



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Herramientas para la detección de menciones a palabras clave
en flujos de audio*

Grado en Ingeniería Mecánica

ALUMNO: Kevin Barroso Marín

DIRECTORES: Norberto Fernández García
José P. González Coma

CURSO ACADÉMICO: 2020-2021

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Herramientas para la detección de menciones a palabras clave
en flujos de audio*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

Universida_{de}Vigo

RESUMEN

El *audio mining* es una técnica que está actualmente en alza, ya que desempeña una serie de funciones muy diversas. Dentro de estas funciones, este trabajo se centra en la detección de menciones a palabras. Esta función tiene multitud de aplicaciones, desde asistentes virtuales o marketing hasta la que más interesa en el objetivo y el contexto de este trabajo, la inteligencia militar.

Para realizar esta operación de búsqueda de menciones a palabras, existen tres alternativas diferentes: *search by sound*, búsqueda de patrones en la señal sonora; *keyword spotting*, detección de menciones a términos usando características extraídas del audio y/o lingüísticas o *speech recognition*, donde el audio se transcribe a texto y ahí se busca la mención. Para encontrar la mejor herramienta que realice la función deseada, se lleva a cabo un estudio de una selección de herramientas que desempeñan la función. Se elige la aplicación Vosk, se procede a su instalación y se describe su funcionamiento. A continuación, con el propósito de averiguar qué factores son los que influyen en la calidad de la transcripción y de la búsqueda de menciones a palabras, se realizan diversas pruebas en diferentes escenarios (incluyendo comunicaciones navales) y se analizan los resultados, para al final concluir si influyen o no y en qué medida los factores estudiados.

PALABRAS CLAVE

Minería de Audio, Detección de menciones, Reconocimiento de voz, Transcripción

AGRADECIMIENTOS

Para empezar, agradecer a los tutores de este TFG Norberto Fernández García y José Pablo González Coma por sus aportaciones y sugerencias, y sobre todo por su constante ayuda para dar forma a este trabajo.

Agradecer a mis compañeros de promoción y amigos por la ayuda a la hora de resolver alguna de las dudas surgidas y por ayudarme en la creación de algunos de los audios usados en el trabajo.

A todos y cada uno de los miembros de mi familia, que me han apoyado incansablemente a lo largo de mi vida. Y especialmente a mis padres y hermano, sin ellos nunca podría haber sido quién soy hoy en día.

A todos, gracias.

CONTENIDO

Contenido	1
Índice de Figuras	4
Índice de Tablas.....	6
1 Introducción y objetivos	8
1.1 Introducción	8
1.1.1 ¿Qué es Audio Mining?	8
1.1.2 ¿Por qué es interesante el Audio Mining?	8
1.2 Objetivos	13
1.2.1 Aplicaciones de la detección de una palabra o secuencia de palabras.....	13
1.2.2 Search by sound	15
1.2.3 Word spotting o Keyword spotting (KWS)	15
1.2.4 Speech recognition.....	15
1.3 Estructura	16
2 Estado del arte	17
2.1 Soundfingerprinting	17
2.1.1 ¿Qué es Soundfingerprinting?	17
2.1.2 Creación y versiones.....	17
2.1.3 ¿Cómo funciona?	18
2.1.4 Otros datos de interés.....	21
2.2 pyAudioAnalysis.....	22
2.2.1 ¿Qué es pyAudioAnalysis?	22
2.2.2 Creación y versiones.....	22
2.2.3 ¿Cómo funciona?	23
2.2.4 Otros datos de interés.....	26
2.3 Pocketsphinx	27
2.3.1 ¿Qué es Pocketsphinx?	27
2.3.2 Creación y versiones.....	27
2.3.3 ¿Cómo funciona?	27
2.3.4 Otros datos de interés.....	28
2.4 Kaldi KWS (Vosk).....	28
2.4.1 ¿Qué es Kaldi?	28
2.4.2 ¿Qué es Vosk?	29
2.4.3 Creación y versiones.....	29
2.4.4 ¿Cómo funciona?	30

2.4.5 Otros datos de interés.....	30
2.5 Otras herramientas	31
2.5.1 pyAudioProcessing	31
2.5.2 Flashlight	32
2.5.3 Lingvo.....	32
2.6 Comparación de herramientas.....	33
3 Desarrollo	34
3.1 Instalación en Python	34
3.1.1 Prueba con archivos de audio	36
3.1.2 Prueba con micrófono.....	37
3.1.3 Prueba de búsqueda de una palabra	38
3.2 Solución de errores en el proceso de instalación	41
4 Pruebas y validación.....	43
4.1 Métricas de evaluación.....	43
4.1.1 Word Error Rate	43
4.1.2 Precisión y recuperación.....	43
4.2 Prueba de calidad del audio	44
4.2.1 Nivel de ruido mínimo.....	44
4.2.2 Nivel de ruido bajo	45
4.2.3 Nivel de ruido medio-alto	46
4.2.4 Pruebas de búsqueda de la mención de términos.....	47
4.2.5 Análisis de resultados	48
4.3 Prueba de contenido del audio	48
4.3.1 Prueba de voz hablada más música	48
4.3.2 Prueba de voz cantada más música.....	50
4.3.3 Análisis de resultados	52
4.4 Prueba de género del hablante	53
4.4.1 Prueba de voz femenina.....	53
4.4.2 Prueba de voz masculina	53
4.4.3 Prueba de búsqueda de menciones a términos.....	54
4.4.4 Análisis de resultados	55
4.5 Prueba de acento del hablante	55
4.5.1 Prueba de andaluz	55
4.5.2 Prueba de argentino	56
4.5.3 Prueba de canario.....	57
4.5.4 Prueba de catalán	57

4.5.5 Prueba de gallego.....	58
4.5.6 Prueba de puertorriqueño.....	59
4.5.7 Comparación de resultados.....	59
4.6 Prueba de idioma.....	60
4.6.1 Modelo en inglés.....	60
4.6.2 Modelo en español.....	61
4.6.3 Análisis de resultados	62
4.7 Prueba con vocabulario técnico	62
4.7.1 Análisis de resultados	65
4.8 Pruebas en comunicaciones navales	65
4.8.1 Prueba de vocabulario naval.....	65
4.8.2 Prueba de comunicación breve del aviso de un movimiento.....	66
4.8.3 Prueba de transcripción de un “hailing”	67
4.8.4 Prueba de búsqueda de mención a términos	69
4.8.5 Análisis de resultados	69
4.9 Pruebas de detección de términos sin contexto.....	71
5 Conclusiones y líneas futuras	73
5.1 Conclusiones	73
5.1.1 Ruido de fondo	73
5.1.2 Nivel de la música de fondo y canciones.....	73
5.1.3 Género del hablante	74
5.1.4 Procedencia geográfica del hablante	74
5.1.5 Idioma	74
5.1.6 Vocabulario.....	74
5.1.7 Comunicaciones navales.....	74
5.1.8 Detección de términos sin contexto	75
5.1.9 Conclusiones personales	75
5.2 Líneas futuras	75
6 Bibliografía.....	76

ÍNDICE DE FIGURAS

Figura 1-1. Ejemplo de señal de audio en la que aparecen voz, música y un pequeño silencio [63].	9
Figura 1-2. Comparación entre una grabación y los parámetros biométricos de las cuerdas vocales (peritaje lingüístico forense) [62].	9
Figura 1-3. Onda de voz estresada (derecha) y no estresada (izquierda) [8].	10
Figura 1-4. Interfaz de la aplicación Google Live Transcribe [10].	10
Figura 1-5. Onda de audio con ruido de fondo (señalado con fondo blanco) en Audacity.	11
Figura 1-6. Detección de una palabra (señalada en blanco) en un volumen de audio [11].	11
Figura 1-7. Esquema del sistema domótico con control por voz [13].	12
Figura 1-8. Logo (izquierda) e interfaz de Soundhound al detectar una canción (ejemplo: Nyno Vargas – Dicen de mí) [17].	12
Figura 1-9. Altavoz de Alexa y ejemplos de posibles preguntas a realizarle [22].	13
Figura 1-10. Submarino de la Armada Española de la clase S-70 [24].	14
Figura 1-11. Fotografía de la interfaz de una de estas aplicaciones. Muestra los resultados más frecuentes [25].	14
Figura 1-12. Ejemplo de huellas digitales representando 1,8 segundos de contenido de audio [26].	15
Figura 2-1. Icono de Emy [26].	17
Figura 2-2. Espectrograma de una conversación (frecuencia frente a tiempo). Límites de 2KHz y 300Hz representados por las líneas blancas [26].	19
Figura 2-3. Espectrograma basado en los picos de frecuencia [26].	19
Figura 2-4. Espectrogramas cortados de un archivo de audio [26].	20
Figura 2-5. Espectrograma de umbral de la señal anteriormente representada [26].	20
Figura 2-6. Dos imágenes agrupadas en un conjunto o cubo de hash [26].	21
Figura 2-7. p' y q' están en el mismo cubo hash [26].	21
Figura 2-8. Interfaz de almacenamiento de huellas digitales Emy [29].	22
Figura 2-9. Diagrama general de pyAudioAnalysis [32].	23
Figura 2-10. Segmentación supervisada mediante HMM. Clasificación binaria entre voz y música [32].	25
Figura 2-11. Identificación visual de las partes más representativas del audio (<i>thumbnail</i>) [32].	25
Figura 2-12. Ejemplo de visualización de similitudes del contenido de los audios analizados, en este caso canciones analizadas [32].	26
Figura 2-13. Ejemplo de creación de un grafo con HMM [42].	28
Figura 2-14. Vista simplificada de los diferentes componentes de Kaldi [44].	29
Figura 2-15. Ejemplo de clasificación en voz, sonido de pájaros o música con pyAudioProcessing [48].	31
Figura 2-16. Ejemplo de clasificación de géneros musicales mediante MFCC [48].	32
Figura 3-1. Última frase del archivo test.wav	36

Figura 3-2. Captura de pantalla del terminal de Ubuntu que muestra en la parte superior las palabras detectadas y sus características y debajo la oración completa.....	37
Figura 3-3. Transcripción mediante test_microphone.py de la frase “este es un ejemplo”.	38
Figura 3-4. Código del script test_words.py con la línea 20 señalada y con las palabras “tres” y “nueve” escritas para buscarlas.....	39
Figura 3-5. Código de test_words.py comentando la funcionalidad de mostrar por pantalla los resultados parciales.	40
Figura 3-6. Resultado de la búsqueda de las palabras “tres” y “nueve” en el audio ejemplo.wav. .	41
Figura 3-7. Captura de pantalla de la interfaz de online-convert.com.	42
Figura 4-1. Transcripción del anuncio de televisión.	48

ÍNDICE DE TABLAS

Tabla 1. Lista completa de características de audio implementadas en pyAudioAnalysis [32].	24
Tabla 2. Tabla comparativa de las herramientas estudiadas.	33
Tabla 3. Análisis de la transcripción del audio sin nada de ruido.	44
Tabla 4. Análisis de la transcripción del audio con ruido ambiental.	45
Tabla 5. Análisis de la transcripción del audio con ruido de tráfico intenso.	46
Tabla 6. Resultados de la búsqueda de menciones de palabras en la prueba de calidad del audio.	47
Tabla 7. Análisis de la transcripción de un anuncio con música.	49
Tabla 8. Resultado de la búsqueda de palabras en el anuncio de televisión.	50
Tabla 9. Análisis de la transcripción de versos de la canción “Yo voy a ser Rey León”.	51
Tabla 10. Análisis de la transcripción de versos de la canción “Tanto la quería”.	51
Tabla 11. Resultados de la búsqueda de palabras en canciones.	52
Tabla 12. Análisis de la transcripción del audio con hablante femenina.	53
Tabla 13. Análisis de la transcripción del audio con hablante masculino.	54
Tabla 14. Resultado de la búsqueda de menciones a términos en la prueba de género del hablante.	54
Tabla 15. Análisis de la transcripción del audio con hablante andaluz.	56
Tabla 16. Análisis de la transcripción del audio con hablante argentino.	56
Tabla 17. Análisis de la transcripción del audio con hablante canario.	57
Tabla 18. Análisis de la transcripción del audio con hablante catalán.	58
Tabla 19. Análisis de la transcripción del audio con hablante gallego.	58
Tabla 20. Análisis de la transcripción del audio con hablante puertorriqueño.	59
Tabla 21. Tabla resumen de los resultados de la prueba de acentos.	59
Tabla 22. Análisis de la transcripción del audio con el modelo en inglés.	61
Tabla 23. Análisis de la transcripción del audio con el modelo en español.	61
Tabla 24. Análisis de la transcripción del audio con vocabulario específico de mitología griega.	63
Tabla 25. Análisis de la transcripción del audio con vocabulario específico sobre motores.	64
Tabla 26. Resultado de la búsqueda de menciones a términos en los audios sobre mitología y motores.	64
Tabla 27. Análisis de la transcripción del audio de prueba de vocabulario naval.	66
Tabla 28. Análisis de la transcripción del audio con un aviso de un movimiento a otro buque.	67
Tabla 29. Análisis de la transcripción del audio de un <i>hailing</i> .	68
Tabla 30. Resultado de la búsqueda de menciones a términos en los audios de comunicaciones navales.	69
Tabla 31. Análisis de la transcripción de un <i>hailing</i> grabado con micrófono.	71
Tabla 32. Prueba de transcripción y búsqueda de palabras en español.	72

Tabla 33. Prueba de transcripción y búsqueda de palabras en inglés.....	72
---	----

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción

En este apartado se va a introducir el contexto del trabajo, definiendo unos conceptos básicos como el de *Audio Mining* y por qué es útil.

1.1.1 ¿Qué es Audio Mining?

La minería de audio o *audio mining* es la “técnica que consiste en la búsqueda automática de patrones en grandes volúmenes de audio, utilizando técnicas de correlación estadística y categorización” [1].

La minería de audio, también se define como “técnica que se utiliza para buscar y analizar archivos de audio en busca de apariciones de palabras o frases habladas” [2].

La investigación seria sobre minería de audio comenzó a fines de la década de 1970. Se han realizado investigaciones en varias escuelas importantes, incluidas la Universidad Carnegie Mellon, la Universidad de Columbia, el Instituto de Tecnología de Georgia y la Universidad de Texas. No fue hasta 2002, cuando la tecnología comenzó a ofrecer niveles aceptables de rendimiento y precisión para uso comercial, según Jeff Karnes, gerente de Virage (un proveedor de minería de audio). Los productos de minería de audio generalmente se han integrado en sistemas más grandes porque “la capacidad de buscar y reproducir contenido es más valiosa como parte de un “producto” que administra grandes archivos de información, como un sistema de administración de medios” [3].

1.1.2 ¿Por qué es interesante el Audio Mining?

El *audio mining* es una técnica muy usada actualmente, ya que desempeña diferentes tareas en multitud de aplicaciones. Estos son algunos ejemplos del empleo de la minería de audio:

- **Segmentación y clasificación de audio:** esta aplicación consiste en la separación del audio en diferentes segmentos, para favorecer la clasificación y la extracción de características de este. Es una parte muy importante en el preprocesado, pero también tiene alguna aplicación directa. Por ejemplo, sirve para identificar y separar los fragmentos de audio que contengan voz, silencio o música, un ejemplo se muestra en la Figura 1-1. Esta técnica es muy útil para herramientas de grabación de voz, (p.ej.: MP3 Recorder Studio), utilizada para detectar silencios y detener la grabación si se detecta un silencio durante un tiempo establecido [4].

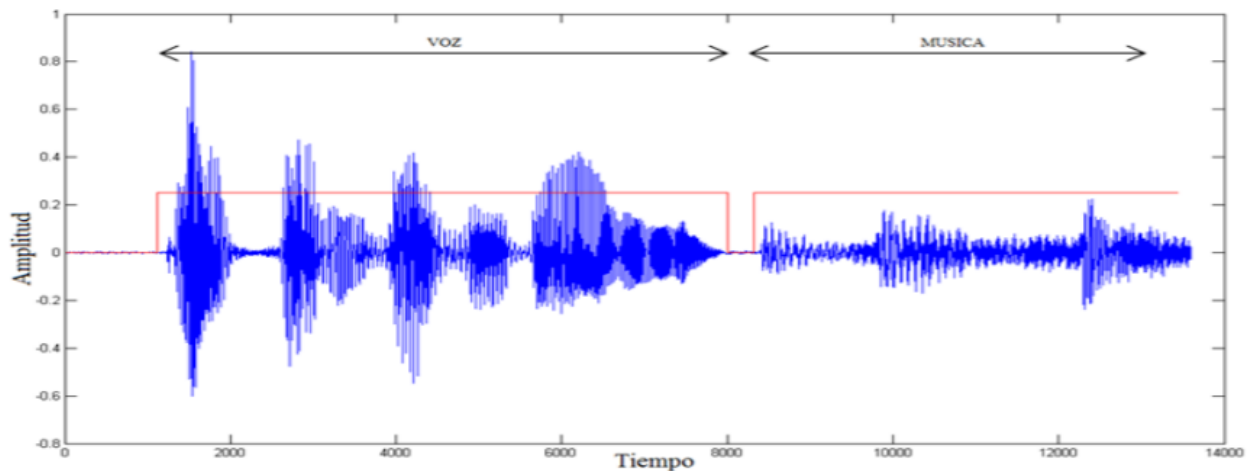


Figura 1-1. Ejemplo de señal de audio en la que aparecen voz, música y un pequeño silencio [63].

- **Identificación del hablante:** esta función consiste en una identificación de la persona que está hablando a través de un análisis sonoro de la voz. Esta aplicación es usada en contextos muy diferentes, entre ellos el que le da la policía científica para el análisis de pruebas sonoras contra un acusado, realizando un peritaje lingüístico para obtener una resolución inculpatória o exculpatória [5]. Este proceso se hace comparando los parámetros biométricos del acusado con el audio examinado como se indica en la Figura 1-2.

También se puede aplicar esta función en empresas, analizando conversaciones con posibles clientes, de modo que identifican el género y edad aproximada del hablante para poder ofrecerle una gama de productos más adecuada a sus características.

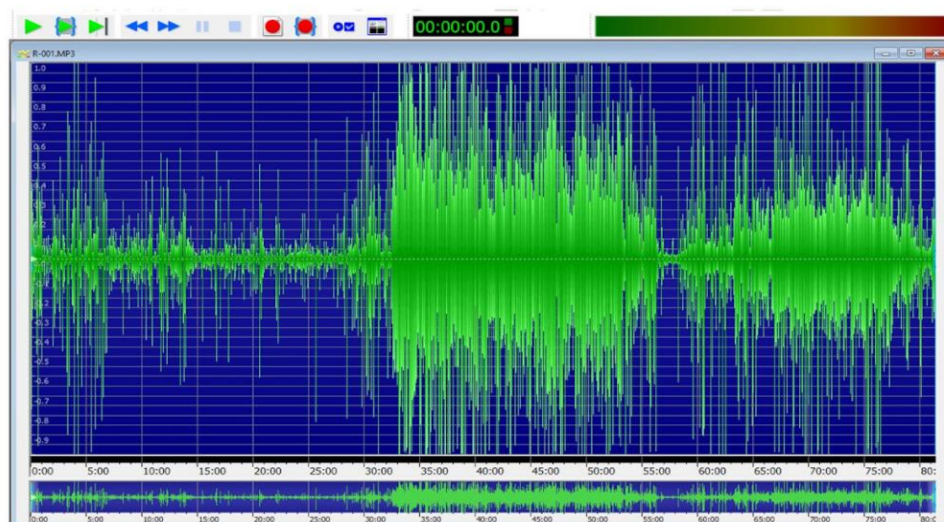


Figura 1-2. Comparación entre una grabación y los parámetros biométricos de las cuerdas vocales (peritaje lingüístico forense) [62]

- **Detectar el nivel de estrés del hablante:** existen herramientas que mediante el análisis de audios detectan el nivel de estrés e incluso las mentiras de un hablante. El PSE (*Psychological Stress Evaluator*) es una herramienta que mide y muestra gráficamente componentes relacionados con el estrés en la voz. Está basado en el análisis de las

pequeñas modulaciones de frecuencia involuntarias provocadas por las oscilaciones de los músculos utilizados en el mecanismo de la voz [6] [7] como a continuación muestra la Figura 1-3.

Es una aplicación muy usada por cuerpos de seguridad, investigación criminal, evaluación de empleados e incluso diagnósticos médicos o psiquiátricos.

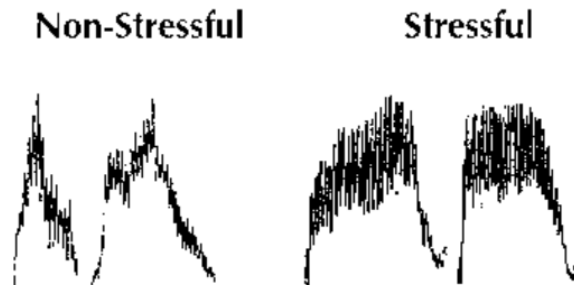


Figura 1-3. Onda de voz estresada (derecha) y no estresada (izquierda) [8].

- **Reconocimiento automático de voz:** está técnica es usada para convertir la voz humana en texto, de modo que el humano pueda establecer una comunicación con el ordenador únicamente a través de la voz. Tiene variedad de aplicaciones, como el dictado de recetas o informes médicos (p.ej.: INVOX Medical) [9]. También es una tarea muy útil para aquellas personas que por su estado físico no puedan hacer uso de la pantalla táctil de un dispositivo móvil y necesiten de un sustituto del teclado para realizar algún tipo de escritura (p.ej.: Google Live Transcribe o Speechnotes).

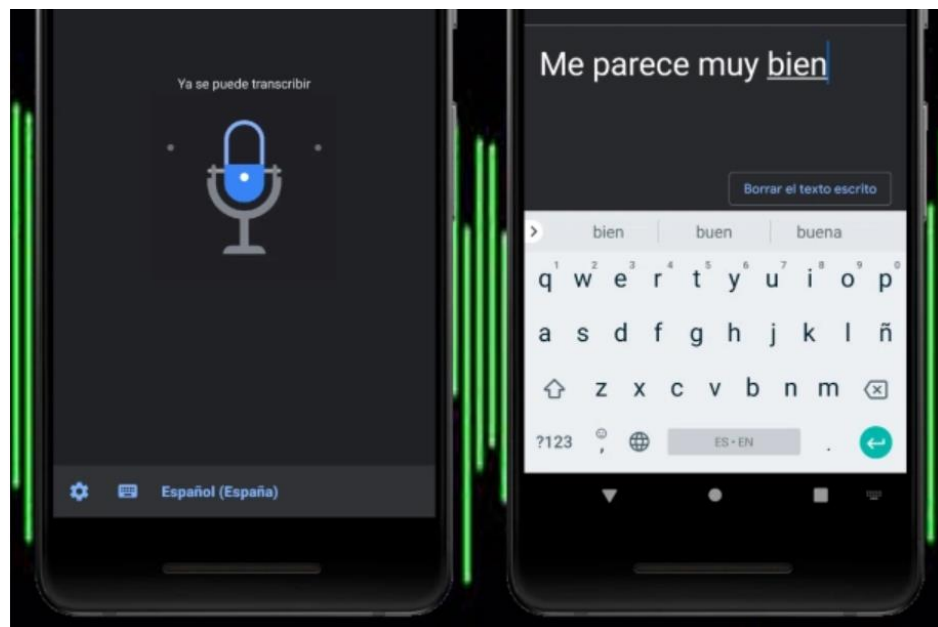


Figura 1-4. Interfaz de la aplicación Google Live Transcribe [10].

- **Identificar el contexto en el que se encuentra el hablante:** esta aplicación se usa para el reconocimiento del ruido de fondo que aparece en el audio para identificar si el hablante se encuentra en la calle, en su casa, en una fábrica, etc. Se basa en la caracterización de los ruidos de fondo, y muchas empresas lo usan para marketing. En una llamada pueden reconocer el ruido de fondo (como en la onda mostrada en la Figura 1-5) para saber las horas en las cuales esta no debe llamar si el individuo está en el trabajo o las horas donde sí

puede hacerlo. Identificando el ruido de fondo se puede detectar que está en casa, si tiene niños o bebés e incluso los canales de televisión que está viendo, para ajustar y personalizar sus ofertas.

Esta técnica también es usada por muchos delincuentes para identificar en qué momento del día una casa se encuentra vacía, ya que una llamada a la víctima puede determinar que esta se encuentra en el trabajo.

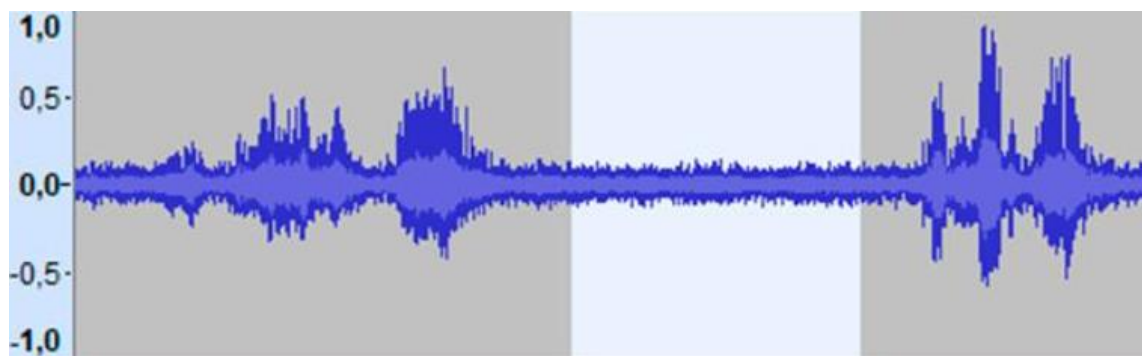


Figura 1-5. Onda de audio con ruido de fondo (señalado con fondo blanco) en Audacity.

- **Detección de palabras o secuencias de palabras en un audio:** consiste en la búsqueda de una palabra en concreto o de una secuencia de estas en un flujo de audio. En la Figura 1-6 podemos observar la búsqueda y coincidencia de la palabra que estamos buscando en el volumen de audio analizado.

En el ámbito militar, esta técnica es muy útil en aplicaciones de inteligencia, por ejemplo, para centrar la atención del analista únicamente en aquellas partes donde se mencionen ciertas ideas, automatizando el procesamiento de grandes volúmenes de audio o vídeo con la finalidad de optimizar el tiempo del operador humano, quien se centrará en las partes importantes del audio.



Figura 1-6. Detección de una palabra (señalada en blanco) en un volumen de audio [11].

En el ámbito civil, una función muy importante en los dispositivos modernos es el reconocimiento de voz, con diferentes aplicaciones como la de sustitución de pantallas táctiles o botones para el manejo de dispositivos electrónicos o incluso para inteligencia artificial. Un ejemplo a pequeña escala de esto pueden ser los altavoces [12] o las casas inteligentes (como muestra la Figura 1-7), a los cuales se les introducen comandos a partir de voz e incluso a medida que avanza el desarrollo de las aplicaciones de este tipo, se le

pueden dar órdenes mediante voz utilizando un vocabulario más extenso que unos simples comandos de voz.

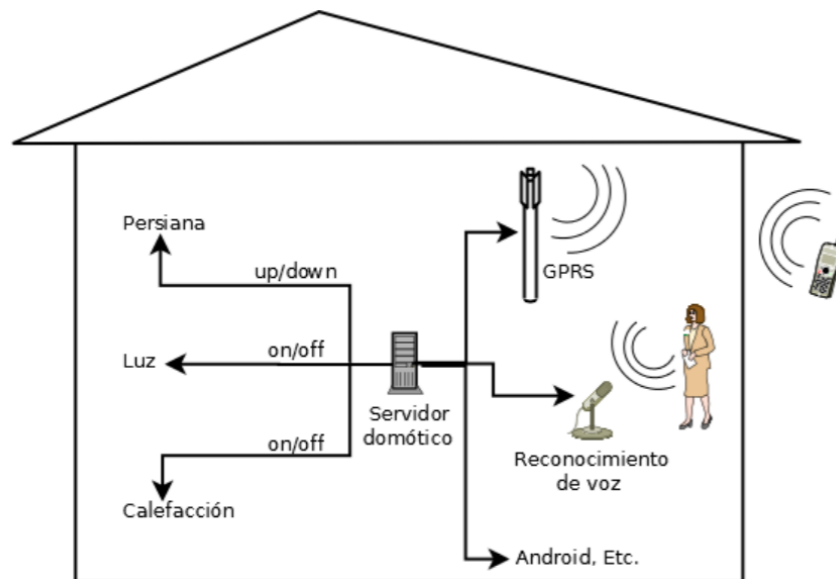


Figura 1-7. Esquema de un sistema domótico con control por voz [13].

- **Búsqueda de audio:** esta técnica consiste en la búsqueda de música a partir de un fragmento de esta. Las aplicaciones que desempeñan esta función escuchan un fragmento de la canción que está sonando en el entorno del usuario y, tras cotejar ese fragmento en su base de datos devuelven al usuario toda la información de la canción sonante. Además, hoy en día, estas tecnologías son capaces de encontrar canciones a partir de fragmentos de canción con mucho ruido de fondo, con algo de voz, e incluso si el usuario la tatarea o silba. Hay multitud de aplicaciones disponibles que desempeñan esta función, algunas mundialmente conocidas como Shazam [14] , Google Sound Search [15] o Soundhound [16]. A continuación, en la Figura 1-8 se muestra el ejemplo de una coincidencia de búsqueda en Soundhound.

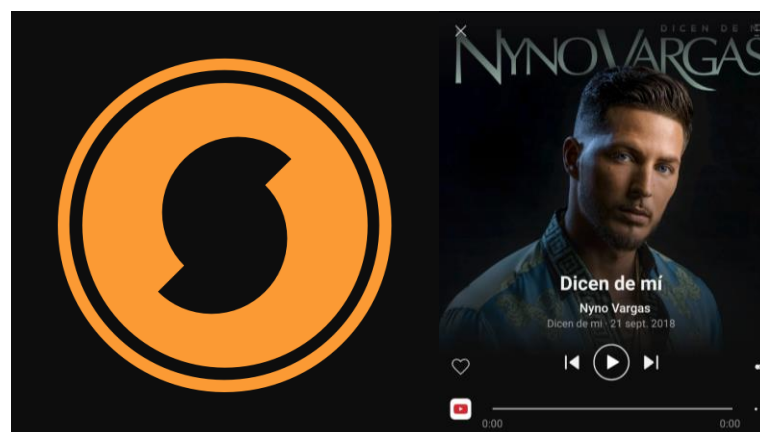


Figura 1-8. Logo (izquierda) e interfaz de Soundhound al detectar una canción (ejemplo: Nyno Vargas – Dicen de mí) [17].

1.2 Objetivos

Dentro de las múltiples tareas que se pueden llevar a cabo usando técnicas de minería de audio, este trabajo se va a centrar en una muy concreta: la detección de una palabra o secuencia de palabras en un audio.

1.2.1 Aplicaciones de la detección de una palabra o secuencia de palabras

Esta tiene varias aplicaciones tanto en ámbito militar como civil. A continuación, se exponen algunas de las más importantes:

- **Asistentes digitales:** un asistente digital es un software que permite interactuar con él mediante comandos de voz para la realización de diversas tareas. Las tareas que son capaces de realizar no se basan en tareas básicas como navegación, preguntas y respuestas o llamadas telefónicas, sino que también permiten tareas más complejas como compras, administración del correo electrónico o calendario, e incluso el control remoto de otros aparatos. Algunos de los principales asistentes de voz son Google Assistant [18], Siri (Apple) [19], Cortana (Microsoft) [20] o Alexa (Amazon) [21].

Para entender cómo funciona, se explicará a través del ejemplo de Alexa. Este asistente fue lanzado en 2014 en Estados Unidos, con el objetivo de crear la forma más natural de interactuar con un sistema de inteligencia artificial. A Alexa se le puede preguntar directamente algunas cuestiones como el tiempo que va a hacer, el estado del tráfico o que reproduzca una canción, etc. A continuación la Figura 1-9 muestra algunos ejemplos más de estos comandos [22].



Figura 1-9. Altavoz de Alexa y ejemplos de posibles preguntas a realizarle [22].

- **Marketing:** muchas empresas deciden monitorizar su marca, es decir, detectar y analizar todo aquello que hable sobre la marca en cuestión. Esta aplicación es ampliamente usada por las empresas en los medios de comunicación como la radio o la televisión, con la intención de encontrar todo aquel flujo de audio en el que se mencione la palabra clave, en este caso el nombre de la empresa o de la marca.

Además, varias empresas utilizan este método en atención al cliente, para detectar ciertas llamadas que requieren de más importancia para la compañía, como por ejemplo la mención de las palabras “dar de baja” [23].

- **Inteligencia militar:** en el ámbito militar, la monitorización de comunicaciones es clave en misiones de inteligencia. La escucha de comunicaciones de buques navegando en la zona de operaciones en búsqueda de palabras concretas, permitiendo al operador decidir dónde debe prestar más atención. En España, sobre todo los submarinos tienen un papel especial en la recolección de inteligencia [24].

En el conflicto de Libia, según expresa la página de la Armada, los submarinos clase S-70 (mostrado en la Figura 1-10) desempeñaron un papel fundamental frente a las costas de Libia, llevando a cabo una serie de operaciones encubiertas de vigilancia del tráfico mercante, así como obtención de inteligencia y monitorización de las comunicaciones.



Figura 1-10. Submarino de la Armada Española de la clase S-70 [24].

- **Identificación de actividades ilegales en conversaciones:** este tipo de tecnología se usa también para la identificación de algún tipo de ilegalidad en conversaciones de presos en las cárceles de Estados Unidos (la Figura 1-11 muestra algunas de estas). El software desarrollado para esta aplicación, es de gran ayuda para resolver crímenes que dejaron algún cabo sin resolver o incluso para prevenir algún asesinato. Por ejemplo, en una llamada telefónica a la cárcel del condado de Suffolk, Nueva York, un recluso amenazó por teléfono con matar a un juez, identificándolo con su nombre incluso. El software de LEO Technologies detectó y alertó sobre la conversación para, en este caso, prevenir un asesinato [25].



Figura 1-11. Fotografía de la interfaz de una de estas aplicaciones. Muestra los resultados más frecuentes [25].

Las plataformas o herramientas desarrolladas para desempeñar la función de indexación de búsqueda de patrones en audio se pueden dividir básicamente en 3 grupos según el método que usen para indexar y buscar en el flujo de audio: *Search by Sound*, *Word spotting* o *Keyword spotting (KWS)* y *Speech recognition*.

1.2.2 Search by sound

Este método se basa en la recuperación de información a partir de la búsqueda de coincidencias entre una entrada de audio y una gran base de datos almacenada. Es decir, busca una similitud entre el patrón sonoro de entrada (el que el usuario introduce) y el patrón sonoro del fichero o flujo de audio (base de datos) que el sistema tiene almacenada.

Este tipo de búsqueda se basa en el uso de huellas de audio (*audio fingerprinting*), que son asociaciones entre elementos multimedia y la información que contiene. Estas huellas no tienen que ser necesariamente detalladas como para que se pueda replicar exactamente la onda del audio inicial, sino que, el principio que han de cumplir, es que la entrada y el contenido de la base de datos al que se quiere llegar en la búsqueda tengan esta misma huella digital. Un ejemplo de huella digital pueden ser cadenas de números enteros obtenidos a partir de herramientas. Las cadenas de números enteros como la que muestra la Figura 1-12, son las que se buscan en la base de datos almacenada [26].

287121152, 1275791411, 539499396, 185209916, 319820588, 1244475492,
1751526233, 1862426472, 306596106, 204033582, 1242574602, 50341460,
773194334, 339627067, 2014447110, 738722113, 587801446, 320160293,
540288008, 1146191520, 461079818, 959409810, 889623606, 16778759,
285488401

Figura 1-12. Ejemplo de huellas digitales representando 1,8 segundos de contenido de audio [26].

La aplicación más común para este método es la identificación de música, donde el usuario graba una parte de una canción que este sonando para que las aplicaciones identifiquen la canción y muestren al usuario toda la información de esta contenida en la base de datos. Por ejemplo, el título de la canción, el artista, disco al que pertenece, duración, etc. Existen multitud de aplicaciones con esta función como Shazam o Google Sound Search.

1.2.3 Word spotting o Keyword spotting (KWS)

Word spotting o *Keyword spotting (KWS)* es un método basado en el análisis de un flujo de audio para la búsqueda de una serie de palabras clave. Mediante este método se pueden identificar palabras o expresiones que sean de importancia para nuestra finalidad. Este método utiliza la transcripción de audio, que se basa en aplicar modelos de lenguaje (gramática y sintaxis) [23] [27].

El KWS está muy extendido en el mundo empresarial, puesto que permite analizar las llamadas de los clientes y detectar algunas palabras clave para la empresa (p.ej. : “dar de baja”) [23] o para rastrear líneas o nombres de productos específicos de las empresas. También es un método usado por los servicios de emergencia para detectar palabras como “muerto” o “arma” en llamadas.

1.2.4 Speech recognition

El reconocimiento automático de voz (ASR) es un método de reconocimiento y transcripción de un lenguaje hablado a texto, para poder analizar y realizar búsquedas directamente en el texto.

Se basa en la generación de una onda eléctrica que se aproxima a la voz humana. Esta onda resultante se compara con la biblioteca almacenada para reconocer la palabra y de esa forma, generar

mediante un sistema lingüístico el texto. De modo que a partir de la voz humana entrante se genera el texto.

Este método tiene multitud de aplicaciones, enfocadas en diferentes funciones, como la seguridad en los vehículos, que incorporan sistemas de navegación por la interfaz del vehículo activados por voz. También se encuentra en auge la implantación de sistemas de *speech recognition* en hospitales y centros de salud para la captura y registro de diagnósticos de pacientes [28].

También se utiliza este método en la implementación de asistentes virtuales, cada vez más integrados en la vida diaria. Otra de las aplicaciones en auge de este método es la autenticación basada en voz, que agrega un nivel aceptable de seguridad en tecnologías de la vida diaria.

1.3 Estructura

El trabajo se organiza en los siguientes capítulos:

- **Capítulo 1: Introducción y objetivos.** Se expone el contexto del trabajo, definiendo algunos conceptos generales. También quedarán reflejadas las aplicaciones del campo estudiado y más concretamente de la función que se va a trabajar a lo largo del trabajo.
- **Capítulo 2: Estado del arte.** Se exponen algunas de las herramientas, indicando las características generales y el funcionamiento de cada una de ellas. Al final de este capítulo se elaborará una tabla comparativa para elegir una de estas herramientas para su desarrollo.
- **Capítulo 3: Desarrollo.** Se proporciona un tutorial para instalar la herramienta elegida y se explica el funcionamiento de esta.
- **Capítulo 4: Pruebas.** Se realizan pruebas para comprobar la influencia de ciertos factores en los resultados de búsqueda de palabras y de transcripción.
- **Capítulo 5: Conclusión y líneas futuras.** Se realiza un resumen y un análisis de los resultados de las pruebas realizadas y se comentan algunas líneas para otros posibles proyectos.

2 ESTADO DEL ARTE

Existen multitud de herramientas para realizar la función de búsqueda de palabras o secuencias de palabras en un volumen de audio. A continuación, se exponen algunas de las más interesantes para compararlas al final de este capítulo.

2.1 Soundfingerprinting

Soundfingerprinting es una herramienta de código libre que cumple la función de búsqueda de palabras o secuencia de palabras en un audio usando la técnica de *Search by sound*.

2.1.1 ¿Qué es Soundfingerprinting?

Es un *framework* (marco de trabajo) de código libre diseñado para el trabajo en el campo del audio y el procesamiento de señales digitales, así como del reconocimiento de audio. Implementa un algoritmo eficiente para una recuperación e inserción de huellas acústicas, con una alta tasa de recuperación y precisión [29].

Soundfingerprinting es muy usado en interfaces como Emy [26] en el ámbito de generación de huellas de audio. Emy permite el almacenamiento de estas huellas para su consulta mediante el algoritmo de Soundfingerprinting, además aporta una interfaz para el uso de la herramienta.



Figura 2-1. Icono de Emy [26].

2.1.2 Creación y versiones

Fue publicada en noviembre de 2011 por Sergiu Ciumac, ingeniero de software especializado en procesamiento de señales digitales, telecomunicaciones, comercio electrónico y desarrollo web. Sergiu Ciumac, además es CEO y cofundador de nemo.ai, un motor de búsqueda personalizada y recomendación de productos para tiendas de comercio electrónico [30].

Actualmente, la última versión disponible es la versión 7.5.5 publicada el 25 de enero de 2021, fruto de muchas mejoras y solución de problemas en el código. Algunas de las versiones contienen mejoras muy notables [29] :

- A partir de la versión 5.0.0, la biblioteca de huellas digitales es compatible con .NET (plataforma de desarrollo gratuita de código abierto para compilar varios tipos de aplicaciones [31]).
- En la versión 5.1.0 la generación de huellas digitales se vuelve de alta precisión, evolucionando a una firma de huellas mucho más resistente al ruido. Por el contrario, las huellas de versiones anteriores dejan de ser compatibles y hay que volver a indexar los datos.
- En la versión 6.0.0, se proporciona una interfaz mejorada, además se puede insertar una nueva clase de datos para representar información de pistas de audio.
- Una de las mejoras más significativas viene en la actualización 6.2.0, donde el *framework* ofrece la posibilidad de consultar fuentes de datos en tiempo real, de forma que en versiones posteriores se pueda supervisar un flujo de datos en tiempo real obteniendo resultados rápidamente.

2.1.3 ¿Cómo funciona?

Este algoritmo, como la gran mayoría de los que trabajan con huellas digitales de audio, realiza dos pasos para su transformación: compresión con pérdida y hash [26].

Para la compresión, todos los algoritmos de huellas digitales decodifican primero el archivo de entrada a un formato WAV sin procesar, enfocando su atención en tres propiedades: número de canales (mono, estéreo, u otros), frecuencia de muestreo y profundidad de bits (bits por muestra). La cantidad de memoria necesaria para almacenar las muestras de un archivo de audio sería demasiado grande, de modo que se pasa a comprimir el archivo basándose en sus propiedades anteriormente mencionadas. La compresión conseguida puede llegar a ser de 32 veces, ya que realiza los siguientes pasos:

- Número de canales: reducción de estéreo a mono. Se consigue una reducción a la mitad eliminando un canal de audio.
- Frecuencia de muestreo: disminución de la frecuencia típica de muestreo de 44KHz a una frecuencia de 5,5KHz, se crea una reducción x8.
- Profundidad de bits: se disminuye el número de bits por muestra, de 32 bits (más usada en formatos de audio) a 16 bits, consiguiendo una reducción x2.

Incluso después de todas las reducciones, la señal de audio no ofrece facilidades a la hora de comparar con la base de datos, debido a que al cambiar el volumen, por ejemplo, se modificarían las propiedades de la onda. Para evitar este problema, el algoritmo trata al sonido como un espectro de frecuencias.

El algoritmo *Soundfingerprinting* tomará las muestras a 5512Hz, frecuencia elegida según el teorema de muestreo de Nyquist-Shannon ($B < F_s/2$) para capturar frecuencias de 0 a 2756Hz. Está demostrado que la gran mayoría de información relevante en una huella se encuentra en este rango de frecuencias. El algoritmo también incorpora un filtro que elimina frecuencias por debajo de 300Hz y por encima de 2KHz, por ser el rango donde se encuentra la mayoría de información relevante de la huella, como muestra la Figura 2-2.

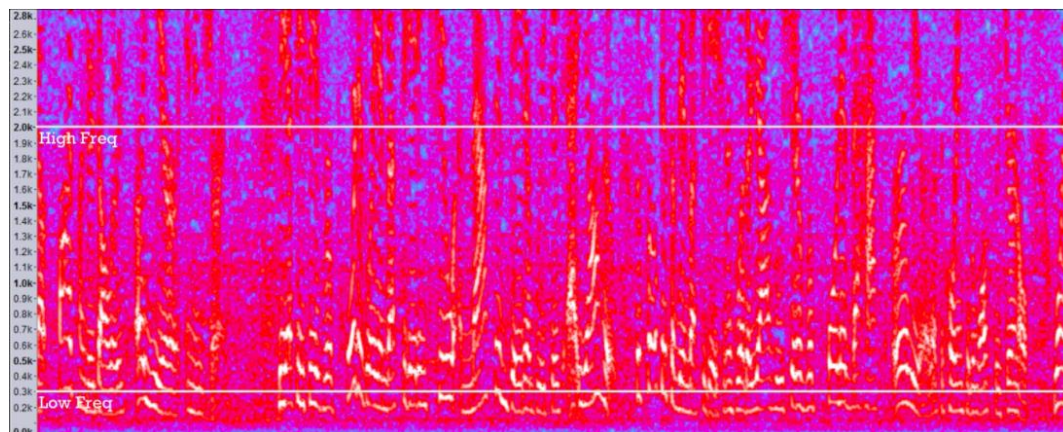


Figura 2-2. Espectrograma de una conversación (frecuencia frente a tiempo). Límites de 2KHz y 300Hz representados por las líneas blancas [26].

El espectrograma mostrado anteriormente representa una conversación entre dos personas de 15s. Las líneas blancas muestran el rango en que trabaja *Soundfingerprinting* para crear huellas digitales de audio, como podemos observar dentro de ese rango se encuentran la gran mayoría de información acústica (más información cuanto más brillante es el área).

Soundfingerprinting se interesa mayormente en los picos de frecuencia, de modo que quedaría un espectrograma como el mostrado en la Figura 2-3.

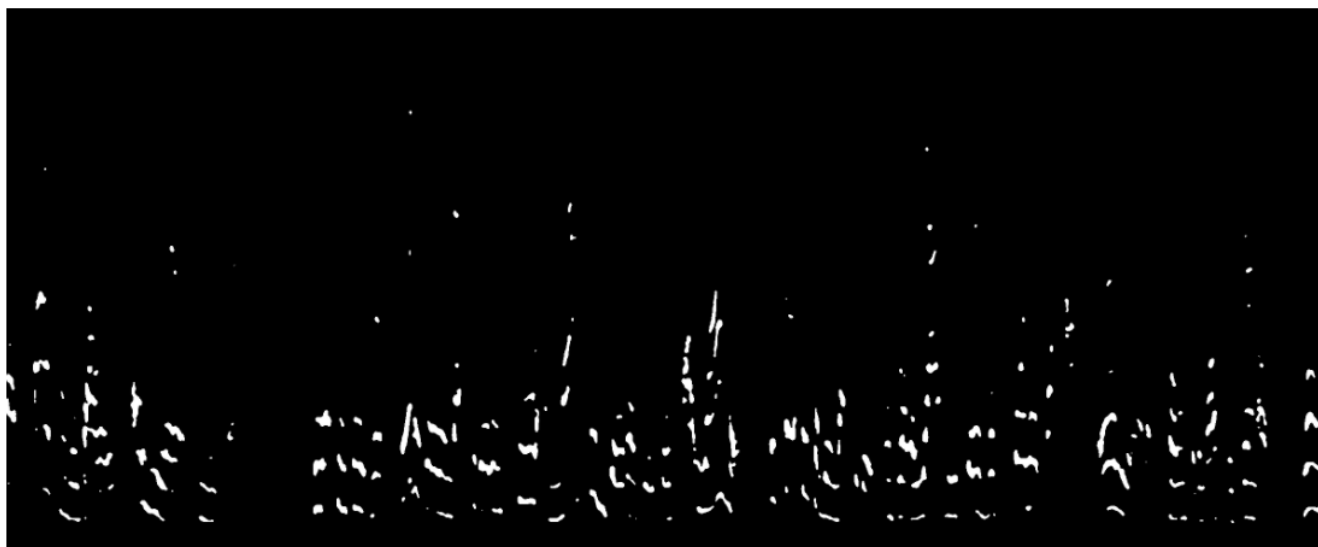


Figura 2-3. Espectrograma basado en los picos de frecuencia [26].

Este algoritmo utiliza 8192 muestras de audio por huella correspondientes a 1,48s. El espectrograma utiliza 64 muestras superpuestas, produciendo imágenes de 128x32 píxeles.

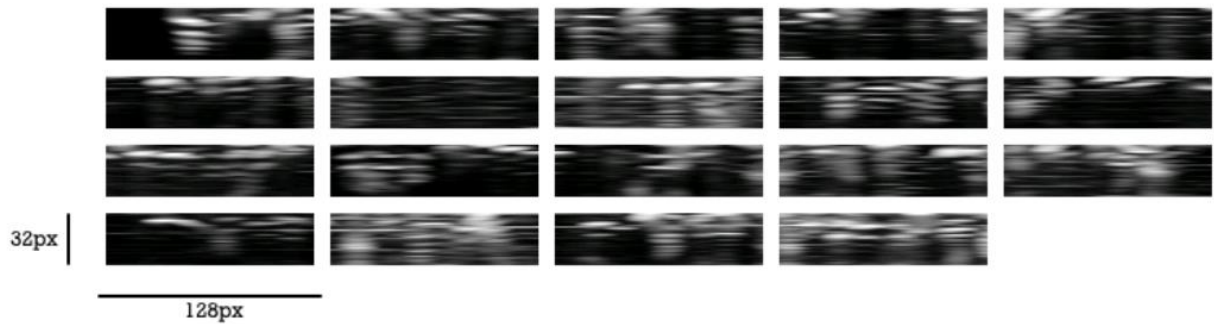


Figura 2-4. Espectrogramas cortados de un archivo de audio [26].

Las imágenes generadas sirven para comparar con imágenes guardadas, pero para facilitar esta comparación, se establecen unos umbrales de picos de frecuencia. Cada píxel funciona de modo que, si está por debajo del umbral, el píxel se convierte en un píxel negro y si está por encima, en uno blanco. Por lo tanto, el espectrograma mostrado en la Figura 2-4 tomará la siguiente forma (Figura 2-5):



Figura 2-5. Espectrograma de umbral de la señal anteriormente representada [26].

El método sigue siendo lento, ya que habría multitud de imágenes en unas pocas horas de contenido de audio, por tanto, *Soundfingerprinting* utiliza una técnica que construye una estructura de datos hash (*Locality Sensitive Hashing*), cuyo propósito general es ignorar pequeñas diferencias entre los puntos cercanos e incluirlos en los mismos grupos o “cubos” (*buckets*). Para construir la estructura de datos LSH, el algoritmo define una métrica de distancia, que aprovecha la escasez de datos y el dominio (0,1) de los mismos (vector de características). La Figura 2-6 muestra un ejemplo de conversión de imágenes a hash.

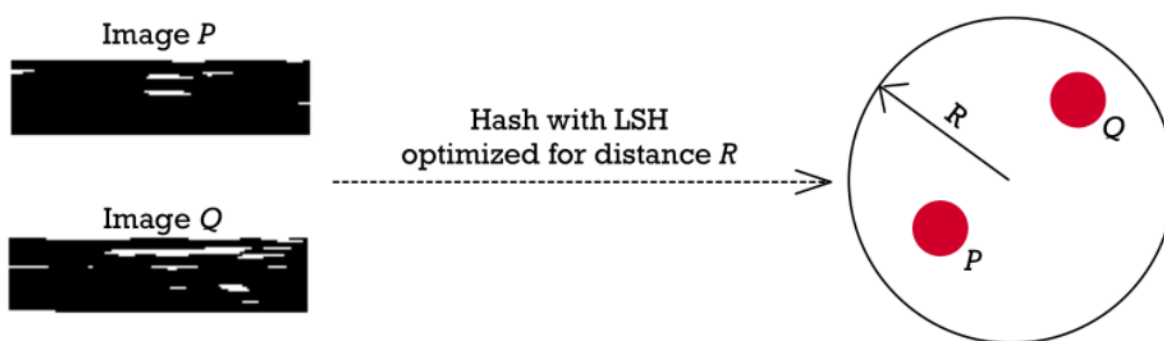


Figura 2-6. Dos imágenes agrupadas en un conjunto o cubo de hash [26].

Para realizar una búsqueda más eficiente, *Soundfingerprinting* divide los puntos en diferentes cubos de hash (Figura 2-7) y solo realiza una búsqueda en aquellos que se encuentran en el mismo cubo de hash, dado que significa que estos estarán relacionados entre sí.

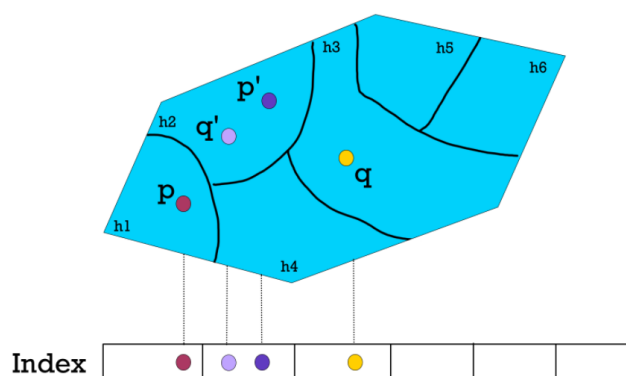


Figura 2-7. p' y q' están en el mismo cubo hash [26].

2.1.4 Otros datos de interés

- El código de esta herramienta fue implementado íntegramente en C#.
- Licencia MIT: posibilidad de uso, modificación y distribución.
- Dispone de una comunidad amplia, ya que cuenta con más de 37000 descargas y 1500 comentarios en la página de GitHub. Además, cuenta con varios foros abiertos.
- 541 estrellas en GitHub.
- Cuenta con una interfaz de uso no comercial especializada en el almacenamiento de huellas digitales (Emy) que permite insertar, consultar y eliminar entradas. A continuación, la Figura 2-8 muestra esta interfaz.

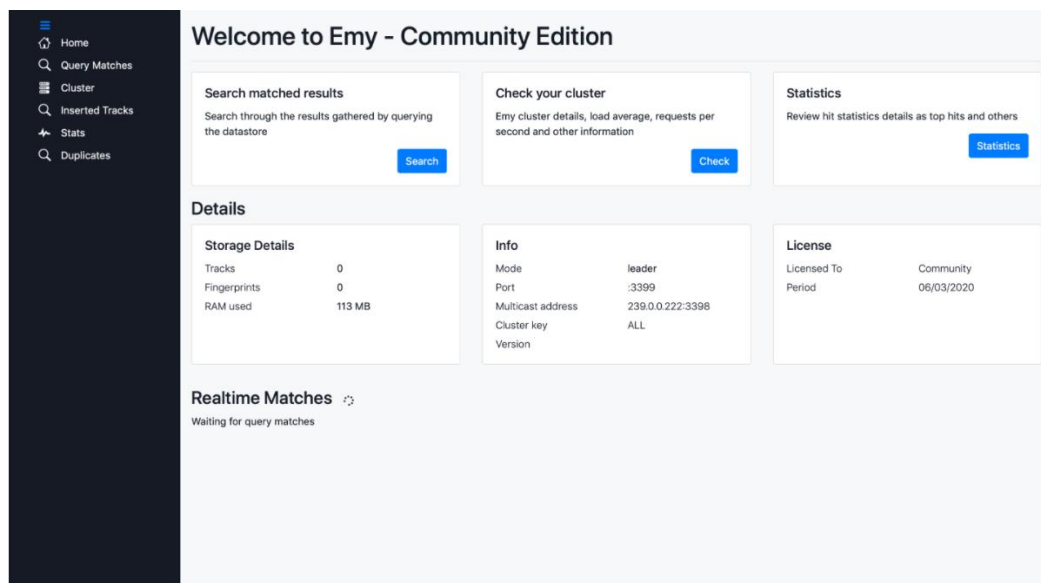


Figura 2-8. Interfaz de almacenamiento de huellas digitales Emy [29].

2.2 pyAudioAnalysis

Estudiamos la herramienta pyAudioAnalysis, que cumple multitud de funciones que son útiles para el objetivo de este trabajo. A continuación, se explicarán todas las funciones que proporciona esta herramienta.

2.2.1 ¿Qué es pyAudioAnalysis?

Es una biblioteca de código abierto de Python que implementa una amplia gama de funcionalidades de análisis de audio y se puede utilizar con varias aplicaciones. Actualmente, las dos principales funciones son: la implementación de una funcionalidad de huellas digitales de audio que se adoptará en el contexto de un sistema de recuperación de audio y la optimización de las funcionalidades de extracción de características. Sin embargo, pyAudioAnalysis ofrece otras funcionalidades como segmentar un flujo de audio, visualizar relaciones de contenido o entrenar y aplicar clasificadores de audio [32].

Con pyAudioAnalysis, también se pueden clasificar segmentos de audio en diferentes conjuntos de clases predefinidas, segmentar una grabación de audio y clasificar los segmentos homogéneos, eliminar áreas de silencio de una grabación de voz, estimar la emoción de un segmento de voz, extraer miniaturas de audio de una pista de música, etc.

2.2.2 Creación y versiones

La librería pyAudioAnalysis fue publicada el 19 de octubre de 2016 por Theodoros Giannakopoulos, ingeniero en informática y telecomunicaciones por la Universidad de Atenas y un máster en Procesamiento de señales e imágenes por la Universidad de Patras. Actualmente Theodoros Giannakopoulos es director de aprendizaje de máquinas en la empresa Behavioral Signals e investigador en NCSR Demokritos [33].

Actualmente, pyAudioAnalysis tiene su versión más reciente en la 0.3.6 (publicada en 2021) después de 13 versiones anteriores para solventar algunos errores de código y para mejorar el rendimiento de la aplicación [34].

2.2.3 ¿Cómo funciona?

pyAudioAnalysis implementa multitud de funciones, se pueden observar en la Figura 2-9. A continuación, se explican cada una de ellas:

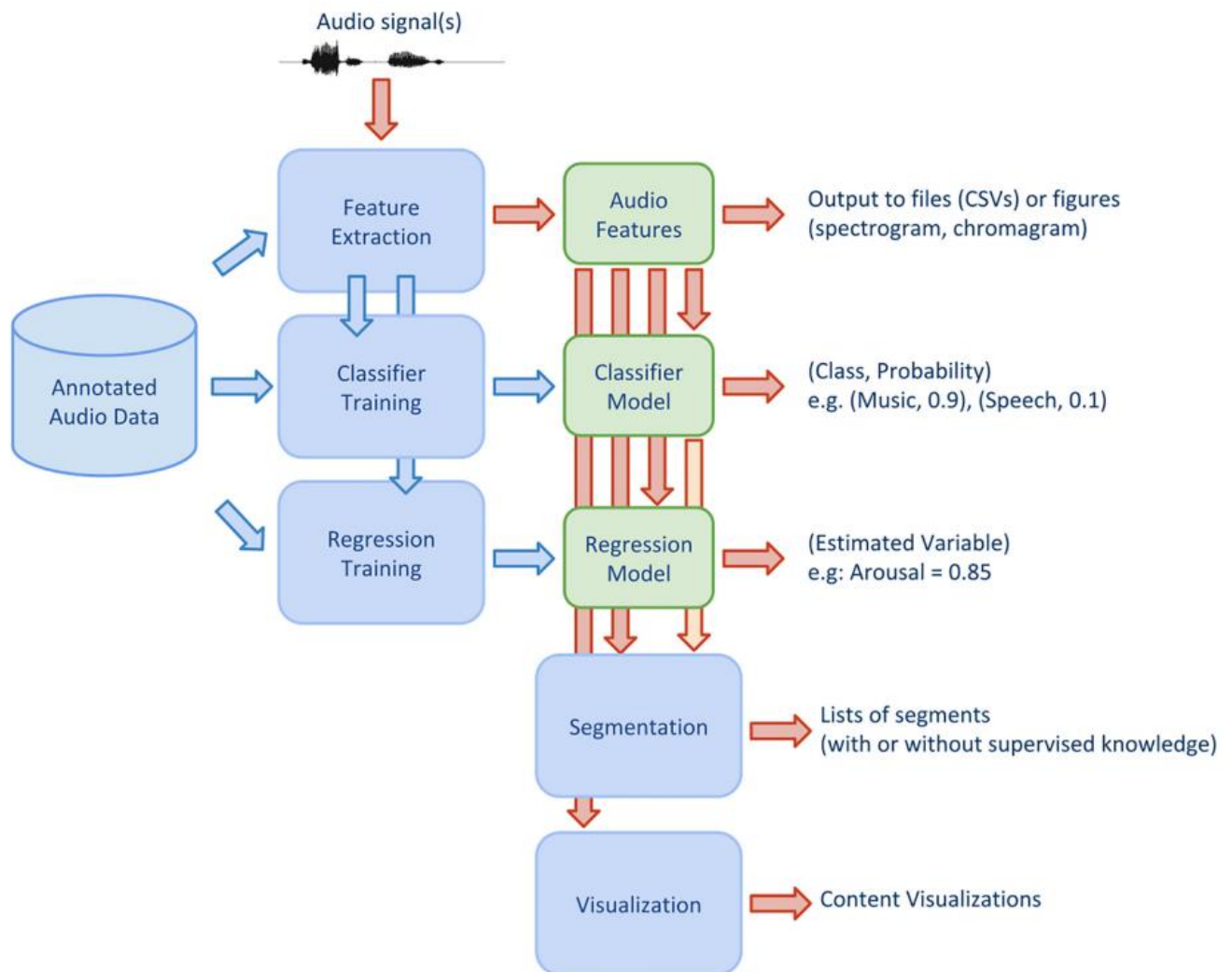


Figura 2-9. Diagrama general de pyAudioAnalysis [32].

- **Extracción de características:** pyAudioAnalysis realiza un análisis a corto y medio plazo. A corto plazo la señal de audio se divide en *frames* y para cada *frame* se calculan 34 características, dando como resultado una secuencia de vectores de características a corto plazo de 34 elementos cada uno. Los tamaños de la ventana a corto plazo son de 20 a 100ms y se realiza usando superposición (el paso del *frame* es más corto que la longitud) o encuadre no superpuesto (paso igual a la longitud). A medio plazo, la señal de audio se divide primero en ventanas (segmentos) de 1 a 10 segundos. Para cada segmento el análisis a medio plazo se utiliza para calcular estadísticas de las características, cada segmento está representado por un conjunto de estadísticas [32].

Índice característica	Nombre	Descripción
1	Zero Crossing Rate	La tasa de cambios de signo de la señal durante la duración de una trama en particular.
2	Energía	La suma de los cuadrados de los valores de la señal, normalizada por la longitud de la trama.
3	Entropía de energía	La entropía de las energías normalizadas de subtramas. Puede interpretarse como una medida de cambios bruscos.
4	Centroide espectral	El centro de gravedad del espectro.
5	Propagación espectral	El segundo momento central del espectro.
6	Entropía espectral	Entropía de las energías espectrales normalizadas para un conjunto de subtramas.
7	Flujo espectral	La diferencia al cuadrado entre las magnitudes normalizadas de los espectros de las dos tramas sucesivas.
8	Caída espectral	La frecuencia por debajo de la cual se concentra el 90% de la distribución de magnitud del espectro.
9-21	MFCCs	<i>Mel Frequency Cepstral Coefficients</i> (Coeficientes Cepstrales de Frecuencia Mel).
22-33	Vector de croma	Una representación de 12 elementos de la energía espectral.
34	Desviación de croma	La desviación estándar de los 12 coeficientes de croma.

Tabla 1. Lista completa de características de audio implementadas en pyAudioAnalysis [32].

- **Clasificación de audio:** la biblioteca proporciona funcionalidades para el entrenamiento de modelos supervisados que clasifican los segmentos. pyAudioAnalysis proporciona vectores de soporte SVM (*Support Vector Machines*, máquina de vectores de soporte) y un clasificador de vecino más cercano (*k-Nearest Neighbor*). También proporciona un procedimiento de validación cruzada para extraer el clasificador con parámetros optimizados. Finalmente se proporcionan funciones de extracción de características, incrustado en el procedimiento de clasificación, de este modo se pueden clasificar archivos de audio desconocidos o incluso grupos de archivos [32].
- **Regresión de audio:** la biblioteca estima el valor de las variables desconocidas en función del vector características respectivo, ya sea por valores reales, por clases discretas (p.ej.: ira, felicidad, etc) o por la estimación de representaciones dimensionales. pyAudioAnalysis soporta el entrenamiento de regresión SVM para asignar las características del audio a las variables, pero para ello se debe introducir primero una serie de segmentos y un archivo (CSV) donde se almacenan los valores reales de la variable de los segmentos. La función de entrenamiento de regresión también contiene un procedimiento de ajuste de parámetros para una validación cruzada similar a la clasificación. La medida del rendimiento del procedimiento de entrenamiento de regresión es el error cuadrático medio (MSE, *Mean Squared Error*) [32].
- **Segmentación de audio:** divide la señal ininterrumpida del audio en segmentos de contenido homogéneo. La biblioteca pyAudioAnalysis proporciona soluciones algorítmicas para dos tipos de segmentación diferentes: supervisado o no supervisado.
 - El supervisado contiene algoritmos que tienen un “conocimiento previo”, es decir, han sido entrenados previamente (p.ej.: esquemas de clasificación preentrenados), donde se adopta un enfoque de segmentación de tamaño fijo o

basado en HMM (*Hidden Markov Models*, Modelos Ocultos de Markov), que son transiciones entre estados según unas reglas probabilísticas obtenidas en un estado de observación. La biblioteca proporciona la capacidad de entrenar el modelo HMM proporcionando datos indicando los segmentos. Por ejemplo, en la Figura 2-10 se observa una clasificación de una señal de radio de forma binaria, música frente a voz [32].

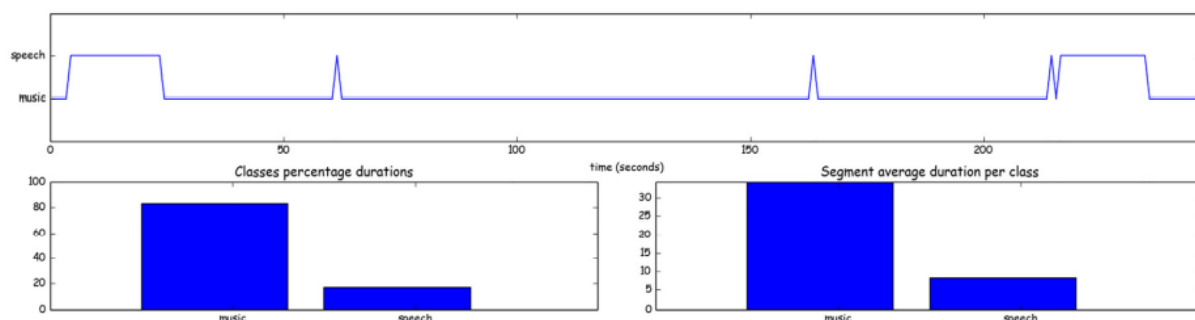


Figura 2-10. Segmentación supervisada mediante HMM. Clasificación binaria entre voz y música [32].

- La segmentación no supervisada no requiere ningún conocimiento previo sobre el contenido del audio involucrado. Las funciones más importantes son las siguientes [32]:
 - Eliminación del silencio: establece los segmentos separándolos en las zonas donde se produce un silencio mediante un modelo SVM entrenado para distinguir zonas de alta energía y baja energía del audio.
 - Registro de hablantes: se trata de una segmentación conjunta y primer paso del agrupamiento. Esta función trata de responder la pregunta “¿Quién habla y hasta cuándo?” y así establecer los segmentos dependiendo del hablante y de las características del habla.
 - *Audio thumbnailing*: se encarga de extraer la parte más representativa de un audio. La biblioteca pyAudioAnalysis se basa en encontrar el máximo de una versión filtrada de una matriz de autocorrelación y se puede visualizar como muestra la Figura 2-11.

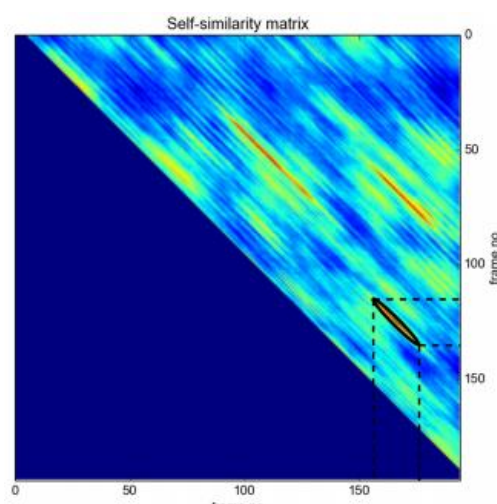


Figura 2-11. Identificación visual de las partes más representativas del audio (*thumbnail*) [32].

- **Visualización del audio:** pyAudioAnalysis ofrece un método de visualización del audio donde muestra las similitudes entre el contenido de las señales de audio analizadas y una serie de WAV almacenados (en la Figura 2-12 se muestra un ejemplo de estas

visualizaciones) e incorpora una función de conversión de MP3 a WAV que produce archivos donde saca información de las categorías de las etiquetas del MP3. La biblioteca calcula una matriz de similitud, y utiliza un análisis de componentes principales (PCA, *Principal Component Analysis*) o de discriminante lineal (LDA, *Linear Discriminant Analysis*) para proyectar la información del audio a una dimensión inferior. [32].

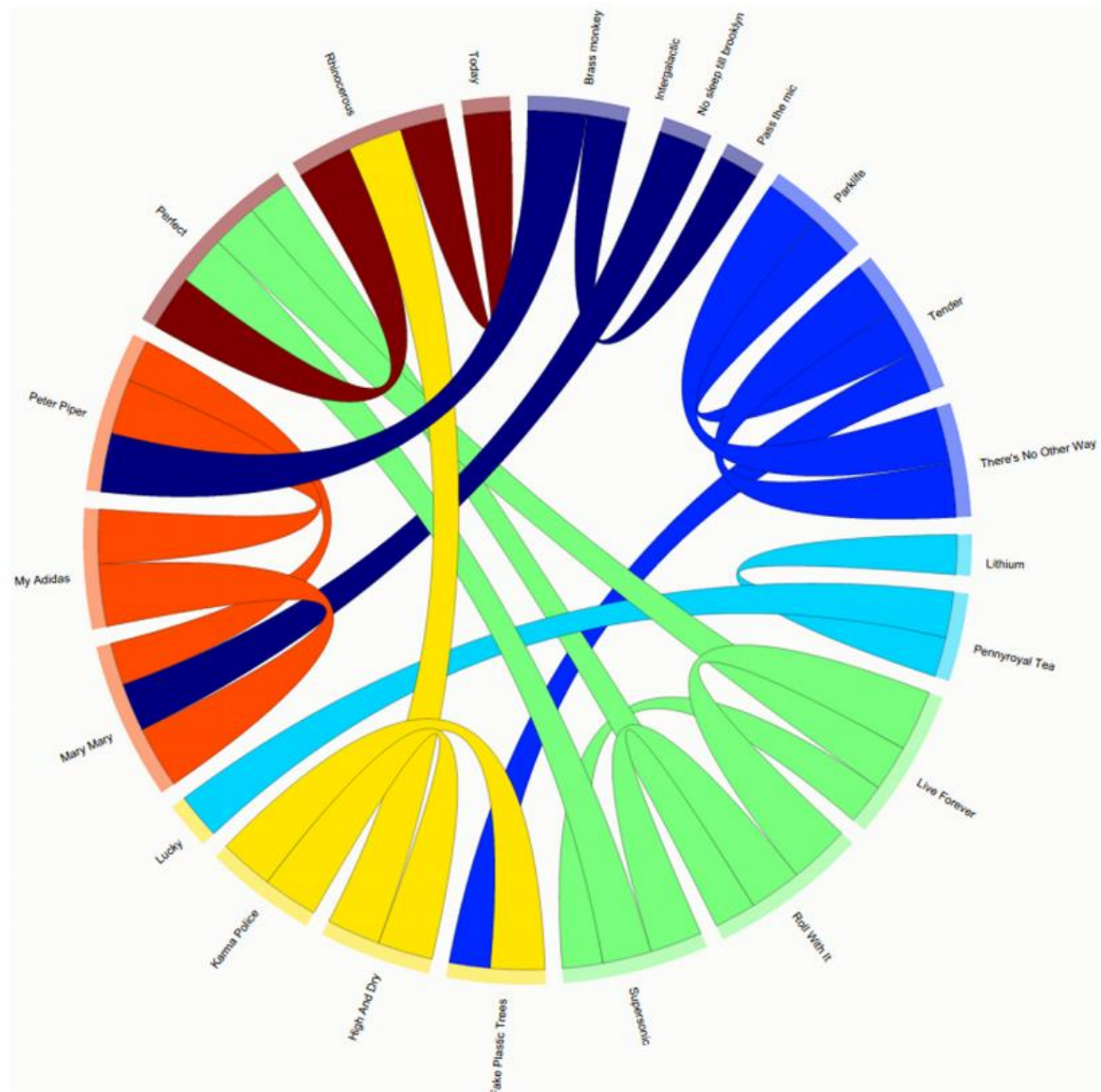


Figura 2-12. Ejemplo de visualización de similitudes del contenido de los audios analizados, en este caso canciones analizadas [32].

2.2.4 Otros datos de interés

- Este proyecto ha recibido financiación del programa de investigación e innovación Horizonte 2020 de la Unión Europea a través del acuerdo de subvención nº 643892.
- Ha sido usada en un gran número de aplicaciones de análisis de audio. Por ejemplo, en aplicaciones para recomendación de películas en función del contenido [35], para la detección de depresión [36], reconocimiento de emociones [37], estimaciones de la calidad de las ciudades en función de la contaminación acústica mediante un análisis del audio [38] y en la segmentación de audio en la puesta a punto del proyecto de BitBite [39].
- El código de esta herramienta fue implementado en Python (92,3%), Shell (4,0%) y HTML (1,9%) entre otros (1,8%).

- Licencia: Apache License 2.0, permite uso comercial, modificación, distribución y uso privado.
- Dispone de una comunidad amplia, con más de 100 dudas o errores de usuarios resueltos y 675 comentarios en la página de GitHub.
- 3700 estrellas en GitHub.

2.3 Pocketsphinx

Para un análisis certero, conviene comparar diversas herramientas que sean capaces de realizar la tarea demandada, en nuestro caso la búsqueda de palabras en un audio. Para ello, analizamos la herramienta Pocketsphinx.

2.3.1 ¿Qué es Pocketsphinx?

Pocketsphinx es una librería de Python que emplea la función de reconocimiento de voz continuo e independiente del hablante. Esta librería implementa una gran cantidad de vocabulario y se usa en su mayoría para conversión de voz a texto.

Para su funcionamiento, Pocketsphinx requiere la instalación de una librería de soporte, Sphinxbase. Ambas forman parte de CMU Sphinx junto con otras dos, una de modelos acústicos y un entrenador acústico (Sphinx4 y Sphinxtrain respectivamente) [40].

2.3.2 Creación y versiones

CMU Sphinx fue originalmente desarrollado por la Universidad de Carnegie Mellon en el año 2000, incluía una serie de programas de reconocimiento de voz y un entrenador del modelo acústico. Pocketsphinx fue creada posteriormente por Dmitry Prazdnichnov en Python, aunque el código original del proyecto Sphinx (CMU Sphinx) está programado en C y C++.

La primera versión para Python de Pocketsphinx fue publicada en febrero de 2015, después de varias actualizaciones hasta la 0.1.15, se publicó la más reciente de ellas la versión 5prealpha en 2020. La aplicación actualmente se encuentra en desarrollo y es por ello que aún se encuentra en fase alfa, lo que significa que puede contener todavía errores o tener pendiente la puesta en práctica de alguna de las funciones, pero mantiene satisfechos la mayoría de requisitos para los que está diseñada [41].

2.3.3 ¿Cómo funciona?

El sistema CMU Sphinx se basa en HMM. En los sistemas que se basan en este modelo estadístico, se representan los fonemas indicando la distribución de todas las muestras que lo componen, construyendo así el modelo acústico del fonema [42].

El sistema transforma los fonemas recibidos en vectores que representan las características de la señal, para más tarde estimar los parámetros del modelo acústico. Este proceso es conocido como el entrenamiento de los modelos acústicos. Para ello el sistema incorpora la librería Sphinxtrain.

Para el reconocimiento de voz, el sistema comienza procesando la señal igual que en el caso del entrenamiento, extrayendo los vectores de características. Este procedimiento se ejecuta en tiempo real y se compara con el modelo acústico. El resultado indica una probabilidad de que un conjunto de vectores concreto pertenezca al fonema correspondiente del modelo acústico.

De esta forma, para un reconocimiento de voz se obtiene la mejor secuencia de vectores posibles que coincida con la señal de entrada. El problema se basa en la búsqueda de grafos del sistema basado en HMM y cada grafo está compuesto por fonemas siguiendo un orden concreto.

Para construir un grafo, se necesita la aplicación de varios componentes, un diccionario que descomponga la palabra en fonemas (para esta función se necesita la aplicación Sphinxbase). Por ejemplo, la palabra en inglés “one” la descompone en los fonemas W, AX y N y construye el grafo

como indica en la Figura 2-13. También necesita la aplicación de un modelo acústico que obtenga el HMM de los fonemas, realizada por Sphinx-4 mediante un módulo de lingüística.

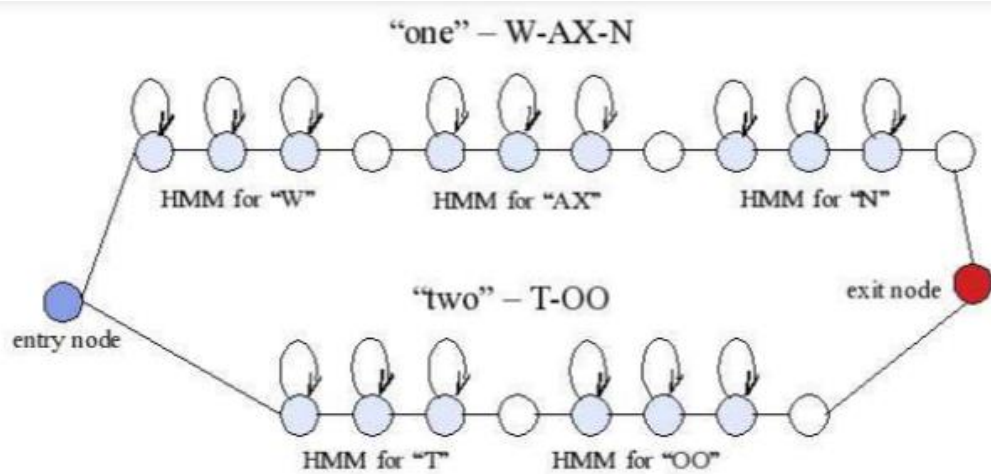


Figura 2-13. Ejemplo de creación de un grafo con HMM [42].

Una vez que el grafo está construido por completo, el sistema compara la secuencia de vectores con las diferentes rutas que componen el grafo hasta encontrar la que más se ajusta, y se genera un resultado con todas las palabras o fonemas que forman el camino para almacenar en una variable la palabra o cadena de palabras reconocidas. Las palabras reconocidas por grafos almacenadas están disponibles para la función requerida por el usuario mediante los comandos de CMU Sphinx [40]. Por ejemplo, el reconocimiento de una palabra dicha por el micrófono y su impresión por pantalla o realizar algunas funciones de entrenamiento para reconocer la palabra que dice el usuario de entre una lista que ofrece por pantalla [43].

2.3.4 Otros datos de interés

- 2800 estrellas en GitHub.
- El código de esta herramienta fue implementado en C y C++, pero adaptado a Python por SWIG (un software de adaptación de lenguajes de programación).
- Licencia BSD.
- Dispone de una comunidad amplia, con más de 1500 comentarios en la página de GitHub.

2.4 Kaldi KWS (Vosk)

Vosk es una herramienta que utiliza Kaldi para llevar a cabo la función de la búsqueda y análisis de flujos de audio. Es una herramienta moderna que se basa en un nuevo método de redes neuronales o *Deep learning*.

2.4.1 ¿Qué es Kaldi?

Kaldi es un conjunto de herramientas (*toolkit*) de código abierto para realización de labores de reconocimiento de voz creada con el objetivo de tener un código moderno y flexible que sea fácil de entender, modificar y ampliar. Las características más importantes de Kaldi son la integración con transductores de estado finito, un amplio soporte de álgebra lineal, un diseño extensible y exhaustivas pruebas. El *toolkit*, como muestra la Figura 2-14, depende de dos bibliotecas que también están disponibles de forma gratuita, OpenFst y bibliotecas de álgebra numérica [44].

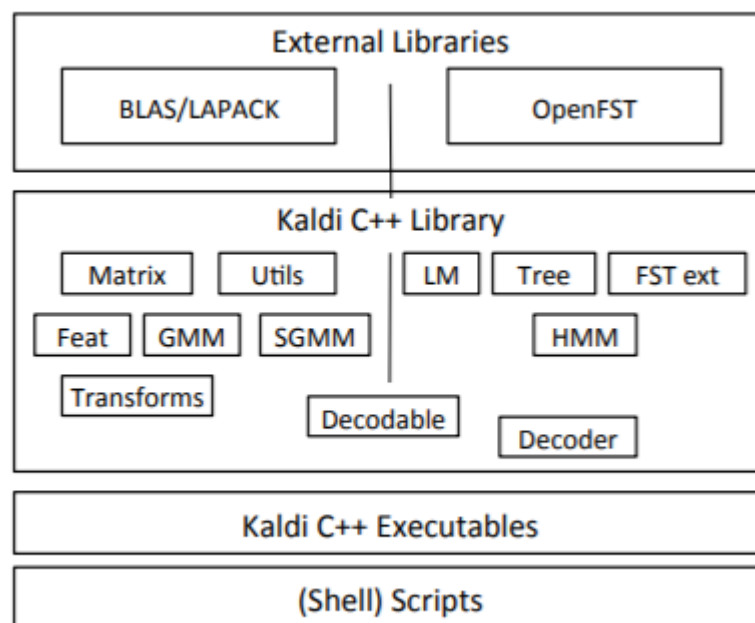


Figura 2-14. Vista simplificada de los diferentes componentes de Kaldi [44].

2.4.2 ¿Qué es Vosk?

Vosk es un conjunto de librerías (entre ellas utiliza Kaldi) para reconocimiento del habla de código abierto y sin conexión basado en el *Deep Learning* y fue introducida en 2019 para modernizar códigos como el de CMU Sphinx. Ofrece modelos de reconocimiento del habla para 17 idiomas, entre ellos español, catalán e inglés. Proporcionan una transcripción continua de gran vocabulario, un vocabulario reconfigurable e identificación del hablante [45].

Vosk ofrece reconocimiento de voz para domótica y asistentes virtuales, además puede crear subtítulos para películas y transcripción de conferencias y entrevistas.

Está implementado en C++ (45%), Java (12%) y Python (12%) entre otros. Cuenta con 1200 estrellas en GitHub y su versión más reciente es la 0.3.17, publicada en 2021.

2.4.3 Creación y versiones

Kaldi comenzó en un taller de la Universidad Johns Hopkins (Baltimore, Maryland) en 2009, bajo el título “Low Development Cost, High Quality Speech Recognition for New Languages and Domains”. Algunos asistentes de ese taller se reunieron nuevamente en la ciudad de Brno (República Checa) en 2010 para crear de una herramienta de propósito general, que en 2011 se presentaría como lo que se conoce hoy en día como Kaldi. Los creadores de Kaldi pertenecen a diferentes instituciones, como Microsoft Research (USA), Saarland University (Alemania), Centre de Recherche Informatique de Montréal (Canadá), Brno University of Technology (República Checa), Tsinghua University (China), etc.

Kaldi ofrece un amplio número de versiones:

- La primera versión fue la 5.0.0 publicada en enero de 2017, y el número de versión cambia repetidamente para cada arreglo que se implementa al código. Debido a este hecho existen versiones entre la 5.0.0 y la 5.0.62.
- La versión 5.1 incorpora importantes cambios como el requerimiento de C++11 para compilar y el uso exclusivo de la última versión de OpenFst que simplifica el código de

Kaldi, por lo que se reescribe parte del código. La versión 5.1 contiene desde la 5.1.0 hasta la 5.1.114.

- La versión 5.2 incorpora algunas actualizaciones importantes como la reformulación y simplificación de los algoritmos de entrenamiento. En esta versión se incluye desde la 5.2.0 hasta la 5.2.216.
- La versión 5.3 cambia los modelos de lenguaje y se añaden extensiones nuevas para soportar algunas operaciones. Incluye desde la versión 5.3.0 hasta la 5.3.118.
- La versión 5.4 añade código base nuevo que mejora el rendimiento en velocidad y eficiencia. Además, también se reorganiza el código nnet3. Y se actualiza hasta la versión 5.4.271.
- La última versión es la 5.5, que cambia la compatibilidad con el soporte de gramática y grafos y decodificación gramatical dinámica.

2.4.4 ¿Cómo funciona?

El módulo de búsqueda de palabras de Kaldi KWS implementa dos características claves, la indexación para una recuperación rápida de palabras clave y un proxy de palabras clave para solucionar problemas de vocabulario (OOV, *out-of-vocabulary*).

Kaldi se centra en la búsqueda de palabras clave (*keywords*), pero también puede realizar la función de búsqueda de fonemas, aunque es más compleja. Dispone de un módulo LVCSR (*Large Vocabulary Continuous Speech Recognizer*), que decodifica la búsqueda y genera las estructuras correspondientes (celosías) y un módulo KWS que hace un índice de estas celosías y busca las palabras clave del índice obtenido.

El sistema LVCSR mediante un análisis extrae 13 características acústicas dimensionales y realiza un entrenamiento acústico de máxima probabilidad, comenzando con un análisis fonético mediante HMM y terminando con un entrenamiento adaptativo del hablante (SAT, *Speaker Adaptive Training*) agrupado por estados. Todo el contenido del entrenamiento se decodifica usando un sistema gaussiano (SGMM, *Singular Gauss-Markov Model*) y se realiza otro entrenamiento de información mutua reforzada.

Las celosías generadas por el sistema se procesan utilizando técnicas de indexación y colaboran en una única estructura de transductor de factor generalizado. En esta, el tiempo de inicio y fin y la probabilidad posterior de celosía (probabilidad de que tome ese valor dependiendo del anterior) de cada *token* de palabra se almacenan como un coste. El transductor se asemeja a un índice invertido de todas las secuencias de palabras. Dada una palabra, se compone con el transductor para obtener todas las menciones de la palabra (o secuencia de palabras) clave junto con toda su información, un ID de la expresión, tiempo de inicio y fin y probabilidad posterior de celosía [46].

En el caso de que una palabra clave no esté en el vocabulario del sistema LVCSR, no aparecería en las celosías de la colección de búsqueda, aunque realmente esta palabra se haya dicho en la misma. Hay una forma de eliminar el problema, creando un sistema de subpalabras. El enfoque de Kaldi KWS es encontrar palabras del vocabulario que sean acústicamente similares para la palabra clave OOV y usarlas como palabras clave proxy en lugar de la palabra clave original. Esta técnica es complementaria a la búsqueda fonética y se puede usar también en mejorar el rendimiento en las palabras clave que aparecen en el vocabulario LVCSR [47].

2.4.5 Otros datos de interés

- 10100 estrellas en GitHub.
- El código de esta herramienta fue implementado principalmente en Shell (45%) y C++ (36%) y Python (8%).

- Licencia Apache 2.0.
- Dispone de una comunidad muy amplia, con más de 9100 comentarios en la página de GitHub.

2.5 Otras herramientas

Se incluyen en esta sección otras herramientas que por su carácter experimental o por disponer de menos documentación o popularidad se han considerado de menor interés.

2.5.1 pyAudioProcessing

Es una librería de Python con la función de procesar datos de audio con el objetivo de crear modelos de aprendizaje automático y clasificación de audios. La tarea de clasificación de audio la realiza usando pequeños conjuntos de datos de 50 muestras por clase y se obtiene la clasificación mediante una combinación de MFCC (*Mel Frequency Cepstral Coefficients*) y GFCC (*Gammatone Frequency Cepstral Coefficients*) [48]. Una aplicación de esta biblioteca puede ser clasificar el audio en tres categorías: música, voz y el sonido de pájaros, como observamos en la Figura 2-15.

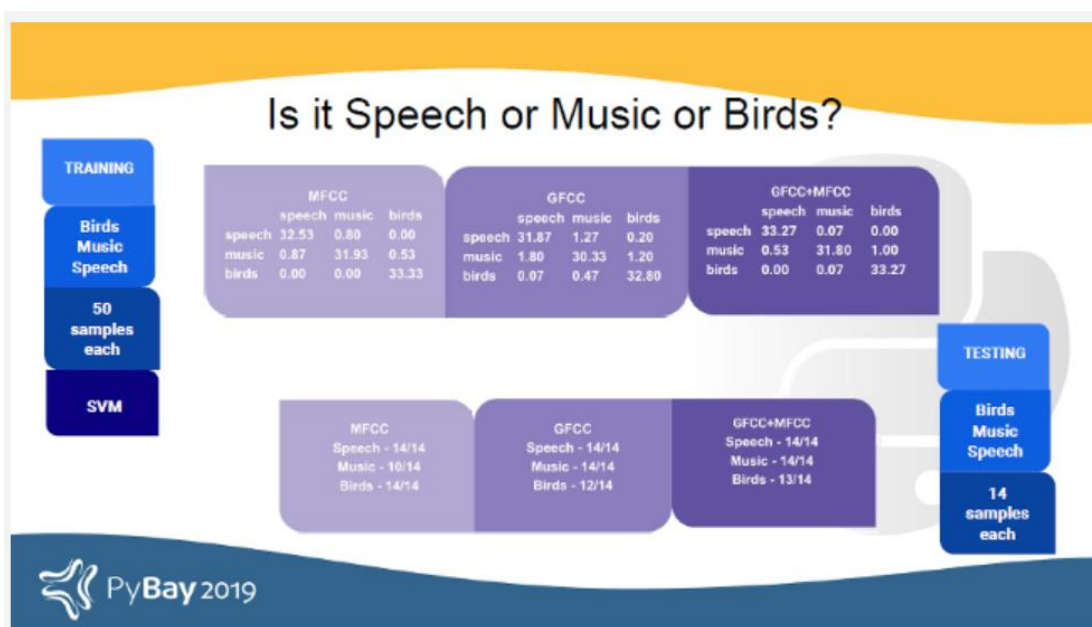


Figura 2-15. Ejemplo de clasificación en voz, sonido de pájaros o música con pyAudioProcessing [48].

También es capaz de clasificar la música en géneros dependiendo de las características extraídas del audio como se muestra en la Figura 2-16.

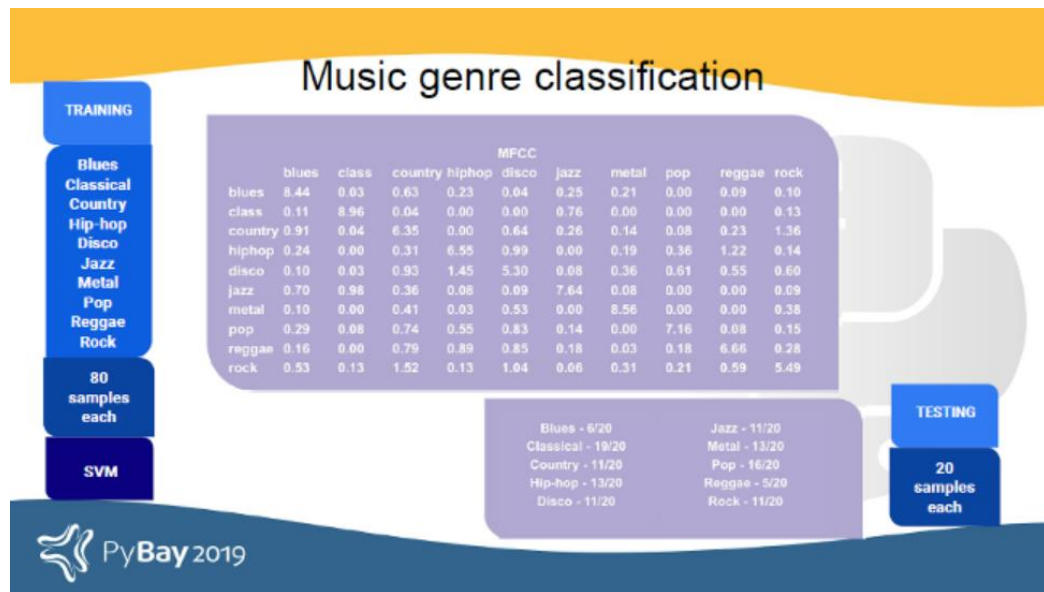


Figura 2-16. Ejemplo de clasificación de géneros musicales mediante MFCC [48].

De la Figura 2-16 también podemos observar que el rendimiento varía en gran medida dependiendo de las categorías, pero se le puede aplicar varios procesos de entrenamiento. La biblioteca extrae comprobaciones de calidad e incorpora funciones relacionadas con el ritmo y otros aspectos musicales además de otras funciones importantes como transcripción de audio a texto. Es fácil de combinar con otras librerías para formar un buen sistema de procesamiento de audio.

pyAudioProcessing está implementado en Python y dispone de una comunidad pequeña: tiene 52 estrellas en GitHub y no llega a 10 comentarios.

2.5.2 Flashlight

Flashlight es una aplicación ASR (*Automatic Speech Recognition*, Reconocimiento automático de voz) que anteriormente recibía el nombre de wav2letter. Proporciona capacidades de formación y deducción para sistemas de reconocimiento de voz de extremo a extremo. La arquitectura del sistema contiene implementaciones muy actualizadas del dominio del habla. Existen modelos de Flashlight ya entrenados pero se pueden entrenar otros para ser utilizados con la librería [49].

Flashlight es una librería publicada por Facebook Research, con una comunidad grande con multitud de variantes y tutoriales. Tiene 1400 estrellas y aproximadamente 1000 comentarios en GitHub.

2.5.3 Lingvo

Lingvo es un *framework* para construir redes neuronales de Tensorflow, particularmente modelos de secuencia [50]. Un modelo secuencia a secuencia (*sequence-to-sequence*) mapea secuencias de entrada de longitud arbitraria a secuencias de salida de longitud también arbitraria. Algunos de los usos más comunes para estos sistemas son traducciones automáticas, reconocimiento de voz y resúmenes de texto.

Lingvo, necesita de un código de entrenamiento para su sistema. A través de la herramienta Codelab se proporciona código de ejemplo sobre cómo llevar a cabo ese proceso de entrenamiento. Codelab genera datos de entrada para entrenar un modelo secuencia a secuencia en Lingvo, especifica y configura los modelos mediante una adaptación de una arquitectura de modelo preexistente para la traducción automática y utiliza un modelo entrenado para obtener deducciones [51].

Tiene una comunidad muy amplia, ya que es un sistema usado por grandes empresas como Google. Cuenta con más de 2200 estrellas en GitHub y más de 2500 comentarios. Está implementado en Python (90%) y C++ (7%).

2.6 Comparación de herramientas

En este apartado se procede a comparar las herramientas estudiadas que cumplen con el objetivo de este trabajo, para la elección de una de ellas para la fase de desarrollo. Las herramientas que aparecen en la sección de “otras herramientas” no aparecerán en la tabla debido a su carácter experimental o por disponer de escasa documentación o popularidad, por lo tanto, solo las herramientas de gran interés serán comparadas.

La tabla se basa en comparar las siguientes características:

- Comunidad: si existe un número grande de usuarios del código, si existen foros para responder preguntas o si existen páginas web con resolución de problemas frecuentes. Los valores que aparecerán en la tabla son: muy grande (10000 < estrellas en GitHub), grande (10000 > estrellas en GitHub >1000), pequeña (1000 > estrellas en GitHub).
- Idiomas soportados: si la herramienta es capaz de analizar flujos de audio en español o en inglés. En caso de herramientas de *Search by sound* el idioma no es importante porque compara directamente el audio.
- Código: si son herramientas de código abierto (gratuitas) o comerciales (de pago).
- Posibilidad de trabajar sobre un flujo de audio: si la herramienta es capaz de trabajar en un audio en tiempo real o si trabaja con archivos de audio almacenados. Los valores de la tabla serán SI/NO.
- Año de creación/actualización: indica el año en que fue creado y el año en que salió su última versión para comprobar si el código es moderno o más antiguo.

	Comunidad	Idiomas soportados (español o inglés)	Código comercial / abierto	Año de creación / actualización	Posibilidad de trabajar sobre un flujo de audio
Soundfingerprinting	Pequeña	Cualquiera (<i>Search by sound</i>)	Abierto	2011 / 2021	Sí
pyAudioAnalysis	Grande	Inglés	Abierto	2016 / 2021	Sí
Vosk (Kaldi)	Muy grande	Español e inglés	Abierto	2019 / 2021	Sí
Pocketsphinx	Grande	Español e inglés	Abierto	2013 / 2020	Sí

Tabla 2. Tabla comparativa de las herramientas estudiadas.

La herramienta elegida para un análisis más profundo será Vosk, ya que dispone de la comunidad más amplia, además también es la herramienta más moderna, de código abierto y con posibilidad de trabajar sobre un flujo en tiempo real en castellano.

3 DESARROLLO

En este capítulo se realiza el estudio y la instalación de la herramienta elegida en el capítulo anterior, en este caso Vosk.

3.1 Instalación en Python

Esta herramienta se puede instalar en Linux (x86_64 o arm64), Raspbian (Raspberry Pi) OSX y Windows [52]. Los únicos requerimientos son los siguientes:

- Versión de Python: de la 3.5 a 3.9 (Linux), desde la 3.6 a 3.7 (ARM), de la 3.8 a 3.9 en OSX y entre la 3.5 y 3.9 para Windows 64 bits.
- Versión de pip: 19.0 o versiones posteriores.

En este trabajo procederemos a la instalación en Linux. En particular, todas las pruebas e instalación se realizarán en un ordenador portátil Acer Aspire A514-53 con un procesador Intel i5 y memoria RAM 12 GB, con un SO (Sistema Operativo) Ubuntu 20.04.2 LTS 64 bits. Si durante el seguimiento de la instalación surge algún problema consultar la sección: 3.2 Solución de errores en el

```
python3 -version  
pip3 -version
```

proceso de instalación. El primer paso es comprobar las versiones de Python y pip, para ello se usan los siguientes comandos:

Una vez comprobada la versión de python y pip, el siguiente paso es proceder a la instalación de la herramienta mediante el comando:

```
sudo pip3 install vosk
```

A continuación, clonamos la librería vosk-api de GitHub mediante el comando:

```
git clone https://github.com/alphacep/vosk-api
```

El siguiente paso es crear el modelo en inglés para ejecutar la primera prueba que viene en la carpeta “example”. Dentro de esta carpeta se tiene que descargar el modelo en inglés, que viene en un fichero comprimido. Los comandos son los siguientes:

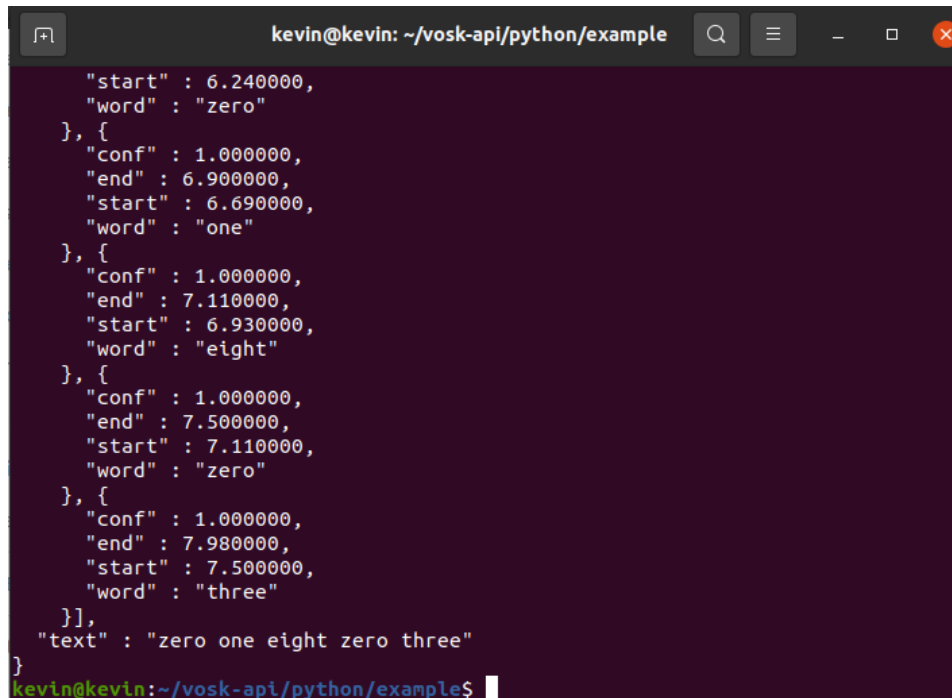
```
cd vosk-api/python/example  
wget https://alphacephei.com/kaldi/models/vosk-model-small-en-us-0.15.zip
```

El fichero comprimido (.zip) se tiene que descomprimir y cambiar el nombre a “model” para facilitar el proceso. Para ello usaremos:

```
unzip vosk-model-small-en-us-0.15.zip  
mv vosk-model-small-en-us-0.15 model
```

El modelo en inglés está creado y en una carpeta con el nombre “model”, ahora podemos realizar la primera prueba en un archivo WAV (*Wave Audio Format*) que viene incorporado en la herramienta. Para ello ejecutamos el script de Python *test_simple.py* y a continuación debemos indicar el nombre del fichero, con el que haremos la prueba. La prueba consiste en la transcripción del contenido de audio del fichero en texto, por lo tanto, deberíamos observar en pantalla el texto “one zero zero zero one” como primera frase del test, y la última que debe aparecer es “zero one eight zero three”. La Figura 3-1 muestra el resultado de la última frase del audio mencionado anteriormente.

```
python3 ./test_simple.py test.wav
```



```

"start" : 6.240000,
"word" : "zero"
}, {
  "conf" : 1.000000,
  "end" : 6.900000,
  "start" : 6.690000,
  "word" : "one"
}, {
  "conf" : 1.000000,
  "end" : 7.110000,
  "start" : 6.930000,
  "word" : "eight"
}, {
  "conf" : 1.000000,
  "end" : 7.500000,
  "start" : 7.110000,
  "word" : "zero"
}, {
  "conf" : 1.000000,
  "end" : 7.980000,
  "start" : 7.500000,
  "word" : "three"
}],
"text" : "zero one eight zero three"
}

```

Figura 3-1. Última frase del archivo test.wav

Una de las ventajas de Vosk es que existen modelos pre-entrenados en múltiples idiomas [53]. En nuestro caso deseamos instalar el modelo en español, por lo tanto, hay que borrar el modelo en inglés, descargar el archivo comprimido en español, descomprimirlo y llamar de nuevo a la carpeta “model”:

```

rm -rf model
wget https://alphacephei.com/kaldi/models/vosk-model-small-es-0.3.zip
unzip vosk-model-small-es-0.3.zip
mv vosk-model-small-es-0.3 model

```

3.1.1 Prueba con archivos de audio

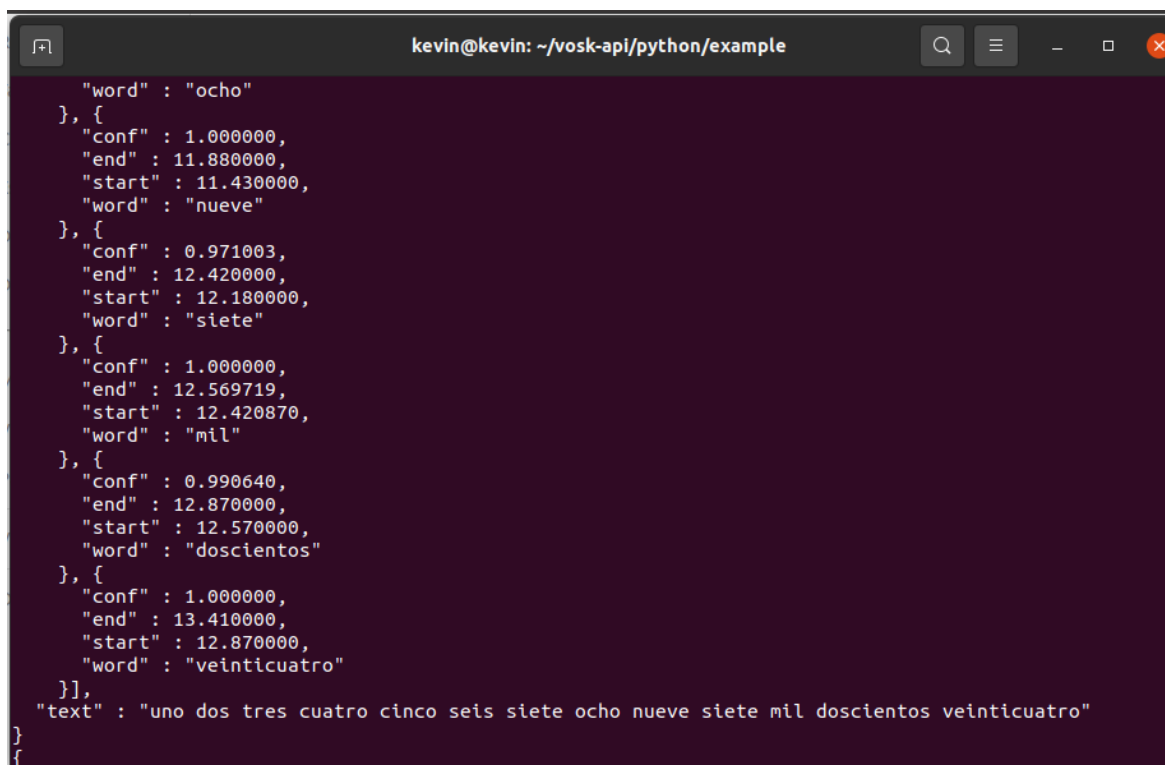
La versión de Vosk descargada no contiene ficheros de prueba para los modelos de otros idiomas distintos del inglés, por lo tanto, el usuario debe descargar un fichero de audio para la prueba del modelo. Vosk incorpora una función de conversión de archivo MP3 (*MPEG-2 Audio Layer III*, formato de compresión de audio digital) a WAV mediante el script *test_ffmpeg.py*, de modo que el archivo de audio para la prueba puede ser también un MP3. Para ello necesitamos tener instalada la aplicación ffmpeg [54]. Supongamos que descargamos un MP3 en español que se llame ejemplo.mp3. Para convertir el contenido del MP3 a texto necesitamos mover el audio de la carpeta de descargas a la carpeta “example”:

```

sudo apt-get install ffmpeg
cd Descargas
cp ejemplo.mp3 ~vosk-api/python/example
cd ~/vosk-api/python/example
python3 test_ffmpeg.py ejemplo.mp3

```

El resultado que obtenemos es una representación palabra a palabra (“partial”), y al final un resultado de la oración entera (“text”). La aplicación va construyendo las diferentes oraciones y las muestra, primero palabra por palabra y después de forma seguida. A continuación, la Figura 3-2 muestra el resultado de la transcripción de un audio de ejemplo grabado personalmente que dice: “1 2 3 4 5 6 7 8 9 7224”.



```

"word" : "ocho"
}, {
  "conf" : 1.000000,
  "end" : 11.880000,
  "start" : 11.430000,
  "word" : "nueve"
}, {
  "conf" : 0.971003,
  "end" : 12.420000,
  "start" : 12.180000,
  "word" : "siete"
}, {
  "conf" : 1.000000,
  "end" : 12.569719,
  "start" : 12.420870,
  "word" : "mil"
}, {
  "conf" : 0.990640,
  "end" : 12.870000,
  "start" : 12.570000,
  "word" : "doscientos"
}, {
  "conf" : 1.000000,
  "end" : 13.410000,
  "start" : 12.870000,
  "word" : "veinticuatro"
}],
"text" : "uno dos tres cuatro cinco seis siete ocho nueve siete mil doscientos veinticuatro"
}

```

Figura 3-2. Captura de pantalla del terminal de Ubuntu que muestra en la parte superior las palabras detectadas y sus características y debajo la oración completa.

3.1.2 Prueba con micrófono

Además de la prueba de transcripción del contenido de audio de un fichero MP3, también permite la transcripción en tiempo real de la voz por micrófono. Para ello la aplicación incorpora un script de Python que lo permite, *test_microphone.py*. Para poder realizar la prueba, la aplicación requiere el paquete de “PyAudio”, de modo que el primer paso será instalar el paquete necesario:

```
pip3 install pyaudio
```

Una vez instalados los paquetes necesarios, se puede ejecutar ya el script con la prueba del micrófono. El resultado es una presentación en pantalla igual que la mostrada en la Figura 3-2 pero que representa en tiempo real lo que se está diciendo a través del micrófono. Para dejar de transcribir es necesario presionar las teclas Ctrl + C. Para llevar a cabo la prueba de micrófono se usa el siguiente comando:

```
python3 test_microphone.py
```

La Figura 3-3 muestra la transcripción de la frase “este es un ejemplo” dicha por el micrófono y posteriormente la detención de transcripción mediante Ctrl + C. Podemos observar que en la prueba de micrófono en la sección de “partial” construye las frases, en caso de ser muy largas las construye palabra a palabra.

```
{
  "partial" : ""
}
{
  "partial" : ""
}
{
  "partial" : "este es un ejemplo"
}
{
  "partial" : "este es un ejemplo"
}
{
  "partial" : "este es un ejemplo"
}
{
  "partial" : "este es un ejemplo"
}
{
  "partial" : "este es un ejemplo"
}
{
  "partial" : "este es un ejemplo"
}
{
  "partial" : "este es un ejemplo"
}
^C
Done
```

Figura 3-3. Transcripción mediante test_microphone.py de la frase “este es un ejemplo”.

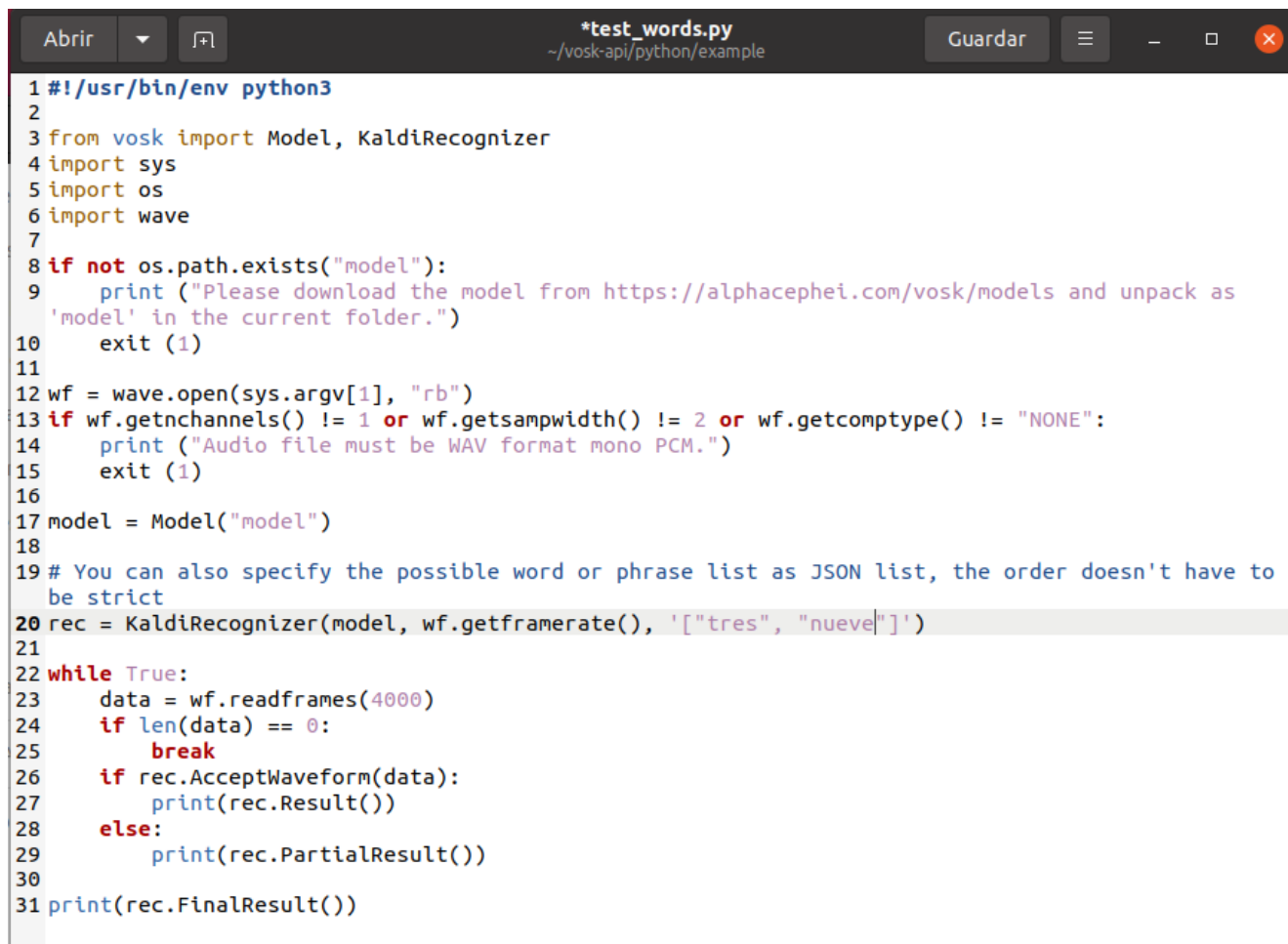
3.1.3 Prueba de búsqueda de una palabra

La librería Vosk de Python incorpora también un script de nombre *test_words.py*, que sirve para buscar una o más palabras en el audio indicado (en formato WAV mono). Para realizar la prueba, primero debemos abrir el código del script *test_words.py*, después localizamos en la línea 20 del código, donde aparece:

```
“rec=KaldiRecognizer(model,wf.getframerate(),’[“””]’)
```

Tal como muestra la Figura 3-4 introducimos la palabra o palabras que queramos buscar entre las comillas. Se puede introducir más de una separándolas con una coma y cada una entre comillas. En este caso la prueba se realizará con el mismo audio que la prueba de la sección 3.1.1, pero en formato

WAV, ejemplo.wav. El contenido del archivo era “1 2 3 4 5 6 7 8 9 7224”, y queremos detectar si el audio contiene la palabra “tres” y “nueve”.



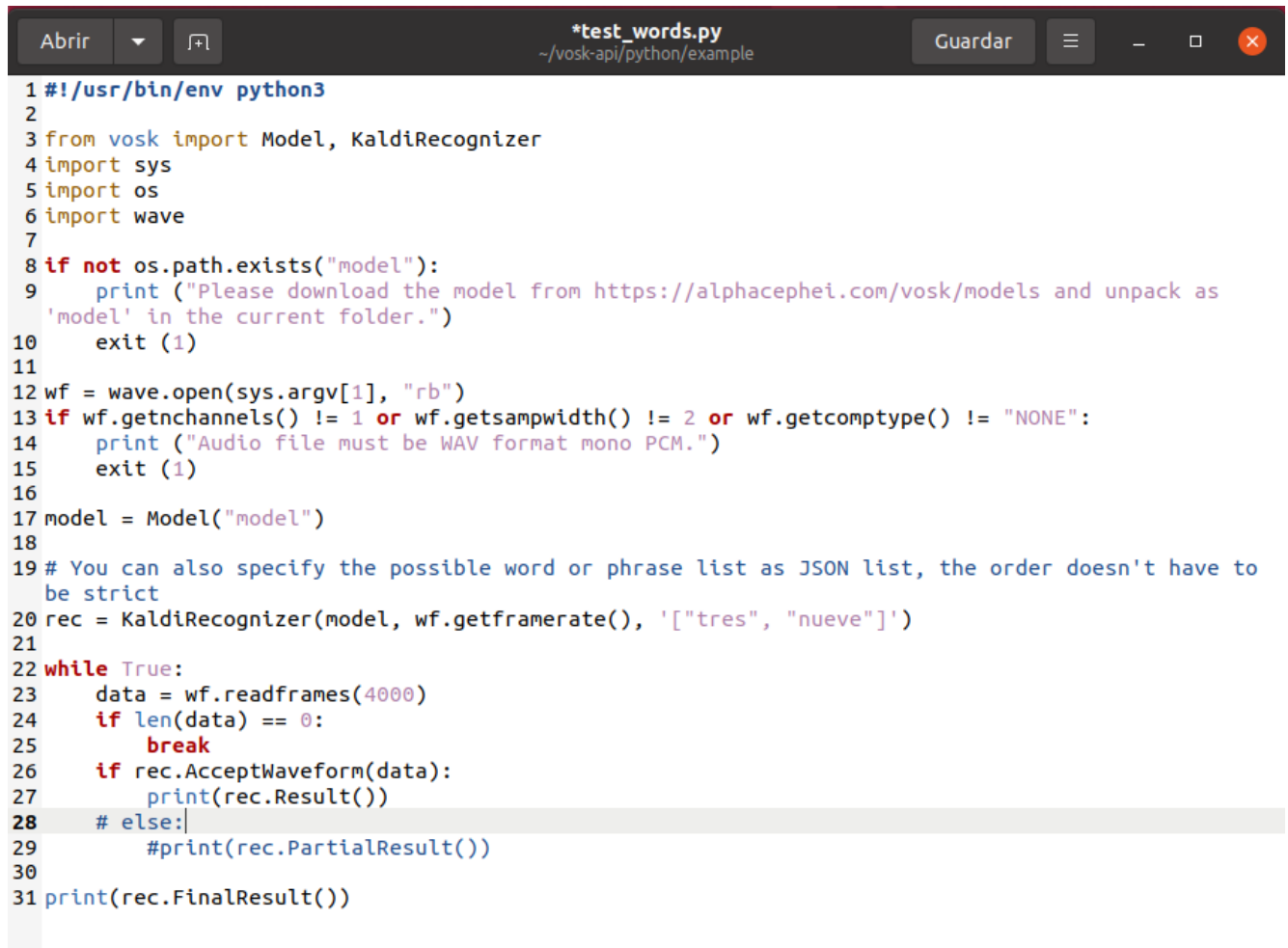
```

1#!/usr/bin/env python3
2
3from vosk import Model, KaldiRecognizer
4import sys
5import os
6import wave
7
8if not os.path.exists("model"):
9    print ("Please download the model from https://alphacephei.com/vosk/models and unpack as
'model' in the current folder.")
10    exit (1)
11
12wf = wave.open(sys.argv[1], "rb")
13if wf.getnchannels() != 1 or wf.getsampwidth() != 2 or wf.getcomptype() != "NONE":
14    print ("Audio file must be WAV format mono PCM.")
15    exit (1)
16
17model = Model("model")
18
19# You can also specify the possible word or phrase list as JSON list, the order doesn't have to
be strict
20rec = KaldiRecognizer(model, wf.getframerate(), ["tres", "nueve"])
21
22while True:
23    data = wf.readframes(4000)
24    if len(data) == 0:
25        break
26    if rec.AcceptWaveform(data):
27        print(rec.Result())
28    else:
29        print(rec.PartialResult())
30
31print(rec.FinalResult())

```

Figura 3-4. Código del script test_words.py con la línea 20 señalada y con las palabras “tres” y “nueve” escritas para buscarlas.

Para evitar que se muestren en pantalla los resultados parciales a modo de tener un resultado más claro, podemos quitar las líneas de código que lo imprimen por pantalla, o para mantenerlas en el código pero que no aparezcan por pantalla podemos ponerlas como un comentario. La Figura 3-5 muestra el código con la función de mostrar los resultados parciales marcados como comentario:



```

1 #!/usr/bin/env python3
2
3 from vosk import Model, KaldiRecognizer
4 import sys
5 import os
6 import wave
7
8 if not os.path.exists("model"):
9     print ("Please download the model from https://alphacephei.com/vosk/models and unpack as
10         'model' in the current folder.")
11     exit (1)
12
13 wf = wave.open(sys.argv[1], "rb")
14 if wf.getnchannels() != 1 or wf.getsampwidth() != 2 or wf.getcomptype() != "NONE":
15     print ("Audio file must be WAV format mono PCM.")
16     exit (1)
17
18 model = Model("model")
19
20 # You can also specify the possible word or phrase list as JSON list, the order doesn't have to
21 # be strict
22 rec = KaldiRecognizer(model, wf.getframerate(), '["tres", "nueve"]')
23
24 while True:
25     data = wf.readframes(4000)
26     if len(data) == 0:
27         break
28     if rec.AcceptWaveform(data):
29         print(rec.Result())
30     # else: | #print(rec.PartialResult())
31
32 print(rec.FinalResult())

```

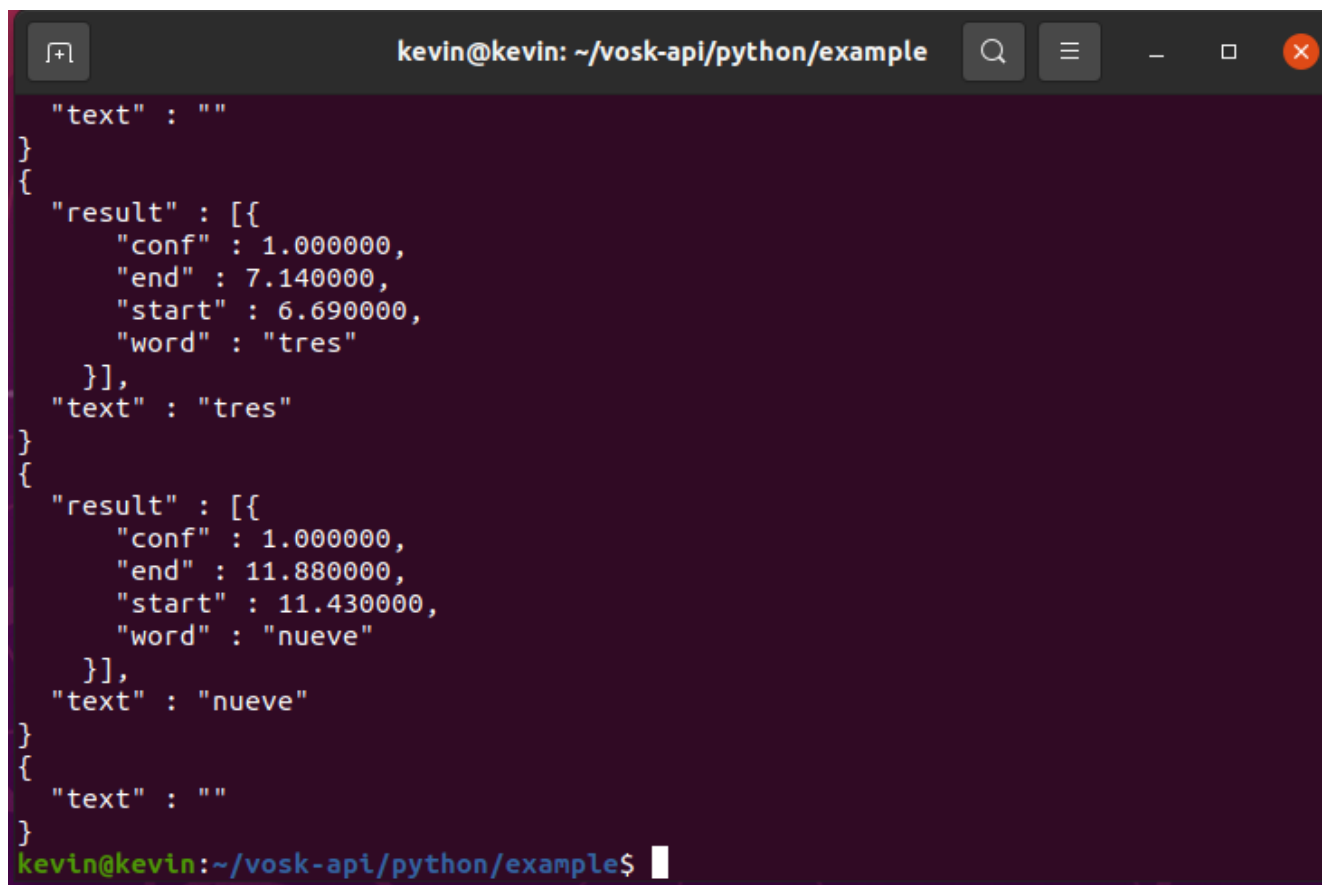
Figura 3-5. Código de test_words.py comentando la funcionalidad de mostrar por pantalla los resultados parciales.

Una vez introducidas las palabras deseadas, ejecutamos el script en el terminal mediante el siguiente comando:

```
python3 test_words.py ejemplo.wav
```

El resultado es el que se muestra en la Figura 3-6, donde se observa la detección de la palabra “tres” a los 6,69 segundos de audio y la palabra “nueve” a los 11,43 segundos de audio. La información que indica cada resultado de la búsqueda de las palabras es:

- La confianza (entre 0 y 1): es la seguridad con la que se ha detectado la palabra.
- El principio de palabra (“start”): es cuando empieza la mención a la palabra en segundos.
- El final de la palabra (“end”): es el tiempo de audio en segundos donde se acaba la palabra

A terminal window with a dark background and light-colored text. The window title is 'kevin@kevin: ~/vosk-api/python/example'. It displays the JSON output of a speech recognition process. The output consists of three JSON objects. The first object has 'text' as an empty string. The second object has 'result' as an array containing one object with 'conf' (1.000000), 'end' (7.140000), 'start' (6.690000), and 'word' ('tres'). The third object has 'text' as 'tres'. The fourth object has 'result' as an array containing one object with 'conf' (1.000000), 'end' (11.880000), 'start' (11.430000), and 'word' ('nueve'). The fifth object has 'text' as 'nueve'. The sixth object has 'result' as an array containing one object with 'conf' (1.000000), 'end' (11.880000), 'start' (11.430000), and 'word' ('nueve'). The seventh object has 'text' as 'nueve'. The eighth object has 'result' as an array containing one object with 'conf' (1.000000), 'end' (11.880000), 'start' (11.430000), and 'word' ('nueve'). The ninth object has 'text' as 'nueve'. The tenth object has 'result' as an array containing one object with 'conf' (1.000000), 'end' (11.880000), 'start' (11.430000), and 'word' ('nueve'). The eleventh object has 'text' as 'nueve'. The twelfth object has 'result' as an array containing one object with 'conf' (1.000000), 'end' (11.880000), 'start' (11.430000), and 'word' ('nueve'). The thirteenth object has 'text' as 'nueve'. The fourteenth object has 'result' as an array containing one object with 'conf' (1.000000), 'end' (11.880000), 'start' (11.430000), and 'word' ('nueve'). The fifteenth object has 'text' as 'nueve'. The prompt 'kevin@kevin:~/vosk-api/python/example\$' is visible at the bottom.

```
kevin@kevin: ~/vosk-api/python/example$  
{  
  "text" : ""  
}  
{  
  "result" : [{  
    "conf" : 1.000000,  
    "end" : 7.140000,  
    "start" : 6.690000,  
    "word" : "tres"  
  }],  
  "text" : "tres"  
}  
{  
  "result" : [{  
    "conf" : 1.000000,  
    "end" : 11.880000,  
    "start" : 11.430000,  
    "word" : "nueve"  
  }],  
  "text" : "nueve"  
}  
{  
  "text" : ""  
}  
kevin@kevin:~/vosk-api/python/example$
```

Figura 3-6. Resultado de la búsqueda de las palabras “tres” y “nueve” en el audio ejemplo.wav.

3.2 Solución de errores en el proceso de instalación

1. No tiene *python3* o *pip3* instalados. Ejecute los comandos:

```
sudo apt-get install pip3  
sudo apt-get install python3
```

2. Al intentar clonar *vosk-api* y utilizar el comando *git clone* puede que aparezca un mensaje de error, eso ocurre si no tiene *git* instalado, para ello ejecute el comando:

```
sudo apt-get install git
```

3. Al realizar la prueba de micrófono puede aparecer un error del tipo “not found sounddevice”, lo que significa que hay que instalar un paquete llamado *sounddevice* para la realización de la prueba. Para instalarlo ejecutaremos el comando:

```
python3 -m pip3 install sounddevice
```

4. En la búsqueda de una palabra en un audio con el script *test_words.py* le aparece en pantalla el texto “Audio file must be WAV format mono PCM”. Es porque el audio seleccionado no está en formato WAV o está en formato WAV estéreo. Para ello se recomienda acudir a un convertidor online como *online_convert.com* [55]. Este convertidor permite convertir todo tipo de archivos de audio a WAV mono. Para ello hay que seleccionar el convertidor WAV (señalado en azul en la Figura 3-7) y arrastrar el archivo al rectángulo verde del conversor (señalado en verde en la Figura 3-7). Importante seleccionar más abajo la opción “Modificar canales de audio: mono” (señalado en rojo en la Figura 3-7). El archivo descargado ya tendrá el formato correcto para poder usarlo con *test_words.py*.

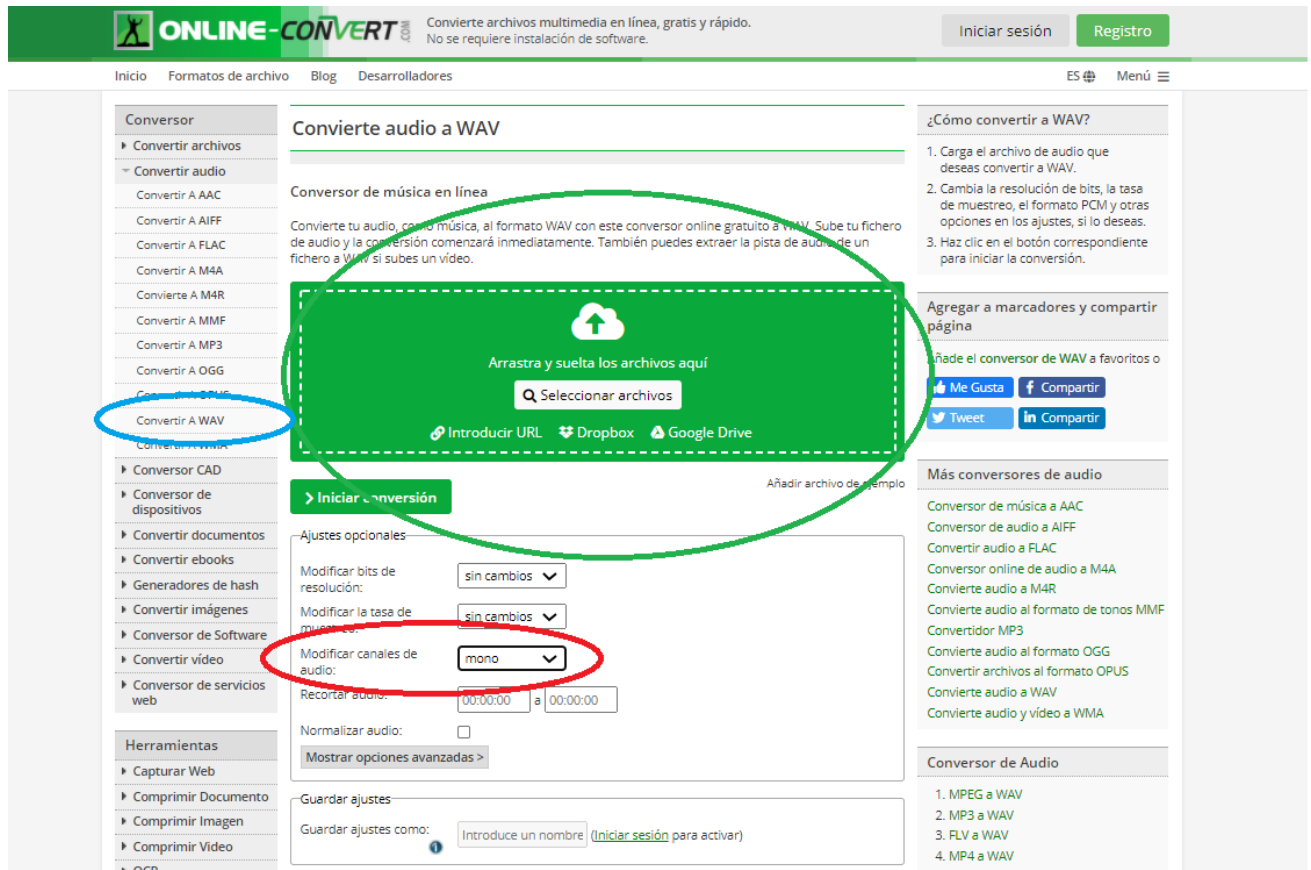


Figura 3-7. Captura de pantalla de la interfaz de *online-convert.com*.

4 PRUEBAS Y VALIDACIÓN

En este capítulo vamos a realizar diferentes pruebas para la evaluación del sistema de reconocimiento de voz y transcripción de voz a texto de Vosk. El objetivo es determinar la calidad o tasa de aciertos que alcanza la herramienta en distintas tareas (transcripción, detección de términos) y diferentes escenarios (idiomas, acentos, hablantes, género de los contenidos, etc.). Para ello el primer paso es establecer una métrica de medida del error.

4.1 Métricas de evaluación

En esta sección se exponen los dos métodos de evaluación de las pruebas, uno para evaluar la calidad de la transcripción y otro para la calidad de la detección de términos.

4.1.1 Word Error Rate

Para evaluar la transcripción de la herramienta, necesitamos primero establecer una métrica de evaluación, en este caso utilizaremos la WER (*Word Error Rate*, Proporción de errores en palabras). Este método calcula una proporción de errores en la transcripción respecto al número de palabras totales, de modo que cuanto más grande sea, más errores contiene el texto analizado [56]. La WER se calcula mediante la siguiente fórmula:

$$WER = \frac{S+B+I}{N}$$

- S: número de sustituciones.
- B: número de borrados.
- I: número de inserciones.
- N: número total de palabras que tiene el texto de referencia.

4.1.2 Precisión y recuperación

Es un método usado en recuperación de información para evaluar las búsquedas de las menciones a términos. Consiste en dos conceptos: recuperación y precisión [57].

- Precisión: es la fracción de todas las instancias relevantes dividido entre las instancias obtenidas.
- Recuperación: es la fracción de instancias relevantes que se han obtenido sobre la cantidad total de instancias relevantes.

$$\text{precisión} = \frac{n^{\circ} \text{ de menciones correctas detectadas}}{n^{\circ} \text{ de menciones totales detectadas}}$$

$$\text{recuperación} = \frac{n^{\circ} \text{ de menciones correctas detectadas}}{n^{\circ} \text{ de menciones reales en el texto}}$$

4.2 Prueba de calidad del audio

En este apartado se comparan tres audios con voz. Uno de ellos con un ruido mínimo, el segundo tendrá ruido de nivel bajo (ruido ambiental) y el tercero tendrá un nivel de ruido medio-alto (ruido de ciudad con tráfico intenso). El contenido del audio va a ser el mismo en los tres casos (a excepción del ruido) y se procederá a analizar las transcripciones de los tres audios con la finalidad de determinar si el ruido de fondo influye en esta transcripción y a qué nivel.

4.2.1 Nivel de ruido mínimo

Este audio se ha generado a partir de una grabación propia, en mismas condiciones (hablante, micrófono) que las demás pruebas de ruido. Los resultados se exponen en la Tabla 3.

Contenido Original	<i>“El Sol es la estrella central del Sistema Solar. Tiene luz propia y nos da luz y calor a la Tierra y a todos los demás astros del Sistema. El Sol es la estrella fuente de toda vida en la Tierra. Sin el Sol no podríamos vivir. El Sol es mucho más grande que la Tierra, tanto como un millón trescientas mil veces, pero en realidad es una estrella pequeña. Casi todas las estrellas más brillantes que vemos por la noche en el cielo son más grandes que el Sol, pero al encontrarse tan alejadas de nosotros se perciben como simples puntos brillantes”.</i>
Texto transcrito	“el sol es la estrella central del sistema solar tiene luz propia y nos da luz y calor a la tierra y a todos los demás nuestros del sistema el sol es la estrella fuente de toda vida en la tierra sin el sol no podríamos vivir el sol es mucho más grande” “que la tierra tanto como un millón trescientos mil veces pero en realidad es una estrella pequeña casi todas las estrellas más brillante que vemos por la noche en el cielo son más grandes que el sol pero al encontrarse tan alejadas de nosotros se perciben como simples puntos brillante”
Análisis de las diferencias	“el sol es la estrella central del sistema solar tiene luz propia y nos da luz y calor a la tierra y a todos los demás nuestros (astros) del sistema el sol es la estrella fuente de toda vida en la tierra sin el sol no podríamos vivir el sol es mucho más grande” “que la tierra tanto como un millón trescientos (trescientas) mil veces pero en realidad es una estrella pequeña casi todas las estrellas más brillante (brillantes) que vemos por la noche en el cielo son más grandes que el sol pero al encontrarse tan alejadas de nosotros se perciben como simples puntos brillante (brillantes)”

Tabla 3. Análisis de la transcripción del audio sin nada de ruido.

A continuación, se va a proceder a calcular la WER, para ello obtenemos los datos de la prueba realizada. Sobre un total de 103 palabras del texto original, se han detectado un total de 4 substituciones, y ninguna eliminación ni inserción. La WER se calcula de la siguiente manera:

$$WER = \frac{S + B + I}{N} = \frac{4 + 0 + 0}{103} = 3,88\%$$

La WER indica muy buenos resultados en la transcripción, con una proporción de errores muy baja.

4.2.2 Nivel de ruido bajo

Este audio ha sido grabado en las mismas condiciones, pero con un altavoz a un metro del micrófono añadiendo ruidos naturales (lluvia y viento) a un volumen medio. Los resultados se exponen en la Tabla 4.

Contenido Original	<i>“El Sol es la estrella central del Sistema Solar. Tiene luz propia y nos da luz y calor a la Tierra y a todos los demás astros del Sistema. El Sol es la estrella fuente de toda vida en la Tierra. Sin el Sol no podríamos vivir. El Sol es mucho más grande que la Tierra, tanto como un millón trescientas mil veces, pero en realidad es una estrella pequeña. Casi todas las estrellas más brillantes que vemos por la noche en el cielo son más grandes que el Sol, pero al encontrarse tan alejadas de nosotros se perciben como simples puntos brillantes”.</i>
Texto transcrito	“el sol es la estrella central del sistema solar tienen el propia y nos da luz y calor a la tierra y a todos los demás actos del sistema el sol es la estrella fuente de toda vida en la tierra sin el sol lo podríamos vivir el sol es mucho más grande que la tierra tanto como” “un millón trescientos mil veces pero en realidad es una estrella pequeño casi todas las estrellas más brillantes que vemos por la noche en el cielo son más grandes que el sol pero al encontrarse están alejadas de nosotros se perciben como siempre puntos brillante”
Análisis de las diferencias	“el sol es la estrella central del sistema solar tienen (tiene) el (luz) propia y nos da luz y calor a la tierra y a todos los demás actos (astros) del sistema el sol es la estrella fuente de toda vida en la tierra sin el sol lo (no) podríamos vivir el sol es mucho más grande que la tierra tanto como” “un millón trescientos (trescientas) mil veces pero en realidad es una estrella pequeño (pequeña) casi todas las estrellas más brillantes que vemos por la noche en el cielo son más grandes que el sol pero al encontrarse están (tan) alejadas de nosotros se perciben como siempre (simples) puntos brillante (brillantes)”

Tabla 4. Análisis de la transcripción del audio con ruido ambiental.

Sobre un total de 103 palabras del texto original, se han detectado un total de 9 substituciones, y ninguna eliminación ni inserción. La WER se calcula como:

$$WER = \frac{S + B + I}{N} = \frac{9 + 0 + 0}{103} = 8,74\%$$

La WER muestra un aumento de ($\approx 5\%$) respecto a la prueba anterior.

4.2.3 Nivel de ruido medio-alto

Este audio ha sido obtenido mediante grabación propia en las mismas condiciones que las pruebas anteriores con la salvedad de un altavoz a un metro añadiendo ruido de ciudad (tráfico intenso) a un volumen alto. Los resultados se exponen en la Tabla 5.

Contenido Original	<i>“El Sol es la estrella central del Sistema Solar. Tiene luz propia y nos da luz y calor a la Tierra y a todos los demás astros del Sistema. El Sol es la estrella fuente de toda vida en la Tierra. Sin el Sol no podríamos vivir. El Sol es mucho más grande que la Tierra, tanto como un millón trescientas mil veces, pero en realidad es una estrella pequeña. Casi todas las estrellas más brillantes que vemos por la noche en el cielo son más grandes que el Sol, pero al encontrarse tan alejadas de nosotros se perciben como simples puntos brillantes”.</i>
Texto transcrito	“el sol es la estrella central del sistema solar tiene luz propia y no esta luz y calor a las diez y a todos los demás gastos del sistema el sol es la estrella fuente de toda vida en la tierra sin el sol no podríamos vivir el sol es mucho más grande” “que la tierra tanto como un millón trescientos mil veces pero en realidad es una estrella pequeña casi todas las estrellas más brillantes que por coche en el suelo son más grandes que el sol pero al encontrarse van alejadas de nosotros se perciben como simples puntos brillante”
Análisis de las diferencias	“el sol es la estrella central del sistema solar tiene luz propia y no (nos) esta (da) luz y calor a las diez (tierra) y a todos los demás gastos (astros) del sistema el sol es la estrella fuente de toda vida en la tierra sin el sol no podríamos vivir el sol es mucho más grande” “que la tierra tanto como un millón trescientos (trescientas) mil veces pero en realidad es una estrella pequeña casi todas las estrellas más brillantes que (vemos) por (la) coche (noche) en el suelo (cielo) son más grandes que el sol pero al encontrarse van (tan) alejadas de nosotros se perciben como simples puntos brillante (brillantes)”

Tabla 5. Análisis de la transcripción del audio con ruido de tráfico intenso.

Sobre un total de 103 palabras del texto original, se han detectado un total de 9 substituciones, dos eliminaciones y ninguna inserción. La WER se calcula como:

$$WER = \frac{S + B + I}{N} = \frac{9 + 2 + 0}{103} = 10,68\%$$

Los resultados de esta prueba muestran un aumento ($\approx 2\%$) respecto al nivel de ruido bajo.

4.2.4 Pruebas de búsqueda de la mención de términos

Vamos a proceder a buscar algunas menciones en los audios anteriores mediante el script `test_words.py`. Los resultados de la búsqueda se exponen en la Tabla 6.

Palabra	Estrella	Sol	Brillantes	Luz
Nº de apariciones en el audio	3	5	2	2
Nº de detecciones correctas en el audio con ruido mínimo	3	5	2	2
Nº detecciones totales con el audio ruido mínimo	4	8	2	2
Nº de detecciones correctas en el audio con ruido bajo	3	5	2	1
Nº de detecciones totales en el audio con ruido bajo	4	9	2	1
Nº de detecciones correctas en el audio con ruido medio-alto	3	5	2	2
Nº de detecciones totales en el audio con ruido medio-alto	4	8	2	3
Recuperación audio con ruido mínimo (%)	100	100	100	100
Recuperación audio con ruido bajo (%)	100	100	100	50,00
Recuperación audio con ruido medio-alto (%)	100	100	100	100
Precisión audio con ruido mínimo (%)	75,00	62,50	100	100
Precisión audio con ruido bajo (%)	75,00	55,56	100	100
Precisión audio con ruido medio-alto (%)	75,00	62,50	100	66,67

Tabla 6. Resultados de la búsqueda de menciones de palabras en la prueba de calidad del audio.

4.2.5 Análisis de resultados

Observamos que el ruido de fondo sí que influye en la transcripción final del audio, ya que la WER del mismo audio sin ruido es del 3% y el del audio con ruido aumenta hasta el 10%. Si necesitamos una transcripción más precisa tenemos que intentar que el audio tenga el menor ruido posible, de esta forma el número de palabras transcritas de forma errónea será menor.

En la prueba de búsqueda de palabras, se observa que las tasas de recuperación y precisión de los tres audios son muy parecidas, por lo que podemos concluir que para las palabras escogidas el nivel de ruido no afecta significativamente en la detección de las menciones y en la precisión de estas detecciones.

4.3 Prueba de contenido del audio

Se realizan pruebas con diferentes contenidos de audio. El primer audio analizado es un anuncio de televisión compuesto de una voz y música de fondo. El segundo audio es una canción, para proceder a comprobar si Vosk reconoce las palabras cuando se altera el ritmo del habla y/o se añade música de fondo.

4.3.1 Prueba de voz hablada más música

En esta prueba se va a analizar un recorte del anuncio de televisión de Cruzcampo [58]. Consiste en una voz narrando y música de guitarra de fondo pero al mismo nivel de sonido. Los resultados se exponen en la Tabla 7.

La Figura 4-1 muestra el resultado de la transcripción del audio mediante el script *test_simple.py*.

```

kevin@kevin: ~/vosk-api/python/example
{
  "conf" : 0.786140,
  "end" : 33.270000,
  "start" : 33.150000,
  "word" : "o"
}, {
  "conf" : 0.527273,
  "end" : 33.810000,
  "start" : 33.271051,
  "word" : "ministro"
}],
  "text" : "las costuras y loft platillo ese texto muchas tal y como tal cual ci
erre de la conchinchina de la línea de la concepción cajeros del supermercado ca
tedrática o ministro"
}
{
  "result" : [{
    "conf" : 0.559499,
    "end" : 35.190000,
    "start" : 35.048185,
    "word" : "no"
  }, {
    "conf" : 0.900620,
    "end" : 35.640095,
    "start" : 35.190000,

```

Figura 4-1. Transcripción del anuncio de televisión.

Contenido Original	“¿Tú sabes por qué a mí se me entendió en todo el mundo? Por el acento. Y no solo me refiero a la forma de hablar, que también. Me refiero a ese pellizco, a esa forma con la que te llenas el pecho de alegría, con la que rebañas un huevo frito, con la que te pintas el rabillo del ojo. Acento es que se te vean las costuras y los dobladillos, que se te escuche hasta el hipo. Da igual si eres de la
--------------------	---

	<i>Conchinchina o de la Línea de la Concepción, cajero del supermercado, catedrática o ministro. Manosea tus raíces que de ahí siempre salen cosas buenas”</i>
Texto transcrito	“Tú” “sabes porque mi se me entendió todo el mundo” “solo me refiero forma tabla que también” “pierre cepillito” “forma con la que te llena de alegría con la que se vaya juntos los fritos con la que que pinta el rápido” “en que te veía” “las costuras y loft platillo ese texto muchas tal y como tal cual cierre de la conchinchina de la línea de la concepción cajeros del supermercado catedrática o ministro” “no sea tus raíces que de ahí siempre salen cosas”
Análisis de las diferencias	“Tú” “sabes porque mi se me entendió todo el mundo (por el acento)” “(y no) solo me refiero (a la) forma (de) tabla (hablar) que también” “(me) pierre (refiero) (a ese) cepillito (pellizco)” “(a esa) forma con la que te llena (llenas) (el pecho) de alegría con la que se vaya (rebañas) (un) juntos (huevo) los fritos con la que que (te) pinta (pintas) el rápido (rabillo) (del ojo)” “(acento) en (es) que (se) te veía (vean)” “las costuras y loft (los) platillo (dobladillos) (que) ese (se) texto (te) muchas (escuche) tal (hasta el) y como (hipo) tal (da) cual (igual) cierre (si eres) de la Conchinchina (o) de la línea de la concepción cajeros del supermercado catedrática o ministro” “no sea (manosea) tus raíces que de ahí siempre salen cosas (buenas)”

Tabla 7. Análisis de la transcripción de un anuncio con música.

Sobre un total de 109 palabras del texto original, se han detectado un total de 22 substituciones, 23 eliminaciones y 1 inserción. La WER se calcula:

$$WER = \frac{S + B + I}{N} = \frac{22 + 23 + 1}{109} = 42,20\%$$

Los resultados muestran una gran proporción de errores en la transcripción debido a la música de fondo.

En el mismo audio se va a proceder a realizar una prueba de búsqueda de palabras. Esta prueba se llevará a cabo mediante el script *test_words.py* de la aplicación de Vosk. El resultado de la búsqueda de palabras se muestra a continuación en la Tabla 8.

Palabra	Acento	Pecho	Dobladillos	Rabillo	Me	Que
Nº de apariciones en el audio	2	1	1	1	3	8
Nº de detecciones correctas	0	1	1	1	1	5
Nº detecciones totales	1	1	1	3	5	12
Recuperación (%)	0	100	100	100	33,33	62,50
Precisión (%)	0	100	100	33,33	20	41,67

Tabla 8. Resultado de la búsqueda de palabras en el anuncio de televisión.

En los resultados podemos observar que la búsqueda de palabras funciona mejor con palabras más largas, ya que las palabras más pequeñas las confunde más fácilmente con otros sonidos parecidos. Por esa razón, las tasas de recuperación y precisión de las palabras más largas son más altas que las tasas de las palabras más cortas.

4.3.2 Prueba de voz cantada más música

Esta prueba consiste en transcribir unos versos de dos canciones, y a continuación analizar la WER. Por último, realizaremos una búsqueda de palabras en las canciones completas.

La primera canción analizada va a ser “Voy a ser Rey León”. Esta prueba la realizaremos mediante *test_simple.py*. Los resultados se exponen en la Tabla 9.

Contenido Original	<i>“Libre para ser el rey, (quítate eso de la cabeza), libre para ser el rey, (es hora de que tú y yo hablemos de verdad), no quiero escuchar a un pajarraco tan vulgar.”</i>
--------------------	---

Texto transcrito	“la canela” “es hora de que tú y yo hablaremos de verdad” “escuchar”
Análisis de las diferencias	“ la canela ” “ es hora de que tú y yo hablaremos de verdad ” “ escuchar ”

Tabla 9. Análisis de la transcripción de versos de la canción “Yo voy a ser Rey León”.

Sobre un total de 33 palabras del texto original, se han detectado un total de 2 substituciones, 20 eliminaciones y ninguna inserción. La WER se calcula:

$$WER = \frac{S + B + I}{N} = \frac{2 + 20 + 0}{33} = 66,67\%$$

Para probar de nuevo, utilizaremos la canción de Andy & Lucas “Tanto la quería”. Los resultados se exponen en la Tabla 10.

Contenido Original	<i>“Porque eres tan hermosa y a la vez tan difícil, porque la vida pasa y pasa y te quiero a mi vera, si me trataste como un juguete sucio y abandonado, si no comprendes que el amar es algo más que besarnos”</i>
Texto transcrito	“eres tan verbos haya daba es tan difícil” “por qué la vida pasa y paz te quiero a mi verga” “si me tratas de jugué sucio ya abandonado” “si no comprendes que la amar es algo más que besar no”
Análisis de las diferencias	“(porque) eres tan verbos (hermosa) haya (y) daba (a la) es (vez) tan difícil” “por qué la vida pasa y paz (pasa) (y) te quiero a mi verga (vera)” “si me tratas de (trataste) (como a un) jugué (juguete) sucio ya (y) abandonado” “si no comprendes que la (el) amar es algo más que besar no (besarnos)”

Tabla 10. Análisis de la transcripción de versos de la canción “Tanto la quería”.

Sobre un total de 42 palabras del texto original, se han detectado un total de 11 substituciones, 5 eliminaciones y ninguna inserción. La WER se calcula:

$$WER = \frac{S + B + I}{N} = \frac{11 + 5 + 0}{42} = 38,09\%$$

Podemos observar en los resultados que muestran las pruebas realizadas que la proporción de errores en la transcripción de una canción pueden ser muy variables puesto que hay una diferencia de aproximadamente un 30%.

En las dos canciones completas, se va a realizar una búsqueda de palabras. Esta prueba se llevará a cabo mediante el script *test_words.py* de la aplicación de Vosk. El resultado de la búsqueda se muestra en la Tabla 11.

Palabra	León	Ser	Pajarraco	Vocación	Amor	Tanto	Estrellas	Senderos
Nº de apariciones en el audio	6	10	1	1	4	9	4	1
Nº de detecciones correctas	0	0	0	1	1	3	3	1
Nº detecciones totales	0	2	0	1	6	8	4	1
Recuperación (%)	0	0	0	100	25	33,33	75	100
Precisión (%)	-	-	-	100	16,67	37,50	75	100

Tabla 11. Resultados de la búsqueda de palabras en canciones.

En las celdas marcadas con un guión “-“, significa que no se puede calcular la precisión ya que no existe ninguna detección, ni correcta ni falsa.

4.3.3 Análisis de resultados

Con una WER del 42,20%, podemos concluir que la música de fondo en una voz es un factor que influye de manera muy negativa a la hora de realizar transcripciones, y que el ritmo al que se dicen las palabras (si se cantan) además de la música también influye muy negativamente. Se ha podido comprobar que, en una de las pruebas de transcripción de unos versos de canción, donde se incluía música y voz cantada (alteración del ritmo) el resultado ha sido de mala calidad (WER = 66,67%).

En la búsqueda de palabras Vosk tiende a reconocer palabras más largas (a excepción de “pajarraco”), ya sea de un hablante con música de fondo o en una canción. En cambio, en la detección de menciones a palabras cortas Vosk no parece tan eficaz, ya que muestra muchos resultados, pero la mayor parte de estos son erróneos (se ve reflejado en la precisión) y además se omite gran parte de las menciones reales a dicha palabra (se ve reflejado en la recuperación).

4.4 Prueba de género del hablante

En esta prueba se comparan dos audios de mismo contenido, generados personalmente con el mismo micrófono, uno con una voz masculina y otro con una voz femenina, ambas personas procedentes del mismo lugar geográfico y recitando con la misma entonación y velocidad, para tratar de dejar como única variable la diferencia de tono de voz entre un hombre y una mujer. Se realizará la prueba para comprobar si el tono de voz influye en la transcripción de la herramienta. En este apartado, la prueba se va a realizar con un fragmento de la canción de Manolo Escobar “Mi Carro”.

4.4.1 Prueba de voz femenina

A continuación, realizamos la prueba con el fichero de voz femenina mediante el script de `Vosk test_simple.py`. Los resultados se exponen en la Tabla 12.

Contenido Original	<i>“Mi carro me lo robaron, estando de romería. Mi carro me lo robaron, anoche cuando dormía. ¿Dónde estará mi carro? Me dicen que le quitaron los clavos que relucían, creyendo que eran de oro de limpios que los tenía.”</i>
Texto transcrito	“mi carro robaron” “mi carro ni lo robaron anoche cuando dormía donde estará mi carro me dicen que le quitaron los clavos que reducía creyendo que eran de oro de limpios que lo tenía”
Análisis de las diferencias	“mi carro (me lo) robaron (estando de romería)” “mi carro ni (me) lo robaron anoche cuando dormía donde estará mi carro me dicen que le quitaron los clavos que reducía (relucían) creyendo que eran de oro de limpios que lo (los) tenía”

Tabla 12. Análisis de la transcripción del audio con hablante femenina.

Sobre un total de 39 palabras del texto original, se han detectado un total de 3 substituciones, 5 eliminaciones y ninguna inserción. La WER se calcula como:

$$WER = \frac{S + B + I}{N} = \frac{3 + 5 + 0}{39} = 20,51\%$$

La transcripción del audio con voz femenina muestra una proporción de errores de palabras del 20,51%.

4.4.2 Prueba de voz masculina

A continuación, realizamos la prueba con el fichero de voz masculina mediante el script de `Vosk test_simple.py`. Los resultados se exponen en la Tabla 13.

Contenido Original	<i>“Mi carro me lo robaron, estando de romería. Mi carro me lo robaron, anoche cuando dormía. ¿Dónde estará mi carro? Me dicen que le quitaron los clavos que relucían, creyendo que eran de oro de</i>
--------------------	---

	<i>limpios que los tenía.”</i>
Texto transcrito	“mi carro me lo robaron estando de romería mi carro me lo robaron anoche cuando dormía donde estará mi carro me dicen que le quitaron los clavos que reducía creyendo que eran de oro del limpios que los tenía”
Análisis de las diferencias	“mi carro me lo robaron estando de romería mi carro me lo robaron anoche cuando dormía donde estará mi carro me dicen que le quitaron los clavos que reducía (relucían) creyendo que eran de oro del (de) limpios que los tenía”

Tabla 13. Análisis de la transcripción del audio con hablante masculino.

Sobre un total de 39 palabras del texto original, se han detectado un total de 2 substituciones, ninguna eliminación ni inserción, lo que resulta en una WER:

$$WER = \frac{S + B + I}{N} = \frac{2 + 0 + 0}{39} = 5,12\%$$

En la prueba de voz masculina, la proporción de errores en la transcripción es 15% menor a la de voz femenina.

4.4.3 Prueba de búsqueda de menciones a términos

Vamos a proceder a buscar algunas menciones en los audios anteriores. Los resultados de la búsqueda se exponen en la Tabla 14.

Palabra	Carro	Me	Romería	Oro
Nº de apariciones en el audio	3	3	1	1
Nº de detecciones correctas (hablante femenina)	3	3	0	1
Nº de detecciones totales (hablante femenina)	3	6	2	1
Nº de detecciones correctas (hablante masculino)	3	3	1	1
Nº de detecciones totales (hablante masculino)	4	8	3	2
Recuperación audio hablante femenina(%)	100	100	0	100
Recuperación audio hablante masculino (%)	100	100	100	100
Precisión audio hablante femenina (%)	100	50	0	100
Precisión audio hablante masculino (%)	75	37,50	33,33	50

Tabla 14. Resultado de la búsqueda de menciones a términos en la prueba de género del hablante.

4.4.4 Análisis de resultados

Para comparar la eficacia de las transcripciones, se comparan usando la WER de cada una de ellas. Los resultados obtenidos son los siguientes:

- Prueba con voz femenina: 20,51%
- Prueba con voz masculina: 5,12%

La WER indica la tasa de palabras con error, de modo que la transcripción con la WER más elevada es la peor. En este caso, los audios son de exacto contenido, recitados de la misma forma.

Se puede observar en los resultados que las pruebas muestran que Vosk es capaz de transcribir de forma mucho más idónea la voz masculina que la femenina, ya que la diferencia de WER es del 15,39%. A partir de este resultado, podemos concluir que el género del hablante parece influir de manera directa en la calidad de la transcripción de un audio.

Por otro lado, en la prueba de búsqueda de menciones a palabras, se pueden observar unas tasas más altas de recuperación en el audio con voz masculina, aunque por el contrario, las tasas más altas de precisión las obtenemos con el audio con voz femenina.

4.5 Prueba de acento del hablante

La finalidad de esta prueba es detectar variaciones en la transcripción de un mismo contenido de audio, pero dicho por personas que representan diferentes dialectos del español. La prueba consiste en la transcripción de una misma frase y el análisis de WER de cada uno de los acentos para compararlos al final de este apartado.

En esta prueba, la frase que se va a utilizar es la frase de Camarón de la Isla: *“La pureza no se puede perder nunca cuando uno la lleva dentro de verdad. Entonces, lo único que veo es que la gente no me comprende como yo canto. Mi manera de sentir la gente todavía no la ha entendido, entonces no les echo cuenta.”* Precediendo la frase anterior, en cada prueba se dirá además *“Prueba español de _____”* siendo el nombre del sitio de procedencia del hablante la palabra que rellena el hueco (p. ej.: *“Prueba español de Argentina”*). De este modo procedemos a realizar la prueba con los audios de grabación propia con diferentes hablantes masculinos usando el mismo micrófono (excepto el puertorriqueño, que también es grabado desde un micrófono, pero diferente al resto. Aunque bajo las mismas condiciones).

4.5.1 Prueba de andaluz

Realizamos la prueba de transcripción mediante el script *test_simple.py*. Los resultados se exponen en la Tabla 15.

Contenido Original	<i>“Prueba español de Andalucía. La pureza no se puede perder nunca cuando uno la lleva dentro de verdad. Entonces, lo único que veo es que la gente no me comprende como yo canto. Mi manera de sentir la gente todavía no la ha entendido, entonces no les echo cuenta”.</i>
Texto transcrito	“prueba de Andalucía” “la pobreza no se puede perder nunca cuando uno las lleva dentro de verdad entonces lo único que veo que la gente no me comprende como yo canto me sentí todavía la gente no entendí entonces no les dio cuenta”.
Análisis de las diferencias	“prueba (español) de Andalucía” “la pobreza (pureza) no se puede perder nunca cuando uno las (la)

	lleva dentro de verdad entonces lo único que veo que la gente no me comprende como yo canto (mi manera) me (de) sentí (sentir) todavía la gente no (la ha) entendí (entendido) entonces no les dio (echo) cuenta”.
--	--

Tabla 15. Análisis de la transcripción del audio con hablante andaluz.

Sobre un total de 49 palabras del texto original, se han detectado un total de 6 substituciones, 5 eliminaciones y ninguna inserción, resultando la WER:

$$WER = \frac{S + B + I}{N} = \frac{6 + 5 + 0}{49} = 22,45\%$$

4.5.2 Prueba de argentino

Realizamos la prueba de transcripción mediante *test_simple.py*. Los resultados se exponen en la Tabla 16.

Contenido Original	“Prueba español de Argentina. La pureza no se puede perder nunca cuando uno la lleva dentro de verdad. Entonces, lo único que veo es que la gente no me comprende como yo canto. Mi manera de sentir la gente todavía no la ha entendido, entonces no les echo cuenta.”
Texto transcrito	“prueba español argentina” “grecia se puede pero nunca cuando uno la gente de verdad entonces lo único que que la gente no me comprende como canto me sentí todavía la gente no le ha entendido” “entonces no les echó cuenta”
Análisis de las diferencias	“prueba español (de) argentina” “(la) grecia (pureza) (no) se puede pero (perder) nunca cuando uno la gente (lleva) de verdad entonces lo único que (veo es que) que la gente no me comprende como (yo) canto (mi manera) me (de) sentí (sentir) todavía la gente no le ha entendido” “entonces no les echó (echo) cuenta”

Tabla 16. Análisis de la transcripción del audio con hablante argentino.

Sobre un total de 49 palabras del texto original, se han detectado un total de 6 substituciones, 9 eliminaciones y ninguna inserción. La WER se calcula como:

$$WER = \frac{S + B + I}{N} = \frac{6 + 9 + 0}{49} = 30,61\%$$

4.5.3 Prueba de canario

Realizamos la prueba de transcripción mediante *test_simple.py*. Los resultados se exponen en la Tabla 17.

Contenido Original	<i>“Prueba español de Canarias. La pureza no se puede perder nunca cuando uno la lleva dentro de verdad. Entonces, lo único que veo es que la gente no me comprende como yo canto. Mi manera de sentir la gente todavía no la ha entendido, entonces no les echo cuenta.”</i>
Texto transcrito	“prueba español le ganaría la pureza no se puede perder nunca cuando uno la lleva dentro de verdad entonces lo único que veo que la gente no me comprende como local mi manera de sentir y todavía la gente no entender entonces no les ello cuenta”
Análisis de las diferencias	“prueba español le (de) ganaría (Canarias) la pureza no se puede perder nunca cuando uno la lleva dentro de verdad entonces lo único que veo (es) que la gente no me comprende como (yo) local (canto) mi manera de sentir y todavía la gente no (la ha) entender (entendido) entonces no les ello (echo) cuenta”

Tabla 17. Análisis de la transcripción del audio con hablante canario.

Sobre un total de 49 palabras del texto original, se han detectado un total de 5 substituciones, 4 eliminaciones y 1 inserción. La WER se calcula a continuación:

$$WER = \frac{S + B + I}{N} = \frac{5 + 4 + 1}{49} = 20,41\%$$

4.5.4 Prueba de catalán

Realizamos la prueba de transcripción mediante *test_simple.py*. Los resultados se exponen en la Tabla 18.

Contenido Original	<i>“Prueba español de Cataluña. La pureza no se puede perder nunca cuando uno la lleva dentro de verdad. Entonces, lo único que veo es que la gente no me comprende como yo canto. Mi manera de sentir todavía la gente no la ha entendido, entonces no les echo cuenta.”</i>
Texto transcrito	“español de cataluña pureza” “se puede perder nunca cuando uno la lleva dentro de verdad entonces puede único qué veo que la gente enorme compran coyoacán mi manera de sentir todavía la gente no va a entender entonces no les ha hecho cuenta”
Análisis de las diferencias	“(prueba) español de cataluña (la) pureza” “(no) se puede perder nunca cuando uno la lleva dentro de verdad entonces puede (lo) único qué veo (es) que la gente enorme (no me)”

	compran (comprende) coyoacán (como yo canto) mi manera de sentir todavía la gente no va (la) a (ha) entender (entendido) entonces no les ha hecho (echo) cuenta”
--	--

Tabla 18. Análisis de la transcripción del audio con hablante catalán.

En este caso los fallos a (ha) y hecho (echo) no se cuentan como error dado que suenan igual y se pueden confundir por la estructura de la frase transcrita.

Sobre un total de 49 palabras del texto original, se han detectado un total de 2 substituciones, 4 eliminaciones y 2 inserciones. La WER se calcula como sigue:

$$WER = \frac{S + B + I}{N} = \frac{8 + 4 + 2}{49} = 28,57\%$$

4.5.5 Prueba de gallego

Realizamos la prueba de transcripción mediante *test_simple.py*, los resultados se exponen en la Tabla 19.

Contenido Original	“Prueba español de Galicia. La pureza no se puede perder nunca cuando uno la lleva dentro de verdad. Entonces, lo único que veo es que la gente no me comprende como yo canto. Mi manera de sentir todavía la gente no la ha entendido, entonces no les echo cuenta.”
Texto transcrito	“español de cali” “la pura eso no se puede perder nunca cuando uno debate entre de verdad entonces lo único que la gente no me compran como you can vies manera de sentir todavía la gente no entiende” “cuenta”
Análisis de las diferencias	“(prueba) español de cali (Galicia)” “la pura eso (pureza) no se puede perder nunca cuando uno (la) debate (lleva) entre (dentro) de verdad entonces lo único que la gente no me compran (comprende) como you (yo) can (canto) vies (mi) manera de sentir todavía la gente no (la ha) entiende (entendido)” “(entonces no les echo) cuenta”

Tabla 19. Análisis de la transcripción del audio con hablante gallego.

Sobre un total de 49 palabras del texto original, se han detectado un total de 9 substituciones, 8 eliminaciones y ninguna inserción. La WER se calcula como:

$$WER = \frac{S + B + I}{N} = \frac{9 + 8 + 0}{49} = 34,69\%$$

4.5.6 Prueba de puertorriqueño

Realizamos la prueba de transcripción mediante *test_simple.py*. Los resultados se exponen en la Tabla 20.

Contenido Original	“Prueba español de Puerto Rico. La pureza no se puede perder nunca cuando uno la lleva dentro de verdad. Entonces, lo único que veo es que la gente no me comprende como yo canto. Mi manera de sentir todavía la gente no la ha entendido, entonces no les echo cuenta.”
Texto transcrito	“prueba español de puerto rico” “la pureza no se puede del nunca cuando uno la lleva dentro de verdad entonces lo único que dado que la gente no me comprende como yo canto mi manera decente todavía la gente delante mío entonces no les hecho cuenta”
Análisis de las diferencias	“prueba español de puerto rico” “la pureza no se puede del (perder) nunca cuando uno la lleva dentro de verdad entonces lo único que dado (veo) (es) que la gente no me comprende como yo canto mi manera decente (de sentir) todavía la gente (no) delante (la ha) mío (entendido) entonces no les hecho (echo) cuenta ”

Tabla 20. Análisis de la transcripción del audio con hablante puertorriqueño.

La palabra hecho (echo) no se contabiliza como error ya que suenan igual.

Sobre un total de 50 palabras del texto original, se han detectado un total de 7 substituciones, 2 eliminaciones y ninguna inserción. La WER se calcula:

$$WER = \frac{S + B + I}{N} = \frac{7 + 2 + 0}{50} = 0,18 = 18,00\%$$

4.5.7 Comparación de resultados

Para comparar la calidad de las transcripciones, nos fijamos en la WER de cada una de ellas. Las WER de cada una de las pruebas son las expuestas en la Tabla 21.

Acento	WER (%)
Andaluz	22,45
Argentino	30,61
Canario	20,41
Catalán	28,57
Gallego	34,69
Puertorriqueño	18,00

Tabla 21. Tabla resumen de los resultados de la prueba de acentos.

La WER indica la tasa de palabras con error, de modo que la transcripción con la WER más elevada es la peor. En este caso, los audios son exactamente iguales: sin ruido de fondo, misma velocidad de habla, una voz masculina y grabados con el mismo micrófono (excepto el puertorriqueño).

Las pruebas realizadas, indican que los acentos que mejor reconoce Vosk, son el de Puerto Rico, Canarias y Andalucía. Los que peor transcribe son el de Galicia y el de Argentina. Como conclusión de estas pruebas, podemos asumir que el acento del hablante tiene influencia en la transcripción a texto de un audio, ya que existe una variación de WER máxima del 24,86%.

4.6 Prueba de idioma

En este apartado se compararán dos audios, uno en español y el otro en inglés. La finalidad de esta prueba es una comparación de la calidad de la transcripción de los diferentes modelos de Vosk, en este caso el modelo en inglés y el modelo en español. En esta prueba, analizaremos la transcripción de dos audios con el mismo contenido y con voces iguales, pero uno en español y otro en inglés. Para realizar esta prueba hay que tener ambos modelos de Vosk descargados. Realizaremos la transcripción de los dos audios y calcularemos la WER, para poder comparar numéricamente la calidad de transcripción del modelo inglés y el modelo en español. Tanto el audio en español [59] como el audio en inglés [60] han sido obtenidos de Internet.

4.6.1 Modelo en inglés

Realizamos la transcripción del audio en inglés. Los resultados se exponen en la Tabla 22.

Contenido Original	<i>“Today we are going to talk about some animals that existed sixty-five million years ago. I’m sure you’ve heard about the dinosaurs and I’m certain you have seen them in many films and cartoons featuring dinosaurs. But not everything that is told in these films is correct so you don’t mind, let’s find now what they have really like and how they lived. The first thing we need to understand is that people and dinosaurs never lived together and dinosaurs lived before human beings existed and they roamed the Earth for approximately one hundred and thirty-five million years.”</i>
Texto transcrito	“today we going to talk about some animals that existed sixty five million years ago i'm sure you've heard about the dinosaurs and i'm certain you have seen them in many films and cartoons featuring dinosaurs but not everything that is told in these films is correct so have you don't mind let's find out what they have really like and how they lived the first thing we need to understand is that people and dinosaurs never lived together the dinosaurs lived before human beings existed and they roam the earth for approximately one hundred and thirty five million years”
Análisis de las diferencias	“today we (are) going to talk about some animals that existed sixty five million years ago i'm sure you've heard about the dinosaurs and i'm certain you have seen them in many films and cartoons featuring dinosaurs but not everything that is told in these films is correct so have you don't mind let's find out what they have really like and how they lived the first thing we need to understand is that people and dinosaurs never lived together the (and) dinosaurs lived before human

	beings existed and they roam (roamed) the earth for approximately one hundred and thirty five million years”
--	--

Tabla 22. Análisis de la transcripción del audio con el modelo en inglés.

Sobre un total de 98 palabras del texto original, se han detectado un total de 2 substituciones, 1 eliminación y ninguna inserción. La WER se calcula:

$$WER = \frac{S + B + I}{N} = \frac{2 + 1 + 0}{98} = 3,06\%$$

La WER de esta prueba con el modelo en inglés muestra una transcripción de muy buena calidad.

4.6.2 Modelo en español

Realizamos la misma prueba que anteriormente, pero con el audio en español. Los resultados se exponen en la Tabla 23.

Contenido Original	<i>“Hoy queremos hablaros de unos animales que existieron hace sesenta y cinco millones de años. Seguro que habéis oído hablar de los dinosaurios, es más, seguro que habéis visto muchas películas y dibujos animados en los que aparecen, pero no todo lo que se dice en esas pelis es cierto. Si os parece bien vamos a conocer cómo eran y cómo vivían los dinosaurios de verdad. Lo primero que tenemos que saber es que los dinosaurios y los seres humanos nunca convivieron. Los dinosaurios dominaron la Tierra durante ciento treinta y cinco millones de años aproximadamente.”</i>
Texto transcrito	“hoy queremos hablar de unos animales que existieran hace sesenta y cinco millones de años seguro que habéis oído hablar de los dinosaurios es más seguro que habéis visto muchas películas y dibujos animados en los que aparecen pero no todo lo que se dice en esas pelis es cierto si os parece bien vamos a conocer cómo eran y cómo vivían los dinosaurios de verdad lo primero que tenemos que saber es que los dinosaurios y los seres humanos nunca convivieron los dinosaurios dominar en la tierra durante ciento treinta y cinco millones de años aproximadamente”
Análisis de las diferencias	“hoy queremos hablar de unos animales que existieran (existieron) hace sesenta y cinco millones de años seguro que habéis oído hablar de los dinosaurios es más seguro que habéis visto muchas películas y dibujos animados en los que aparecen pero no todo lo que se dice en esas pelis es cierto si os parece bien vamos a conocer cómo eran y cómo vivían los dinosaurios de verdad lo primero que tenemos que saber es que los dinosaurios y los seres humanos nunca convivieron los dinosaurios dominar en (dominaron) la tierra durante ciento treinta y cinco millones de años aproximadamente”

Tabla 23. Análisis de la transcripción del audio con el modelo en español.

Sobre un total de 96 palabras del texto original, se han detectado un total de 7 substituciones, ninguna eliminación y 1 inserción. La WER se calcula:

$$WER = \frac{S + B + I}{N} = \frac{2 + 0 + 1}{96} = 0,0313 = 3,13\%$$

La WER muestra un resultado similar al del modelo en inglés.

4.6.3 Análisis de resultados

En el caso de los audios analizados tanto en inglés como español, se obtiene una transcripción de calidad, con una WER cercano al 3% en ambos modelos. Por lo tanto, tanto el modelo en español como el modelo en inglés son válidos. Según nuestra prueba, el idioma en el que está el audio (español o inglés) no es el factor que más influye a la hora de obtener una transcripción precisa.

4.7 Prueba con vocabulario técnico

En este apartado se van a realizar pruebas de transcripción con dos audios de vocabulario más técnico y con nombres propios. El primero es una explicación sobre mitología griega, donde aparecen palabras específicas sobre el tema y nombres propios. El segundo audio analizado va a ser un podcast llamado “¿Cómo funciona? Motor diésel y motor de gasolina (Diesel vs Otto)” de JF Calero [61].

En primer lugar, se va a realizar la transcripción de la explicación sobre mitología griega. La prueba se va a realizar mediante el script `test_simple.py`. Los resultados se exponen en la Tabla 24.

Contenido Original	<i>“Zeus o Júpiter en el mundo romano, es el principal dios de la mitología grecolatina. Dicen los textos antiguos que Rea y su hermano Crono tuvieron seis hijos: Deméter, Hades, Hera, Hestia, Poseidón y Zeus. La madre de Crono, Gea le advirtió que uno de sus descendientes le derrotaría. Por ello, el titán devoraba a sus hijos nada más nacer. Sin embargo, Rea consiguió salvar a Zeus de la voracidad de su padre dándole una piedra envuelta en pañales en lugar de al dios recién nacido. Este fue llevado a la isla de Creta y criado allí por una de las ninfas alimentándose con néctar, ambrosía y leche de la cabra Amaltea. Una vez creció, se propuso derrotar a su padre Crono y a sus tíos los titanes, hijos de Urano y Gea en una época de guerra conocida como la titanomáquia.”</i>
Texto transcrito	“zeus o júpiter en el mundo romano es el principal dios de la mitología greco latina dicen los textos antiguos quería y su hermano crono tuvieron seis hijos de meter a vez era existía poseidón y zeus la madre de crono gea le admire” “tio que uno de sus descendientes la derrota haría por ello el titán devoraba a sus hijos nada más nacer sin embargo rea consiguió salvar a zeus de la voracidad de su padre dándole una piedra envuelta en pañales el lugar de al dios recién nacido este fue llevado a la isla de creta y criado allí por una de las” “ninfas alimentándose conectar ambrosía y leche de la cabra mal tea una vez creció se propuso derrotar a su padre crono y a sus tíos los titanes hijos de urano y gea en una época de guerra conocida como la

	titanio magia”
Análisis de las diferencias	“zeus o júpiter en el mundo romano es el principal dios de la mitología greco latina dicen los textos antiguos querría (que Rea) y su hermano crono tuvieron seis hijos de meter (Deméter) a vez (Hades) era (Hera) existía (Hestia) poseidón y zeus la madre de crono gea le admire (advirtió)” “tio que uno de sus descendientes la (le) derrota haría (derrotaría) por ello el titán devoraba a sus hijos nada más nacer sin embargo rea consiguió salvar a zeus de la voracidad de su padre dándole una piedra envuelta en pañales el (en) lugar de al dios recién nacido este fue llevado a la isla de creta y criado allí por una de las” “ninfas alimentándose conectar (con néctar) ambrosía y leche de la cabra mal tea (Amaltea) una vez creció se propuso derrotar a su padre crono y a sus tíos los titanes hijos de urano y gea en una época de guerra conocida como la titanio magia (titanomáquia)”

Tabla 24. Análisis de la transcripción del audio con vocabulario específico de mitología griega.

Sobre un total de 142 palabras del texto original, se han detectado un total de 14 substituciones, ninguna eliminación y 1 inserción. La WER se calcula:

$$WER = \frac{S + B + I}{N} = \frac{14 + 0 + 1}{142} = 10,56\%$$

Los resultados muestran una transcripción de una calidad alta, y ya se pueden observar algunos patrones en el tipo de palabras donde se concentran los errores.

A continuación, se realiza la misma prueba con el podcast mencionado anteriormente. Los resultados se exponen en la Tabla 25.

Contenido Original	<i>“En un Otto la combustión normal empieza con una chispa que inflama la mezcla de aire y gasolina, eso crea un frente de llama que se propaga por la cámara a una velocidad relativamente lenta. Voy a citar el que ha sido el manual para muchos ingenieros industriales sobre esta materia el libro “Motores de combustión interna alternativos”, de los profesores Muñoz y Paire. Dice el libro a propósito de los motores Otto: “esta propagación así definida, está incluida dentro de lo que se conoce como deflagración, que corresponde a una combustión que se propaga a velocidades inferiores a la del sonido””.</i>
Texto transcrito	“en un auto la combustión normal empieza con una chispa que inflama la mezcla de aire y gasolina eso crea un frente de llama que se propaga por la cámara a una velocidad relativamente lenta voy a citar el que ha sido el manual para muchos ingenieros industriales sobre esta materia el libro motores de combustión interna alternativos de los” “profesores muñoz y país dice el libro a propósito de los motores soto

	esta propagación así definida está incluida dentro de lo que se conoce como deflagración que corresponde a una combustión que se propaga velocidades inferiores a la del sonido”
Análisis de las diferencias	“en un auto (Otto) la combustión normal empieza con una chispa que inflama la mezcla de aire y gasolina eso crea un frente de llama que se propaga por la cámara a una velocidad relativamente lenta voy a citar el que ha sido el manual para muchos ingenieros industriales sobre esta materia el libro motores de combustión interna alternativos de los” “profesores muñoz y país (Paire) dice el libro a propósito de los motores soto (Otto) esta propagación así definida está incluida dentro de lo que se conoce como deflagración que corresponde a una combustión que se propaga (a) velocidades inferiores a la del sonido”

Tabla 25. Análisis de la transcripción del audio con vocabulario específico sobre motores.

Sobre un total de 102 palabras del texto original, se han detectado un total de 3 substituciones, 1 eliminación y ninguna inserción. La WER se calcula:

$$WER = \frac{S + B + I}{N} = \frac{3 + 1 + 0}{102} = 3,92\%$$

En esta prueba la herramienta muestra una transcripción muy precisa del podcast sobre motores.

Por último, realizamos una prueba de detección de palabras. Las palabras a buscar serán dos palabras comunes (una larga y una corta) de cada audio y nombres propios, los resultados aparecen en la Tabla 26.

	Términos comunes					Nombres propios			
Palabra	Voracidad	Titán	Combustión	Chispa	De	Hades	Rea	Otto	Paire
Nº de apariciones en el audio	1	1	3	1	10	1	2	2	1
Nº de detecciones correctas	1	1	3	1	8	0	2	2	0
Nº de detecciones totales	4	7	4	1	19	0	15	2	0
Recuperación (%)	100	100	100	100	80	0	100	100	0
Precisión (%)	25	14,86	75	100	42,11	-	13,33	100	-

Tabla 26. Resultado de la búsqueda de menciones a términos en los audios sobre mitología y motores.

4.7.1 Análisis de resultados

En esta prueba de vocabulario, podemos obtener dos conclusiones generales, una es que el modelo de Vosk en español permite reconocer palabras de temas muy concretos, como por ejemplo de la mitología griega o sobre motores. Y la otra conclusión es que el fallo más común en la transcripción de este tipo de audios son los nombres propios. Aunque alguno de los buscados se detectó el 100% de las veces con una precisión del 100% hay margen de mejora, lo que probablemente requería adaptar el modelo de Vosk.

En la prueba de búsqueda de menciones a palabras, se observa un porcentaje de recuperación alto, ya que la mayoría de menciones se encuentran, pero se puede observar que en algunos nombres propios (p.ej.: Hades, Paire) la tasa de recuperación es de 0. Por otro lado, en las palabras más corrientes las tasas de recuperación son muy altas (la más pequeña es de un 80%). Las tasas de precisión observadas son mucho más altas en palabras largas, las cuales hay menor probabilidad de confundir, que en las palabras cortas, que se pueden confundir con algunos fonemas de palabras más largas.

4.8 Pruebas en comunicaciones navales

En esta sección realizaremos pruebas de búsqueda de palabras y de transcripción en diferentes grabaciones de comunicaciones navales. Estas grabaciones han sido proporcionadas por la Escuela Naval Militar. En primer lugar, se va a realizar las pruebas de vocabulario en una conversación entre dos unidades. Después se va a realizar la transcripción de una breve comunicación entre dos buques (en inglés) y por último de un *hailing* (en inglés). Un *hailing* es una realización de una serie de preguntas de un buque militar a otro buque.

4.8.1 Prueba de vocabulario naval

Se va a realizar una prueba de transcripción de una prueba de vocabulario naval en español. Los resultados se exponen en la Tabla 27.

Contenido Original	<i>“delta charlie de delta alfa cambio, adelante para delta charlie cambio aquí delta alfa vamos a pasar el vocabulario aquí delta charlie recibido alfa, bravo, Charlie, delta, echo, foxtrot, golf, hotel, india, juliet, kilo, lima, Mike, november, óscar, papa, Quebec, romeo, sierra, tango, uniform, Víctor, whiskey, x-ray, yankee, zulu pollo, gallina, gallo, hierros, regalo, regalo plus, mairena, historias, toro, goma, madre, agua, cacahuete, sorpresas, tirado, real Madrid, campeón, cacahuete caliente, cacao delta charlie de delta alfa por esa parte hemos finalizado aquí delta charlie recibido en un minuto transmitimos nosotros por el seis nueve aquí delta alfa recibido”</i>
Texto transcrito	“charly de alta” “adelante para detectar y capital”

	“aquí el vocabulario” “aquí destacar y recibido” “alza bravo hotel india kilo lima papá romeo sierra tango uniforme tú sede la ira” “pollo mayo regalo regalo mairena podrías roma madrid agua sorpresas tirado real madrid cacahuete cacao” “me gustaría por otra parte tengo finalizado” “recibido y transmitir nosotros por aquí” “aquí es alta recibido”
Análisis de las diferencias	“(delta) charly de (delta) alta (alfa)” “adelante para detectar (delta) y capital (Charlie) (cambio)” “aquí (delta alfa vamos a pasar) el vocabulario” “aquí destacar (delta) (Charlie) y recibido” “alza (alfa) bravo (Charlie delta echo foxtrot golf) hotel india kilo lima (mike november oscar) papá (Quebec) romeo sierra tango uniforme (uniform) tú sede la ira (victor whiskey x-ray yankee zulu)” “pollo (gallina) mayo (gallo) (hierros) regalo regalo (plus) mairena podrías (historias) (toro) roma (goma) madrid (madre) agua (cacahuete) sorpresas tirado real madrid (campeón) cacahuete (caliente) cacao” “me gustaría (delta charlie) (de delta alfa) por otra (esa) parte tengo (hemos) finalizado” “(aquí delta charlie) recibido (en un minuto) y transmitir (transmitimos) nosotros por aquí (el seis nueve)” “aquí es (delta) alta (alfa) recibido”

Tabla 27. Análisis de la transcripción del audio de prueba de vocabulario naval.

Sobre un total de 98 palabras del texto original, se han detectado un total de 25 substituciones, 31 eliminaciones y 2 inserciones. La WER se calcula:

$$WER = \frac{S + B + I}{N} = \frac{25 + 31 + 2}{98} = 59,18\%$$

El resultado obtenido muestra una transcripción con un elevado número de errores.

4.8.2 Prueba de comunicación breve del aviso de un movimiento

Esta prueba se va a realizar con el modelo en inglés. Los resultados se exponen en la Tabla 28.

Contenido Original	“will be overtaking in your starboard side, I repeat, will be overtaking your starboard side”
--------------------	---

Texto transcrito	“the out a couple of days” “i repeat mere be over to hear the part about that”
Análisis de las diferencias	“the out a couple of days (will be overtaking your starboard side)” “i repeat mere (will) be over to hear (overtaking) the part about that (your starboard side)”

Tabla 28. Análisis de la transcripción del audio con un aviso de un movimiento a otro buque.

Sobre un total de 15 palabras del texto original, se han detectado un total de 11 substituciones, ninguna eliminación y ninguna inserción. La WER se calcula como:

$$WER = \frac{S + B + I}{N} = \frac{11 + 0 + 0}{15} = 73,33\%$$

El resultado muestra un muy elevado porcentaje de errores en la transcripción, aunque por la longitud de la prueba este número no es del todo concluyente en el resultado de las pruebas de comunicaciones navales.

4.8.3 Prueba de transcripción de un “hailing”

Se va a realizar una prueba de transcripción de un *hailing*. Los resultados se exponen en la Tabla 29.

Contenido Original	“Vessel in my bow it is NATO warship foxtrot one zero three operating in NATO operation Sea Guardian in Mediterranean Sea in order to deter and contain maritime security bridge. Switch to channel six nine. Good morning sir what is your plan? What is your IMO number? What is the last port of call and date of departure? What is the port of call and date of arrival? What is your owner? Do you know Operation Sea Guardian, sir? NATO units are conducting Operation Sea Guardian in area to deter, defend and protect against terrorism and terrorist support activities. Please, report any suspicious activities or information NATO warship in NATO ship report number is zero, zero, four, four, one, nine, two, three, nine, five, six, five, seven, four or via mail to info@ship.nato. Thank you for your cooperation sir.”
Texto transcrito	“major a mofo pts” “nato or she of all three or pretty a tv” “not op operation she got down to admit it or not into the dirt on put their my immaturity for its feet awkward a tunnel she's nice” “but money what is a flat” “what is your i phone number” “what does not go on fate of the episodes”

	“what's this” “have gone on a of arrival” “what is your owner” “what is your ear” “do you know of what is your own theory on earth” “it's not a huge is adequate you see operators she got that area for their to retire defense is there are protest against terrorists on story it supports activities please report any suspicious activities or information that oversees it as” “if he said the phone number it's judo pseudo for for one nine two three four five six five center for be made to people are she be taught me to into a for your supposed to set up as if you're not a is that the war see a moratorium italian food” “he” “it's”
Análisis de las diferencias	“major a mofo pts” “nato or she of all three or pretty a tv” “not op operation she got down to admit it or not into the dirt on put their my immaturity for its feet awkward a tunnel she's nice” “but money what is a flat” “what is your i phone number” “what does not go on fate of the episodes” “what's this” “have gone on a of arrival” “what is your owner” “what is your ear” “do you know of what is your own theory on earth” “it's not a huge is adequate you see operators she got that area for their to retire defense is there are protest against terrorists on story it supports activities please report any suspicious activities or information that oversees it as” “if he said the phone number it's judo pseudo for for one nine two three four five six five center for be made to people are she be taught me to into a for your supposed to set up as if you're not a is that the war see a moratorium italian food” “he” “it's”

Tabla 29. Análisis de la transcripción del audio de un *hailing*.

Sobre un total de 139 palabras del texto original, se han detectado un total de 94 substituciones, ninguna eliminación y 7 inserciones. La WER se calcula como:

$$WER = \frac{S + B + I}{N} = \frac{94 + 0 + 7}{139} = 72,66\%$$

Este resultado apoya al anterior, de forma que en este caso los resultados ya son más concluyentes, dado que la WER es muy similar.

4.8.4 Prueba de búsqueda de mención a términos

A continuación, se va a realizar una prueba de búsqueda de mención a términos en las tres pruebas anteriores de comunicaciones navales. Los resultados se obtienen mediante el uso de *test_words.py* y se muestran en la Tabla 30.

	Vocabulario			Aviso de movimiento		Hailing		
Palabra	Delta	November	Cacahuete	Starboard	Vessel	Bow	Warship	Port
Nº de apariciones en el audio	10	1	2	2	1	1	2	2
Nº de detecciones correctas	8	0	2	2	1	1	2	2
Nº de detecciones totales	9	0	2	3	15	29	12	26
Recuperación (%)	80	0	100	100	100	100	100	100
Precisión (%)	88,89	-	100	66,67	6,67	3,45	16,67	7,69

Tabla 30. Resultado de la búsqueda de menciones a términos en los audios de comunicaciones navales.

4.8.5 Análisis de resultados

Tanto la transcripción del modelo en inglés como del modelo español se ven afectados por la calidad del audio y por el vocabulario muy específico, ya que el vocabulario naval solamente se usa para la comunicación entre buques o con estaciones en tierra. El vocabulario específico, sumado a una calidad baja de audio (son comunicaciones VHF entre buques) provoca una mala transcripción, y por ende la búsqueda de palabras será dificultosa.

Para comprobar si el problema es la calidad o es el vocabulario, realizamos una prueba más. En esta prueba grabamos personalmente un audio con el mismo contenido que la prueba de *hailing* y comparamos los resultados. A continuación, en la Tabla 31 se muestran los resultados de esta nueva prueba.

Contenido Original	“Vessel in my bow it is NATO warship foxtrot one zero three operating in NATO operation Sea Guardian in Mediterranean Sea in
--------------------	--

	order to deter and contain maritime security bridge. Switch to channel six nine. Good morning sir what is your plan? What is your IMO number? What is the last port of call and date of departure? What is the port of call and date of arrival? What is your owner? Do you know Operation Sea Guardian, sir? NATO units are conducting Operation Sea Guardian in area to deter, defend and protect against terrorism and terrorist support activities. Please, report any suspicious activities or information NATO warship in NATO ship report number is zero, zero, four, four, one, nine, two, three, nine, five, six, five, seven, four or via mail to info@ship.nato. Thank you for your cooperation sir.”
Texto transcrito	“brazil a mobo it is nato warship foxtrot one zero three operating in nato operation see guardian in that the mediterranean sea in order to that their and contain married him sick or” “the bridge” “switch to channel sick nine” “good morning sir what is your plan” “what is your email number” “what is the last part” “of call and they have the porter” “what is the port of call and they have a driver what is your owner the you know operation several cancer” “nato unit are conducting operations who are the and in area to the to defend and protect against terrorism and thirty support activity” “please report any suspicious activities or him for me who donate the worship a nato ship report number is zero zero for for one nine two three nine five six” “five seven four or via mail to infer at a ship that nato” “thank you for your cooperation to”
Análisis de las diferencias	“brazil (vessel) (in) a (my) mobo (bow) it is nato warship foxtrot one zero three operating in nato operation see (sea) guardian in that the mediterranean sea in order to that their (deter) and contain married him (maritime) sick or (security)” “the bridge” “switch to channel sick nine” “good morning sir what is your plan” “what is your email (IMO) number” “what is the last part (port)” “of call and they (date) have the porter (of departure)” “what is the port of call and they (date) have a driver (of arrival) what is your owner the you know operation several cancer (sea guardian) (sir)”

	“nato unit are conducting operations who are the and (sea guardian) in area to the to (deter) defend and protect against terrorism and thirty (terrorism) support activity” “please report any suspicious activities or him for me who donate the worship a nato ship report number is zero zero for (four) for (four) one nine two three nine five six” “five seven four or via mail to infer (info) at a ship that (dot) nato” “thank you for your cooperation to (sir)”
--	--

Tabla 31. Análisis de la transcripción de un *hailing* grabado con micrófono.

Sobre un total de 139 palabras del texto original, se han detectado un total de 28 substituciones, 2 eliminaciones y 5 inserciones. La WER se calcula de la siguiente forma:

$$WER = \frac{S + B + I}{N} = \frac{28 + 2 + 5}{139} = 25,18\%$$

El resultado de esta prueba de transcripción muestra que el mal resultado de la prueba de comunicación naval depende en buena parte de la calidad de las comunicaciones. Por otro lado, también se ha comprobado que en el caso del primer audio “vocabulario naval”, las letras (alfa, bravo, charlie, delta, echo, etc.) no aparecen en el modelo en español, por lo tanto, habría que incluirlas.

En la prueba de búsqueda de menciones a términos, se observan unas buenas tasas de recuperación, esto se traduce en que la mayoría de las veces que se menciona este término en el audio, Vosk es capaz de encontrarlo y mostrarlo al usuario. Por el contrario, las tasas de precisión en algunas palabras son muy bajas, lo que significa que Vosk muestra muchas más menciones a los términos de lo que en realidad se menciona en el audio, de modo que cuesta diferenciar cual son las menciones correctas.

4.9 Pruebas de detección de términos sin contexto

En este apartado se va a proceder a grabar una serie de audios en inglés y español con solamente una palabra en cada uno de ellos, de esta forma se va a evaluar la herramienta Vosk como a una herramienta de detección de menciones a términos en el caso en el que no haya otras palabras que puedan ser utilizadas como contexto para apoyar el proceso de reconocimiento. Esta prueba se lleva a cabo con los audios de palabras aisladas, y se evalúa mediante una tasa de aciertos (lo que hemos llamado recuperación en las anteriores secciones). Las palabras incluidas en la prueba deben ser variadas, desde palabras técnicas a nombres propios, incluyendo vocabulario naval y algunas palabras más comunes.

Los audios van a ser grabados con el mismo micrófono, con el mismo hablante y con el mismo nivel de ruido. Los resultados de las pruebas se separan en dos tablas, una para las palabras en inglés (Tabla 33) y otra en español (Tabla 32). Para la transcripción de la palabra contenida en el audio se va a usar el script *test_simple.py* y para la búsqueda de la mención a la misma palabra se usará el script *test_words.py*, de modo que cabe la posibilidad que la palabra esté mal transcrita pero que la búsqueda sea correcta.

Palabra	Proa	Estribor	Barcelona	Boston	Bioetanol	Selva	Neutrino	Pureza
Transcripción	Pero a	Estribor	Barcelona	Boston	Bioetanol	Selva	Neutro China	Pureza
Búsqueda (✓/✗)	✓	✓	✓	✓	✓	✓	✗	✓

Tabla 32. Prueba de transcripción y búsqueda de palabras en español.

Palabra	Bow	Starboard	Barcelona	Boston	Bioethanol	Jungle	Neutrino	Purity
Transcripción	Bow	Start board	Barcelona	Boston	Bio ethanol	Jungle	Neutrino	Purity
Búsqueda (✓/✗)	✓	✓	✓	✓	✓	✓	✓	✓

Tabla 33. Prueba de transcripción y búsqueda de palabras en inglés.

A continuación, se va a calcular la tasa de acierto en la transcripción y en la búsqueda de cada uno de los modelos para comprarlos numéricamente. Primero

$$tasa\ acierto\ transcripción_{esp} = \frac{n^{\circ} \text{ palabras bien transcritas}}{n^{\circ} \text{ palabras totales}} = \frac{8}{10} = 80,00\%$$

$$tasa\ acierto\ búsqueda_{esp} = \frac{n^{\circ} \text{ palabras encontradas}}{n^{\circ} \text{ palabras totales}} = \frac{9}{10} = 90,00\%$$

$$tasa\ acierto\ transcripción_{eng} = \frac{n^{\circ} \text{ palabras bien transcritas}}{n^{\circ} \text{ palabras totales}} = \frac{9}{10} = 90,00\%$$

$$tasa\ acierto\ búsqueda_{eng} = \frac{n^{\circ} \text{ palabras encontradas}}{n^{\circ} \text{ palabras totales}} = \frac{10}{10} = 100\%$$

En esta prueba podemos observar que, aunque la herramienta no transcriba bien la palabra a la hora de buscar la mención sí que es capaz de reconocerla (caso de “proa” o “starboard”). Aunque también hay términos (p.ej.: “neutrino”) que son de un vocabulario muy específico, y que el modelo en español no parece capaz de transcribirla bien ni de encontrar la mención a este término.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

Uno de los principales objetivos del trabajo era la selección de una herramienta para realizar el análisis más detallado. De entre todas las estudiadas la elegida fue Vosk, que es una herramienta de código abierto, de fácil obtención e instalación que permite realizar transcripciones de audios, búsqueda de palabras o secuencias de palabras en un flujo de audio y que, además, incorpora funciones de transcripción en tiempo real de voz capturada a través del micrófono.

En este trabajo se han realizado diversas de pruebas para determinar cuáles son los factores más influyentes a la hora de determinar la calidad de la transcripción y detección de menciones a términos en un audio mediante la API Vosk. La métrica usada para la medición de la calidad de las transcripciones ha sido la WER (Word Error Rate) mientras que se ha usado precisión y recuperación para evaluar la detección de menciones. En las próximas secciones se detallan las principales conclusiones respecto a cada uno de los parámetros analizados.

5.1.1 Ruido de fondo

Las pruebas con el mismo audio y diferentes ruidos nos demuestran que el nivel de ruido influye de manera directa en la transcripción, a más ruido peor calidad de transcripción. Asimismo, también obtenemos una WER relativamente baja comparado con otras pruebas, por lo que el nivel de ruido influye, pero a los niveles en los que se han realizado las pruebas, no ha resultado ser una de las características más influyentes para una transcripción precisa.

El nivel de ruido, según las pruebas realizadas, no influye de manera significativa en la búsqueda de menciones a palabras, ya que las tasas de recuperación y precisión son muy parecidas entre los tres audios.

5.1.2 Nivel de la música de fondo y canciones

Por otra parte, también se hna analizado las transcripciones de un anuncio con música y voz, con una transcripción poco precisa (WER = 42,40%). De modo que el hecho de que haya música y voz a la vez en un audio, siendo el nivel de volumen semejante entre ellos, perjudica el nivel de calidad de la transcripción. Para comprobar si además de la música el ritmo de la voz también influye, se realizaron pruebas con dos canciones, mostrando resultados semejantes a los del anuncio. En cambio, una de las canciones analizadas muestra una transcripción de muy baja calidad. Por lo tanto, el factor de música y alteración del ritmo del habla (cantado) son factores que influyen de manera negativa.

Tanto en el anuncio como en las canciones puestas a prueba, el resultado en cuanto a la búsqueda de menciones a términos o palabras es más precisa en palabras más largas que en palabras más cortas, que parece que se pueden confundir con otros fonemas de palabras más largas.

5.1.3 Género del hablante

Otro factor a analizar era el factor del género del hablante, los resultados muestran que la voz de un hablante masculino proporciona mejor transcripción que la voz de un hablante femenino, aunque este no es de los factores más influyentes de los analizados.

En el audio con voz masculina se observa una mayor tasa de recuperación que en la voz femenina, por el contrario, la tasa de precisión es más elevada en el audio con voz femenina. En conclusión, a nivel del experimento realizado, en la búsqueda de menciones a términos es mejor la voz femenina o la voz masculina dependiendo de si se desea una mayor tasa de recuperación (voz masculina) o si se desea una mayor tasa de precisión de los resultados (voz femenina).

5.1.4 Procedencia geográfica del hablante

El factor geográfico también podía ser determinante, para ello, se analizaron diferentes audios con mismo contenido, pero con hablantes de procedencia muy diferente. El factor del acento del hablante a priori podría ser uno de los factores menos influyentes. Sin embargo, tras las pruebas realizadas se concluye en que es uno de los más influyentes en la transcripción de Vosk. La WER de los diferentes acentos es muy variable, desde aproximadamente un 40% a un 18%.

5.1.5 Idioma

El factor del idioma podría ser determinante, pero tras las pruebas realizadas, los modelos en español e inglés muestran resultados muy semejantes, con una tasa de error de menos del 4%. En conclusión, el idioma no parece un factor influyente en la calidad de la transcripción del audio, y por ende de la búsqueda de palabras en estos audios debido a los dos modelos analizados son igualmente válidos.

5.1.6 Vocabulario

Esta prueba toma resultados de dos audios sobre temas muy específicos. Los resultados de transcripción de los audios sobre temas como la mitología griega y los motores, que contenían vocabulario específico, fueron muy positivos, realizando una transcripción de muy buena calidad y precisión.

En la prueba de búsqueda de menciones a palabras, se puede confirmar que el principal fallo ocurría en la detección de los nombres propios. También se confirma la tendencia de las malas precisiones en la búsqueda de palabras cortas.

5.1.7 Comunicaciones navales

Los resultados de las comunicaciones navales no fueron tan positivos, de lo que se podría deducir que el vocabulario marítimo no se incluye en ninguno de los dos modelos. También el hecho de que las comunicaciones navales suelen tener una menor calidad de audio (debida a diferentes factores) influye a la hora de analizar estos flujos de audio, tanto para transcripción como para búsqueda de palabras. Para confirmar esa teoría se observan los resultados de la transcripción con el mismo

contenido del audio, pero grabado personalmente, donde se mejora la calidad del audio. Los resultados demuestran que la WER más elevada de las comunicaciones navales, se debe, en su mayor parte, a la calidad más baja del audio.

5.1.8 Detección de términos sin contexto

La prueba se realiza con 20 palabras, 10 en el modelo en español y 10 en el modelo en inglés. Las tasas de acierto en la transcripción y en la búsqueda del modelo en inglés son superiores a las del modelo en español. Esto podría significar que el vocabulario en el modelo en inglés es más amplio que el vocabulario de temas específicos en el modelo en español. Además, podemos observar como a pesar de que la palabra no ha sido transcrita en el texto (mediante *test_simple.py*) se puede encontrar la mención a esta palabra (mediante *test_words.py*).

5.1.9 Conclusiones personales

A título personal, este TFG me ha ayudado a mejorar la capacidad de síntesis y de análisis, también me ha introducido en el mundo del *Audio Mining* y de todas sus aplicaciones. No hay duda que el estudio de este campo se va a seguir desarrollando e irá tomando un papel más relevante, de modo que seguro que en mi carrera retomaré el análisis de audios para extraer la máxima información de estos.

5.2 Líneas futuras

- En este trabajo, todas las pruebas se han realizado en castellano y en inglés (con sus respectivos modelos), podría ser de interés evaluar el funcionamiento en otros idiomas de la herramienta elegida (Vosk API), puesto que dispone de modelos de más de 15 idiomas.
- Podría explorarse también el funcionamiento de la herramienta sobre datos recibidos por Internet (RTP, *Real Time Protocol*), leídos directamente del micrófono y obtenidos de un módulo SDR (*Software Defined Radio*) en tiempo real, ya que en este trabajo se han realizado sobre grabaciones.
- Estudiar los mecanismos que ofrece la herramienta para adaptar modelos existentes o entrenar modelos nuevos específicamente adaptados al escenario de análisis de comunicaciones marítimas.
- Aunque en este trabajo se han hecho pruebas diversas (pues el objetivo era comprobar el impacto de múltiples factores en el funcionamiento de la herramienta) el tamaño del corpus utilizado en cada prueba ha sido limitado (pues las pruebas se han hecho a mano) extender la evaluación a escenarios más exigentes en este sentido podría ser otra futura línea de trabajo a explorar.

6 BIBLIOGRAFÍA

- [1] D. Moelants, O. Cornelis, M. Leman, J. Gansemans, R. De Caluwe, G. De Tré, T. Matthé y A. Hallez, «Problems and Opportunities of Content-based Analysis and Description of Ethnic Music,» *International Journal of Intangible Heritage*, vol. 02, nº ISSN: 1975-3586, 2007.
- [2] M. Kaur Mand, D. Nagpal y Gunjan, «An Analytical Approach for Mining Audio Signals,» *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, nº 9, 2013.
- [3] N. Leavitt, «Let's hear it for audio mining,» *IEEE Journals & Magazines*, vol. 35, nº 10, pp. 23-25, 2002.
- [4] My Music Tools, «Grabadora de voz con detección de silencio,» [En línea]. Available: <https://www.mymusictools.com/es/articles/grabadora-de-voz-con-deteccion-de-silencio.htm>. [Último acceso: 27 01 2021].
- [5] L. Colls Palomero, «Identificación del hablante a partir del análisis acústico de una muestra de voz en condiciones de cansancio,» Trabajo de fin de grado, Universitat de Girona, 2016.
- [6] J. Masip, E. Garrido y C. Herrero, «La detección de la mentira mediante la medida de la tensión en la voz: una revisión crítica,» *Estudios de psicología*, vol. 1, nº 25, pp. 13-30, 2004.
- [7] D. Haddad, S. Walter, R. Ratley y M. Smith, «Investigation and Evaluation of Voice Stress Analysis Technology,» 2002.
- [8] E. A. Massanet Avelar, «PSE (Psychological Stress Evaluator),» 30 10 2017. [En línea]. Available: [https://mentirapedia.com/index.php/PSE_\(Psychological_Stress_Evaluator\)#:~:text=El%20Evaluador%20del%20Estr%C3%A9s%20Psicol%C3%B3gico,el%20estr%C3%A9s%20de%20la%20voz..](https://mentirapedia.com/index.php/PSE_(Psychological_Stress_Evaluator)#:~:text=El%20Evaluador%20del%20Estr%C3%A9s%20Psicol%C3%B3gico,el%20estr%C3%A9s%20de%20la%20voz..) [Último acceso: 27 01 2021].
- [9] INVOX Medical, «La Inteligencia Artificial en el reconocimiento de voz,» INVOX Medical, [En línea]. Available: <https://invoxmedical.com/noticias/la-inteligencia-artificial-en-el-reconocimiento-de-voz/>. [Último acceso: 27 01 2021].

- [10] J. M. López, «Cómo convertir voz a texto en tiempo real,» Blogthinkbig.com, 15 03 2019. [En línea]. Available: <https://blogthinkbig.com/voz-a-texto>. [Último acceso: 28 01 2021].
- [11] J. Reig, Escritor, *Fotografía de detección de una palabra en un audio..* [Performance]. 2010.
- [12] N. Fresko, «Alexa Skills Kit Blog,» Amazon, 22 07 2020. [En línea]. Available: <https://developer.amazon.com/en-US/blogs/alexa/alexa-skills-kit/2020/07/31-new-features-to-unlock-more-natural-and-immersive-alexa-experiences>. [Último acceso: 22 01 2021].
- [13] J. Panta Marínez, «Control Domótico por voz,» 27 09 2012. [En línea]. Available: <https://riunet.upv.es/bitstream/handle/10251/17631/Memoria.pdf?sequence=1>. [Último acceso: 30 01 2021].
- [14] Apple Inc., «Shazam,» Apple, [En línea]. Available: <https://www.shazam.com/es>. [Último acceso: 02 02 2021].
- [15] Google, «Google play música,» Google, [En línea]. Available: <https://support.google.com/googleplaymusic/answer/2913276?hl=es>. [Último acceso: 02 02 2021].
- [16] SoundHound Inc., «Soundhound,» Soundhound Inc., [En línea]. Available: <https://www.soundhound.com/>. [Último acceso: 02 02 2021].
- [17] *Soundhound logo.* [Performance]. Soundhound Inc..
- [18] Google, «Hey Google. Google assistant,» Google, [En línea]. Available: https://assistant.google.com/intl/es_es/. [Último acceso: 02 02 2021].
- [19] Apple, «Antes de que digas nada, Siri hace más que nunca,» Apple Inc., [En línea]. Available: <https://www.apple.com/es/siri/>. [Último acceso: 02 02 2021].
- [20] Microsoft 2021, «Cortana,» Microsoft 2021, 2021. [En línea]. Available: <https://www.microsoft.com/en-us/cortana>. [Último acceso: 02 02 2021].
- [21] Amazon, «Amazon Alexa,» Amazon, [En línea]. Available: <https://developer.amazon.com/es-ES/alexa>. [Último acceso: 02 02 2021].
- [22] J. Castilla, «Alexa (Echo) es un asistente personal muy versátil,» DomóticaDoméstica, 18 03 2016. [En línea]. Available: <http://www.domoticadomestica.com/alexa-echo-asistente-personal-versatil/>. [Último acceso: 28 01 2021].
- [23] SYSCOM, «Word Spotting,» SYSCOM, 2019. [En línea]. Available: <http://www.syscom.es/analitica-de-lenguaje-natural-2/word-spotting/>. [Último acceso: 23 01 2021].
- [24] Armada Española, «Submarinos clase "Galerna" (S-70),» [En línea]. Available: <https://armada.defensa.gob.es/ArmadaPortal/page/Portal/ArmadaEspañola/buquessubmarinos/prefLang-es/01s70>. [Último acceso: 28 01 2021].
- [25] C. Francescani, «US prisons and jails using AI to mass-monitor millions of inmate calls,» *abcNEWS*, 24 10 2019.
- [26] S. Ciurac, «How does Audio fingerprinting work,» EmySound, 12 06 2020. [En línea]. Available: <https://emysound.com/blog/open-source/2020/06/12/how-audio-fingerprinting->

- works.html. [Último acceso: 20 01 2021].
- [27] Mitel, «Keyword Spotting,» Mitel Networks, 2021. [En línea]. Available: <https://www.mitel.com/features-benefits/keyword-spotting>. [Último acceso: 23 01 2021].
- [28] IBM Cloud Education, «What is speech recognition?,» IBM, 02 09 2020. [En línea]. Available: <https://www.ibm.com/cloud/learn/speech-recognition#:~:text=Speech%20recognition%2C%20also%20known%20as,speech%20into%20a%20written%20format..> [Último acceso: 24 01 2021].
- [29] S. Ciurac, «GitHub/soundfingerprinting,» GitHub, 13 11 2011. [En línea]. Available: <https://github.com/AddictedCS/soundfingerprinting/blob/develop/README.md>. [Último acceso: 24 01 2021].
- [30] nemo.ai, «visely.io,» nemo.ai, 2020. [En línea]. Available: <https://visely.io/>. [Último acceso: 25 01 2021].
- [31] T. Dykstra y olprod, «Introducción a .NET,» Microsoft, 16 11 2020. [En línea]. Available: <https://docs.microsoft.com/es-es/dotnet/core/introduction>. [Último acceso: 26 01 2021].
- [32] T. Giannakopoulos, «pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis,» *PLOS one*, vol. 10, n° 12, 2015.
- [33] T. Giannakopoulos, «Github/tyiannak,» Github, [En línea]. Available: <https://tyiannak.github.io/about.html>. [Último acceso: 01 02 2021].
- [34] PyPI, «pyAudioAnalysis 0.3.6,» Python Software Foundation, [En línea]. Available: <https://pypi.org/project/pyAudioAnalysis/#history>. [Último acceso: 29 01 2021].
- [35] T. Lehinevych, N. Kokkinis-Ntrenis, T. Giannakopoulos y S. Konstantopoulos, «Discovering similarities for content-based recommendation and browsing in multimedia collections,» 2015.
- [36] J. M. Girard y J. F. Cohn, «Automated Audiovisual Depression Analysis,» National Center for Biotechnology Information, U.S National Library of Medicine, 19 11 2019. [En línea]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4539261/>. [Último acceso: 02 02 2021].
- [37] K. Tsikas, L. Watts, T. Giannakopoulos, A. Papangelis, R. Gatchel, V. Karkaletsis y F. Makedon, «A multimodal adaptive dialogue manager for depressive and anxiety disorder screening: A Wizard-of-Oz experiment,» de *8th ACM International Conference on Pervasive Technologies Related to Assistive Environments, PETRA 2015*, Corfu, Greece, 2015.
- [38] T. Giannakopoulos, G. Siantikos, N. Votsi y J. Pantis, «Automatic soundscape quality estimation using audio analysis,» de *8th Pervasive Technologies Related to Assistive Environments PETRA 2015*, 2015.
- [39] BitBite, «About BitBite,» BitBite Inc., [En línea]. Available: <https://bitbite.app/about/>. [Último acceso: 02 02 2021].
- [40] CMUSphinx, «Building an application with PocketSphinx,» CMUSphinx, [En línea]. Available: <https://cmusphinx.github.io/wiki/tutorialpocketsphinx/#pocketsphinx-api-core-ideas>. [Último acceso: 03 02 2021].

- [41] PyPI, «pocketsphinx 0.1.15,» Python Software Foundation, 2015. [En línea]. Available: <https://pypi.org/project/pocketsphinx/>. [Último acceso: 03 02 2021].
- [42] J. Vallinot Sánchez, «Módulo de transcripción VolP-Texto para una plataforma de interceptación legal de comunicaciones,» Universidad Carlos III de Madrid, 2012.
- [43] Tennyson, «Python Speech Recognition demo. Pocketsphinx Python module,» Youtube, 07 06 2020. [En línea]. Available: <https://www.youtube.com/watch?v=xRYIkLakufs&t=104s>. [Último acceso: 03 02 2021].
- [44] D. G. A. Povey, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, Motlíček, Petr, T. Qian, P. Schwarz, J. Silovsky, G. Stemmer y K. Vesely, «The Kaldi Speech Recognition toolkit,» 2012.
- [45] alphacep, «Vosk API,» GitHub, 01 09 2019. [En línea]. Available: <https://github.com/alphacep/vosk-api/graphs/contributors>. [Último acceso: 05 02 2021].
- [46] D. Povey, «Keyword Search in Kaldi,» KALDI, [En línea]. Available: <http://kaldi-asr.org/doc/kws.html>. [Último acceso: 04 02 2021].
- [47] G. Chen, O. Yilmaz, J. Trmal, D. Povey y S. Khudanpur, «Using proxies for OOV keywords in the keyword search task,» Center for Language and Speech Processing & Human Language Technology Center of Excellence, 2013.
- [48] J. Singh, «An introduction to audio processing and machine learning using Python,» opensource.com, 19 09 2019. [En línea]. Available: <https://opensource.com/article/19/9/audio-processing-machine-learning-python>. [Último acceso: 04 02 2021].
- [49] Facebook Research, «Flashlight,» GitHub Inc., 16 12 2018. [En línea]. Available: <https://github.com/facebookresearch/flashlight>. [Último acceso: 04 02 2021].
- [50] Tensorflow, «Lingvo,» GitHub, 19 08 2018. [En línea]. Available: <https://github.com/tensorflow/lingvo/graphs/contributors>. [Último acceso: 05 02 2021].
- [51] Colaboratory Research Google, «Introduction to Lingvo,» [En línea]. Available: <https://colab.research.google.com/github/tensorflow/lingvo/blob/master/codelabs/introduction.ipynb>. [Último acceso: 04 02 2021].
- [52] Alpha Cephei, «VOSK Installation,» [En línea]. Available: <https://alphacephei.com/vosk/install>. [Último acceso: 14 02 2021].
- [53] Alpha Cephei, «VOSK: Models,» [En línea]. Available: <https://alphacephei.com/vosk/models>. [Último acceso: 14 02 2021].
- [54] FFmpeg.org, «FFmpeg,» [En línea]. Available: <https://ffmpeg.org/>. [Último acceso: 14 02 2021].
- [55] online convert, «Convierte audio a WAV,» [En línea]. Available: <https://audio.online-convert.com/es/convertir-a-wav>. [Último acceso: 16 02 2021].
- [56] Wikipedia, «Word Error Rate,» Wikipedia, 06 06 2020. [En línea]. Available: https://es.wikipedia.org/wiki/Word_Error_Rate. [Último acceso: 26 02 2021].
- [57] Wikipedia, «Precision and recall,» Wikipedia, 22 02 2021. [En línea]. Available: https://en.wikipedia.org/wiki/Precision_and_recall. [Último acceso: 01 03 2021].

- [58] Cruzcampo, «Con mucho acento,» Cruzcampo, [En línea]. Available: <https://www.cruzcampo.es/con-mucho-acento>. [Último acceso: 22 02 2021].
- [59] H. L. Español, «Dinosaurios,» Youtube, [En línea]. Available: <https://www.youtube.com/watch?v=dTgofE8nrk0>. [Último acceso: 20 02 2021].
- [60] H. L. English, «DINOSAURS: all you need to know,» Youtube, [En línea]. Available: <https://www.youtube.com/watch?v=G3gXWDYpLAE>. [Último acceso: 20 02 2021].
- [61] J. Calero, «¿Cómo funciona? Motor diesel y motor de gasolina (Diesel vs Otto),» LISTEN NOTES, 28 07 2020. [En línea]. Available: <https://www.listennotes.com/podcasts/el-podcast-de-jf/c%C3%B3mo-funciona-motor-diesel-y-kcOe1qUnJ5r/>. [Último acceso: 25 02 2021].
- [62] Perito Informático Judicial, «Cómo peritar un archivo de audio,» GlobatikaLab, [En línea]. Available: <https://peritosinformaticos.es/como-peritar-un-archivo-de-audio/>. [Último acceso: 27 01 2021].
- [63] B. García Naranjo, «Trabajo fin de grado. Segmentación de audio broadcast,» Universidad Autónoma de Madrid, 2016.