrenamiento de las curvas a priori al aplicar VTLN.*

El α óptimo define el nuevo banco de filtros de Mel con el que se debe realizar la parametrización acústica de las señales para una nueva etapa de reconocimiento, siguiendo el procedimiento explicado en la sección 3.3.1. El resultado de este proceso se obtienen los valores de POS y WD para cada una de las señales. Con estos datos se entrenan las curvas de Bayes, necesarias para la próxima etapa de la implementación.

4.3.2. Generación de scores objetivos de la base de test

En la Figura 4.4 se muestra un diagrama de bloques de etapa de generación de scores objetivos de la base de test al utilizar VTLN. Esta técnica se aplica considerando las mismas condiciones utilizadas al entrenar las curvas *a priori*.

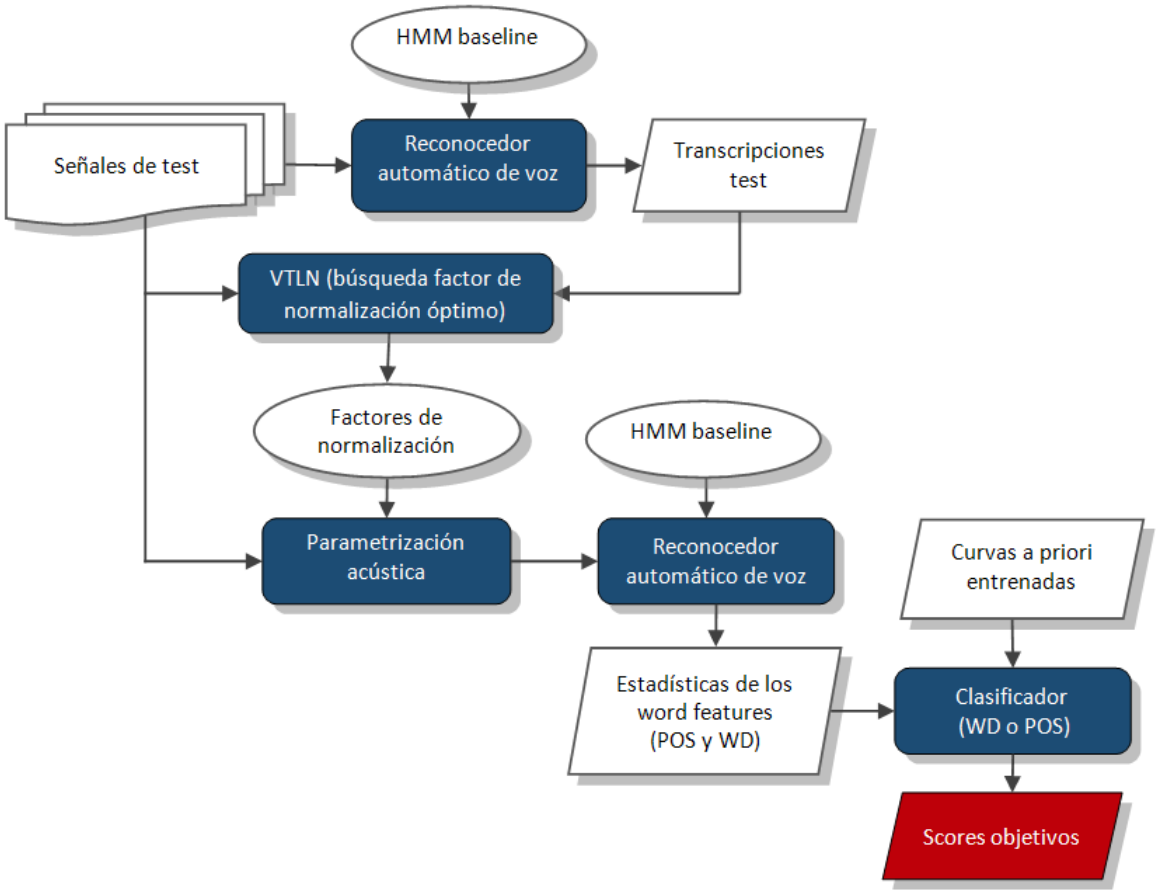


Figura 4.4: *Diagrama de bloques que describe el proceso de test al aplicar VTLN.*

4.4. Condiciones de evaluación

El reconocedor del sistema de evaluación de pronunciación utilizado en las pruebas de este capítulo considera un modelo acústico entrenado utilizando HTK y considerando como unidad básica el trifonema. Cada trifonema fue modelado con una topología de izquierda-a-derecha (*left-to-right*) de tres estados sin transiciones de salto de estados, con ocho densidades Gaussianas por estado y con matrices de covarianza diagonales. En el sistema, la frase reconocida corresponde a la primera hipótesis (la más probable) dentro de la lista de las N-mejores obtenida por la decodificación de Viterbi. Se consideró un máximo de 10 hipótesis en la lista de Viterbi.

Los modelos acústicos del inglés fueron entrenados con el Corpus CSR-I WSJ0 (Garofalo *et al.*, 1993). Las señales de voz de esta base de datos fueron grabadas con micrófonos de alta calidad y con una tasa de muestreo de 16 [kHz]. Durante la parametrización de las señales se utilizó un banco de Mel de 18 filtros. Las 20055 uteraciones que la componen fueron utilizadas para el entrenamiento de los modelos acústicos. Para el entrenamiento de los modelos acústicos del español se utilizó la base de entrenamiento de EL40 (4994) uteraciones. Finalmente, se agregaron 21 señales grabadas por lingüistas expertos del idioma inglés para agregar trifonemas a la base de datos que resultaban necesarios para realizar las pruebas con MLLR. En total la base contiene 25071 señales.

El modelo de lenguaje del ASR está compuesto por las palabras objetivo y las palabras competidoras, las cuales fueron obtenidas del Corpus CSR-I WSJ0 (Garofalo *et al.*, 1993) utilizando un método basado en la distancia K-L entre palabras (Molina *et al.*, 2008). Cada palabra objetivo tiene asociadas diez palabras competidoras. Se incluye además una variante de la palabra objetivo con pronunciación en español, la cual utiliza modelos acústicos obtenidos a partir de señales grabadas por locutores nativos del español. Notar que las palabras

competidoras no están relacionadas directamente con la palabra objetivo excepto por la distancia K-L utilizada para su elección. A modo de ejemplo, las palabras competidoras de la palabra *boyfriend* del modelo CPC son las siguientes: *boyfriend2* (versión en español) *; clarified; padded; buttons; fivefold; rebuilding; vanished; highs; riding; y variations*. Excepto por la versión en español de la palabra objetivo (en algunos de los casos), las palabras competidoras no se asemejan fonéticamente a la palabra *boyfriend*. Dado que la implementación de MLLR y VTLN es no supervisada, las transcripciones reconocidas pueden contener estas palabras, lo que puede tener un efecto en el desempeño de estas técnicas. Por tal motivo, las pruebas se realizaron considerando dos configuraciones de palabras competidoras: 10 palabras competidoras y la versión en español (CPC); y sólo la versión en español (SPC).

Para obtener una nota objetiva de la evaluación de pronunciación a partir del ASR se consideraron dos métricas de confiabilidad: WD, definida en la ecuación (2.42); y POS, como se muestra en la ecuación (2.43). Como medida de evaluación de rendimiento del CAPT se consideró la correlación entre: las notas objetivas, entregadas por el reconocedor (WD o POS); y las notas subjetivas, calificadas por evaluadores expertos del idioma inglés.

A continuación se describen las tres bases de test utilizadas para evaluar el rendimiento del sistema con las dos técnicas de robustez.

4.4.1. Base de test lingüistas (BTL)

Esta base de test está compuesta por uteraciones de 20 palabras objetivo: *Against; Behave; Boyfriend; Chocolate; College; Doesn't; Example; Handsome; Hospital; Mouth; Scientist; Should; Special; Student; Television; Thirty two; Tourism; Vegetable; Vibration; y Yesterday*. Estas palabras fueron escogidas por expertos en fonética del idioma inglés con el fin de conseguir un conjunto de test fonéticamente balanceado. A continuación, se eligieron 4

categorías distintas de errores de pronunciación, considerando dos o tres ejemplos de pronunciaciones erróneas por categoría. Estos errores van desde errores menores a errores más significativos que corresponden a pronunciaciones de la palabra objetivo utilizando las reglas fonéticas del idioma español. Como resultado se obtuvieron 5 categorías de pronunciación, donde 5 equivale a la pronunciación correcta de la palabra objetivo y 1 equivale a la peor pronunciación posible (pronunciación usando reglas fonéticas del idioma español). Estas transcripciones fueron grabadas por 9 expertos del idioma inglés utilizando dos micrófonos de bajo costo. Se usó una tasa de muestreo de 16 [kHz]. En total se grabaron 3811 uteraciones, las cuales fueron evaluadas por los mismos expertos del idioma inglés aplicando la escala de evaluación mencionada antes.

La base de datos mencionada anteriormente se dividió en: un conjunto de evaluación formado por 2800 uteraciones (7 locutores, 400 palabras por locutor); y conjunto de test formado por 800 uteraciones (2 locutores, 400 palabras por locutor). Nótese que la suma de ambas bases es menor que el total de elocuciones grabadas (3811). Esto se debe a que se eliminaron aleatoriamente señales para conseguir un conjunto balanceado por locutor para facilitar la elección de ventanas al realizar la adaptación. La base de evaluación fue utilizada para el entrenamiento de las f.d.p *a priori* de acuerdo a la ecuación (2.41).

También se consideró una evaluación con esta base de datos considerando sólo 2 niveles: aceptable (5) o inaceptable (1). En este caso los conjuntos de evaluación y test se redujeron a 1458 y 420 uteraciones respectivamente. Nuevamente, el conjunto de evaluación fue utilizado para el entrenamiento de las f.d.p, para el caso de 2 niveles de evaluación.

4.4.2. Base de test de alumnos (BTA)

En BTA se grabaron uteraciones de ocho locutores adultos no expertos del inglés, los que pronunciaron las mismas palabras de BTL. Estas señales fueron evaluadas por los mismos expertos del inglés de BTL como: aceptable (5) o inaceptable (1). En total se grabaron 100 señales, utilizando micrófonos de bajo costo y en un ambiente de ruido poco controlado. Estas señales no están etiquetadas por locutor, por lo que no es posible realizar la adaptación considerando más de una señal.

4.4.3. Base de test de niños (BTN)

En esta base de datos se consideraron elocuciones provistas por un grupo de cinco estudiantes no hablantes nativos del inglés, con edades de 11 o 12 años de edad. Cada uno grabó 3 o 4 veces cada una de las siguientes palabras: *Gently, Counter, Healthy and Race*. Estas señales fueron evaluadas por los mismos expertos del inglés de BTL considerando los 2 niveles de evaluación mencionados anteriormente. La base completa considera 72 uteraciones, las cuales fueron grabadas considerando las mismas condiciones de BTA. Al igual que en BTA, se desconoce cuales señales pertenecen a cada uno de los locutores. Este hecho impide realizar pruebas con esta base de datos con ventanas mayores que una señal.

4.5. Resultados y discusión

En esta sección se muestran los resultados de las pruebas de rendimiento del sistema de evaluación de pronunciación utilizando la implementación de MLLR y VTLN presentada en este capítulo. El desempeño se midió utilizando la correlación entre: los scores objetivos,

obtenidos del reconocedor; y los scores subjetivos, correspondientes a calificaciones de las señales realizadas por evaluadores expertos del idioma inglés. La correlación se calcula de la siguiente forma:

$$\rho_{X,Y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2\right]^{1/2}} \tag{4.1}$$

donde X corresponde a los scores objetivos de las uteraciones, Y a los scores subjetivos y $\bar{x}$ y $\bar{y}$ corresponden a los valores promedio respectivos de estas series.

La adaptación con MLLR fue realizada con los siguientes ajustes: un árbol de regresión con 32 nodos terminales; número de ocupación de 700; y matriz de transformación por tramos (ecuación (3.1)). Esta es la configuración *default* de MLLR utilizando HTK (Young *et al.*, 2001).

En la Tabla 4.1 se muestra el rendimiento del sistema obtenido para la base BTL, considerando 5 niveles de evaluación. VTLN muestra un comportamiento contradictorio, mejorando la correlación del clasificador POS para ambos tipos de modelos de palabras competidoras (CPC y SPC), pero produciendo una degradación del rendimiento con WD. Se puede observar también que todas las configuraciones de MLLR producen una disminución de la correlación entre los scores subjetivos y objetivos para ambos clasificadores, la que puede llegar a un 23,3 % de diferencia con respecto al *baseline*. Notar que la menor degradación ocurre al considerar la ventana más pequeña entre las que se probaron con MLLR, observándose un peor desempeño al aumentar la cantidad de uteraciones utilizada. Este resultado sugiere que el utilizar una mayor cantidad de información para la adaptación dificulta la tarea de calificación objetiva del sistema.

Configuración	Correlación promedio de los scores subjetivos - objetivos obtenida con el clasificador WD	Correlación promedio de los scores subjetivos - objetivos obtenida con el clasificador POS
Baseline	$0,584 \quad (0,591)$	$0,608 \quad (0,595)$
VTLN 1 señal	$0,561 \quad (0,583)$	$0,614 \quad (0,603)$
MLLR 5 señales	$0,521 \quad (0,519)$	$0,544 \quad (0,527)$
MLLR 25 señales	$0,448 \quad (0,477)$	$0,503 \quad (0,493)$
MLLR 100 señales	$0,494 \quad (0,467)$	$0,489 \quad (0,477)$

Tabla 4.1: *Correlación promedio de los scores subjetivos - objetivos obtenida con los clasificadores WD y POS en BTL considerando 5 scores de evaluación para las siguientes configuraciones: Baseline (sin técnicas de robustez); VTLN con una ventana de 1 señal; MLLR con ventana de 5, 25 y 100 señales. Los valores sin paréntesis corresponden a los obtenidos con el modelo de palabras competidoras CPC. Con paréntesis se muestran los de SPC.*

En la Figura 4.5 se muestran las reducciones del WER obtenidas para las configuraciones anteriores versus el score subjetivo de cada señal. Es posible observar que MLLR produce una reducción de hasta un 7,92 % al considerar todos los scores. Notar que para los tres tamaños de ventanas MLLR presenta una disminución en el WER de las señales con score 5 (hasta un máximo de 15,8 % con ventana de 5 señales), pero se obtiene un mejor desempeño del ASR en todos los otros niveles de evaluación. VTLN por otra parte presenta un leve aumento del WER, el que resulta significativo para el caso de las señales con score 5 (5 % de reducción del WER).

Un mejor reconocimiento implica que la palabra objetivo se encuentra en primer lugar de la lista de las N-mejores de Viterbi para una mayor cantidad de señales. Esto puede afectar de forma negativa a los clasificadores, lo que queda ejemplificado al analizar el caso de POS (ecuación (2.43)). En la Figura 4.6 se observa el valor de POS en la base de evaluación

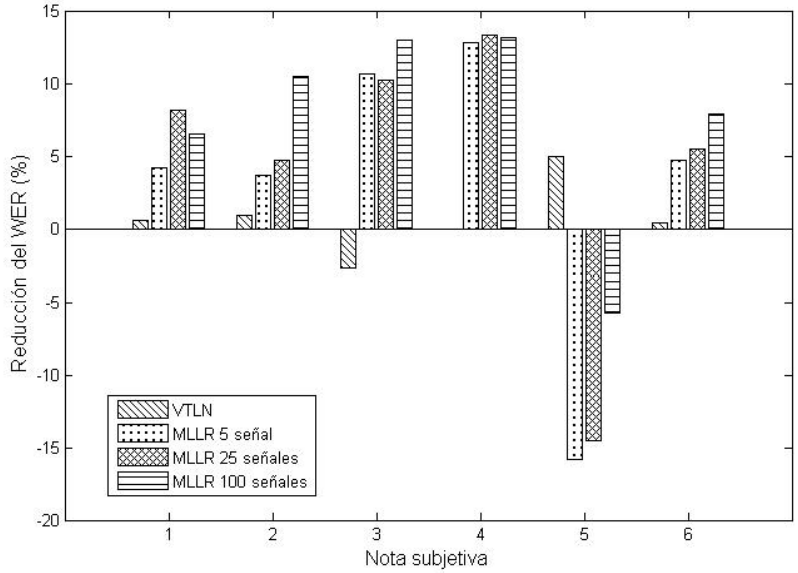
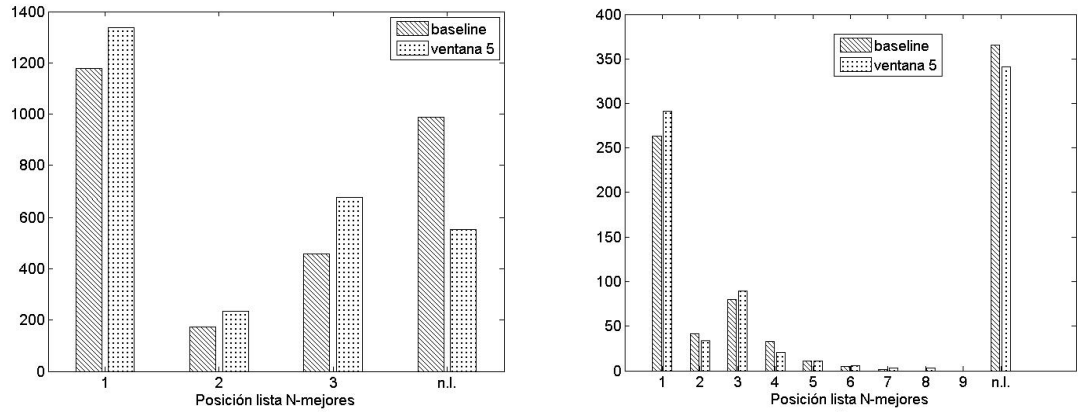


Figura 4.5: *Reducción del WER obtenidas para cada uno de los 5 scores en la base BTL con las siguientes técnicas: VTLN con una señal; MLLR con 5, 25 y 100 señales. Modelo de palabras competidoras es CPC. El WER del baseline es de 25; 78,18; 78,57; 84,62; y 92,78, para el score 5; 4; 3; 2; y 1 respectivamente. El WER del baseline considerando todos los scores es de 66,63.*

utilizada para entrenar las curvas y en la base de test de BTLN para las configuraciones baseline y MLLR utilizando 5 señales de adaptación.

La métrica de confiabilidad POS es utilizada para realizar una calificación objetiva basada en la posición de la palabra objetivo dentro de la lista de las N-mejores entregada por el reconocedor. De esta forma, la aparición de la palabra objetivo en las primeras posiciones de esta lista implicaría que se está en presencia de una palabra correctamente pronunciada, la que se evaluaría con nota alta (4 o 5). Si por otra parte la palabra objetivo se encuentra al final o fuera de esta lista, el clasificador consideraría la pronunciación como deficiente y la evaluaría con nota 1 o 2. Como se puede ver en la Figura 4.6, la palabra objetivo aparece con mayor frecuencia en las primeras posiciones de la lista de N-mejores de Viterbi. Esto dificulta el proceso de discriminación entre clases (scores) realizado con POS para la evaluación



(a) **(b)**

Figura 4.6: *Histograma de los valores de POS obtenidos en la base de evaluación utilizada para entrenar las curvas a priori (Figura (a)) y la base de test de BTL (Figura (b)) con las siguientes configuraciones: baseline; y mllr con 5 señales. Modelo de palabras competidoras es CPC.n.l. indica que la palabra objetivo no se encuentra en la lista de las N-mejores de Viterbi (se consideró N=10. En la Figura (a) se muestran solamente los scores 1,2,3 y n.l dado que no se registraron otros valores de POS en la base de entrenamiento de las curvas a priori.)*

objetiva de las señales y puede ser la causa de la baja correlación obtenida con MLLR.

Los resultados obtenidos en BTL considerando dos niveles de evaluación se pueden ver en la Tabla 4.2. En este caso VTLN obtiene buenos resultados en todas las configuraciones, con mejoras en la correlación de hasta 2,3 % con el modelo de palabras competidoras CPC y de 4,5 % considerando sólo la versión "españolizada" de la palabra objetivo como competidora (modelo SPC). Esta diferencia en desempeño puede estar relacionada con la elección de las palabras competidoras, las que al no haber sido elegidas utilizando algún criterio fonético subjetivo afectan negativamente el proceso de normalización (dado que se está calculando la máxima verosimilitud considerando transcripciones muy distintas a las correctas).

Por otra parte, se puede observar en la Tabla 4.2 que MLLR presenta una degradación de

Configuración	Correlación promedio de los scores subjetivos - objetivos obtenida con el clasificador WD	Correlación promedio de los scores subjetivos - objetivos obtenida con el clasificador POS
Baseline	$0,695 \quad (0,712)$	$0,759 \quad (0,704)$
VTLN 1 señal	$0,711 \quad (0,742)$	$0,768 \quad (0,736)$
MLLR 5 señales	$0,535 \quad (0,459)$	$0,548 \quad (0,457)$

Tabla 4.2: *Correlación promedio de los scores subjetivos - objetivos obtenida con los clasificadores WD y POS en BTL considerando 2 scores de evaluación para las siguientes configuraciones: Baseline (sin técnicas de robustez); VTLN con una ventana de 1 señal; MLLR con ventana de 5, 25 y 100 señales. Los valores sin paréntesis corresponden a los obtenidos con el modelo de palabras competidoras CPC. Con paréntesis se muestran los de SPC.*

hasta un 35 %, que se obtiene al considerar el modelo SPC y el clasificador WD. Un análisis de la variación del WER similar al realizado anteriormente muestra que el WER aumenta en un 94,6 % para el score 5 (el WER del *baseline* es de 15,42) y se reduce en un 16,35 % (WER del baseline es de 88,33 %) para la nota 1. Este comportamiento es similar al que se observa en la Figura 4.5. Una posible explicación es que la base de test de BTL se encuentra ordenada solamente por locutor y no por score, ya que el sistema es no supervisado. Por ende, al agrupar las señales dado un cierto tamaño de la ventana de adaptación se consideran señales evaluadas subjetivamente con score 5 y 1. Lo anterior provoca que el reconocimiento de las señales con score 5 se vea afectado negativamente, mientras que el de las señales con score 1 aumenta. Esto indicaría que resulta aconsejable considerar una sola señal para cada adaptación, con el fin de evitar esta causa de degradación del sistema.

En la Tabla 4.3 se muestran los resultados obtenidos en las pruebas con la base BTA. MLLR se excluye del análisis ya que dada la configuración utilizada para este método no se calcula ninguna matriz de transformación con una ventana de una señal. Como se ex-

Configuración	**Correlación promedio de los scores subjetivos - objetivos obtenida con el clasificador WD**	**Correlación promedio de los scores subjetivos - objetivos obtenida con el clasificador POS**
Baseline	$0,581 \quad (0,534)$	$0,678 \quad (0,571)$
VTLN 1 señal	$0,491 \quad (0,517)$	$0,613 \quad (0,577)$

Tabla 4.3: *Correlación promedio de los scores subjetivos - objetivos obtenida con los clasificadores WD y POS en BTA considerando 2 scores de evaluación para las siguientes configuraciones: Baseline (sin técnicas de robustez); y VTLN con una ventana de 1 señal. Los valores sin paréntesis corresponden a los obtenidos con el modelo de palabras competidoras CPC. Con paréntesis se muestran los de SPC.*

plicó anteriormente, las uteraciones de esta base no se encuentran etiquetadas por locutor lo que impide el uso de una mayor cantidad de señales en la adaptación. El desempeño de VTLN es irregular en este caso, presentando mejoras leves (1,05 %) al considerar el clasificador POS y el modelo SPC. En esta configuración se presenta además la menor degradación obtenida con el clasificador WD.

Configuración	**Correlación promedio de los scores subjetivos - objetivos obtenida con el clasificador WD**	**Correlación promedio de los scores subjetivos - objetivos obtenida con el clasificador POS**
Baseline	$0,598 \quad (0,501)$	$0,813 \quad (0,598)$
VTLN 1 señal	$0,682 \quad (0,573)$	$0,745 \quad (0,598)$

Tabla 4.4: *Correlación promedio de los scores subjetivos - objetivos obtenida con los clasificadores WD y POS en BTN considerando 2 scores de evaluación para las siguientes configuraciones: Baseline (sin técnicas de robustez); y VTLN con una ventana de 1 señal. Los valores sin paréntesis corresponden a los obtenidos con el modelo de palabras competidoras CPC. Con paréntesis se muestran los de SPC.*

Finalmente, en la Tabla 4.4 se encuentran los resultados obtenidos en las pruebas con la base BTN. Nótese que el clasificador WD presenta mejoras en la correlación (con ambos modelos de palabras competidoras) de hasta un 14,04 %. El desempeño obtenido con POS resulta menos favorable, pero se puede observar que el mejor caso nuevamente corresponde al utilizar el modelo SPC, lo que se asemeja de cierta forma a lo observado en la Tabla 4.3.

4.6. Conclusiones

En este capítulo se han descrito las distintas etapas que componen el proceso de implementación de MLLR y VTLN en un sistema de evaluación de pronunciación basado en reconocimiento de voz. Estas técnicas se aplicaron tanto para el entrenamiento de las curvas a priori de los clasificadores WD y POS como para la evaluación del rendimiento del sistema.

Se observaron disminuciones entre 10,52 % y 35 % en la correlación entre scores subjetivos y objetivos con respecto al *baseline* al utilizar MLLR. Los resultados obtenidos indican una mayor degradación del sistema al considerar ventanas con más señales de adaptación. Además, las pruebas de evaluación del reconocimiento de la palabra objetivo mostraron que se produce un aumento considerable del WER para las señales con score subjetivo igual a 5 y una disminución de este valor para el resto (4, 3, 2 y 1).

Estos resultados sugieren que la aplicación de MLLR considerando varias señales de adaptación no es recomendable. En primer lugar, se produce un aumento importante del reconocimiento del sistema para todas las señales, sin importar la calidad de la pronunciación. Esto dificulta el proceso de discriminación entre las distintas clases y produce una menor correlación en el sistema. Otra causa posible de la degradación es la elección de las palabras competidoras, las cuales no se relacionan fonéticamente (de forma subjetiva) con la

palabra objetivo. De esta forma, al considerar una ventana mayor que una señal (necesario para aplicar MLLR eficientemente) y un modo no supervisado, la información de adaptación se degrada excesivamente ya que se consideran señales con niveles de pronunciación muy variados en una misma ventana. Esto produce una transformación inadecuada de los HMMs.

En los casos estudiados VTLN presentó mejoras en la correlación con respecto al *baseline* de hasta un $14,37\,\%$ y disminuciones de hasta un $15,49\,\%$. Los mejores resultados se obtuvieron con la base BTL de 2 scores y la base BTN, observándose una mejora promedio de la correlación de un 3,1 % y 5,01 % respectivamente con ambos clasificadores. Sin embargo, el método no fue eficaz para reducir las condiciones de *mismatch* presentes en la base de adultos no nativos del inglés, donde se observó una reducción de la correlación promedio de $6,8\,\%$. Las pruebas realizadas indican que VTLN se ve afectado por el tipo de modelos de palabras competidoras de forma similar a MLLR (aunque en menor medida debido a que se utilizan ventanas de menor tamaño). La correlación promedio utilizando el modelo SPC aumentó en un 2,61 % al aplicar VTLN, mientras que la obtenida con el modelo CPC disminuye en 2,34 %. Esto sugiere que resulta necesario considerar modificaciones a la implementación de esta técnica de normalización o al modelo de palabras competidoras antes de que su uso sea recomendable completamente.

Capítulo 5

Conclusiones

5.1. Conclusiones y análisis finales

En este trabajo se ha propuesto la implementación de *Maximum Likelihood Linear Regression* (MLLR) y *Vocal Tract Length Normalization* (VTLN) para reducir los problemas de robustez frente a cambios de locutor en evaluación de pronunciación automática basada en ASR.

MLLR es una técnica que modifica el modelo acústico del reconocedor mediante una transformación lineal de las medias de las componentes Gaussianas (Leggetter & Woodland, 1994). Por otra parte, VTLN realiza una normalización del banco de filtros de Mel utilizado para el procesamiento de las señales, con el fin de reducir la degradación del rendimiento del sistema debido a diferencias en la longitud del tracto vocal de los locutores (Lee *et al.*, 1998; Panchapagesan & Alwan, 2008). Estas técnicas se implementaron de forma no supervisada, considerando una cantidad de información limitada para la adaptación.

Los experimentos realizados en ASR demostraron la eficacia de las metodologías propuestas, llegando a reducciones del WER con respecto al sistema base de $30,56\,\%$ con MLLR y $16,23\,\%$ con VTLN en las mejores configuraciones (25 señales y 1 señal respectivamente). Este alto rendimiento se mantiene a pesar de considerar pocos datos, obteniéndose mejoras promedio del WER de $19,4\,\%$ y $6,34\,\%$ en MLLR con 5 señales y VTLN con 1 señal respectivamente. Estos resultados son comparables a los observados en la literatura (Lee, 2008; Leggetter & Woodland, 1994) al utilizar un método no supervisado. Cabe destacar que las pruebas realizadas con la configuración de MLLR indican la indican la necesidad de ajustar adecuadamente los parámetros de adaptación para obtener un rendimiento óptimo. Variaciones en la cantidad de matrices de transformación calculadas produjeron diferencias de hasta un $20\,\%$ en la reducción del WER con respecto al sistema base.

Al implementarse las técnicas en el sistema de evaluación de pronunciación descrito en (Molina *et al.*, 2008) se obtienen resultados menos favorables que en ASR. VTLN presentó un mejor rendimiento que MLLR, obteniéndose un aumento promedio de la correlación entre scores objetivos y subjetivos de $3{,}1\,\%$ y $5{,}01\,\%$ para dos de las bases de datos consideradas. Sin embargo algunas de las configuraciones que se probaron produjeron una degradación al desempeño, la que puede llegar hasta $15,49\,\%$ en el peor caso. Con MLLR se observan disminuciones entre $10{,}52\,\%$ y $35\,\%$ en la correlación entre scores subjetivos y objetivos con respecto al *baseline*. Como se muestra en el capítulo 4, este bajo rendimiento de MLLR está asociado a dos problemas principales: el aumento del reconocimiento de la palabra objetivo para los scores más bajos y una reducción para las señales con nota subjetiva igual a 5, lo que dificulta el proceso de discriminación entre clases; y la aplicación del método de forma no supervisada y utilizando un modelo de palabras competidoras sin relación fonética directa a la palabra objetivo, lo que implica que el proceso de adaptación disminuye el rendimiento del sistema. Este último efecto negativo resulta más notorio en MLLR que en VTLN debido a la utilización de ventanas de mayor tamaño, lo que produce una degradación

de la calidad de la información de adaptación.

En base a los análisis y resultados presentados en esta memoria, resulta recomendable la implementación de alguna de las técnicas propuestas para brindar robustez al proceso de reconocimiento de voz. Estos métodos pueden implementarse de forma transparente al locutor y mejoran significativamente el desempeño del sistema. En cuanto a la aplicación a los sistemas de evaluación de pronunciación, el comportamiento errático de VTLN y el bajo desempeño de MLLR evitan que sea posible sugerir su uso permanente considerando el estado actual de estas técnicas y del sistema de evaluación.

5.2. Trabajos propuestos a futuro

Si bien se ha presentado una metodología para la utilización de MLLR y VTLN en un sistema de evaluación de pronunciación, los resultados obtenidos evitan que sea recomendable su aplicación inmediata en el estado actual. Como parte de un trabajo futuro se proponen las siguientes tareas: realizar nuevas pruebas considerando otras bases de datos con más señales y etiquetadas por locutor (BTA y BTN contienen 100 y 72 señales respectivamente); modificar el conjunto de palabras competidoras utilizando información fonética para analizar el efecto que tiene en el desempeño con las técnicas de robustez; e implementación de MLLR en el reconocedor sin hacer uso de HTK para el proceso de adaptación, con el fin de tener un mayor control sobre esta técnica.

Glosario

Alineamiento: Proceso de asociación de cada vector de la secuencia de observación O con un estado s, perteneciente al modelo HMM evaluado, se obtiene la secuencia S.

ASR: *Automatic Speech Recognition*.

Baseline: Caso base o de referencia de un experimento.

CAPT: *Computer Aided Pronunciation Training*.

CDHMM: *Continuous Density Hidden Markov Model*.

Clustering: Proceso de organizar objetos en grupos cuyos miembros son similares de cierta forma.

CMN: *Cepstral Mean Normalization*.

Coeficientes Cepstrales: Parámetros acústicos que caracterizan a una señal de voz. Se basan en análisis en frecuencia de la señal.

Conjunto de Entrenamiento: Señales acústicas que se utilizan para determinar los parámetros de los modelos que describen el ASR o CAPT.

Conjunto de Test: Señales acústicas que evalúan el reconocedor y que no fueron utilizadas para el entrenamiento de los modelos que describen el ASR.

DCT: *Discrete Cosine Transform*.

DFT: *Discrete Fourier Transform*.

EM: *Expectation Maximization*.

Estado: Etapa de un HMM que representa un período estacionario de una señal acústica. Su valor es escalar.

Frame: Ventana o segmentación de la señal acústica, unidad mínima de análisis.

HMM: *Hidden Markov Model*.

LDC: *Linguistic Data Consortium*.

MAP: *Maximum A Posteriori*.

MFCC: *Mel Frequency Cepstral Coefficient*.

Mismatch: Situación que ocurre cuando las condiciones de evaluación y entrenamiento difieren. Pueden ser diferencias en el ambiente, locutor, ruido, etc.

MLLR: *Maximum Likelihood Linear Rregression*.

RM Task: *Resource Management Task*.

Score objetivo: Nota asociada a una uteración obtenida a partir de la evaluación realizada por el CAPT.

Score subjetivo: Nota asociada a una uteración obtenida a partir de la evaluación realizada por un evaluador experto del idioma inglés.

SMAP: *Structural Maximum A Posteriori*.

Unvoiced: Sonido producido sin vibración de las cuerdas vocales.

VILTS: *Voice Interactive Language Training System*.

Voiced: Sonido producido con vibración de las cuerdas vocales.

VTLN: *Vocal Tract Length Normalization*.

WER: *Word Error Rate*.

Referencias

AFIFY, M., & SIOHAN, O. 2000. Constrained Maximum Likelihood Linear Regression for Speaker Adaptation. *In: Sixth international conference on spoken language processing*. ISCA.

BECCHETTI, C., & RICOTTI, L.P. 1999. *Speech recognition: theory and C++ implementation*. John Wiley & Sons, Inc. New York, NY, USA.

COLE, R., MARIANI, J., USZKOREIT, H., ZAENEN, A., & ZUE, V. 1996. *Survey of the State of the Art in Human Language Technology*. http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html. Center for Spoken Language Understanding (CSLU).

FRANCO, H., NEUMEYER, L., KIM, Y., & RONEN, O. 1997. Automatic Pronunciation Scoring for Language Instruction. *In: Ieee international conference on acoustics speech and signal processing*. Institute of Electrical Engineers Inc (IEE).

GAROFALO, J., GRAFF, D., PAUL, D., & PALLETT, D. 1993. Continous Speech Recognition (CSR-I) Wall Street Journal (WSJ0) news, complete. *Linguistic data consortium, philadelphia*.

GAUVAIN, J.L., & LEE, C.H. 1994. Maximum a posteriori estimation for multivariate

Gaussian mixture observations of Markov chains. *Speech and audio processing, ieee transactions on*, **2**(2), 291–298.

HAMAKER, J.E. 1999. MLLR: A Speaker Adaptation Technique for LVCSR. *Lecture for a course at isip-institute for signal and information processing*.

HUANG, X., & LEE, KF. 1993. On speaker-independent, speaker-dependent, and speaker-adaptivespeech recognition. *Speech and audio processing, ieee transactions on*, **1**(2), 150–157.

HUO, Q., & LEE, C.H. 1997. On-line adaptive learning of the continuous density hidden Markovmodel based on approximate recursive Bayes estimate. *Speech and audio processing, ieee transactions on*, **5**(2), 161–172.

INDRAYANTI, L., USAGAWA, T., CHISAKI, Y., & DUTONO, T. 2006. Evaluation of Pronunciation by means of Automatic Speech Recognition System for Computer Aided Indonesian Language Learning. *Pages 553–556 of: Information technology based higher education and training, 2006. ithet'06. 7th international conference on*.

JELINEK, F. 1998. *Statistical Methods for Speech Recognition*. Massachusetts Institute of Technology (MIT) Press.

KWAN, K.Y., LEE, T., & YANG, C. 2002. Unsupervised N-Best Based Model Adaptation Using Model-Level Confidence Measures. *In: Seventh international conference on spoken language processing*. ISCA.

LAMEL, L., RABINER, L., ROSENBERG, A., & WILPON, J. 1981. An improved endpoint detector for isolated word recognition. *Acoustics, speech, and signal processing [see also ieee transactions on signal processing], ieee transactions on*, **29**(4), 777–785.

LAURILA, K., VASILACHE, M., & VIIKKI, O. 1998. A combination of discriminative and maximum likelihood techniques for noise robust speech recognition. *In: Acoustics, speech and signal processing, 1998. proceedings of the 1998 ieee international conference on*, vol. 1.

LEE, C.H. 2008. A critical overview on model adaptation in speech, language and media processing. *In: Isca dl at univ. of chile*. ISCA DL.

LEE, L., ROSE, R., & MIT, C. 1998. A frequency warping approach to speaker normalization. *Speech and audio processing, ieee transactions on*, **6**(1), 49–60.

LEGGETTER, C.J., & WOODLAND, PC. 1994. *Speaker Adaptation of HMMS Using Linear Regression.* University of Cambridge, Department of Engineering.

MOLINA, S., YOMA, N.B., WUTH, J., & VIVANCO, H. 2008. ASR based pronunciation evaluation with automatically generated competing vocabulary and classifier fusion. *Speech communication*.

MOLINA, S., YOMA, N.B., HUENUPÁN, F., GARRETÓN, C., & WUTH, J. 2009. Maximum entropy based reinforcement learning using a confidence measure in speech recognition for telephone speech. *Ieee transactions on audio speech and language processing (submitted)*.

MYRVOLL, T.A. 2002. *Adaption of Hidden Markov Models Using Maximum a Posteriori Linear Regression with Hierchical Priors*. Ph.D. thesis, Norwegian University of Science and Technology, Faculty of Information Technology, Mathematics and Electrical Engineering.

PANCHAPAGESAN, S., & ALWAN, A. 2008. Frequency warping for VTLN and speaker

adaptation by linear transformation of standard MFCC. *Computer speech & language*.

PAUL, D.B., & BAKER, J.M. 1992. The Design for the Wall Street Journal-based CSR Corpus. *In: Second international conference on spoken language processing*.

PICONE, JW, INC, T.I., & DALLAS, TX. 1993. Signal modeling techniques in speech recognition. *Proceedings of the ieee*, **81**(9), 1215–1247.

PITZ, M., MOLAU, S., SCHLÜTER, R., & NEY, H. 2001. Vocal Tract Normalization Equals Linear Transformation in Cepstral Space. *In: Seventh european conference on speech communication and technology*.

POTAMIANOS, A., & NARAYANAN, S. 2003. Robust recognition of children's speech. *Speech and audio processing, ieee transactions on*, **11**(6), 603–616.

PRICE, P., & RYPA, M. 1998. Speech Technology and Language Learning: Some Examples from VILTS The Voice Interactive Language Training System. *In: Proc. aatoll conference*.

PYLKKONEN, J. Estimating VTLN Warping Factors by Distribution Matching. *log*, **1**, 2.

RABINER, L.R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the ieee*, **77**(2), 257–286.

SAVOJI, MH. 1989. A robust algorithm for accurate endpointing of speech signals. *Speech communication*, **8**(1), 45–60.

SCHWARTZ, R., CHOW, Y., KIMBALL, O., ROUCOS, S., KRASNER, M., MAKHOUL, J., BERANEK, B., NEWMAN, I., & CAMBRIDGE, MA. 1985. Context-dependent modeling for acoustic-phonetic recognition of continuous speech. *In: Acoustics, speech, and signal processing, ieee international conference on icassp'85.*, vol. 10.

SENEFF, S., HURLEY, E., LAU, R., PAO, C., SCHMID, P., & ZUE, V. 1998. GALAXY-II: A Reference Architecture for Conversational System Development. *In: Fifth international conference on spoken language processing*. ISCA.

SHINODA, K., & LEE, C.H. 1997. Structural MAP speaker adaptation using hierarchical priors. *Pages 381–388 of: Automatic speech recognition and understanding, 1997. proceedings., 1997 ieee workshop on*.

WANG, S., CUI, X., & ALWAN, A. 2007. Speaker Adaptation With Limited Data Using Regression-Tree-Based Spectral Peak Alignment. *Ieee transactions on audio speech and language processing*, **15**(8), 2454.

YANG, X., MILLAR, J.B., & MACLEOD, I. 1996. On the Sources of Inter-and Intra-speaker Variability in the Acoustic Dynamics of Speech. *In: Fourth international conference on spoken language processing*. ISCA.

YOUNG, S., EVERMANN, G., KERSHAW, D., MOORE, G., ODELL, J., OLLASON, D., VALTCHEV, V., & WOODLAND, P. 2001. The HTK Book (for HTK Version 3.1). *Cambridge university engineering department*.

ZHAN, P., & WAIBEL, A. 1997. *Vocal tract length normalization for large vocabulary continuous speech recognition*. School of Computer Science, Carnegie Mellon University.