Desarrollo de aplicación de software para la detección de voces de niños de 3 a 5 años

Diego Alejandro Duran Ruiz
Jonathan Arley Torres Castañeda

Trabajo de Grado para Optar al Título de Ingeniero Electrónico

Director
Franklin Alexander Sepúlveda Sepúlveda
PhD. En Procesamiento de Señales

Codirectora
Janeth Suarez Brand
Instituto de la Comunicación Humana

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Bucaramanga
2017

Dedicatoria

A Dios y a la Virgen, por fortalecerme en los momentos de dificultad que se presentan en el diario vivir.

A mi madre, a mis familiares que le debo todo y que tiene todo mi amor, a mi familia presente en Bucaramanga.

A todas las oraciones que se hicieron por mi persona en el proceso de grado muy agradecido estoy

A la memoria de Wilmer Alexis Rueda Arias, luchador de la vida

Diego Alejandro Duran Ruiz

A Dios, por guiarme, bendecirme, guiarme y darme fuerzas para superar los momentos difíciles.

A mi familia que ha sido mi constante apoyo.

A los profesores que tuve durante mi carrera universitaria, no los olvidaré.

Jonathan Arley Torres Castañeda

Agradecimientos

Los autores expresan sus agradecimientos a:

- Dios, por permitirnos desarrollar y terminar este proyecto.
- Magister (c) Jonathan Arley Monsalve Salazar, nuestra mano derecha y principal ayuda.
- PhD. Franklin Sepúlveda, por toda su valiosa ayuda en cada tramo del proyecto.
- Al Grupo CEMOS por todo el apoyo en este momento de nuestras vidas.
- Al Grupo HDSP por toda su valiosa ayuda y comprensión en todo el desarrollo del proyecto.
- Al Centro de la Comunicación Humana de la Universidad Nacional que nos facilitó nuestra primera base de datos.
- A todas aquellas personas que nos colaboraron con el uso de su voz para la Base de datos de Adultos, especialmente aquellas personas de la calle, vendedores ambulantes, policías, tenderos, secretarios, estudiantes, personal de oficios varios y personal de oficina.
- Nuestros padres y amigos que de una manera u otra nos brindaron su colaboración en el desarrollo del proyecto.

Contenido

Lista de Figuras

Pág.

Lista de Tablas

Pág.

Lista de Apéndices

Pág.

Resumen

Título: Desarrollo de aplicación de software para la detección de voces de niños de 3 a 5 años.[*]

Autores: Diego Alejandro Duran Ruiz
Jonathan Arley Torres Castañeda[**]

Palabras claves: Segmentación de hablantes, Fonoaudiología, mezclas gaussianas.

Descripción:

Este proyecto presenta y describe la implementación de una aplicación de software para la detección de voces de niños de 3 a 5 años. En el desarrollo se utilizaron diferentes bases de datos de audios, principalmente las señales de voz suministradas por la fonoaudióloga Janeth Suarez Brand. La base de datos está compuesta por 19 audios, cada uno con una duración promedio de 42 minutos. Cada audio corresponde a la conversación entre la fonoaudióloga y niños entre 3 a 5 años de edad. También, se recolectaron audios de diferentes personas y edades en nuestro entorno con el fin de realizar pruebas adicionales. La aplicación se desarrolló mediante el uso de herramientas de procesamiento de señales de voz y utilizando como herramienta de modelado a las Mezclas Gaussianas.

Por medio de una interfaz gráfica amigable con el usuario se busca integrar los métodos utilizados. Basta con cargar el audio que se quiere segmentar, seleccionar la opción de segmentación y como resultado se generarán los intervalos de tiempo correspondientes a la voz del niño, al mismo tiempo que las distingue del ruido de fondo y de las voces de los adultos. El software ha sido desarrollado en Python 2.7.12.

La segmentación de audios con voces infantiles podría llegar a usarse en Fonoaudiología para propósitos de análisis del habla durante la Primera Infancia.

---

[*] Trabajo de grado

[**] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: PhD Franklin Alexander Sepulveda Sepulveda

Abstract

Title: Development of software application for the detection of voices of children from three to five years[*]

Authors: Diego Alejandro Duran Ruiz
Jonathan Arley Torres Castañeda[**]

Keywords: Speaker diarization, Speech-Language Pathology, gaussian mixtures.

Description:

This project presents and describes the implementation of a software application for the detection of voices of children from 3 to 5 years. In the development, different databases of audios were used, mainly the voice signals supplied by the speech therapist Janeth Suarez Brand. The database is composed of 19 audios, each with an average duration of 42 minutes. They are composed by conversations between the speech therapist and children between 3 and 5 years of age. Additionally, audios from different people and ages were collected in order to perform additional test procedures. The application was developed through the use of voice signal processing tools and using Gaussian Mixtures as a modeling tool.

Through a user-friendly graphical interface, the methods used are integrated. First, you need to load the audio you want to segment, select the segmentation option and as a result will generate the time intervals corresponding to the child's voice, while distinguishing them from the background noise and voices of adults. The software has been developed in Python 2.7.12.

The segmentation of audios with children's voices could be used in Speech-Language Pathology for purposes of speech analysis during Early Childhood.

[*] Bachelor thesis

[**] Faculty of Physical-Mechanical Engineering, School of Electrical Engineering, Electronics and Telecommunications. Director. PhD. Franklin Alexander Sepulveda Sepulveda

Introducción

En el proceso de detección de voces se busca determinar la respuesta a la pregunta ¿Quién habla y cuándo?, en una grabación de audio o video. Como definición más formal se habla de obtener la identidad de un hablante y los intervalos durante los cuales está hablando (Anguera, Bozonnet, 2012). La detección de hablantes ha recibido mucha atención recientemente especialmente por el *National Institute of Standards and Technology* (NIST) y generalmente tienen varios dominios de estudio entre los que se encuentran, las conversaciones telefónicas, los programas de TV y radio y las reuniones entre hablantes de número desconocido. Las aplicaciones de la detección de hablantes incluyen sistemas de reconocimiento de datos, sistemas de transcripción con gran cantidad de datos, sistemas de etiquetado de audio, sistemas de archivo y monitoreo de audio, conteo de hablantes, enrutamiento de llamadas y detección de diálogos (Moattar, Homayounpour, 2012).

Para realizar una segmentación de hablantes se deben tener en cuenta una gran variedad de parámetros como son la calidad del audio, el aprovechamiento del ambiente, la presencia de ruidos molestos, el género y el tono del habla y la simultaneidad de los hablantes. El estado del arte actual se adentra en temas específicos como dialectos (Das, Allayear, Amin, Rahman, 2016), y sobre reconocimiento de hablantes, cuyo problema principal es la simultaneidad en el momento en el que hablan (Moattar, Homayounpour, 2012). Particularmente el habla de los niños difiere de la de los adultos en muchos aspectos y características importantes. Entre las que más se destacan el tono,

la frecuencia de los formantes, la duración de la conversación. En lingüística estos cambios también se relacionan con la pronunciación y la gramática (Ghai, Sinha, 2010).

De la misma forma no se encontró un sistema de segmentación de hablantes, para dos personas y que esté enfocado en Niños de Primera Infancia, dado que las investigaciones solo incluyen reconocimiento de Niños en edades escolares de 6 años en adelante (Safavi, Najafian, Hanani, 2008).

Las características cambiantes del habla pueden ser sorteadas con herramientas que se basen en detección de cambios (Moattar, Homayounpour, 2012), y para nuestro caso, detección de cambios para dos hablantes (Adami, Kajarekar, Hermansky, 2002). Básicamente acerca de la voz humana se necesita obtener la información de cuatro formantes F1 al F4, la cual es más importante si logra obtener los formantes F1(Frecuencias por debajo de 1.2KHz) que contiene las características principales de la voz, y el formante F3(Frecuencias entre 2.0KHz y 5.8KHz) que contiene los valores agudos y fricativos. En el caso de los niños y los adultos el género resulta un factor influyente pues los estudios demuestran que estos formantes aumentan su ancho de banda entre 11% y un 38% tanto para Mujeres como para Niños que coincidirían debido a la agudeza y tono de la voz con diferencia de los Varones que tendrían menos ancho de banda (Safavi, Najafian, Hanani, 2008).

Viendo las problemáticas actuales en donde en el sector médico cada vez es más importante el uso de nuevas tecnologías. En particular, herramientas que puedan facilitar este tipo de procesos que son tediosos manualmente. Es necesario contar con un sistema que identifique los momentos

en los que participa el niño dentro de una conversación. Para nuestra población infantil es muy determinante saber la calidad del habla desde edades tempranas dado que este será un patrón que regirá a lo largo de su vida (Schuster, Pancoast, 2014).

Para el caso particular de Colombia, actualmente se cuenta con aproximadamente 8,5 millones de niños entre los 0 y los 9 años en todo el país (DANE, 2016). Hasta donde se investigó en los antecedentes en esta materia, no se pudo encontrar alguna aplicación de software dedicada a la tarea de detección de voces de niños de la primera infancia para el caso del idioma castellano de Colombia.

Tener un sistema de detección de voces de niños a disposición es particularmente importante en aquellos oficios y tareas que implican la realización de entrevistas con niños, como lo es el caso de las tareas de terapia del habla, realizadas por fonoaudiólogos.

1. Planteamiento del problema

Actualmente en Colombia aquellos niños con problemas del habla de algunas regiones del país requieren consulta de un Fonoaudiólogo con el objetivo de realizar un análisis del habla desde tempranas edades.

Durante la terapia del habla y las entrevistas con niños, se generan registros de audio de mediana duración (aprox. 33 min.). Por tanto, se requiere de herramientas que faciliten el análisis de los audios. Entre ellas se requieren sistemas de segmentación de hablantes que permita determinar aquellos instantes donde interviene el niño durante la entrevista.

Con el objetivo de realizar procesos de segmentación de hablantes se han propuesto varios métodos. Abordando el problema de forma general se presenta un algoritmo que utiliza modelos de mezclas Gaussianas para representar la voz de una cantidad de hablantes desconocida (Moattar, Homayounpour, 2012). Asimismo, conocer de antemano la cantidad de hablantes mejora el desempeño de los sistemas de segmentación (Meignier, Moraru, Fredouille, 2006). Sin embargo, en nuestro problema se sabe de antemano que en la conversación participan una persona adulta y un niño. Esta información ayudaría a mejorar el desempeño y reducir la complejidad del sistema. El enfoque general que consta de un algoritmo de segmentación y clustering muy usado (Anguera, Bozonnet, 2012), En algunos enfoques se pueden ver inclusive detalladamente algoritmos más específicos; que utilizan técnicas que permitirían diferenciar el niño a partir de frecuencias fundamentales (Martins, Troncoso, Abad, 2009). Asimismo ciertos autores se centran en la

busqueda de las regiones del habla en este caso (Reynolds, Quaitieri, 2000), e inclusive un uso específico con algoritmos de distancia (Meignier, Moraru, Fredouille, 2006).

De otra parte, se han planteado sistemas de segmentación de hablantes para cuando la cantidad de hablantes corresponde a 2 personas. En (Adami, Karejakar, Hermansky, 2002) se propone realizar los siguientes pasos: extracción de parámetros (MFCC), *voice activity detection* (VAD), Caracterización de Modelo GMM-UBM utilizando agrupamiento y Resegmentación. Por otra parte, en (Bazyar, 2014) se utiliza el criterio de razón de verosimilitud (GLR, *Generalized Log-Likelihood Ratio*) para obtener los puntos de cambios de hablante para luego asignarlos mediante un proceso de agrupación para dos clases.

Algunos procesos utilizan máquinas de soporte vectorial (SVM, *Support Vector Machines*) de tal manera que se separa la voz de una entrevistadora de la de niños de 6-18 meses (Schuster, S., Pancoast, 2014). A pesar de los métodos planteados anteriormente, no se cuenta con métodos de segmentación dirigida a la población de la primera infancia del castellano colombiano, y particularmente en el rango de 3 a 5 años de edad.

Particularmente contamos con varias herramientas como son las Mezclas Gaussianas GMM, la extracción de parámetros utilizando Coeficientes de Mel MFCC y los sistemas de detección de voz VAD, que sumadas en un mismo algoritmo pueden extraer de forma categórica los cambios de voces entre Niños y Adultos.

Este proyecto pretende abordar desde una perspectiva fácil, amena y rápida a través del software Python 2.7.12, la unión de librerías robustas que permitan realizar la extracción de parámetros teniendo en cuenta estas características de la voz de Niños y Adultos, teniendo como contexto una base de datos que contiene ruido ambiente y que no posee simultaneidad en el hablante.

Dado lo anterior, consideramos necesario desarrollar un sistema de detección de hablantes pertenecientes al rango de edades de 3 a 5 años. Adicionalmente, y dado que es una herramienta dirigida a fonoaudiólogos y no a ingenieros, se considera necesario crear una interfaz gráfica que muestre los segmentos de tiempo en los que participa el niño.

2. Objetivos

2.1. Objetivo General

Desarrollar una aplicación de software que permita de manera automática determinar aquellos segmentos de voz en los que hablan niños de 3 a 5 años mediante el uso de herramientas relacionadas con el modelado estadístico de señales de voz.

2.2 Objetivos Específicos

1. Desarrollar algoritmo de detección de hablantes en grabaciones en la que participan dos hablantes (niño y adulto) del Castellano Colombiano.
2. Desarrollar una interfaz gráfica de usuario que muestre y archive los resultados de la segmentación y que permita la navegación dentro de la señal de audio por parte del usuario en el Software Praat.
3. Evaluar el desempeño de la aplicación de software desarrollada mediante el uso de métricas propias de la rama de la segmentación de hablantes.

2.3 Alcances

Se pretende realizar un sistema que permita determinar los segmentos de voces de niños mediante el uso de herramientas de procesamiento de señales de voz y de herramientas de modelado tales como modelos de mezclas Gaussianas. Estos modelos podrán determinar quién habló y en qué momento lo hizo. Adicionalmente, a través de una interfaz gráfica se busca integrar

los métodos utilizados y mostrar las zonas dentro del audio que corresponde a la voz del niño, al tiempo que diferencia estas del ruido de fondo y de las voces de los adultos.

3. Metodología

La segmentación engloba los cambios en el habla, intentando establecer los límites en los que pasa de un hablante a otro en una grabación de audio (Ugarte Echeverría, 2010). Los métodos de segmentación de hablantes más conocidos son:

Método de detección de silencios: Se utiliza un umbral que determina la probabilidad de existencia de un silencio. Se determinan los instantes de silencio entre 2 hablantes iguales o diferentes. se usa también información extra para decodificar cómo realizar una separación por género. Como ya se ha dicho antes, puede no haber una relación clara entre la existencia de un silencio y el cambio de un hablante en una grabación

Métodos basados en modelos: Los puntos son clasificados por un algoritmo de máxima verosimilitud de tal manera que la frontera entre uno y otro modelo serán los puntos de cambio en la segmentación. De esta forma, es necesario un conocimiento previo para inicializar los modelos de hablantes. En los casos más simples se usa un Modelo Universal conocido como UBM que es el modelo genérico del hablante o de los hablantes este puede basarse en varias técnicas como cadenas de Markov (HMM, *Hidden Markov Models*), SVM y GMM.

Métodos basados en distancias: Son las técnicas de segmentación más comunes. En estos métodos, la distancia métrica entre dos segmentos de análisis consecutivos es usada como decisión de medida para determinar el punto de cambio. Estos métodos requieren una detección límite que se encuentra sintonizada de manera empírica para detectar cambios de audio.

Métodos usando algoritmos híbridos: Estos combinan distancia y técnicas de modelo. Generalmente se usa el algoritmo de distancia para hacer una primera segmentación de la señal de audio. Con esos segmentos se crea un modelo de hablante. Ahora se puede realizar una re-segmentación basada en modelos haciendo más robusto el sistema (Moattar, Homayounpour, 2012).

Separamos los silencios a través de un método de detección de silencios VAD. Utilizamos un método basado en modelos GMM obteniendo parámetros del habla.

Algunos métodos de segmentación se muestran en las figuras 1 y 2. El hecho de clasificar y detectar cambios sumado a un algoritmo de clustering, puede servir como una medida no supervisada tal como se ve en la figura 1. Generalmente la idea es poder obtener un modelo que pueda adaptarse en a medida que pueda ir recibiendo más datos (Kwon, Narayanan, 2003b).

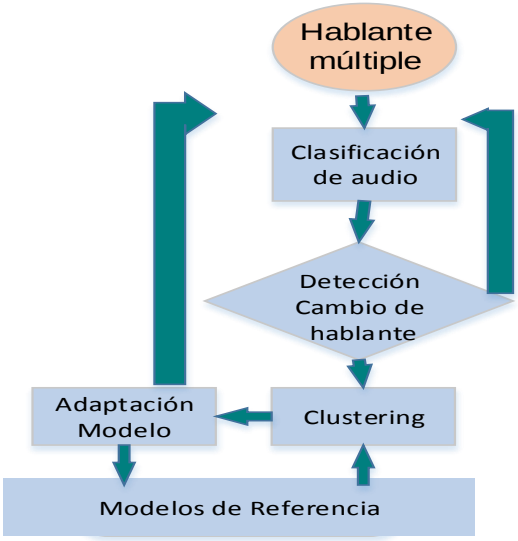

Figura 1. Método de segmentación de hablantes basado en detección de cambios. Adaptado de Kwon & Narayanan (2003).

En la figura 2, aunque puede existir una re-segmentación en el tramo final dependemos de un método BIC para probabilísticamente determinar los cambios de hablantes y posteriormente con un clustering buscar la región más probable de los datos. (Lu, Zhang, 2005).

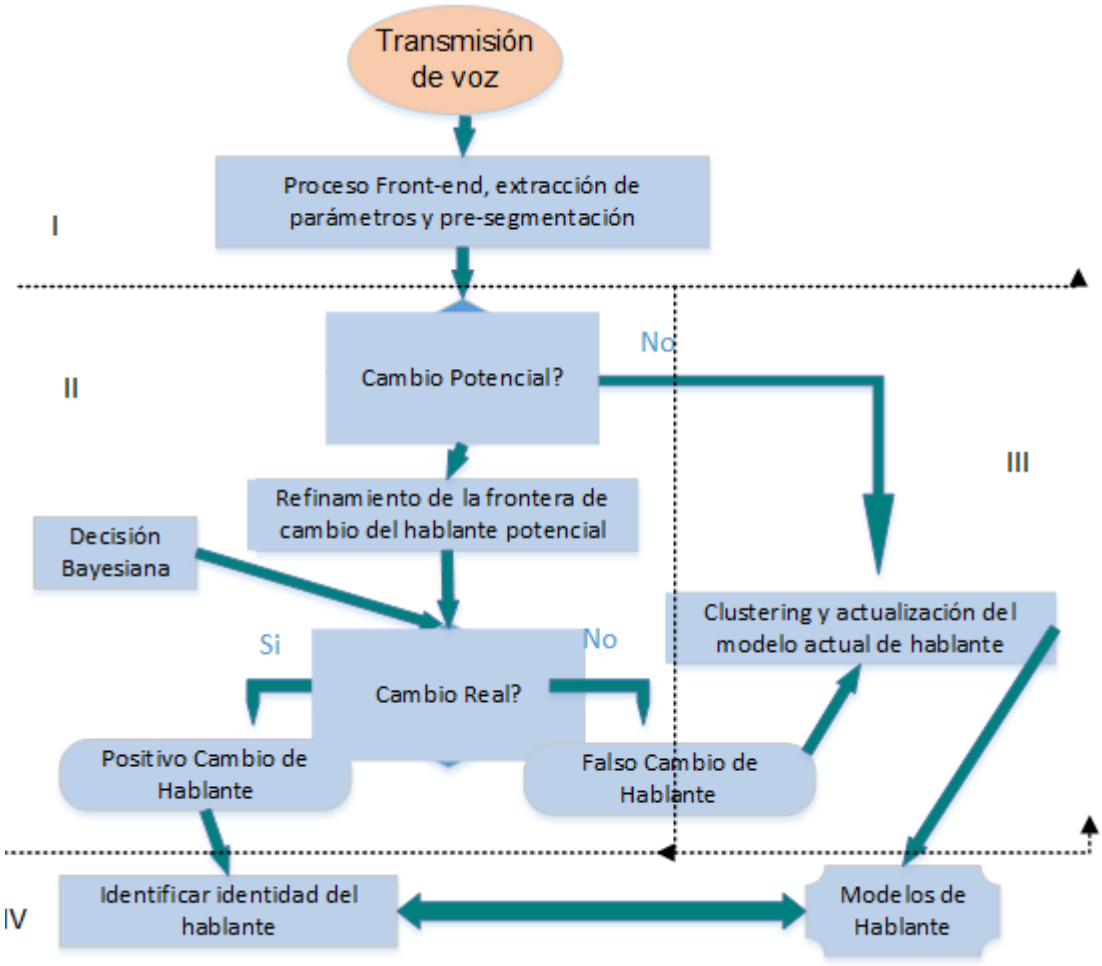

Figura 2. Método de segmentación alternativo I) Etapa de acondicionamiento. II) Etapa de segmentación. III) Etapa de clustering. IV)Etapa de re-segmentación. Adaptado de Lu & Zhang (2005).

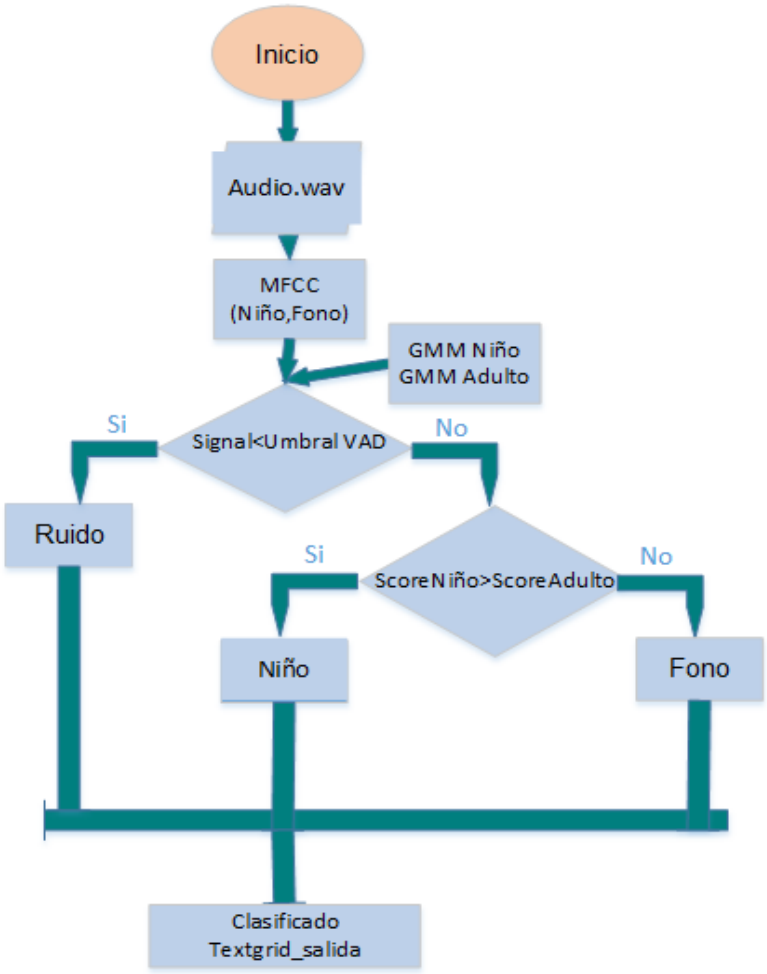

Figura 3. Diagrama de bloques – Segmentación.

En la figura 3, se resume el método de segmentación que hemos propuesto. Para probar nuestro programa una vez obtenido el modelo GMM-UBM para cada clase ya sea niño o adulto, se realizan las tres etapas de segmentación : Extracción de parámetros MFCC, Detección de Voz VAD y Entrada de Modelo Entrenado GMM-UBM, al obtener esta caracterización, el VAD separará el

ruido y dejará las otras dos clases obteniendo la probabilidad del Niño y del Adulto dependiendo de cuál es mayor, pertenecerá a alguna de las dos clases que ya fue parametrizada con la unión del modelo GMM-UBM, y los coeficientes del audio a segmentar. Esto producirá la fusión de las tres etiquetas en un único Textgrid de salida clasificado.

3.1 Extracción de parámetros

Los MFCC son los parámetros que tienen en cuenta la percepción de la voz humana con respecto a las frecuencias, por lo cual, son ideales para el reconocimiento de voz. Los coeficientes derivados del proceso de extracción constituyen una buena representación de las características dominantes en acústica, así como información para las ventanas de tiempo seleccionadas (Ali, Mansor, Khuan, & Zabidi, 2012).

Para hallar los MFCC, el primer paso consiste en tomar la onda de entrada y someterla a un banco de filtros de ancho de banda delimitados que generan valores de energía asociados a cada rango de frecuencia del filtro, estos producen una aproximación al contenido espectral de la señal. Estos filtros, de tipo pasa-banda, están traslapados y distribuidos a lo largo del rango de frecuencias de la señal de voz. Las frecuencias centrales de estos filtros son determinadas de tal manera que todos los filtros compartan el mismo ancho de banda de acuerdo a la Escala Mel. La técnica de coeficientes cepstrales en la Escala de Mel es un método ampliamente utilizado para obtener los vectores característicos de la señal de voz (Porras, Mendoza, 2016).

En la figura 4 se presenta un conjunto de pasos comúnmente usadas para esta primera etapa.

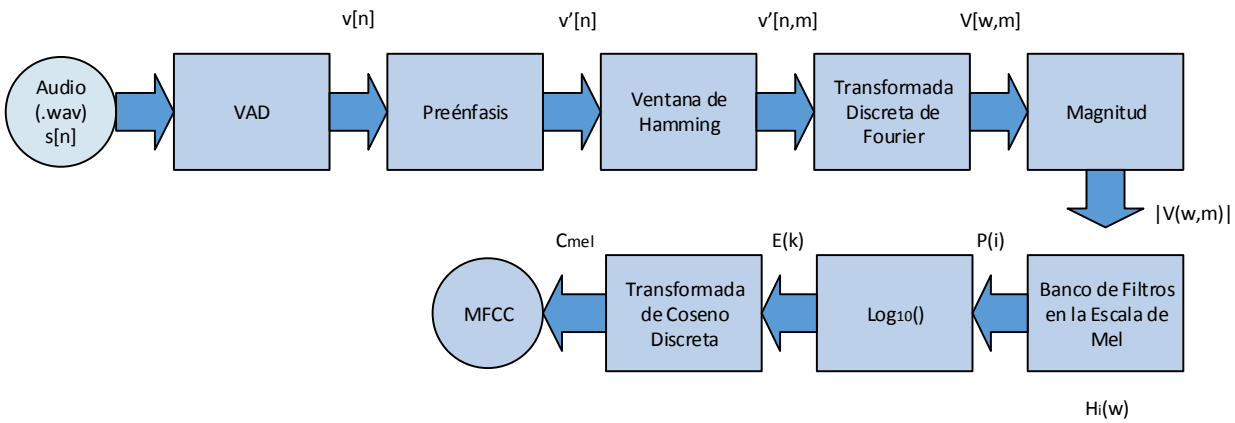

Figura 4. Extracción de coeficientes de Mel. Adaptado de Mello (2010).

Preénfasis: A la señal de voz identificada por el VAD se le aplica el filtro de preénfasis. Tiene como objetivo compensar la atenuación de aproximadamente 20 dB/década que se produce en la producción del habla.

La señal pre-enfatizada se obtiene a través del siguiente filtro:

$$\boldsymbol{y[n] = x[n] - ax[n-1]; a \in [0.95, 0.98]} \quad (1)$$

El filtro aumenta la amplitud de la señal para altas frecuencias y la atenúa para bajas frecuencias.

Ventana de Hamming: Cuando se aplica la aplica la Transformada de Fourier sobre el segmento finito se requiere que la señal sea periódica y continua dentro del intervalo. Para evitar, efectos indeseados en el espectro como discontinuidad en los extremos se aplica un enventanado. En particular, la ventana de Hamming se rige por el siguiente modelo matemático.

$$\boldsymbol{w(n,\alpha) = (1-\alpha) - \alpha cos\left(\frac{2\pi n}{N-1}\right); 0 \leq n \leq N-1} \quad (2)$$

Las curvas de las ventanas de Hamming para diferentes valores de $\boldsymbol{\alpha}$ se muestra en la Figura 5.

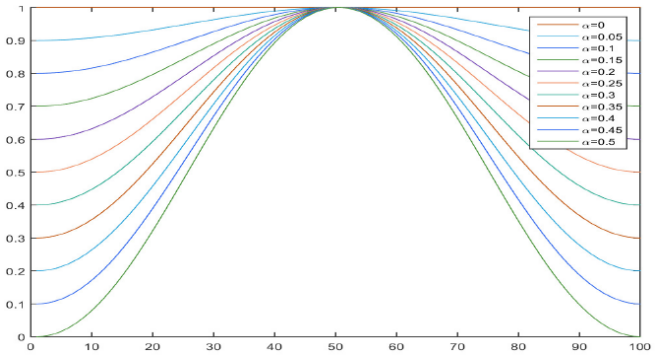

Figura 5. Ventanas de Hamming. Adaptado de Das, *et al.* (2016).

Transformada Discreta de Fourier (DFT): Luego de que la señal discreta ha sido enventanada podemos aplicar esta Transformada para adquirir la magnitud de la respuesta frecuencial.

$$\boldsymbol{X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}}; 0 \leq k \leq K-1} \quad (3)$$

A partir de este momento, se desprecia la fase, se utiliza solamente la magnitud.

Banco de filtros en la Escala de Mel: Se realiza la conversión a la Escala de Mel. La conversión de Hertz a Mel está dada por la siguiente fórmula:

$$\boldsymbol{M(f) = 2595log_{10}\left(1 + \frac{f}{700}\right) = 1125ln\left(1 + \frac{f}{700}\right)}$$

(4)

Energía del banco de filtros logarítmicos: Obtenemos el logaritmo de las energías del banco de filtros. El banco de filtros se muestra en la figura 6.

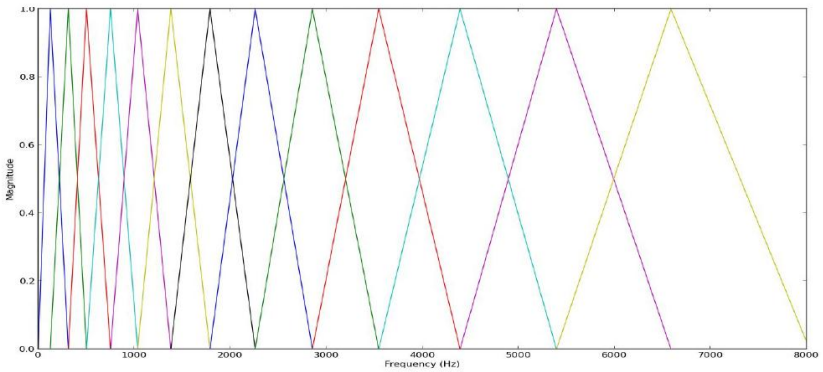

Figura 6. Banco de filtros de Mel. Adaptado de Das *et al.* (2016).

Transformada Coseno Discreta: Al aplicar esta transformada obtenemos la información necesaria acerca de las características de la voz en pocos coeficientes.

3.1.1 Librería Bob. Bob es una herramienta desarrollada por Biometrics Group en Suiza. Se encuentra desarrollada en C++ y Python. Sus funcionalidades se centran en el desarrollo de librerías para el desarrollo de aplicaciones basadas en el procesamiento de imágenes y el reconocimiento biométrico.

Es muy útil para poder realizar extracción de parámetros utilizados en segmentación de audio tales como MFCC, LPC(*Linear-Prediction-Coefficients*) y LFCC(*Lineal-Frequency-Cepstral-Coefficients*) que determinan características en el reconocimiento del habla y la asociación de modelos universales de hablantes (Anjos, 2014).

3.2 Detección Activa de la Voz (VAD)

El VAD juega un papel central en los algoritmos de sustracción espectral debido a que su precisión afecta drásticamente el nivel de supresión de ruido y la cantidad de distorsión del habla que se produce (Adami *et al.*, 2002). Para el presente trabajo de grado el VAD es una tarea importante y tiene por finalidad principal determinar aquellos segmentos donde las personas hablan, eliminando los silencios. Por otra parte, la misma detección de puntos de inicio/fin puede ayudar a determinar los límites de cambio entre hablantes y a determinar cuando las condiciones del ambiente cambian. Sin embargo, puede no haber una relación clara entre la existencia de un silencio y el cambio de un hablante en una grabación (Moattar, Homayounpour, 2002). Lo que se busca en este caso es que se permita reducir el número de muestras de tal manera que se pueda obtener solo las muestras hablantes descartando ruidos y silencios.

Existen diferentes algoritmos de detección de voz. A continuación, presentaremos el más importante (Meduri & Ananth, 2012).

VAD basado en la tasa de cruce por cero y energía: Este método es el más sencillo y rápido. Consiste en separar los segmentos donde hay voz y donde no hay voz. Se dice que hay un cruce por cero cuando dos muestras sucesivas poseen signos diferentes. Se puede definir:

$$Z_n = \sum_{m=-\infty}^{\infty} |sgn[x(m)] - sgn[x(m-1)]| w(n-m) \quad (5)$$

$$sgn[x(m)] = \begin{cases} 1 x(n) \geq 0 \\ -1 x(n) < 0 \end{cases} \quad (6)$$

$$w(n) = \begin{cases} \frac{1}{2N} 0 \leq n \leq N-1 \\ 0 \quad Casocontrario \end{cases} \; N \; es \; la \; duración \; de \; la \; ventana \quad (7)$$

La tasa de cruce por cero indica la presencia o ausencia de voz. Si la tasa es alta, se considera un segmento sin voz. En el caso contrario, si la tasa es baja se considera el segmento como voz.

3.2.1 Librería WebRTCVad. Es una herramienta desarrollada por la empresa WEBRTC con el apoyo de Google y Opera entre otros (Dutton, 2012), para el reconocimiento de segmentos de audio de habla. Su objetivo es evaluar un umbral de energía de manera que al superarlo pueda tenerse como resultado un ‘1’ para ‘voz’ y un ‘0’ para no-voz. Se utilizan pequeños fragmentos de 30 [ms] que son posibles de obtener de tal forma que actúan como un búfer que compara las energías hasta que estas lleguen al tope de 300[ms].

3.3 Modelado mediante mezclas gaussianas

Para poder realizar una interfaz de segmentación se debe tener en cuenta que el término

detección de hablantes según la literatura en general se separa en dos procesos: segmentación y clustering, los cuales harían más viable y robusto cualquier modelo basado en entrenamiento. De esta forma al realizarlo se podrá realizar la verificación del hablante. Utilizamos mezclas gaussianas, a continuación, daremos una breve explicación:

$$\boldsymbol{Y} = [\boldsymbol{Y_1}, \boldsymbol{Y_2}, \dots, \boldsymbol{Y_D}]^{\boldsymbol{T}} \quad (8)$$

Y es una variable aleatoria real *D*-dimensional. Se dice que la distribución de *Y* sigue una distribución mezcla finita si su función de densidad de probabilidad se puede escribir como una combinación lineal de funciones de densidad de probabilidad (pdf, *probability density functions*) elementales (García Herreño, 2010)

$$p(y \mid \theta) = \sum_{i=1}^{I} \alpha_i p(y \mid C = i, \beta_i)\, i \in \{1, \dots, I\} \quad (9)$$

Donde *I* representa el número de distribuciones elementales (componentes) de la mezcla. C es el conjunto $(1,2, \dots, I)$, θ es el conjunto $\{\alpha_1, \dots, \alpha_I, \beta_1, \dots, \beta_I\}$. Representa el número de parámetros

β es el conjunto $\{\beta_1, \dots, \beta_I\}$ de parámetros asociados a cada distribución de la mezcla y α es el conjunto $\{\alpha_1, \dots, \alpha_I\}$, representa la probabilidad o peso de cada distribución de la mezcla.

La mezcla Gaussiana está compuesta por distribuciones que llevan el mismo nombre como se observa en la Figura 7. Por tal motivo un modelo de mezclas Gaussianas es una distribución

probabilística cuya función de probabilidad es una combinación lineal de distribuciones Gaussianas, de la siguiente forma:

$$p(y \mid \theta) \sum_{i=1}^{I} \alpha_i N(y \mid C = i, \beta_i) \quad (10)$$

Siendo

$$N(y \mid C = i, \beta_i) = \frac{1}{(2\pi)^{\frac{D}{2}}} exp\left(\frac{-1}{2}(y - \mu_i)^T \Sigma (y - \mu_i)\right) \quad (11)$$

Y los parámetros μ_i, Σ_i de cada Gaussiana corresponde a la media y la varianza.

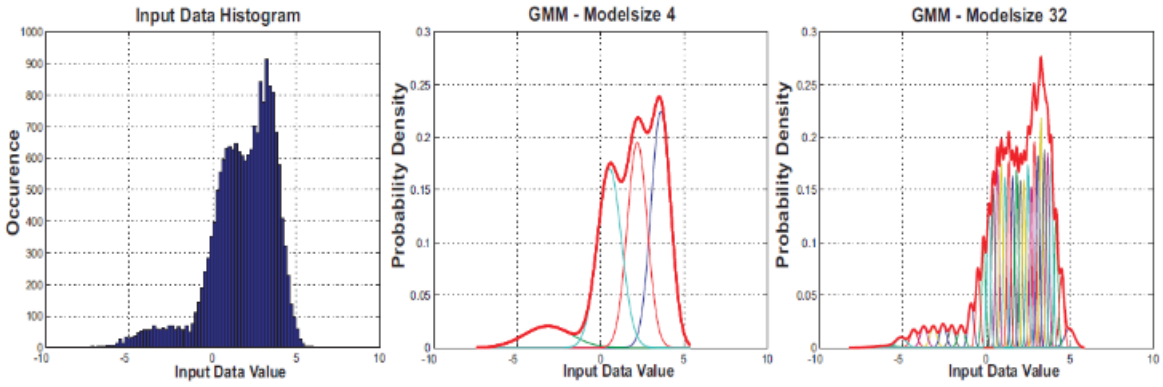

Figura 7. Histograma y densidad de probabilidad de mezclas gaussianas Adaptado de Ugarte Echeverría (2010).

Modelado de mezclas gaussianas usando scikit-learn: Herramienta de Python que usa técnicas supervisadas y no supervisadas para crear estructuras de datos correlacionados. Se ha popularizado para el uso de base de datos académicos y comerciales. El objetivo es que con sencillas herramientas y aprovechando la calidad del código, la colaboración, la documentación y el alto rendimiento de esta librería se puedan caracterizar modelos con el nivel de robustez que requieren

los sistemas de producción. El proceso realizado se muestra en la figura 8.

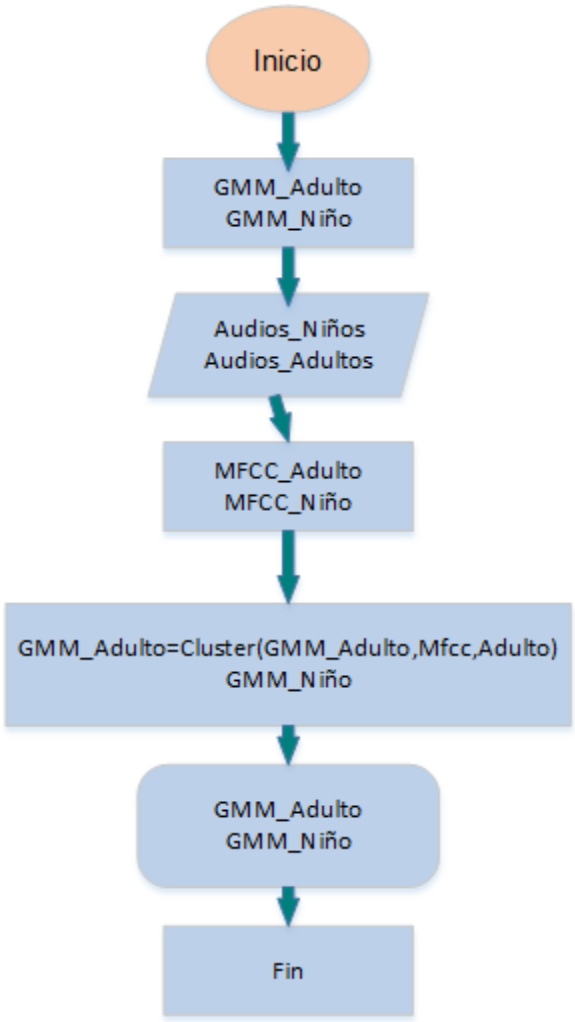

Figura 8. Diagrama de bloques – Entrenamiento

El entrenamiento toma grandes cantidades de datos y tiende a obtener resultados de salida parecidos a lo que se pide que se reconozca (Wu, Zhou, Li, 2014). Para el caso de realizar el reconocimiento de la voz es necesario analizar que debe ser el mismo ambiente (con Ruido o sin

Ruido), y debe haber equilibrio en el dominio del tiempo y con los géneros masculino y femenino. También que se debe tomar un audio para entrenar y otro distinto para segmentar.

3.4 Medidas de evaluación del desempeño

3.4.1 Valores de verdad. Si bien en la literatura nos es posible obtener un valor de acierto o error (Meduri & Ananath, 2012), en este caso lo que nos importa es poder obtener una validación que nos permita evaluar punto a punto en el dominio del tiempo, por tal motivo nuestras métricas se basan en valores positivos y valores negativos de referencia que permiten evaluar cada punto de frontera, punto a punto. En nuestro caso el Textgrid exportado en el programa Praat se compara con un Textgrid de mismo audio, realizado manualmente.

En la tabla 1 se visualiza como la comparación del sistema con la referencia produce cuatro valores: Verdadero Positivo, Verdadero Negativo, Falso Positivo, Falso Negativo (Liu, Shriberg, 2007).

Tabla 1.

Parámetros utilizados para validación

Referencia vs Sistema	Sistema Verdadero	Sistema Falso
Referencia verdadera	Verdadero Positivo	Falso Negativo
Referencia falsa	Falso Positivo	Verdadero Negativo

Nota: Adaptado de “Comparing Evaluation Metrics for Sentence Boundary Detection “ por Liu & Shriberg, 2007.

3.4.2 Métricas. De acuerdo con (Liu, Shriberg, 2007) y obteniendo nuestro porcentaje de verdaderos y falsos podremos obtener las métricas propias de los análisis de frontera. Cada una de las siguientes medidas se debe tomar como un índice estadístico no solamente como medición física de precisión, exactitud y cercanía. También tenemos en cuenta que precision y recall son medida de la relevancia (Powers, 2014)

$$\boldsymbol{Precision(P)} = \frac{\boldsymbol{V_P}}{\boldsymbol{V_P + F_P}} \quad (12)$$

$$\boldsymbol{Recall(R)} = \frac{\boldsymbol{V_P}}{\boldsymbol{V_P + F_N}} \quad (13)$$

$$\boldsymbol{Accuracy(A)} = \frac{\boldsymbol{V_P + V_N}}{\boldsymbol{V_P + V_N + F_P + F_N}} \quad (14)$$

3.4.3 Elección VAD. Para determinar el mejor algoritmo de detección de voz se tomaron 300 muestras, despreciándose las que tenían una duración menor a 0,5 s. Se usaron las demás muestras para calcular las métricas de segmentación. Como objeto de prueba se usaron dos audios de la base de datos, evaluados sobre 60 minutos de audio. Se tomaron dos fuentes el VAD usado más popularmente al que nombramos VAD_marshbrook y el basado en la aplicación WebBRTCVad de Google. Para esta validación como se dijo anteriormente, se podrían usar métricas específicas de VAD, ver (Meduri & Ananath, 2012).

En nuestro caso verificamos todas las medidas de error con el NIST error rate, el CER y la F-measure que son ampliamente usadas en medidas de frontera (Liu, Shriberg, 2007).

Tabla 2.

Comparación VAD: Audio 66

PARÁMETRO	AUDIO_66 WebRTCVad	AUDIO_66 marshbrook
Número muestras	123	147
Verdaderos positivos	67	49
Verdadero negativos	39	52
Falsos Positivos	15	27
Falsos Negativos	2	19
NIST error rate	0,246377	0,676471
CER	0,138211	0,312925
Precision	0,817073	0,644737
Recall	0,971014	0,720588
F-measure	0,887417	0,680556

Tabla 3.

Comparación VAD: Audio 85

PARÁMETRO	AUDIO_85 WebRTCVad	AUDIO_85 marshbrook
Número muestras	145	151
Verdaderos positivos	76	54
Verdaderos negativos	51	52
Falsos Positivos	17	30
Falsos Negativos	1	15
NIST error rate	0,233766	0,652174
CER	0,124138	0,298013
Precision	0,817204	0,642857
Recall	0,987013	0,782609
F-measure	0,894118	0,705882

3.4.4 Validación VAD. Es posible realizar la validación del VAD, tomando como objetivo separar primero esta clase y así de esta forma poder obtener la segunda clase que sería separar entre Niños y Adultos. Para realizar la validación de nuestro sistema usamos generalmente los siguientes pasos de acuerdo a (Liu, Shriberg, 2007):

-Verificación de parámetros de verdaderos positivos, verdaderos negativos, falsos positivos, falsos negativos validando punto a punto en el dominio del tiempo.

-Determinación de parámetros de precision, recall y accuracy.

Como observación cabe notar que en el Textgrid que se encuentra segmentado a mano se encuentran tres clases: Niño, Fono y Ruido de Fondo y al realizar la separación de Hablante y Ruido, solo quedan dos clases para analizar en el Textgrid que se automatizó con la aplicación. Por esta razón, se debe tener en cuenta que cuando en el Textgrid de referencia detecte Ruido y en ese caso sea Ruido no será un error no obstante no lo podrá comparar. El proceso realizado se muestra en la figura 9.

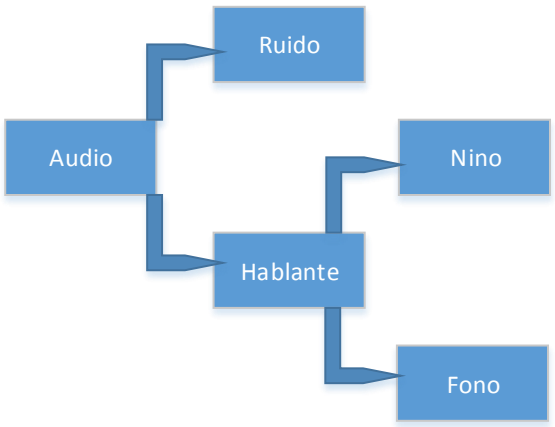

Figura 9. Criterios de selección para la evaluación

Una vez obtenido la separación de las dos clases es posible obtener entre Niños y Adultos la validación comparando archivos Textgrid que ya se han segmentado a mano, con los que se han hecho automáticamente. El proceso realizado se muestra en la figura 10.

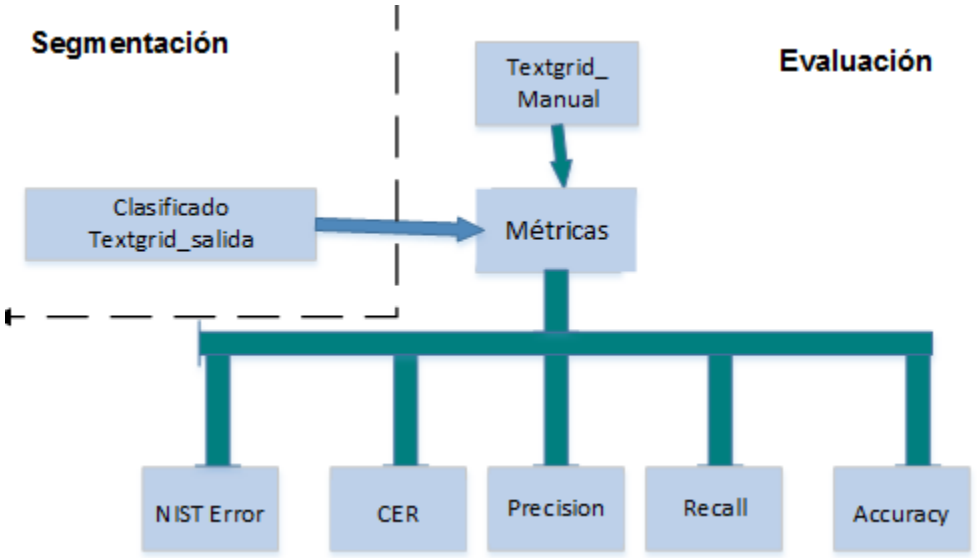

Figura 10. Diagrama de bloques – Evaluación

Para poder evaluarlo introducimos la comparación de dos clases entre el Textgrid a Mano y el Textgrid Clasificado, obteniendo las métricas propias de la segmentación (Liu, Shriberg, 2007).

3.5 Base de datos

La base de datos contiene 19 audios suministrados por la codirectora Janneth Suárez[1] en los que participan niños de entre 3 y 5 años, así como, la fonoaudióloga que los atiende. Los archivos de audio son de formato .wav. Están codificados en PCM con una frecuencia de muestreo de 8kHz en 32 bits. Tienen una duración promedio de 38 minutos. También incluye fuentes externas como grabaciones a personas que tienen una duración entre 10 a 25 segundos separando adultos de niños.

1 Fonoaudióloga. Instituto de la Comunicación Humana. UNAL

Para entrenar nuestro sistema tomamos de la base de datos solamente los segmentos de voz de niño y adulto, de la misma forma tenemos en cuenta la duración total y los géneros, debido a que tenemos audios con ruido utilizamos audios de la web exportados de la aplicación Youtube. De esta manera, dentro del modelo tendríamos datos con ruido, datos sin ruido y podríamos hacer pruebas para estos dos tipos de ambiente para construir un modelo más universal.

Tabla 4.

Clasificación respecto a duración y género

Duración y Género/Clase	Audios entrenados niño	Audios entrenados adulto
Numero de Audios totales	52 archivos	46 archivos
Duración Total	1072 [s]	1072[s]
Género masculino	440	720
Género femenino	632	352

El trabajo previo consiste en utilizar el software Praat para etiquetar el audio de tal forma que sea posible convertirlo en formato texto mostrando el hablante, clasificándolo por Ruido, Adulto y Niño. En total se introdujeron 46 Audios de Adultos y 52 Audios de Niños y se guardó el modelo. En otras palabras, se generan los resultados que sirven como referencia para verificar la fiabilidad del sistema desarrollado.

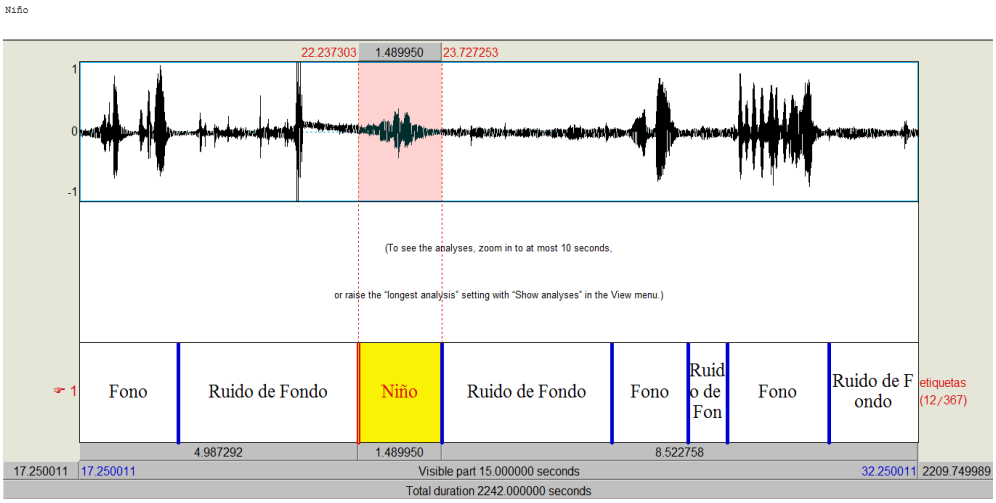

Figura 11. Etiquetado del archivo en Praat

Dado que todos los audios tenían por lo menos 2 hablantes fue necesario realizar una edición básica del audio usando este programa. En otras palabras, recortar y pegar los segmentos para un solo hablante. Con éstos, generar archivos de audio separados.

4. Resultados

4.1 Interfaz gráfica

En el software Python 2.7.12 utilizando la herramienta gráfica Tkinter (Shipman, 2014) podemos obtener nuestra interfaz gráfica de usuario que nos permite obtener un entrenamiento o una segmentación. En la segmentación podemos obtener los siguientes datos de tal manera que realizando el proceso (con un mismo audio) o (con audios diferentes) se pueda obtener un resultado.

4.2 Pruebas Iniciales

Para definir lo parámetros más apropiados es necesario hacer una variación de un parámetro dejando los otros dos sin moverse, tomando como parámetros importantes número de mezclas gaussianas, numero de cepstrales y paso temporal de nuestro programa.

4.2.1 Búsqueda mediante número de mezclas. Usando los audios de prueba se busca una mejora del desempeño aumentando o disminuyendo el número de mezclas gaussianas usadas dejando como máximo número de coeficientes 90 y paso 0.35 cercano al VAD.

Tabla 5.

Variación de número de mezclas gaussianas con paso y coeficientes definidos

#Mezclas	Precision	Recall	Accuracy	Promedio
2	0.335112	0.63510	0.49651	0,488907333
4	0.57082	0.64620	0.57438	0,597133333
8	0.95375	0.71256	0.74744	0,804583333
16	0.92587	0.67894	0.71254	0,77245
32	0.90393	0.71396	0.74072	0,786203333
64	0.96656	0.76754	0.81115	0,848416667
128	0.95798	0.80083	0.83718	0,86533
256	0.95574	0.75559	0.80387	0,8384
512	0.96093	0.80065	0.84166	0,867746667
1024	0.96093	0.77319	0.81626	0,850126667

4.2.2 Búsqueda mediante número de coeficientes. Usando los audios de prueba se busca una mejora del desempeño aumentando o disminuyendo el número de coeficientes de Mel que se extraen, tomando como base el escenario de las 512 mezclas gaussianas usadas anteriormente con el paso igual a 0.35. Utilizando la Conversacion_1.wav.

Tabla 6.

Variación de número de coeficientes, teniendo fijo el número de mezclas y el paso del programa

# Coeficientes	Precision	Recall	Accuracy	Promedio
50	0,49811	1.0	0.64386	0,71399
60	0.66086	1.0	0.75602	0,803826667
70	0.67201	1.0	0.76486	0,81229
80	0,79158	0,97025	0,77540	0,845743333
90	0.96093	0.80065	0.84166	0,867746667

4.2.3 Búsqueda mediante paso del programa. Usando los audios de prueba se busca una mejora del desempeño aumentando o disminuyendo el número de coeficientes de Mel que se extraen, tomando como base el escenario de las 512 mezclas gaussianas usadas anteriormente con el número de coeficientes igual a 90.

Tabla 7.

Variación del paso del programa teniendo como fijos el número de coeficientes y el número de mezclas gaussianas

Paso	Precision	Recall	Accuracy	Promedio
0.35	0.96093	0.80065	0.84166	0,867746667
0.40	0.92938	0.77009	0.78586	0,8284433
0.45	0.93853	0.72536	0.76188	0,80859
0.50	0.94037	0.78275	0.80424	0,842453333
0.55	0.96313	0.75956	0.79452	0,83907

4.3 Herramientas

4.3.1 Librería Scikit-Learn. Utilizando las herramientas de esta librería es posible obtener resultados de segmentación, clustering y ubicación de regiones de tal forma que sea posible obtener una separación de modelos de niño y adulto. Usando la herramienta gmmset podremos variar el número de mezclas gaussianas a implementar y podremos obtener variaciones en nuestro paso de programa ya sea para acercarlo o alejarlo del registro del VAD. Usamos los siguientes parámetros:

Tabla 8.

Parámetros utilizados en el apartado gmmset de Scikit-Learn

Parámetro	Valor
Número de iteraciones	200
Número de gaussianas	16
Concurrencia	4
Verbosidad	0

4.3.2 Librería Bob. En este caso usando los parámetros más clásicos, se usan un extractor de coeficientes usado en la librería BOB, con los siguientes parámetros.

Tabla 9.

Parámetros utilizados en los coeficientes de Mel

Parámetro	Valor
Coeficiente de preénfasis	0.95
Numero de coeficientes	50
Número de Filtros de Mel	56
Ancho de enventanado	30 [ms]
Frecuencia máxima de Filtros de Mel	6000 [Hz]

4.3.3 Librería WebRTCVad. Los parámetros utilizados fueron usados tanto en el entrenamiento como en la segmentación.

Tabla 10.

Parámetros utilizados en WebRTCVad

Parámetro	Valor
Unidad de frames de recorrido	30 [ms]
Unidad de Audio tomada en el VAD	300 [ms]
Mínima unidad de audio para comparación de VAD	500[ms]
Mínima unidad de audio para validación de VAD	500[ms]

4.4 Pruebas

4.4.1 Resultados VAD. Para este caso se probaron los audios de tal manera que se les asignó una ponderación de acuerdo a su duración, tomándose cinco conversaciones extraídas de internet en el ámbito sin ruido.

Tabla 11.

Métricas Conversación 1, Número de filtros de Mel = 55, MFCC = 80, GMM =128, 50 iteraciones

	Clase					
	Hablante			Ruido		
Paso	%Precision	%Recall	%Accuracy	%Precision	%Recall	%Accuracy
0.35	0.60433	1.0	0.61808	1.0	0.08340	0.61808

Tabla 12.

Métricas Conversación 2, Número de filtros de Mel = 55, MFCC = 80, GMM =128, 50 iteraciones

	Clase					
	Hablante			Ruido		
Paso	%Precision	%Recall	%Accuracy	%Precision	%Recall	%Accuracy
0.35	0.78945	0.98425	0.78541	0.68463	0.11520	0.78541

Tabla 13.

Métricas Conversación 3, Número de filtros de Mel = 55, MFCC = 80, GMM =128, 50 iteraciones

	Clase					
	Hablante			Ruido		
Paso	%Precision	%Recall	%Accuracy	%Precision	%Recall	%Accuracy
0.35	0.70042	0.97778	0.72565	0.89387	0.30912	0.72565

Tabla 14.

Métricas Conversación 4, Número de filtros de Mel = 55, MFCC = 80, GMM =128, 50 iteraciones

	Clase					
	Hablante			Ruido		
Paso	%Precision	%Recall	%Accuracy	%Precision	%Recall	%Accuracy
0.35	0.78590	0.36808	0.41008	0.18232	0.58423	0.41008

4.4.2 Pruebas para validación de niños y adultos. Se utilizaron pruebas de audios en los que no se contaba con el ruido de la base de datos.

Tabla 15.

Métricas Conversación 1, Número de filtros de Mel = 55, MFCC = 80, GMM =128, 50 iteraciones

	Clase					
	Niño			Adulto		
Paso	%Precision	%Recall	%Accuracy	%Precision	%Recall	%Accuracy
0.35	0.50466	0.99613	0.64663	0.99519	0.45004	0.64663

Tabla 16.

Métricas Conversación 2, Número de filtros de Mel = 55,MFCC = 80, GMM =128, 50 iteraciones.

	Clase					
	Niño			Adulto		
Paso	%Precision	%Recall	%Accuracy	%Precision	%Recall	%Accuracy
0.35	0.80123	0.83162	0.76811	0.70864	0.66499	0.76811

Tabla 17.

Métricas Conversación 3, Número de filtros de Mel = 55, MFCC = 80, GMM =128, 50 iteraciones

	Clase					
	Niño			Adulto		
Paso	%Precision	%Recall	%Accuracy	%Precision	%Recall	%Accuracy
0.35	0.96093	0.80065	0.84166	0.61909	0.89884	0.81381

Tabla 18.

Métricas Conversación 4, Número de filtros de Mel = 55, MFCC = 80, GMM =128, 50 iteraciones

	Clase					
	Niño			Adulto		
Paso	%Precision	%Recall	%Accuracy	%Precision	%Recall	%Accuracy
0.35	0.35913	0.92537	0.42186	0.81740	0.16825	0.42186

5. Recomendaciones

La librería Bob es la que más se recomienda usar para poder obtener mejores resultados de coeficientes de Mel, sin importar no solo su rapidez sino ciertas mejoras que pueden darse en los parámetros de los filtros de Mel que permiten recuperar la información del formante B3 (Frecuencias por encima de 2.1kHz hasta 6.0 kHz).

Al trabajar con tipos de entrenamiento lo más recomendable es usar un entrenamiento en igualdad de tiempo para las clases y en igualdad de género dada la poca diferencia entre las características del habla entre una mujer y un niño por tanto deben extraerse características del hombre que permitan diferencia los formantes.

El número de filtros de Mel que se usan influye en la obtención del formante F3, pues es muy probable que en ciertas voces haya intermitencias que no son deseadas es decir pequeños fragmentos de otras veces que se insertan.

A veces es muy necesario tal cual se hizo en el software cancelar ciertos ruidos muy pequeños si bien la agresividad del VAD es buena para evitar errores en fragmentos pequeños en los que ocurren silencios menores a 0.5, mejorando la fiabilidad.

Los niños entre 3 y 5 años para la realización de métodos de entrenamiento si bien presentan mayor dificultad en cuanto al reconocimiento entre ellos mismos (Safavi, Najafian, Hanani, 2008), que niños de edades superiores, pueden modelarse con niños de entre 6 y 9 años dado que si bien los fonemas son parecidos, la calidad de las mediciones con ruido ambiente son similares, además es más fácil de determinar el género en este tipo de edades que entre 3 y 6 años.

6. Conclusiones

Para obtener un grado de precisión alto en nuestro programa debemos buscar primero las mezclas gaussianas y luego ir aumentando el número de coeficientes de Mel. En nuestro caso se usan 512 mezclas gaussianas con 90 coeficientes que permiten obtener un mejor desempeño.

Utilizar solo una de las medidas que se han implementado para este caso, (precision, recall y accuracy) no suele ser muy confiable para tal objetivo es mejor usar el promedio de las tres, dado que si se tienen tres valores superiores al 80% se podrá tener en cuenta esta afirmacion.

Referencias Bibliográficas

Adami, A. G., Kajarekar, S. S., & Hermansky, H. (2002). A New Speaker Change Detection Method for Two-speaker Segmentation. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, *4*, 3908–3911. Retrieved from http://www.icsi.berkeley.edu/ftp/pub/speech/papers/icassp02-adami.pdf

Ali, M. Z. M., Mansor, W., Khuan, L. Y., & Zabidi, A. (2012). Simulink model of Mel Frequency Cepstral Coefficient analysis for extracting asphyxiated infant cry features. *2012 International Conference on Biomedical Engineering, ICoBE 2012*, (February), 475–478. doi:10.1109/IcoBE.2012.6179062.

Anguera, X., & Bozonnet, S. (2012). Speaker diarization: A review of recent research. *Audio, Speech, and …*, (August), 1–15. doi:10.1109/TASL.2011.2125954

Anjos, A. . Bob's Audio I/O Routines. [En línea]. Idiap Research Institute, Martigny, Switzerland. (2014) (Recuperado en 23 de Septiembre de 2017) Disponible en https://pythonhosted.org/bob/temp/bob.io.audio/doc/index.html

Bazyar, M. (2014). A New Speaker Change Detection Method in a Speaker Identification System for two-Speakers Segmentation. *2014 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)* , 141–145.

Das, P. P., Allayear, S. M., Amin, R., & Rahman, Z. (2016). Bangladeshi Dialect Recognition using Mel Frequency Cepstral Coefficient , Delta , Delta-delta and Gaussian Mixture Model, 359–364.

Dutton, S. Getting Started with WebRTCVad. [En línea]. WebRTC. (2012). (Recuperado en 14 de Junio de 2017) Disponible en https://www.html5rocks.com/en/tutorials/webrtc/basics/

García Herrero, A. (2015). Algoritmos para la estimación de modelos de mezclas gaussianas.

Gihai, S., & Sinha, R. (2010). Exploring the effect of differences in the acoustic correlates of adults' and children's speech in the context of automatic speech recognition. Eurasip Journal on Audio, Speech, and Music Processing, 2010(January 2010). https://doi.org/10.1155/2010/318785

Kwon, S., Narayanan, S., Analysis, S., & Media, I. (2003). A Method For On-Line Speaker Indexing Using Generic Reference Models, 2–5.

Liu & Shriberg., Comparing Evaluation Metrics for Sentence Boundary Detection. Dept of Computer Science, University of Texas at Dallas, Richardson, TX, U.S.A. International Computer Science Institute, Berkeley, CA, U.S.A.

Lu, L., & Zhang, H. (2005). Unsupervised speaker segmentation and tracking in real-time audio content analysis.

Martins, R., Trancoso, I., Abad, A., & Meinedo, .H (2009). Detection of Children's Voices Intituto Superior T ´, 77–80.

Meduri, S. S., & Ananth, R. (2011). A Survey and Evaluation of Voice Activity Detection Algorithms.

Meignier, S., Moraru, D., Fredouille, C., Bonastre, J. F., & Besacier, L. (2006). Step-by-step and integrated approaches in broadcast news speaker diarization. *Computer Speech and Language*, *20*(2-3 SPEC. ISS.), 303–330. http://doi.org/10.1016/j.csl.2005.08.002

Mello, C. A. (2010). Processamento digital de voz. Universidade Federal do Pernambuco Disponible en: http://www.cin.ufpe.br/~cabm/pds/PDS_Aula11_Voz.pdf

Moattar, M. H., & Homayounpour, M. M. (2012). A review on speaker diarization systems and approaches. *Speech Communication*, *54*(10), 1065–1103. doi:10.1016/j.specom.2012.05.002

Porras, D., & Mendoza, S. (2016). *Desarrollo de un sistema de verificación de hablante usando modelos de Mezclas Gaussianas*. Universidad Industrial de Santander. 62. Retrieved from http://tangara.uis.edu.co/biblioweb/tesis/2016/161127.pdf

Powers, D. M. W. (2007). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation, (December).

Reynolds, D. A., Quaitieri, T. F., & Dunn, R. B. (2000). Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, *10*(1-3), 19–41. http://doi.org/10.1006/dspr.1999.0361

Safavi, S., Najafian, M., Hanani, A., Russell, M., & Carey,. School of Electronic, Electrical and Computer Engineering, University of Birmingham. Speaker Recognition for Children's Speech, 15–18.

Schuster, S., Pancoast, S., Ganjoo, M., Frank, M. C., & Jurafsky, D. (2014). Speaker-independent detection of child-directed speech Department of Electrical Engineering , Stanford University ,

Stanford , CA ° Department of Psychology , Stanford University , Stanford , CA Department of Linguistics , Stanford University , Stanford , CA, 366–371.

Shipman, J. W. (2014). Tkinter 8.5 reference: a GUI for Python. New Mexico Tech Center. Retrieved from http://infohost.nmt.edu/tcc/help/pubs/tkinter/tkinter.pdf

Ugarte Echeverría, E. (2010). Implementación y comparación de algoritmos basados en técnicas biométricas de voz para reconocimiento del locutor en colaboración con la empresa IECISA. Escuela Técnica Superior De Ingenieros Industriales y de Telecomunicación. Universidad Pública de Navarra. Pamplona, Navarra, 1-102.

Wu, Y., Zhou, X., & Li, T. (2014). A Real-time Speaker Recognition System with GUI. Tsinghua University. Retrieved from https://github.com/ppwwyyxx/speaker-recognition

Apéndice

Apéndice A
Tutorial de instalación de herramientas

Sistema:

- Linux Ubuntu 14.04
- Python 2.7.12
- Compilador Pycharm (opcional)

Pasos de instalacion para Ubuntu

Utilizando Ctrl + Alt + T se genera una terminal para poder escribir las instrucciones de instalacion

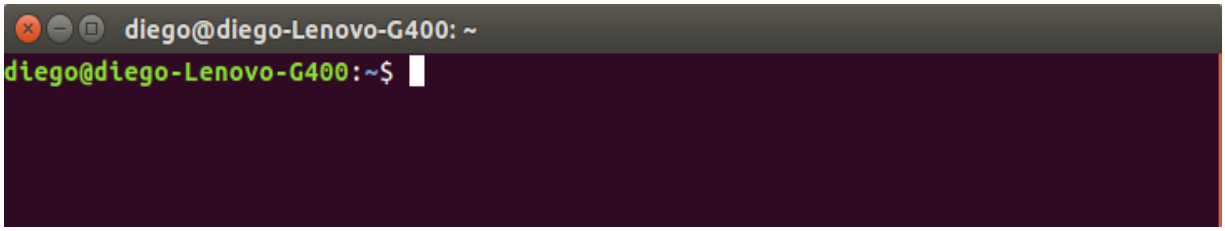

Figura. 12 . Ingreso a la terminal de Ubuntu

- Instalación de herramientas de audio

Digitamos en la consola de ubuntu

sudo apt-get install build-essential autoconf libtool pkg-config python-opengl python-imaging python-pyrex python-pyside.qtopengl idle-python2.7 qt4-dev-tools qt4-designer libqtgui4 libqtcore4 libqt4-xml libqt4-test libqt4-script libqt4-network libqt4-dbus python-qt4 python-qt4-gl libgle3 python-dev libssl-dev

- Instalación de librería WebRTCVad

A través de la instrucción

sudo apt-get install webrtcvad

Confirmando, mediante

pip install webrtcvad

- Instalación de librería BOB

Se deben instalar de forma adecuada los siguientes paquetes

- scipy

sudo apt-get install scipy

- numpy

$ pip install numpy

- Se instalan las correspondientes dependencias de ubuntu

sudo apt-get install libboost-all-dev
sudo apt-get install libblitz0-dev
sudo apt-get install cmake
sudo apt-get install libhdf5-serial-dev
sudo apt-get install libtiff5
sudo apt-get install libtiff5-dev
sudo apt-get install libtiff-tools
sudo apt-get install giflib-dbg

- Instalar las dependencias de bob

$ pip install bob.extension
$ pip install bob.blitz
$ pip install bob.core
$ pip install bob.io.base
$ pip install bob.io.image

$ pip install bob.io.audio

sudo pip2 install bob.measure

Nota:

Puede ocurrir el siguiente error dada la no instalación del paquete sox

RuntimeError: pkg-config package `sox' was not found

Se deben corroborar tres cosas:

- Correcta instalación del numpy, se actulizan las librerías a través del siguiente comando

sudo easy-install -U numpy

- Instalación del Sox

sudo apt-get install sox

- Verificar que el sistema use Python puro y no Anaconda

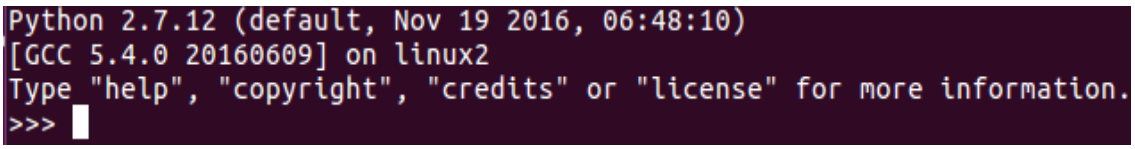

Fig. 13 . Terminal de Ubuntu que muestra el tipo de instalación de Python

Para el caso de Anaconda se usa la siguiente orden

conda install bob.io.audio

- Instalar librerias y componentes adicionales

sudo apt-get install python-software-properties #to install "add-apt-repository"

sudo add-apt-repository ppa:biometrics/bob

sudo apt-get update

- Reinstalar Bob

sudo apt-get install bob

- Instalación de la librería PyAudio

$ pip install pyaudio

O se instala la ultima versión de PyAudio

sudo apt-get install python-pyaudio python3-pyaudio

- Problemas comunes

error: command 'x86_64-linux-gnu-gcc' failed with exit status 1

Se soluciona ejecutando

sudo apt-get install portaudio19-dev

Posteriormente volvemos a instalar nuestra librería

$ pip install pyaudio

- Instalación de la librería pysst

$ pip install pysst

- Instalación de la librería scikit-learn y del paquete scikits.talkbox

$ pip install scikit-learn

$ pip install scikits.talkbox

Apéndice B
Manual para uso de interfaz de entrenamiento

1. Entrenamiento de audio:

1.1 Entrenamiento de la primera clase:

Una vez pulsado el botón que se encuentra señalando la clase que necesitamos (sea niño o adulto) se procede a obtener a buscar los audios con los cuales vamos a entrenar y le damos click en Almacenar.

Figura 14. Entrenamiento primera clase (Niño)

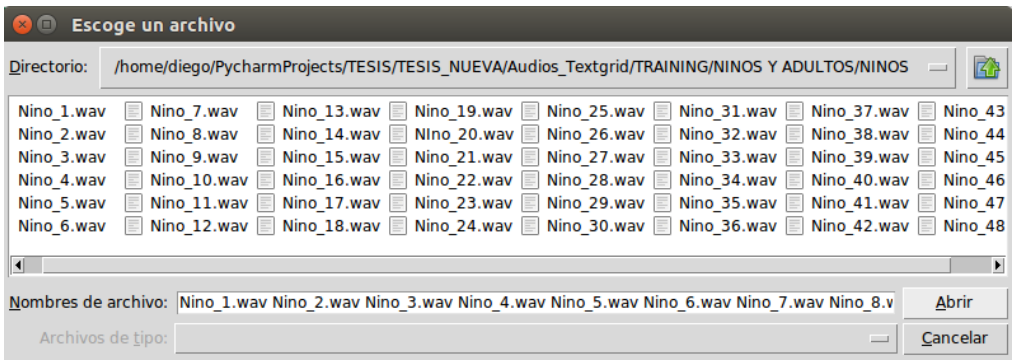

Figura 15. Selección de audios de entrenamiento (Niño)

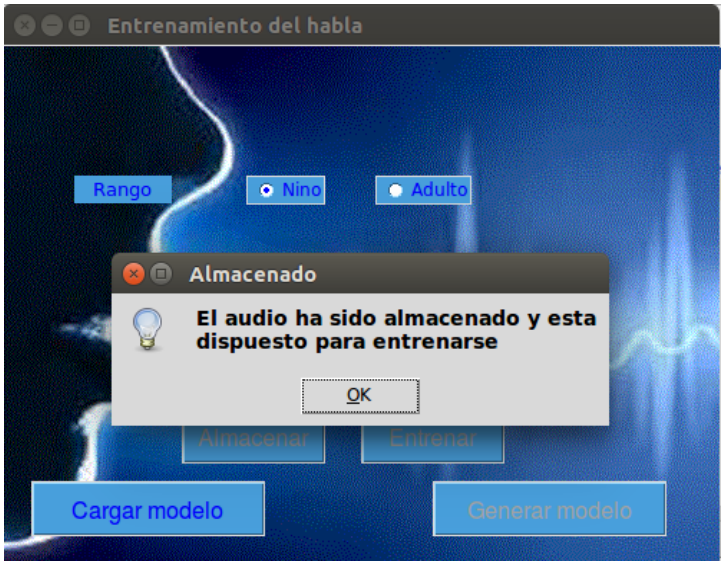

Figura 16. Entrenamiento de niños terminado

1.2 Entrenamiento de la segunda clase:

Se realiza el mismo procedimiento anterior para la segunda clase seleccionando los audios con los que vamos a entrenar

Figura 17. Entrenamiento segunda clase (Niño)

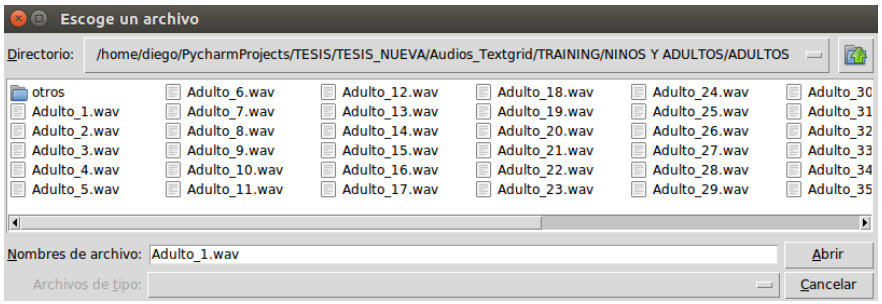

Figura 18. Selección de audios de entrenamiento (Adulto)

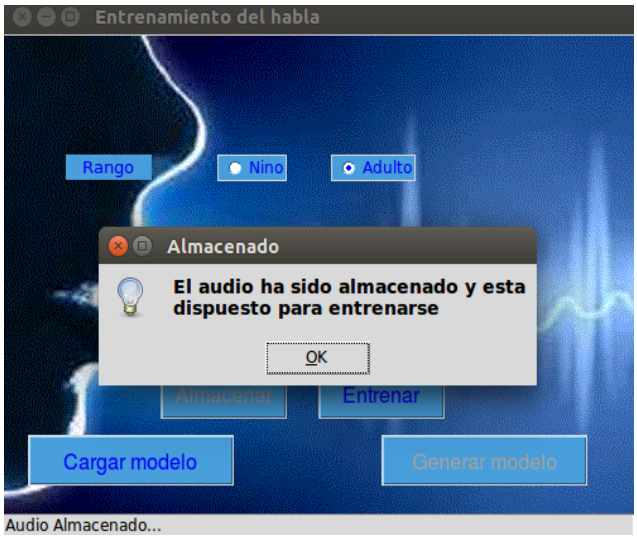

Figura 19. Entrenamiento de adultos terminado

1.3 Datos de entrenamiento: Una vez se hayan entrenado las dos clases es posible generar los datos usando el botón “Entrenar”, que generará los datos del UBM a utilizar.

Figura 20. Generación de datos del UBM

Figura 21. Datos del UBM generados

2. Generación de modelo:

Pulsando el botón “Generar modelo” se puede obtener un archivo en formato binario.mdl que contendrá la información entrenada. Aparecerá un mensaje en la barra de estatus mostrando que se ha generado.

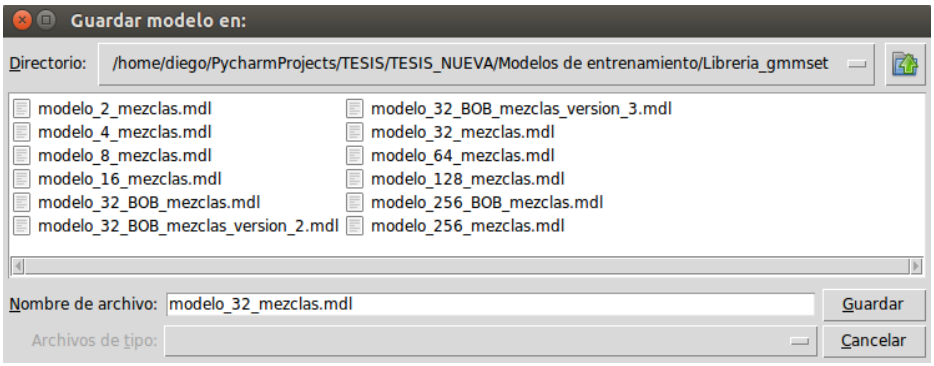

Figura 22. Generación del modelo

3. Carga de modelo:

Pulsando el botón "Generar modelo" se puede obtener un archivo en formato binario .mdl que contendrá la información entrenada. Aparecerá un mensaje en la barra de estatus mostrando que se ha generado.

Figura 23. Modelo generado

PROCESO DE SEGMENTACIÓN

1. Ingresamos al botón Segmentación de la Interfaz Inicial.

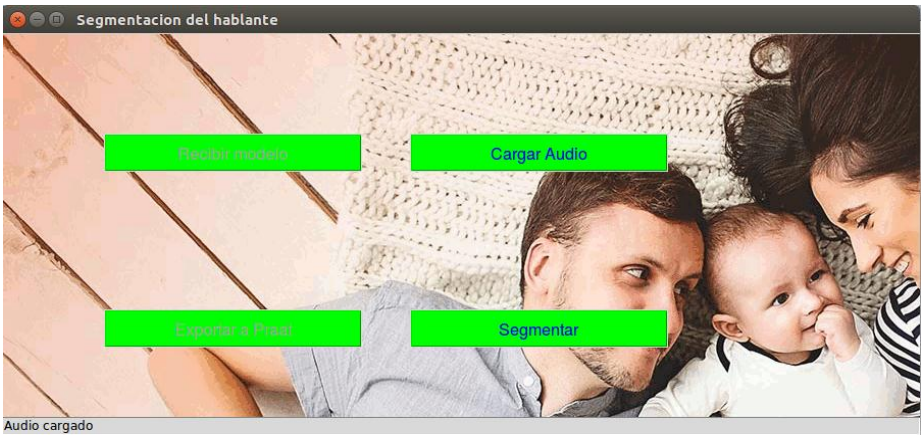

Figura 24. Interfaz inicial de segmentación

2. Pulsamos “Cargar Audio” para poder subir el audio que queremos segmentar, el cual cargara en la consola

3. Pulsamos “Segmentar” y así obtenemos la salida de nuestra etiqueta, tiempo inicial y tiempo final.

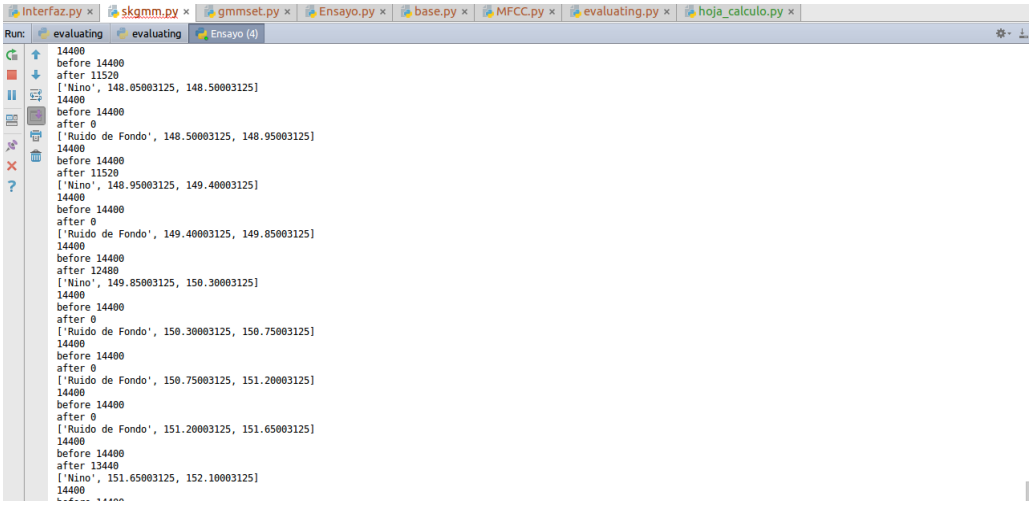

Figura 25. Etiquetas generadas por el programa

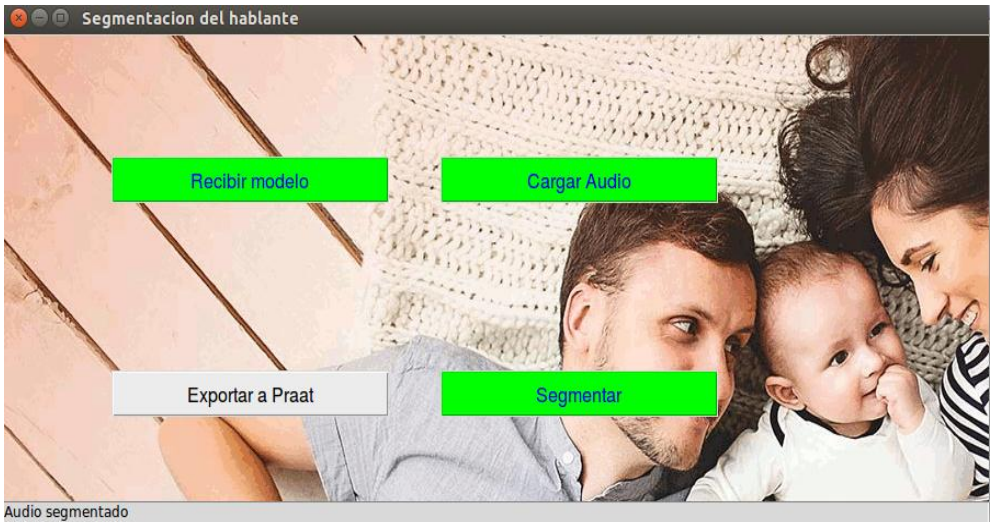

Figura 26. Exportación a Praat de los resultados

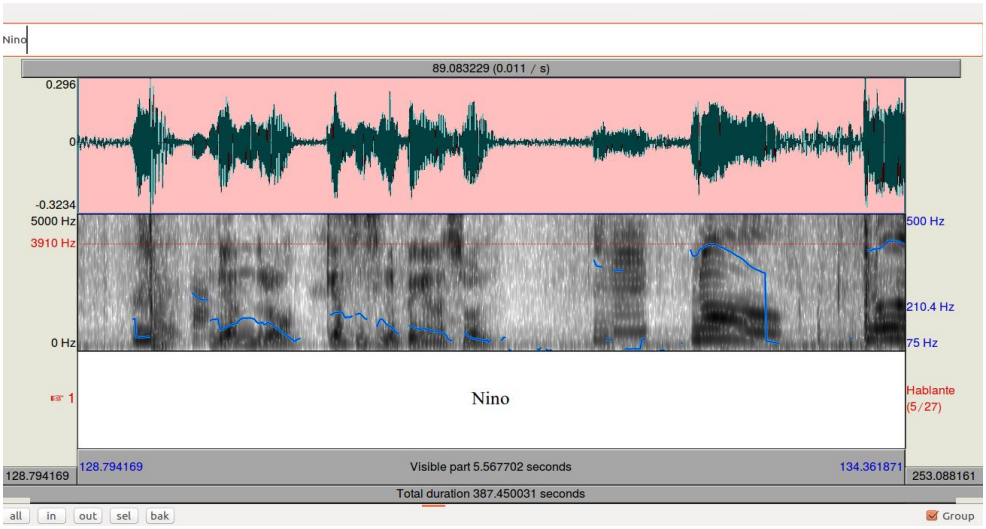

Figura 27. Datos visualizados en Praat