SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

Sistema Nacional de Educación Superior Tecnológica
Dirección General de Educación Superior Tecnológica

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

"POR MI PATRIA Y POR MI BIEN"

Un Método para la Identificación Automática de Lenguas Basado en la Transformada Wavelet

PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS
EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

I.S.C. JOSE MANUEL VARGAS MARTINEZ

NÚMERO DE CONTROL:

G00070864

DIRECTOR:

DR. ARTURO HERNANDEZ RAMIREZ

CODIRECTOR

ANA LILIA REYES HERRERA
Instituto Nacional de Astrofísica Óptica y Electrónica

Sistema Nacional de Educación Superior Tecnológica
Dirección General de Educación Superior Tecnológica

SUBSECRETARÍA DE EDUCACIÓN SUPERIOR
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA
INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

SECRETARÍA DE EDUCACIÓN PUBLICA
SEP

Cd. Madero, Tam., a 26 de Noviembre de 2008.

Área: Posgrado
Nº Oficio: U5.427/08
Asunto: Autorización de Impresión de Tesis

C. ING. JOSÉ MANUEL VARGAS MARTÍNEZ.
P r e s e n t e.

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su examen de grado de Maestro en Ciencias en Ciencias de la Computación, se acordó autorizar la impresión de su tesis titulada:

"Un Método para la Identificación Automática de Lenguas Basado en la Transformada Wavelet"

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con Usted el logro de esta meta. Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

A t e n t a m e n t e
"POR MI PATRIA Y POR MI BIEN"

M.P. María Yolanda Chávez Cinco
Jefa de la División

S.E.P.
DIVISION DE ESTUDIOS
DE POSGRADO E
INVESTIGACION
I T C M

" 2008, Año de la Educación Física y el Deporte "

MYCHC ' NLCO ' ajm*

Ave. 1o. De Mayo y Sor Juana I. de la Cruz, Col. Los Mangos, C.P. 89440 Cd. Madero, Tam.
Teléfono : 01 (833) 357 48 20 al 29
Internet : www.itcm.edu.mx Correo Electrónico : itcm@itcm.edu.mx

Dedicatoria

Dedico este trabajo a mis padres:
Adelina y José Manuel.

Agradecimientos

Quiero agradecer en primer lugar a Dios, que ha escuchado mis plegarias, y en que me apoyo día a día.

A mis padres, los cuales me han dado lo mejor de sí mismos, a mi madre por los días de desvelo, por su comprensión, su cariño y por su fuerza para salir adelante, a mi padre, por su fuerza, su honestidad, sus consejos, en fin, por muchas cosas, gracias.

Al Dr. Arturo Hernández Ramírez, un gran maestro, del cual he aprendido mucho, gracias por su confianza y sabiduría, su sentido común y profesionalismo, y por el cual esta tesis tomo forma.

A la Dra. Ana Lilia Reyes Herrera, por indicarme el camino a seguir en esta tesis, a su tiempo y confianza conmigo, muchas gracias.

Al Maestro Apolinar Ramírez Saldivar, gran maestro, gracias por sus consejos, su confianza, por su profesionalismo, su sentido crítico e incisivo.

A mis amigos del alma, a Jessica Rojas, Marco Aguirre, Hugo Fuentes, Los cuatro fantásticos, con los que he convivido estos años, a Jessica por su chispa, a Marco por sus bromas y a Hugo por su sinceridad.

A mis casi hermanos Roy, Oscar, José Miguel y RICC, a los cuales agradezco su ayuda y quienes han creído en mí y me han alentado, cuando mas he necesitado han estado ahí.

A mi entrañable amigo Rolando Demes al cual admiro, por lograr lo que ha logrado, gracias por creer en mí, y enseñarme que muchas cosas son posibles.

Al Capitán Noe Martínez y su Familia, gran amigo, gracias por sus consejos y su confianza

A los tres Pedros, Pedro Tomas, Pedro Luis y Pedro Torres, por la convivencia amena en el LVR, y que demostraron a muchos que al final valen los hechos, no las palabras.

A mis compañeros linuxeros Jorge Curiel, Alfredo y Peter con los que se puede hablar de linux y hacer bromas de Windows y que ¡viva Linux!

A mis compañeros de maestría de otras generaciones, A Edgar y Violeta por su compañerismo, a los ocurrentes Marcela y Gilberto, a Alberto y Paco.

A todos aquellos que pudiera olvidar ahora, por sus atenciones, gracias.

Resumen

La identificación automática de lenguas habladas (LID) es aquella que tiene como objetivo el determinar qué lengua habla un hablante cualquiera utilizando una muestra de voz, todo esto mediante computadora. LID está relacionada con el reconocimiento del habla, con la diferencia de que no busca entender el mensaje hablado, solo en qué lengua fue expresado. Sus aplicaciones son variadas: a) como procesamiento previo a sistemas de traducción multilingüe, o para la canalización de los hablantes con el personal adecuado (en llamadas telefónicas de soporte y emergencia), b) también en uso de interfaces de voz que eligen el idioma más adecuado para el usuario, c) la interacción multilingüe en la que se busca que dos hablantes se comuniquen usando su lengua nativa, mediante un traductor de tiempo real. Su importancia se ha incrementado hoy en día debido a la globalización.

En la actualidad los sistemas LID más exitosos son aquellos que utilizan información fonotáctica, los cuales, utilizan información fonética y acústica para obtener características más finas de las lenguas desde el punto de vista lingüístico, pero implica un coste computacional y lingüístico alto. Por otro lado existen los trabajos del tipo acústico que evitan estos costes con el fin de aplicarse a casi cualquier lengua incluso aquellas que tienen pocos recursos lingüísticos, como las lenguas indígenas, en particular las de México, pero este enfoque posee niveles de identificación menores, y por lo tanto, es una línea de investigación abierta.

El presente trabajo utiliza el enfoque acústico, basado como muchos trabajos en el ritmo de los lenguajes hablados, el método propuesto utiliza como técnica de procesamiento de las señales acústicas la Transformada Wavelet Db2, y enfatiza el uso de muestras cortas de habla como su principal ventaja, puede manejar como mínimo 4 segundos de habla, y un número de atributos menor a 200 por clasificador, con buenos resultados arriba de 90% de clasificación correcta, lo que indica la posibilidad de aplicar este método en la construcción de un sistema real de identificación de lenguas en un futuro no muy lejano.

Summary

The spoken language identification (LID) has as goal to decide what language speaks an unknown talker using a sample of voice, all this by means of computer. LID is related with the automatic speech recognition, but it doesn't search to understand the spoken message, only the language used. Its applications are varied, such as pre-processing to systems of multilingual translation, for channeling of talkers with the appropriated personnel (in telephone calls of support and also emergency) in voice interfaces, that select the language more appropriated for the user, the multilingual interaction which search that two talkers communicate themselves using their native language, by means of real-time translator. Nowadays its importance has increased because of the worldwide globalization and problems like the terrorism.

At present, the systems more successful are those that phonotactics information, they use phonetic and acoustic information to obtain fine features of language from the linguistic viewpoint, but this implies a high computer and linguistic costs. In another hand, there are acoustic works that avoid these costs in order to applying to almost any language, even those that have poor linguistic resources, like the native languages, specially from Mexico, but the acoustic works have low levels of identification, therefore, a line of research.

The present work has acoustic approach, which is based like many others in the spoken language rhythm, it presents a method which use an acoustic signals processing technique, the Db2-Wavelet Transform, that emphasizes, the use of speech short samples as its main advantage, this can manage as minimum 4 second of speech, and a small number to 200 attributes per classifier, with good results greater than 90%, which indicate the possibility of applying this method in the building of a real system of language identification, in the not too distant future.

Índice

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

—Aguarden se escucha algo my débil,….Un sonido robótico salió del intercomunicador.

—Señor, tengo fluidez en 6 millones de formas de comunicación, y esta señal, no es usada por la alianza, podría tratarse de un código imperial—C3PO.

—Sea lo que sea, no es amistoso—Han Solo.

—George Lucas, StarWars Episodio V

1.1. Antecedentes

C3PO es un personaje de ciencia ficción, que aparece en la saga de películas *StarWars* (Guerra de las Galaxias) de George Lucas, este es un personaje artificial, el cual es un robot de protocolo, que tiene la capacidad de interactuar con diferentes razas de seres en la galaxia donde se desarrolla la historia, y sobre todo hacer que se entiendan unos con otros, mediante traducción de lo que dicen cada uno de los hablantes, pero su más grande cualidad es que es capaz de **identificar la lengua que cada uno habla**, y aunque él no pueda identificar la lengua correcta, puede encontrar su relación con otras lenguas de su base de conocimiento y esto le permite inferir lo que trata de decir el hablante, este personaje combina tres campos de estudio del habla, **El reconocimiento automático del habla**, **la síntesis de voz** y **la identificación automática de lenguas habladas**. C3PO es la gran meta o gran sueño que aun no ha alcanzado el campo de la identificación automática de lenguas.

La identificación automática de lenguas habladas se encuentra emparentada con el reconocimiento automático del habla, pero a diferencia de este no se interesa por el mensaje. La identificación automática de lenguas trata sobre la identificación de una lengua hablada tomando una muestra de habla de un hablante cualquiera, es decir, la capacidad de tomar señales digitales de habla con una computadora y determinar por métodos computacionales, en qué lengua fue expresada la señal de habla.

Las aplicaciones de la identificación del lenguaje se pueden agrupar en dos grandes categorías: el procesamiento previo para sistemas y el procesamiento previo para humanos. El procesamiento previo para sistemas se refiere a la necesidad de los reconocedores de habla de saber qué lengua tienen que procesar, de otro modo se tendrían que tener varios reconocedores corriendo en paralelo tratando de identificar las palabras expresadas. Por ejemplo, un aeropuerto que colecta información de portavoces multilingües para recuperar información de los vuelos, tiene la necesidad de saber la lengua y usar el traductor apropiado para el almacenaje de dicha información; otro ejemplo es un sistema de traducción multilingüe, que requiere de la identificación para usar el traductor correcto.

El procesamiento previo para humanos se refiere a que en algunos casos es necesario saber qué lengua hablan. Por ejemplo, en llamadas telefónicas para pedir información sería más eficiente si la persona recibe información en su propio idioma, en llamadas de emergencia la tensión hace que las personas hablen en su idioma nativo, en otros casos hablantes de otras lenguas no pueden comunicarse y se puede tener una idea equivocada de lo que necesitan. Otro ejemplo, sería la interacción multilingüe, donde dos personas de diferentes lenguas tratan de comunicarse mediante un traductor que les habla en su propia lengua, que es el caso de C3PO o un C3PO telefónico donde cada hablante detrás de cada teléfono solo oye el mensaje del otro en su propio idioma.

La tarea no es trivial, como muchas capacidades humanas que la inteligencia artificial ha tratado de emular, las primeras investigaciones se remontan hasta la década de 1970, donde se buscaron sonidos particulares a cada lengua con el fin de identificarla, con el paso del tiempo el

tipo de características se fueron refinando, y también las técnicas de identificación. Muthusamy menciona 3 categorías de métodos para identificar lenguas [Muthusamy 1992]:

- Identificación usando características acústicas
- Identificación usando categorías fonéticas amplias
- Identificación usando categorías fonéticas finas

[Reyes 2007] menciona por su parte tres enfoques para resolver el problema, el primero usa segmentación del habla en fonemas y usa un reconocedor de fonemas, creando modelos de los lenguajes mediante el uso de árboles de fonemas llamados *n-grams*. Este enfoque requiere de grandes recursos lingüísticos conocidos como información fonotáctica, que son exclusivos de cada idioma, precisa esta metodología un etiquetado manual de los fonemas, además los n-grams necesitan gran procesamiento computacional y el agregar un nuevo idioma implica también un gran costo desde el punto de vista lingüístico. Debido al coste de los n-grams y el etiquetado manual de los fonemas, surgió el segundo enfoque una adecuación del método, que sustituye los fonemas, por una secuencias de fonemas, conocidas como *tokens* [Torres-Carrasquillo 2002].

Aunque los resultados de los anteriores son los mejores, su aplicación en lenguas de pocos recursos lingüísticos es difícil, debido a eso un tercer enfoque ha sido estudiado desde hace varias décadas, que utiliza directamente características acústica de las señales, como la prosodia y el ritmo de las lenguas. Los dos primeros enfoques se pueden ubicar en la identificación de categorías fonéticas finas y el último en el de características acústicas de Muthusamy.

1.2. Definición del Problema

A lo largo del tiempo se ha detectado una problemática clara en las investigaciones de este tipo, la cual radica en que teniendo una señal de habla digitalizada, se deben elegir ciertas características (frecuencia fundamental, timbre, tono, entonación) o unidades (vocales, consonantes) con el fin de distinguir lenguas.

Nuestra problemática se resume en qué extraer y como extraerlo, para poder caracterizar a la señal de habla y poder identificar lenguas habladas.

1.3. Justificación

Las diversas aplicaciones de la identificación automática de lenguas mencionadas anteriormente son importantes. Pero la razón principal de este proyecto está basada en la necesidad de construir un identificador de lenguas que se prescinda del uso de información fonotáctica (información lingüística léxica y gramatical) y que trabaje con únicamente información acústica en segmentos muy cortos de habla. Con el fin de aplicarse en un futuro a lenguas marginadas (Como las lenguas indígenas de Mexico). Para probar el método, se usan 9 lenguas del mundo de una base de datos estándar llamada OGI_TS.

1.4. Objetivos

1.4.1. Objetivo general

La identificación de lenguas mediante la extracción de información acústica de la señal de habla

1.4.2. Objetivos específicos

- Entender en forma teórica y práctica las característica de la Transformada Wavelet Db2, técnica que utilizada para extraer características de la señal de voz.
- Entender el concepto de ritmo de los lenguajes, así como su extracción usando la Transformada Wavelet Db2.
- Adquirir la base de datos de lenguas necesarias para hacer el entrenamiento y pruebas, que permitirá cumplir los objetivos anteriores.
- Utilizar segmentos cortos de habla (10, 5, y 4 segundos), así como la eliminación de pausas largas en las señales de habla.

1.5. Aportación

La aportación de este proyecto consiste principalmente en la metodología utilizada, se aporta un método que es capaz de identificar pares de lenguas con el uso mínimo de 4 segundos de habla y con un número de atributos debajo de 300, lo que permitiría aplicarlo a la realidad. Se aporta un método fácil de implementar, en lenguajes de programación conocidos como lo son java y matlab.

1.6. Limitaciones

Se limitará al uso de 9 lenguas, Inglés, Alemán, Español, Japonés, Tamil, Mandarín, Coreano, Farsi y Vietnamita.

Se limitará al uso de clasificadores binarios de lenguas (2 lenguas), debido a que mediante el uso de características acústicas, los mejores resultados son obtenidos de esta forma.

Se limita al uso de características acústicas, pues su aplicación a futuro es en lenguas de recursos lingüísticos limitados.

Se contempla el uso de habla espontánea, además en ella dos aspectos sumamente difíciles:

- Diferentes hablantes: los hablantes no se repiten (Independencia del hablante)
- Diferente Género: Hombres y Mujeres

Se limita a la construcción de un método de identificación de lenguas, no un sistema aplicado en situaciones reales.

1.7. Organización de la Tesis

La tesis está organizada de la siguiente forma:

Capítulo 2 Marco Teórico. En este capítulo se exhibe el fundamento acústico, de procesamiento digital de señales, matemático, técnicas de minería de datos y aprendizaje de máquina, que hacen posible el presente trabajo.

Capítulo 3 Estado del Arte. Se presenta un panorama de los trabajos más interesantes que conforman la historia de la identificación automática de lenguas a lo largo de casi 4 décadas.

Capítulo 4 Metodología. En este capítulo se describe la aplicación del conjunto de técnicas mencionadas en el capítulo 2, con las que se resuelve el problema de identificar lenguas.

Capitulo 5 Pruebas y Resultados. En este capítulo se describen la conformación de pruebas y los resultados obtenidos de ellas.

Anexo A Wavelets Haar y Db2. Se describe la obtención de los coeficientes filtro de la Wavelet de Haar y Db2

Anexo B Códigos. Se muestran los códigos fuente de las técnicas usadas en el proceso de identificación de lenguas.

Capítulo 2

Marco Teórico

—Hola, ¿HAL puedes escucharme?, ¿HAL? — Dave Bowman — Afirmativo, Dave, te escucho — HAL

—Abre las puertas del compartimento del pod — Dave Bowman — Lo siento Dave, me temo que no puedo hacer eso — HAL

— ¿Cual es el problema? — Dave Bowman —Yo pienso que tú sabes muy bien cuál es el problema — HAL

— ¿De qué estás hablando HAL? — Dave Bowman — La misión es muy importante para mí, como para permitir que la arriesgues — HAL

— ¿Yo no sé de qué estás hablando? — Dave Bowman — Yo se que tú y Frank están pensando desconectarme, y me temo que eso, es algo que no puedo permitir — HAL

— ¿Pero de donde sacaste esa idea? — Dave Bowman —Dave, aunque tomaste precauciones en el pod contra mi capacidad de oír, yo puedo ver, tus labios moverse — HAL

— Stanley Kubrick y Arthur C. Clarke, 2001: A Space Odissey

En este capítulo se describen los conceptos básicos y el fundamento matemático necesario para realizar este trabajo. Se comenzará hablando de la señal del habla, la importancia del ritmo de los lenguajes, y las técnicas de procesamiento de señales, minería de datos y aprendizaje de máquina necesarias para este trabajo.

2.1. El Habla

HAL 9000 es un personaje de ciencia ficción muy conocido, que aparece en la película *2001: A Space Odissey*, de Stanley Kubrick, HAL es un agente artificial que interactúa con humanos mediante el uso del lenguaje natural, usando para ello reconocimiento de voz y síntesis de la misma, además, llegado cierto momento logra ser capaz, de leer los labios de una de las personas con las que interactúa. HAL es un agente sofisticado, engloba reconocimiento y síntesis de voz, retención y extracción de información, e inferencia, un agente aun difícil de construir en la realidad, debido a que muchas de estas capacidades siguen siendo investigadas en la actualidad.

El Lenguaje Natural que emula HAL es muy avanzado, su experiencia con el reconocimiento de voz lo conduce a relacionarlo con los movimientos de los labios de Dave, quien planea desconectarlo, pues debido a una falla, HAL se vuelve radical en cuanto a su misión. El habla que es su principal habilidad lleva intrínseca una gran dificultad de emular; a continuación se mostrarán algunas de las características que hacen del habla un área muy interesante.

2.1.1. El Proceso de Producción y Percepción del Habla en Seres Humanos.

El proceso de producción comienza cuando el hablante formula un mensaje que espera trasmitir a un oyente a través del habla. La contraparte del proceso de formulación del mensaje, es el proceso de creación de texto impreso, expresando las palabras del mensaje. El siguiente paso es la conversión del texto en un código de lenguaje. Esto burdamente corresponde a la conversión del texto impreso en una secuencia de fonemas, correspondientes a los sonidos que forman las palabras, a lo largo de la secuencia con marcadores de prosodia denotando duración de sonidos, volumen de sonidos, y acento del "pitch" (Frecuencia más baja en el espectro de frecuencias del habla) asociado a los sonidos [Rabiner y Juang 1993].

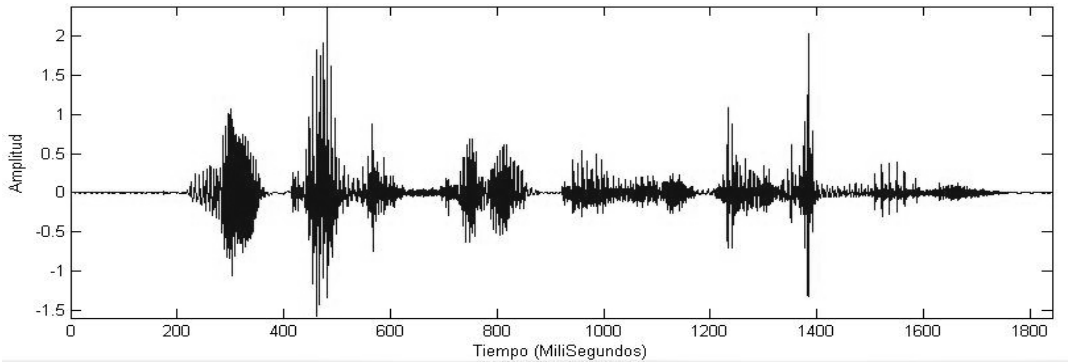


Figura 2.1. Ejemplo de la señal del habla

Una vez que la señal de sonido es generada y propagada hacia el oyente, comienza el proceso de percepción. Primero el oyente procesa la señal acústica a los largo de la membrana basilar en el interior de la oreja, la cual provee de un análisis espectral de la señal de entrada. Un proceso de transducción neuronal convierte la señal espectral de la salida de la membrana basilar, en señales activas en el nervio auditivo, correspondiendo burdamente a un proceso de extracción de características. De una manera que no es aun bien comprendida, la actividad neuronal a los largo del nervio auditivo es convertida en un código de lenguaje, en los centros más altos de procesamiento dentro del cerebro y se logra finalmente la comprensión del mensaje [Rabiner y Juang 1993].

2.1.2. Funcionamiento del Mecanismo Vocal Humano en el Proceso de Producción del Habla

El tracto vocal comienza en la abertura de las cuerdas vocales o glotis y termina en los labios (figura 2.2). El tracto vocal se compone de la faringe (la conexión del esófago a la boca), o cavidad oral, en los varones promedio la longitud total del tracto vocal es de 17 cm. El área trasseccional del tracto vocal determinado por las posiciones de la lengua, labios, mandíbula y velo del paladar varía de 0 (completamente cerrada) a cerca de 20 cm^2. El tracto nasal comienza en el velo del paladar y termina en las ventanas de la nariz. Cuando el velo del paladar[1] es bajado el tracto nasal está acústicamente acoplado a el tracto vocal para producir los sonidos nasales del habla.

[1] Consiste en un tejido blando, situado en la parte más posterior del paladar, que termina en un pliegue denominado úvula o, comúnmente, campanilla

Un diagrama esquemático del mecanismo vocal humano, se muestra en la figura 2.3. Como el aire es expelido por los pulmones a través de la tráquea, la tensión de las cuerdas vocales dentro de la laringe es causada por la vibración que produce el flujo de aire.

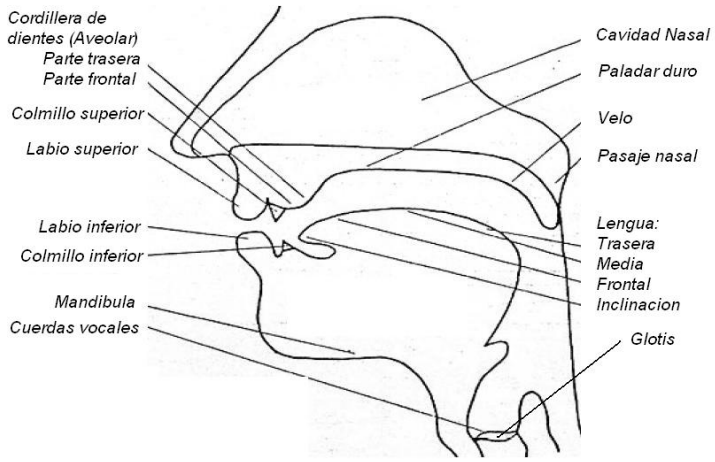


Figura 2.2. Aparato fonatorio

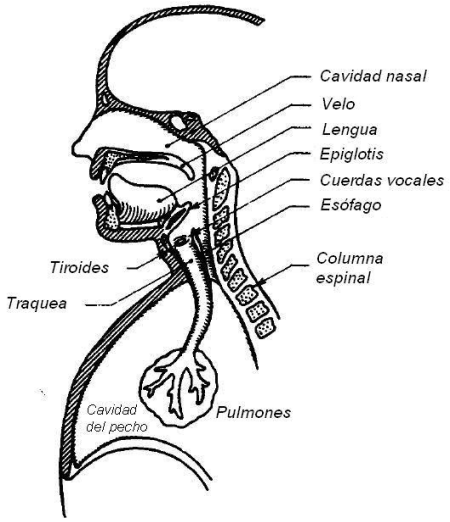


Figura 2.3. Vista esquemática del mecanismo vocal humano

La figura 2.4 muestra la velocidad del volumen glotal y su presión acústica producida en la boca, para un sonido de vocal típica. La forma de onda glotal muestra un desarrollo gradual a un tren de pulsos de aire aproximadamente-periódico, tomando cerca de 15 milisegundos para un estado de extensión fija.

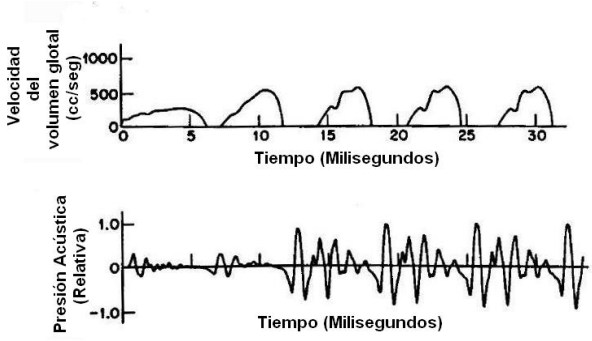


Figura 2.4. Velocidad del volumen glotal y presión acústica

Una representación simplificada del mecanismo completo fisiológicamente para crear habla es mostrada en la figura 2.5. Los pulmones y los músculos asociados actúan como fuente para la excitación del mecanismo vocal. El músculo obliga a desalojar aire fuera de los pulmones y a través de los bronquios y tráquea. Cuando las cuerdas vocales están tensas, el flujo de aire causa que vibren, produciendo los llamados sonidos sonoros del habla (voiced). Cuando las cuerdas vocales son relajadas, a fin de producir un sonido, el aire fluye o debe pasar por una contracción en el tracto vocal con lo cual llega a ser turbulento, produciendo los llamados sonidos sordos del habla (unvoiced) [Rabiner y Juang 1993].

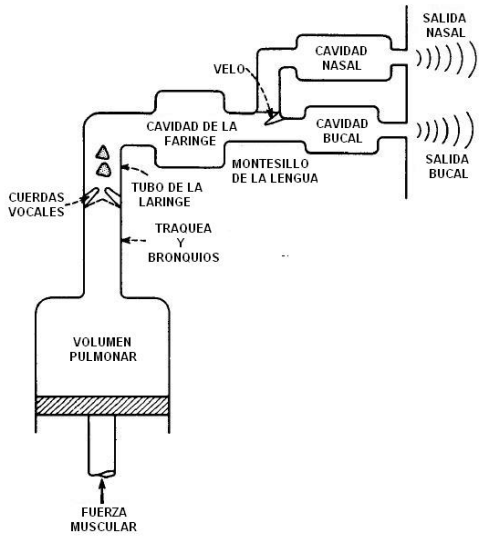


Figura 2.5. Representación esquemática de un sistema de generación de habla

El habla es producida como una secuencia de sonidos. Por lo tanto, el estado de las cuerdas vocales así como también las posiciones, formas y tamaños de varias articulaciones, cambian a lo largo del tiempo para reflejar el sonido que está siendo producido.

2.1.3. La Representación del Habla en los Dominios del Tiempo y la Frecuencia

La señal del habla es una señal de variación lenta en el tiempo, en el sentido que, cuando examinamos sobre un periodo de tiempo suficientemente corto (entre 5 y 100 milisegundos) sus características son bastante estacionarias; sin embargo sobre periodos largos de tiempo (sobre el orden de 1/5 segundos) las características cambian para reflejar los diferentes sonidos que están siendo emitidos.

Existen algunas maneras de clasificar (etiquetar) eventos en el habla. Tal vez el más simple, es mediante el estado de la fuente de producción del habla - las cuerdas vocales. Esta es una convención aceptada para usar la representación de tres estados los cuales son (1) silencio (S), donde el habla no es producida; (2) Sordas (U), en el cual las cuerdas vocales no vibran, además la forma del habla resultante es no periódica o aleatoria en esencia; (3) Sonoras (V), en el cual las cuerdas vocales están tensas y por lo tanto vibran periódicamente cuando el aire fluye desde los pulmones, además la forma de onda del habla resultante es aproximadamente-periódica. Se debe aclarar que la segmentación de la forma de onda en regiones bien definidas como zonas de silencio, sordas y sonoras no es exacta, es a menudo difícil distinguir un sonido sordo débil de un silencio, o un sonido sonoro débil de un sonido sordo o un silencio.

Una manera alternativa es la caracterización de la señal del habla mediante la representación espectral. Tal vez la representación más popular de este tipo es la del espectrograma del sonido, la cual es una representación de la intensidad del habla en tres dimensiones, en bandas de frecuencia diferentes, sobre el tiempo [Rabiner y Juang 1993].

2.1.4. El Ritmo

La noción de ritmo es compleja y extensa, pudiendo ser definida desde distintas disciplinas que lo incorporan como un eje central de sus actividades o como un componente de los fenómenos que estudian. Pero en particular desde una mirada lingüística, [Liberman y Prince 1977], señalan que el ritmo es de naturaleza jerárquica y está representado a diferentes niveles como una alternancia de pulsos fuertes y débiles, los cuales corresponden a las sílabas.

La importancia del ritmo del habla tiene que ver con la diferenciación o discriminación de los lenguajes. La clasificación fonética del ritmo de los lenguajes como el *stress-timing* (Inglés, Alemán) o *syllable-timing* (Español, Francés, Italiano) es atribuido a [Pike 1946] y [Abercrombie 1967] incluyendo lenguajes *mora-timing* (lenguajes orientales) en la clasificación, lo cual sugiere que estas tres categorías rítmicas se pueden aplicar a todas los lenguajes del mundo.

Pero en trabajos más recientes basados en la duración de intervalos entre los acentos en los lenguajes de tipo *syllable-timing* y *stress-timing* proveen una estructura en la cual estas dos categorías binarias son sustituidas por una continua [Dauer-1983]. Dauer encuentra ciertas características fonéticas y fonológicas en los leguajes de las dos familias:

- Estructura silábica: los lenguajes de tipo stress-timing tienen una gran variedad de silabas más que los lenguajes del tipo sylable-timing, por lo tanto tienden a tener silabas más pesadas
- Reducción Vocal: en lenguajes stress-timing, las silabas no acentuadas tiene un sistema de reducción vocal, y las vocales no acentuadas son consistentemente cortas.

Las diferencias rítmicas entre los lenguajes entonces están principalmente relacionadas con sus estructuras silábicas y la presencia (o ausencia) de reducción vocal. En los diferentes trabajos de lingüística y psicolinguística existe controversia sobre el estatus del ritmo en lenguajes del mundo ilustrando dramáticamente la dificultad de segmentar el habla en unidades rítmicas correctas. Aun si existe una correlación entre la señal del habla y el ritmo lingüístico, extender una relación pertinente parece difícil [Ramus 1999]. Pero en el ámbito del lenguaje natural en el área de inteligencia artificial se sospecha que pueda encontrarse en las bajas frecuencias de las señales.

2.2. Señales y Sistemas

El habla es una señal y por lo tanto puede ser tratada como tal. A continuación se menciona el habla desde el punto de vista de señales, debido a que las señales manejadas en este proyecto son discretas.

Matemáticamente podemos modelar señales y sistemas como funciones. Una señal es una función que mapea un dominio, a menudo tiempo y espacio, en un rango, frecuentemente una medida física tal como la presión del aire o la intensidad de la luz. Un sistema es una función que mapea señales desde su dominio – sus señales de entrada – en señales en su rango – sus señales de salida. El dominio y el rango son conjuntos de señales (espacios de señales). Así los sistemas operan sobre funciones [Lee y Varaiya 2000].

2.2.1. Señales de Audio

Nuestros oídos son sensibles al sonido lo que consiste físicamente en variaciones rápidas en la presión del aire que reciben. Así el sonido puede ser representado como una función

Sonido: Tiempo → Presión

donde la ***Presión*** es un conjunto que consiste en todos los valores de la presión del aire y el Tiempo es un conjunto que representa el intervalo sobre el cual la señal dura.

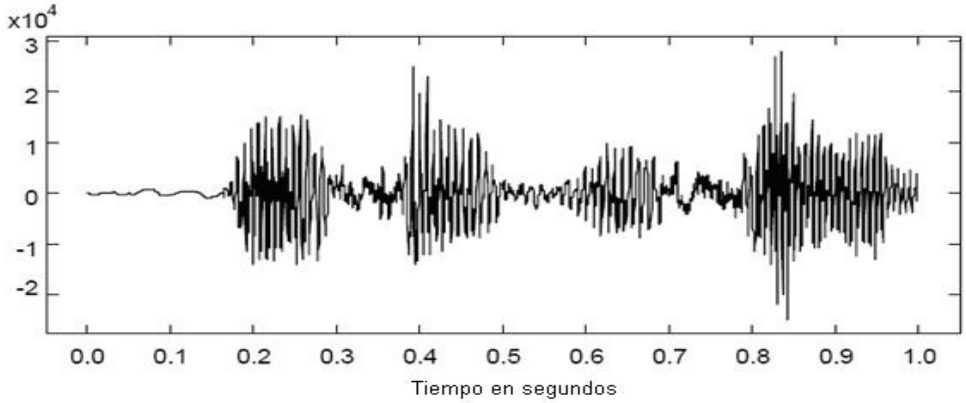


Figura 2.6. Forma de onda de un fragmento de habla

Por ejemplo, un segundo de una señal de voz es una función de la forma

Voz: [0, 1] → Presión

donde [0,1] representa un segundo del tiempo.

En la figura 2.6 el eje vertical no representa directamente la presión del aire. Esto es obvio porque la presión del aire no es negativa. De hecho los valores posibles de la función de voz son representados en la figura 2.6 como enteros de 16 bits, adecuados para su almacenamiento en una computadora. Llamaremos al conjunto de enteros de 16 bits *Ints16*={-32768, …, 32768}. El hardware de audio de la computadora es responsable de convertir los miembros del conjunto *Ints16* en presión del aire.

Nuestros oídos, después de todo no son sensibles a la constante presión del aire en el ambiente. Así, tomamos *Presión = Reales*, los números reales, donde la presión negativa significa una caída en la presión relativa a la presión del aire en el ambiente [Lee y Varaiya 2000].

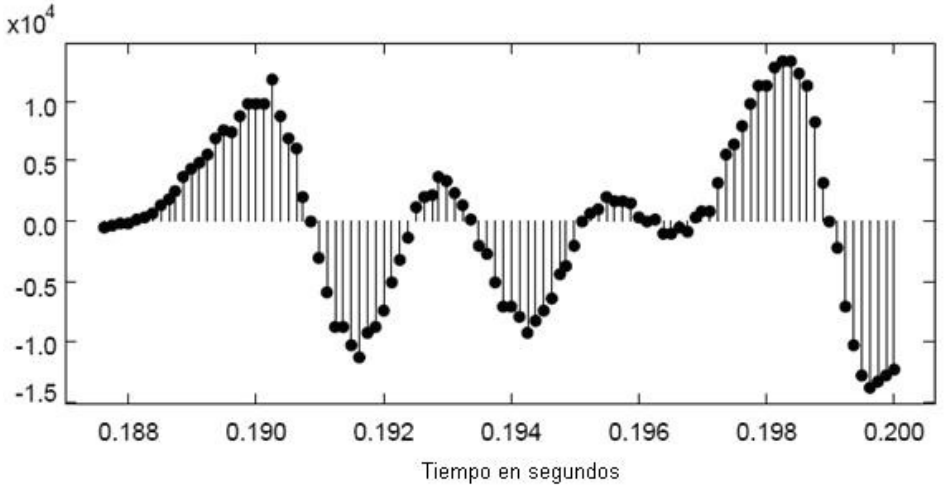


Figura 2.7. Representación en tiempo discreto de una señal de habla

Los ejes horizontales de la figura 2.6 sugieren que el tiempo varía de forma continua de 0 a 1. Sin embargo, una computadora no puede manejar directamente tal continuidad. El sonido es representado no como una forma de onda continua, pero sí como una lista de números (para la

calidad de voz 8000 números por segundo). Un acercamiento de un sección de la forma de onda del habla es mostrada en la figura 2.7 y muestra el graficado de 100 puntos de datos (llamados **muestras**). Ya que hay 8000 muestras por segundo, los 100 puntos en la figura 2.7 representan 100/8000 segundos o 12.5 milisegundos de habla [Lee y Varaiya 2000].

Una señal de un segundo de voz de tiempo discreto en una computadora es una función

Voz (Computadora): Tiempo Discreto → *Ints16*

donde T**iempo discreto = {0, 1/8000, 2/8000,…, 8000/8000}** es el conjunto de muestras de tiempo.

2.2.2. Señales Discretas y Muestreo

Las señales discretas a menudo surgen de dominios continuos mediante muestreo. El dominio continuo tiene un número infinito de elementos. Aunque el dominio [0, 1] ⊂ Tiempo, este representa un intervalo de tiempo finito, el cual tiene un número infinito de elementos. Una manera común para aproximar una función es muestreando uniformemente su dominio continuo.

Ejemplo. Si muestreamos 10 segundos a lo largo del dominio de la voz

Voz*: [0.10] → *Presión

tomaremos de la señal 10 000 instantes en un segundo

Voz muestreada: {0, 0.0001, 0.0002,…, 9.9998, 9.9999, 10} → *Presión*

Como se nota el muestreo uniforme significa la recolección de un conjunto espaciado uniformemente de puntos del dominio continuo [0, 1].

2.2.3. Cuantificación

Aunque *Voz muestreada* es un ejemplo que tiene un dominio finito, podemos sin embargo no poderlo almacenar en una cantidad finita de memoria. Para ver porqué, supongamos que el rango de la *Presión* de la función *Voz muestreada* es el intervalo continuo [a, b]. Para representar cada valor en [a, b] se requiere una precisión infinita. Por ejemplo, una palabra de 8 bits de largo puede tener $2^8 = 256$ valores diferentes. Así debemos aproximar cada número en el rango [a, b] por uno de los 256 valores.

El método más común es **cuantificar** la señal, un acercamiento común, es la elección de 256 valores uniformemente espaciados en el rango [a, b] y aproximar cada valor en el rango [a, b] por uno de esos 256 valores. Una alternativa de la aproximación, llamada **truncamiento**, es la elección del más grande de los 256 valores que es menor o igual al valor deseado [Lee y Varaiya-2000].

Ejemplo: la figura 2.8 muestra una señal *PureTone*, *SampledPureTone* obtenida después de muestrear, y una *DigitalPureTone* cuantificada obtenida usando 4 bits o un truncado de nivel 16. La señal *PureTone* tiene un dominio y un rango continuo, mientras *SampledPureTone* (representada por círculos) tiene un dominio discreto y un rango continuo, y *DigitalPureTone* (descrita con x's) tiene un dominio y un rango discreto. Solamente el último de estos puede ser representado precisamente en una computadora.

Es costumbre llamar una señal con dominio y un rango continuo como *PureTone* una **señal analógica,** y una señal con dominio y rango discreto como *DigitalPureTone* una **señal digital** [Lee y Varaiya 2000]**.**

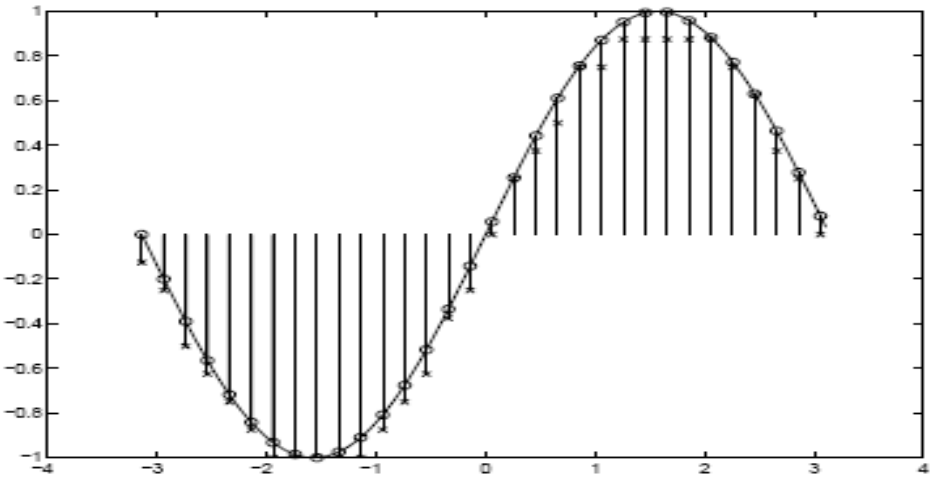


Figura 2.8. PureTone (curva continua), SampledPureTone (circulos), DigitalPureTone (x's)

2.3. Las Series de Fourier

En esta sección se menciona las propiedades que tiene una base ortonormal, como *las series de Fourier* y que permite la representación de una señal en términos de estas series, y tales propiedades se aplican de la misma forma a las Wavelets. La transformación de una función en sus componentes Wavelets tiene mucho en común con la transformación en sus componentes de Fourier.

Jean Batiste Fourier descubrió que podía descomponer cualquier cantidad de funciones en funciones componente de una función trigonometría periódica estándar. Aquí solo consideraremos funciones definidas en el intervalo [-π, π]. Las funciones seno y coseno están definidas sobre todo $\mathbb{R}$ y tienen periodo 2π. La representación de Fourier se aplica a las funciones de cuadrado-integrable. Específicamente decimos que la función *f* pertenece a un espacio de cuadrado-integrable $L^2[a,b]$ si

$$\int_a^b f^2(x)dx < \infty$$

El resultado de Fourier establece que cualquier función $f \in L^2[-\pi,\pi]$ puede ser expresada como una suma infinita de funciones dilatadas seno y coseno [Odgen 1997].

$$f(x) = \frac{1}{2}a_0 + \sum_{j=1}^{\infty}(a_j \cos(jx) + b_j \sin(jx)), \tag{2.1}$$

para un cálculo apropiado del conjunto de coeficientes $\{a_0, a_1, b_1, ...\}$ (coeficientes de Fourier). La sumatoria en (2.1) es hasta infinito, pero una función puede ser bien aproximada (en el sentido de L^2) mediante una sumatoria finita con límite superior *J*:

$$S_J(x) = \frac{1}{2}a_0 + \sum_{j=1}^{J}(a_j \cos(jx) + b_j \sin(jx)). \tag{2.2}$$

Esta representación en series de Fourier es extremadamente útil ya que, cualquier función en L^2 puede ser escrita en términos de funciones de bloques de construcción muy simples: senos y cósenos. Esto es debido al hecho que el conjunto de funciones $\{\sin(j\cdot), \cos(j\cdot), j = 1, 2, ...\}$, junto con la función constante, forman una base para el espacio de funciones $L^2[-\pi, \pi]$ [Odgen 1997].

Como un ejemplo, mostraremos el desarrollo en serie de Fourier de la siguiente función

$$f(x) = \begin{cases} x + \pi, & -\pi \le x \le -\pi/2 \\ \pi/2, & -\pi/2 < x \le \pi/2 \\ \pi - x, & \pi/2 < x \le \pi \end{cases} \tag{2.3}$$

Serie de fourier con límite superior infinito de *f*(*x*).

$$S(x) = \frac{3\pi}{16} + \frac{4\sum_{J=1}^{\infty} \frac{\cos\left(\frac{\pi J}{4}\right)\cos\left(\frac{Jx}{2}\right) - \cos\left(\frac{\pi J}{2}\right)\cos\left(\frac{Jx}{2}\right)}{J^2}}{\pi} \tag{2.4}$$

En la práctica no es necesario calcular hasta infinito como se ha mencionado, aquí se muestra un ejemplo de cuatro aproximaciones *J*=1, *J*=3, *J*=10, *J*=20.

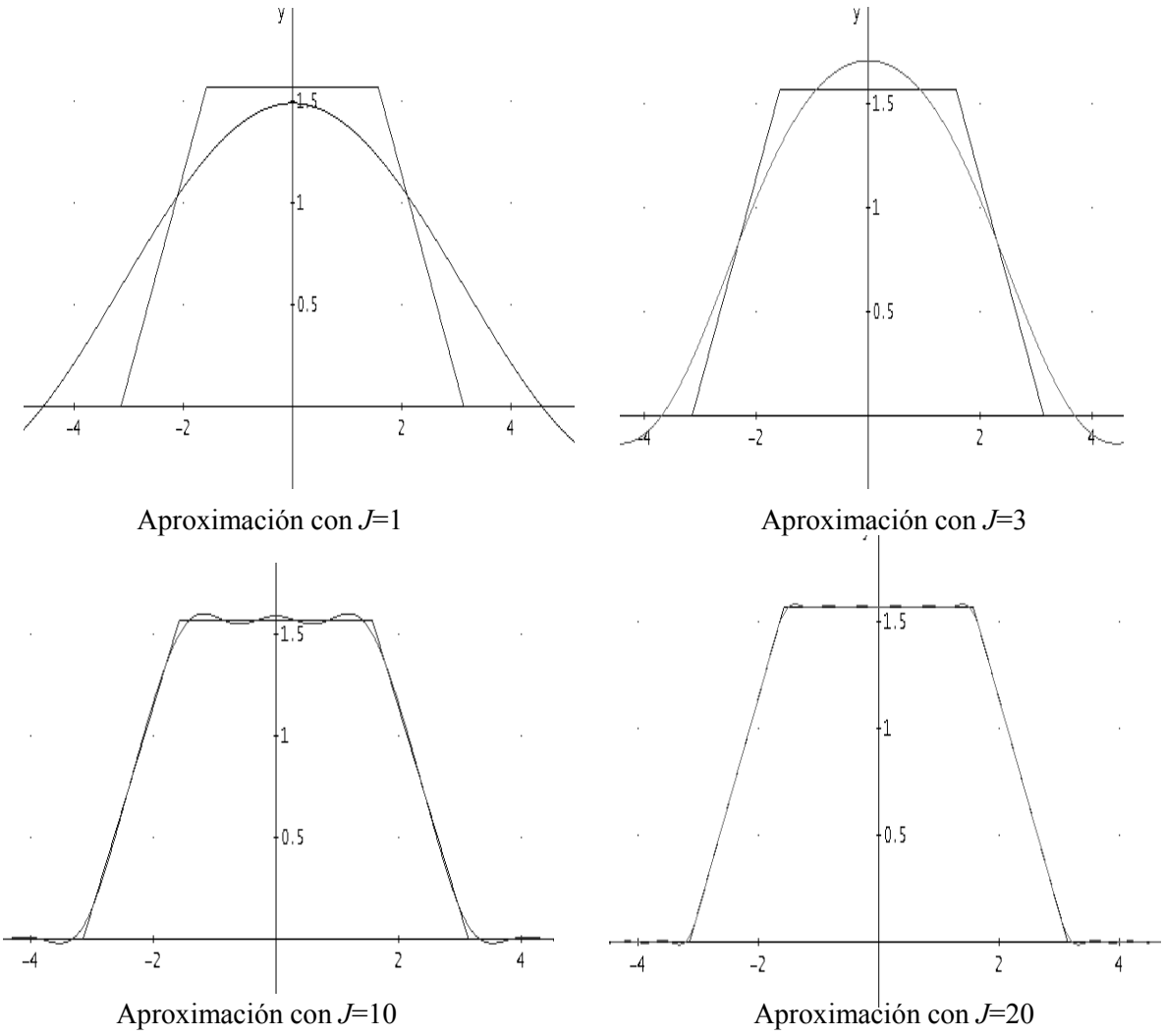


Figura 2.9. Una función ejemplo y sus representaciones en sumas de Fourier

En este ejemplo (Figura 2.9), usando 4 pares de funciones bases en la reconstrucción, da una representación bastante buena, aun así esta representación puede ser mejorada mediante el incremento de *J*. El siguiente asunto a considerar es el cálculo de los coeficientes $\{a_0, a_1, b_1, a_2, b_2, ...\}$. Para $j \geq 1$, los coeficientes de Fourier pueden ser calculados tomando el producto interno de la función f y las correspondientes funciones base:

$$a_j = \frac{1}{\pi}\langle f, \cos(j\cdot)\rangle = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(jx)dx, j = 0,1,..., \quad (2.5)$$

$$b_j = \frac{1}{\pi}\langle f, sen(j\cdot)\rangle = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)sen(jx)dx, j = 1,2,.... \quad (2.6)$$

Los coeficientes en (2.5) y (2.6) están dados en términos del *producto interno* entre dos funciones en L^2:

$$\langle f, g \rangle = \int f(x) g(x) dx,$$

donde la integral es tomada sobre el conjunto apropiado de $\mathbb{R}$. La norma L^2 de una función es definida por:

$$\|f\| = \sqrt{\langle f, f \rangle} = \sqrt{\int f(x) dx}.$$

La representación en (2.1) se aplica uniformemente para toda $x \in [-\pi, \pi]$ bajo ciertas restricciones sobre *f*. La función de ejemplo de la Figura 2.9 tiene discontinuidades en su derivada, pero la representación de Fourier converge en todos los puntos. Para cualquier función en $L^2[-\pi, \pi]$, la representación truncada (Figura 2.1) converge en el sentido L^2:

$$\|f - S_j\|^2 \to 0$$

cuando $J \to \infty$. En términos prácticos, esto significa que muchas funciones pueden ser descritas usando solamente pocos coeficientes.

Definición 1 Dos funciones $f_1, f_2 \in L^2[a, b]$ son *ortogonales* sí $\langle f_1, f_2 \rangle = 0$.

Definición 2 *Una secuencia de funciones* $\{f_j\}$ *es ortonormal si los* f_j*'s son ortogonales en pares y* $\|f_j\| = 1$ *para toda j.*

El requerimiento de ortogonalidad es cubierto con las funciones seno y coseno. Normalizando la base de esta manera permite escribir la representación (2.1) con las expresiones para cálculo de los coeficientes (2.5) y (2.6) como:

$$f(x) = \langle f, h_0 \rangle h_0(x) + \sum_{j=1}^{\infty} \left(\langle f, g_j \rangle g_j(x) + \langle f, h_j \rangle h_j(x) \right)$$

Definición 3 *Decimos que una secuencia de funciones* $\{f_j\}$ *es un sistema ortonormal completo (CONS) si las* f_j*'s son ortogonales por pares,* $\|f_j\| = 1$ *para cada j, y solamente la función ortogonal a cada* f_j *es la función cero.*

Así definido, el conjunto $\{h_0, g_j, h_j : j = 1, 2, ...\}$ es sistema *ortonormal* completo para $L^2[-\pi, \pi]$.

2.3.1. La Transformada de Fourier (FT)

Nosotros podemos pensar en una función no periódica derivada de una función periódica con un periodo que se extiende desde $-\infty$ hasta ∞. Entonces, para una señal que es una función de tiempo con un periodo de $-\infty$ hasta ∞, podemos formar la integral

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \tag{2.7}$$

y asumiendo que existe para cada valor de frecuencia en radianes ω, podemos llamar a la función $F(\omega)$ la *Transformada de Fourier* o la *Integral de Fourier* [Karris 2003].

La transformada de Fourier es, en general, una función compleja. Podemos expresar esta como la suma de su parte real e imaginaria o en forma exponencial

$$F(\omega) = Re\{F(\omega)\} + jIm\{F(\omega)\} = |F(\omega)| e^{j\varphi(\omega)}$$

La transformada inversa de Fourier es definida como

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \tag{2.8}$$

La importancia de este hecho es que podemos descomponer una señal (Análisis) y observar información en el dominio de la frecuencia, la cual no es posible observar en el dominio del tiempo y a su vez regresar (Síntesis) dicha señal al dominio del tiempo sin la perdida de información.

Supongamos que tenemos una señal (figura 2.10, señal sólida) y que para analizarla necesitamos $F(\pi)$, para lo cual la función de análisis $e^{-j\omega t}$ de la parte real (Figura 2.5, señal punteada) queda $Re\{e^{-j\pi t}\}$, por supuesto que esta representa una señal oscilante con frecuencia en radianes de $\omega = \pi$ [Stark 2005].

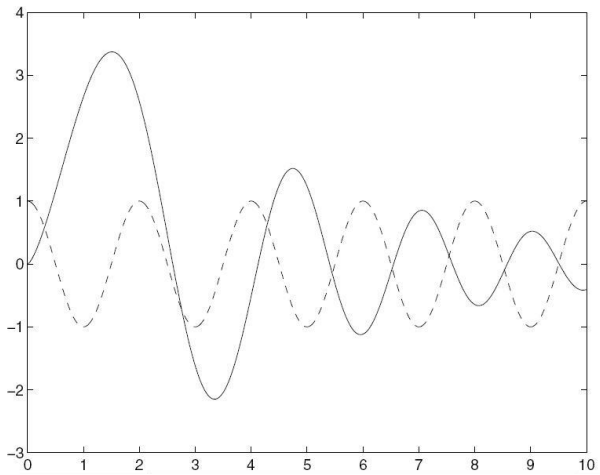


Figura 2.10. Señal (línea solida) y función de análisis (línea punteada)

¿Por qué mide $F(\pi)$ la aparición de $\omega = \pi$ en la señal? Esto se debe, a que si, en un intervalo de tiempo la señal oscila con una frecuencia en radianes de $\omega = \pi$, la señal y la función de análisis tienen una fase de movimiento mutua, por lo tanto, esto provee una contribución diferente de cero a $F(\pi)$.

Sin embargo, en la Transformada de Fourier, no es posible *localizar* (ubicar en el tiempo o espacio), sabemos que existe una contribución a $F(\pi)$ por parte de $\omega = \pi$, y lo sabemos si el valor de $F(\pi)$ en valor absoluto es grande, pero no sabemos donde apareció, ya que la función de análisis se extiende sobre todo el eje real. Por lo tanto, no es posible localizar en el dominio del tiempo dicha fase de movimiento mutua entre la señal y la función de análisis.

Para un manejo más práctico se estableció en términos discretos la Transformada de Fourier, llamada Transformada Discreta de Fourier

$$F_k = \sum_{n=0}^{N-1} f_n e^{-j(k\Delta\omega)(n\Delta t)} \tag{2.9}$$

En la práctica se utiliza un algoritmo numérico de la Transformada Discreta de Fourier, llamada Transformada Rápida de Fourier

$$X_k = \sum_{n=0}^{N-1} x(n) W_n^{nk} \tag{2.10}$$

2.3.2. La Transformada de Fourier de Tiempo Corto (STFT)

Algunos estudios requieren que sea posible localizar en el tiempo ciertas características encontradas en el dominio de la frecuencia, por lo cual fue necesario encontrar una técnica capaz de mostrar al mismo tiempo información del dominio del tiempo y el dominio de la frecuencia. Así Denis Gabor en 1946, tomó la Transformada de Fourier y la adaptó para analizar una pequeña sección de la señal en el tiempo, una técnica llamada *Windowing*. El windowing requiere el uso de una función ventana, la cual es, una función matemática que se utiliza cuando se requiere limitar en longitud una señal (la ventana rectangular es la más sencilla). La adaptación de Gabor es llamada *Transformada de Fourier de Tiempo Corto* (Short Time Fourier Transform, STFT) y por la técnica empleada, también se le conoce como *Windowed Fourier Transform*.

La STFT observa la aparición de una frecuencia en radianes ω en *un cierto tiempo t*. La función de análisis usada es la siguiente $e^{-j\omega t} w(u-t)$. Donde $w(u)$ es la función ventana. La expresión $w(u-t)$ indica que una función ventana se mueve a un tiempo t por lo regular centrada en 0. La transformación no solo depende de ω también de t y de la forma de la ventana utilizada [Stark 2005].

$$F_{\omega}(\omega,t)=\int_{-\infty}^{\infty} f(u)e^{-j\omega u}w(u-t)du \tag{2.11}$$

Por ejemplo, si tenemos una señal $f(u)$ a analizar, usando una ventana rectangular de tamaño 2, centrada en 0, y queremos analizar la aportación de $\omega=\pi$ en dicha señal, en el tiempo t=8, se observaría de la siguiente forma

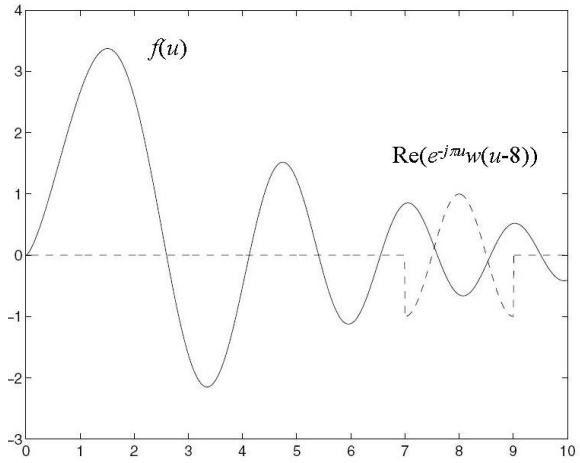


Figura 2.11. Señal (línea solida) y función de análisis (línea punteada) con ω=π

Aunque la función de análisis heredada de la FT se extiende sobre todo el eje real, solo observamos lo que permite la ventana, dando origen a la función de análisis de la STFT.

El análisis hecho por la transformación permite no solo observar información global de la aportación de $\omega=\pi$ a F_{ω}, también, como un extra, saber cuándo ha sucedido, esto es debido a que la función de análisis es fácilmente localizada en "el tiempo de análisis t" (Figura 2.11). Solo existe un pequeño inconveniente en el procedimiento, si necesitamos información más detallada de la señal alrededor de t=8, es necesario aumentar la frecuencia ω, por ejemplo a 6π (Figura 2.12), pero la ventana es de tamaño fijo y no adaptativa. Posiblemente solo una pequeña parte alrededor de t=8, sea requerida, el resto, aunque no interesa, será analizado.

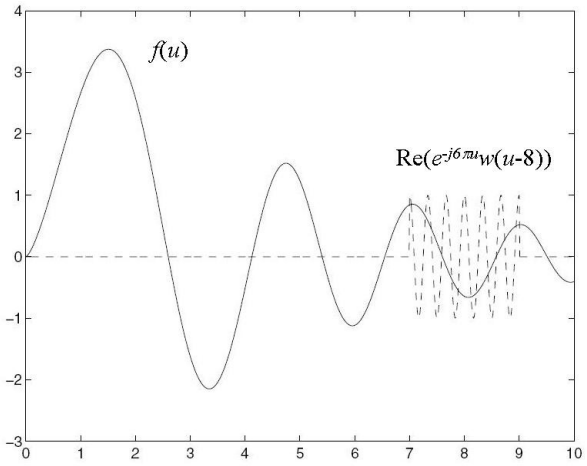


Figura 2.12. Señal (línea solida) y función de análisis (línea punteada) con $\omega=6\pi$

2.4. Introducción a Wavelets

El uso de wavelets hace su aparición formalmente en la última década del siglo XX, debido a los trabajos de Mallat y Daubechies, con fuertes aplicaciones en la ingeniería, la ciencia, las finanzas, el estudio del habla, el procesamiento de imágenes, el Internet, en fin un sin número de áreas que las utilizan.

La idea fundamental detrás de las wavelets es el analizar de acuerdo a la escala. Las wavelets son funciones que satisfacen ciertos requerimientos matemáticos y son usadas en representación de datos o de otras funciones. La idea no es nueva. La aproximación usando superposición de funciones tiene su origen en los inicios del siglo XIX, cuando Joseph Fourier descubrió que podía superponer senos y cosenos para representar otras funciones. Sin embargo en el análisis wavelet, la escala que usamos para ver los datos desempeña un rol especial. Los algoritmos wavelets procesan los datos en diferentes escalas o resoluciones. Si vemos una señal con una ventana grande notaríamos los rasgos totales. De forma similar si vemos una señal con una ventana pequeña notaríamos pequeños rasgos. El resultado en el análisis wavelet es ver ambos el bosque y los árboles, dicho de otra forma.

El procedimiento de análisis wavelet es para adoptar una función prototipo de la wavelet, llamada *analyzing wavelet* o *wavelet madre*. El análisis temporal es llevado a cabo con una versión contraída y de alta frecuencia del prototipo, mientras el análisis de frecuencia se lleva a

cabo con una versión dilatada y de baja frecuencia del prototipo, el análisis wavelet permite tener información tiempo-frecuencia.

Desde una perspectiva histórica las wavelets de tiempo-frecuencia tienen su origen en la década de 1940 con Denis Gabor y John Von Newman, tienen una historia relativamente larga en el procesamiento de señales. Muchas de las contribuciones fundamentales fueron conseguidas por físicos, las wavelets de tiempo-frecuencia se han utilizado mucho en el procesamiento del habla.

Pero en cambio las wavelets de tiempo-escala, tienen una historia más corta aplicadas al procesamiento de señales e imágenes. En la historia de las matemáticas, el análisis wavelet muestra muchos orígenes diferentes, mucho del trabajo fue llevado a cabo en la década de 1930, y con el tiempo, los esfuerzos separados no parecen ser parte de una teoría coherente. Si existe un orden por el cual se deba mencionar el trabajo que ha culminado en la teoría wavelet es el siguiente:

Antes de 1930 la principal rama de las matemáticas en dirección a los wavelets comenzó con Joseph Fourier (1807) el cual descubrió que cualquier función 2π periódica puede ser representada por sumas de senos y cósenos, después de Fourier los matemáticos descubrieron tres asuntos que podían estudiar, ellos podían cambiar la noción de función y adaptarla a las series de Fourier, podían modificar la definición de convergencia de las series y la búsqueda de sistemas ortogonales para los cuales no sucede la divergencia. Finalmente el mejor concepto funcional para las series de Fourier fue creado por Henri Lebesgue [Jaffard et al. 1962].

Debido a esto los matemáticos fueron cambiando sus nociones de frecuencia a nociones de escala. La primera mención de los wavelets aparece en un apéndice en la tesis de A. Haar (1909). El sistema de Haar es el primero de los sistemas de wavelets que tiene soporte compacto, lo cual significa que desaparece fuera de un intervalo finito. El escalamiento es llevado a cabo de forma diádica (potencias de dos). Desafortunadamente las funciones base de Haar no son continuamente diferenciables lo cual limita algo sus aplicaciones. Posteriormente de 1910 a 1920 Schauder, soluciona este problema con funciones base triangulares sobre un espacio Banach [Jaffard et al. 1962].

En los años de 1930, algunos grupos trabajando independientemente investigaron la representación de funciones usando *funciones base de escala variante.* Mediante el uso de la función base de escala variante de Haar, Paul Levy, un físico de los años de 1930, investigó el movimiento Browniano, un tipo de señal aleatoria. El encontró la función base de Haar superior a las funciones base de Fourier para estudiar pequeños detalles complicados en el movimiento Browniano. La wavelet de Lusin fue desarrollada en esa época, el trabajo de Lusin es un ejemplo de expansiones de wavelets continuas [Jaffard et al. 1962].

Entre 1960 y 1980 los matemáticos Guido Weiss y Ronald R. Coifman estudiaron los elementos más simples de un espacio de función, llamado *átomos*, con la meta de encontrar los átomos para una función común y encontrar las "reglas de montaje" que permite la reconstrucción de todos los elementos del espacio de función usando esos átomos. En 1980 Grossman y Morlet, un físico y un geofísico, definieron ampliamente las wavelets en el contexto de la física cuántica, se debe a Grossman y Morlet el uso de de la palabra "*Ondelette*" o wavelet y su primera definición formal [Jaffard et al. 1962].

En 1985 Stephane Mallat dio a las wavelets un salto y comienzo en su fundamento teórico, a través de su trabajo en procesamiento digital de señales. El descubrió las relaciones entre los filtros espejo en cuadratura, algoritmos piramidales y bases de wavelets ortonormales. Inspirado en parte por estos resultados Yves Meyer construye las primeras wavelets no triviales. A diferencia de las wavelets de Haar, las wavelets de Meyer son continuamente deferenciables; sin embargo estas no tienen soporte compacto. Un par de años después, Ingrid Daubechies uso el trabajo de Mallat para construir un conjunto de funciones de base ortonormal de wavelets que son tal vez las más elegantes, y se han convertido en la piedra angular de las aplicaciones wavelet de hoy en día [Jaffard et al. 1962].

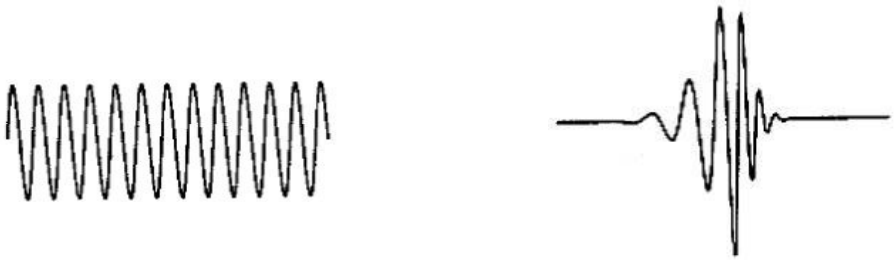
2.4.1. ¿Qué es una Wavelet?

La palabra francesa *Ondelette* fue introducida por primera vez en 1980 por Morlet y Grossman, básicamente quiere decir *onda pequeña*. Aunque, *Wavelet*, la versión en idioma inglés es más

usada en la actualidad, en español recibe varios nombres, como por ejemplo ondeleta, ondicula y ondita.

Una wavelet es una onda pequeña de duración finita, y que en promedio tiene un valor de cero. Comparando las wavelets con las ondas seno, que son la base del análisis de Fourier, los senos no tienen una duración finita, pues se extienden desde $-\infty$ hasta ∞. Además las ondas seno son suaves y predecibles, en cambio las wavelets son irregulares y asimétricas [Burrus et al 1997].

En primera instancia existen dos tipos de wavelet, la wavelet madre y la wavelet padre, que son usadas para el análisis mediante wavelets, análogamente a los senos y cósenos en el análisis de Fourier.



(a) Onda seno (b)Wavelet Daubechies Db10

Figura 2.13. Una onda y una wavelet

2.4.2. La Transformada Wavelet Continua (CWT)

La Transformada Wavelet Continua fue introducida en 1984 por Morlet y Grossman, utilizándola para analizar señales sísmicas, con algún tipo de modificación a la STFT de Gabor

$$F_{\omega}(\omega,t)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(u)e^{-j\omega u}w(u-t)du$$

Para $f\in L^2(\mathbb{R})$ una señal de tiempo continuo y $w\in L^2(\mathbb{R})$ una función ventana. Aquí $L^2(\mathbb{R})$ denota el espacio de funciones de cuadrado integrable sobre $\mathbb{R}$. La modificación a la

STFT fue la siguiente: la función de análisis ($e^{-j\omega u}$) y la función ventana ($w(u-t)$) son sustituidas por una función ψ que puede ser cambiada de escala

$$W_{\psi}(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(u)\psi\left(\frac{t-b}{a}\right)dt \tag{2.12}$$

para alguna función wavelet $\psi \in L^2(\mathbb{R})$, el parámetro de escala $a > 0$ y el parámetro de traslación $b \in \mathbb{R}$. Aunque esto parece ser una nueva transformada, es conocida actualmente como la fórmula de reproducción de Caldéron [Jansen y Oonincx 2005].

La transformada Wavelet tiene la propiedad de actuar como microscopio. Pero en contraste con la de Fourier no observa el aporte de la frecuencia en radianes, en vez de eso observa el tamaño de los detalles de *a* en cierto tiempo *b*. En lugar de tamaño de los detalles podemos hablar de factores de escala, existe una relación entre estos y la frecuencia, es decir la frecuencia y el tamaño de los detalles son inversamente proporcionales, a mayor frecuencia menor será el tamaño de detalle y viceversa [Stark 2005]. Existe una constante β tal que

$$a = \frac{\beta}{\omega}$$

Además la función de análisis ψ es llamada wavelet madre y da origen, mediante el uso del factor de escala *a* y traslación *b*, a una familia de funciones wavelet. Una función wavelet de dicha familia generada por la wavelet madre quedaría de la siguiente manera

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{2.13}$$

2.4.3. La Transformada Wavelet Discreta (DWT)

Con el fin de optimizar el análisis de señales continuas surgió la llamada Transformada Wavelet Discreta (DWT), la cual discretiza la función wavelet.

Sabemos que una función wavelet continua es de la siguiente forma

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$$

su continuidad radica en que *a* y *b* cambian en forma continua, dado que se desea hacer cálculos computacionales es necesario discretizar dicha función.

Una forma de discretizar, es muestrear los parámetros *a* y *b*, usando una discretización logarítmica de *a*, por lo cual se debe conectar esto a un número de saltos tomados de *b* localizaciones, es decir, para conectar *b* con *a* es necesario dar un número de saltos *b* los cuales son proporcionales a la escala *a.* de tal forma que una función wavelet discretizada queda de la siguiente forma

$$\psi_{mn} = \frac{1}{\sqrt{a_0^m}}\psi\left(\frac{t-nb_0a_0^m}{a_0^m}\right) \tag{2.14}$$

Donde *m* y *n*, controlan el escalado y traslación (saltos) respectivamente, a_0 es un parámetro de escala fijo que debe ser mayor a 1 y b_0 es el parámetro de traslación, el cual debe ser mayor que 0 [Addison 2002].

Existe otro tipo de discretización llamado *dyadic grid arrangement*, se debe a que los parámetros elegidos para a_0 y b_0 son 2 y 1 respectivamente, lo que hace que la discretización logarítmica sea de potencias de 2, esta es la más eficiente y permite la construcción de bases de wavelets ortogonales. El término diádico, indica que se usan potencias de 2 para escalamiento y traslación. Sustituyendo a_0 y b_0, con 2 y 1 queda

$$\psi_{mn} = \frac{1}{\sqrt{2^m}}\psi\left(\frac{t-n2^m}{2^m}\right)$$

y en forma más compacta

$$\psi_{mn} = 2^{-m/2}\psi\left(2^{-m}t - n\right) \tag{2.15}$$

2.4.4. El Análisis Multiresolución (MRA) y la Transformada Rápida Wavelet

Como se vio en la sección anterior se puede discretizar la función wavelet, esto permite aproximar señales continuas y discretas con una función wavelet discreta. La descomposición para las señales discretas tiene que estar dada en función del número de puntos *N*, el cual debe ser potencia de dos.

Es posible aproximar con una wavelet una función mediante dilataciones y traslaciones diádicas hasta un número de *j* suficientemente bueno, pero resulta mucho mejor aproximar dicha función f mediante el uso de 2 funciones de análisis, las cuales son la base de un sistema ortonormal, asociadas a un análisis multiresolución. Estas son conocidas como *función de escala* y *función wavelet*. La *función de escala* o también llamada *wavelet padre* está dada por

$$\phi(t) = \sum_k h_k (2t - k) \tag{2.16}$$

donde h_k representa un conjunto de *coeficientes filtro* o *máscara*. La función wavelet o wavelet madre es generada de la función de escala y está dada por

$$\psi(t) = \sum_k g_k (2t - k) \tag{2.17}$$

donde g_k representa los coeficientes filtro o máscara producto de $g_k = (-1)^k h_{1-k}$. Este tipo de wavelets son conocidas como *wavelets de primera generación*. De las wavelets de primera generación más conocidas está la familia Daubechies. Estas bases llamadas ortonormales generan un conjunto de wavelets mediante traslaciones y dilataciones diádicas de las dos funciones no solo de la función wavelet

$$\phi_{j,k} = 2^{j/2}\phi(2^j t - k),$$
$$\psi_{j,k} = 2^{j/2}\psi(2^j t - k)$$

Es posible asociar estas funciones mediante el análisis multiresolución de Mallat. El introdujo las bases de wavelets ortonormales como una descomposición en $L^2(\mathbb{R})$ de una sucesión de espacios encajados $\{V_j, j \in \mathbb{Z}\}$ tal que

$$\ldots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots$$

donde $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$ (El espacio $L^2(\mathbb{R})$ es la cerradura de todos los subespacios V_j) y $\bigcap_{j\in\mathbb{Z}} V_j = \varnothing$ (La intersección de todos los V_j es vacía). La multiresolución es reflejada por el siguiente requerimiento $f \in V_j \Leftrightarrow f(2t) \in V_{j+1}, j \in \mathbb{Z}$, esto es equivalente a $f(t) \in V_0 \Leftrightarrow f(2^j t) \in V_j$, todos los espacios son versiones escaladas de un espacio de referencia V_0 [Li et al 2002].

La wavelet padre se relaciona con el análisis multiresolución dado que, puede generar fácilmente una secuencia de subespacios. Primero todas las traslaciones $\phi(t)$, por ejemplo $\phi(t-k), k \in \mathbb{Z}$ expanden un subespacio llamado V_0. De la misma manera $2^{1/2}\phi(2t-k), k \in \mathbb{Z}$ expande otro subespacio llamado V_1. Esto implica que ϕ cae dentro de V_1 y sus traslaciones $\phi(t-k), k \in \mathbb{Z}$ también caen dentro de V_1. Por lo tanto V_0 esta encajado dentro de V_1.

Pero, ¿qué es lo que pasa en el espacio complemento ortogonal de V_0 en V_1 llamado W_0?, este espacio está relacionado con la wavelet madre por las siguientes razones: se puede observar que ψ cae dentro de V_1, también sus traslaciones $\psi(t-k)$. Además que ψ es ortogonal a ϕ, por lo cual una traslación de ϕ es ortogonal a una traslación de ψ. Así las traslaciones de ψ expanden el espacio complemento W_0. De forma similar una $j, \psi_{j.k}, k \in \mathbb{Z}$ expande un espacio W_j que es el complemento ortogonal del espacio V_j en V_{j+1} [Li et al. 2002], así

$$V_{j+1} = V_j \oplus W_j$$

De forma similar

$$V_1 = V_0 \oplus W_0$$

Lo cual se extiende a

$$V_2 = V_0 \oplus W_0 \oplus W_1$$

Y en general

$$L^2(\mathbb{R}) = V_0 \oplus W_0 \oplus W_1 \oplus W_2 \ldots$$

$$L^2(\mathbb{R}) = V_0 \oplus \bigoplus_{j \geq 0} W_j \tag{2.18}$$

V_0 Puede ser representado por [Burrus et al. 1997]

$$V_0 = W_{-\infty} \oplus \cdots \oplus W_{-1}$$

De tal forma que

$$L^2(\mathbb{R}) = \bigoplus_{j \in \mathbb{Z}} W_j \tag{2.19}$$

Por lo tanto $L^2(\mathbb{R})$ es descompuesto en una secuencia infinita de espacios. Así, una función $f(t)$ puede ser representada según la expresión (2.18) como

$$f(t) = \sum_{k \in \mathbb{Z}} a_{0,k} \phi_{0,k}(t) + \sum_{j=0}^{\infty} \sum_{k \in \mathbb{Z}} d_{j,k} \psi_{j,k}(t) \tag{2.20}$$

donde

$$a_{j,k} = \; < f(t), \phi_{0,k} > \; = \int f(t) \phi_{0,k}(t) dt$$

$$d_{j,k} = \; < f(t), \psi_{j,k} > \; = \int f(t) \psi_{j,k}(t) dt$$

Se puede ver (2.20) como una aproximación de *f* en una escala *j*=0 (primer término de 2.20), además de información extra a una escala más fina (segundo término de 2.20). Una aplicación del análisis multiresolución es la transformada rápida wavelet o también conocido como *algoritmo piramidal* de Mallat [Li et al. 2002].

Para ver cómo funciona el algoritmo piramidal, observemos que dada una función $f \in L^2(\mathbb{R})$ podemos encontrar una *J* tal que $f_J \in V_J$ se aproxime a f hasta una precisión definida. Si $g_i \in W_i$, $f_i \in V_i$, entonces $f_J = f_{J-1} + g_{J-1} = \sum_{i=1}^{M} g_{J-1} + f_{J-M}$. Así el proceso es como sigue: dada una $f_J \in V_J$, se descompone f_n en dos partes donde una parte es V_{J-1} y la otra es W_{J-1}, el siguiente paso será descomponer V_{J-1} obtenido en paso previo, en dos partes una V_{J-2} y la otra W_{J-2}, este procedimiento es repetido. Recordemos que $\phi(t) = \sum_{-\infty}^{\infty} h_k \phi(2t-k)$ y que $\psi(t) = \sum_{-\infty}^{\infty} g_k \phi(2t-k)$, donde $g_k = (-1)^k h_{1-k}$. Además, que la wavelet padre o función de escala está relacionada con el espacio V_0 y que la función wavelet o wavelet madre está relacionada con el espacio complemento W_0 [Li et al. 2002].

Desde el punto de vista del procesamiento digital de señales los coeficientes filtro o máscara $\{h_k\}$ y $\{g_k\}$ son llamados *filtros espejo en cuadratura* (Quadrature Mirror Filters, QMF), donde h_k representa un filtro pasa bajos y g_k un filtro pasa altos, de tal manera que $\{h_k\}$ y $\{g_k\}$, son representados por el par filtro (*H*, *G*) [Stark 2005], así, para señales discretas, elegimos un tamaño de *J*, el cual representa la resolución de la descomposición, relacionada con los puntos *N* de la señal que deben ser potencia de 2, de tal manera que $N = 2^J$. Así, comenzando de f, las señales a^1 y d^1 son generadas mediante la aplicación del par filtro (*H*, *G*) a *f,* procediendo de la misma manera se aplica el par filtro a la señal a a^1, obteniendo las señales a^2 y d^2, y así, sucesivamente se aplica el par filtro a $a^2, a^3, ..., a^{J-1}$, y finalmente se tendrá la señal descompuesta después de *J* pasos [Stark 2005].

$$\begin{array}{l} f \xrightarrow{G} d^1 \\ H\downarrow \\ a^1 \xrightarrow{G} d^2 \\ H\downarrow \\ a^2 \xrightarrow{G} d^3 \\ H\downarrow \\ a^3 \\ \vdots \\ a^{J-1} \xrightarrow{G} d^J \\ H\downarrow \\ a^J \end{array} \tag{2.21}$$

Usualmente a^J es llamada función de aproximación, y los restantes d^j (j=1, 2, 3,…, J) como señales de detalles. Así, a^J es obtenida de la aplicación sucesiva de un filtro pasa bajos, mientras que para las señales de detalles son la aplicación de un filtro pasa altos directo.

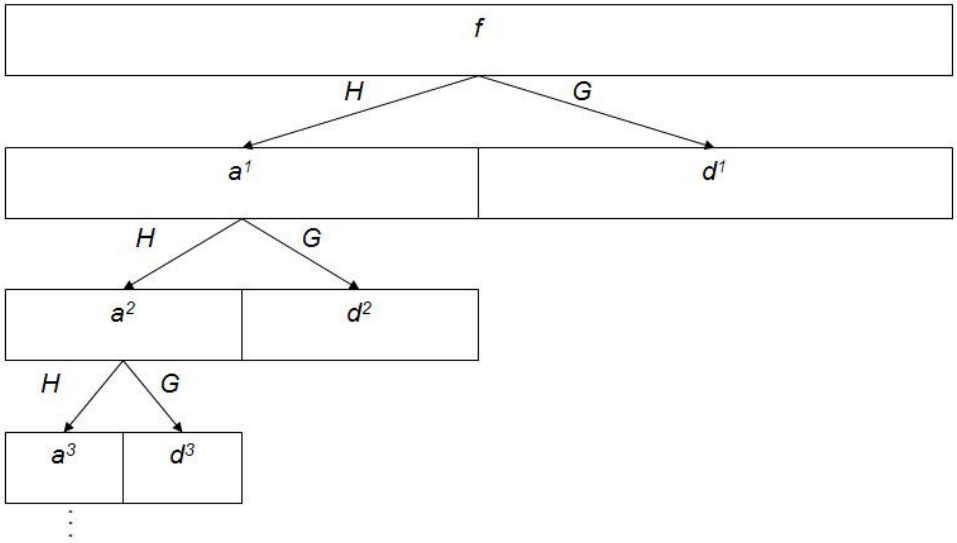


Figura 2.14. Se muestra la descomposición DWT de J pasos o algoritmo piramidal

Se debe recordar que las señales a^j y d^j son el producto de la expansión de ϕ y ψ sobre f respectivamente, mediante traslaciones diádicas.

2.4.5. Haar y Db2 Parte de Una Gran Familia

Por lo que se vio anteriormente solo queda elegir el tipo de wavelet que se usará en el algoritmo de descomposición, existen una variedad de wavelets, y entre ellas está la wavelet más sencilla, pero no menos importante, conocida como wavelet de Haar. La función de Haar es una auténtica wavelet, aunque no es usada mucho en la práctica. La wavelet de Haar no es nueva, habiendo sido desarrollada en 1910, mucho tiempo después todo mundo comenzó a hablar de "wavelets".

La wavelet padre o función de escala (2.16) para Haar está formada por dos coeficientes distintos de cero $h_0 = h_1 = 1$

$$\phi(t) = \phi(2t) + \phi(2t-1) \tag{2.22}$$

La solución de la función de escala es un pulso cuadrado

$$\phi(t) = \begin{cases} 1, & 0 \le t < 1 \\ 0, & \text{en otra parte} \end{cases} \tag{2.23}$$

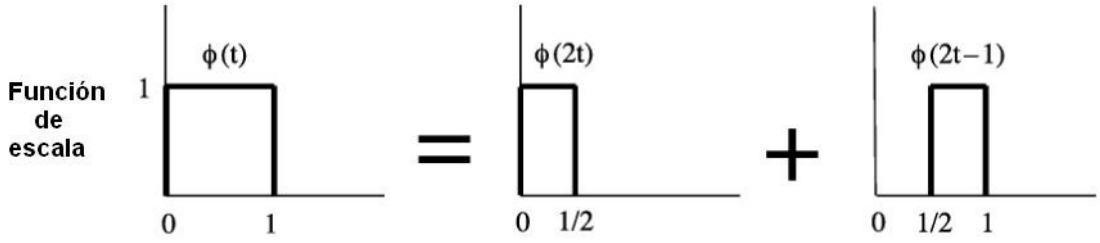


Figura 2.15. Función de escala o wavelet padre de Haar $\phi(t)$

Los coeficientes h_0 y h_1 definen la ecuación de refinamiento o wavelet madre, mediante la fórmula

$$g_k = (-1)^k h_{1-k} \tag{2.24}$$

de esta forma, los coeficientes son $g_0 = 1$ y $g_1 = -1$

$$\psi(t) = \phi(2t) - \phi(2t-1) \tag{2.25}$$

La solución para esta función es una onda cuadrada

$$\psi(t) = \begin{cases} 1, & 0 \le t < \frac{1}{2} \\ -1, & \frac{1}{2} \le t < 1 \\ 0, \text{ en otro parte} \end{cases} \tag{2.26}$$

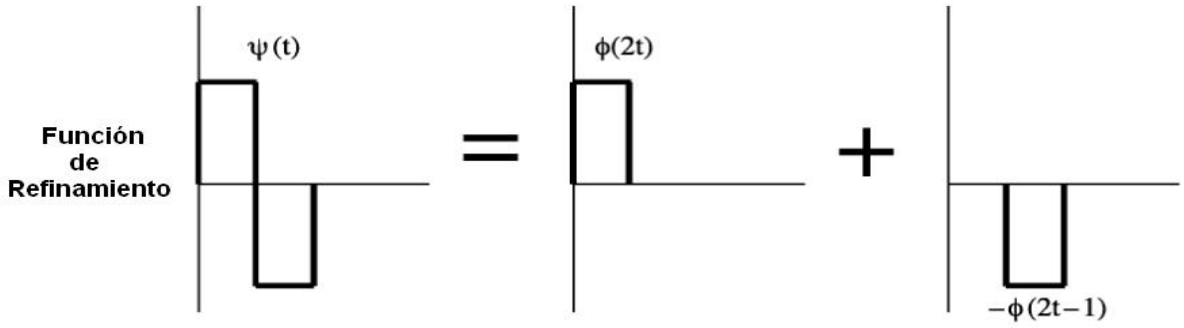


Figura 2.16. Función de refinamiento o wavelet madre de Haar $\psi(t)$

La wavelet de Haar es la primera de una gran familia de wavelets ortonormales que fueron desarrolladas por Ingrid Daubechies [Daubechies 1992], conocida como la familia Dbn, donde cada miembro de esta familia posee un numero natural *n*=1, 2, 3,… y a la vez cada miembro cuenta con 2*n* coeficientes filtro [Stark 2005]. Aunque la wavelet de Haar ya existía, Daubechies se inspiró en sus propiedades y en el trabajo de Mallat para desarrollar la familia Dbn. De esta forma la wavelet de Haar es conocida como Db1, la cual posee dos coeficientes filtro.

La siguiente wavelet de esta familia es la Db2, y está formada por cuatro coeficientes filtro. Para lograr obtener estos coeficientes en necesario el planeamiento de un conjunto de

ecuaciones no lineales y al resolver dicho sistema se obtienen los coeficientes filtro de la función de escala y mediante ellos se obtienen los coeficientes filtro de la función de refinamiento. Los coeficientes filtro para la función de escala Db2 son $h_0 = \frac{1+\sqrt{3}}{4}$, $h_1 = \frac{3+\sqrt{3}}{4}$, $h_2 = \frac{3-\sqrt{3}}{4}$, $h_3 = \frac{1-\sqrt{3}}{4}$ y para la función de refinamiento son $g_0 = \frac{1+\sqrt{3}}{4}$, $g_1 = -\frac{3+\sqrt{3}}{4}$, $g_2 = \frac{3-\sqrt{3}}{4}$, $g_3 = -\frac{1-\sqrt{3}}{4}$, (Ver anexo A). Las ecuaciones de escala y refinamiento son las siguientes:

$$\phi(t) = \frac{1+\sqrt{3}}{4}\phi(2t) + \frac{3+\sqrt{3}}{4}\phi(2t-1) + \frac{3-\sqrt{3}}{4}\phi(2t-2) + \frac{1-\sqrt{3}}{4}\phi(2t-3) \tag{2.27}$$

$$\psi(t) = \frac{1+\sqrt{3}}{4}\phi(2t) - \frac{3+\sqrt{3}}{4}\phi(2t-1) + \frac{3-\sqrt{3}}{4}\phi(2t-2) - \frac{1-\sqrt{3}}{4}\phi(2t-3) \tag{2.28}$$

Las soluciones para estas ecuaciones están dadas por las siguientes wavelets

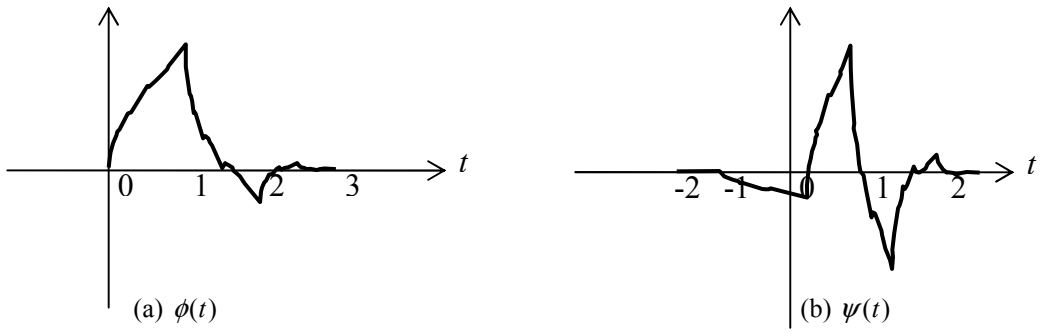


Figura 2.17. Función de escala (Wavelet Padre) y función de refinamiento (Wavelet Madre) de Db2

Pero aun falta un último paso para tener los coeficientes listos para el MRA. Entre las ecuaciones que forman el sistema a resolver, de donde se obtienen los coeficientes, se encuentra una función de estabilidad, la cual asegura que la función de escala conserve su área bajo cada iteración y está dada por

$$\sum_{k=0}^{N-1} h_k = 2 \tag{2.29a}$$

donde *N* representa los 2*n* coeficientes filtro, claramente al sumar los coeficientes de la función de escala de Haar se cumple que $h_0 + h_1 = 2$ y Db2 también cumple dicha condición dado que $h_0 + h_1 + h_2 + h_3 = 2$. Por otro lado se exige una normalización de dicha condición [Daubechies 1992] que obliga a que (2.29a) este dada por

$$Nor\left(\sum_{k=0}^{N-1} h_k\right) = \sqrt{2} \tag{2.29b}$$

lo cual es equivalente a

$$\frac{1}{\sqrt{2}} \sum_{k=0}^{N-1} h_k = \sqrt{2} \tag{2.30}$$

La normalización implica que todo coeficiente h_k sea dividido por $\sqrt{2}$, por lo cual, para la función de escala de Haar los coeficientes filtro normalizados son $h_0 = \frac{1}{\sqrt{2}}$, $h_1 = \frac{1}{\sqrt{2}}$ y deduciendo también obtenemos los de la función de refinamiento $g_0 = \frac{1}{\sqrt{2}}$, $g_1 = -\frac{1}{\sqrt{2}}$. Y su par de funciones (escala y refinamiento) están dadas por

$$\phi(t) = \frac{1}{\sqrt{2}} \phi(2t) + \frac{1}{\sqrt{2}} \phi(2t-1) \tag{2.31}$$

$$\psi(t) = \frac{1}{\sqrt{2}} \phi(2t) - \frac{1}{\sqrt{2}} \phi(2t-1) \tag{2.32}$$

De la misma forma se procede con los coeficientes de Db2, para la función de escala $h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}$, $h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}$, $h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}$, $h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}$ y para la función de refinamiento $g_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}$, $g_1 = -\frac{3+\sqrt{3}}{4\sqrt{2}}$, $g_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}$, $g_3 = -\frac{1-\sqrt{3}}{4\sqrt{2}}$.

Sus funciones están dadas por

$$\phi(t) = \frac{1+\sqrt{3}}{4\sqrt{2}}\phi(2t) + \frac{3+\sqrt{3}}{4\sqrt{2}}\phi(2t-1) + \frac{3-\sqrt{3}}{4\sqrt{2}}\phi(2t-2) + \frac{1-\sqrt{3}}{4\sqrt{2}}\phi(2t-3) \quad (2.33)$$

$$\psi(t) = \frac{1+\sqrt{3}}{4\sqrt{2}}\phi(2t) - \frac{3+\sqrt{3}}{4\sqrt{2}}\phi(2t-1) + \frac{3-\sqrt{3}}{4\sqrt{2}}\phi(2t-2) - \frac{1-\sqrt{3}}{4\sqrt{2}}\phi(2t-3) \quad (2.34)$$

Los coeficientes normalizados son utilizados en el algoritmo de descomposición piramidal o transformada rápida wavelet, los cuales conforman los *filtros espejo en cuadratura* o *par filtro* (*H*, *G*) ya mencionados en el apartado 2.4.4.

Para concluir, Db1 y Db2 forman parte de una gran familia la ya mencionada Dbn, la elección de las wavelet depende de la aplicación, y los recursos computacionales disponibles. Por ejemplo Db2, necesita menos recursos que Db10, pero Db10 cuenta con otras características que no tiene Db2. El algoritmo de descomposición mediante Db2 es mostrado en el Anexo B de este documento.

2.5. Reducción de la Dimensionalidad y Minería de Datos

Una vez lograda la aplicación de toda la teoría anterior, nos enfrentamos a un problema con los resultados de la transformada, el manejo de grandes cantidades de datos (en este caso señales de voz mapeadas), que incide en el uso de grandes recursos de memoria primaria, secundaria y de procesamiento. Este problema es conocido como la dimensionalidad de los datos y en este caso el manejo de altas dimensiones de datos o como algunos mencionan *la maldición de la dimensionalidad,* una frase acuñada por Richard Bellman, debido a la dificultad de optimizar en espacios de productos cartesianos, mediante enumeración exhaustiva (búsqueda exhaustiva) [Donoho 2000].

La disciplina encargada de esta cuestión es el *Análisis de datos*, se encarga de observar y resumir datos con el fin de extraer información útil y desarrollar conclusiones. Nació como parte de la estadística. John Wilder Tukey profesor emérito de Princeton, vislumbro al Análisis de Datos separado de la Estadística, en los años de 1960, Tukey fue el fundador de las estadísticas robustas, el procesamiento de señales no lineales y las transformadas rápidas de Fourier.

En 1962 publicó "The Future of Data Analysis" donde expuso su visión. Además, causó polémica con la publicación "Data Analysis, including Statistics" junto a Fred Mosteller, donde mencionaba que el análisis de datos había sido un campo potencialmente enorme en el cual la estadística, junto con la teoría de la probabilidad y la teoría de decisiones la habían fijado a un pequeño segmento.

Tukey vio cristalizada su visión años más tarde creándose grandes comunidades de estadísticos académicos e industriales, las cuales enfatizan el uso de análisis de datos sobre el análisis matemático y la prueba [Donoho 2000]. Sin embargo Donoho menciona que debe ser reconsiderada la conexión del análisis de datos con las matemáticas, ya que en la actualidad son requeridas nuevas técnicas para manejo de la dimensionalidad, y estas requieren de una ciencia básica, las matemáticas.

Existe una disciplina más que se encarga del manejo de grandes cantidades de datos y es llamada Minería de datos (Data Mining). La Minería de Datos se centra en la búsqueda de patrones interesantes y regularidades importantes en grandes bases de datos [Fayad et al 1996]. Esta disciplina utiliza técnicas de análisis de datos, que le permiten extraer patrones, tendencias y regularidades para entender mejor los datos, predecir comportamientos futuros. La minería de datos se diferencia del análisis de datos, en que no transforma y facilita el acceso a la información para que sea analizada más fácilmente. La minería de datos "analiza" los datos.

El presente trabajo utiliza una técnica de minería de datos llamada Ganancia de Información, la cual proviene de la teoría de la Información, la cal será descrita más adelante.

2.5.1. Formalización de la Reducción de la Dimensionalidad

Para lograr el manejo de la dimensionalidad de los datos, es necesario usar la llamada *reducción de la dimensionalidad*, se puede decir que esta busca una representación de baja dimensionalidad que capture el contenido esencial de los datos originales, mediante algún criterio que determine

cuales son estos contenidos esenciales [Fodor 2002]. Al conjunto de elementos multivariado[2] que forman los vectores o matrices de datos se les suele llamar: variables, características o atributos.

La reducción de la dimensionalidad puede ser vista como la transformación de los atributos que residen en un espacio de dimensión alta a un sub-espacio de dimensionalidad menor, de tal manera que la transformación asegure la máxima preservación posible de la información (error mínimo de reconstrucción) [Bharat 2006].

Los problemas de reducción de la dimensionalidad puede ser formulados de la siguiente manera: Sea $X = \{x_1, x_2, x_3..., x_n\}$ un conjunto de *n* datos en un espacio *d-dimensional*, es decir, $x_i \in \mathbb{R}^d$, entonces una técnica de reducción de dimensionalidad trata de encontrar un conjunto de salida correspondiente a patrones $Y = \{y_1, y_2, y_3, ..., y_n\}$ tal que , $y_i \in \mathbb{R}^m$, donde $m \ll d$ y Y proporciona la representación más fiel en un espacio de dimensionalidad menor [Bharat 2006].

2.5.2. Técnicas de reducción de la Dimensionalidad

Una técnica clásica consiste en usar medidas estadísticas, tales como la media, la varianza, la desviación estándar, máximo, mínimo entre otras, pero existen técnicas más especializadas divididas en dos grupos: *lineales y no lineales*. Las técnicas lineales son aquellas que utilizan combinaciones lineales de los atributos para generar la representación de baja dimensionalidad y las no lineales usan obviamente transformaciones no lineales, las cuales son más complicadas que las lineales [Fodor 2002]. Otra clasificación para estas técnicas es la que las divide en técnicas de *Selección de características* y *Extracción de características*.

La selección de características o atributos, tiene por objetivo el encontrar un subconjunto de variables del conjunto de variables originales, las cuales capturan el contenido del conjunto original. Existen dos formar de aplicarlo como filtro o *wrapper*. Algunos ejemplos de estas técnicas en el modo de Wrapper

- Búsqueda exhaustiva
- Primero mejor (Best First)

[2] Elemento de más de una variable producto de una medición de una *unidad experimental* (la señal de habla)

- Recocido simulado
- Algoritmos genéticos
- Selección voraz hacia adelante (Greedy Forward Selection)
- Selección voraz hacia atrás (Greedy Backward Elimination)

y como filtro

- Ganancia de información
- Algoritmo Relief
- Tasa de información

Varios de estos métodos son descritos en [Hall y Holmes 2003].

La extracción de características o atributos, tiene por objetivo encontrar un subconjunto de atributos que describan al conjunto original, producto de combinaciones lineales o no lineales, del conjunto original. Algunas de las técnicas que caen en esta categoría son:

- Análisis de componentes principales (PCA)
- Reducción de la dimensionalidad multifactorial
- Análisis de Factores (FA)
- Análisis de Factores principales (PFA)
- Análisis de Factores de Máxima Verosimilitud
- Análisis de componentes independientes, entre otros

En [Fodor 2002] se puede observar estas y otras técnicas de extracción de atributos.

2.5.3. Ganancia de Información

En el presente trabajo se hace uso de la técnica de selección de atributos llamada ganancia de información (Gain information). Dicha técnica, es muy usada en aplicaciones de categorización de textos. Además, la ganancia de información y su modificación llamada tasa de información, son la base de los árboles de decisión (ID3, C45 o J48), se basa en la entropía, la cual puede

entenderse como el grado de desorden de la información o variables. Así, si A es un atributo y C es la clase de ese atributo, las ecuaciones (2.35) y (2.36), definen la entropía de la clase y la entropía de la clase dado el atributo A

$$H(C) = -\sum_{c \in C} p(c) \log_2 p(c) \tag{2.35}$$

$$H(C \mid A) = -\sum_{a \in A} p(a) \sum_{c \in C} p(c \mid a) \log_2 p(c \mid a) \tag{2.36}$$

De (2.37), la cantidad por la cual la entropía de la clase *C* decrece refleja, la información adicional en la clase, la cual es suministrada por el atributo *A* y se le llama ganancia de información. A cada atributo A se le asigna un resultado basado en la ganancia de información entre sí misma y la clase

$$\begin{aligned} \mathrm{IG}_i &= H(C) - H(C \mid A_i) \\ &= H(A_i) - H(A_i \mid C) \\ &= H(A_i) + H(C) - H(A_i, C) \end{aligned} \tag{2.37}$$

Esta técnica usa variables nominales (no numéricas), por lo cual se requiere discretizacion de las variables [Fayyad e Irani 1993]. Para tomar la decisión de cuales variables deben ser seleccionadas, ganancia de información usa un criterio de selección (ranker), es decir, se enlista los atributos de acuerdo a su resultado IG_i y una función de restricción los filtra. Las variables restantes son las seleccionadas.

Tabla 2.1. El clima de catorce juegos de tenis y la decisión de jugar o no

Ambiente	Temperatura	Humedad	Viento	Juega
Soleado	Caluroso	Alta	Falso	No
Soleado	Caluroso	Alta	Cierto	No
Nublado	Caluroso	Alta	Falso	Si
Lluvioso	Templado	Alta	Falso	Si
Lluvioso	Frio	Normal	Falso	Si
Lluvioso	Frio	Normal	Cierto	No
Nublado	Frio	Normal	Cierto	Si
Soleado	Templado	Alta	Falso	No
Soleado	Frio	Normal	Falso	Si
Lluvioso	Templado	Normal	Falso	Si
Soleado	Templado	Normal	Cierto	Si
Nublado	Templado	Alta	Cierto	Si
Nublado	Caluroso	Normal	Falso	Si
Lluvioso	Templado	alta	Cierto	No

Tabla 2.2. Atributo perspectiva y las probabilidades condicionales de su datos

Ambiente			Total	Ambiente p(c|a)			Total
	Si	No			Si	No	
Soleado	2	3	5	Soleado	2/5	3/5	5/5
Nublado	4	0	4	Nublado	4/4	0/4	4/4
Lluvioso	3	2	5	Lluvioso	3/5	2/5	5/5

Así por ejemplo para la tabla 2.1 de juegos de tenis, tenemos 14 juegos en los cuales la hay dos clases ***si*** y ***no*****,** su entropía está dada por

$$H(C) = -p(si)\log_2 p(si) - p(no)\log_2 p(no)$$
$$= -\left(\frac{9}{14}\log_2\frac{9}{14}\right) - \left(\frac{5}{14}\log_2\frac{5}{14}\right) = 0.9403 bits$$

Para el C su entropía dado *Ambiente* es, según 2.36

$$H(C \mid Ambiente) = -\frac{5}{14}\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) - \frac{4}{14}\left(\frac{4}{4}\log_2\frac{4}{4} + \frac{0}{4}\log_2\frac{0}{4}\right)$$
$$-\frac{5}{14}\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right) = -\frac{5}{14}(-0.971) - \frac{4}{14}(0) - \frac{5}{14}(-0.971) = 0.6936 bits$$

y la ganancia de información de perspectiva es

$$IG_{Ambiente} = H(C) - H(C \mid Ambiente) = 0.9403 - 0.6936 = 0.2467 bits$$

Así, para cada atributo se calcula la ganancia de información y se filtra mediante un umbral. Por ejemplo si, el umbral fuera de 0.3 bits el atributo Ambiente no se tomaría en cuenta.

2.6. Aprendizaje de Máquina (Machine Learning) mediante Naive Bayes

El aprendizaje de máquina es parte de la Inteligencia Artificial, y se encarga de la cuestión de construir programas de computadora que sean capaces de mejorar con el uso de la experiencia [Mitchell 1997].

Existen diversos métodos para construir estos sistemas de aprendizaje automático, existen una variedad, algunos provenientes de métodos de minería de datos, y otros que son en sí mismos un área con fundamentos propios. Entre ellos podemos encontrar los métodos bayesianos, las redes neuronales, los árboles de decisión. Entre los métodos de tipo bayesiano destaca Naive Bayes, el cual es sencillo de implementar, se utiliza tanto para biclasificación como para multiclasificación. Aunque está diseñado para variables nominales, mediante una discretización es posible manejar variables numéricas.

En Naive Bayes, se utilizan todas las variables y considera que las contribuciones de cada una de ellas son igualmente importantes en la decisión, además considera a cada variable independiente una de otra dada una clase, lo cual no corresponde en la realidad, no todas las variables son igualmente importantes e independientes una de otra [Witten y Frank 2000]. La independencia de las variables permite la multiplicación de las probabilidades con el fin de calcular la verosimilitud (likelihood) de la clase.

Naive Bayes está basado en la regla de Bayes, la cual dice que si se tiene una hipótesis *h* y una evidencia *E* que es relevante para la hipótesis, entonces

$$P(h \mid E) = \frac{P(E \mid h)P(h)}{P(E)} \tag{2.38}$$

donde

$P(h)$, es la probabilidad independiente de *h*

$P(E)$, es la probabilidad independiente de *E*

$P(E \mid h)$, probabilidad condicional de *E* dado *h*

$P(h \mid E)$, probabilidad condicional de *h* dado *E*

Así, basados en la regla de Bayes, se puede calcular la *hipótesis máxima a posteriori* (*maximum a posteriori hypothesis*)

$$h_{MAP} \equiv \arg\max_{h \in H} P(h \mid E)$$

$$h_{MAP} \equiv \arg\max_{h \in H} \frac{P(E \mid h)P(h)}{P(E)}$$

$$h_{MAP} \equiv \arg\max_{h \in H} P(E \mid h)P(h) \qquad (2.39)$$

donde, *H* es el conjunto de todas las hipótesis y h_{MAP} la hipótesis máxima a posteriori. La probabilidad independiente de *E*, es omitida en h_{MAP}, dado que representa una constante independiente de *E* [Mitchell 1997], la formula (2.39) es la que es aplicada finalmente.

Tabla 2.3. Tabla de datos del clima

Perspectiva			Temperatura			Humedad			Viento			Juega	
	Si	No		Si	No		Si	No		Si	No	Si	No
Soleado	2	3	Caluroso	2	2	Alta	3	4	Falso	6	2	9	5
Nublado	4	0	Templado	4	2	Normal	6	1	Cierto	3	3		
Lluvioso	3	2	Frío	3	1								

Tabla 2.4. Probabilidades condicionales de los atributos del clima de los catorce juegos de tenis

Perspectiva			Temperatura			Humedad			Viento			Juega	
	Si	No		Si	No		Si	No		Si	No	Si	No
Soleado	2/9	3/5	Caluroso	2/9	2/5	Alta	3/9	4/5	Si	6/9	2/5	9/14	5/14
Nublado	4/9	0	Templado	4/9	2/5	Normal	6/9	1/5	No	3/9	3/5		
Lluvioso	3/9	2/5	Frío	3/9	1/5								

En la tabla 2.3 se observa los datos del clima y en la tabla 2.4 las probabilidades condicionales en 14 juegos de tenis y las decisiones que se tomaron de jugar o no, así, por ejemplo si tenemos un nuevo juego con las siguientes características, Ambiente: soleado, Temperatura: frío, Humedad: alta, Viento: si, entonces debemos calcular su *hipótesis máxima a posteriori* con **Si** y con **No**

MAP de $Si = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0053$
MAP de $No = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 4/14 = 0.0206$

En algunos casos se puede asumir, que todas las hipótesis son igualmente probables a priori, así, por ejemplo $P(h_i) = P(h_j)$, para toda $h_i, h_j \in H$, lo que quiere decir que se asume una *probabilidad apriori uniforme*, esto facilita el cálculo de h_{MAP}

$$h_{ML} \equiv \arg\max_{h \in H} P(E \mid h) \tag{2.40}$$

esta es llamada *Hipótesis de Máxima Verosimilitud* (Maximum Likelihood Hypothesis).

¿Cómo funciona el clasificador? Pues ahora suponga que se tiene un conjunto de instancias de entrenamiento o experiencia y un conjunto finito de clases *C*, la tarea del clasificador es predecir correctamente la clase de una nueva instancia con *n* atributos $\langle a_1, a_2, ..., a_n \rangle$.

$$c_{MAP} = \arg\max_{c_j \in C} P(c_j \mid a_1, a_2, ..., a_n)$$

$$c_{MAP} = \arg\max_{c_j \in C} \frac{P(a_1, a_2, ..., a_n \mid c_j) P(c_j)}{P(a_1, a_2, ..., a_n)}$$

$$c_{MAP} = \arg\max_{c_j \in C} P(a_1, a_2, ..., a_n \mid c_j) P(c_j) \tag{2.41}$$

donde, *C* es el conjunto de todas las clases.

Así, para el ejemplo de los juegos de tenis el clasificador Naive Bayes sería

$$c_{NB} = \arg\max_{c_j \in [si,no]} P(c_j) \prod_i P(a_i \mid c_j)$$

y para los 36 clasificadores binarios de lenguas usados en este proyecto donde los atributos se determinan mediante medidas estadísticas aplicadas a los coeficientes wavelets

$$c_{NB_1} = \arg\max_{c_j \in [Inglés, Alemán]} P(c_j) \prod_i P(a_i \mid c_j)$$

$$c_{NB_2} = \arg\max_{c_j \in [Inglés, Español]} P(c_j) \prod_i P(a_i \mid c_j)$$

$$\vdots$$

$$c_{NB_{36}} = \arg\max_{c_j \in [Tamil, Farsi]} P(c_j) \prod_i P(a_i \mid c_j)$$

En este caso los clasificadores de lenguas no usan variables nominales, como en el ejemplo de los juegos de tenis, por lo cual se discretizan mediante el uso de media y desviación estándar, y mediante ellas es calculada su densidad de probabilidad [Witten y Frank 2005][3]

[3] Data Mining: Practical Machine Learning Tools and Techniques, p. 92-93, 2005

Capítulo 3

Estado del Arte

Estaban rodeados por una especie de osos de peluche de 1 metro de altura con lanzas en forma amenazadora, en la luna forestal Endor — ¡Oh!, mi cabeza, ¡Dios mío! —C3PO
Los seres comenzaron a hacer alabanzas y C3PO les habló en un lenguaje que entendieron
¿Entiendes algo de lo que dicen? — Luke Skywalker
Claro, amo Luke, recuerde que tengo fluidez en más de 6 millones de formas de comunicación — C3PO
¿Qué les estás diciendo? — Han Solo
"Hola", creo yo, podría equivocarme, usan un dialecto primitivo, pero creo que piensan que soy una especie de dios — C3PO — Chewbacca, emitió una risa, y Luke y Han marcaron una sonrisa.
¿Por qué no usas tu influencia divina y nos sacas de aquí? — Han Solo

— George Lucas StarWars Episodio VI

El estudio del habla abrió un campo de actividad científica y tecnológica hace ya varias décadas, en 1950 cuando los laboratorios Bell implementaron los primeros reconocedores de dígitos, mediante voz, basados en características fonético-acústicas. En el campo del lenguaje natural dos de los problemas que se abordan actualmente son el reconocimiento del habla y la síntesis de voz; sin embargo, el entender y emular la manera en que el ser humano habla y

escucha ha sido todo un reto y que ha necesitado la conjunción de diversas disciplinas con resultados limitados, no obstante solo parece ser cuestión de tiempo y esfuerzo. Un investigador importante en el área de reconocimiento del habla es el Ing. Eléctrico Lawrence Rabiner el cual junto a su colega Biing-Hwang Juang publicaron un documento llamado *Fudamentals of speech recognition* en 1993, que ha llegado a ser una referencia ineludible en la materia, y que enfatiza el uso de métodos de reconocimiento de patrones para esta tarea.

En este mismo orden aparecieron otros problemas ligados al reconocimiento del habla, como lo son: el reconocimiento de emociones, la identificación del hablante, la identificación de voz activa y la identificación de lenguas habladas, siendo esta última en la que se enfoca esta sección.

La identificación automática de lenguas a diferencia del reconocimiento automático del habla, no requiere saber lo que dice el hablante. Se puede decir que, es el problema de identificar una lengua hablada mediante una muestra de voz de un hablante cualquiera [Muthusamy 1992]. Su importancia se elevó debido a la globalización, el uso de interfaces multilingües ha cobrado importancia por ejemplo en aeropuertos, y en la red telefónica mundial, la mayoría de las bases de datos utilizadas para la identificación de lenguas son telefónicas, muestreadas a 8000 Hertz (OGI-TS, CallFriend, CallHome, SpechDat-M).

A manera de reseña se mencionan los proyectos más relevantes de la identificación automática de lenguas.

3.1. La Identificación Automática de Lenguas en la Década de 1970

Hablar de la identificación automática de lenguas es remontarse a 1970, cuando la compañía Texas Instruments llevo a cabo un proyecto de esta naturaleza, usando una base de datos basada en lecturas, para 7 lenguajes, con 100 hablantes (50 hombres, 50 mujeres), para lo cual uso detección de “sonidos de referencia” o secuencias de sonidos, particulares a cada lenguaje, además de estimación de verosimilitudes logarítmicas (log likelihoods) de los lenguajes. En el

proceso también se utilizó secuencia de fonemas, el proyecto mostró cuatro estudios desde 1973 a 1980. Los resultados en clasificación por pares utilizando vecinos cercanos y mínima verosimilitud fue de alrededor de 62%.

Por aquellos años, se buscaban nuevas características para la discriminación de los lenguajes dado que el estado del arte en aquellos años estaba enfocado a la información de carácter fonético (pausas, fricativas, vocales, silencios), se decidió extraer características acústicas a la información fonética, como es el caso de [House y Neuberg 1977], ellos utilizaron técnicas markovianas para el proceso de discriminaron de las lenguas.

Tomando dichas técnicas, las de Markov (donde su aplicación primaria fue el procesamiento de lenguaje escrito) [Li y Edwads 1980] las combinaron con el uso de una categoría más amplia de información fonética (silabas, vocales, fricativas, detección de voz, aspiración, etc.), con lo que se construyeron dos modelos: uno basado en segmentos y otro basado en silabas.

3.2. La Identificación Automática de Lenguas en la Década de 1980

El uso de información acústica la cual evitara el uso de unidades fonéticas tales como fonemas, vocales, consonantes, silabas, etc. fue una manera de trabajar para diversos investigadores en la década de 1980, elevando la importancia del procesamiento digital de señales. Un ejemplo fue la aplicación de análisis LPC (Análisis de Codificación de Predicción Lineal), Coeficientes Cepstrales[4], Coeficientes de Autocorrelación, extraídos mediante ventanas móviles, esta investigación llevada a cabo por [Cimarusti e Ives 1982] utilizó una función polinomial como clasificador. Así como estas características también se emplearon la prosodia y el ritmo.

[4] Producto de la aplicación del cepstrum, lo cual es la aplicación de la Transformada de Fourier al espectro de decibeles y tomar dicha transformada como una señal, la palabra cepstrum viene spectrum.

[Foil 1986] tuvo la particularidad de ser el primero en usar grabaciones provenientes de señales de radio con alto contenido de ruido que investigaciones anteriores no tuvieron y de utilizar segmentos de menos de 10 segundos de duración. Foil aplicó detección del Pitch[5] y agrupación de formantes, en esta misma dirección se desarrolló la investigación de [Goodman et al. 1989].

3.3. La Identificación Automática de Lenguas en la Década de 1990

La década de 1990 trajo nuevas investigaciones y aportaciones a la identificación de lenguas, las cuales han definido el rumbo en los últimos años, una de ellas proviene de un fuerte estudio lingüístico llamado fonotáctico, que puede ser visto como un conjunto de reglas léxicas y gramaticales, aplicado al reconocimiento de fonemas y modelado de los idiomas. En esta sección se hará un estudio más detallado pues son estas investigaciones muy interesantes para el presente estudio. Algunos de los investigadores fueron los siguientes:

[Sugiyama 1991] aplica algunas características acústicas anteriormente citadas y agrega los vectores de cuantificación como una manera de discriminación. [Muthusamy et al. 1992] por su parte combina redes neuronales con características fonéticas tales como vocales, fricativas, pausas, entre otras, clasifica 4 lenguajes utilizando muestras cortas de tiempo de 5.7s y 17.2s provenientes de una base de datos tipo telefónica (muestreada a 8000 m/s).

No existe una claridad en la clasificación de los distintos métodos de identificación automática de lenguas de lenguas, pero [Muthusamy 1992] establece las siguientes:

- Identificación usando características acústicas (LPC, Cepstrales, SDC (Shifted Delta Cepstral), Pitch,…).
- Identificación usando categorías fonéticas amplias (Fricativas, vocales, pausas,…).

[5] Frecuencia fundamental de la voz, es la frecuencia mas baja del espectro de frecuencias, los armónicos son múltiplos del Pitch

- Identificación usando categorías fonéticas finas (fonemas, alófonos, características fonotácticas).

[Berkling 1996] Kay Margarethe Berkling es una de las investigadoras que basa su metodología en el uso de información fonotáctica, los fonemas toman importancia en su investigación, trata de demostrar que la parte lingüística es determinante en este tipo de investigaciones. En la mayoría de los trabajos con fonemas su usan mono-fonemas, es decir fonemas que solo ocurren en determinada lengua, pero en su investigación utiliza polifonemas, es decir fonemas presentes en casi todos los lenguajes, dados estos fonemas lleva a cabo agrupación jerárquica o clustering jerárquico (aprendizaje no supervisado) lo que le permite distinguir entre distintos idiomas, estableciendo un modelo a seguir para los investigadores que prefieren usar fonemas y características fonotácticas (figura 3.1).

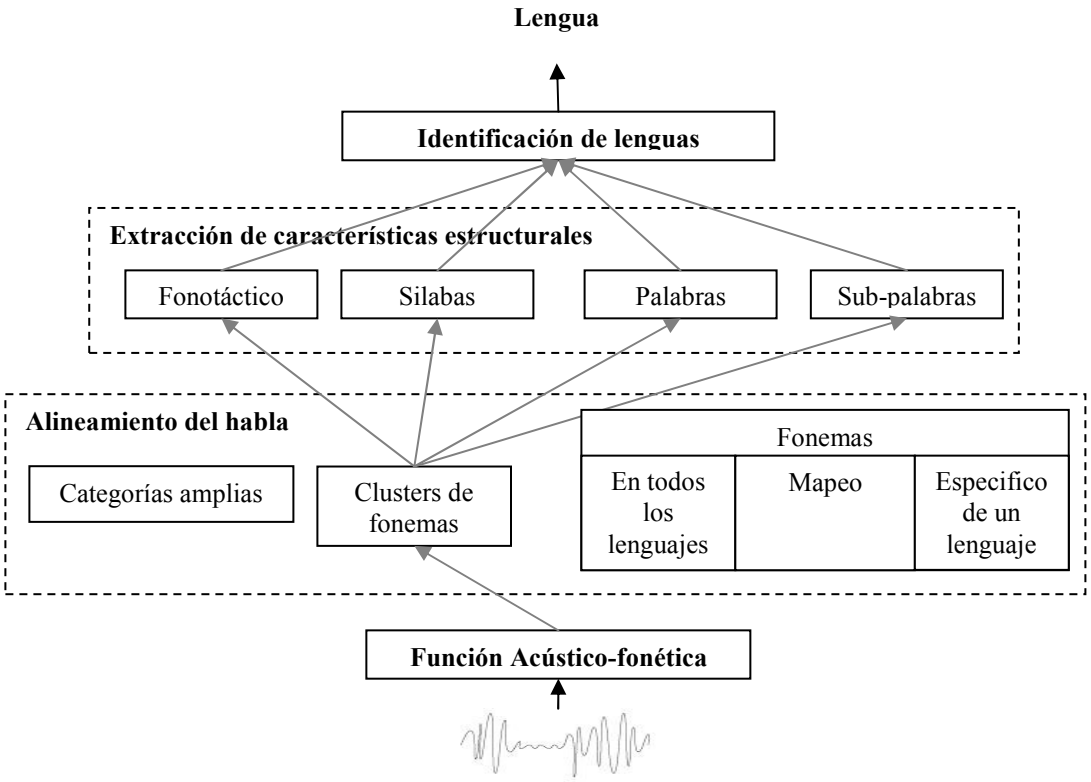


Figura 3.1. Modelo de Berkling

Navratil Jiri, es uno investigador muy representativo del estilo acústico-fonotáctico, la combinación provee una metodología fuerte, incluye además el uso de *n*-grams, los cuales, son árboles de fonemas, donde *n* representa el número de ramas que poseen estos árboles.

La metodología de Navrátil se compone en primer lugar de un extractor de características (en este caso fonemas) para alimentar a un reconocedor de fonemas con una arquitectura de decodificación de árbol binario de fonemas (*bigrams*) (parte fonotáctica) y también a un conjunto de modelos de pronunciación dependiente del lenguaje para cada uno de los fonemas (parte acústica), y finalmente ambas partes después de ser procesadas alimentan a una red neuronal que se encarga de la multiclasificación.

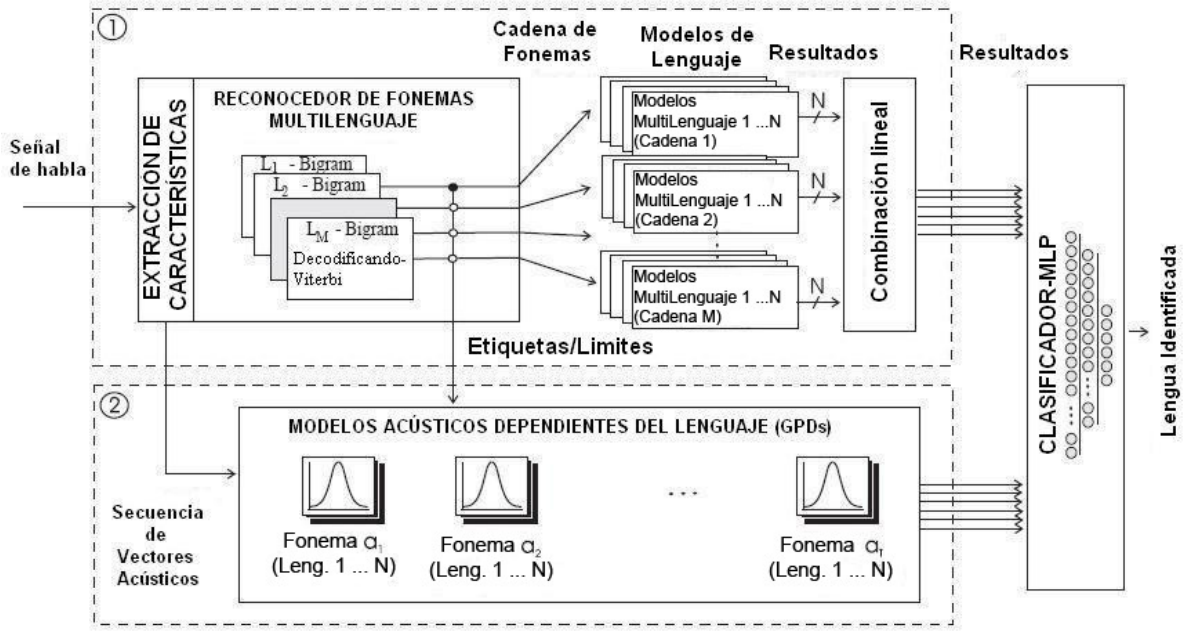


Figura 3.2. Modelo del sistema acústico-fonotáctico de Navratil

Navrátil trabaja usando para entrenamiento, la base de datos telefónica OGI_TS, de la cual toma 9 idiomas, Inglés, Alemán, Hindi, Japonés, Mandarín, Español, Francés, Tamil, y Vietnamita y para pruebas la base de datos de prueba NIST. El tamaño de las muestras de habla es de 45/10 segundos, haciendo dos experimentos, el primero con los primeros 6 lenguajes y el segundo con los 9. Los resultados tienen los siguientes grados de error:

9.8%/0.8% y 14.7%/5.6% para 10s y 45s de los 6 y 9 lenguajes respectivamente [Navrátil y Zühlke-1998].

Diamantino Caseiro propone una metodología (figura 3.3) únicamente fonotáctica, muy semejante al componente fonotáctico de Navratil, este tiene un extractor de características, mediante coeficientes cepstrales mel, 12 coeficientes cepstrales delta, energía y energía delta (delta o de diferencia[6]), que alimenta a un reconocedor de fonemas basado en HMM (Modelos Ocultos de Markov) y llena un conjunto de modelos de lenguaje basado en *bigrams* de fonemas, los cuales tienen como función calcular la verosimilitud (likelihood) de cada lenguaje y estas verosimilitudes son usadas por el clasificador para hacer la multiclasificación.

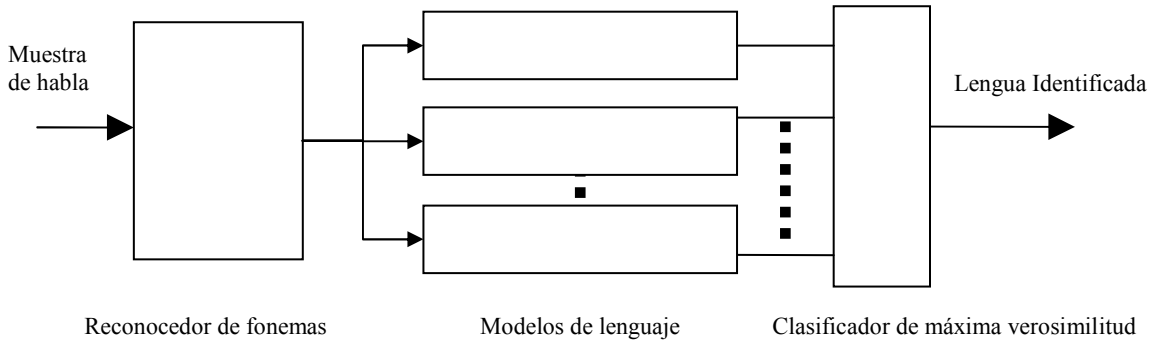


Figura 3.3. Arquitectura del sistema de Caseiro

La base lingüística es mínima ya que solo usa un lenguaje, el portugués, lo que hace que el reconocedor de fonemas busque información más específica, es decir, utiliza como en el caso de Berkling polifonemas (fonemas presentes en todos lenguajes) específicamente los del portugués, que le permitió clasificar los lenguajes restantes. El entrenamiento y pruebas fueron realizados utilizando la base de datos telefónica de lenguajes europeos *Speech-Dat-M* la cual posee 7 lenguajes, Inglés, Francés, Alemán, Italiano, Portugués, Español y Francés Suizo, utilizando los primeros 6, y eligiendo 9 muestras de cada hablante en cada lengua. Los resultados muestran en la tabla siguiente:

[6] Delta, se refiere a calcular la diferencia de un coeficiente en un instante con otro en un instante posterior, o varios instantes posteriores.

Tabla 3.1. Matriz de confusión del sistema de Caseiro

	Inglés	Español	Alemán	Portugués	Italiano	Frances
Inglés	**81.40%**	0.40%	11.10%	1.70%	2.70%	2.80%
Español	1.90%	**70.60%**	2.30%	3.70%	15.90%	5.60%
Alemán	8.60%	1.50%	**82.40%**	1.20%	2.00%	4.20%
Portugués	2.50%	3.10%	1.90%	**87.80%**	1.70%	3.10%
Italiano	4.90%	14.40%	4.10%	1.10%	**70.00%**	5.60%
Francés	2.10%	2.90%	4.30%	1.20%	4.00%	**85.5%**

Fred Cummins establece una metodología basado en la prosodia y el ritmo de los lenguajes, específicamente la F0 (Frecuencia Fundamental de la voz), el sistema se compone de un extractor de características, las cuales son: ΔF0 y ΔEnv.

La ΔF0, es una log F0 calculada cada milisegundo, se le aplica una primera diferencia (Δ, delta o diferencia), posteriormente dicho vector es cambiado de frecuencia a 100Hz, y suavizado mediante una ventana rectangular de 15 puntos, finalmente es normalizado en el intervalo [-1 1]. La ΔEnv es una medida aplicada a la señal de voz en el dominio del tiempo, fue calculada mediante la aplicación de un filtro Butterworth pasabandas de bajo orden centrado en 1000Hz con un ancho de banda de 500Hz a la señal de habla. El resultado del filtrado conserva una banda de frecuencias con información de los formantes (Resonancias) libre de F0 y energía de las Fricativas

En el trabajo de [Cummins 1997] y [Scott 1993] sugieren que este rango de frecuencias es importante para la percepción del ritmo. El habla filtrada fue rectificada tomando valores absolutos de cada muestra, y suavizada mediante un filtro Butterworth pasabajos de 10Hz, y se le aplica la misma metodología que a log F0 (primera diferencia, remuestreo, suavizado y reescalado). Cummins uso una red neuronal del tipo LSTM (Long Short Term Memory) [Hochreiter y Schmidhuber 1997] que funciona como el clasificador binario del sistema.

El equipo de investigación de Fred Cummins realizó pruebas sobre la base de datos OGI_TS, ya mencionada anteriormente, Cummins se restringió a solo 5 idiomas, Inglés, Japonés, Español, Mandarín y Alemán. En su estudio experimental utilizó 50 muestras para

cada idioma y los evaluó por pares generando 10 clasificadores (Tabla 3.2) [Cummins et al-1999].

Tabla 3.2. Resultados de los clasificadores binarios del método de Cummins

	Alemán	Español	Japonés	Mandarín
Ingles	52	56	50	59
Alemán	-	51	5	58
Español	-	-	59	50
Japonés	-	-	-	63

3.4. La Identificación Automática de Lenguas en el Nuevo Milenio

El nuevo milenio trajo consigo el afianzamiento de técnicas como la fonotáctica y de procesamiento digital de señales, así como la aparición de nuevas técnicas, una de ellas es la inclusión de la *Transformada Wavelet*, la cual posee ciertas características que funcionan mejor para señales en particular como por ejemplo la señal del habla.

Torres-Carrasquillo, en su trabajo (GMM-SDC) propone un modelo que no requiera de reglas léxicas y gramaticales y evite el uso de *n-grams* de alto coste computacional utilizado en el PPR-LM (Parallel Phone Recognition and Language Model). Para lograrlo el sistema se basa en un extractor de características mediante SDC (Shifted Delta Ceptral) los cuales representan las diferencias entre coeficientes cepstrales, posteriormente son tratados por un conjunto de sistemas de simbolización GMM para cada lenguaje. La verosimilitud producida por los sistemas de simbolización es evaluada por un conjunto de modelos de lenguaje y sus resultados alimentan a un clasificador basado en modelos de mezclas gaussianas (GMM) que funciona como multiclasificador.

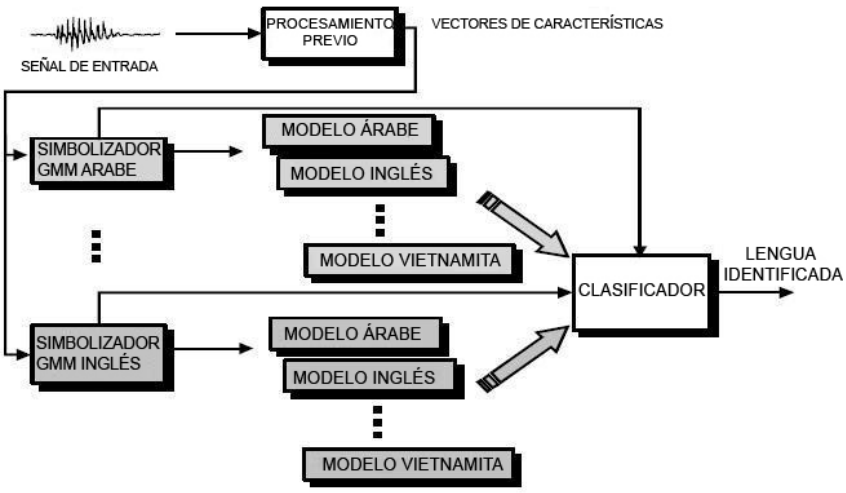


Figura 3.4. Arquitectura del modelo de Torres-Carrasquillo

Para realizar el entrenamiento y pruebas se utilizan dos bases de datos telefónicas, la *CallFriend* que contiene 12 lenguajes los cuales son: Árabe, Inglés, Farsi, Francés, Alemán, Hindi, Japonés, Coreano, Mandarín, Español, Tamil y Vietnamita. La otra base de datos utilizada fue la ya mencionada OGI_TS utilizando los lenguajes anteriores con excepción del Árabe. Los resultados muestran para CallFreind una tasa de error del 6.9% contra 7.84% del sistema PPRLM, para OGI_TS cuenta con 29% contra 21% del PPRLM con 10 segundos y 24% contra 11% del PPRLM para muestras de 45 segundos [Torres-Carrasquillo et al-2002].

[Rouas et al. 2003] En sus investigaciones, Rouas propone un método basado en la entonación y el ritmo, usando unidades de intervalos vocálicos y consonánticos para caracterizar el ritmo, lo que él llama seudo-silabas. Sus trabajos, están basados en la teoría de Ramus de la habilidad de los recién nacidos de distinguir idiomas mediante segmentación vocálica y consonántica [Ramus, 2000]. Su sistema está integrado por una fase de generación de seudosilabas el cual está formado a su vez de un módulo de segmentación de la señal, posteriormente un módulo de detección de voz activa y detección de vocales, la siguiente fase es la extracción de características donde 2 módulos paralelos obtienen las características de medidas vocálicas y consonánticas (Duración de consonantes, Duración de las vocales, complejidad del grupo de consonantes) y las de entonación aplicadas a la F0 (media, desviación estándar, skewness y Kurtosis). Ambos módulos alimentan un clasificador basado

en Modelos de Mezclas Gaussianas (GMM) que funcionan como clasificador de dos modelos de lenguajes (biclasificador).

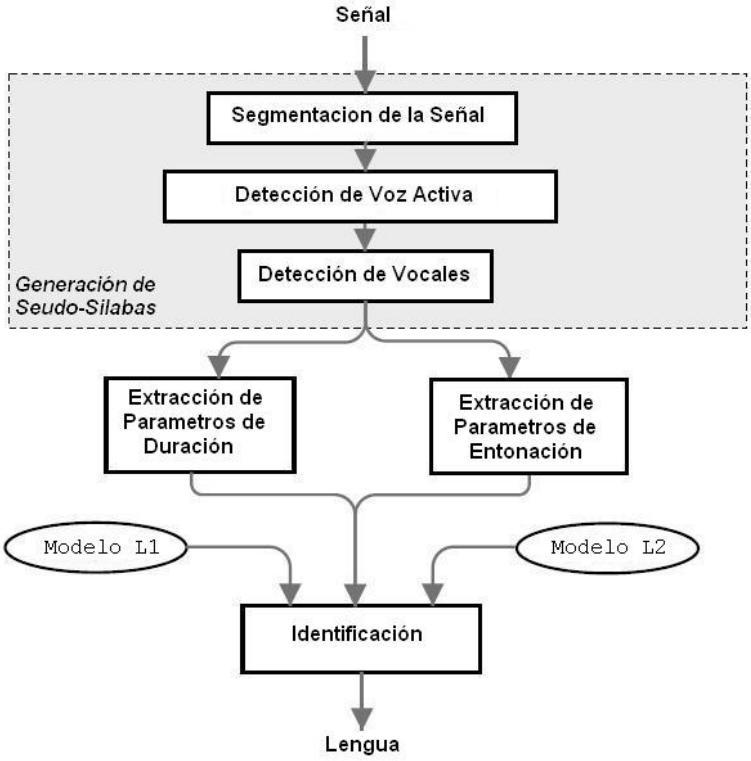


Figura 3.5. Arquitectura del modelo de Rouas

Utiliza la base de datos OGI_TS, los idiomas clasificados fueron 10: Inglés, Alemán, Francés, Español, Mandarín Vietnamita, Japonés, Coreano, Tamil y Farsi. En promedio los resultados que obtiene son de un 70% de clasificación.

Tabla 3.3. Resultados de los clasificadores binarios del método de Rouas

	Alemán	Francés	Español	Mandarín	Vietnamita	Japonés	Coreano	Tamil	Farsi
Ingles	60	52	68	75	68	68	79	77	76
Alemán	-	56	59	62	66	66	71	70	72
Francés	-	-	64	61	58	56	55	60	69
Español	-	-	-	81	62	63	76	65	67
Mandarín	-	-	-	-	50	50	74	74	76
Vietnamita	-	-	-	-	-	69	56	71	67
Japonés	-	-	-	-	-	-	66	59	67
Coreano	-	-	-	-	-	-	-	62	75
Tamil	-	-	-	-	-	-	-	-	70

El *MITLL System* 2005, este proyecto fue la conjunción de diversos proyectos del tipo fonotáctico llevado a cabo por el MIT Lincoln Laboratory e IBM TJ. Watson Research Center, que puede ser consultado en [Campbell et al. 2006]. Algunos de los modelos fonotácticos se muestran en la tabla 3.4.

Tabla 3.4. Tabla 3.4. Los 6 modelos que integraron el MITLL System

Modelo	Descripción
GMM-SDC (Gaussian Mixture Model with Shifted Delta Cepstral Features)	El sistema consiste en un modelo dependiente del lenguaje, incorpora un UBM (Universal Background Model), extrae características mediante SDC (Shifted Delta Cepstral). Utilizando para construir el modelo de cada lenguaje un simbolizador GMM, el método usado para la clasificación fue el Modelo de Mezclas Gaussianas (GMM)
SVM-SDC (Support Vector Machine with Shifted Delta Cepstral Features)	El sistema se basa en una máquina de soporte vectorial GLDS y Extracción de características mediante SDC (Shifted Delta Cepstral) con una configuración similar al GMM-SDC. Utilizando para construir el modelo de cada lenguaje un simbolizador GMM.
PPR-LM (Parallel Phone Recognition followed by Language Models Classifiers)	El sistema se basa en el reconocimiento de fonemas paralelos , los modelos de cada lenguaje fueron construidos mediante el uso de árboles de fonemas ternarios (Trigrams)
PPR-LM-lattice (Parallel Phone Recognition followed by Language Models Classifiers using Phone Lattices).	El sistema se basa en el reconocimiento de fonemas paralelos mediante phone lattices (celosías de fonemas), los modelos de lenguaje son construidos mediante tri-grams y simbolizadores.
PPR-SVM-lattice (Parallel Phone Recognition followed by Support Vector Machine using Phone Lattices).	El sistema se basa en el reconocimiento de fonemas paralelos mediante phone lattices (celosías de fonemas), los modelos de lenguaje son construidos utilizando unigrams y bigrams.
PPR-BT (Parallel Phone Recognition followed by Binary Tree Language Models)	Este sistema usa reconocimiento de fonemas y emplea para generar los modelos de lenguaje arboles binarios de fonemas (bigrams) el cual se compone de los fonemas más comunes del lenguaje y permite calcular su probabilidad.

En la tabla anterior se muestran los trabajos de tipo fonotáctico los cuales han logrado mucha aceptación por sus buenos resultados, por tal razón se combinaron en MITTLL System (figura 3.6)

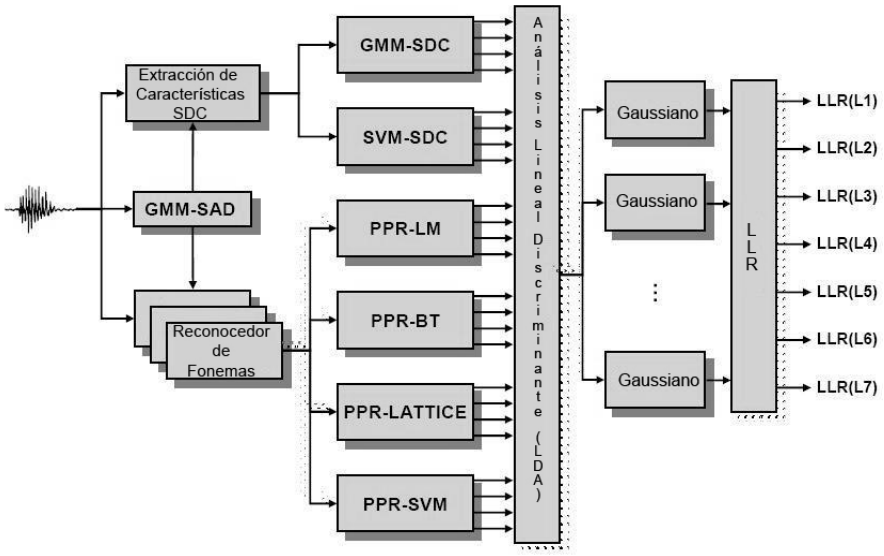


Figura 3.6. Modelo del MITLL-System

El sistema uso para su entrenamiento y pruebas una combinación de 5 bases telefónicas cubriendo 22 lenguajes esta fue llamada **xcorpus** y las bases de datos fueron: OGI_TS, CallFriend, CallHome, Fisher y Mixer, a las últimas 4 se les aplicó detección de voz activa (detección de los bloques de habla, sin pausas, ruido o alguna clase de sonido distinto del habla), el método de clasificación empleado fue el Modelo de Mezclas Gaussianas.

[Reyes 2007] La investigación de Reyes Herrera sigue la ruta trazada por Cummins y Rouas, con la diferencia fundamental de que incorpora por primera vez, en esta clase estudios, la transformada Wavelet para caracterizar la señal del habla, tiene como precedente su utilización en el reconocimiento del locutor, en el reconocimiento del habla, en la detección de voz activa entre otras [Gupta y Gilbert 2001; Hioka y Hamada 2003].

El procedimiento seguido en este trabajo usa la transformada wavelet Daubechies Db2 con cuatro coeficientes y normalizados a [-1, 1]. El método se basa en la idea de [Mallat, 1999] de que los coeficientes de mayor magnitud representan las bajas frecuencias y los coeficientes de menor magnitud corresponden a las altas frecuencias. De esta forma, se truncan los coeficientes wavelet de acuerdo a su magnitud con un umbral del 1%, esto le

permite, en primer término, eliminar algunos coeficientes, y en segundo, aplicar ganancia de información. Lo anterior conlleva un manejo más adecuado de la información ya que, por ejemplo, en la clasificación de una pareja de lenguajes (inglés/alemán) de 131,072 coeficientes, se reducen a 1,310 coeficientes relevantes y posteriormente con la ganancia de información a únicamente 641.

Finalmente, usando la base de datos OGI_TS, con los mismos idiomas que Rouas, con excepción del Francés, compara sus resultados con los de él, por ejemplo, para una muestra de 50 segundos (Tabla 3.4) sus resultados muestran que el uso de la transformada wavelet es muy pertinente en esta clase de estudios.

Tabla 3.5. Resultados de los clasificadores binarios del método de Reyes usando 50 segundos

	Alemán	Español	Mandarín	Vietnamita	Japonés	Coreano	Tamil	Farsi
Ingles	97	97	93	94	96	95	99	96
Alemán	-	93	94	93	98	98	94	91
Español	-	-	91	86	92	98	91	94
Mandarín	-	-	-	95	95	93	89	94
Vietnamita	-	-	-	-	93	96	95	95
Japonés	-	-	-	-	-	93	89	94
Coreano	-	-	-	-	-	-	95	91
Tamil	-	-	-	-	-	-		90

Capítulo 4

Metodología

La metodología que se usa para resolver el problema de la identificación de lenguas para este proyecto conjuga diversos tópicos entre ellos, wavelets, estadística, minería de datos y aprendizaje de máquina.

La idea de usar wavelets en investigaciones de este tipo es debido fundamentalmente al trabajo de Mallat, ya mencionado. La transformada wavelet descompone señales en subespacios sucesivos encajados, separando las altas y bajas frecuencias, lo cual se conoce como análisis multiresolución. Dado que una resolución representa un subespacio, la separación de frecuencias es clara, la importancia de esta cuestión, radica en que, fenómenos de la prosodia como el ritmo al que se le atribuye la distinción de lenguajes, se encuentran presumiblemente en las frecuencias bajas y una de las propiedades de las wavelets es lograr alta resolución en frecuencias bajas en periodos cortos de tiempo.

Nuestra metodología se enfoca en ciertas cuestiones que el trabajo de Reyes no contempla:

1. Una aplicación real necesita una cantidad de habla corta para identificar lenguas en un tiempo razonable (Reyes necesita 50 segundos, mas el procesamiento).

2. La aplicación de un modulo de Detección de Voz Activa (VAD), para la eliminación ciertos sonidos y artefactos presentes en la voz, en nuestro caso eliminación de pausas largas.
3. Reyes no usa medidas estadísticas sobre los coeficientes wavelet, como si lo hace Rouas sobre la F0.
4. El aprovechar las propiedades de las wavelets, como la resolución de frecuencias bajas en periodos cortos de tiempo requiere usar señales cortas, es decir sub-señales de la original (segmentación).

Así, la metodología está diseñada para identificar lenguas con muestras cortas de habla, usando medidas estadísticas simples sobre los coeficientes wavelet Db2 de la señal segmentada, además de detección de voz activa (VAD) simple para la eliminación de pausas largas en las señales de habla.

El sistema está formado de 6 partes importantes, que procesan las muestras de audio, mediante el uso de herramientas computacionales, entre ellas el software Praat [Boersma y Weenink 2002], Matlab y Weka [Witten and Frank 2005]. A continuación se muestra el diagrama general del sistema (figura 4.1).

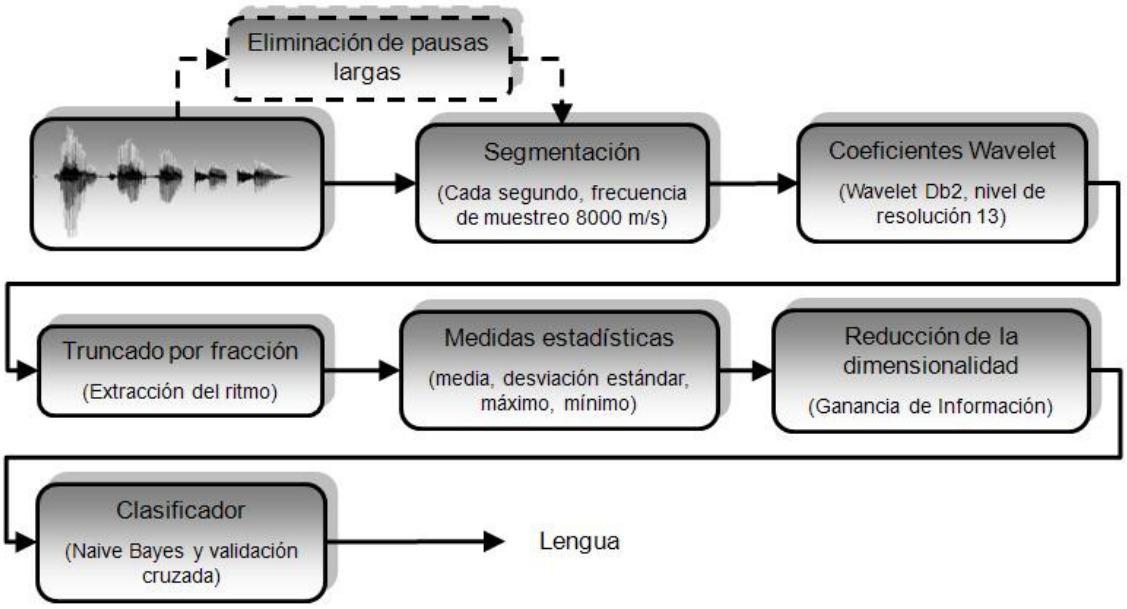


Figura 4.1. Sistema de identificación de lenguas en un diagrama a bloques

4.1. Segmentación

La primera parte del procesamiento, comprende la segmentación de la señal en pequeñas señales de 1 segundo, aunque para el estudio wavelet no es necesaria la segmentación de señales como sucede en estudios que utilizan Transformada de Fourier. Se sabe que en el área de procesamiento de imágenes, JPG2000 la cual es una aplicación de wavelets usa segmentación (crea subimágenes) para una mejor compresión. Para nuestro caso, lo que se desea es una mejor representación de la señal en términos del ritmo con menos datos, y dado que las wavelets logran una alta resolución en bajas frecuencias (donde se encuentra el ritmo) con periodos cortos de tiempo, la segmentación resulta conveniente. El número de segmentos será igual al número de segundos que dure la señal de habla.

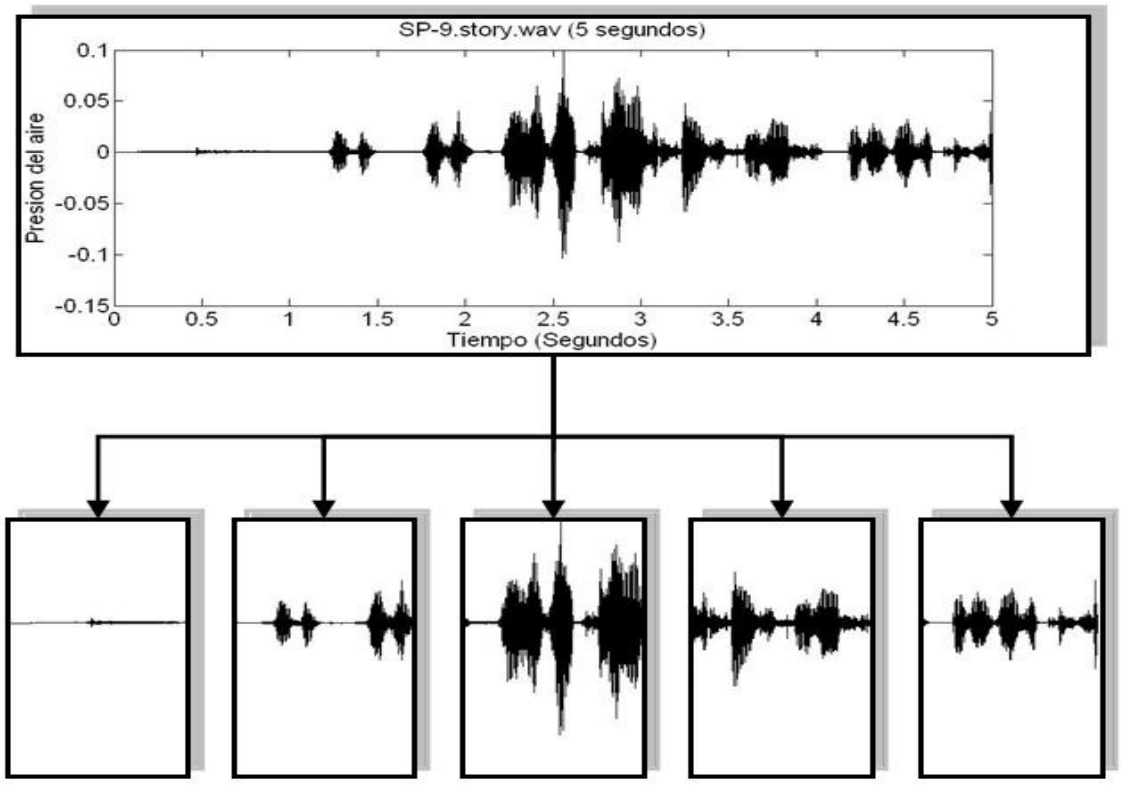


Figura 4.2. Ejemplo de una señal de 5 segundos de habla (SP-9.story) y sus 5 segmentos

Como puede observarse en la figura 4.2, la señal tiene una duración de 5 segundos, se ha dividido la señal es 5 segmentos, y dado que cuenta con una frecuencia de muestreo de 8kHz, cada segundo tiene 8000 muestras/segundo.

4.2. Coeficientes Wavelet

La trasformada wavelet Db2 se utiliza en este punto, la cual utiliza cuatro coeficientes filtro para descomponer las señales como se dijo, de un segundo, por lo tanto se aplicarán las transformaciones a todos los segmentos de un segundo de la señal, la trasformada es representada finalmente con los coeficientes wavelet producidas por la transformación.

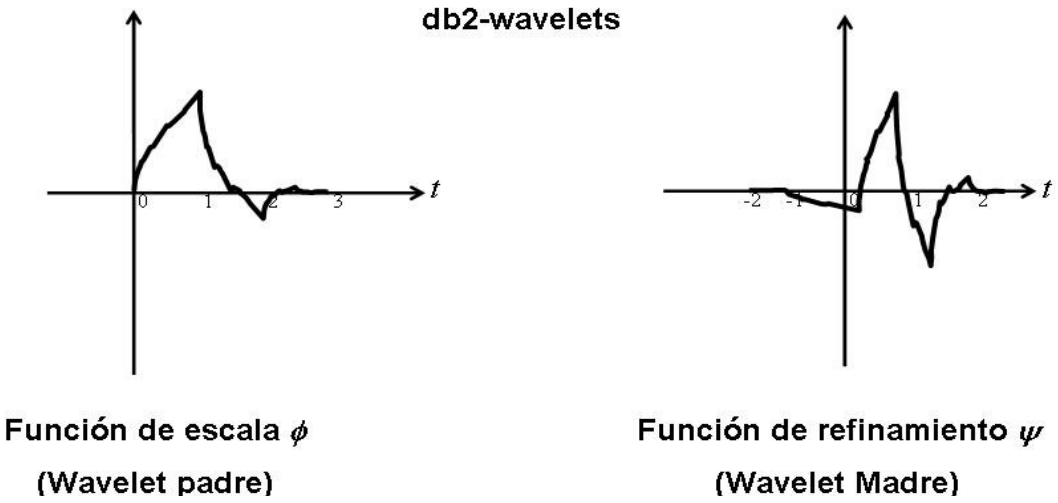


Figura 4.3. Funciones wavelet

Figura 4.3. Función de escala (pasa bajos) y refinamiento (pasa altos) de Db2-wavelets. Cada función funciona como un filtro, la wavelet padre como un filtro pasa bajos y la wavelet madre como filtro pasa altos, el árbol de descomposición permite ver este proceso.

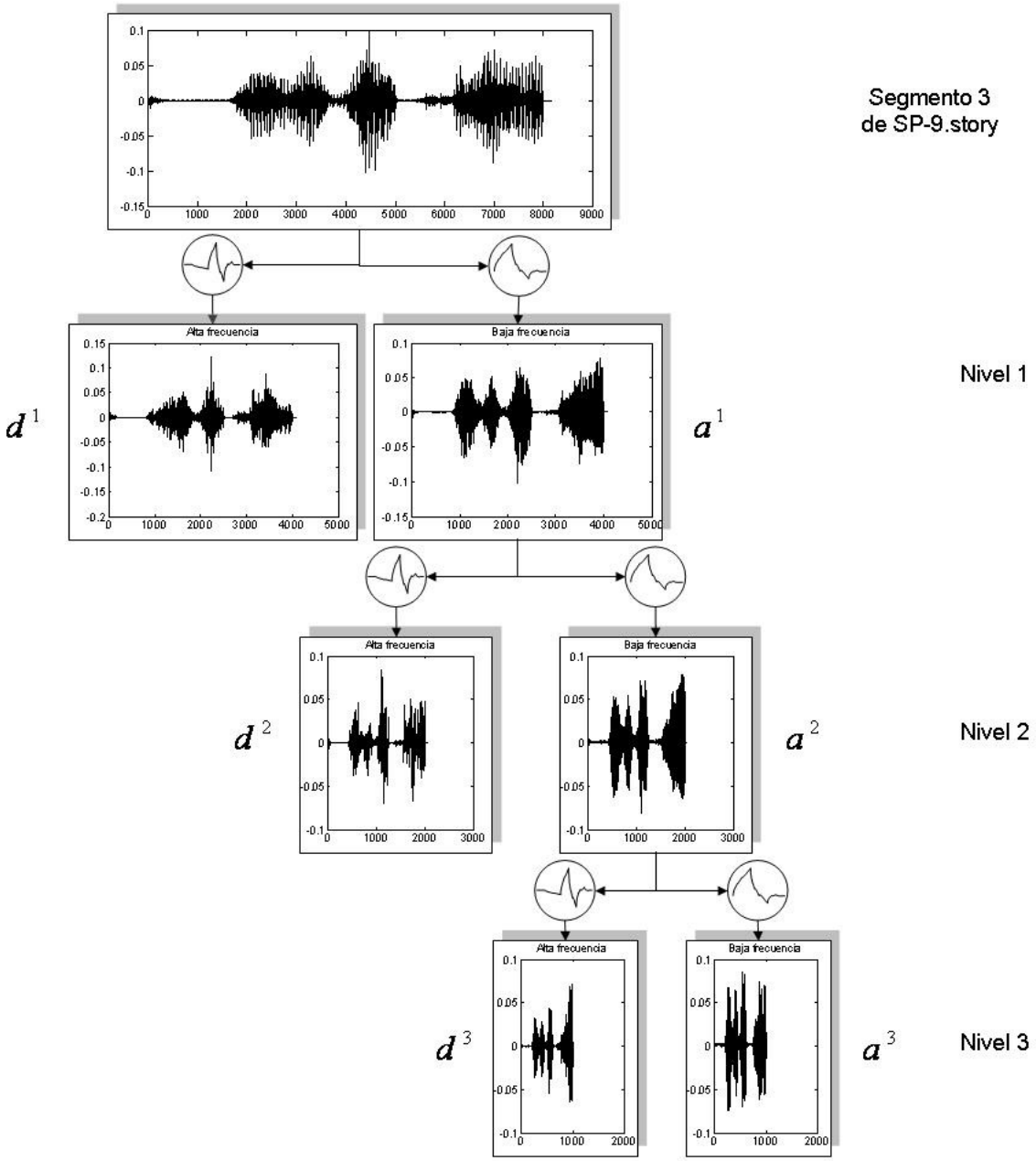


Figura 4.4. Se muestra una descomposición wavelet del tercer segmento de "SP-9.story.wav" con una resolución de j=3

En la figura 4.4 se puede observar la descomposición del tercer segmento de la señal mostrada en la figura 4.2 con un nivel de resolución de 3, puede observarse cómo cada filtro (figura 4.3), genera una señal del exactamente la mitad de la señal anterior, a lo que se le llama *decimation* (diezmado). Esta descomposición es llamada en cascada o algoritmo piramidal de Mallat, se puede observar gráficamente en la figura 4.4 y también se puede expresar en coeficientes en el siguiente formato [Walker 2008] como: (a^3 | d^3| d^2 | d^1)

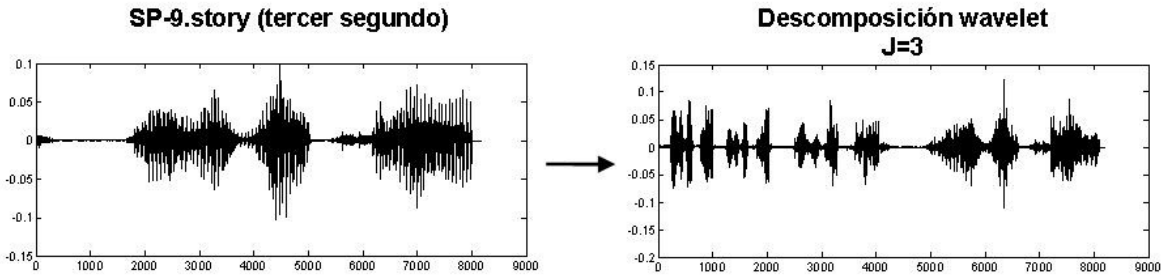


Figura 4.5. Descomposición con j=3

Podemos observar en la figura 4.5 la descomposición wavelet en 3 niveles. Observando detenidamente la segunda mitad de la descomposición, notamos que se compone aproximadamente de la misma señal pero a una escala menor, pues ha sido filtrada. Para los estudios de señales de voz una herramienta muy útil es el escalograma (*scalogram*) es análogo al *espectrograma*, algunos lo llaman plano tiempo-frecuencia de tejas, los tonos más intensos reprendan bajas frecuencias [Addison 2002].

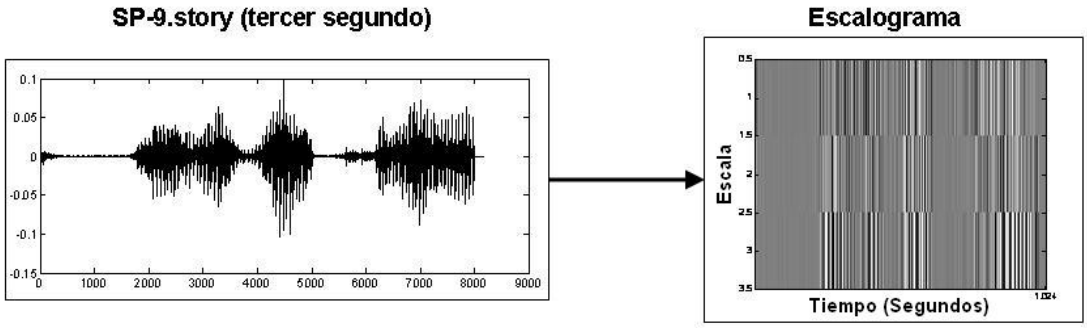


Figura 4.6. Escalograma de los 3 primeros niveles de descomposición

Mostrar 3 niveles es sencillo, pero es necesario para realizar una buena descripción de la señal un nivel de resolución de acuerdo con el teorema de Nyquist, dado que ese nivel de resolución es potencia de 2 (por las divisiones) se busca la potencia de 2 más cercana hacia arriba con respecto a la frecuencia de muestreo (Nyquist) y la duración de la señal, la señal cuenta con 8000 muestras/segundo y 1 segundo de duración, se requiere una potencia igual o mayor a $s \cdot fm = 1 \cdot 8000 = 8000$, la más cercana es 2^{13}=8192. Ya que se requieren 8192 muestras, se llenan con ceros los 192 que faltan en la señal que será transformada y el número de coeficientes

resultantes será igual al número de muestras, 8192. Los coeficientes de la figura 4.7 pueden ser expresados como: (a^{13} | d^{13}| d^{12} | d^{11}| d^{10}| d^{9}| d^{8}| d^{7}| d^{6}| d^{5}| d^{4}| d^{3}| d^{2}| d^{1}), correspondiente a cada nivel del escalograma de la figura 4.7.

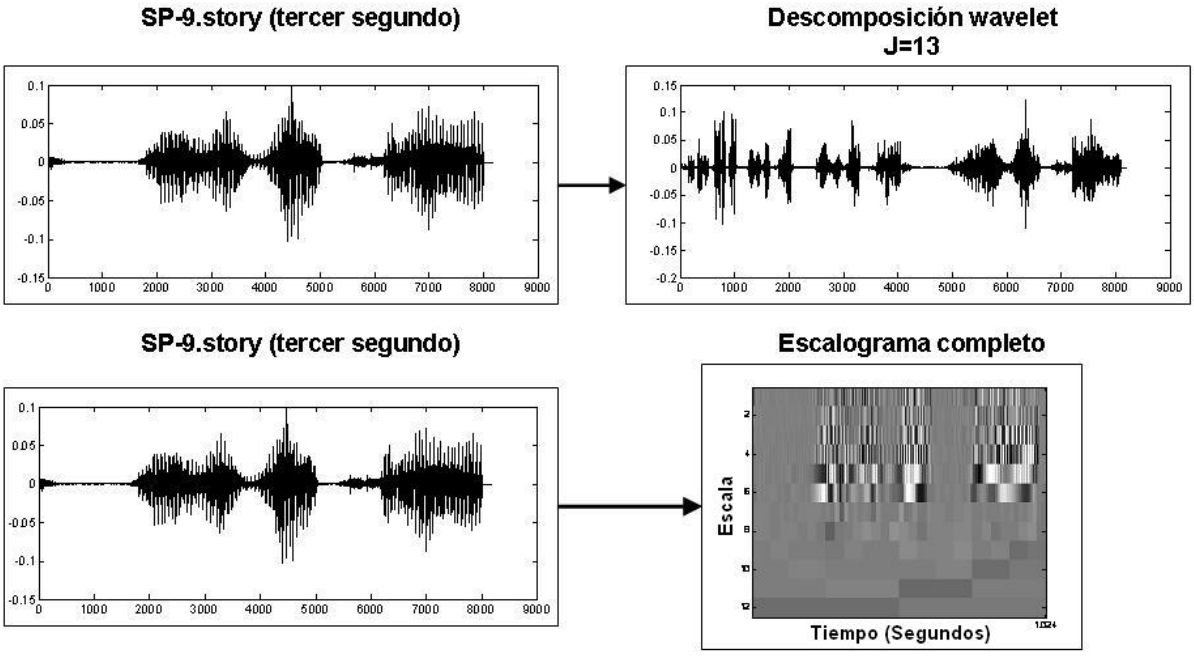


Figura 4.7. Se muestra la descomposición en 13 niveles y su escalograma correspondiente

Todo esto se aplica a cada segmento, en el ejemplo de la figura 4.2 se tienen 5 segmentos, por cada uno se tienen 8192, lo cual da 40960 coeficientes en total. Así, por ejemplo, una señal de 30 segundos, tendrá 245760 coeficientes. Debido a la gran cantidad de información que se produce, aplicamos técnicas para reducir la dimensionalidad de los coeficientes.

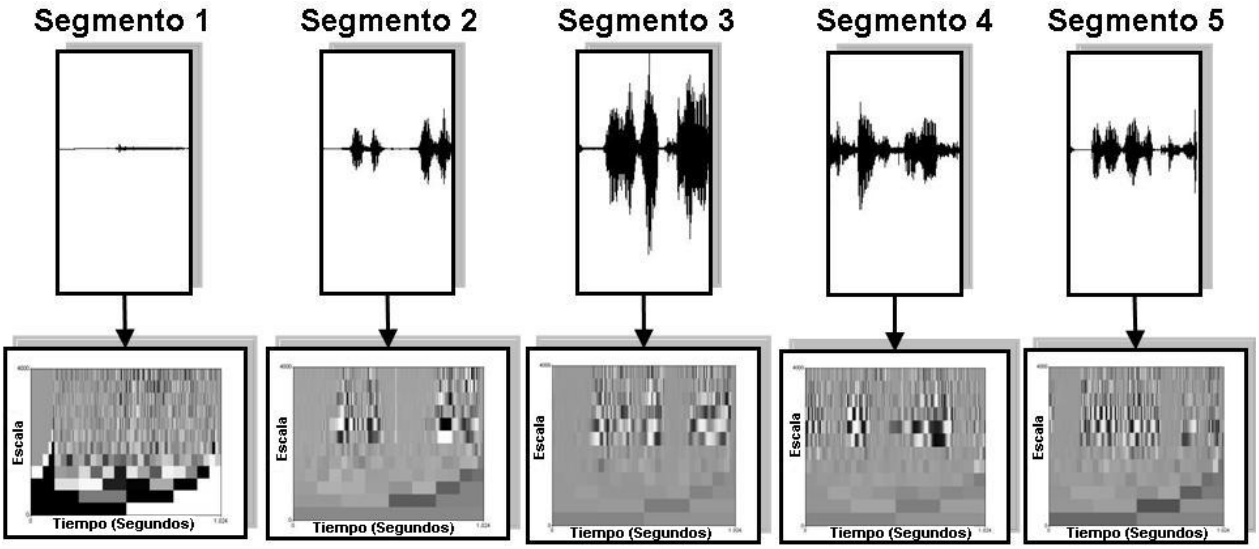


Figura 4.8. Segmentos de la señal SP-9.story de 5 segundos y sus transformadas wavelet representadas en escalogramas

4.3. Truncado por Fracción

El truncado por umbral es una técnica usada en compresión y eliminación de ruido de señales e imágenes, dado que las wavelets descomponen las señales en subbandas, el ruido es separado, además valores frecuenciales y de intensidad, imperceptibles para el oído y ojo humano son cercanos a cero. La idea es eliminar o establecer a cero aquellos coeficientes que no cumplen con cierto requerimiento en este caso un valor de umbral. Nuestro truncado no busca como meta principal eliminar ruido o la compresión, la meta es una representación del ritmo de los lenguajes, ubicado supuestamente en las bajas frecuencias de la voz.

El tipo de truncado que se utiliza es llamado truncado por fracción, pues se conserva una fracción del conjunto original de datos inalterado, mientras los demás se establecen a cero debido a cierta condición, mientras en el truncado por umbral no se sabe que fracción se conservará. De los coeficientes wavelet que resultan del cálculo, los coeficientes de mayor magnitud representan las bajas frecuencias y los de menor magnitud las altas frecuencias. Por tal razón, usamos un truncado con una fracción del 1%, esto nos permite usar solo el 1% de los coeficientes de mayor magnitud, una cantidad ya probada en el trabajo de [Reyes 2007], que se cree representa el ritmo, y que logra que dichos coeficientes aporten el mayor peso de información para la siguiente parte del proceso.

Ejemplificado en el escalograma los coeficientes de mayor magnitud son los de colores más intensos (oscuros y blancos) como se dijo anteriormente cada coeficiente parece una teja, así se conservaran alrededor de 82 tejas de tiempo-frecuencia (1% de 8192).

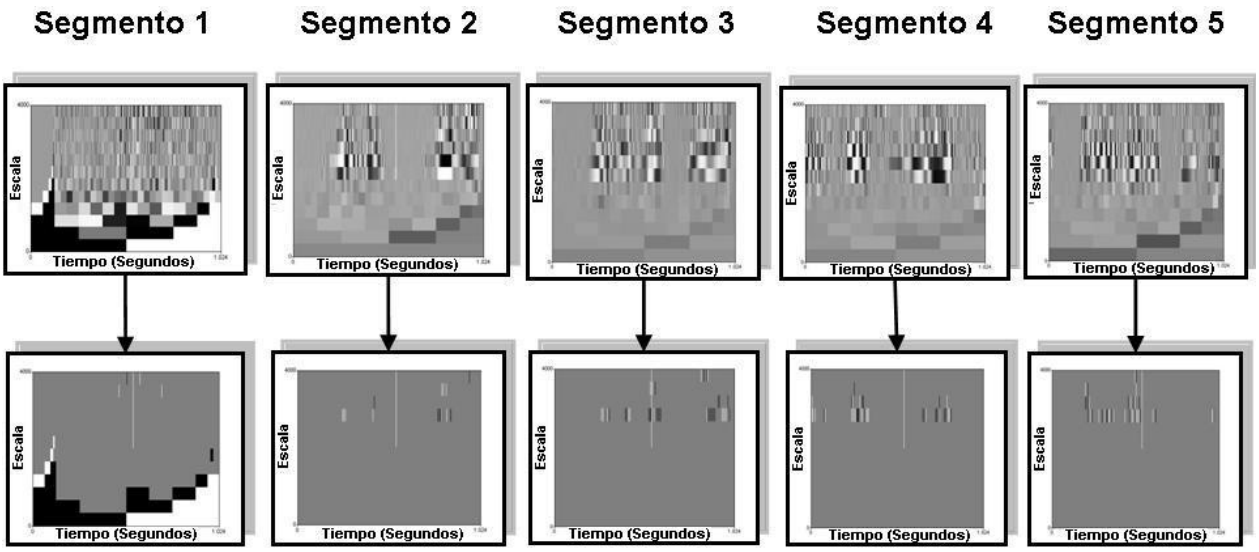


Figura 4.9. Las transformadas wavelet de los 5 segmentos son truncadas

Como se muestra en la figura 4.9 el 99% de los coeficientes son cero debido al truncado, un truncado por umbral puede aplicarse como alternativa, solo si se sabe que buscar.

4.4. Medidas Estadísticas

La problemática en cuestión es el manejo de la gran cantidad de coeficientes, por ejemplo 245760 para cada muestra de 30 segundos, lo cual nos lleva a enfrentarnos con la dimensionalidad de los datos [Donoho 2000] y como una primera manera de reducirlos y a la vez describirlos para caracterizar la señal, se emplean las medidas estadísticas de tendencia central, que se han utilizado como una manera de reducir la dimensión capturando el contenido original de los datos [Fodor 2002]. Las medidas utilizadas en este trabajo fueron: media de los coeficientes, desviación estándar de los coeficientes, máximo valor de los coeficientes y mínimo valor de los coeficientes. De tal manera que teniendo por ejemplo 245760 coeficientes o características wavelet, se reducen a 32768 atributos estadísticos, ya que tenemos 8192 atributos por cada medida estadística (media, desviación estándar, máximo, mínimo) (Tabla 4.1).

Tabla 4.1. Medidas estadísticas aplicadas a cada uno de los coeficientes

Descripción	formulas	Cantidad
Media	$Pc_i = \frac{1}{S}\sum_{k=1}^{S} c_{ik}$ Donde i =1,2,3, …, 8192; S = Número de segundos	8192
Desviación estándar	$Dc_i = \sqrt{\frac{1}{S-1}\sum_{k=1}^{S}\left(c_{ik} - Pc_i\right)^2}$ Donde i =1,2,3, …, 8192; S = Número de segundos	8192
Máximo	$Maxc_i = \max_{k=1}^{S}(c_{ik})$ Donde i =1,2,3, …, 8192; S = Número de segundos	8192
Mínimo	$Minc_i = \min_{k=1}^{S}(c_{ik})$ Donde i =1,2,3, …, 8192; S = Número de segundos	8192
Total de atributos		32768

4.5. Reducción de la Dimensionalidad

Una vez que las características estadísticas han sido extraídas, se construyen los clasificadores en formato weka[7], a pesar de la reducción mediante variables estadísticas, las instancias son muy grandes, y se procede a aplicar ganancia de información para seleccionar a los atributos más significativos en cada clasificador, esta técnica como se menciona en la sección 2.18, funciona como filtro, el umbral elegido para filtrar fue de 0, lo que indica que aquellos que tengan ganancia de información 0 o debajo de cero serán eliminados. La aplicación de esta técnica mediante código puede ser vista en la sección B.5.

[7] Formato de archivo de datos para trabajar en weka, donde estos datos se pueden manejar con las diferentes herramientas de Minería de Datos y Aprendizaje de Maquina que weka posee.

La técnica permitió reducir el número de atributos de 32768 a un rango entre 700 a 150 aproximadamente. Más precisamente, en las muestras de 30 segundos el clasificador Inglés-Alemán se reduce a 546 atributos, mientras que el clasificador Tamil-Farsi se reduce a 661. Asimismo, en las muestras de 10 segundos el clasificador Inglés-Alemán se reduce a 241, mientras que el clasificador Tamil-Farsi se reduce a 280 atributos, la tendencia indica que con menor tiempo menos atributos son requeridos.

4.6. Clasificador Mediante Validación Cruzada

La validación cruzada es un método utilizado en estadística, el cual divide en subconjuntos iguales, a un conjunto de datos, donde cada subconjunto es analizado o clasificado (en nuestro caso) con el resto, lo que permite analizar un conjunto completo sin tener que dividirlo y solo analizar una parte. De esta forma fue posible clasificar todas las muestras de cada clasificador binario, al ser 9 idiomas se generan 36 clasificadores distintos.

$$c_{NB_1} = \underset{c_j \in [Inglés, Alemán]}{\arg\max} P(c_j) \prod_i P(a_i \mid c_j)$$

$$c_{NB_2} = \underset{c_j \in [Inglés, Español]}{\arg\max} P(c_j) \prod_i P(a_i \mid c_j)$$

$$\vdots$$

$$c_{NB_{36}} = \underset{c_j \in [Tamil, Farsi]}{\arg\max} P(c_j) \prod_i P(a_i \mid c_j)$$

La configuración para la validación cruzada usando Naive Bayes fue de 10 subconjuntos (folds) (ver sección 2.6). La implementación en código de esta parte puede ser vista en la sección B.5

4.7. Sistema con Eliminación de Pausas

Una problemática que se encontró en la base de datos fue la presencia de grandes espacios sin la presencia de voz, pausas o silencios que se pensó que no aportarían nada al nivel de identificación alcanzado en la metodología anterior. Por tal motivo se llevó a cabo un estudio rápido de las señales para tratar de eliminar esa dificultad. El nuevo sistema se basa en los puntos

anteriores pero antes de todo se realiza eliminación de pausas como preprocesamiento (ver Figura 4.1) representado por el bloque punteado, el cual se describe a continuación (Figura 4.10).

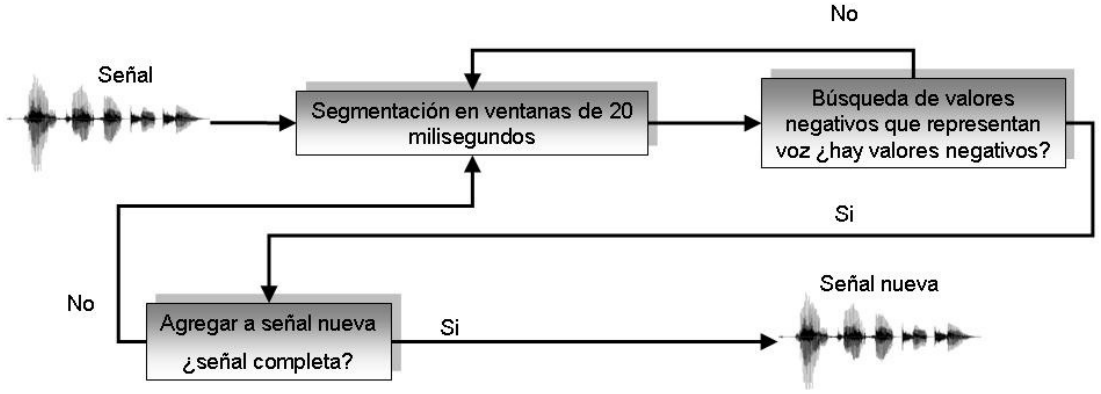


Figura 4.10. Bloque de eliminación de pausas

4.7.1. Descripción de Eliminación de Pausas

El objetivo en este preprocesamiento de la base de datos OGI_TS es eliminar las pausas largas de las muestras de audio.

Para ello se implemento un algoritmo de Detección de Voz Activa (VAD) simple. En la eliminación de pausas largas se ha utilizado un método propio sobre la base de datos telefónica OGI_TS; dadas las siguientes condiciones encontradas en los archivos de audio se planteo el algoritmo:

a) El flujo de habla y sonidos emitidos por el sistema de habla, produce cambios en la presión del aire, lo que quiere decir que existen cruces por cero en gran cantidad (de positivo a negativo).

b) Las pausas o silencios no producen mucho cambios solo producen valores desde 0 hasta algún valor positivo y si al caso algún cruce por cero sin llegar a ser voz.

La elaboración del algoritmo implica entonces la búsqueda de valores negativos que indican el cambio de la presión del aire, para tal efecto se usa una ventana de búsqueda de tamaño estándar en el análisis de voz de 20 milisegundos, que equivale a 160 *muestras* (valores producto

del muestreo, $Nm = \frac{milisegundos}{1000} \cdot Fm$, donde *Nm* es el número de muestras y *Fm* es la *frecuencia de muestreo*) de la señal, dada su *frecuencia de muestreo* de 8000 Hz, el algoritmo queda entonces:

Algoritmo VAD para eliminación de pausas

1.-Leer señal
2.-Calcular segmento de 20 ms
3.-Para 1 hasta tamaño de la señal
    3.1.-Obtener segmento
    3.2.-Buscar valores negativos en el segmento
    3.3.-Si existen
        3.3.2.-Agregar a nueva señal
    3.4.-Fin si
    3.5.-Siguiente segmento
4.-Fin para
5.-Guardar señal nueva

El algoritmo fue implementado en el lenguaje de programación de laboratorio Matlab (Matrix Laboratory). Dada su capacidad de manejo de matrices y vectores, resulta sencillo el manejo de los vectores de voz, su excelente graficado en 2D, permite observar los resultados de este reprocesamiento sobre OGI, y las funciones sobre archivos wav permiten el manejo de las señales de audio de manera eficiente. Este algoritmo fue aplicado a toda la base de datos, el resultado fue una nueva base de datos con algunos archivos reducidos, los demás son poco afectados o sin afectación, debido a la fluidez de los hablantes. Ejemplo de un audio reducido de 50 a 30 milisegundos

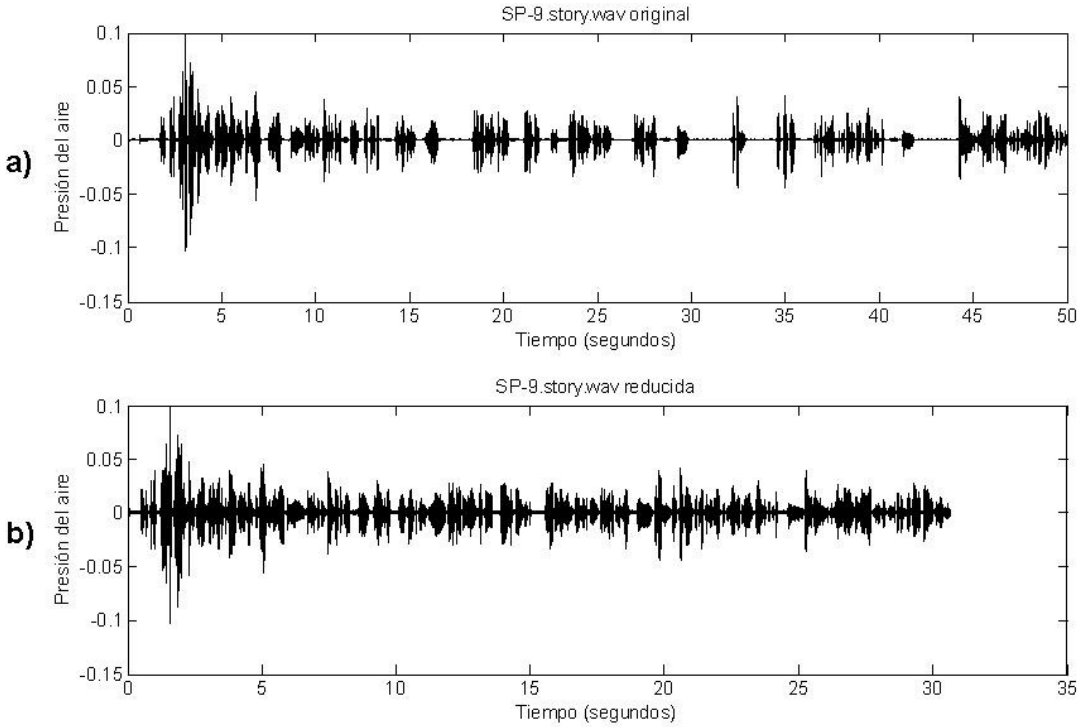


Figura 4.11. Señal original de SP-9.story.wav a) y se versión reducida b)

En apariencia, La figura 4.11 a) es una señal distinta a 4.11 b), pero es solo que fueron recorridos los bloques de habla, pero haciendo un acercamiento:

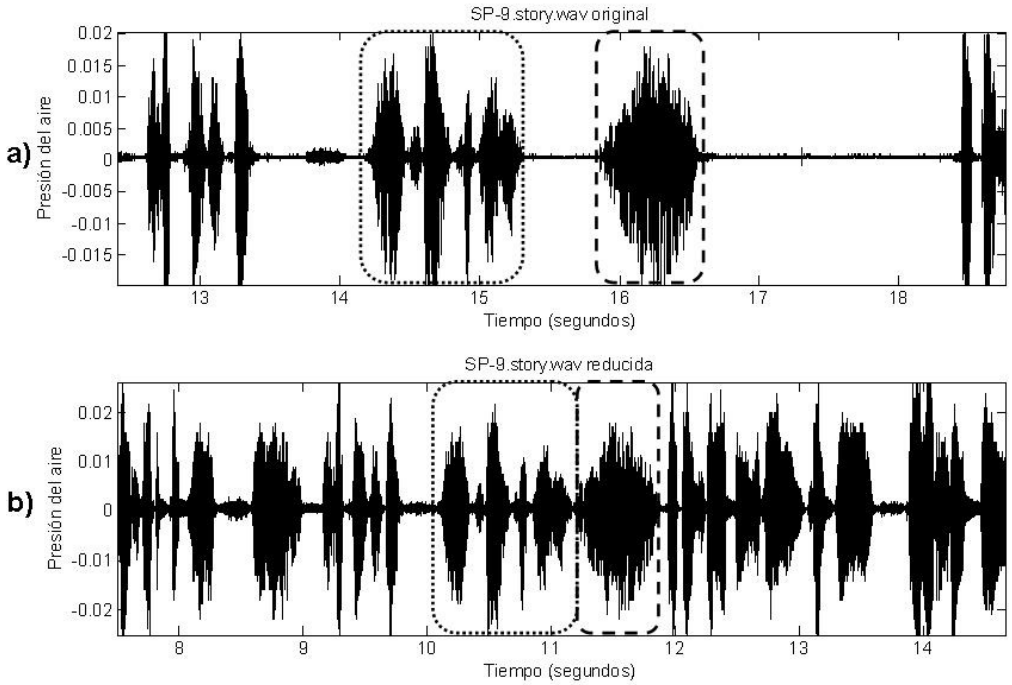


Figura 4.12. Se muestra un acercamiento de SP-9, de la señal original a) y la señal reducida b).

Como puede verse los bloques encerrados en los recuadros del mismo punteado son el mismo bloque de voz de la señal SP-9.story.wav, lo que quiere decir que el espacio entre los bloques de líneas punteadas de la señal original (figura 4.12 a)) fue eliminado, por cumplir con la condición de no tener valores negativos, haciendo un acercamiento a lo que sucede entre el segundo 15.3 y 15.9 de la señal original, observamos:

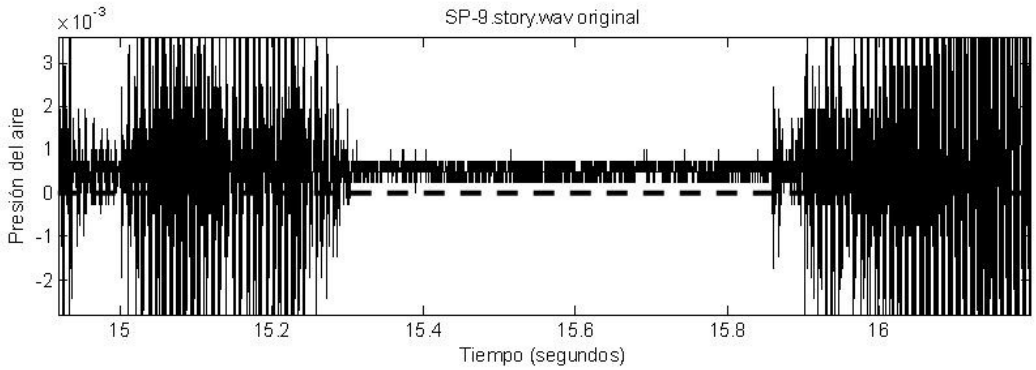


Figura 4.13. Se muestra un acercamiento de la señal original SP-9 entre el segundo 15.2 y 16.

La Figura 4.13, muestra el acercamiento que corresponde al espacio entre los bloques de la figura 4.12 a), se ha trazado una línea recta punteada, a la altura de cero y puede observarse que esta pausa, no cruza nunca el cero, solo una oscilación alcanza 0, ni siquiera hay un pequeño cruce, por lo cual el algoritmo lo eliminó.

Es necesario decir que este algoritmo considera que si existe solo un valor negativo se debe eliminar, pues en pausas existen algunos valores que no son voz y producen el cruce. Pocos son los archivos que son afectados de gran manera, pues hay pausas que contienen sonidos originados por el sistema de habla de muy baja intensidad, ejemplo

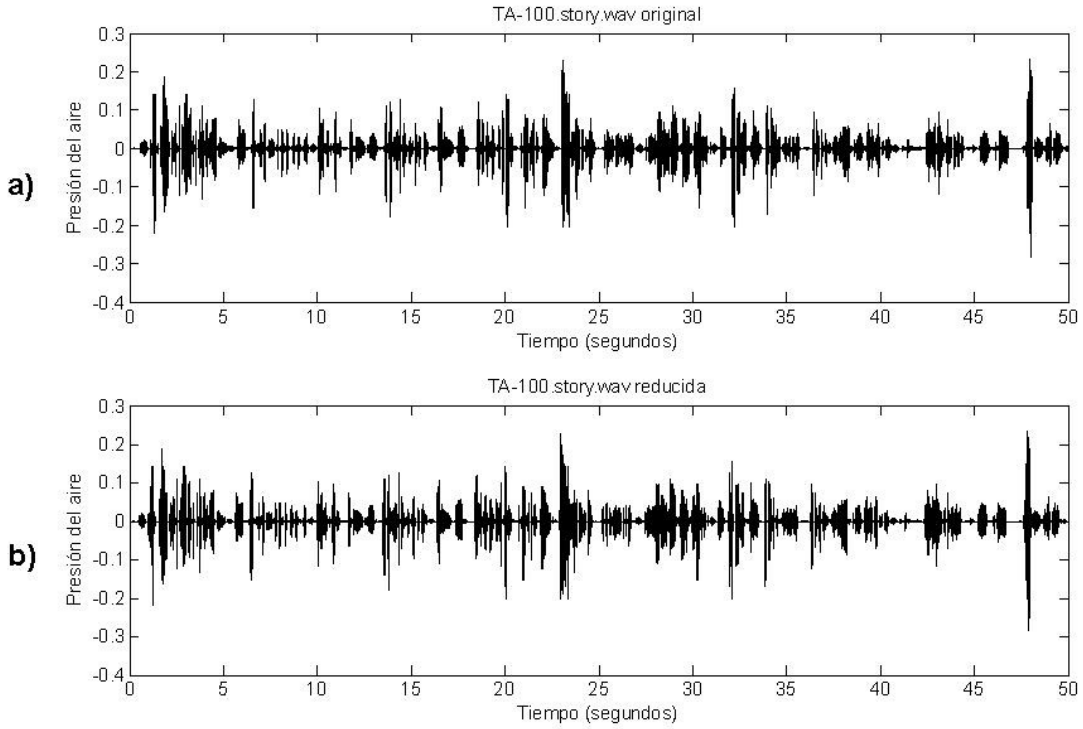


Figura 4.14. Señal TA-100.story normal a) y reducida (sin efecto alguno) b)

La señal TA-100.story.wav no es afectada de manera alguna por el algoritmo, pues en sus pequeñas pausas existen cruces por cero, producto de alargamiento de sonidos o respiración.

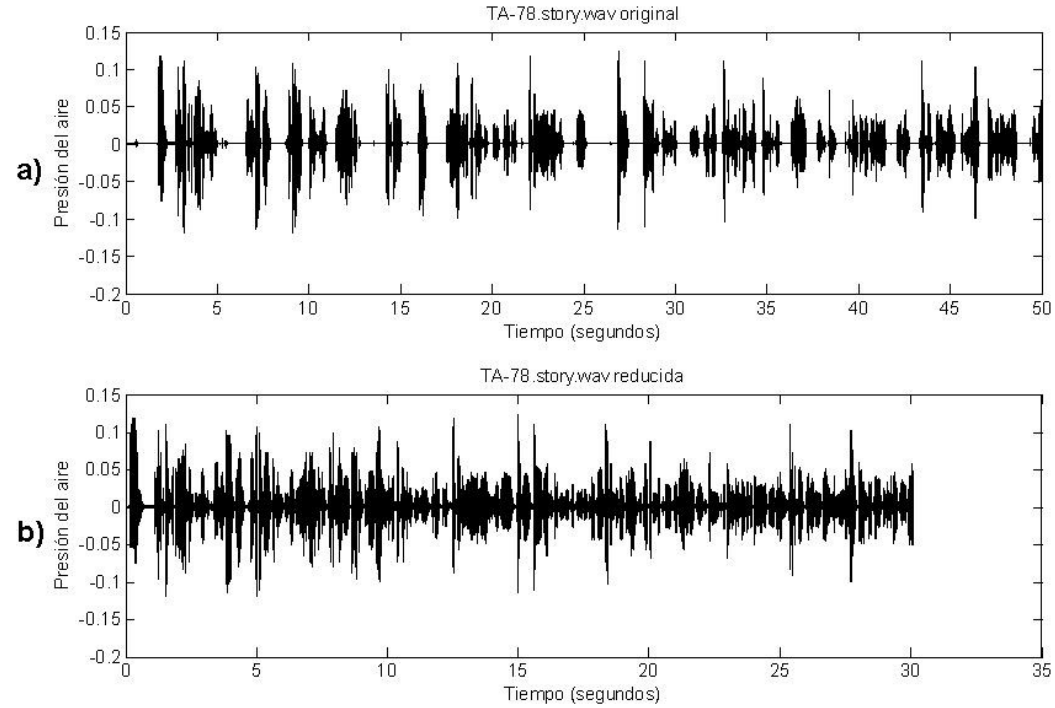


Figura 4.15. Señal TA-78.story.wav normal a) y su versión reducida b)

Otro caso de silencios muy grandes es el archivo TA-78.story (figura 4.15), al igual que en SP-9.story, la señal en los silencios no produce cambio en la presión del aire y por tal razón son eliminados tales silencios o pausas.

Algoritmo 2

Como la reducción de las pausas largas es completo y los demás archivos de audio tienen pausas cortas se modificó el algoritmo, conservando cierto tamaño de pausa para hacer las señales reducidas similares a las demás, el cambio radica en que si se detecta silencio se conserva por separado y mientras se detecta voz se sigue almacenando, si un segmento de voz es detectado, la pausa acumulada se vaciará solo en el tamaño de pausa establecido si es mayor se elimina el resto y si es menor se vacía completo, inmediatamente se coloca el segmento de voz detectado.

Algoritmo 2 de eliminación de pausas

1.-Leer señal
2.-Calcular segmento de 20 ms
3.-Para 1 hasta tamaño de la señal
    3.1.-Obtener segmento
    3.2.-Buscar valores negativos en el segmento
    3.3.-Si existen
        3.3.1.-Si hay pausa y es menor a pausa establecida
            3.3.1.1.-Agregarla
        3.3.2.-De otro modo
            3.3.2.1 Agregar solo el tamaño de pausa establecida y eliminar el resto
        3.3.2.-Fin si
        3.3.3.-Agregar a nueva señal el segmento de voz
    3.4.-De otra forma
        Guardar pausa
    3.5.-Fin si
    3.6.-Siguiente segmento
4.-Fin para
5.-Guardar señal nueva

Un ejemplo del efecto de este algoritmo es la señal SP-9.story.wav

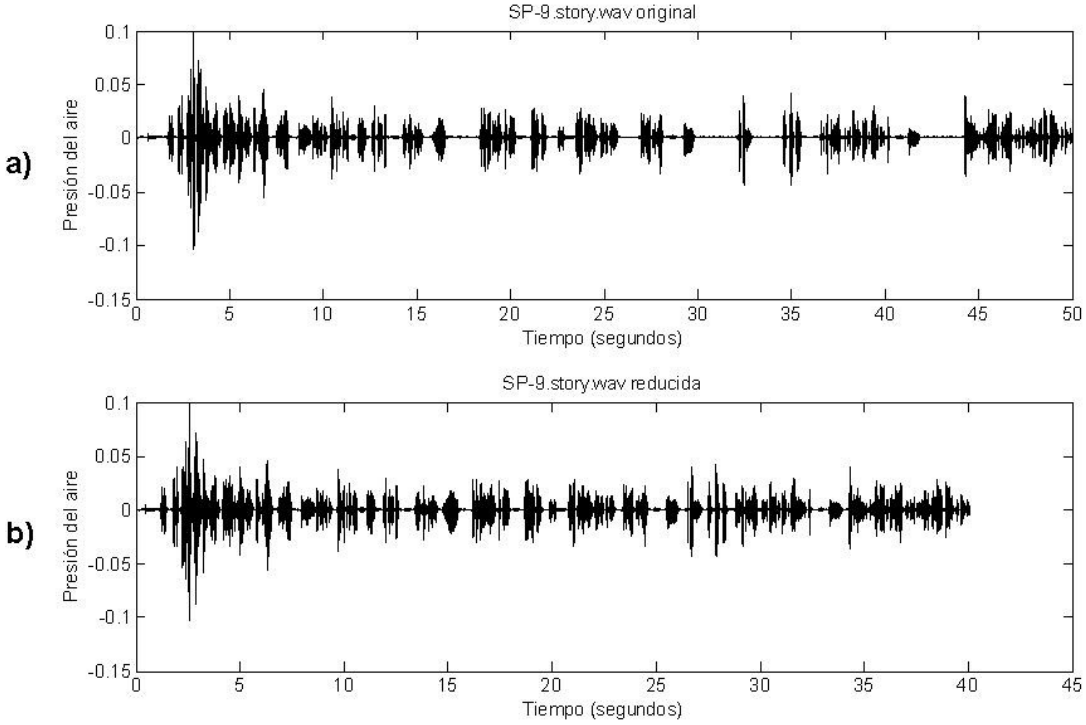


Figura 4.16. Señal SP-9.story.wav normal a) y la versión reducida de la misma b)

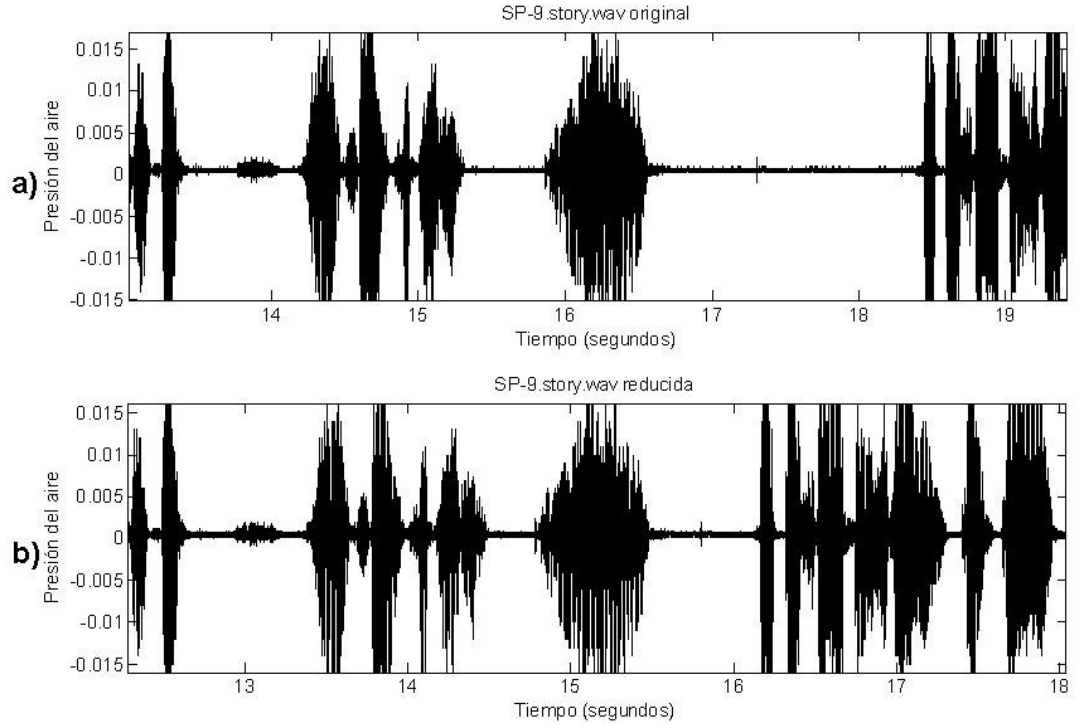


Figura 4.17. Acercamiento a una sección de la señal SP-9.story.wav normal a) y la versión reducida de la misma b)

La figura 4.16 muestra el efecto del algoritmo, que a simple vista cumple su objetivo, se conservan los segmentos y reducen las pausas, pero no las elimina por completo. En la figura 4.17 se observa claramente que los bloques son conservados solo las pausas son reducidas a un tamaño estándar o conservadas si son menores al estándar.

El algoritmo cumple su objetivo en reducción de pausas largas, la eliminación de las pausas completas es una tarea aun pendiente, pues requiere más que el simple estudio de la oscilación de la señal, por la presencia de ruido, respiración, etc., para ello existen técnicas de detección de voz activa que van desde los cruces por cero hasta los wavelets.

El siguiente paso es aplicar toda la metodología anterior mostrada en la figura 4.1 en el principio de este capítulo.

Capítulo 5

Pruebas y Resultados

Las pruebas son una parte importante de la investigación, así se define un conjunto de pruebas para validar el funcionamiento de la metodología empleada, sus resultados nos darán los parámetros para saber si se han conseguido los objetivos.

5.1. La Base de Datos OGI_TS

La OGI_TS es una base datos telefónica concebida con el fin de investigar la identificación automática de lenguas y el reconocimiento de habla multilingüe. Está integrada por 22 idiomas presentes en los Estados Unidos [Lander et al. 1995], mediante llamadas telefónicas, las cuales son grabadas; la base de datos se compone de 3 grupos principales de grabaciones, a) Respuestas a información específica, b) habla continua de tópicos selectos y c) habla improvisada.

En nuestro caso, usamos el tercer grupo, el de habla improvisada o espontánea, de la cual se toman 50 muestras por idioma, de una duración de 50 segundos de los siguientes 9 idiomas: Inglés, Alemán, Español, Mandarín, Vietnamita, Japonés, Coreano, Tamil y Farsi, en total 450 hablantes diferentes.

5.2. Tamaño de la Muestra de Habla, Eliminación de Pausas y Número de Clasificadores

Uno de nuestros objetivos ha sido el uso de muestras cortas de habla, por tal motivo, no utilizamos las muestras de habla completas solo 30, 10, 5 y 4 segundos de la muestra de 50 segundos original. También se realizaron pruebas agregando la eliminación de pausas largas mencionada en la metodología del sistema. En nuestra experimentación, se contempla la clasificación por pares de lenguajes, al ser 9 idiomas, se generan por combinación sin repetición, 36 clasificadores distintos por ejemplo: Inglés-Alemán, Inglés-Español,…,Coreano-Farsi, Tamil-Farsi. Cada clasificador cuenta con 100 muestras de habla (50 de cada idioma). Para evaluar las 100 muestras de habla se utiliza validación cruzada de 10 conjuntos de prueba y Naive Bayes como método de clasificación.

5.3. Resultados de la Experimentación

Las transformadas wavelet Db2 son calculadas con el programa de tratamiento de habla *Praat* [Boersma y Weeink, 2002]. Cabe aclarar que para realizar esta misma tarea, también usamos *Matlab* (Ver Anexo B.2), pero por cuestiones prácticas preferimos usar *Praat* (Ver anexo B.1). Los resultados de nuestra experimentación los organizamos en tablas, en donde cada casilla indica el porcentaje de exactitud de los dos idiomas. Cabe mencionar que nuestros resultados los obtuvimos usando el software *weka* [Witten y Frank 2005]. En las tablas que mostramos, en cada casilla ponemos los resultados de la muestra

Tabla 5.1. Porcentajes de clasificación con muestras normales de 30 segundos

	Alemán	**Español**	**Mandarín**	**Vietnamita**	**Japonés**	**Coreano**	**Tamil**	**Farsi**
Inglés	81	83	88	84	91	82	83	92
Alemán	-	94	95	95	93	91	84	84
Español	-	-	96	92	91	91	91	88
Mandarín	-	-	-	93	93	91	86	86
Vietnamita	-	-	-	-	90	94	80	93
Japonés	-	-	-	-	-	93	88	88
Coreano	-	-	-	-	-	-	86	85
Tamil	-	-	-	-	-	-	-	86

Tabla 5.2. Porcentajes de clasificación con muestras normales de 10 segundos

	Alemán	**Español**	**Mandarín**	**Vietnamita**	**Japonés**	**Coreano**	**Tamil**	**Farsi**
Inglés	88	97	89	91	87	86	92	92
Alemán	-	91	90	94	87	96	91	91
Español	-	-	91	90	91	94	89	95
Mandarín	-	-	-	90	94	88	90	93
Vietnamita	-	-	-	-	94	94	86	94
Japonés	-	-	-	-	-	93	87	92
Coreano	-	-	-	-	-	-	85	93
Tamil	-	-	-	-	-	-	-	93

Tabla 5.3. Porcentajes de clasificación con muestras normales de 5 segundos

	Alemán	**Español**	**Mandarín**	**Vietnamita**	**Japonés**	**Coreano**	**Tamil**	**Farsi**
Inglés	93	95	93	96	95	88	95	100
Alemán	-	94	93	92	93	95	94	92
Español	-	-	92	92	93	95	95	93
Mandarín	-	-	-	89	96	91	93	95
Vietnamita	-	-	-	-	90	92	93	95
Japonés	-	-	-	-	-	95	92	94
Coreano	-	-	-	-	-	-	87	91
Tamil	-	-	-	-	-	-	-	92

Tabla 5.4. Porcentajes de clasificación con muestras normales de 4 segundos

	Alemán	**Español**	**Mandarín**	**Vietnamita**	**Japonés**	**Coreano**	**Tamil**	**Farsi**
Inglés	94	96	90	94	97	90	95	97
Alemán	-	96	93	92	95	96	93	91
Español	-	-	94	92	94	92	89	95
Mandarín	-	-	-	91	96	96	92	96
Vietnamita	-	-	-	-	93	90	94	97
Japonés	-	-	-	-	-	93	97	94
Coreano	-	-	-	-	-	-	91	94
Tamil	-	-	-	-	-	-	-	89

Aunque los resultados parecen ser parecidos entre las diferentes pruebas las de menor tiempo tienen los mejores resultados una grafica de barras de las 4 pruebas anteriores da una idea de esta tendencia.

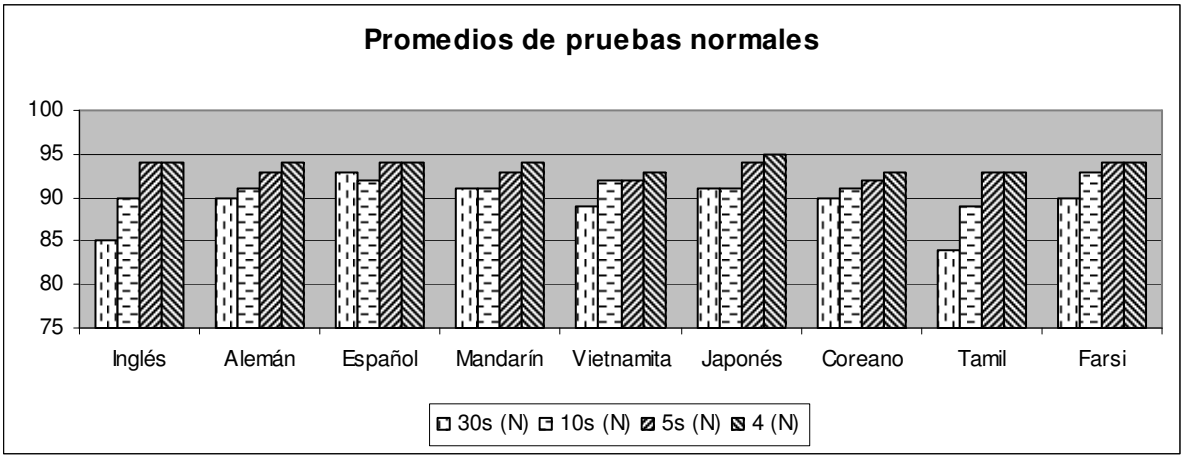


Figura 5.1. Gráfica que muestra las 4 pruebas normales por idioma

Las siguientes tablas muestran las mismas pruebas pero con archivos de audio sin pausas largas, con el fin de observar el efecto de la eliminación de dichas pausas en los resultados.

Tabla 5.5. Porcentajes de clasificación de muestras de 30 segundos (Eliminación de pausas)

	Alemán	Español	Mandarín	Vietnamita	Japonés	Coreano	Tamil	Farsi
Inglés	81	83	88	84	91	82	83	92
Alemán	-	94	95	95	93	91	84	84
Español	-	-	96	92	91	91	91	88
Mandarín	-	-	-	93	93	91	86	86
Vietnamita	-	-	-	-	90	94	80	93
Japonés	-	-	-	-	-	93	88	88
Coreano	-	-	-	-	-	-	86	85
Tamil	-	-	-	-	-	-	-	86

Tabla 5.6. Porcentajes de clasificación de muestras de 10 segundos (Eliminación de pausas)

	Alemán	Español	Mandarín	Vietnamita	Japonés	Coreano	Tamil	Farsi
Inglés	92	91	89	87	91	89	94	94
Alemán	-	91	89	95	82	93	89	90
Español	-	-	93	88	89	92	89	89
Mandarín	-	-	-	92	94	91	86	92
Vietnamita	-	-	-	-	91	87	87	92
Japonés	-	-	-	-	-	88	86	89
Coreano	-	-	-	-	-	-	87	93
Tamil	-	-	-	-	-	-	-	87

Tabla 5.7. Porcentajes de clasificación de muestras de 5 segundos (Eliminación de pausas)

	Alemán	Español	Mandarín	Vietnamita	Japonés	Coreano	Tamil	Farsi
Inglés	92	96	91	94	89	94	93	92
Alemán	-	93	89	96	90	93	94	95
Español	-	-	91	95	91	91	91	91
Mandarín	-	-	-	90	96	93	91	97
Vietnamita	-	-	-	-	86	97	89	94
Japonés	-	-	-	-	-	90	93	93
Coreano	-	-	-	-	-	-	89	93
Tamil	-	-	-	-	-	-		98

Tabla 5.8. Porcentajes de clasificación de muestras de 4 segundos (Eliminación de pausas)

	Alemán	Español	Mandarín	Vietnamita	Japonés	Coreano	Tamil	Farsi
Inglés	93	97	90	93	94	94	94	93
Alemán	-	91	95	94	90	93	94	97
Español	-	-	92	90	96	92	96	94
Mandarín	-	-	-	90	94	94	94	97
Vietnamita	-	-	-	-	92	94	91	97
Japonés	-	-	-	-	-	91	95	93
Coreano	-	-	-	-	-	-	94	93
Tamil	-	-	-	-	-	-	-	94

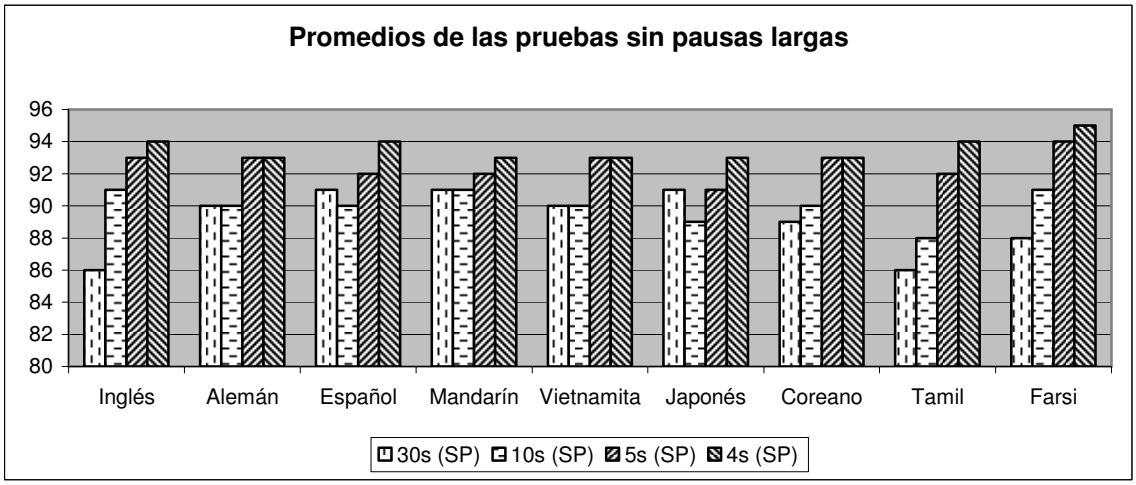


Figura 5.2. Gráfica de las pruebas con eliminación de pausas largas por idioma

Como en las pruebas anteriores se observa en la gráfica 5.2 la tendencia de menos tiempo mayor nivel de clasificación correcta.

5.4. Promedios de las 8 Pruebas

Para ver de una manera más simple los resultados de los experimentos se presentan los resultados por idioma en la siguiente tabla, donde aparece la prueba normal (N) y su compañera sin pausas (SP).

Tabla 5.9. Se muestra los porcentajes de clasificación por idioma de todos los experimentos

Prueba	Inglés	Alemán	Español	Mandarín	Vietnamita	Japonés	Coreano	Tamil	Farsi
30s (N)	85	90	93	91	89	91	90	84	90
30s (SP)	86	90	91	91	90	91	89	86	88
10s (N)	90	91	92	91	92	91	91	89	93
10s (SP)	91	90	90	91	90	89	90	88	91
5s (N)	94	93	94	93	92	94	92	93	94
5s (SP)	93	93	92	92	93	91	93	92	94
4s (N)	94	94	94	94	93	95	93	93	94
4s (SP)	94	93	94	93	93	93	93	94	95

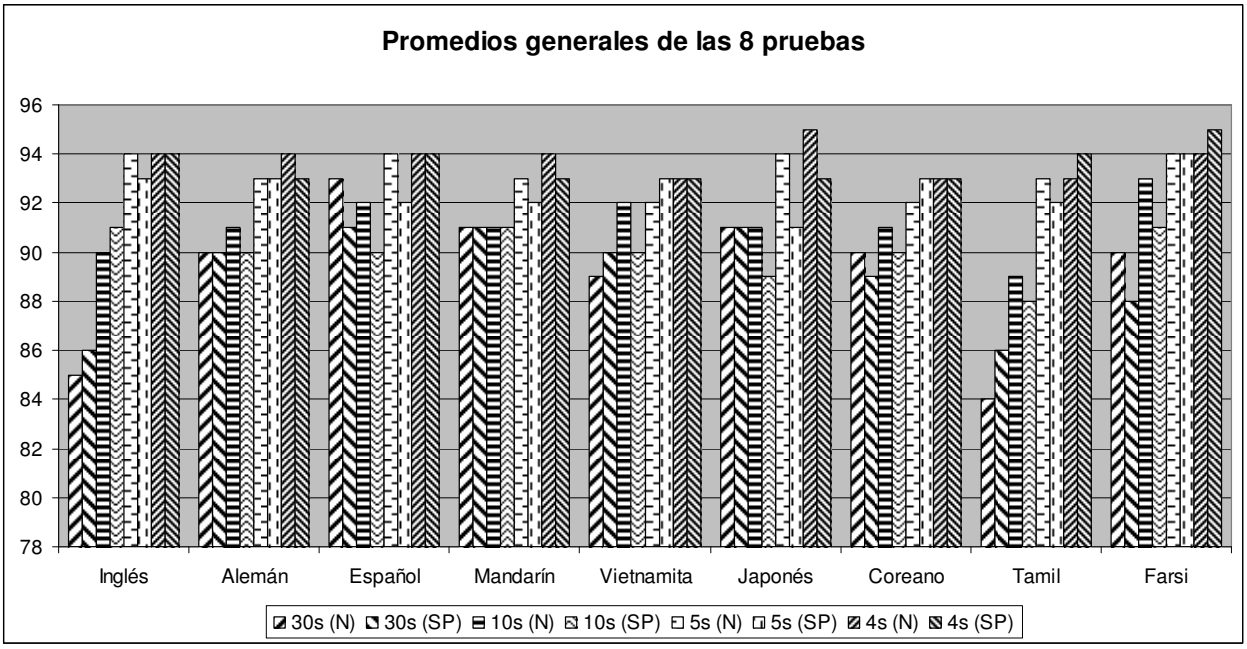


Figura 5.3. Porcentajes de clasificación por idioma en una gráfica de barras

5.5. Conclusiones

La metodología de solución permite lograr niveles de clasificación buenos, pero no superan los mostrados en el estado del arte por Reyes, con respecto a la prueba de 50 segundos.

Se tienen los mejores resultados arriba de 92% usando muestras cortas de habla de 5 y 4 segundos de habla, además de requerir menos atributos para lograr una buena clasificación de las lenguas. Esto marca una tendencia clara en los resultados de las pruebas comparados con los resultados de Reyes. La tendencia observada es contraria a la que muestra el trabajo de Reyes, es decir que en lugar de ser mayor nivel de clasificación con más tiempo, es menor, pero aumenta si se disminuye el tiempo, lo que nos ofrece la posibilidad de construir una aplicación real.

El algoritmo de eliminación de pausas logra quitar las pausas largas sin perder la información, las pruebas usando estas señales logran casi los mismos niveles de clasificación que las pruebas normales, recordando que es habla efectiva y que no corresponden estrictamente al mismo segmento de habla de las pruebas normales.

Con excepción de las pruebas de 5 y 4 segundos de cualquier naturaleza, de acuerdo a nuestra investigación la combinación de cualquier lengua con Tamil resulta en los niveles más bajos de clasificación que son arriba de 85% de clasificación, lo que indica que es una lengua difícil de clasificar y puede ser observado en la tabla 5.8.

5.6. Contrastes y Aportaciones Principales

5.6.1. Contrastes

La presente investigación presenta ciertas coincidencias y diferencias con la investigación de [Reyes 2007] las coincidencias son el uso de las Wavelets para extracción de características y usar el 1% de los coeficientes para extracción del ritmo.

La diferencia es el manejo de los coeficientes wavelet, agregando para esto la segmentación de la señal, en señales de un segundo, lo que permite el manejo de menos coeficientes para el total de la señal, así como, también el poder aplicar medidas estadísticas a los coeficientes.

La segmentación combinada con medidas estadísticas simples, logra que entre menos segundos se ocupen para analizar la señal, mayor es el porcentaje de clasificación correcta, esta es otra diferencia con respecto a Reyes.

Pero los resultados de Reyes disminuyen con menos tiempo de habla y los resultados de este trabajo aumentan con menos tiempo. El trabajo de Reyes tiene los mejores resultados con el uso de 50 segundos y este trabajo tiene buenos resultados con tan solo 4 segundos.

5.6.2. Aportaciones

Después de lo expuesto anteriormente las principales aportaciones de este proyecto son las siguientes:

La aportación de este proyecto es un método de identificación de lenguas que es independiente de información fonotáctica, tiene resultados de clasificación correcta arriba de 92%, con el uso mínimo de cuatro segundos de habla.

Otra aportación es que esta metodología es aplicable a un producto real, ya que solo necesita 4 segundos de habla para analizar, lo cual indica que es viable su implementación, con las dificultades que conlleva.

5.7. Trabajos Futuros

Existen un sin número de posibles cambios a la metodología que pueden bien ser aprovechados para futuros trabajos, con respecto a la segmentación, la wavelet elegida, la caracterización del ritmo mediante minería de datos, entre otros asuntos.

Pero en forma específica, un trabajo futuro es la multiclasificación, es decir dejar de usar solo 2 lenguas, y migrar a un mayor número de ellas, las pruebas realizadas indican que se tiene un buen nivel a pares, y es posible observar que sucede con un número mayor.

Un trabajo futuro es el uso de una función wavelet diferente a la Db2, para lo cual existe un número amplio de familias wavelet a elegir.

Otro es la eliminación de pausas con un algoritmo más sofisticado, que incluya otros aspectos como la energía, o wavelets, y logre la obtención de habla efectiva como pre-procesamiento al sistema de identificación de lenguas.

Para reforzar el método se sugiere un trabajo futuro sobre OGI_TS, usando segundos aleatorios de la señal evaluada e inicios aleatorios de dicha señal, para tomar un segmento a evaluar, con el fin de demostrar que el método funciona con cualquier muestra de habla de dicha señal.

La identificación de lenguas indígenas es un objetivo que se está buscando, por lo cual la construcción de una base de datos con un buen número de lenguas es un trabajo a futuro muy importante, que serviría para investigaciones futuras.

Anexo A

Wavelets Haar y Db2

El sistema de Haar el cual apareció a principios del siglo XX, fue extendido muchos años más tarde, después de la aparición del concepto de Wavelet y la unificación que hiciera Stephan Mallat de las técnicas para el cálculo de wavelets como los algoritmos piramidales de procesamiento de imágenes, filtros espejo en cuadratura de procesamiento digital de señales, y las bases ortonormales de Morlet en Sismología, fue posible para Ingrid Daubechies terminar el trabajo que sabiéndolo o no comenzó Haar, creando una base ortonormal de wavelets, la cual es llamada Db*n*.

A.1. ¿Cómo fueron encontradas estas Wavelets?

La idea principal está en la similitud de ellas mismas. Comienza con una función $\phi(x)$ que está compuesta por una versión más pequeña de ella misma. Esta es la ecuación de refinamiento (o doble escala, dilatación).

$$\phi(x) = \sum_{k=-\infty}^{\infty} a_k \phi(2x-k)$$

Las a_k's son llamadas coeficientes filtro o mascaras. La función $\phi(x)$ es llamada la función de escala y bajo ciertas condiciones,

$$\psi(x) = \sum_{-\infty}^{\infty} (-1)^k b_k \phi(2x-k) = \sum_{-\infty}^{\infty} (-1)^k a_{1-k} \phi(2x-k)$$

genera una wavelet.

A.2. ¿Cuáles son estas condiciones?

Primero la función de escala debe preservar su área bajo cualquier iteración, así, $\int_{-\infty}^{\infty}\phi(x)dx=1$, integrando la función de refinamiento entonces

$$\int_{-\infty}^{\infty}\phi(x)dx=\sum a_k\int_{-\infty}^{\infty}\phi(2x-k)dx=\frac{1}{2}\sum a_k\int_{-\infty}^{\infty}\phi(u)du$$

Por lo tanto $\sum a_k=2$. Así la estabilidad de la iteración fuerza a una condición sobre el coeficiente a_k . Segundo la convergencia de la wavelet requiere la condición $\sum_{k=0}^{N-1}(-1)^k k^m a_k=0$, donde $m=0,1,2,...,\frac{N}{2}-1$. Tercero requiriendo la ortogonalidad de las wavelets fuerza la condición $\sum_{k=0}^{N-1}a_k a_{k+2m}=0$, donde $m=0,1,2,...,\frac{N}{2}-1$. Finalmente si la función de escala requiere ortogonalidad $\sum_{k=0}^{N-1}a_k^2=2$. Resumiendo

Estabilidad

$$\sum_{k=0}^{N-1}a_k=2\,, \qquad \text{(A.1)}$$

Convergencia

$$\sum_{k=0}^{N-1}(-1)^k k^m a_k=0 \qquad \text{(A.2)}$$

Ortogonalidad de las wavelets

$$\sum_{k=0}^{N-1}a_k a_{k+2m}=0 \qquad \text{(A.3)}$$

Ortogonalidad de las funciones de escala

$$\sum_{k=0}^{N-1}a_k^2=2 \qquad \text{(A.4)}$$

Este tipo de wavelet está restringida a ser cero fuera de un pequeño intervalo, esto lo lleva a cabo la propiedad de soporte compacto, la ecuación de refinamiento asegura que la función wavelet sea no diferenciable en todas partes.

A.3. Generación de Haar y Daubechies wavelets

Para Haar el número de coeficientes es *N*=2. La condición de estabilidad fuerza $a_0 + a_1 = 2$, la condición de exactitud implica, $a_0 - a_0 = 0$, y la de ortogonalidad da $a_0^2 + a_1^2 = 2$, la única solución es que $a_0 = a_1 = 1$, entonces la ecuación de refinamiento queda

$$\phi(x) = \phi(2x) + \phi(2x-1) \tag{A.5}$$

La función de refinamiento es satisfecha por una función caja

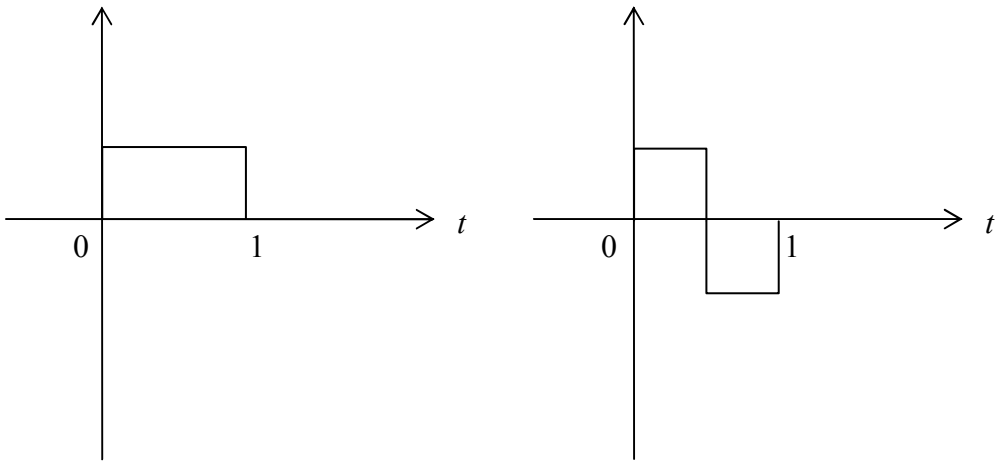
$$B(x) = \begin{cases} 1 & 0 \le x < 1 \\ 0 & \text{de otra forma} \end{cases} \tag{A.6}$$

Una vez que es elegida la función de refinamiento, obtenemos una wavelet simple, la función wavelet queda

$$\psi(x) = \phi(2x) - \phi(2x-1) \tag{A.7}$$

La función wavelet es satisfecha por una función de onda cuadrada

$$H(x) = \begin{cases} 1 & 0 \le x < \frac{1}{2} \\ -1 & \frac{1}{2} \le x < 1 \\ 0 & \text{de otra forma} \end{cases} \tag{A.8}$$



(a) $\phi(t)$ (b) $\psi(t)$

Figura A.1. Se muestra la función de refinamiento (a) y la función wavelet de Haar (b)

Todo esto para generar Haar o también llamada Db1, y ahora se obtiene Db2 la primera wavelet generada por Daubechies, las wavelet Dbn tienen un número de coeficientes del orden de $N=2n$, es decir db1 tiene $N=2$, Db2 tiene $N=4$, así que se generan las ecuaciones que cumplen con las propiedades mencionadas anteriormente.

Las ecuaciones para las mascaras o filtros de Db2 son:

$$a_0 + a_1 + a_2 + a_3 = 2 \tag{A.9}$$

$$a_0 - a_1 + a_2 - a_3 = 0 \tag{A.10}$$

$$-a_1 + 2a_2 - 3a_3 = 0 \tag{A.11}$$

$$a_0 a_2 + a_1 a_3 = 0 \tag{A.12}$$

$$a_0^2 + a_1^2 + a_2^2 + a_3^2 = 2 \tag{A.13}$$

Resolviendo el sistema

De (A.9) y (A.10)

$$a_0 + \cancel{a_1} + a_2 + \cancel{a_3} = 2$$
$$a_0 - \cancel{a_1} + a_2 - \cancel{a_3} = 0$$
$$2a_0 + 2a_2 = 2$$
$$a_0 + a_2 = 1;$$
$$a_2 = 1 - a_0 \qquad \text{(A.14)}$$

Multiplicando (A.10) por (-) tenemos

$$\cancel{a_0} + a_1 + \cancel{a_2} + a_3 = 2$$
$$-\cancel{a_0} + a_1 - \cancel{a_2} + a_3 = 0$$
$$2a_1 + 2a_3 = 2$$
$$a_1 + a_3 = 1;$$
$$a_3 = 1 - a_1 \qquad \text{(A.15)}$$

Sustituyendo (A.14) y (A.15) en (A.11)

$$-a_1 + 2(1 - a_0) - 3(1 - a_1) = 0$$
$$2a_1 - 2a_0 = 1;$$
$$a_1 = \frac{1 + 2a_0}{2} \qquad \text{(A.16)}$$

Sustituyendo (A.14) y (A.16) en (A.12)

$$a_0(1 - a_0) + \frac{1 + 2a_0}{2} a_3 = 0$$
$$a_3 = a_0(a_0 - 1)\frac{2}{1 + 2a_0}$$

$$a_3 = \frac{2a_0(a_0 - 1)}{1 + 2a_0} \tag{A.17}$$

Sustituyendo (A.14), (A.15) y (A.17) en (A.13) tenemos

$$a_0^2 + \frac{(1+2a_0)^2}{4} + (1-a_0)^2 + \frac{4a_0^2(a_0-1)^2}{(1+2a_0)^2} = 2$$

Eliminando el común denominador

$$4(1+2a_0)^2 a_0^2 + (1+2a_0)^2 + 4(1+2a_0)^2(1-a_0)^2 + 16a_0^2(a_0-1)^2 = 2 \cdot 4(1+2a_0)^2 = 8(1+2a_0)^2$$

Desarrollándolo por partes

Primer término

$$4(1+2a_0)^2 a_0^2 = 4(1+4a_0+4a_0^2)a_0^2 = 16a_0^4 + 16a_0^3 + 4a_0^2$$

Segundo término

$$(1+2a_0)^4 = (1+4a_0+4a_0^2)^2 = 1+16a_0^2+16a_0^4+8a_0+8a_0^2+32a_0^3 = 16a_0^4+32a_0^3+24_0^2+8a_0+1$$

Tercer término

$$4(1+2a_0)^2(1-a_0)^2 = 4(1+4a_0+4a_0^2)(1-2a_0+a_0^2) =$$
$$4 - 8\cancel{a_0} + 4\cancel{a_0^2} + 16\cancel{a_0} - 32\cancel{a_0^2} + 16\cancel{a_0^3} + 16\cancel{a_0^2} - 32\cancel{a_0^3} + 16_0^4$$
$$16a_0^4 - 16a_0^3 - 12_0^2 + 8a_0 + 4$$

Cuarto término

$$16a_0^2(a_0-1)^2 = 16a_0^2(a_0^2 - 2a_0 + 1) = 16a_0^4 - 32a_0^3 + 16a_0^2$$

Término del lado derecho de la ecuación

$$8(1+2a_0)^2 = 8(1+4a_0+4a_0^2) = 32a_0^2+32a_0+8$$

Formando el lado izquierdo de la ecuación

$$\begin{aligned} &16a_0^4+16a_0^3+4a_0^2 \\ &16a_0^4+32a_0^3+24_0^2+8a_0+1 \\ &16a_0^4-16a_0^3-12_0^2+8a_0+4 \\ &16a_0^4-32a_0^3+16a_0^2 \end{aligned}$$

Sumando los cuatro términos del lado izquierdo de la ecuación tenemos

$$64a_0^4+32a_0^2+16a_0+5$$

Igualándolo con el lado derecho tenemos

$$64a_0^4+32a_0^2+16a_0+5 = 32a_0^2+32a_0+8$$

$$64a_0^4-16a_0-3=0 \qquad \text{(A.18)}$$

Como se observa se tiene raíces cuartas, obteniendo estas raíces de (A.18) $a_0 = \frac{1+\sqrt{3}}{4}$

Sustituyendo a_0 en (A.16)

$$a_1 = \frac{1+2a_0}{2},$$

$$a_1 = \frac{1+2\frac{1+\sqrt{3}}{4}}{2} = \frac{1+\frac{1+\sqrt{3}}{2}}{2} = \frac{\frac{3+\sqrt{3}}{2}}{2} = \frac{3+\sqrt{3}}{4}$$

Sustituyendo a_1 en (A.15) tenemos

$$a_3 = 1 - a_1$$

$$a_3 = 1 - \left(\frac{3+\sqrt{3}}{4} \right) = \frac{4}{4} - \left(\frac{3+\sqrt{3}}{4} \right) = \frac{1-\sqrt{3}}{4}$$

Sustituyendo a_0 en (A.14) tenemos

$$a_2 = 1 - a_0$$

$$a_2 = 1 - \left(\frac{1+\sqrt{3}}{4} \right) = \frac{4}{4} - \frac{1+\sqrt{3}}{4} = \frac{3-\sqrt{3}}{4}$$

Las soluciones son $a_0 = \frac{1+\sqrt{3}}{4}$, $a_1 = \frac{3+\sqrt{3}}{4}$, $a_2 = \frac{3-\sqrt{3}}{4}$, $a_3 = \frac{1-\sqrt{3}}{4}$

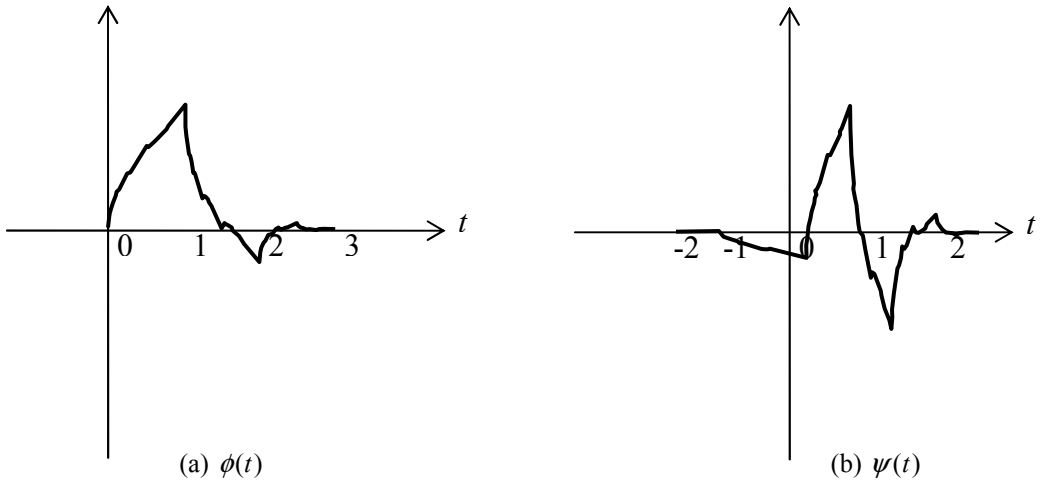


Figura A.2. Se muestra la función de refinamiento (a) y la función wavelet Db2(b)

Anexo B

Códigos

B.1. Código script “extractcoef.script” de transformadas wavelet mediante Praat

```
#Este programa se encarga de calcular la transformada wavelet db2 a
#señales de audio
#Código Jose Manuel Vargas, ITCM-DEPI 2008
#lee la lista de idiomas
Read Strings from raw text file... idiomas.txt
xx=1
id=1
final=0
#Manda a procesar 450 archivos de 50 en 50
while xx <450
#Toma el nombre del idioma dependiendo de “id”
select Strings idiomas
idioma$=Get string... 'id'
#se calcula el final del segmento
final=xx+49
#llama al procedimiento extraccion
call   extraction xx final 'idioma$'
xx=final +1
id=id+1
endwhile

#---------------------------------------------------------------

procedure extraction inicio fin nombre2$
#Código Jose Manuel Vargas, ITCM-DEPI

#Lee dos archivos con los nombres de los 450 archivos, la segunda lista
representa los nombres de los archivos, dentro de Praat, pues cambian
Read Strings from raw text file... archivos1.txt
Read Strings from raw text file... archivos2.txt
j=1
k=1
# este ciclo se encarga de mandar a cada nombre de audio al procedimiento
programage
for it from inicio to fin
```

```
 select Strings archivos1
 nombre$=Get string... 'it'
 select Strings archivos2
 nombre1$=Get string... 'it'
#Se manda a llamar a programege enviando le los nombres del archivo y el
idioma
 call programage 'nombre$' 'nombre1$' 'nombre2$'
# printline 'nombre$'
endfor
select Strings archivos1
Remove
select Strings archivos2
Remove
#select Strings idiomas
#Remove
endproc

procedure programage nombre$ nombre1$ nombre2$
#Este procedimiento segmenta la señal en segundos y aplica la transformada
Wavelet a cada uno de ellos
#Código Original: Ana Lilia Reyes Herrera-INAOE
#Modificación: José Manuel Vargas Martínez-ITCM-DEPI

#Carga una lista de números del 1 al 50, dado que no cuenta con funciones de
conversión de enteros a texto
Read Strings from raw text file... lista.txt
#se forma la ruta donde se guardaran los archivos ".wavelet"
ruta0$ = "OGI_Story_res\" + nombre2$ + "\"
#Ruta de la base de dato de archivos de audio
ruta$ = "Nueva_OGI_Story_50s\" + nombre2$ + "\"
printline 'ruta$'
#Ruta completa del archivo de audio a segmentar
archivo$ = ruta$ + nombre$ + ".wav"
#Lectura del archivo
Read from file... 'archivo$'

#este ciclo se encarga de segmentar el archivo, calcular su transformada
wavelet, truncarla y posteriormente guardarla
#el ciclo depende de la duracion de la señal o el tiempo deseado
for i from 0 to 44
select Sound 'nombre1$'
#Se extrae un segmento de 1 segundo desde i hasta i+1
Edit
  editor Sound 'nombre1$'
  Select... 'i' 'i'+1
  Extract selection (preserve times)
Close
#Se aplica la transformada wavelet
 To Wavelet... 4
#Se trunca en una fraccion del 1%
 Truncate by fraction... 0.01
 select Strings lista
 num$=Get string... 'i'+1
 select Wavelet untitled_1
#se forma el nombre del archive que sera guardado
 archivot$=ruta0$+ nombre$+ num$ +".wavelet"
#Se guarda
```

```
 Write to text file... 'archivot$'
 select Sound untitled
 Remove
 select Wavelet untitled
 Remove
 select Wavelet untitled_1
 Remove
Endfor

select Strings lista
Remove
select Sound 'nombre1$'
Remove
Endproc
```

B.2. Código "extractcoef.m" para calcular las transformadas wavelet db2, mediante Matlab

```
function extractcoef
%Esta funcion se encarga de calcular las transformadas wavelets
%(Coefcientes wavelet) de 450 archivos de audio
%Codigo: Jose Manuel Vargas Martinez, ITCM-DEPI
clear all;
Numero_elem=50;%Numero de señales por idioma
Numero_segun=45;%Numero de segundos por señal
f=fopen('idiomas.txt','r');%Archivo que contine los nombres de los idiomas
lang=textscan(f,'%s',9);%Son cargados
la=lang{:};% y convertidos
%este ciclo manda a procesar los 9 idiomas
for i=1:9
    %Coeficientes se encarga de procesar cada idioma o lengua
    coeficientes(char(la(i)), Numero_elem, Numero_segun);
end

%-------------------------------------------------------------------------

function coeficientes(idioma,elementos, s)
%Coeficientes se encarga de manejar cada lista de idioma, y manda a segmentar
cada una de las señales de dicho idioma o lengua
%Codigo: Jose Manuel Vargas Martinez, ITCM-DEPI
pf=fopen([idioma '.txt'],'r');%Se abre la lista de señales
C = textscan(pf, '%s',50);%se cargan a memoria
C=C{:};%Se extraen del cell
origen='OGI_Story_50s';%origen del archivo
destino='OGI_Story_res';%destino del archivo .wavelet
disp(idioma);
%El ciclo recorrera a todas las señales de la lengua y las enviara a
%segmentar
for i=1:elementos
    segmentacion(char(C(i)),s,origen,destino,idioma);
    disp(i)
end
fclose(pf);

%-------------------------------------------------------------------------
```

```
function segmentacion(nombre,s,origen,destino,idioma)
%Este programa se encarga de segmentar una señal de habla y descomponerna en
coeficientes wavelet db2, para su truncado, ademas del calculo de medidas
estadisticas
%Codigo: Jose Manuel Vargas Martinez, ITCM-DEPI
ruta=[origen '\' idioma '\' nombre '.wav']; %Ruta del archivo a segmentar
x=wavread(ruta); %Se carga a memoria
%punt=find(nombre=='.');
inicio=1; %Indice que se actualiza de 8000 en 8000
ad=zeros(192,1); %Ceros para rellenado del segmento
%ciclo que segmenta la señal, calcula la transformada wavelet, la trunca y la
guarda
for i=1:s
    tem=x(inicio:8000*i);%Se extrae un segmento de 1 segundo de inicio hasta
8000*i
    tem=[tem; ad]; %Se rellena de ceros, para que sea potencia de 2
    CW=CodageDaubechie(nextpow2(8000),tem); %Se calcula su transformada
    %Lineas correspondientes al truncado mediante fraccion
    [or idx]=sort(abs(CW)); %Se ordena por magnitud (Valor absoluto) de menor
a mayor
    CW(idx(1:8110))=0; %Los primeros 8110 (99%) se mandan a cero y se conserva
el 1% inalterado
    %-------------------------------------------------------
    nombrear=[destino '\' idioma '\' nombre int2str(i) '.wavelet'];%Se forma
la ruta completa del archivo
    %Se guarda el archivo los coeficientes se escriben con una presicion de
    %9 digitos despues del punto
    pr=fopen(nombrear,'w+');
    for j=1:8192
        fprintf(pr,'%.9f ',CW(j));
        fprintf(pr,'\n');
    end
    fclose(pr); %Se cierra el archivo
    inicio=(i*8000)+1; %Se actualiza inicio
    tem=[];
end



%-------------------------------------------------------------------------

function DJ=CodageDaubechie(J,CJ)
%Este codigo correspende al algoritmo de descomposicion wavelet db2, llamado
algoritmo piramidal de Mallat
%Codigo:Ionut Danaila,Pascal Joly, Sidi Mahmoud Kaber, Marie Postel, An
Introduction to Scientific Computing,Twelve Computational Projects Solved with
MATLAB, Capitulo 6 MRA (Analisis Multiresolucion)

DJ=CJ;
%Coeficientes Wavelet, para Db2 son 4
 C0=(1+sqrt(3))/(4*sqrt(2));
 C1=(3+sqrt(3))/(4*sqrt(2));
 C2=(3-sqrt(3))/(4*sqrt(2));
 C3=(1-sqrt(3))/(4*sqrt(2));

%analysis
if size(CJ,1)<size(CJ,2)%Si el vector es en fila
```

```
    for j=J-1:-1:0 % Nivel de resolucion

%DJ se toma en forma de anillo (Wrap-around) el principio es el fin y el fin
el principio, para C0 y C3 respectivamente; los cuatro coeficientes se
trasladan en la señal diadicamente (de dos en dos), Matlab nos ahorra hacer el
ciclo de 1 hasta 2^j, y manejarlo como una multiplicación de un vector por un
escalar y la suma final de los vectores que se almacenaran en  DJ(1:2^j) y
DJ(2^j+1:2^(j+1))

%Filtrado de Baja frecuencia, mediante wavelet padre (subespacio Vj)
        DJ(1:2^j)=C0*[CJ(2^(j+1)),CJ(2:2:2^(j+1)-2)]...
            +C1*CJ(1:2:2^(j+1))...
            +C2*CJ(2:2:2^(j+1))...
            +C3*[CJ(3:2:2^(j+1)),CJ(1)];


%filtrado de alta frecuencia, mediante wavelet madre (subespacio Wj)
        DJ(2^j+1:2^(j+1))=C3*[CJ(2^(j+1)),CJ(2:2:2^(j+1)-2)]... %
            -C2*CJ(1:2:2^(j+1))...
            +C1*CJ(2:2:2^(j+1))...
            -C0*[CJ(3:2:2^(j+1)),CJ(1)];
        CJ=DJ;

    end
else %Si es vector columna

    for j=J-1:-1:0% Nivel de resolucion

%Filtrado de Baja frecuencia, mediante wavelet padre (subespacio Vj)
        DJ(1:2^j)=C0*[CJ(2^(j+1));CJ(2:2:2^(j+1)-2)]...
            +C1*CJ(1:2:2^(j+1))...
            +C2*CJ(2:2:2^(j+1))...
            +C3*[CJ(3:2:2^(j+1));CJ(1)];

%filtrado de alta frecuencia, mediante wavelet madre (subespacio Wj)
        DJ(2^j+1:2^(j+1))=C3*[CJ(2^(j+1));CJ(2:2:2^(j+1)-2)]...
            -C2*CJ(1:2:2^(j+1))...
            +C1*CJ(2:2:2^(j+1))...
            -C0*[CJ(3:2:2^(j+1));CJ(1)];
        CJ=DJ;
    end
end
```

B.3. Código que convierte los archivos .wavelet en variables estadisticas

```
function principal
%Este programa se encarga de procesar los archivos de OGI_Story_res
%para lo cual los procesa por idioma y los convierte en archivos
%de conocimiento por idioma, que se guardan en la carpeta atributos
%Codigo:Jose Manuel Vargas Martínez

clear all;
Numero_elem=1;%es el numero de archivos que se procesaran
```

```
Numero_segun=5;%es el numero de segundos que se procesaran
f=fopen('idiomas.txt','r');%se abre el archivo de nombres de idioma
lang=textscan(f,'%s',9);%se cargan los nombres
la=lang{:};%extraccion del contenido de la cell obsteniendo un cellstring
for i=1:9
    %se llama a atributos que se ebcarga de calcular los estadisticos
    atributos(char(la(i)), Numero_elem, Numero_segun);
end

%-------------------------------------------------------------------------

function atributos(Name, Numero_elem, Numero_segun)
%Esta funcion se encarga de extraer las caracteristicas estadisticas
%de un señal que ha sido mapeada en wavelets
%Codigo:Jose Manuel Vargas Martinez, ITCM-DEPI 2008
clc
disp(Name);
N_coef=8192;%Numero de coeficientes
NameListFile=[Name '.txt'];%Archivo de la lengua que contiene la lista de las
señales
NameFileAtri=['Atributos\' Name '.txt'];%Archivo  donde se guardara los
atributos estadisticos
ruta=['OGI_Story_res\' Name '\'];%Ruta de los archivos wavelet
sounds=fopen(NameListFile,'r');%Archivo que contiene la lista es abierto
fil=fopen(NameFileAtri,'w+');%Archivo que almacenara atributos es arbierto
%Se convierte un cell a cellstring
sounds_lista = textscan(sounds, '%s',50);
sounds_lista=sounds_lista{:};
%---------------------------------------
m=[];
for i=1:Numero_elem %procesa cada archivo en la lista hasta Numero_elem
    nombrear = char(sounds_lista{i});
    for s=1:Numero_segun%procesa cada segundo del archivo actual hasta
Numero_segun
        rn=[ruta,nombrear,int2str(s),'.wavelet'];%Arma cada 1 de los
Numero_segun archivos .wavelet
        %-----------------------------------------------------
        archivo=fopen(rn,'r');%archivo wavelet
        fseek(archivo,70,'bof');%se ubica en 70 caracteres debido al formato
de praat
        %y carga la lista de coeficientes en un vector cell
        %con una precicion de 20 numeos despues del punto
        C = textscan(archivo, '%*s %*s %*s %.20f',N_coef);
        c=C{:};%se convierte de cell a lista de enteros
        %-------------------------------------------------------
        %el segmento de codigo anterior delimitado, corresponde a archivos
wavelet hechos en praat, para los hechos en matlab, se resume en una sola
        %linea:
        %c=load(rn);

        m=[m c];%se guarada c en una matriz
        fclose(archivo);
        %disp(i);
    end
    clear('C')
    clear('c');
```

```
    %La matriz contiene todos los Numero_segun cargados en columnas
    %pero como Matlab calcula estadisticos por columnas, hay que transponer
    %la matriz
    m=m';
    %y claculamos las variables estaditicas
    media=mean(m);
    %varianza=var(m);
    desviacion=std(m);
    maximo=max(m);
    minimo=min(m);
   %formamos el vector de atributos para la señal
    atri=[media desviacion maximo minimo];
    %y la guardamos en el archivo del idioma correspondiente con una
    %precision de 9 digitos despues del punto
    for j=1:N_coef*4
        fprintf(fil,'%.9f ',atri(j));
    end
    fprintf(fil,'\n');
    clear('media');clear('desviacion');clear('maximo');clear('minimo')
    clear('m');
    m=[];
    disp(s);
end
fclose(fil)
clear all;
```

B.4. Código que crea los archivos .arff binarios (dos lenguas) para ser evaluados en weka

```
function combinar
%Este programa se encarga de formar archivos weka binarios
%para posteriormente clasificarlos en weka
%Codigo: Jose Manuel Vargas Martinez, ITCM-DEPI 2008
global C file idio ii jj tam1
%Nombres de idioma (para ya no cargarlos de archivo))
idiomas=char('Ingles','Aleman','Espanol','Mandarin','Vietnamita','Japones','Ko
reano',...
    'Tamil','Farsi');
idio=cellstr(idiomas);
clear('idiomas');
%los nombres de los archivos arff tendran solo las iniciales
nom='IAEMVJKTF';
ext='.arff';%extencion de los archivos

%este ciclo construira 36 archivos arff
for i=1:8
    archivo1=[idio{i,:} '.txt'];%se forma el nombre del primer archivo de un
idioma
    A=load(archivo1);%se carga a la matriz A
    tam1=size(A,1);%se obtiene su tamaño
    for j=i+1:9
        archivo2=[idio{j,:} '.txt'];%se forma el segundo nombre del archivo
        B=load(archivo2);%Se carga
        C=[A;B];%Se forma una matriz C con A y B
        clear('B')%se linera memoria
        %Se forma el nombre del archivo A-B.arff
        volcado=['weka\' nom(i) '-' nom(j) ext];
```

```
        file=fopen(volcado,'w+');%es abierto para escritura
        ii=i;jj=j;%se respaldan indices para usarlos
        cabecera();%Se imprime la cabecera del archivo
        %cabecera_c();
        %se guarda la matriz C depues de la cabecera
        fprintmat(C, file, idio, ii, jj, tam1);
        clear('C')
        fclose(file);
    end
    clear('A')
end

%--------------------------------------------------------------------------

function cabecera
%Esta Funcion crea una cabecera para un archivo .arff
%Codigo:Jose Manuel Vargas Martinez, ITCM-DEPI
%file es el nombre el archivo
%idio contiene los combres de idioma
%ii y jj son los idiomas que estan siendo procesados
global file idio ii jj
atributos=char('Media','Desviacion','Maximo','Minimo');
a=cellstr(atributos);
clear('atributos');
primero='@RELATION lenguajes';
fprintf(file,'%s\n',primero);
for i=1:4
    for j=1:8192
        linea=['@ATTRIBUTE' ' ' a{i,:} int2str(j) ' ' 'REAL'];
        fprintf(file,'%s\n',linea);
    end
    fprintf(file,'\n');
end
linea=['@ATTRIBUTE class    {' idio{ii,:} ',' idio{jj,:} '}'];
fprintf(file,'%s\n',linea);
linea='@DATA';
fprintf(file,'%s\n\n',linea);
%fclose(file);

%--------------------------------------------------------------------------

function fprintmat(C, file, idio, ii, jj, tam1)
%Esta función se encarga de imprimir la matriz C con formato de weka
%Codigo Jose Manuel Vargas Martinez, ITCM-DEPI 2008

%C es la matriz que contiene a los idiomas A y B
%file es el nombre el archivo
%idio contiene los nombres de idioma
%ii y jj son los idiomas A y B que estan siendo procesados
%tam1 es el tamaño de A
t=size(C);%se obtiene el tamaño de C (filas y columnas)
for i=1:t(1)%el ciclo va de 1 hasta el tamaño de c en filas
    pos=ii;%se encarga de saber que idioma es procesado
    if i>tam1%en caso de que ya no queden instancias de A
        pos=jj;%se cambia a B
    end
```

```
    for j=1:t(2)%el ciclo va de 1 hasta el tamaño d C en columnas
        fprintf(file, '%.9f',C(i,j));%se graba cada coeficiente con precision
de 9
        if j==t(2)%si se llego al final
            fprintf(file, ', %s\n',idio{pos,:});%se graba el nombre del idioma
y se genera un salto
        else
            fprintf(file, ', ');%sino se pone una coma para el coeficiente que
sigue
        end
    end
end
```

B.5. Código que clasifica mediante validación cruzada usando Naive Bayes, usando funciones weka en NetBeans

```
package pruebaweka;
import java.io.*;
// FileReader
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.util.*;
// Random
import weka.core.*;
// Instances, Instance
import weka.filters.supervised.attribute.*;
// Discretize
import weka.classifiers.*;
import weka.attributeSelection.*;
// Classifier, Evaluation

public class Main {

    static int classifier = 1, numfolds = 10;
    static int atributos=0;
    static double perct=0;
    static double pct = 50.0;

public static void main(String[] args) throws Exception{
//este programa se encarga de clasificar un conjunto de instancias de lenguas
mediante el uso de la tecnica Naive Bayes
//Basado en Pruebaweka Codigo: Marco Antonio Aguirre Lam y Jose Manuel Vargas
Martines ITCM-DEPI 2006
//Modificacion para este proyecto: Jose Manuel Vargas Martinez, ITCM-DEPI 2008
String bufer;
//Lista de archivos arff que seran evaluados
FileInputStream fis=new FileInputStream("UCI\\weka\\nombres.txt");
//Tabla de resultados de las clasificaciones
FileOutputStream fos=new FileOutputStream("UCI\\weka\\tabla.txt");
//Lectura y escritura de dichos archivos respectivamente
BufferedReader d= new BufferedReader(new InputStreamReader(fis));
BufferedWriter w=new BufferedWriter (new OutputStreamWriter(fos));
int contador=0;//Contador de los archivos arff
do
{
 bufer=d.readLine();//Se lee un nombre
```

```
 contador++;
 String linea="UCI\\weka\\",conv="",linean="";;
 linean=linea.concat(bufer);
 if(bufer!=null)//si no ha llegado al final
  {
    clasificar(linean);//ha clasificar el archivo arff
    //Desplegado en pantalla
    System.out.println(bufer);
    System.out.print(' ');
    System.out.print(perct);
    System.out.print(' ');
    System.out.print(atributos);
    //Grabado en archivo
    w.write(bufer);
    w.write(' ');
    w.write(conv.valueOf(perct));
    w.write(' ');
    w.write(conv.valueOf(atributos));
    w.newLine();
   }

 }while(contador<36);
 d.close();
 w.close();
 //clasificar();

}

public static void clasificar(String nombre_archivo_entrenamiento) throws
Exception{
//esta funcion se encarga de cargar el archivo arff, filtrarlo y clasificarlo
Classifier C = seleccionarClassificador(classifier);//Se elige el clasificador
Instances entrenamiento = cargarInstancias(nombre_archivo_entrenamiento);//Se
cargan las instancias
Instances filtradas;//Donde se depositaran una vez filtradas las instancias
boolean terminado, cargado;//Para verificacion
InfoGainAttributeEval filtro =new InfoGainAttributeEval();//Filtro de
seleccion de atributos
Ranker busqueda = new Ranker();//Ranker, enlistador, buscador
//Se establecen los parametros del ranker
busqueda.setGenerateRanking(true);//por dfecto
busqueda.setNumToSelect(-1);//por defecto
busqueda.setStartSet("");//por defecto
busqueda.setThreshold(0.0);//se elige un umbral de cero aportacion
//Objeto de seleccion de atributos
weka.filters.supervised.attribute.AttributeSelection Seleccion= new
weka.filters.supervised.attribute.AttributeSelection();
//Se establecen sus parametros
Seleccion.setEvaluator(filtro);//Evaluador, tipo de filtro ganacia de
informacion
Seleccion.setSearch(busqueda);//Buscada, mediante ranker
cargado=Seleccion.setInputFormat(entrenamiento);//se copia el formato de las
instancias, devuelve true si fue exitoso
//se copian las instancias, ya establecido el formato
for(int i=0;i<entrenamiento.numInstances();i++)
{
  Seleccion.input(entrenamiento.instance(i));
```

```
}
//Se hace el filtrado
terminado=Seleccion.batchFinished();
filtradas=Seleccion.getOutputFormat();//se copia el formato a filtradas
//se copian las instancias a filtradas
while(Seleccion.numPendingOutput()>0)
{
 filtradas.add(Seleccion.output());
}
//Se colocan aleatoriamente
filtradas.randomize(new Random(1));
Evaluation Eval;//Objeto para evaluar las instancias filtradas
//Se llama a validacionCruzada mandando el clasificador, las instancias y el
numero del folds
Eval=ValidacionCruzada(C, filtradas, numfolds);//Devolviendo Eval con la
informacion
perct=Eval.pctCorrect();//Porcentaje de clasificacion correcta
atributos=filtradas.numAttributes();//obtencion del numero de atributos des
pues de filtrar
System.out.println();
}

public static Classifier seleccionarClassificador(int i) {
//Eleccion del clasificador
switch(i) {
 case 0: return new weka.classifiers.trees.j48.J48();
 case 1: return new weka.classifiers.bayes.NaiveBayes();
 case 2: return new weka.classifiers.bayes.BayesNet();
 default: return new weka.classifiers.bayes.BayesNetK2();
 }
}


public static Instances cargarInstancias(String nombre_archivo) throws
Exception{
  Instances I;
  //se carga el archivo de instancias
  I = new Instances(new FileReader(nombre_archivo));
  I.setClassIndex(I.numAttributes()  - 1);

  return I;
}

public static Evaluation ValidacionCruzada(Classifier C, Instances
entrenamiento, int numfolds)throws Exception {
//objeto de evaluation, inicializado con las instancias de entrenamiento
filtradas
Evaluation E = new Evaluation(entrenamiento);
//Ejecucion de la clasificacion mediante validacion cruzada usando Naive
Bayes
E.crossValidateModel(C, entrenamiento, numfolds);
return E;
}
```

Referencias

[Abercrombie 1967] Abercrombie D., *Elements of General Phonetics*, Edinburgh University Press, Edinburgh, 1967.

[Addison 2002] Addison Paul S., *The Illustrated Wavelet Transform Handbook : Introductory Theory and Applications in Science, Engineering, Medicine and Finance*, Institute on Physics Publishing, 2002.

[Berkling 1996] Berkling Kay Margarethe, *Automatic Languaje Identification with Sequences of Language-Independent Phoneme Clusters*, Tesis Doctoral, Oregon Graduate Institute of Science and Technology, 1996.

[Bharat 2006] Bharat Ravisekar, "A Comparative Analysis of Dimensionality Reduction Techniques", College of Computing Georgia Institute of Technology, 2006.

[Boersma y Weenink 2002] Boersma Paul and Weenink David, "Praat: Doing Phonetics by Computer" (Version 4.0.5) [Computer program].http://www.praat.org/, 2002.

[Burrus et al.] Burrus C. Sidney, Gopinath Ramesh A., Guo Haitao, *Introduction to Wavelets and Wavelet Transform*, Prentice Hall, 1997.

[Campbell et al. 2006] Campbell William, Gleason Terry, Navrátil Jiri, Reynolds Douglas, Shen Wade, Singer Elliot, and Torres-Carrasquillo Pedro, "Advanced Language Recognition using Cepstra and Phonotactics: MITLL System Performance on the NIST 2005 Language Recognition Evaluation", In *IEEE Odyssey 2006: The Speaker and Language Recognition Workshop*, (San Juan, Puerto Rico), June 2006.

[Caseiro y Trancoso 1998] Caseiro D., Trancoso I., "Language Identification Using Minimum Linguistic Information", *10th Portuguese on Pattern Recognition (RECPAD'98)*, Lisbon Portugal, 1998.

[Cimarusti e Ives 1982] Cimarusti D. and Ives R. B., "Development of An Automatic Identication System of Spoken Languages": Phase 1, *In proceedings 1982 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Paris, France, May 1982.

[Cumins et al. 1999] Cummins, Fred, Felix Gers , Jürgen Schmidhuber, "Language Identification from Prosody Without Explicit Features", *EUROSPEECH-99*, pp. 371-374, 1999.

[Daubechies 1992] Daubechies Ingrid, *Ten Lectures on Wavelets*, Vol. 61, SIAM Press, Philadelphia, PA. USA, 1992.

[Dauer 1983] Dauer R. M., "Stress-Timing and Syllable-Timing Reanalyzed", *Journal of Phonetics*, 11:51-62, 1983.

[Donoho 2000] Donoho D., "High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality", American Math. Society, Available at: www stat.stanford.edu/˜donoho/Lectures/AMS2000/, 2000.

[Fayyad et al. 1996] Fayad U. M., Piatetsky-Shapiro G., Smyth P., Uhturudsamy, R., *Advances in Knowledge Discovery and Data Mining*, San Mateo, AAAI Press, EE.UU, 1996.

[Fayyad e Irani 1993] Fayyad U.M., and Irani K. B., "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning". *Proc. of the Thirteenth International Joint Conference on Artificial Intelligence*. San Francisco: Morgan Kaufman, pp. 1022-1027, 1993.

[Fodor 2002] Fodor I. K., "A Survey of Dimension Reduction Techniques", LLNL Technical Report, UCRL-ID-148494, 2002.

[Foil 1986] Foil J. T., "Language Identification Using Noisy Speech", In *Proceedings 1986 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Tokyo, Japan, 1986.

[Goodman et al. 1989] Goodman F.J., Martin A.F., and Wohlford R.E., "Improved Automatic Language Identification in Noisy Speech". In *Proceedings 1989 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Glasgow, Scotland, May 1989.

[Gupta y Gilbert 2001] Gupta M., Gilbert A., "Robust Speech Recognition using Wavelet Coefficient Features", *ASRU '01, IEEE*, pp. 445- 448, 2001.

[Hall y Holmes 2003] Hall Mark A. And Holmes Geoffrey, "Benchmarking Attribute Selection Techniques for Discrete Class Data Mining", *IEEE Transactions on Knowledge and Data Engineering*. vol. 15, no. 6, pp. 1437-1447, 2003.

[Hioka y Hamada 2003] Hioka Yusuke, Hamada Nazomu, "Voice Activity Detection with Array Signal Processing in the Wavelet Domain", *IEICE TRANSACTIONS on Fundamentals of Electronics*, Vol E86-A, No 11, 2003.

[Hochreiter y Schmidhuber] Hochreiter S. and Schmidhuber J., "Long Short-Term Memory", *Neural Computation 9(8)*, pp. 1735-1780, 1997.

[House y Neuberg 1977] House A. S. and Neuberg E. P., "Toward Automatic Identication of the Language of An Utterance, I. Preliminary methodological considerations. *Journal of the Acoustical Society of America*, 62(3):708-713, 1977.

[Jaffard et al.] Jaffard Stéphane, Meyer Ives, Ryan D. Robert, *Wavelets Tools for Science and Technology*, SIAM, 1962.

[Jansen y Oonincx 2005] Jansen Maarten, Oonincx Patrick, *Second Generation Wavelets and Applications*, Springer-Verlag London 2005.

[Karris 2003] Karris Steven T., *Signals and Systems*, Orchad Publications, Second Edition, 2003.

[Lander et al. 1995] Lander, T., Cole A. Ronald, Oshika B. T., Noel M., "The OGI 22 language telephone speech corpus". *EUROSPEECH-1995*, pp. 817-820, 1995.

[Lee y Varaiya 2000] Lee Edward A., Varaiya Pravin, *Structure and Interpretation of Signals and Systems*, Addison Wesley, 2000.

[Liberman y Prince 1977] Liberman Mark, and Prince Alan, "On stress and linguistic rhythm". *Linguistic Inquiry* 8(2):249-336, 1977.

[Li et al. 2002] Li Tao, Li Qi, Zhu Shenghuo, Ogihara Mitsunori, "A Survey on Wavelet Applications in Data Mining", *ACM SIGKDD Explorations Newsletter*, Volume 4 , Issue 2, Pages: 49 – 68, December 2002.

[Li y Edwards] Li K.P. and Edwards T. J., "Statistical models for automatic language identication", In *Proceedings 1980 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Denver, CO, April 1980.

[Mallat 1999] Mallat S., *A Wavelet Tour of Signal Processing*, Academic Press. Second Edition. ISBN 0-12-466606-X, 1999.

[Mitchell 1997] Mitchell Tom M., *Machine Learning*, McGraw-Hill Science/Engineering/Math, March 1, 1997.

[Muthusamy 1992] Muthusamy Yeshwant K., "A Review of Research in Automatic Language Identification", Technical Report No. CS/E 92-009, Center for Spoken Language Understanding, Oregon Graduate Institute, 1992.

[Muthusamy et al. 1992] Muthusamy Yeshwant, Berkling Kay, Arai Takayuki, Cole Ronald, Barnard Etienne, "A Comparison of Approaches to Automatic Language Identification Using Telephone Speech", In *EUROSPEECH'93*, 1307-1310, 1993.

[Navrátil y Zühlke 1998] Navrátil, J. Zühlke W., "An efficient phonotactic-acoustic system for language identification", Acoustics, Speech and Signal Processing, 1998. *Proceedings of the 1998 IEEE*, pp. 781-784 Vol.2, 1998.

[Odgen 1997] Odgen R. Todd, *Essential Wavelets for statistical Applications and Data Analysis*, Birkäuser, 1997.

[Pike 1947] Pike K. L., *The Intonation of American English*, University of Michigan Press, Ann Arbor, Mich., 1947.

[Rabiner y Juang 1993] Rabiner Lawrence, Juang Biing-Hwang., *Fundamentals of Speech Recognition*, Prentice-Hall Signal Processing Series, 1993.

[Ramus et al. 1999] Ramus Franck, Nespor Marina, Mehler Jacques," Correlates of Linguistic Rhythm in the Speech Signal", *Cognition 73*, pp. 265-292, 1999.

[Ramus et al. 2000] Ramus F., Hauser M. D., Miller C., Morris D. And Mehler J., "Language Discrimination by Human Newborns and by Cotton-Top Tamarin Monkeys", *Science 288*, 349-351, 2000.

[Reyes 2007] Reyes Herrera Ana Lilia, *Un Método para la Identificación Automática de Lenguaje Hablado Basado en Características Suprasegméntales*, Tesis doctoral, Instituto Nacional de Astrofísica Óptica y Electrónica, 2007.

[Stark 2005] Stark Hans-Georg, *Wavelets and Signal Processing*, Springer Berlin Heidelberg New York, 2005.

[Stein 2000] Stein Jonathan Y., *Digital Signal Processing: A Computer Science Perspective*, John Wiley and Sons Inc., 2000.

[Sugiyama 1991] Sugiyama M., "Automatic Language Recognition Using Acoustic Features". *In Proceedings 1991 IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, May 1991.

[Torres-Carrasquillo 2002] Torres-Carrasquillo Pedro A., Singer Elliot, Kohler Mary A., Greene Richard J., Reynolds Douglas A., Deller Jr. J. R.**,** "Approaches to Language Identification Using Gaussian Mixture Models and Shifted Delta Cepstral Features", In *ICSLP-2002*, pp. 89-92, 2002.

[Rouas et al. 2003] Rouas J.-L., Farinas J., Pellegrino F. and André-Obrecht R., "Modeling Prosody for Language Identification on Read and Spontaneous Speech", *Proc. IEEE ICASSP2003*, vol 1, pp. 40-43, 2003.

[Walker 2008] Walker, James S., *A Primer on Wavelets and their Scientific Applications*, Chapman and Hall, Second Edition, pp. 5-31, 2008.

[Witten y Frank 2005] Witten Ian H. and Frank Eibe**,** *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.