UNIVERSIDAD DE CUENCA
FACULTAD DE INGENIERÍA
ESCUELA DE INFORMÁTICA

"APLICACIÓN DE TECNOLOGÍAS DE SEGMENTACIÓN DE AUDIO Y RECONOCIMIENTO AUTOMÁTICO DE DIALECTO PARA LA OBTENCIÓN DE INFORMACIÓN DE DIÁLOGOS CONTENIDOS EN AUDIO"

TRABAJO DE TITULACIÓN
PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO DE
SISTEMAS

AUTOR:

ERIK ALEJANDRO SIGCHA QUEZADA
C.I. 0104947114

DIRECTOR Y CO-AUTOR:

ING. JORGE MAURICIO ESPINOZA MEJÍA, PHD.
C.I. 0102778818

CUENCA – ECUADOR

2017

Resumen

El interés de la comunidad científica en la identificación de contenidos audiovisuales ha crecido considerablemente en los últimos años, debido a la necesidad de ejecutar procesos automáticos de clasificación y monitoreo del cada vez mayor contenido transmitido por diferentes medios como televisión, radio e internet. En este artículo se propone una arquitectura para la extracción de información a partir de audio, con la finalidad de aplicarlo al análisis de contenidos televisivos en el contexto ecuatoriano. Para esto, se definen dos servicios, un servicio de segmentación de audio y un servicio de transcripción. El servicio de segmentación identifica y extrae los segmentos de audio que contienen narrativa, música, o narrativa sobre música. Mientras que, el servicio de transcripción hace un reconocimiento de los segmentos de tipo narrativa para obtener su contenido como texto. Estos servicios y las herramientas que los conforman han sido evaluados con el fin de medir su rendimiento y, en el caso de las herramientas usadas, definir cuál de estas es la que mejor se ajusta a la definición de la arquitectura. Los resultados de las evaluaciones realizadas sobre la arquitectura propuesta demuestran que la construcción de un sistema de reconocimiento de habla que haga uso de distintas herramientas de código abierto existentes ofrece un mayor nivel de precisión que un servicio de transcripción de disposición general.

Palabras clave: *Reconocimiento Automático del Habla, Segmentación Automática de Audio, Python, Servicios Web, Habla a Texto, Televisión Digital, Análisis de Audio.*

Erik Alejandro Sigcha Quezada

Abstract

The interest of the scientific community in the identification of audiovisual content has grown considerably in recent years, due to the need to execute automatic classification and monitoring processes on the increasing content broadcasted by different media such as television, radio and internet. This article proposes an architecture for extracting information from audio, with the purpose of applying it to the analysis of television contents in the Ecuadorian context. For this, two services are defined, an audio segmentation service and a transcription service. The segmentation service identifies and extracts audio segments containing speech, music, or speech with musical background. Whereas, the transcription service recognizes the speech segments to obtain its content as text. These services and the tools that conform them have been evaluated in order to measure their performance and, in the case of the tools used, to define which of these is the one that best fits the definition of the architecture. The results of the evaluations carried out on the proposed architecture demonstrate that the construction of a speech recognition system, that makes use of different existing open source tools, offers a higher level of precision than a general availability transcription service.

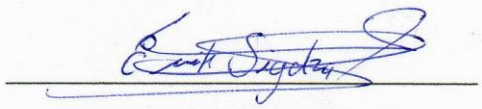
Keywords: *Automatic Speech Recognition, Audio Segmentation, Python, Web Services, Speech to Text, Digital TV, Audio Analysis.*

Erik Alejandro Sigcha Quezada

Índice General

Yo, ERIK ALEJANDRO SIGCHA QUEZADA, autor del Trabajo de Titulación "Aplicación de Tecnologías de Segmentación de audio y Reconocimiento automático de dialecto para la obtención de información de diálogos contenidos en audio", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 10 de mayo de 2017

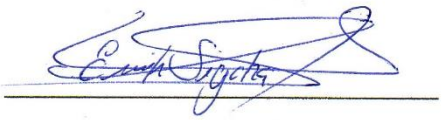


Erik Alejandro Sigcha Quezada

C.I: 0104947114

Yo, ERIK ALEJANDRO SIGCHA QUEZADA, autor del Trabajo de Titulación "Aplicación de Tecnologías de Segmentación de audio y Reconocimiento automático de dialecto para la obtención de información de diálogos contenidos en audio", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniero de Sistemas. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor

Cuenca, 10 de mayo de 2017



Erik Alejandro Sigcha Quezada

C.I: 0104947114

Aplicación de Tecnologías de Segmentación de Audio y Reconocimiento Automático de Dialecto para la obtención de información de diálogos contenidos en audio.

Erik Alejandro Sigcha Quezada[1], **Jorge Mauricio Espinoza Mejía**[2]

[1]Facultad de Ingeniería, Carrera de Ingeniería de Sistemas, Universidad de Cuenca, Av. 12 de abril y Agustín Cueva, Campus Central, Ciudadela Universitaria, Cuenca, Ecuador

[2]Departamento de Ciencias de la Computación, Universidad de Cuenca, Av. 12 de abril y Agustín Cueva, Edificio Tecnológico de Ingeniería, tercer piso, Ciudadela Universitaria, Cuenca, Ecuador

Autores para correspondencia: erik.sigchaq@ucuenca.ec, mauricio.espinoza@ucuenca.edu.ec

RESUMEN

El interés de la comunidad científica en la identificación de contenidos audiovisuales ha crecido considerablemente en los últimos años, debido a la necesidad de ejecutar procesos automáticos de clasificación y monitoreo del cada vez mayor contenido transmitido por diferentes medios como televisión, radio e internet. En este artículo se propone una arquitectura para la extracción de información a partir de audio, con la finalidad de aplicarlo al análisis de contenidos televisivos en el contexto ecuatoriano. Para esto, se definen dos servicios, un servicio de segmentación de audio y un servicio de transcripción. El servicio de segmentación identifica y extrae los segmentos de audio que contienen narrativa, música, o narrativa sobre música. Mientras que, el servicio de transcripción hace un reconocimiento de los segmentos de tipo narrativa para obtener su contenido como texto. Estos servicios y las herramientas que los conforman han sido evaluados con el fin de medir su rendimiento y, en el caso de las herramientas usadas, definir cuál de estas es la que mejor se ajusta a la definición de la arquitectura. Los resultados de las evaluaciones realizadas sobre la arquitectura propuesta demuestran que la construcción de un sistema de reconocimiento de habla que haga uso de distintas herramientas de código abierto existentes ofrece un mayor nivel de precisión que un servicio de transcripción de disposición general.

Palabras clave: *Reconocimiento Automático del Habla, Segmentación Automática de Audio, Python, Servicios Web, Habla a Texto, Televisión Digital, Análisis de Audio.*

1. INTRODUCCIÓN

Actualmente, Ecuador se encuentra en un proceso de adopción de Televisión Digital Terrestre por medio del estándar Japonés-Brasileño (ISDB-Tb), lo cual motiva el desarrollo de investigaciones relacionados con la interacción multimedia y accesibilidad a la información presentada por un medio televisivo. Además, desde el año 2013, el país ha adoptado políticas que se enfocan al monitoreo de contenido dentro de los medios a través de la Ley Orgánica de Comunicación (artículos 32, 65, 66 y 69). Dentro de este contexto, analizar los contenidos que se muestran en una transmisión televisiva es un tema fundamental para poder presentar una programación a una determinada audiencia en un momento adecuado. Por ejemplo, en el trabajo presentado en (Campoverde Llanos & Guerrero Fernández de Córdova, 2015), se aplicaron tecnologías de reconocimiento de objetos para identificar la presencia de armas de fuego en video, para clasificar contenidos violentos. Como conclusión de este trabajo, se expuso una línea de investigación futura orientada al análisis de audio, el cual combinado al análisis de video puede obtener resultados más acertados.

En artículos como (Robert-Ribes, 1998) y (Imai et al., 2004) se analizan las ventajas y desventajas que surgen al realizar un análisis sobre el audio de un video mediante la aplicación de tecnologías de Reconocimiento Automático de Habla (RAH) con el objetivo de obtener una herramienta de subtitulado automático. Existen también varios trabajos orientados a comparar herramientas de código abierto para RAH como (Kłosowski et al, 2014) y (Gaida et al., 2014), pero son pocos los recursos de información que aplican dichas herramientas al idioma español y ejecuten una evaluación de precisión en las transcripciones (Varela et al., 2003; Niculescu & de Jong, 2008). Además, aplicativos descritos en (Schuster, 2010) y (Aron, 2011) evidencian el gran avance que se ha dado en el campo del reconocimiento de voz en los últimos años, lo que ha permitido aplicar la tecnología de procesamiento natural de lenguaje, en asistentes personales instalados en teléfonos inteligentes. Basado en estos recursos y los resultados alcanzados por diferentes propuestas, en este trabajo se propone estudiar la aplicación de herramientas para segmentación de audio y RAH para el análisis de contenidos de señales televisivas y radiales en español latinoamericano.

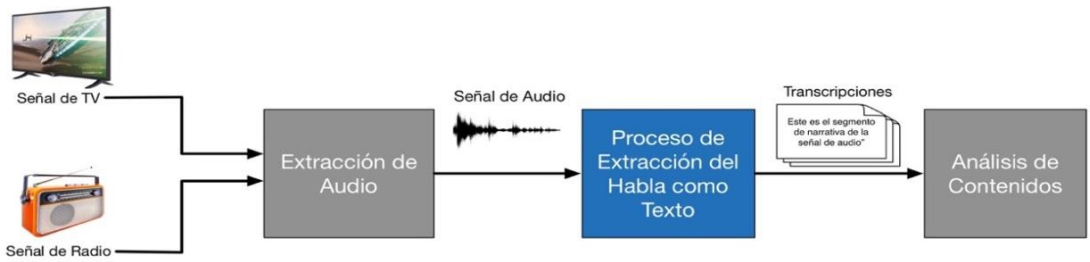


Figura 1. Flujo del proceso extracción de contenidos televisivos o radiales para su análisis.

De manera general un proceso de extracción y análisis de contenidos a partir de señales de audio, puede ser enfocado a través de los módulos descritos en la Figura 1. Como paso inicial la recepción de las señales de TV o Radio, deberá ser procesada por un *módulo de extracción de audio*, encargado de obtener únicamente la señal de audio de las transmisiones, luego esta señal es enviada a un *módulo de extracción de habla*, el cual obtendrá los textos de las transcripciones de dichos audios. Como paso final se envían dichas transcripciones a un proceso de *análisis de contenidos*. El trabajo descrito en este artículo se enfoca únicamente en el desarrollo del módulo del *proceso de extracción del habla como texto* (bloque central en la Figura 1), el cual recibirá como insumo de entrada una señal de audio y generará las transcripciones correspondientes.

Según (Robert-Ribes, 1998), el análisis de los contenidos de una señal de audio debe considerar recursos de información útiles para realizar la clasificación de elementos tales como ruidos, música de fondo, sonidos de ambiente y diálogos. Aquí se ve la necesidad de aplicar un pre-procesamiento a la señal de audio, con el fin de identificar qué secciones del audio contienen estos elementos. Para lograr esto, artículos como (Giannakopoulos, 2015), (Gallardo-Antolín & Hernández, 2010) y (Bachu et al., 2008), presentan herramientas y técnicas de análisis de audio como: regresión, clasificación y segmentación de audio, que pueden resultar útiles al momento de definir un proceso de identificación de contenidos. Luego, considerando los diálogos presentes en un audio como uno de los principales recursos de información, resulta fundamental obtener una transcripción de dichos audios, donde el reto principal es desarrollar una herramienta de software que permita realizar un proceso de subtitulado automático de todo el contenido. Por esta razón, el presente trabajo adopta un proceso que permita, en una primera

instancia, identificar las secciones del audio que contienen diálogos y luego aplicar, a estas secciones, algoritmos de subtitulado automático.

En el presente artículo se presenta la arquitectura de un servicio que ejecuta el proceso de extracción de habla por medio de la aplicación de las tecnologías de segmentación automática de audio y RAH, las cuales asisten en la ejecución de los procesos de identificación de contenidos y subtitulado automático, respectivamente. Para esto, se han tomado como base trabajos previos como (Kłosowski et al., 2014), (Gaida et al., 2014) y (Giannakopoulos, 2015), con el fin de definir el o los productos de software de código abierto disponibles y que se ajusten de mejor manera a la definición de esta arquitectura orientada al análisis de contenidos de difusión televisiva o radial. Adicionalmente, se han ejecutado diferentes evaluaciones para medir el rendimiento de las herramientas seleccionadas y de la arquitectura descrita en este trabajo, con la finalidad de hacer una comparación de estos resultados con los resultados obtenidos al aplicar una herramienta similar de disposición general.

El resto de este artículo está organizado de la siguiente forma. La sección 2 resume los conceptos básicos de las tecnologías de segmentación automática de audio y RAH. En la sección 3 se hace una revisión de los trabajos relacionados con la propuesta presentada en este artículo. La arquitectura del *servicio de segmentación y transcripción de audio* y los módulos que la conforman se describen en la sección 4. La sección 5 describe las evaluaciones realizadas sobre el servicio de segmentación y transcripción de audio y las herramientas utilizadas para su implementación. Como parte final, en la sección 6 se presentan las conclusiones y las líneas futuras de investigación.

2. REVISIÓN DE TECNOLOGÍAS APLICADAS.

Los elementos tecnológicos que conforman el bloque de procesamiento para la extracción de habla contenida en audio se basan en las tecnologías de segmentación automática de audio y RAH. En esta sección se revisan los conceptos básicos necesarios para comprender cada una de estas tecnologías.

2.1. Segmentación Automática de Audio.

La segmentación automática de audio consiste en dividir una señal de audio digital en segmentos, los cuales contendrán información de audio de algún tipo específico como diálogos de personas, música, vocalizaciones de animales, sonidos de ambiente o ruidos (Fakotakis et al. 2014). El número de tipos de segmentos identificables en este proceso depende del escenario de aplicación de esta tecnología.

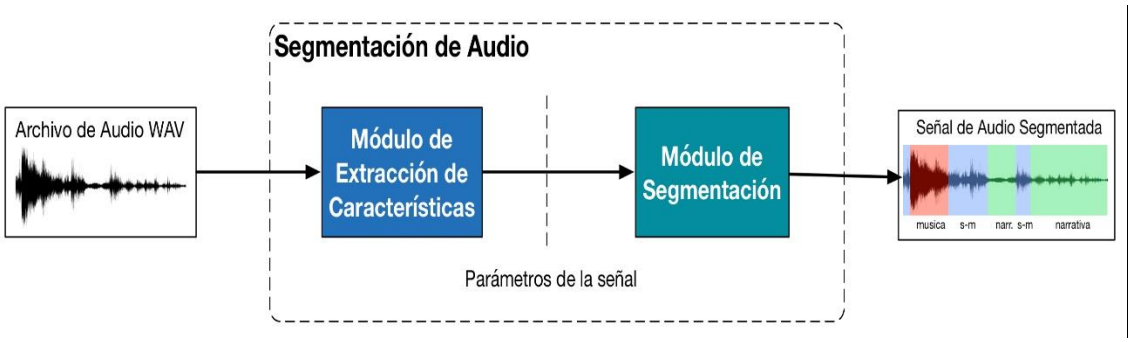


Figura 2. Arquitectura general de un sistema de segmentación automática de audio.

Arquitectura Básica.

A pesar de que no es posible hablar de una arquitectura general para soportar la segmentación automática de audio, un buen número de propuestas encontradas en la literatura ((Castán et al., 2014), (Kulkarni et al., 2001) o (Gallardo-Antolín & Hernández, 2010)) coinciden en el uso de dos procesos para ejecutar la división del audio en segmentos identificables: extracción de características y segmentación. Otros autores como (Pikrakis, 2008), (Fakotakis et al. 2014) y (Abad et al., 2008) completan esta configuración básica con módulos adicionales, para ejecutar tareas de segmentación inicial o de post-procesamiento del audio. En este trabajo se adopta como elementos tecnológicos indispensables para ejecutar el proceso de segmentación del audio, los módulos identificados en la Figura 2. A continuación se describe en detalle los módulos ilustrados en la figura.

- **Módulo de Extracción de características:** este módulo tiene como objetivo representar la señal de audio de manera discreta, para esto la señal es dividida en marcos (*frames*) superpuestos de muestras de audio y por cada marco se obtiene un vector de características paramétricas. Los descriptores de audio más utilizados para obtener los vectores de características paramétricas son: los coeficientes MFCC (Mel-Frequency Cepstral Coefficients) y la tasa de cruces por ceros (ZCR – Zero Crossing Rate) (Fakotakis et al. 2014).
    - **Coeficientes MFCC:** son parámetros espectrales basados en la escala de Mel, la cual se construye siguiendo un esquema de funcionamiento parecido al del oído humano (Barrobés, 2012).
    - **Tasa de Cruces por Ceros:** es la medida del número de veces que la amplitud de una señal toma el valor de cero en un determinado intervalo de tiempo (Bachu et al., 2008).
- **Módulo de Segmentación:** se encarga de analizar los vectores de características paramétricas extraídos para determinar en qué puntos la señal de audio cambia de un tipo acústico a otro, es decir, identifica el inicio, el fin y el tipo de cada segmento. Para resolver el problema de la segmentación de audio se utiliza uno de dos enfoques, el enfoque de segmentación y clasificación o el enfoque de segmentación por clasificación (Castán et al., 2014).
    - **Enfoque de Segmentación y Clasificación:** este enfoque realiza la segmentación en dos pasos. En el primer paso se detectan los límites de cada segmento y en el segundo se clasifica cada segmento delimitado aplicando algoritmos de aprendizaje automático (Castán et al., 2014).
    - **Enfoque de Segmentación por Clasificación:** consiste en clasificar pequeñas muestras consecutivas con tamaño fijo de la señal de audio. Para esto se extraen vectores de características de cada muestra y se clasifica cada una de manera individual, en lugar de clasificarlos por grupos (Fakotakis et al., 2014; Goméz Rincon, 2015).

2.2. Reconocimiento Automático del Habla.

El Reconocimiento Automático del Habla es el proceso de convertir una señal de voz a una secuencia de palabras, por medio de un algoritmo implementado como un programa de computadora (Anusuya & Katti, 2010).

Arquitectura Básica.

Un reconocedor automático del habla puede ser visto como un bloque de procesamiento que recibe una señal de audio de entrada y produce una transcripción (texto) como salida (Barrobés, 2012). Autores como (Anusuya & Katti, 2010), (Huang & Deng, 2010) y (Saon & Chien, 2012) especifican diferentes arquitecturas en sus trabajos, las cuales se diferencian por el grado de detalle que se ha aplicado para definirlas. Sin embargo, en este trabajo se reconoce que una arquitectura compuesta principalmente por un módulo de extracción de características y un módulo de decodificación (como se ilustra en la Figura 3), puede cubrir la necesidad de una gran parte de reconocedores de habla automáticos.

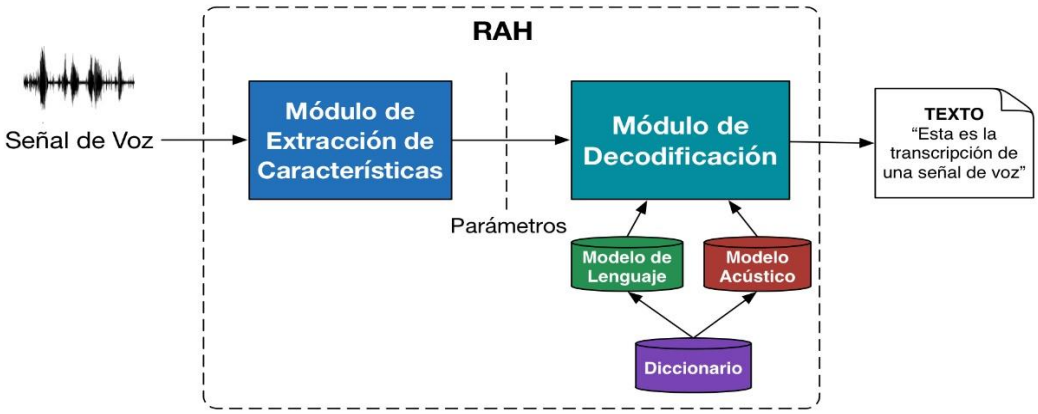


Figura 3. Arquitectura básica de un reconocedor automático de habla.
Fuente: Adaptado de Barrobés, H. D., & Ruiz, M. (2012).

A continuación se explican en detalle el proceso que realiza cada módulo y los demás componentes que conforma la arquitectura general básica de un RAH.

- **Módulo de extracción de características:** Al igual que en la segmentación automática de audio, el proceso de reconocimiento automático del habla debe primero convertir la señal de voz en una secuencia discreta de parámetros. Este módulo es el encargado de este proceso en la arquitectura, para lo cual utiliza dos fases: la fase de entramado y la fase de estimación de parámetros (Anusuya & Katti, 2010; Gaikwad et al., 2010). En la fase de entramado se obtienen pequeños tramos de la señal original, para lograr esto, la señal se multiplica, cada cierto intervalo de tiempo, por otra señal denominada ventana (diferente de cero en un pequeño intervalo de tiempo). En la fase de estimación de parámetros se calculan valores o características de cada tramo. Los coeficientes más aplicados para la obtención de características de una señal de voz para RAH, son los coeficientes MFCC (ver sección 2.1) (Ittichaichareon et al., 2012).
- **Módulo de Decodificación:** se encarga de encontrar la secuencia de palabras que con mayor probabilidad producen la secuencia de vectores de características obtenidos por el módulo de extracción de características (Anusuya & Katti, 2010;

Gaikwad et al., 2010; Barrobés, 2012). Para esto, se requiere dos modelos probabilísticos, el modelo de lenguaje y el modelo acústico. Estos modelos se generan en una etapa de entrenamiento utilizando un conjunto de datos denominado corpus[1].

- **Diccionario:** el diccionario de un RAH es un listado de todas las palabras que este será capaz de reconocer (Barrobés, 2012), junto con su descomposición en unidades acústicas como sílabas o fonemas. Si una palabra no está en el diccionario y aparece en el reconocimiento, esta se denomina "palabra fuera del vocabulario (*OOV – out of vocabulary*)". La Tabla 1 muestra un ejemplo de cómo se realiza la descomposición en fonemas de algunas palabras contenidas en un diccionario.

Palabras	Descomposición fonética
abejas	a b e x a s
abierto	a b i e r(t o
abrazo	a b r(a s o
…	…

Tabla 1. Ejemplo de la estructura de un diccionario.

- **Modelo Acústico:** El modelo acústico es una representación de todos los sonidos o unidades acústicas del lenguaje que se pueden reconocer. Esto significa que, para cada unidad acústica utilizada en la conformación del diccionario de un RAH se debe construir un modelo que relacione dicha unidad acústica con el vector de características que se producirá al pronunciarla. El enfoque de reconocimiento de patrones es el más utilizado para la construcción de los modelos acústicos para RAH (Anusuya & Katti, 2010; Gaikwad et al., 2010; Barrobés, 2012). Clasificados dentro de este enfoque se encuentran los denominados Modelos Ocultos de Markov (HMM – *Hidden Markov Models*), los cuales han sido utilizados en reconocedores como Kaldi (Povey et al., 2011), CMU Sphinx (Walker et al., 2004), HTK (Young et al., 2006), SPOJUS++ (Fujii et al., 2011) y AT&T Watson Recognizer (Goffin et al., 2005).
    - **Modelos HMM:** son modelos estadísticos útiles para representar datos secuenciales en el tiempo o en el espacio como videos, música, imágenes, texto y señales de habla (Saon & Chien, 2012).
- **Modelo de Lenguaje:** El modelo de lenguaje es una representación probabilística de las relaciones que existen entre las palabras del diccionario. En otras palabras, en el modelo de lenguaje se especifican todas las combinaciones de palabras con sentido semántico que pueden formarse a partir del diccionario y su probabilidad de aparición. Para la creación del modelo de lenguaje se aplican los n-grams, que se definen como secuencias de n palabras, siendo 2-gram para frases de dos palabras y 3-gram para frases de tres palabras, etc. (Jurafsky & Martin, 2014). Los n-gram más usados para el RAH son los 2-gram (*bigram*) y 3-gram (*trigram*) (Barrobés, 2012).

Una vez descritos los conceptos básicos de las tecnologías de segmentación automática de audio y RAH, a continuación se describen los trabajos relacionados con esta propuesta.

[1] Un corpus es un conjunto estructurado de ejemplos reales de uso de la lengua.

3. TRABAJOS RELACIONADOS

En esta sección se describen algunos trabajos relacionados con el proceso de creación o aplicación de herramientas de segmentación de audio y reconocimiento automático del habla para el análisis de información en medios audiovisuales.

3.1. Segmentación Automática de Audio

Existen varios trabajos que abordan el tema de la creación de sistemas de segmentación de audio y cada uno difiere en los tipos de segmentos identificables, pues cada uno de los trabajos analizados tiene un propósito específico. Por ejemplo, (Kulkarni et al., 2001) aplica la segmentación para obtener los componentes estructurales de canciones por medio de la identificación de segmentos vocales, segmentos no vocales y silencios. En (Bietti et al., 2015) se presentan dos experimentos de segmentación, el primero se basa en la identificación del sonido de eventos como ruidos de puerta, sofá o llaves. El segundo experimento es una segmentación que identifica la ocurrencia de notas musicales. En cambio en (Bachu et al., 2008) se tiene como objetivo separar las secciones que contienen voz, de las secciones que contienen silencios o ruidos de respiración de una señal de tipo habla, ya segmentada previamente, aplicando la tasa de cruces por ceros.

Otro aspecto diferenciador en los trabajos de segmentación de audio es con respecto a las arquitecturas propuestas. En (Castán et al., 2014) y (Pikrakis et al., 2008) se aplica un módulo de post-procesamiento de resultados para refinar la respuesta de la segmentación. Además del módulo de post-procesamiento, Pikrakis et al. propone una segmentación en tres etapas. La primera realiza una segmentación inicial dejando secciones sin segmentar, la segunda se encarga de clasificar las secciones sin segmentar y la última etapa refina los resultados. La arquitectura descrita en (Zahid et al., 2015) es similar a la de Pikrakis et al., con la diferencia de que este trabajo no utiliza el módulo de post-procesamiento.

Además en la literatura se identificaron trabajos como (Giannakopoulos, 2015) y (Gallardo-Antolín & Hernández, 2010), en los cuales se describen herramientas de software para ejecutar el proceso de segmentación. Giannakopoulos presenta la librería de código abierto PyAudioAnalysis[2] la cual, además de la segmentación, ofrece varias funcionalidades para el análisis de audio. Esta herramienta está a disposición de la comunidad científica junto con los modelos de segmentación evaluados en su trabajo. La propuesta de Gallardo-Antolín describe un sistema de segmentación, llamado UPM-UC3M, el cual utiliza herramientas de entrenamiento de modelos HMM junto con la herramienta OpenSMILE[3] para la extracción de características.

De la revisión realizada sobre el área de segmentación automática de audio se puede concluir que es muy difícil encontrar un sistema de segmentación que satisfaga las necesidades específicas de identificar las secciones de narrativa de un audio y que ponga a disposición de la comunidad científica una herramienta de software. Esto debido a que, la mayoría de autores no ofrecen detalles sobre el software aplicado para la construcción de sus sistemas y por lo general solo utilizan una técnica de segmentación o se enfocan en presentar los aspectos teóricos de sus sistemas. De todas las propuestas analizadas, el trabajo presentado en (Giannakopoulos, 2015) es el único que presenta una herramienta

[2] PyAydioAnalysis, https://github.com/tyiannak/pyAudioAnalysis
[3] OpenSMILE, http://audeering.com/technology/opensmile/

de segmentación junto con los modelos evaluados. Además la herramienta ofrece la posibilidad de aplicar dos técnicas o enfoques de segmentación diferentes.

3.2. Reconocimiento Automático del Habla.

Varias aplicaciones de reconocimiento automático del habla han sido identificadas en la literatura relacionadas con el objetivo de este trabajo. Como ejemplo, en (Stüker, 2007), el autor emplea RAH para generar transcripciones de discursos del Parlamento Europeo de manera automática, sin embargo en la propuesta no se da detalles de las herramientas de software aplicadas. En (Guinaudeau et al., 2010) se propone emplear RAH para la identificación de tópicos abordados en videos, pero el autor pone mayor énfasis en la explicación de los conceptos y no ofrece detalles sobre las herramientas usadas. En (Ranchal, 2013) el objetivo es crear un sistema que ayude en la toma de notas de clases, para esto el autor propone el uso de herramientas propietarias como motor de la función de reconocimiento. Adicionalmente, en (Schneider et al., 2012) se propone un sistema que permite hacer recomendaciones de escenas de video en redes sociales por medio de la selección de citas o frases mencionadas en secciones del video las cuales son transcritas aplicando RAH.

Un aspecto fundamental a considerar en el RAH, es el idioma a reconocer. En (Nicalescu & de Jong, 2008) y (Varela et al., 2003), se construyen sistemas de reconocimiento de habla que coinciden con el idioma objetivo de este trabajo, sin embargo en estos trabajos se utilizan RAH poco difundidos en la comunidad o versiones antiguas de algún RAH de acceso libre. Además, un elemento importante a tener en cuenta para construir un RAH es el corpus que utiliza para la creación de los modelos acústicos. Existen varios trabajos como (Gretter, 2014), (Gravier et al., 2012), (Rodríguez-Fuentes et al., 2012), (Rousseau et al., 2012) y (Panayotov et al., 2010), en los cuales se presenta varios corpus aplicables al RAH, pero solo los autores Gretter y Rodríguez presentan corpus aplicables al idioma español. El Consorcio de Datos Lingüísticos (LDC[4]) ofrece un amplio catálogo de corpus pero requiere una membresía pagada para poder acceder a estos datos.

Por otra parte, varias herramientas se han creado para ayudar a la implementación de funciones de reconocimiento del habla. Trabajos como (Gaida et al., 2014), (Yao et al., 2010) y (Morbini, 2013) hacen una revisión y evalúan algunas herramientas RAH de código abierto. En (Plátek & Jurcıcek, 2014) y (Alumäe, 2014) se presentan complementos para la creación de reconocedores basados en la nube o como servicios web.

La mayor parte de los trabajos revisados utilizan procesos similares al propuesto en este trabajo, pero son muy pocos los que analizan un reconocimiento del habla en idioma español latinoamericano. Además, para la construcción de un RAH solo se han identificado los corpus presentados en (Pineda et al., 2004) y (Hernández-Mena & Herrera-Camacho, 2014) como aplicables a la implementación del reconocedor propuesto. Hay que destacar que propuestas como las presentadas en (Gaida et al., 2014) y (Alumäe, 2014) pueden servir como punto de partida para la construcción de herramientas de RAH.

[4] LDC, https://www.ldc.upenn.edu/

4. SERVICIO DE SEGMENTACIÓN Y TRANSCRIPCIÓN DE AUDIO.

En esta sección se presenta la arquitectura implementada para soportar los servicios de segmentación y transcripción de audio, los cuales ejecutan el proceso de extracción del habla a partir de una señal de audio (ver elemento central en la Figura 1). El proceso de segmentación y transcripción de audio recibe como entrada una señal de audio en un formato determinado e identifica las secciones del audio que son narrativa limpia, es decir diálogos de personas sin ningún otro tipo de sonido. El siguiente paso en el proceso, es reconocer las secciones del audio identificadas como narrativa y obtener un texto que contenga las transcripciones de dichas secciones. La Figura 4 ilustra los módulos propuestos para la extracción del habla de un audio.

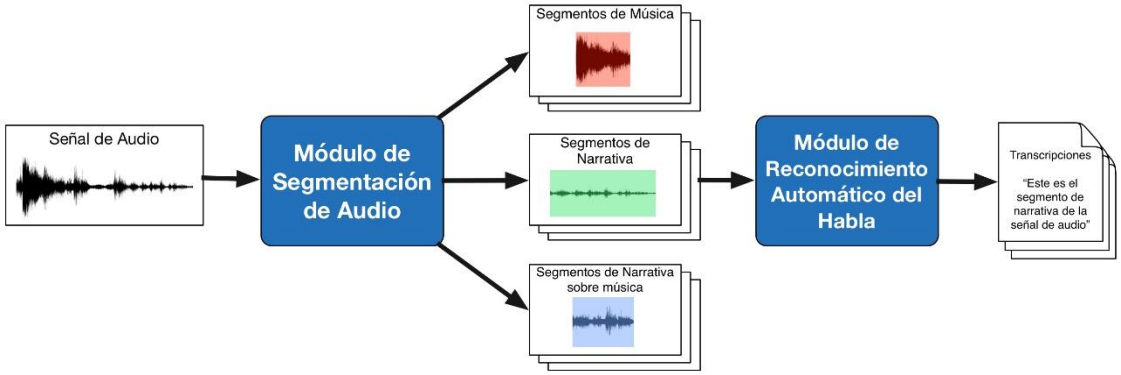


Figura 4. Proceso de análisis de una señal de audio para la extracción de habla como texto.

Para realizar la implementación del proceso ilustrado en la Figura 4, se ha diseñado la arquitectura mostrada en la Figura 5, en la cual se reconocen los siguientes componentes:

- ***Servicio de Segmentación Automática del Audio:*** El objetivo de este servicio es determinar los segmentos de tipo *música*, *narrativa o narrativa sobre música*, obteniendo para cada segmento su inicio y duración, de manera que sea posible su extracción del audio original
- ***Servicio de Transformación de Habla a Texto:*** Este servicio ejecuta un algoritmo de reconocimiento del habla sobre los segmentos obtenidos en el servicio de segmentación. El resultado final de este análisis es un texto con las transcripciones de las secciones que contengan narrativa de una señal de audio.
- ***Interfaz Web:*** Permite probar la funcionalidad de los servicios de segmentación y transcripción de audio, y los resultados obtenidos en cada parte del procesamiento.

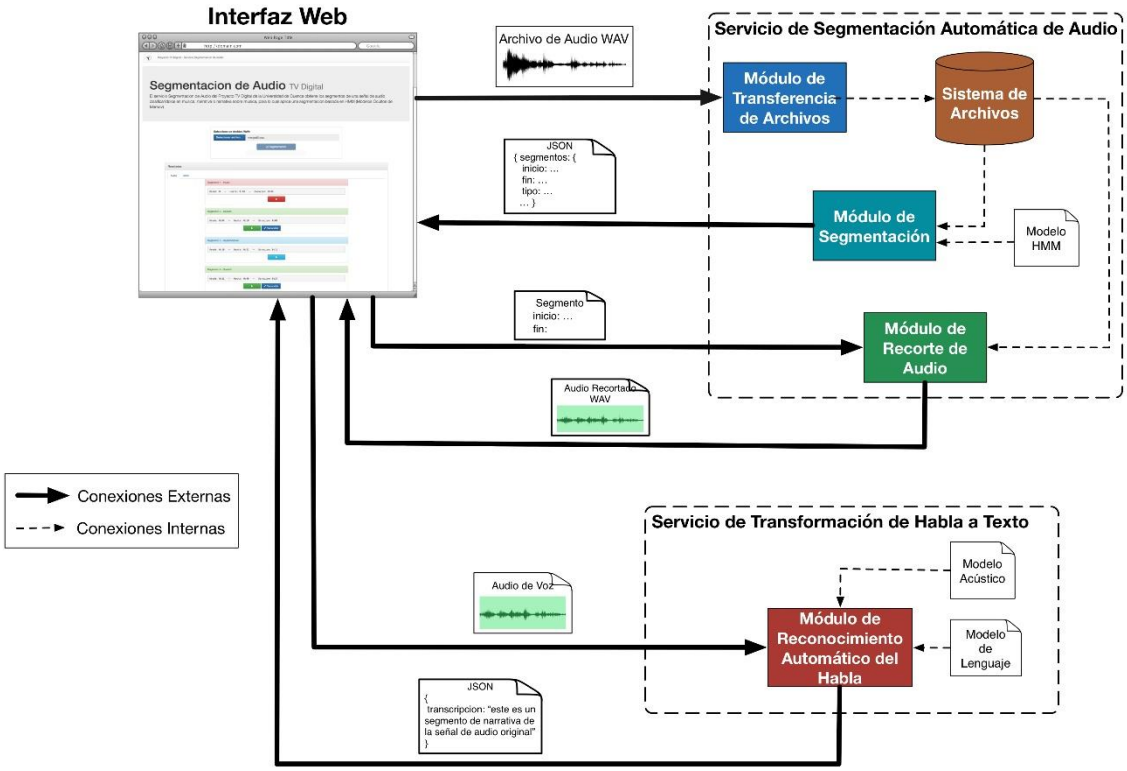


Figura 5. Arquitectura del Servicio de Segmentación y Transcripción de Audio

Las herramientas y algoritmos empleados en los servicios que conforman la arquitectura fueron evaluados previo a la implementación. Los resultados y la metodología utilizada en dichas evaluaciones se presentan en la sección 5.

A continuación se ofrecen más detalles de los componentes de la arquitectura.

4.1. Servicio de Segmentación Automática de Audio.

El servicio de segmentación automática de audio recibe como entrada un archivo de audio en formato WAV, el cual es analizado con el fin de extraer las partes de audio que contengan narrativa. Este servicio se implementó como un servicio Web REST[5] y se encarga de procesar y responder solicitudes utilizando el protocolo HTTP[6]. Para el desarrollo de los módulos, que conforman el servicio de segmentación, se utilizó el lenguaje de programación Python, esto debido a que, la librería PyAudioAnalysis, la cual es el núcleo del servicio, está escrita en dicho lenguaje de programación. Además, se utilizó el framework para aplicaciones Web Tornado[7], el cual también está escrito en Python y provee varias funcionalidades para la construcción de servidores web. Este servicio, consta de los tres módulos descritos a continuación:

[5] Siglas en ingles de Representational State Transfer es una arquitectura que proporciona una API para realizar operaciones entre un cliente y un servidor.
[6] Siglas en inglés de Hypertext Transfer Protocol.
[7] Tornado, http://www.tornadoweb.org/en/stable/

Módulo de Transferencia de Archivos.

La tarea de este módulo es transferir los archivos de audio provistos por el usuario, a un sistema de archivos local para que puedan ser procesados por los otros módulos. Así, el módulo se encarga de recibir un archivo de audio en formato WAV y copiarlo a un directorio de archivos en el servidor. El archivo debe ser enviado al servidor utilizando el método HTTP POST. Entonces, el módulo de transferencia de archivos recibe la petición y accede a su contenido para obtener el archivo, luego genera un identificador aleatorio de 6 alfanuméricos y guarda el archivo en el servidor. Como resultado, este módulo proporciona la ruta del directorio y nombre en donde se guardó el archivo.

Módulo de Segmentación de Audio.

Este módulo determina las partes del archivo de audio que son música, narrativa o narrativa sobre música y envía esta información como texto en formato JSON[8] (*JavaScript Object Notation*), a una interfaz web para presentar el resultado de la segmentación de una forma apropiada al usuario. El texto JSON describe las posiciones de la colección en las que inician y terminan segmentos de un solo tipo, como se observa en la Figura 6. Para la implementación del módulo de segmentación de audio se utilizó la librería PyAudioAnalysis.

Además, para poder ejecutar el proceso de segmentación se propuso un modelo HMM utilizando como tipos acústicos identificables en la segmentación: música, narrativa sobre música, y solo narrativa. Los detalles sobre el proceso de entrenamiento del modelo se describen en la sección 5.1.


```
{
  "segments": [
    {
      "start": 0,
      "end": 4,
      "number": 0,
      "label": "music"
    },
    {
      "start": 4,
      "end": 10,
      "number": 1,
      "label": "speech"
    },
    {
      "start": 10,
      "end": 21,
      "number": 2,
      "label": "s-m"
    },
  "id": "kimzbt"
}
```


Figura 6. Texto JSON resultante del proceso de segmentación.

[8] JSON es un formato de texto ligero para el intercambio de datos.

Módulo de Recorte de Audio.

Una vez conocidos los segmentos de cada tipo contenidos en el archivo de audio, es necesario obtener dichos segmentos como archivos de audio independientes. El módulo de recorte de audio recibe el tiempo inicial y la duración del segmento que se desea extraer del audio original, realiza un recorte de este archivo y envía el archivo resultante al usuario.

Para poder recortar los archivos WAV se desarrolló un script que recibe como entrada el segundo de inicio y la duración del segmento. Para implementar esta función se utilizaron herramientas de manipulación de archivos WAV de la librería SciPy[9]. El resultado del módulo de recorte es un archivo WAV del segmento que se desea extraer del audio original. Este archivo se guarda en un directorio en el sistema de archivos del servidor.

4.2. Servicio de Transformación de Habla a Texto.

El servicio de Transformación de Habla a Texto aplica RAH para transcribir un archivo de audio que contiene voz. Este servicio está conformado por un único módulo, denominado Módulo de Reconocimiento Automático del Habla, el cual recibe un archivo de audio en formato WAV y lo analiza para obtener su transcripción. Para realizar el reconocimiento es necesario disponer de un modelo acústico y un modelo de lenguaje (ver sección 2.2). El procedimiento aplicado para la creación y evaluación de estos modelos se presenta en la sección 5.2. La implementación del módulo central que ejecuta este servicio, se describe a continuación.

Módulo de Reconocimiento Automático del Habla.

Este módulo recibe como insumo de entrada un archivo de audio en formato WAV, el cual es procesado y da como resultado un texto en formato JSON con la transcripción de dicho audio. Para implementar este módulo se utilizó el servidor de reconocimiento del habla kaldi-gstreamer-server[10], el cual ofrece varias funcionalidades tales como: i) reconocimiento en tiempo real, ii) acceso a diferentes protocolos de comunicación cliente-servidor basados en peticiones HTTP y Web Sockets[11] y iii) características de escalabilidad para el realizar reconocimientos concurrentes (Green et al., 2015). El servidor kaldi-gstreamer-server se basa en el reconocedor Kaldi[12], y utiliza el framework multimedia GStreamer[13] para manipular los archivos de audio.

El proceso para realizar la implementación del servidor se encuentra en el sitio web de kaldi-gstremer-server. Hay que mencionar que, para realizar un reconocimiento es necesario escribir archivos de configuración para la ejecución de trabajadores (*workers*). Estos trabajadores son los que realmente llevan a cabo el reconocimiento y son controlados por el servidor.

[9] SciPy, https://www.scipy.org
[10] Kaldi GStreamer server, https://github.com/alumae/kaldi-gstreamer-server
[11] Web Socket es una tecnología que proporciona un canal de comunicación bidireccional y full dúplex entre un navegador y un servidor.
[12] Kaldi ASR, http://kaldi-asr.org
[13] Gstreamer, https://gstreamer.freedesktop.org

Para el desarrollo del Servicio de Transformación de Habla a Texto se utilizó el protocolo cliente-servidor basado en Web Sockets, lo cual permite realizar un reconocimiento en tiempo real, debido a que ofrece transcripciones parciales a medida que se vaya procesando el archivo de audio de entrada. Por este motivo, es necesario dividir el audio en bloques binarios que son enviados con una tasa de cuatro bloques por segundo.

4.3. Interfaz Web.

La interfaz web se desarrolló con el objetivo de mostrar el proceso de análisis de una señal de audio (ilustrado en la Figura 4), demostrando, al mismo tiempo, la funcionalidad de los servicios de segmentación y transcripción de audio, y los resultados obtenidos en cada parte del procesamiento. Para la construcción de esta interfaz se utilizó el lenguaje de programación JavaScript junto con el framework para creación de aplicaciones web AngularJS[14].

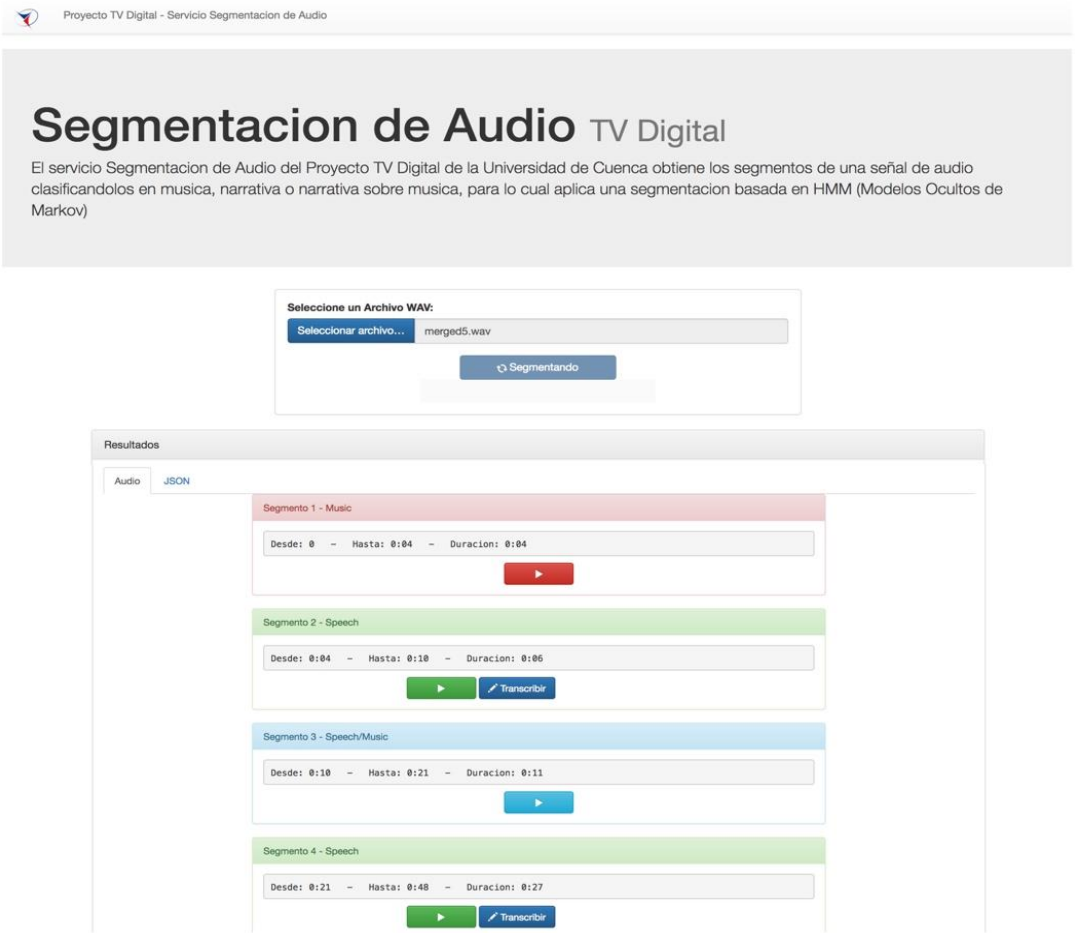


Figura 7. Captura de la interfaz Web desarrollada para consumir el Servicio de Segmentación de Audio.

[14] AngularJS, https://angularjs.org

Para utilizar el servicio de segmentación el usuario debe proveer un archivo de audio en formato WAV. El resultado de este proceso es un conjunto de segmentos presentados en paneles, como se observa en la Figura 7. Luego, el usuario puede seleccionar un segmento específico y escuchar su contenido, para lo cual la interfaz hace una petición al módulo de recorte de audio, obtiene el archivo recortado referente al segmento que seleccionó y lo reproduce. En caso de tratarse de un segmento de tipo narrativa, la interfaz presenta un botón para transcribir el segmento (ver botón Transcribir en la Figura 7).

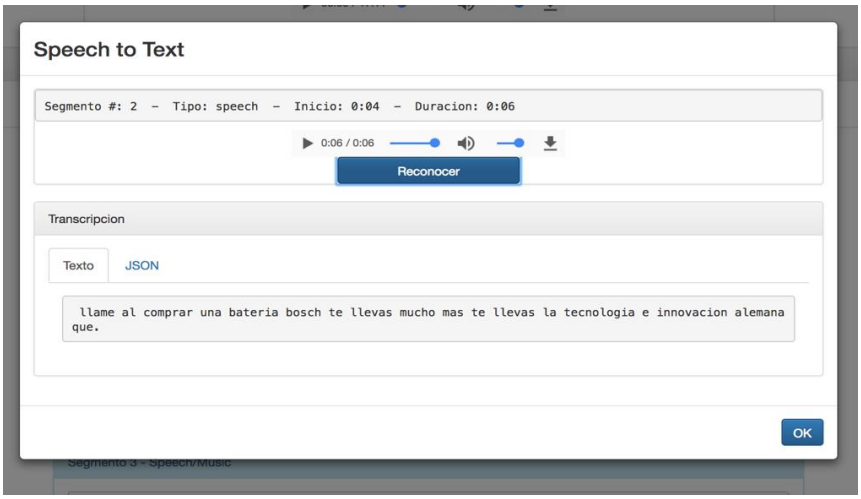


Figura 8. Captura de la interfaz web desarrollada para consumir el servicio de Transformación de Habla a Texto.

Al hacer clic en el botón Transcribir, se abre una nueva ventana, la cual presenta la información de ese segmento junto con un botón que inicia el reconocimiento, como se observa en la Figura 8. Si el usuario hace clic en el botón Reconocer, la interfaz inicia el reconocimiento utilizando el servicio Web de Transformación de Habla a Texto y presenta el resultado del reconocimiento al usuario como un texto en formato plano o JSON.

5. EVALUACIÓN.

En esta sección se describen los experimentos llevados a cabo para i) identificar las herramientas más adecuadas para implementar los servicios de segmentación de audio y transformación del habla a texto, y ii) evaluar la precisión obtenida por el servicio completo de segmentación y transcripción de audio usando las herramientas seleccionadas.

5.1. Evaluación de herramientas para la segmentación y transcripción de audio a texto.

La Figura 9 ilustra el proceso de evaluación aplicado para la selección de las herramientas utilizadas en la implementación del: servicio de segmentación de audio (ver sección 5.1.1) y el servicio de transcripción del habla a texto (ver sección 5.1.2).

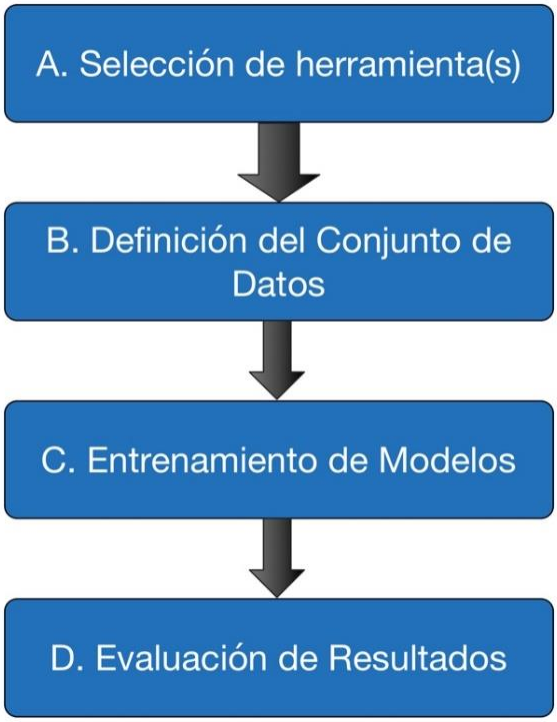


Figura 9. Flujo del proceso de evaluación aplicado.

Como primer paso se identificaron herramientas que permitan cumplir las funcionalidades de los servicios de segmentación y transcripción de audio a texto. Para ello se consideró herramientas de código abierto o que permitan ser configuradas de acuerdo a las necesidades. El segundo paso fue la definición de un conjunto de datos que permita evaluar la precisión de la herramienta en cuestión. La siguiente parte consistió en el entrenamiento de los modelos requeridos aplicando las funcionalidades de cada herramienta seleccionada sobre el conjunto de datos definido. Como paso final se aplicaron los modelos entrenados para realizar las tareas de segmentación o transcripción, según el caso; usando para ello una parte del conjunto total de datos. Esto con el objetivo de medir el rendimiento de las herramientas evaluadas aplicando alguna métrica o métricas.

5.1.1. Evaluación de herramientas de Segmentación de Audio.

En este punto se describe la evaluación de las herramientas de segmentación de audio tomando como referencia el flujo de proceso ilustrado en la Figura 9.

A. Selección de Herramientas.

Como primera parte de la evaluación se buscaron herramientas que permitan realizar el proceso automático de segmentación de audio. Como se mencionó en la sección 3.1, la única herramienta disponible fue la librería PyAudioAnalysis presentada en (Giannakopoulos, 2015). Esta librería ofrece dos funciones de segmentación basadas en dos enfoques respectivamente: enfoque de segmentación y clasificación y enfoque de segmentación por clasificación.

Esta librería ha sido creada para su uso en el análisis de señales de audio debido a que ofrece una amplia gama de funcionalidades junto con un diseño de programación fácil de usar y completo. Además, está escrita en el lenguaje de programación Python, el cual es el lenguaje de programación utilizado en el entorno de desarrollo seleccionado para la creación del servicio presentado en la sección anterior.

Para el presente análisis se evaluaron las dos funcionalidades de segmentación de audio provistas por la librería:

- Segmentación de tamaño fijo, que se basa en el enfoque de segmentación y clasificación.
- Segmentación basada en modelos HMM, que aplica el enfoque de segmentación por clasificación.

Estas dos funcionalidades permiten realizar una segmentación de audio supervisada, es decir, permiten entrenar los modelos requeridos que luego serán utilizados para segmentar un conjunto de datos de prueba.

B. Conjunto de Datos.

Actualmente, el Departamento de Ciencias de la Computación de la Universidad de Cuenca tiene registrado un conjunto de datos para el desarrollo de procesos de monitoreo de medios audiovisuales. De este conjunto de datos se seleccionó una muestra de audios como conjunto de datos para la evaluación de la librería PyAudioAnalysis.

Características del Conjunto de Datos para Segmentación	
Lenguaje:	Español Latinoamericano
País:	Ecuador
Tipo de grabación:	Capturado del *streaming* disponible en Internet de cada emisora
Contenido:	Programación Radial (Locuciones, Música, Publicidad)
Duración:	Aprox. 33 horas
Frecuencia de muestreo:	44.1 kHz
Canales de audio:	2
Formato del audio:	WAV
Ambiente de grabación:	Transmisión radial
Cantidad:	42 archivos con duraciones en su mayoría de 1 hora
Otros comentarios:	Contenido radial de 4 emisoras del Ecuador

Tabla 2. Resumen de las características del conjunto de datos usado en la evaluación de los algoritmos de segmentación.

Los archivos seleccionados contienen programación radial de 4 emisoras, con una duración de 33 horas aproximadamente. Esta programación comprende contenidos de locuciones, publicidad y música. La Tabla 2 resume algunas características de los archivos utilizados para evaluar los algoritmos de segmentación.

A diferencia de la segmentación binaria (narrativa con música) usada en (Giannakopoulos, 2015), en este trabajo se definieron tres tipos acústicos para la segmentación: solo música, narrativa y música y solo narrativa. Para ello se construyó un conjunto de datos etiquetando manualmente los archivos de audio. Se revisó el contenido del audio, utilizando una herramienta de edición para visualizar la forma de onda de cada

archivo de audio, de manera que se puedan observar los cambios significativos en la señal y de esta forma, acelerar el proceso de etiquetado. Además, se redujo los canales de audio de los archivos WAV del conjunto de datos, puesto que la librería PyAudioAnalysis solo procesa archivos con un canal de audio. Para este procesamiento se utilizó la herramienta de manipulación de audio SoX[15].

En la Tabla 3 se muestra cómo se distribuyen los archivos luego del etiquetado manual de los segmentos. Como se puede ver, la mayor parte del contenido radial es de tipo música, mientras que los contenidos de tipo solo narrativa y narrativa sobre música tienen una duración similar. Finalmente, para la evaluación se separó el conjunto de datos como sigue: aproximadamente 25 horas para entrenamiento y 7 horas y 30 minutos para pruebas. Hay que destacar que la cantidad de audio usada en este análisis es tres veces mayor a la cantidad utilizada en (Pikrakis, 2008), el cual se toma como referencia en (Giannakopoulos, 2015) para el desarrollo de pruebas.

Radios	Música (mins)	Narrativa (mins)	Narrativa/música (mins)	Totales (mins)
Radio A	568,2	33,5	75,6	677,4
Radio B	226,8	5,6	55,9	288,4
Radio C	342,8	272,3	85,0	700,0
Radio D	371,0	8,8	94,7	474,5
Totales	1508,8	320,2	311,3	2140,3

Tabla 3. Cantidad de minutos por radio y por tipo acústico del conjunto de datos utilizados.

C. Entrenamiento de Modelos.

PyAudioAnalysis ofrece dos funcionalidades de segmentación: i) segmentación basada en tamaño fijo y ii) segmentación basada en HMM. Por este motivo, fue necesario desarrollar un proceso de entrenamiento para cada una de estas funcionalidades. A continuación se describen algunos aspectos referentes a cada uno de los entrenamientos realizados.

i. Entrenamiento de Modelos para segmentación basada en tamaño fijo.

Este enfoque de segmentación requiere la aplicación de clasificadores para etiquetar los segmentos identificados como un tipo u otro. PyAudioAnalysis ofrece herramientas para el entrenamiento de modelos clasificadores basados en algoritmos de SVM (*Support Vector Machine*) o algoritmos KNN (*k-Nearest Neighbors*).

Para poder entrenar estos modelos supervisados es necesario agrupar los audios en diferentes directorios dependiendo del tipo acústico de los archivos (narrativa, narrativa sobre música, y música). Dado que, el conjunto de datos utilizados pertenece a audios que son grabaciones radiales, los cuales pueden contener los tipos acústicos descritos previamente en un solo archivo, fue necesario recortar los audios dependiendo de los segmentos etiquetados en la construcción del conjunto de datos. Para realizar este pre-procesamiento se desarrollaron varios scripts con la herramienta SoX.

[15] SoX - Sound Exchange, http://sox.sourceforge.net

Una vez establecidos los directorios con los segmentos de los audios de prueba para cada tipo, se procedió a utilizarlos para entrenar los modelos KNN y SVM. Para entrenar estos modelos supervisados, se utilizó la función trainClassifier de PyAudioAnalysis, la cual recibe los siguientes parámetros de entrada:

- Las rutas de los directorios que contiene los audios de entrenamiento de cada tipo.
- El método que se aplicará para entrenar el modelo (SVM o KNN).
- La ruta del archivo resultante al aplicar esta función.

El resultado de este proceso fue un conjunto de archivos que contienen los valores de las definiciones de los modelos clasificadores entrenados.

ii. Entrenamiento de Modelos para segmentación basada en HMM.

La librería PyAudioAnalysis ofrece funcionalidades para la construcción de modelos HMM a partir de archivos de audio etiquetados manualmente. Para esto, el usuario debe proveer, a más de los archivos de audio, un conjunto de datos que contenga el punto de inicio, el punto final y la etiqueta de cada segmento.

Para realizar el entrenamiento de los modelos HMM para la segmentación automática, se utilizó la funcionalidad trainHMMsegmenter_fromdir, la cual recibe los siguientes parámetros de entrada:

- Ruta del directorio que contiene los archivos de audio junto con los archivos de etiquetado de cada segmento.
- Ruta del Directorio en el que se crearán los archivos de descripción del modelo HMM.
- Tamaños para cada marco que se aplicará en la extracción de características.

El archivo resultante del proceso de entrenamiento contendrá los valores de transición y matrices que definen un modelo HMM de segmentación de audio.

D. Evaluación de Resultados.

La fase de evaluación de resultados comprende la realización de un proceso de segmentación, sobre una parte del conjunto de datos, aplicando los modelos entrenados en la fase anterior. Esto, con la finalidad de obtener resultados de la segmentación, los cuales serán analizados aplicando alguna métrica.

Métricas de Evaluación.

La métrica utilizada en esta evaluación es el promedio de porcentaje de precisión de la segmentación. Para calcular esta medida se utilizó la herramienta segmentationEvaluation de la librería de PyAudioAnalysis, que ejecuta una segmentación sobre un conjunto de audios y calcula un valor de precisión obtenida por cada audio segmentado. El cálculo de la precisión de la segmentación se realiza aplicando la siguiente formula:

$$\%\ de\ precisión\ en\ la\ Segmentacion = \frac{tiempo\ del\ audio\ segmentado\ correctamente}{duración\ de\ tiempo\ total\ del\ audio} * 100$$

Además, la herramienta de evaluación de la segmentación calcula medidas como promedio, mediana y valores máximos y mínimos de los valores de precisión obtenidos sobre un conjunto de archivos de prueba.

Resultados.

La Tabla 4 muestra el promedio del porcentaje precisión en el reconocimiento realizado sobre el conjunto de datos de prueba. En la tabla se puede observar que todos los valores estadísticos calculados demuestran que el algoritmo de segmentación basado en modelos HMM ofrece mayor precisión que los demás, lo que verifica lo dicho en (Giannakopoulos, 2015).

	Modelo KNN	Modelo SVM	Modelo HMM
Precisión Promedio	85,7	85,4	89,9
Precisión Mediana	84,4	84,8	89,6
Precisión Mínima	78,6	76,2	77,7
Precisión Máxima	98,8	99,0	100,0

Tabla 4. Comparativa de los porcentajes de precisión obtenidos por cada uno de los algoritmos de segmentación.

Como una evaluación adicional, se probaron tres configuraciones diferentes en los valores de parámetros de entrada del entrenamiento de los modelos HMM, con el objetivo de comprobar si existe una mejora en los valores de precisión obtenidos en la segmentación. La Tabla 5 muestra una comparativa de los resultados de precisión utilizando distintos valores para el tamaño de los marcos en los que se divide el archivo de audio en la fase de extracción de características. Los valores para los marcos que se han probado son 1, 0.5 y 0.1 segundos.

	Modelo HMM mw 1.0 - ms 1.0	Modelo HMM mw 0.5 - ms 0.5	Modelo HMM mw 0.1 - ms 0.1
Precisión Promedio	89,9	85,4	79,6
Precisión Mediana	89,6	90,0	88,7
Precisión Mínima	77,7	72,0	26,6
Precisión Máxima	100,0	95,2	95,2

Tabla 5. Comparativa de los porcentajes de precisión obtenidos por cada una de las configuraciones probadas sobre el algoritmos de segmentación basada en modelos HMM.

Como se muestra en la Tabla 5, reducir el tamaño de los marcos para la extracción de características disminuye el porcentaje de la precisión en la segmentación de audio. Esto se debe a que, de la señal de audio, se extrae una mayor cantidad de marcos, incrementando el número de vectores de características que se analizará, lo que a su vez, incrementa la probabilidad de que se etiquete incorrectamente uno de estos marcos. La Figura 10 muestra los gráficos resultantes al realizar una segmentación sobre un mismo archivo de audio aplicando valores de tamaño de marco de 1 (a), 0.5 (b) y 0.1 (c) segundos, lo cual ilustra lo mencionado anteriormente. Por esta razón, se ha establecido que el valor del tamaño de marco más óptimo para entrenar un modelo HMM es un segundo.

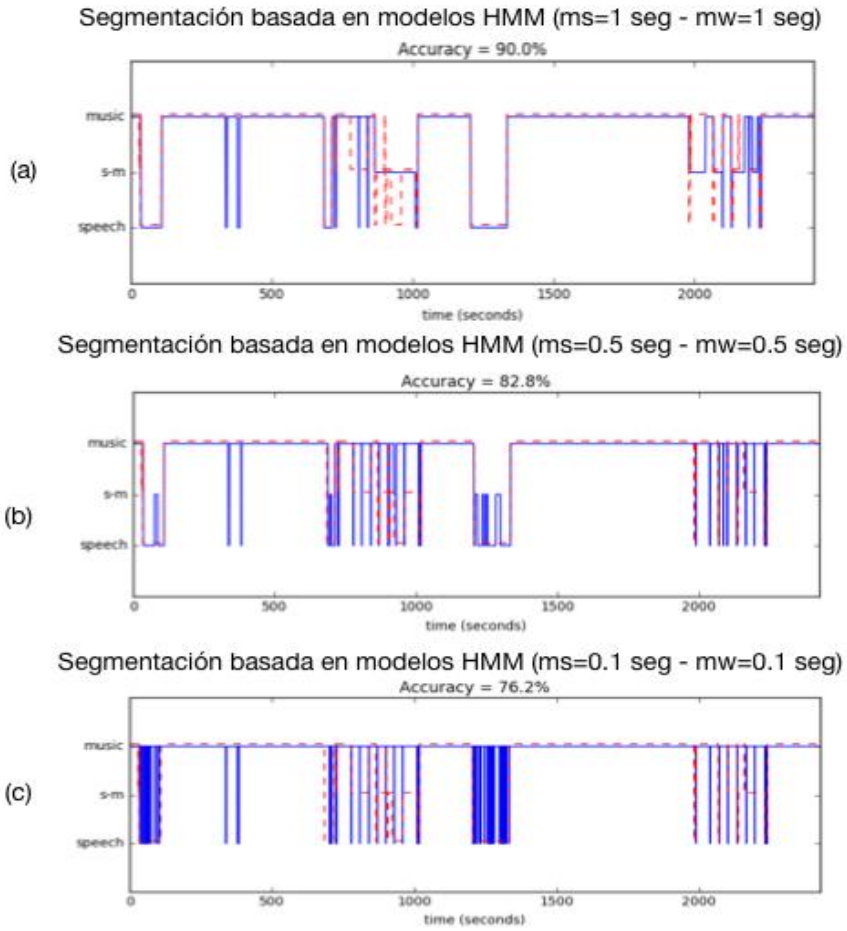


Figura 10. Gráficas obtenidas al aplicar un proceso de segmentación de audio sobre un mismo archivo del conjunto de datos con distintas configuraciones.

5.1.2. Evaluación de herramientas de transcripción del habla a texto.

El objetivo de esta evaluación fue comparar algunas herramientas RAH y seleccionar una de ellas para aplicarla en la definición de la arquitectura del servicio presentado en la sección 4.2.

A. Selección de herramientas.

Las herramientas RAH de código abierto que se evaluaron son las siguientes:

- **HTK (v3.4.1):** es un conjunto de herramientas para construir y manipular modelos HMM. HTK ha sido usada principalmente en el ámbito del reconocimiento del habla, pero puede utilizarse en otras aplicaciones como síntesis del habla, reconocimiento de caracteres, etc. (Young et al., 2006).
- **Kaldi:** es un conjunto de herramientas para el reconocimiento del habla desarrollado en C++. Su características principal es que se basa en Transductores de estado finito (FST – *Finite State Transducers*) (Povey et al., 2011).
- **CMU Sphinx-4:** es un framework modular escrito en Java, se utiliza para el desarrollo de sistemas de reconocimiento automático del habla basados en

modelos HMM (Walker et al., 2004). También existe una versión más liviana llamada pocketsphinx destinada para el desarrollo de aplicaciones para dispositivos móviles (Huggins-Daines et al., 2006).

A pesar de que existen otras herramientas RAH disponibles, la elección de estas herramientas se debió principalmente a: i) la alta difusión en la comunidad, ii) la facilidad de obtención y uso y iii) la comparación realizada en (Gaida et al., 2014), la cual evalúa las mismas herramientas aplicando un conjunto de datos de diálogos en idiomas inglés, alemán y japonés. El objetivo de este análisis es realizar una comparativa similar de dichas herramientas pero aplicándolas al reconocimiento del habla en idioma español latinoamericano, el cual es el idioma objetivo de este trabajo.

B. Conjunto de Datos.

DIMEX100 es un corpus lingüístico creado para ser aplicado en el desarrollo de tecnologías de lenguaje y proveer una base empírica para estudios fonéticos del idioma español mexicano (Pineda et al., 2004). El corpus DIMEX100 fue creado en el Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas de la UNAM. Los diálogos seleccionados para su construcción fueron obtenidos a partir de la conformación de un subconjunto de 5010 oraciones del Corpus230 (Villaseñor et al., 2001). Para la grabación de los audios se reclutaron a 100 personas, quienes grabaron 50 oraciones individuales y 10 comunes, resultando en un total de 6000 archivos de audio. Además de los archivos de audio y transcripciones, el corpus DIMEX100 ofrece los archivos necesarios para crear un reconocedor del habla utilizando la herramienta CMU Sphinx.

Nombre del Corpus:	**Dimex100**
Lenguaje:	Español Latinoamericano
País:	México
Tipo de grabación:	Grabación de Estudio
Contenido:	Subconjunto de 5010 oraciones del Corpus230
Duración de audio:	Aprox. 6 horas 10 minutos
Frecuencia de muestreo:	44.1 kHz
Formato del audio:	WAV - 16 bits
Ambiente de grabación:	Libre de ruido
Independencia de Hablante:	Si
Similitud:	Hablantes entre 16 y 36 años, nivel de educación secundario y superior, lugar de origen México. 49% hombres y 51% mujeres.
Granularidad de transcripción a fonemas:	3 Niveles basados en Mexbet - (T54 complejo, T44 intermedio, T22 simple)
Otros comentarios	Lexicón propio - 11477 palabras contando repeticiones

Tabla 6. Resumen de Características del corpus DIMEX100.

La Tabla 6 resume las características del corpus, como los niveles de granularidad de la transcripción a fonemas, los cuales se refieren al nivel de complejidad que se aplica para la descomposición de palabras al conformar el diccionario. Para la descomposición en fonemas se utilizó el alfabeto Mexbet[16] (Cuetara-Priede, 2004) que comprende un

[16] Mexbet es un alfabeto fonético computacional especializado en el español de México. Fuente: http://turing.iimas.unam.mx/~luis/DIME/DIMEx100/manualdimex100/mexbet.html

conjunto de 37 alófonos[17]. Para realizar el entrenamiento y pruebas de los modelos requeridos para el reconocimiento del habla se separó el corpus DIMEX100 así: audios de 80 locutores para entrenamiento y audios de 20 locutores para pruebas.

C. Entrenamiento de Modelos.

El entrenamiento de los modelos requeridos para la evaluación de las herramientas RAH comprende las fases de: i) extracción de características, ii) entrenamiento de modelos acústicos, iii) definición de un diccionario y iv) entrenamiento del modelo de lenguaje. A continuación se describen los detalles de cada una de estas fases.

i. Extracción de Características.

Para realizar la extracción de características en los tres reconocedores se ha tratado de establecer los mismos parámetros de configuración para todos por igual. Pero, debido a que cada herramienta ofrece sus propios scripts o comandos, no es posible asegurar que todas las configuraciones aplicadas sean exactamente iguales entre todos los reconocedores.

	CMU Sphinx	Kaldi	HTK
Formato:	MFCC	MFCC	MFCC
Sample Rate:	44.1 kHz	44.1 kHz	44.1 kHz
Window size (por defecto):	25.625 ms	25 ms	25 ms
Shift size (por defecto):	10 ms	10 ms	10 ms
Comando o Script utilizado:	make_feats.pl	make_mfcc.sh	HCopy

Tabla 7. Resumen de los parámetros básicos y herramientas aplicados para la extracción de características de cada reconocedor.

La Tabla 7 presenta un resumen de los parámetros aplicados en la extracción de características de cada reconocedor. Se puede observar que para ejecutar la extracción de características con el reconocedor HTK, se ha utilizado la herramienta HCopy con los parámetros de configuración basados en la secuencia de pasos para entrenamiento de modelos acústicos denominada Spanish_Voxforge_HTK_ASR[18], la cual se basa en el procedimiento de referencia creado en (Vertanen, 2006) para llevar a cabo experimentos de entrenamiento y pruebas de reconocimiento.

El reconocedor Kaldi ofrece varios scripts y ejemplos para cada una de las fases del reconocimiento. Para ejecutar la extracción de características se utilizó el script make_mfcc.sh.

La extracción de características con CMU Sphinx-4 se realizó mediante el script sphinxtrain el cual engloba los procesos de entrenamiento y pruebas como uno solo. Los parámetros de configuración aplicados para la extracción son los mismos que se han especificado en los tutoriales de entrenamiento disponibles en la página web de CMU Sphinx[19].

[17] Alófono es el sonido propio de la pronunciación de un fonema.
[18] Spanish voxforge htk asr, https://github.com/nassosoassos/spanish_voxforge_htk_asr
[19] CMU Sphinx, http://cmusphinx.sourceforge.net

ii. Modelos Acústicos.

Para entrenar el modelo acústico se utilizó el mismo conjunto de datos de entrenamiento previamente definido para todos los reconocedores, el cual consiste en casi 5 horas de audio contenido en 4800 archivos de audio. Como se mencionó con anterioridad, cada herramienta tiene su propio formato para especificar configuraciones, por lo que no se puede asegurar que los modelos acústicos han sido entrenados exactamente igual para cada reconocedor.

El procedimiento aplicado para el entrenamiento de los modelos acústicos en esta evaluación se puede resumir en los siguientes pasos:

a. **Entrenamiento de los modelos independientes de contexto (Modelos HMM de Monofonemas[20]):** en el primer paso del entrenamiento, se descompone cada una de las transcripciones en fonemas para cada audio. A partir de la descomposición de estos fonemas y, utilizando los vectores de características obtenidos del conjunto de entrenamiento en la fase de extracción de características, se procede a inicializar y entrenar unos modelos HMM para cada fonema. Además, se agregan modelos adicionales que representan silencios y cualquier otra expresión no vocal. El resultado de este proceso serán unos modelos HMM iniciales denominados Modelos de Monofonemas o Modelos Independientes de Contexto (Barrobés, 2012).
b. **Creación de alineaciones de entrenamiento:** utilizando los modelos construidos en el paso anterior, se efectúa un primer reconocimiento sobre los datos de entrenamiento. Esto con el fin de alinear las palabras de las transcripciones originales con los vectores de reconocimiento en cada archivo de audio.
c. **Entrenamiento de modelos dependientes de contexto (Modelos HMM de Trifonemas[21]):** El resultado del paso anterior son transcripciones a nivel de fonemas, las cuales son convertidas en transcripciones a nivel de trifonemas, lo que genera una lista de todos los trifonemas que aparecen en el conjunto de entrenamiento. A partir de estas nuevas transcripciones se procede a entrenar los modelos dependientes de contexto o denominados Modelos de Trifonemas (Barrobés, 2012). Estos modelos son el resultado final de la fase de entrenamiento.

A continuación se mencionan detalles específicos del entrenamiento realizado por cada herramienta.

- **Modelos Acústicos en HTK**: Para el entrenamiento en HTK se ha utilizado el script Spanish_Voxforge_HTK_ASR creado para entrenar un modelo acústico básico a partir de los audios disponibles en el sitio Voxforge[22]. Estos scripts se han modificado de manera que el entrenamiento se realice utilizando los datos que se definieron en el conjunto de datos.

[20] Monofonema es una denominación usada por los RAH para referirse a los fonemas individuales con el objetivo de diferenciar las fases del entrenamiento del modelo acústico.
[21] Trifonema es un grupo de tres fonemas que toman en cuenta un fonema central y el efecto coarticulatorio de los fonemas anterior y posterior.
[22] VoxForge, http://www.voxforge.org

- **Modelos Acústicos en Kaldi**: La herramienta Kaldi ofrece varios ejemplos para la construcción de reconocedores, por lo cual, para entrenar un modelo acústico se tomó como referencia el script del tutorial de la página web de Kaldi llamado "Kaldi for dummies tutorial"[23]. Este RAH aplica los transductores de estados finitos (FSTs) (Mohri et al., 2008), los cuales permiten codificar eficientemente todas las variedades de fuentes de conocimiento (modelos de lenguaje, diccionarios de pronunciación, árboles contextuales de decisión y topologías HMM) aplicadas en un sistema de reconocimiento. Esto quiere decir que, el reconocedor Kaldi se basa en modelos HMM codificados de manera más eficiente.
- **CMU Sphinx**: Este reconocedor ofrece un conjunto de herramientas llamado sphinxtrain, el cual reduce el proceso de entrenamiento a la ejecución de un comando en terminal, por lo que no es necesaria la creación de scripts que automaticen el proceso. Todos los parámetros necesarios para especificar una configuración pueden ser ajustados mediante un archivo de texto, para lo cual se requiere un nivel alto de conocimientos. En este caso se tomó como referencia la configuración sugerida en los tutoriales de la página web de CMU Sphinx.

iii. Diccionario.

El corpus DIMEX100 incluye tres archivos de diccionario diferentes. Cada uno de estos archivos corresponde a cada nivel de granularidad para transcripción en fonemas. El nivel de granularidad se refiere al número de fonemas que conforman el alfabeto aplicado en la descomposición de palabras. En las evaluaciones realizadas en (Pineda et al., 2004), utilizando el reconocedor Sphinx, se determinó que no existe una gran diferencia en los resultados obtenidos al aplicar un nivel de granularidad u otro. Motivo por el cual, para esta evaluación se utilizó el nivel T22 que corresponde al alfabeto fonético más simple de 17 consonantes y 5 vocales del idioma español mexicano (Pineda et al., 2004). El diccionario de DIMEX100 aplicado contiene un total de 11477 palabras contando transcripciones de pronunciaciones diferentes para una misma palabra.

iv. Modelo de Lenguaje.

Aunque algunos de los reconocedores evaluados ofrecen herramientas propias para la construcción de modelos de lenguaje y gramáticas aplicables al reconocimiento del habla (Young et al., 2006; Walker et al., 2004). En esta evaluación se construyó un modelo de lenguaje común para los tres reconocedores usando la herramienta SRILM (v.1.7.1). SRILM es un conjunto de herramientas de software de código abierto para el modelado estadístico de lenguajes y otras tareas relacionadas (Stolcke et al., 2011). El formato de los modelos de lenguaje usados es el formato ARPA 3-gram (Stolcke, 2002; Jurafsky & Martin, 2014) ya que, todos los reconocedores son compatibles con este formato.

En el caso del reconocedor Kaldi, fue necesario ejecutar unos cuantos scripts más para codificar el modelo de lenguaje ARPA en un archivo FST. Además, para la evaluación de los modelos acústicos se probaron dos experimentos, con el objetivo de conocer la influencia del modelo de lenguaje en el reconocimiento del habla.

[23] Kaldi for Dummies Tutorial, http://kaldi-asr.org/doc/kaldi_for_dummies.html

- **Experimento 1:** Se construyó el modelo de lenguaje a partir de las transcripciones correspondientes al conjunto de datos de entrenamiento solamente.
- **Experimento 2:** Se construyó un modelo de lenguaje a partir de la totalidad de las transcripciones del corpus DIMEX100.

De estos experimentos resultaron dos modelos de lenguaje, el primero, al que se le denominó Modelo de Lenguaje incompleto, se construyó sin incluir las oraciones que se reconocerán en la evaluación. El segundo modelo de lenguaje, llamado Modelo de Lenguaje Completo, contiene todas las oraciones que se pronunciaron en el reconocimiento de prueba.

D. Evaluación de Resultados.

En esta fase se ejecutó un proceso de reconocimiento con cada uno de los RAH sobre una parte del conjunto de datos. Los resultados obtenidos fueron evaluados aplicando dos métricas. A continuación se describen los detalles de esta fase.

Decodificación.

Una vez obtenidos todos los insumos necesarios para el reconocimiento, la última fase de la evaluación es realizar un reconocimiento sobre los datos de prueba, lo que quiere decir que, se procede a ejecutar un proceso de decodificación sobre los vectores de características correspondientes a los audios de prueba utilizando el modelo acústico y el modelo de lenguaje construidos en las fases previas.

Cada reconocedor ofrece funcionalidades propias para realizar la tarea de decodificación. A continuación se detallan aspectos específicos de cada herramienta.

- **Decodificación con HTK**: Para esta evaluación se utilizó el decodificador HDecode, debido al formato ARPA 3-gram del modelo de lenguaje y al tamaño del diccionario a ser utilizados en este análisis.
- **Decodificación con Kaldi:** Se utilizó el script denominado decode.sh, este script recibe como parámetros de entrada un archivo de configuración simple, las rutas de los directorios que contienen los modelos acústico y de lenguaje en formato FST y el directorio donde se crearán los archivos resultantes de la decodificación.
- **Decodificación con CMU Sphinx-4:** La herramienta sphinxtrain de CMU Sphinx, al finalizar el entrenamiento del modelo acústico, inicia automáticamente el proceso de decodificación del conjunto de pruebas. Para esto, se utilizó un script escrito en Perl, denominado psdecode.pl. En la evaluación realizada se utilizaron los parámetros de configuración para el decodificador sugeridos en los ejemplos de construcción de modelos acústicos de la página web oficial de CMU Sphinx.

Métricas de Evaluación.

La evaluación de un RAH se basa principalmente en dos métricas: Tasa de Error por palabras (WER), cuyo objetivo es medir la precisión del reconocimiento y Factor de Tiempo Real (RTF) que mide el tiempo que se requiere para ejecutar el reconocimiento (Anusuya & Katti, 2010).

- **Tasa de Error por Palabras (WER - Word Error Rate):** para medir la tasa de error por palabras se aplica la siguiente formula (Anusuya & Katti, 2010):

$$WER = \frac{S + D + I}{\#\ palabras\ en\ la\ transcripción\ referencia} * 100$$

  Donde:
  - S es el número de palabras sustituidas en la transcripción.
  - D es el número de palabras borradas u omitidas en el reconocimiento
  - I es el número de palabras insertadas que no pertenecen a la transcripción real.

  Para medir la tasa de error por palabra se debe tener transcripciones de referencia de los audios del conjunto de prueba.
  Como cada uno de los reconocedores evaluados ofrece sus propias herramientas para medir la precisión de los modelos entrenados, se ha efectuado la medición de la tasa de error por palabra utilizando cada una de las herramientas propias de cada reconocedor.
- **Factor de Tiempo Real (RTF – Real-Time Factor):** el factor de tiempo real mide la velocidad a la que se realiza el reconocimiento comparándola con la duración de la señal de voz reconocida (Platek, 2014). Para medir el factor de tiempo real se utiliza la siguiente formula:

$$RTF = \frac{tiempo\ de\ decodifación\ del\ audio\ A}{duración\ del\ audio\ A}$$

  Para medir el RTF se tomó la información presentada en archivos de log o durante la ejecución del proceso de decodificación de cada reconocedor.

Resultados.

Los resultados de las evaluaciones realizadas en el conjunto de datos de prueba se muestran en la Tabla 8. Como indican los valores obtenidos para las métricas WER y RTF, Kaldi es el reconocedor que ofrece mejor precisión y mejor eficiencia con respecto al tiempo, comparado con los otros reconocedores, motivo por el cual se seleccionó como motor RAH del servicio presentado en la sección 4. Aunque la diferencia en la precisión entre Kaldi y HTK es de aproximadamente 1%, hay que resaltar que Kaldi requiere una cantidad de tiempo cuatro veces menor a la requerida por HTK para realizar un reconocimiento. Esto puede ser debido a que, Kaldi utiliza una codificación más eficiente, basada en Transductores de Estado Finito, para representar los modelos acústicos y de lenguaje, lo que hace más eficiente la decodificación.

En general, los tres reconocedores evaluados ofrecen un rendimiento aceptable para efectuar una transcripción automática de habla a texto ya que, ofrecen una precisión mayor al 80% y realizan un reconocimiento en un tiempo menor a la mitad de la duración del audio reconocido.

Otro aspecto observable en la Tabla 8 es la disminución del WER en más del 15%, cuando se utilizó el modelo de lenguaje completo, lo que significa que, un modelo de lenguaje construido con los textos de las oraciones que se reconocerán aumenta la precisión del RAH, a pesar de que no se haya entrenado el modelo acústico con el audio de dichas oraciones.

Reconocedor		CMU Sphinx-4	Kaldi	HTK
WER	Modelo de Lenguaje incompleto	38,99%	29,13%	30.27%
	Modelo de Lenguaje Completo	18,80%	13,58%	14.76%
RTF Promedio		0,28	0,1079	0,425

Tabla 8. WER y RTF obtenidos al reconocer los audios del conjunto de pruebas del corpus DIMEX100.

5.2. Evaluación del Servicio de Transformación de Habla a Texto.

La evaluación realizada sobre el servicio de transformación de habla a texto consistió en medir la precisión obtenida al realizar un reconocimiento aplicando el servicio de transformación de habla a texto (ver sección 4.2) sobre un conjunto de datos de pruebas, y comparar los resultados con la precisión obtenida al transcribir el mismo conjunto de datos aplicando el servicio *Speech To Text* de IBM[24].

No se hizo una evaluación comparativa del servicio de segmentación automática de audio debido a que no se encuentra disponible un servicio que ofrezca resultados similares a los conseguidos con el servicio implementado en este trabajo. Además, el modelo HMM utilizado en su implementación ya fue evaluado en el punto 5.1.1.

A. Selección de herramientas.

Las herramientas evaluadas fueron:

- Servicio de Transformación de Habla a Texto (ver sección 4.2).
- Servicio Speech To Text de IBM: es un servicio de pagado de disposición general y convierte la voz de audio en texto escrito, ofrece varios SDKs[25] para ser utilizado en algunos lenguajes de programación como Java, Python, etc.

Para desarrollar la evaluación, se utilizaron los scripts desarrollados en la evaluación realizada en el punto 5.1.2 de este trabajo, los cuales se han aplicado sobre los modelos entrenados para implementar el servicio de Transformación de Habla Texto expuesto en la sección 4.2.

B. Conjunto de Datos.

El conjunto de datos usado para entrenar los modelos requeridos está conformado por el corpus DIMEX100 (Pineda et al., 2004), el corpus CIEMPIESS (Hernández-Mena & Herrera-Camacho, 2014) y un corpus creado a partir de las algunos de los segmentos identificados como narrativa del conjunto de datos utilizado en la evaluación realizada en el punto 5.1.1. Este conjunto de datos se formó con archivos de audio obtenidos de grabaciones de transmisiones radiales ecuatorianas, por lo cual se le denominó Radios Ecuador. Para extraer los segmentos de narrativa de estos audios se utilizó el mismo script

[24] IBM Speech to Text, https://speech-to-text-demo.mybluemix.net
[25] Siglas en inglés de Software Development Kit.

basado en SoX desarrollado en el punto 5.1.1 y luego fueron transcritos a texto manualmente.

La Tabla 9 muestra varias características de los tres corpus utilizados. En esta tabla se muestra que los archivos de audio que conforman los corpus DIMEX100 y Radios Ecuador tienen el mismo formato y frecuencia de muestreo, pero el corpus CIEMPIESS está conformado por archivos de tipo SPH y frecuencia de muestreo de 16 kHz. Por este motivo se redujo la frecuencia de muestreo de los archivos de los corpus DIMEX100 y Radios Ecuador, y además, se cambió el formato de los archivos de audio del corpus CIEMPIESS a WAV aplicando la herramienta SoX.

- **Corpus:**	**Dimex100**	**CIEMPIESS**	**Radios Ecuador**
Lenguaje:	Español Latinoamericano	Español Latinoamericano	Español Latinoamericano
País:	México	México	Ecuador
Tipo de grabación:	Grabación de Estudio	Grabaciones obtenidas de un Podcast	Archivos obtenidos de grabaciones de radio
Tipo de Contenido:	Subconjunto de 5010 oraciones del Corpus230	Programación Radial	Programación Radial
Duración de audio (Aproximado):	6 horas 10 minutos (6000 audios)	18 horas	1 hora 30 minutos
Frecuencia de muestreo:	44.1 kHz	16 kHz	44.1 kHz
Formato del audio:	WAV	SPH (PCM)	WAV
Ambiente de grabación:	Libre de ruido	Transmisión Radial	Transmisión Radial
Independencia de Hablante:	Si	Si	Si
Similitud:	Hablantes entre 16 y 36 años, nivel de educación secundario y superior, lugar de origen México	78% hablantes masculinos, 22% femeninos,	No determinado
Granularidad de transcripción a fonemas:	3 - (T54 complejo, T44 intermedio, T22 simple)	4 - (T50 complejo sin tildes y con tildes, T22 simple sin tildes y con tildes)	Ninguno
Otros comentarios:	Lexicón propio - 11477 palabras contando repeticiones	Lexicón propio 50000 palabras sin repetición	Ninguno

Tabla. 9. Resumen de las características de los tres corpus utilizados como conjunto de datos.

El total del tiempo de duración del conjunto de datos utilizado para la creación y evaluación de los modelos usados en el servicio de transformación de habla a texto, es de aproximadamente 25 horas con 40 minutos distribuido en más de 20000 archivos de audio. De este conjunto de datos fue necesario definir una parte como conjunto de entrenamiento y otra como conjunto de pruebas, esto con el objetivo de realizar evaluaciones posteriores. Por este motivo, se procedió a dividir cada corpus por separado quedando de la siguiente manera.

- **DIMEX100:** se utilizaron los conjuntos de entrenamiento y pruebas definidos en el punto 5.1.2.
- **CIEMPIESS:** en (Hernández-Mena & Herrera-Camacho, 2014), los creadores del corpus definieron conjuntos de prueba y entrenamiento, los cuales fueron utilizados en este trabajo.

- **Radios Ecuador:** se tomó aproximadamente una hora del corpus para entrenamiento y 20 minutos del corpus, para pruebas.

La duración total del conjunto de datos para evaluaciones es de 2 horas con 15 minutos aproximadamente. Este conjunto de datos contiene más de 2000 archivos de audio con formato WAV.

C. Entrenamiento de Modelos.

Esta fase comprende el proceso de entrenamiento de los modelos que conforman el servicio de transformación de habla a texto (ver Figura 5).

Modelo Acústico.

Para la creación del modelo acústico requerido, se utilizaron los scripts desarrollados en el punto 5.1.2. El resultado del proceso de entrenamiento son modelos HMM de trifonemas codificados en un archivo utilizando FST, el cual debe ser provisto como parámetro en el archivo de configuración al momento de inicializar un trabajador en el servidor kaldi-gst-streamer.

Modelo de Lenguaje.

El modelo de lenguaje se creó utilizando la herramienta SRLIM, a partir de un archivo de texto que contiene todas las oraciones que conforman el conjunto de datos definido para la creación de este servicio. El formato del modelo de lenguaje es ARPA 3-gram, el cual es el mismo que se usó en el punto 5.1.2. Este modelo de lenguaje se codificó en un archivo FST para que sea compatible con Kaldi y el servidor kaldi-gstreamer.

D. Evaluación de Resultados.

Los resultados evaluados son las transcripciones procedentes del reconocimiento sobre el conjunto de pruebas aplicando los dos servicios: i) Servicio de Transformación de Habla a Texto (ver sección 4.2) y ii) Speech to Text IBM. A continuación se describen las métricas y resultados de la evaluación.

Métricas de Evaluación.

En la evaluación realizada en esta sección se ha utilizado la métrica Tasa de Error por Palabra (WER) (ver sección 5.1.2). Para calcular el valor del WER se utilizó el script score.sh, incluido en los archivos de instalación de Kaldi.

Aparte, con la finalidad de consumir el servicio Speech To Text de IBM se desarrolló un script que utiliza la herramienta curl[26] para la transferencia de archivos de audio empleando el protocolo HTTP. Además, fue necesario crear una cuenta de IBM para poder generar credenciales de uso del servicio. Las transcripciones resultantes se obtienen en textos de formato JSON. Para medir la tasa de error por palabras en las transcripciones resultantes del servicio Speech to Text, se utilizó el script word_align.pl de la herramienta sphinxtrain incluida en el reconocedor CMU Sphinx. Este script requiere que los archivos de las transcripciones resultantes y de referencia tengan un

[26] Curl, https://curl.haxx.se

formato específico, por lo cual se ejecutó un procesamiento sobre los textos JSON resultantes del reconocedor de IBM.

Resultados.

La Tabla 10 presenta los resultados obtenidos al ejecutar un reconocimiento utilizando dos servicios sobre el mismo conjunto de datos. Se puede observar que, el servicio desarrollado en este trabajo ofrece un menor WER, aunque ambos servicios ofrecen una precisión aceptable. Hay que mencionar que, el reconocedor de IBM transparenta varios aspectos complejos como la generación de los modelos requeridos, pero no se tiene acceso a la personalización de la configuración del servicio para realizar tareas como por ejemplo la inclusión de palabras al vocabulario del reconocedor. Además, el modelo de lenguaje del reconocedor de IBM no fue entrenado con las transcripciones reales del conjunto de datos utilizado para evaluar los reconocedores, motivo por el cual se puede suponer que deben haber palabras fuera del vocabulario del reconocedor de IBM, las cuales influyen en la obtención de una menor precisión.

	Servicio de Transformación de Habla a Texto	**Servicio *Speech To Text* IBM**
WER	10.83%	16.49%

Tabla 10. Valores del WER obtenidos al realizar un reconocimiento aplicando el servicio desarrollado en este trabajo y aplicando el servicio de reconocimiento de IBM.

6. CONCLUSIONES

En este artículo se ha presentado una arquitectura para la ejecución de un proceso de extracción de información a partir de señales de audio mediante la aplicación de las tecnologías de segmentación de audio y RAH. Con el fin de medir el rendimiento de las herramientas utilizadas en el proceso de segmentación y transcripción de audio a texto, se definieron diferentes evaluaciones. El resultado de la evaluación de los algoritmos de segmentación utilizando la librería PyAudioAnalysis permitió conocer que, el enfoque de segmentación por clasificación aplicando modelos HMM ofrece una mayor precisión en sus resultados que el enfoque de segmentación y clasificación basada en distancias. Sin embargo, hay que señalar que no se pudo alcanzar el nivel de precisión obtenido en (Giannakopoulos, 2015), esto presumiblemente debido a que se segmentaron los audios en tres tipos y no solamente dos.

En lo que respecta a la evaluación de los RAH, se identificó que el reconocedor Kaldi ofrece un mejor rendimiento en cuanto a precisión y velocidad del reconocimiento que los demás. Los resultados obtenidos en esta evaluación verifican lo dicho en (Gaida et al., 2014), pero con la diferencia de que se ha aplicado el reconocimiento al idioma español latinoamericano y se ha incluido la medición del Factor de Tiempo Real. Como conclusión del proceso de evaluación, se pudo determinar que, se requiere bastante tiempo para comprender los conceptos básicos, ejecutar un reconocimiento y optimizar configuraciones de un RAH, por este motivo no se han explotado todas las funcionalidades que ofrecen las tres herramientas evaluadas y se ha optado por utilizar como referencia tutoriales básicos sugeridos por los propios creadores de cada reconocedor. Además, se conoció que, para obtener un nivel de precisión aceptable, el modelo de lenguaje debe ser construido a partir de las expresiones que se reconocerán.

Como parte final de este trabajo, se implementó la arquitectura propuesta como un conjunto de servicios web y, en el caso de los resultados del servicio de transcripción, se hizo una comparación con los resultados obtenidos al aplicar un servicio de disposición general desarrollado por un tercero. El resultado de esta comparación mostró que, aunque no es mucha la diferencia, el servicio propuesto en este artículo ofrece mejores resultados en lo que a precisión de las transcripciones se refiere.

Como futuras líneas de investigación que puedan abordarse para contribuir con el desarrollo de los temas propuestos en este trabajo se propone, realizar un análisis de los resultados que se obtienen al aplicar RAH sobre las secciones identificadas como narrativa sobre música por el servicio de segmentación. También, implementar un proceso de refinamiento de los resultados de segmentación aplicando alguna de las arquitecturas hibridas propuestas en (Fakotakis et al. 2014), con el fin de mejorar la precisión de la segmentación. La creación de varios corpus textuales orientados a la construcción de modelos de lenguaje específicos para cada tipo de contenido de televisión y radio establecidos en la Ley Orgánica de Comunicación.

AGRADECIMIENTOS

El trabajo presentado en este artículo es parte del proyecto de investigación “Empleo de tecnologías semánticas para el análisis de contenido multimedia transmitido para televisión digital terrestre” financiado por la Dirección de Investigación de la Universidad de Cuenca (DIUC).

REFERENCIAS

Abad, A., Meinedo, H., & Neto, J. (2008, September). Automatic classification and transcription of telephone speech in radio broadcast data. In *International Conference on Computational Processing of the Portuguese Language* (pp. 172-181). Springer Berlin Heidelberg.

Aron, J. (2011). How innovative is Apple's new voice assistant, Siri?. *New Scientist*, *212*(2836), pag 24.

Alumäe, T. (2014). Full-duplex Speech-to-text System for Estonian. In *Baltic HLT* (pp. 3-10).

Anusuya, M. A., & Katti, S. K. (2010). Speech recognition by machine, a review. *arXiv preprint arXiv:1001.2267*.

Bachu, R. G., Kopparthi, S., Adapa, B., & Barkana, B. D. (2008, June). Separation of voiced and unvoiced using zero crossing rate and energy of the speech signal. In *American Society for Engineering Education (ASEE) Zone Conference Proceedings* (pp. 1-7).

Barrobés, H. D., & Ruiz, M. (2012). Reconocimiento Automático del Habla. Universitat Oberta de Catalunya. Disponible en: https://www.exabyteinformatica.com/uoc/Audio/Procesamiento_de_audio/Procesamiento_de_audio_(Modulo_7).pdf,

Bietti, A., Bach, F., & Cont, A. (2015, April). An online EM algorithm in hidden (semi-) Markov models for audio segmentation and clustering. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on* (pp. 1881-1885). IEEE.

Campoverde Llanos, E. G. and Guerrero Fernández de Córdova, A. M. (2015). Aplicación de Tecnologías Semánticas y Técnicas de Reconocimiento de Objetos para la Identificación de Armas de Fuego en Video. Universidad de Cuenca. Disponible en: http://dspace.ucuenca.edu.ec/handle/123456789/22839

Castán, D., Ortega, A., Miguel, A., & Lleida, E. (2014). Audio segmentation-by-classification approach based on factor analysis in broadcast news domain. *EURASIP Journal on Audio, Speech, and Music Processing*, *2014*(1), 1-13.

Cuétara, J. A. V. I. E. R. (2004). *Fonética de la ciudad de México. Aportaciones desde las tecnologías del habla* (Doctoral dissertation, Tesis para obtener el título de Maestro en Lingüística Hispánica).

Fakotakis, N., Mporas, I., & Theodorou, T. (2014). An overview of automatic audio segmentation. *International Journal of Information Technology and Computer Science (IJITCS)*, *6*(11), 1.

Fujii, Y., Yamamoto, K., & Nakagawa, S. (2011, March). Large vocabulary speech recognition system: SPOJUS++. In *MUSP* (pp. 110-118).

Gaida, C., Lange, P., Petrick, R., Proba, P., Malatawy, A., & Suendermann-Oeft, D. (2014). Comparing open-source speech recognition toolkits. *Tech. Rep., DHBW Stuttgart*.

Gaikwad, S. K., Gawali, B. W., & Yannawar, P. (2010). A review on speech recognition technique. *International Journal of Computer Applications*, *10*(3), 16-24.

Gallardo, A., & San-Segundo, R. (2010). UPM-UC3M system for music and speech segmentation. *Jornadas de Tecnología del Habla FALA 2010*.

Giannakopoulos, T. (2015). pyaudioanalysis: An open-source python library for audio signal analysis. *PloS one*, *10*(12), e0144610.

Goffin, V., Allauzen, C., Bocchieri, E., Hakkani-Tur, D., Ljolje, A., Parthasarathy, S., ... & Saraclar, M. (2005, March). The AT&T Watson speech recognizer. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on* (Vol. 1, pp. I-1033). IEEE.

Gómez Rincón, E. (2015). Segmentación de audio mediante características cromáticas en ficheros de noticias (Bachelor's thesis). Universidad Autónoma de Madrid. Disponible en: http://hdl.handle.net/10486/668164

Gravier, G., Adda, G., Paulson, N., Carré, M., Giraudel, A., & Galibert, O. (2012). The ETAPE corpus for the evaluation of speech-based TV content processing in the French language. In *LREC-Eighth international conference on Language Resources and Evaluation* (p. na).

Green, P., Marxer, R., Cunningham, S., Christensen, H., Rudzicz, F., Yancheva, M., & Desideri, L. (2015, September). Remote speech technology for speech professionals-the CloudCAST initiative. In *6th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)* (p. 97).

Gretter, R. (2014, May). Euronews: a multilingual speech corpus for ASR. In *LREC* (pp. 2635-2638).

Guinaudeau, C., Gravier, G., & Sébillot, P. (2010, September). Improving ASR-based topic segmentation of TV programs with confidence measures and semantic relations. In *INTERSPEECH* (pp. 1365-1368).

Hernández-Mena, C. D., & Herrera-Camacho, J. A. (2014). CIEMPIESS: A New Open-Sourced Mexican Spanish Radio Corpus. In *LREC* (Vol. 14, pp. 371-375).

Huang, X., & Deng, L. (2010). An overview of modern speech recognition. In *Handbook of Natural Language Processing, Second Edition* (pp. 339-366). Chapman and Hall/CRC.

Huggins-Daines, D., Kumar, M., Chan, A., Black, A. W., Ravishankar, M., & Rudnicky, A. I. (2006, May). Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on* (Vol. 1, pp. I-I). IEEE.

Imai, T., Kobayashi, A., Sato, S., Homma, S., Onoe, K., & Kobayakawa, T. (2004). Speech recognition for subtitling Japanese live broadcasts. *Proc. ICA* Vol. 1, 165-168.

Ittichaichareon, C., Suksri, S., & Yingthawornsuk, T. (2012, July). Speech recognition using MFCC. In *International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012) July* (pp. 28-29).

Jurafsky, D., & Martin, J. H. (2014). N-Grams. En Jurafsky, D., & Martin, J. H. Speech and language processing (Vol. 3). Pearson.

Kłosowski, P., Dustor, A., Izydorczyk, J., Kotas, J., & Ślimok, J. (2014). Speech recognition based on open source speech processing software. *International Conference on Computer Networks*, 308-317, Springer International Publishing.

Kulkarni, A., Iyer, D., & Sridharan, S. R. (2001). Audio segmentation. In *IEEE, International Conference on Data Mining, ICDM* (pp. 105-110).

Mohri, M., Pereira, F., & Riley, M. (2008). Speech recognition with weighted finite-state transducers. In *Springer Handbook of Speech Processing* (pp. 559-584). Springer Berlin Heidelberg.

Morbini, F., Audhkhasi, K., Sagae, K., Artstein, R., Can, D., Georgiou, P., & Traum, D. (2013, August). Which ASR should I choose for my dialogue system. In *Proceedings of the 14th annual SIGdial Meeting on Discourse and Dialogue* (pp. 394-403).

Niculescu, A. I., & de Jong, F. M. G. (2008). Development of a speech recognition system for Spanish broadcast news.

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015, April). Librispeech: an ASR corpus based on public domain audio books. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on* (pp. 5206-5210). IEEE.

Pikrakis, A., Giannakopoulos, T., & Theodoridis, S. (2008). A speech/music discriminator of radio recordings based on dynamic programming and bayesian networks. *IEEE Transactions on Multimedia*, *10*(5), 846-857.

Pineda, L. A., Pineda, L. V., Cuétara, J., Castellanos, H., & López, I. (2004, November). DIMEx100: A new phonetic and speech corpus for Mexican Spanish. In *Ibero-American Conference on Artificial Intelligence* (pp. 974-983). Springer Berlin Heidelberg.

Platek, O. (2014). Speech Recognition using KALDI (Doctoral dissertation, Master thesis).

Plátek, O., & Jurcıcek, F. (2014, June). Free on-line speech recogniser based on Kaldi ASR toolkit producing word posterior lattices. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)* (pp. 108-112).

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N. & Silovsky, J. (2011). The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding* (No. EPFL-CONF-192584). IEEE Signal Processing Society.

Ranchal, R., Taber-Doughty, T., Guo, Y., Bain, K., Martin, H., Robinson, J. P., & Duerstock, B. S. (2013). Using speech recognition for real-time captioning and lecture transcription in the classroom. *IEEE Transactions on Learning Technologies*, *6*(4), 299-311.

Robert-Ribes, J. (1998, December). On the use of automatic speech recognition for TV captioning. In *ICSLP*.

Rodríguez-Fuentes, L. J., Penagarikano, M., Varona, A., Diez, M., & Bordel, G. (2012, May). KALAKA-2: a TV Broadcast Speech Database for the Recognition of Iberian Languages in Clean and Noisy Environments. In *LREC* (pp. 99-105).

Rousseau, A., Deléglise, P., & Esteve, Y. (2012, May). TED-LIUM: an Automatic Speech Recognition dedicated corpus. In *LREC* (pp. 125-129).

Saon, G., & Chien, J. T. (2012). Large-vocabulary continuous speech recognition systems: A look at some recent advances. *IEEE Signal Processing Magazine*, *29*(6), 18-33.

Schneider, D., Tschöpel, S., & Schwenninger, J. (2012). Social recommendation using speech recognition: Sharing TV scenes in social networks. In *WIAMIS* (pp. 1-4).

Schuster, M. (2010). Speech recognition for mobile devices at Google. *Pacific Rim International Conference on Artificial Intelligence*, 8-10, Springer Berlin Heidelberg.

Stolcke, A. (2002, September). SRILM-an extensible language modeling toolkit. In *Interspeech* (Vol. 2002, p. 2002).

Stolcke, A., Zheng, J., Wang, W., & Abrash, V. (2011, December). SRILM at sixteen: Update and outlook. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop* (Vol. 5).

Stüker, S., Fügen, C., Kraft, F., & Wölfel, M. (2007, August). The ISL 2007 English speech transcription system for european parliament speeches. In *INTERSPEECH* (pp. 2609-2612).

Varela, A., Cuayáhuitl, H., & Nolazco-Flores, J. A. (2003). Creating a Mexican Spanish version of the CMU Sphinx-III speech recognition system. *Iberoamerican Congress on Pattern Recognition*, 251-258. Springer Berlin Heidelberg.

Vertanen, K. (2006). Baseline WSJ acoustic models for HTK and Sphinx: Training recipes and recognition experiments. Technical report). Cambridge, United Kingdom: Cavendish Laboratory.

Walker, W., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E. & Woelfel, J. (2004). Sphinx-4: A flexible open source framework for speech recognition.

Yao, X., Bhutada, P., Georgila, K., Sagae, K., Artstein, R., & Traum, D. R. (2010, May). Practical Evaluation of Speech Recognizers for Virtual Human Dialogue Systems. In *LREC*.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X. & Valtchev, V. (2006). The HTK book (v3. 4). *Cambridge University*.

Zahid, S., Hussain, F., Rashid, M., Yousaf, M. H., & Habib, H. A. (2015). Optimized audio classification and segmentation algorithm by using ensemble methods. *Mathematical Problems in Engineering*, *2015*.