UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de la Telecomunicación

TRABAJO FIN DE GRADO

Desarrollo de un sistema de Búsqueda de Palabras Clave en voz mediante Ejemplos

Sergio Cortés Álvarez
Tutor: Doroteo Torre Toledano

Abril 2020

Desarrollo de un sistema de Búsqueda de Palabras Clave en Voz mediante Ejemplos

AUTOR: Sergio Cortés Álvarez
TUTOR: Doroteo Torre Toledano

AUDIAS – Audio, Data Intelligence and Speech
Dpto. Tecnología Electrónica y Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Abril de 2020

Resumen (castellano)

Este Trabajo Fin de Grado, consiste y tiene como objetivo, la búsqueda de palabras clave en voz, es decir, el fin es buscar y detectar palabras en audios de larga duración a través de los datos de la voz.

Para ello, se han utilizado audios y consultas pertenecientes a la base de datos MAVIR, que anteriormente se han utilizado en evaluaciones de Albayzin Search on Speech.

A partir de esto, se ha desarrollado un sistema QbE STD (Query-by-Example Spoken Term Detection) con el que se han recuperado datos de un repositorio de voz a partir de la palabra de interés. A partir de los audios, las consultas y el reconocedor BUT (Brno University of Technology) para el reconocimiento de fonemas, se han extraído características cómo son los posteriorgramas fonéticos, que representan la probabilidad de cada fonema en cada instante de tiempo, con los que se ha obtenido una matriz de probabilidades a posteriori. Posteriormente, se ha desarrollado el algoritmo S-DTW (Subsequence – Dynamic Time Warping) cuyo objetivo es buscar un ejemplo de la palabra en el repositorio a través de un archivo de audio y hacerlo de forma rápida y precisa. A partir de dicho sistema, se han obtenido datos como el momento en el que se encuentra la consulta en el repositorio, la duración de dicha coincidencia, y la puntuación o score, que es la inversa del coste medio del camino óptimo encontrado para la consulta.

Finalmente, una vez obtenida la salida del sistema QbE STD en formato XML, se utilizará como entrada en el sistema scoring con el que obtendremos las métricas ATWV (Actual Term Weighted Vaue) y MTWV (Maximun Term Weighted Value), que son las métricas a optimizar.

Abstract (English)

This Final Thesis, consists of and aims, searching for keywords in voice, that is to say, the purpose is to search for and detect words in long-duration audios through voice data.

For this reason, audios and queries belonging to the MAVIR database have been used, that have previously been used in evaluations of Albayzin Search on Speech.

From this, a QbE STD (Query-by-Example Spoken Term Detection) system has been developed from which data has been retrieved from voice repository from the word of interest. From the audios, queries and the BUT (Brno University of Technology) recognizer for phoneme recognition, characteristics such as phonetic posteriorgrams have been extracted that represents the probability of each phoneme at each instant of time, with which a posteriori probability matrix has been obtained. Subsequently, the S-DTW (Subsequence – Dynamic Time Warping) algorithm has been developed whose objective is to search an example of the word in the repository through an audio file and do it quickly and accurately. From this system, data have such as the moment that the query is in the repository, the duration of the match, and the score, which is the inverse of the average cost of the optimal path found for the query, has been obtained.

Finally, once the output of the QbE STD system has been obtained in XML format, it will be used as input in the scoring system with which we will obtain the ATWV (Actual Term Weighted Vaue) y MTWV (Maximun Term Weighted Value) metrics, that are the metrics to optimize.

Palabras clave (castellano)

Consulta, repositorio, reconocedor de voz, matriz de probabilidades a posteriori, consulta mediante ejemplo, alineamiento temporal dinámico, algoritmos de reconocimiento.

Keywords (inglés)

Query, repository, voice recognizer, posteriori probability matrix, query by example, dynamic time warping, recognition algorithms.

Agradecimientos

Agradecer a toda mi familia, especialmente a mis padres, todo el apoyo y los ánimos que me han dado a lo largo de estos bonitos, pero a la vez, difíciles años.

Agradecer también a todos mis amigos, que siempre están ahí, todos los buenos ratos que me hacen pasar. Han conseguido entre otras muchas cosas, evadirme de todas las épocas de exámenes tan duras que se pasan a lo largo de esta carrera.

También, no me puede faltar mi grupo de compañeros de "teleco", que han pasado a formar una parte muy importante en mi vida. Que bien nos lo hemos pasado durante estos años y cuánto nos hemos reído…

Dar la gracias por supuesto a mi Lau, que durante estos últimos años me ha apoyado mucho y siempre ha estado a mi lado.

Asimismo, agradecer a mi tutor Doroteo, toda la ayuda que me ha facilitado para conseguir por fin, realizar este Trabajo de Fin de Grado.

Muchas gracias a todos.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

1.1 Motivación

A día de hoy en el mundo, se está viviendo un importante cambio en el área de la tecnología. La tecnología en general, está optimizando y depurando muchas de las actividades que realizamos diariamente, aportando un gran beneficio a todos los seres humanos, ahorrando entre otras cosas tiempo y esfuerzo.

En el caso de la comunicación, es el principal medio de transmisión de los seres humanos, por lo que su evolución es de vital importancia en el mundo actual. Es entre otras cosas, por lo que siempre me ha parecido muy interesante este sector, y el motivo principal por el que elegí este trabajo que está relacionado con uno de los medios de comunicación más utilizados por el ser humano, la voz.

Por otro lado, cabe destacar, que el reconocimiento de voz está teniendo cada vez mayor importancia, y que en la actualidad hay numerosas aplicaciones y dispositivos que utilizan esta forma de comunicación para mejorar sus prestaciones.

Haciendo referencia a este trabajo, hay que señalar, que debido a la importancia y al volumen que tiene mucha de la información audiovisual y multimedia que existe, cada vez hay más información almacenada en la red, por lo que es importante que aparezcan nuevos métodos para facilitar la búsqueda de dicha información [1].

Una utilidad que puede tener la búsqueda de palabras, y que me ha motivado a realizar este trabajo, es, por ejemplo, que podría ser usado por las fuerzas y cuerpos de seguridad del estado en el caso de que se quiera saber si en una grabación sospechosa se han dicho ciertas palabras.

La motivación principal de este trabajo es la consecución de un sistema QbE STD (Query-by-Example Spoken Term Detection) que sea capaz de buscar palabras de forma precisa, para que, en un futuro con numerosas evoluciones y desarrollos de estos sistemas de búsqueda, se facilite la obtención de contenidos audiovisuales de forma rápida, y sea algo que ayude y ahorre tiempo a las personas en su vida cotidiana.

1.2 Objetivos

El principal objetivo de este Trabajo de Fin de Grado es, como ya se ha mencionado anteriormente, el desarrollo de un sistema de búsqueda de palabras clave mediante la voz, analizar los resultados, e intentar ajustar ciertos parámetros para que el sistema sea lo más preciso posible. Para llegar a dicho objetivo, es necesario ir paso a paso obteniendo distintos propósitos.

El primer objetivo de este trabajo ha sido entender y analizar las evaluaciones y trabajos previos realizados. En este caso, he prestado especial atención a las evaluaciones Search on Speech de los años 2016 y 2018, en las que participa GMT-UVigo (Group of Multimedia Technology de la Universidad de Vigo), la Universidad San Pablo – CEU (Centro de Estudios Universitarios), y la Escuela Politécnica Superior de la Universidad Autónoma de Madrid.

El siguiente objetivo, ha sido generar el sistema que constituye la parte principal de este proyecto, el sistema QbE STD (Query-by-Example Spoken Term Detection). Se ha utilizado el reconocedor BUT (Brno University of Technology) para obtener ciertas características de los audios y consultas que tenía como entrada. En este caso, se han obtenido los llamados posteriorgramas fonéticos, con los que se han formado matrices que formarán la entrada para el algoritmo de búsqueda empleado, el S-DTW (Subsequence – Dynamic Time Warping).

El siguiente objetivo ha consistido en desarrollar dicho algoritmo. Su finalidad, es que a partir de las entradas de una "matriz de posteriors" de la consulta y otra del repositorio, se obtenga el momento en el que se encuentra la consulta en el audio, si es que se encuentra, y la puntuación de dicha búsqueda, a partir de ciertos umbrales específicos.

A continuación, hay que generar la salida del sistema QbE STD en formato XML. Para ello, hay que ir probando con varias consultas e ir generando el archivo que posteriormente servirá como entrada en el sistema de puntuación.

Para finalizar, una vez obtenida la salida del sistema scoring y comprobado que no hay ningún error en ella, habrá que ajustar los parámetros del sistema QbE STD para que los resultados del ATWV (Actual Term Weighted Vaue) y MTWV (Maximum Term Weighted Vaue) sean los mejores posibles. Para ello, se puede comparar con los resultados publicados de la evaluación Search on Speech Albayzin del año 2016.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Introducción.** En dicho capítulo, se explican las motivaciones que me han llevado a la realización de este trabajo. También se aclaran los principales objetivos que pretende dicho Trabajo de Fin de Grado.

- **Estado del Arte.** En este apartado, se pretende situar al reconocimiento de voz en la actualidad y resaltar la importancia que tiene. Se describe como se produce la voz en los seres humanos y lo que son los fonemas, que van a suponer una parte fundamental en el trabajo. Se explica resumidamente la historia del reconocimiento de voz, y se describe el esquema principal de los reconocedores de voz. Por último, dicha sección nos sitúa un poco en el sistema que va a ser desarrollado, y describe de forma breve las evaluaciones que se han llevado a cabo anteriormente y en las que se va a basar el trabajo.

- **Diseño.** En esta sección, se describirá el esquema del sistema que va a ser desarrollado. Nos situará un poco para poder entender lo que ha sido implementado y podremos diferenciar cada una de sus partes y elementos que forman parte del sistema. Se explicará de dónde vienen los datos que van a ser utilizados y los que han sido elegidos.

- **Desarrollo.** Es la parte más laboriosa del trabajo. Posiblemente la parte más subjetiva. Se explican todas las partes desarrolladas del algoritmo de reconocimiento, para finalmente obtener los resultados que serán analizados.

- **Integración, pruebas y resultados.** Todo sistema implementado, necesita de una batería de pruebas y unos resultados finales para intentar optimizarlo. En este apartado, se explican las pruebas que se han realizado y el análisis que se ha llevado a cabo de los resultados obtenidos. Se intentará averiguar los parámetros con los que se obtiene un mejor rendimiento.

- **Conclusiones y trabajo futuro**. En este último capítulo del trabajo, se describirán las conclusiones que se han sacado del desarrollo y de los resultados obtenidos del sistema, y se hipotetizará sobre algún trabajo futuro que se podrá llevar a cabo.

2 Estado del arte

2.1 Reconocimiento de voz en la actualidad. Tecnologías del habla

La tecnología y el reconocimiento de voz están viviendo una importantísima evolución en la actualidad. Las tecnologías del habla comienzan a ser una realidad en el mundo, y cada vez muchas más personas hacen uso de dichas técnicas. El habla, es el medio de comunicación más usado por el ser humano, por lo que tiene sentido que cada vez haya más avances.

Gracias al reconocimiento de voz y a sus utilidades, podemos obtener numerosos beneficios entre los que destaca principalmente el ahorro de tiempo y esfuerzo ya que en lugar de buscar algo de forma manual o escrita, simplemente con la voz podemos encontrarlo. Simplemente pulsando un botón y expresándonos, podemos disponer de aquello que buscamos sin necesidad de realizar más acciones.

Por otro lado, los sistemas de reconocimiento de voz son cada vez más precisos y más efectivos. Es otro de los motivos por los que cada vez más personas utilizan estas tecnologías. Cada vez son más rápidos y más precisos.

A continuación, se muestra una gráfica en la que podemos observar el reconocimiento de palabras a lo largo de los años por Google (de acuerdo a un estudio elaborado por el departamento de ingeniería de Baidu) [2]. Se puede observar, que, con el paso del tiempo, los reconocedores de voz son capaces de entender un mayor número de palabras.

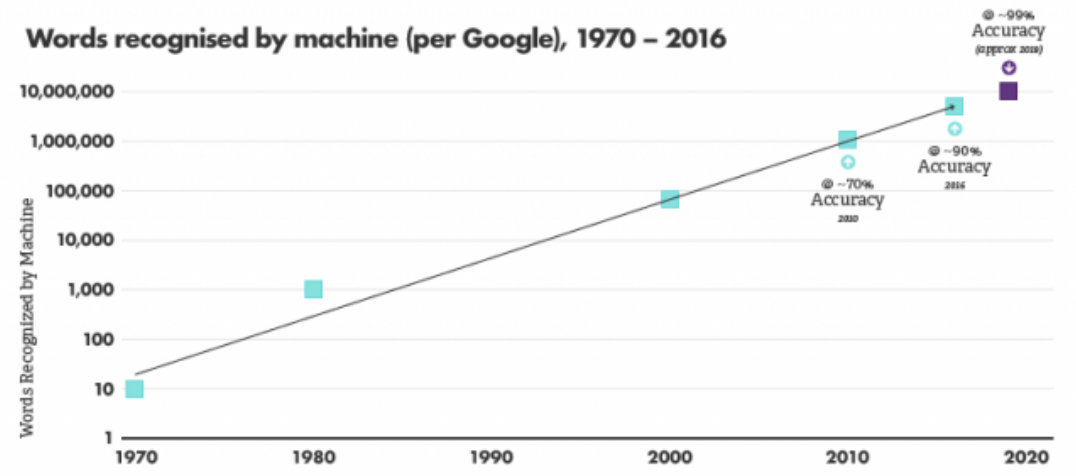


Figura 1: Gráfica reconocimiento de palabras [2]

Es por todo esto, entre otras cosas, que cada vez se hace más importante la investigación y el desarrollo en el reconocimiento de voz y en las tecnologías del habla.

Dentro del reconocimiento de voz, la búsqueda de palabras cada vez tiene una mayor importancia, ya que, como se ha comentado con anterioridad, cada vez hay más contenidos de imagen, audio y video en la red.

2.2 La producción de la voz y los fonemas

Antes de empezar con el grueso del trabajo, es importante conocer como produce la voz el ser humano y lo que son los fonemas.

La voz es el sonido que se produce al pasar el aire por el aparato fonador. Para que se genere la voz humana, entran en funcionamiento numerosos órganos del cuerpo humano: órganos de respiración (pulmones, tráquea…), órganos de fonación (laringe, cuerdas vocales…), y los órganos de articulación (lengua, dientes, labios…). La señal de voz, forma mediante sonidos el habla humana, es decir, produce la comunicación entre las personas. Las cuerdas vocales son las encargadas de producir sonidos sonoros (cuando se encuentran en tensión al atravesarlas el aire), mientras que los sonidos sordos se generan cuando las cuerdas vocales no vibran y se produce una oclusión que ocasiona una interrupción del flujo de aire.

El tracto vocal, está formado por tres cavidades: cavidad faríngea (después de la laringe), cavidad oral (labios, dientes…) y cavidad nasal (encima del paladar, se divide en las fosas nasales). La laringe excita estas cavidades y genera frecuencias de resonancia llamadas formantes, que se encargan de la diferenciación de sonidos.

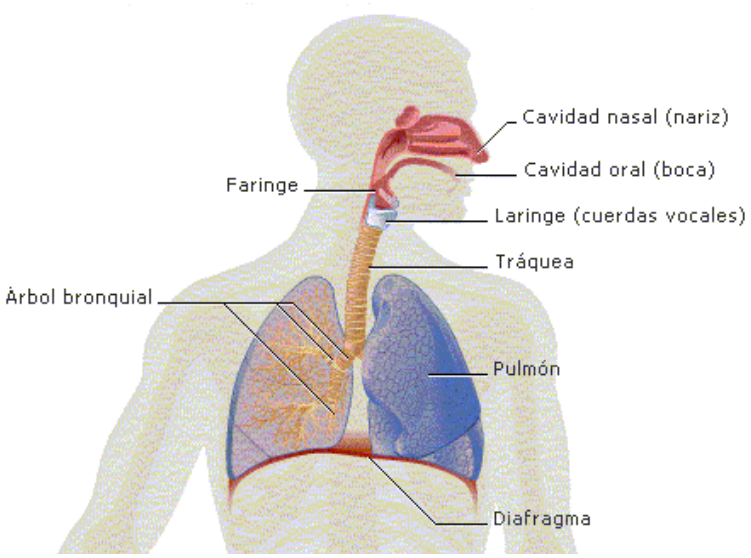


Figura 2: Aparato Fonador Humano

Los fonemas son las unidades fonológicas de la lengua, es decir, son cada uno de los distintos sonidos que se producen en el habla. Cabe destacar, que el español tiene 30 fonemas divididos en fonemas vocálicos y fonemas consonánticos. Pueden producir un cambio de significado en una palabra [3]. Por otro lado, los alófonos representan las pronunciaciones reales de un fonema. Un fonema puede disponer de varios alófonos (lo podemos ver por ejemplo con el fonema /b/ en las palabras barco y sabio).

2.3 Los sistemas de reconocimiento de voz

En este apartado se hablará de la evolución de los sistemas de reconocimiento de voz, del esquema de dichos sistemas, y de los algoritmos principales para reconocimiento.

2.3.1 Evolución del reconocimiento de voz

La información relativa a esta sección ha sido obtenida a partir de la referencia [5].

En 1920 aparece la primera máquina capaz de reconocer el habla. Era un perro llamado Rex que se movía cuando detectaba un tono de 500 hz. Dicha frecuencia es igual a la del primer formante de la pronunciación de la palabra Rex.

En 1952 aparece la primera computadora capaz de reconocer voz. Reconocía dígitos del 1 al 9. Posteriormente, se creó un sistema que llegó a reconocer vocales y consonantes.

A lo largo de los años 60, aparece en Japón el sistema DTW (Dynamic Time Warping), del que se hablará más adelante con más profundidad. Por otro lado, la Universidad de Carnegie Mellon lleva a cabo una investigación continua en el reconocimiento de voz.

Durante el año 1969 nace ARPA (Agencia de Proyectos para la Investigación Avanzado de Estados Unidos) que posteriormente pasó a denominarse DARPA. Esta agencia, conectó cuatro sistemas en una red que se denominó APRANET cuya misión era mantener las comunicaciones en caso de guerra. Dicha red permitía la entrada y salida de conexiones, y que los usuarios pudieran conectarse desde cualquier parte de la red. APRANET se extendió por el mundo académico y aparecieron nuevas redes. Con la unificación de dichas redes, se presupone que nace Internet [4].

Durante los años 70 hubo una importante evolución. Se empezaron a reconocer palabras aisladas. IBM (International Business Machines Corporation) empezó a desarrollar proyectos de reconocimiento con más vocabulario. Se desarrolló el sistema Harpy cuya arquitectura se parecía a la de los sistemas actuales de reconocimiento. Es a lo largo de estos años donde aparecen los modelos ocultos de Markov (HMM según la nomenclatura inglesa), aunque fue en los años 80 donde tuvieron un mayor desarrollo. Más adelante se explicará con más detenimiento.

En los años 80, se introdujeron las redes neuronales en el reconocimiento de la voz.

A lo largo de los años 90 aparecen los primeros ordenadores y las primeras aplicaciones. Aparecen también sistemas de dictado.

En los años 2000 aparece VoiceXML, que es un estándar de documentos donde se especifican medios interactivos entre humanos y ordenadores.

A partir del año 2010 se comienzan a aplicar sistemas de aprendizaje profundo basados en redes neuronales, que constituyen el estado del arte actual.

2.3.2 Esquema de los sistemas de reconocimiento de voz

Como aparece en la siguiente figura, los principales pasos en un sistema de reconocimiento de voz son los siguientes:

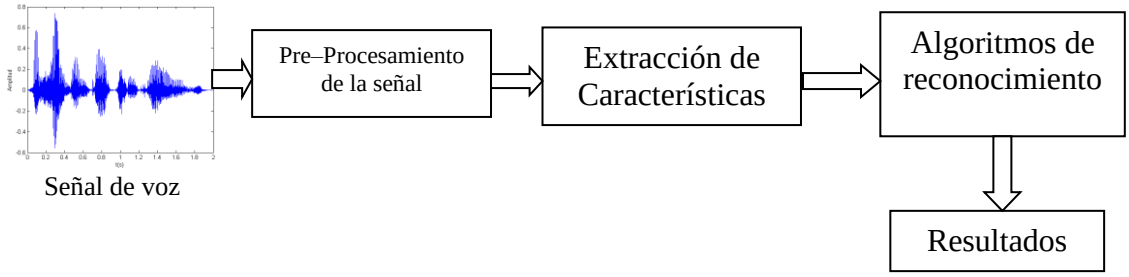


Figura 3: Esquema sistema reconocimiento de voz

- **Señal de voz:** Se adquiere la señal voz procedente de cualquier fuente de sonido.

- **Pre-Procesamiento de la señal:** El objetivo del pre-procesamiento es que la señal de voz pueda ser procesada adecuadamente por los algoritmos de reconocimiento. Para ello se llevan a cabo diferentes medidas como la eliminación del offset dc (tensión continua), la eliminación del mayor ruido posible, se realiza la detección del principio y del fin de la palabra, preénfasis (aumento del nivel de altas frecuencias en proporción al ruido de dichas frecuencias) …

- **Extracción de características:** En esta etapa, se obtienen distintos parámetros importantes de la señal que facilitarán su análisis. Destacan los posteriorgramas fonéticos, los coeficientes cepstrales de Mel y los modelos de predicción lineal. Más adelante se explicará con más detenimiento.

- **Algoritmos de reconocimiento:** El objetivo principal de dichos algoritmos es realizar el reconocimiento de las señales de entrada con respecto a la base de datos que se va a utilizar. Los principales algoritmos de reconocimiento son: DTW (Algoritmo de reconocimiento mediante alineamiento temporal dinámico) y su variación S-DTW (Subsequence – Dynamic Time Warping), los modelos ocultos de Markov (HMM), y las redes neuronales (NN).

- **Resultados:** A partir de los cuales podemos obtener porcentajes y conclusiones de cómo funciona cada algoritmo, de las palabras que se han encontrado, falsas alarmas, ocurrencias no detectadas…

2.3.2.1 Extracción de características

La extracción de características es un paso muy importante en los sistemas de reconocimiento de voz, ya que dichas características serán la entrada de los algoritmos de reconocimiento. En esta etapa se obtienen un conjunto de parámetros que contienen información muy destacada de la señal de voz que la representan de una forma eficiente. A continuación, se explican los métodos de extracción de características más importantes.

2.3.2.1.1 Coeficientes Cepstrales de las frecuencias de Mel (MFCC)

Los Coeficientes Cepstrales de las frecuencias de Mel (MFCC) están basados en la percepción auditiva humana y muestran características y parámetros asociados al tracto vocal. El objetivo, es obtener una representación compacta y eficiente para entender la señal de voz de forma clara.

Sobre los coeficientes cepstrales de Mel hay que destacar que vienen originados por la transformada de Fourier, aunque lo que de verdad hay que destacar es que las bandas de Mel están representadas logarítmicamente por medio de la escala de Mel, donde el punto de referencia se define igualando un tono de 1000 Hz a 1000 mels.

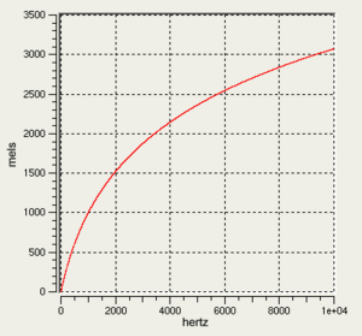


Figura 4: Escala de Mel

A continuación, tal y como se detalla en la siguiente figura, se explica paso a paso el procedimiento para la obtención de los coeficientes cepstrales de Mel:

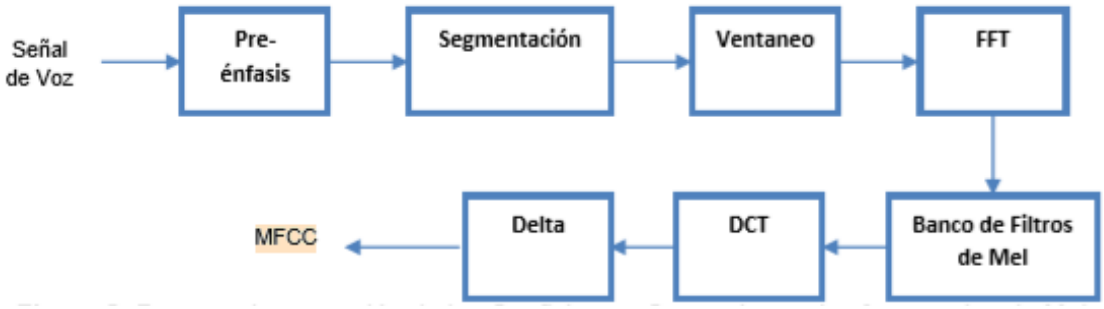


Figura 5: Diagrama de bloques para obtener los MFCC [6]

- **Pre-énfasis:** El principal objetivo de dicho filtro es acentuar las altas frecuencias de la señal, se aumentará el valor de la señal en dichas frecuencias.

- **Segmentación:** La señal es segmentada en bloques de entre 10 y 30 ms.

- **Ventaneo:** Se utiliza principalmente para evitar las discontinuidades al principio y al final del segmento. La ventana más utilizada en la venta Hamming.

- **FFT:** Se aplica la Transformada de Fourier a cada segmento para pasar del dominio del tiempo al dominio de la frecuencia.

- **Banco de Filtros Mel:** Se aplican una serie de filtros triangulares separados de la misma forma que se encuentran en la escala de Mel. Se transforman las frecuencias lineales en frecuencias de Mel.

- **DCT (Transformada del Coseno Discreta):** Su objetivo es comprimir y concentrar la información en pocos coeficientes, de manera que se eliminan las altas frecuencias suavizando así la señal. El resultado es lo que se denomina coeficientes cepstrales de Mel, MFCC.

- **Delta:** Debido a que la señal cambia a lo largo del tiempo, es necesario añadir estas características. Es por ello que se calcula la derivada en el tiempo de los coeficientes cepstrales de Mel, con lo que se obtiene como resultado los Delta MFCC.

2.3.2.1.2 Modelos de Predicción Lineal (LPC)

De acuerdo a la referencia [7], el objetivo principal de los Modelos de Predicción Lineal, es predecir la muestra presente a partir de una combinación lineal de las muestras pasadas. Es una de las técnicas más utilizadas. Representa el espectro de una señal de voz que se aproxima mucho a la envolvente espectral del tracto vocal.

La siguiente figura muestra el modelo en el que se basa LPC (Linear Predictive Coding):

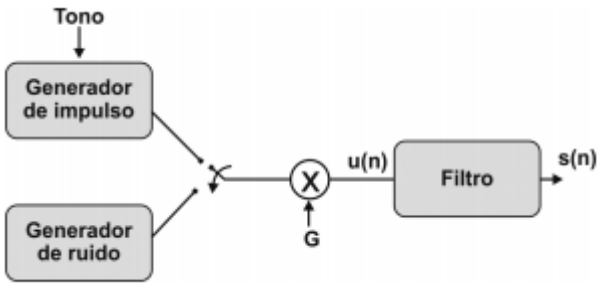


Figura 6: Modelo LPC

El valor actual de la señal se puede obtener como una combinación lineal de varias muestras anteriores. Su fórmula matemática sería:

$$s[n] = \sum a_k s[n-k] + Gu[n].$$

donde G es la ganancia, a_k serían los coeficientes de predicción lineal LPC y *u(n)* es la señal de excitación.

Entre los métodos más utilizados para obtener los coeficientes de predicción lineal destaca el algoritmo Levinson-Durbin, cuyo objetivo es obtener dichos coeficientes a partir del error de predicción.

2.3.2.1.3 Posteriorgramas Fonéticos

La información correspondiente a este apartado, ha sido adquirida a partir de la referencia [8].

Los posteriorgramas fonéticos, son constituidos por la probabilidad de cada fonema en cada instante de tiempo. Es la técnica que ha sido empleada en el desarrollo del sistema QbE STD (Query-by-Example Spoken Term Detection) de este trabajo.

Dichos posteriorgramas, se calculan para cada unidad temporal tanto de la consulta a realizar como del repositorio en el que se va a buscar, de tal manera que se obtienen dos vectores de igual dimensión, cuya longitud será la del número de fonemas totales más una unidad de no habla para representar fenómenos como la risa, el ruido, el silencio …

Cada fonema se representa con tres estados, que son el principio, el fin y el centro del fonema. La suma de las tres probabilidades formará la probabilidad total del fonema.

En este caso los posteriorgramas han sido obtenidos a partir del reconocedor de BUT (Brno University of Technology). Se explicará más adelante.

2.3.2.2 Algoritmos de Reconocimiento

Una vez extraídas las características de cada una de las unidades temporales tanto del audio como del repositorio, el siguiente paso lo conforman los algoritmos de reconocimiento de voz. Su principal objetivo es reconocer el audio o consulta en el repositorio o la base de datos en cuestión. A continuación, se explican alguno de ellos.

2.3.2.2.1 DTW (Algoritmo de reconocimiento mediante alineamiento temporal dinámico)

El contenido de esta sección se basa en [10].

El objetivo del algoritmo DTW (Dynamic Time Warping) es medir la similitud entre dos secuencias que pueden variar en el tiempo, en el espacio y en velocidad.

DTW, es un método que permite encontrar a través de una distorsión no lineal, con respecto a una variable que suele ser el tiempo, la mejor correspondencia entre dos secuencias.

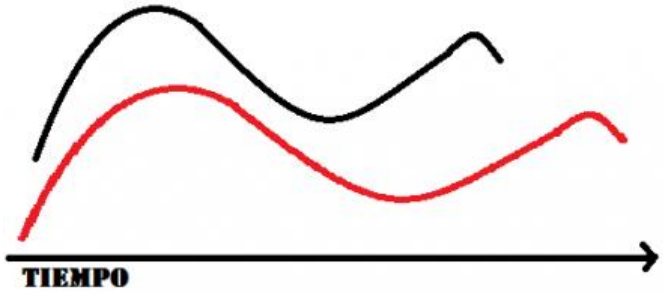


Figura 7: Dos secuencias para alinear [9]

La idea de este alineamiento temporal, es que el eje de la señal de búsqueda se comprima y se expanda de forma no lineal para alinear los vectores de características entre la señal de prueba y la de referencia.

Se suele crear una tabla o matriz para calcular lo que se conoce como camino de alineamiento. Se coloca una señal en un eje y la otra en el otro, y se va calculando la distancia entre cada punto de la señal de búsqueda con cada punto de la señal de referencia. El cálculo, se lleva a cabo a partir de la siguiente fórmula:

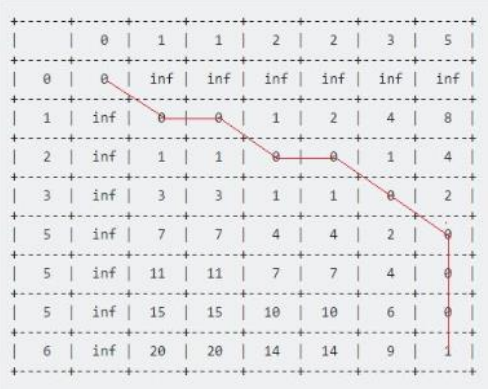
$$Tabla[i][j] = d(i,j) + min(Tabla[i-1][j], Tabla[i-1][j-1], Tabla[i][j-1]),$$

donde d en nuestro caso sería el valor de la matriz de coste en el punto *(i,j)*, y Tabla es la matriz que vamos formando a partir de las dos secuencias. Otro posible valor de d sería la distancia entre los dos puntos de cada señal *(d(x,y) = |x-y|)*. En el caso de tener por ejemplo dos vectores $x = \{1,2,3,5,5,5,6\}$ e $y = \{1,1,2,2,3,5\}$ la matriz quedaría de la siguiente forma, siendo la primera fila y la primera columna todo infinito debido a que los puntos no se comparan con nada:

	0	1	1	2	2	3	5
0	0	inf	inf	inf	inf	inf	inf
1	inf	0	0	1	2	4	8
2	inf	1	1	0	0	1	4
3	inf	3	3	1	1	0	2
5	inf	7	7	4	4	2	0
5	inf	11	11	7	7	4	0
5	inf	15	15	10	10	6	0
6	inf	20	20	14	14	9	1

Figura 8: Matriz de cálculo del camino de alineamiento

Para calcular el camino de alineamiento, empezaríamos en la posición de abajo a la derecha y nos iríamos moviendo al mínimo valor entre las posiciones de arriba, de la izquierda y de la diagonal de arriba a la izquierda, así hasta llegar a la posición (0,0).



	0	1	1	2	2	3	5
0	0	inf	inf	inf	inf	inf	inf
1	inf	0	0	1	2	4	8
2	inf	1	1	0	0	1	4
3	inf	3	3	1	1	0	2
5	inf	7	7	4	4	2	0
5	inf	11	11	7	7	4	0
5	inf	15	15	10	10	6	0
6	inf	20	20	14	14	9	1

Figura 9: Camino de alineamiento

Según el camino de alineamiento, podemos interpretar lo que pasó con la señal temporalmente.

Un movimiento horizontal representa que la señal se aceleró, un movimiento vertical representa que la señal se desaceleró, y un movimiento en diagonal representa que durante ese tiempo las señales fueron iguales.

2.3.2.2.2 S-DTW (Subsequence – Dynamic Time Warping)

Este algoritmo, es una modificación del algoritmo DTW (Dynamic Time Warping), cuyo objetivo es encontrar una señal de menor longitud en una señal de mayor longitud. Se emplea en los casos en los que las secuencias a comparar tienen longitudes muy diferentes. Este es el algoritmo de reconocimiento empleado en el sistema QbE STD (Query-by-Example Spoken Term Detection) desarrollado en este trabajo. En siguientes apartados se explicará con más detalle cómo se ha desarrollado.

2.3.2.2.3 Modelos Ocultos de Markov (HMM)

El modelo oculto de Markov es un modelo estadístico donde su objetivo principal es determinar los parámetros desconocidos u ocultos de la cadena a partir de los parámetros observables.

Un proceso de Markov es un proceso estocástico que sirve para representar secuencias de variables aleatorias entre sí. En otras palabras, la probabilidad del siguiente estado sobre una secuencia completa depende de estados previos al estado actual [11].

En el reconocimiento de voz se emplea para configurar una frase completa, una palabra o un fonema.

En la siguiente figura se muestra una cadena de Markov con tres estados donde {1,2,3} representan los estados del modelo y $a_{(i,j)}$ es la probabilidad de transición del estado i al estado j.

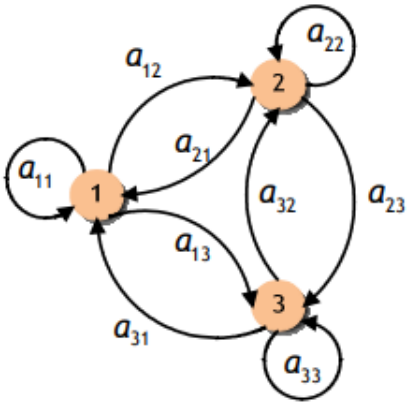


Figura 10: Cadena de Markov con 3 estados [12]

Por otro lado, las cadenas de Markov se corresponden con una serie de estados por los que va pasando la secuencia. De esta manera se puede calcular la probabilidad de que se recorran ciertos estados en el orden que se quiera. Por ejemplo, en este caso y con este modelo podríamos calcular la probabilidad de que se produjera la secuencia {2,3,3,2,1} cuyo cálculo sería:

$$p(O|modelo) = p(2,3,3,2,1|modelo) = p(3) * p(3|2) * p(3|3) * p(2|3) * p(1|2)$$

En estos procesos de Markov, cada estado es un evento observable. Estos procesos se pueden desarrollar de manera que la observación sea aleatoria y depende de cada uno de estos estados del sistema. Este modelo es el que se conoce realmente como Modelo Oculto de Markov (HMM).

De esta manera, los estados están ocultos y son observados a través de otros procesos. Por lo que en base a la Figura 10 habría unas salidas observables y unas probabilidades de salida.

Por lo tanto, se dice que un Modelo Oculto de Markov no depende solamente de la observación de los estados, sino que es una serie de procesos estocásticos que se producen en cada estado.

2.3.2.2.4 Redes Neuronales (NN)

Las Redes Neuronales Artificiales (ANN) son un sistema inspirado en las redes neuronales biológicas y consiste en un conjunto de unidades llamadas neuronas artificiales conectadas entre sí para transmitir señales. La información atraviesa dicha red neuronal produciendo unos valores de salida [13].

Cada enlace entre la entrada y la neurona tiene un peso. Este peso se multiplica con el valor de la entrada y el resultado se le pasa a la neurona. Cada neurona también tiene un peso y una función de activación (función que define la salida a través de la entrada). Los pesos en los enlaces pueden aumentar o disminuir las funciones de activación de las neuronas adyacentes.

La característica principal de las redes neuronales es que son capaces de adquirir el conocimiento a partir del entorno, del estudio y de la experiencia, es decir, las neuronas en función de ciertas entradas se ajustan para producir salidas adecuadas. Se puede decir que llevan a cabo un aprendizaje automático.

Dentro de las Redes Neuronales Artificiales (ANN) podemos encontrar las Redes Neuronales Profundas (DNN) que tiene múltiples capas entre la entrada y la salida.

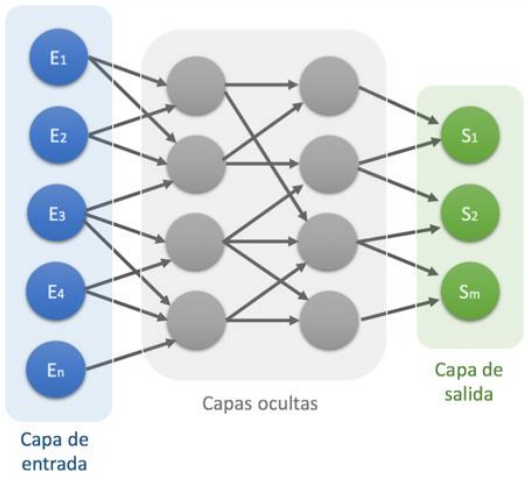


Figura 11: Red Neuronal Profunda (DNN) [14]

La figura muestra una red neuronal profunda (DNN). En el caso de que fuera una red Neuronal Artificial (ANN) convencional, sólo tendría una capa oculta como máximo.

2.4 Sistemas de detección de términos hablados de consulta por ejemplo (QbE STD)

El sistema principal que ha sido desarrollado en el trabajo ha sido el QbE STD (Query-by-Example Spoken Term Detection). Antes de comenzar con el diseño y el desarrollo, conviene entender en qué consisten estos sistemas.

Como ya se ha mencionado en los apartados anteriores, el sistema QbE STD tiene como objetivo buscar en un repositorio o audio muy grande una consulta.

QbE STD podemos dividirlo en dos términos:

- QbE (Query by Example): Consulta por ejemplo. Se destina a buscar una consulta, que es un ejemplo de un audio o un segmento de voz, en otro objeto.

- STD (Spoken Term Detection): Detección de términos hablados. Su principal objetivo es encontrar palabras dentro de un repositorio de audio. También se le suele llamar STD basada en texto, ya que su entrada es un texto. En el QbE STD, la entrada es de voz.

El sistema QbE STD tiene 3 enfoques diferentes: métodos basados en transcripciones de palabras o subpalabras para la consulta, métodos basados en la coincidencia de características y enfoque híbrido [15]. A continuación, se explican estos métodos más detalladamente:

- **Basados en transcripciones de palabras o subpalabras para la consulta:** esto métodos, generalmente, emplean STD basada en texto, por ello necesitan la transcripción de la consulta en palabras o subpalabras. Se emplean diferentes algoritmos de reconocimiento como son los modelos ocultos de Markov, algoritmo de reconocimiento mediante alineamiento temporal dinámico (DTW - Dynamic Time Warping)) …

- **Basados en la coincidencia de características:** Se extraen una serie de características tanto de la consulta como del repositorio. En dicha extracción de características destacan los posteriorgramas fonéticos, que como ya se ha explicado anteriormente es la probabilidad de cada fonema en cada instante de tiempo y que cada uno de ellos se divide en tres estados (principio, centro y fin del fonema). En este caso, se emplea el algoritmo DTW o alguna variante de éste como es el algoritmo S-DTW. Con este enfoque, no es necesario el conocimiento previo del lenguaje que tiene los datos del habla. Puede desarrollarse de una manera eficaz independientemente del lenguaje. Dicho método es el que se ha implementado en este trabajo.

- **Enfoque híbrido:** Este enfoque consiste en combinar los dos métodos individuales. Por ejemplo, la fusión entre una regresión logística de localización de palabras clave y sistemas basados en DTW utilizando reconocedores de fonemas.

El esquema para el sistema QbE STD basado en la coincidencia de características y que se ha utilizado en este trabajo es el siguiente:

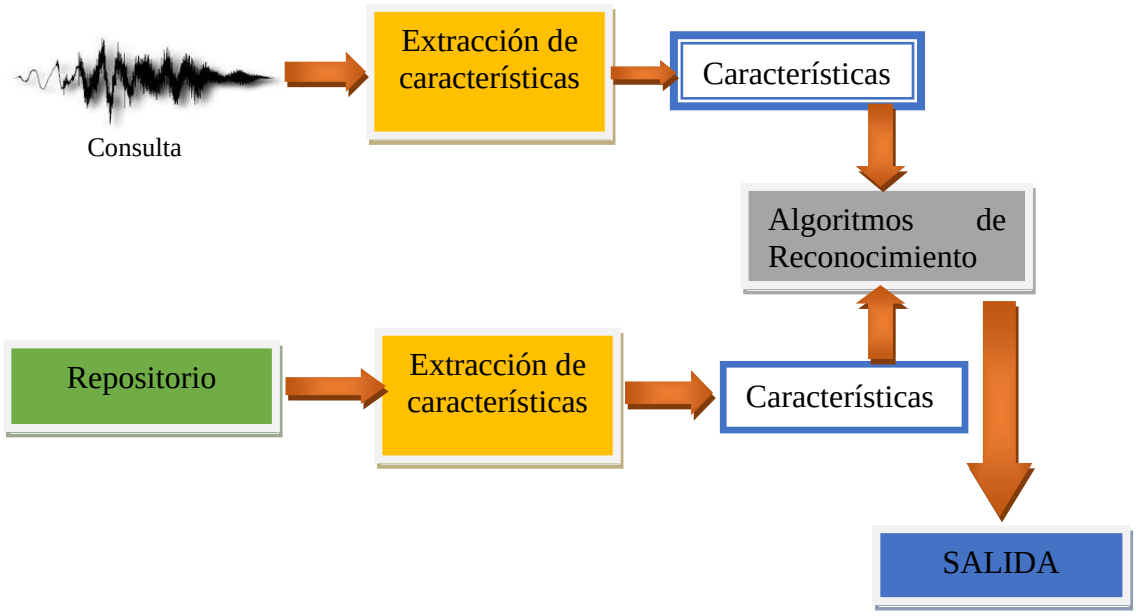


Figura 12: Esquema de un Sistema QbE STD

Como se puede observar en la figura, disponemos de una consulta (palabra a buscar) y un repositorio (audio donde se buscará la consulta). Se extraen las características tanto de uno como de otro (en nuestro caso van a ser los posteriorgramas fonéticos). Dichos posteriorgramas serán la entrada para el algoritmo de reconocimiento empleado. En nuestro sistema QbE STD implementado, dicho algoritmo es el S-DTW. Por último, se obtiene la salida, que será información de dónde se encontró la palabra, la duración de la coincidencia y algunos otros parámetros.

2.5 Evaluaciones anteriores de Albayzin

La búsqueda de palabras mediante voz tiene cada vez una mayor importancia en nuestras vidas, por lo que cada vez es normal que haya un mayor número de desarrollos y evaluaciones.

Es aquí donde aparecen las evaluaciones de Albayzin, que llevan a cabo la búsqueda de palabras de audio dentro de un audio mucho más grande y cuyo objetivo es evaluar el progreso de los sistemas de búsquedas de palabras en voz, especialmente de QbE STD (Query-by-Example Spoken Term Detection). Se han hecho evaluaciones en los años 2012, 2014, 2016 y 2018, en las que ha participado el Grupo de investigación AUDIAS.

A lo largo de este trabajo, se va hacer referencia a las evaluaciones de Albayzin Search On Speech del año 2016 y del año 2018, en las que participa GMT-UVigo (Group of Multimedia Technology de la Universidad de Vigo), la Universidad San Pablo – CEU (Centro de Estudios Universitarios), y la Escuela Politécnica Superior de la Universidad Autónoma de Madrid, ya que son las investigaciones y evaluaciones en las que se ha basado el trabajo y con las que se han comparado los resultados.

2.5.1 GMT-U Vigo Albayzin 2016

El sistema QbE STD presentado por el Grupo GMT (Group of Multimedia Technology) de la Universidad de Vigo en la evaluación ALBAYZIN 2016 consistió en la fusión de tres sistemas:

- Extracción de características: para representar las consultas y los repositorios a través de vectores. Se utilizan diferentes características para sacar un conjunto para la consulta y otro para el repositorio con vectores del mismo tamaño. Dichas características eran distintas probabilidades, posteriorgramas…

- Algoritmo de búsqueda: Se lleva a cabo utilizando el algoritmo S-DTW (Subsequence – Dynamic Time Warping), que es una evolución del DTW (Dynamic Time Warping). Dicho algoritmo utiliza como entrada los dos conjuntos de características y tiene como métrica el coeficiente de correlación de Pearson.

- Fusión: Su objetivo es combinar las puntuaciones de las salidas de los subsistemas para la normalización de la puntuación.

Los resultados de dicha evaluación se obtuvieron a partir de diferentes parámetros como son el valor promedio ponderado por término (ATWV - Actual Term Weighted Value), el valor ponderado del término máximo (MTWV - Maximum Term Weighted Value), la probabilidad de falsa alarma, y la probabilidad de detección errónea.

2.5.2 Albayzin 2018 Search On Speech

Esta evaluación expone el sistema presentado por el equipo de investigación AUDIAS-CEU en el año 2018. En dicha evaluación, se desarrolla el algoritmo QbE STD ya que, según resultados de evaluaciones anteriores, se considera que podría ser el sistema de reconocimiento de voz más preciso. Lo más destacable de dicha evaluación es lo siguiente:

- Se utiliza material procedente de tres bases de datos (MAVIR, COREMAH y una base de datos con programas de RTVE)

- A partir del reconocedor de BUT (Brno University of Technology), se extraen los posteriorgramas fonéticos, que son las probabilidades de cada fonema en cada instante de tiempo. A partir de ellos, se forma la matriz de probabilidades a posteriori, en la que se representa cada fonema en cada instante de tiempo con tres probabilidades.

- Se desarrolla el algoritmo S-DTW. Dicho algoritmo toma como entrada las matrices de posteriorgramas fonéticos de la consulta y del repositorio. Se forma la matriz de costes comparando par a par de ambas matrices. Se compone una matriz de coste acumulado, donde podremos encontrar los caminos óptimos, buscando los valores mínimos, de las consultas a buscar.

- Por último, se obtuvieron los resultados finales a partir de los términos de ATWV y MTWV, que se compararon con los obtenidos en evaluaciones anteriores y se sacaron numerosas conclusiones.

3 Diseño

Como ya se ha comentado en apartados anteriores, el sistema que ha sido desarrollado en este Trabajo de Fin de Grado ha sido el QbE STD (Query-by-Example Spoken Term Detection). En dicho sistema, se han utilizado los posteriorgramas fonéticos y se ha desarrollado un algoritmo S-DTW (Subsequence – Dynamic Time Warping) como algoritmo de reconocimiento. Posteriormente se han analizado los resultados y se han sacado conclusiones. A continuación, se va a explicar el diseño que se ha llevado a cabo en este trabajo.

3.1 Esquema de diseño

La siguiente figura, muestra el proceso que se ha llevado a cabo a lo largo de este trabajo e investigación.

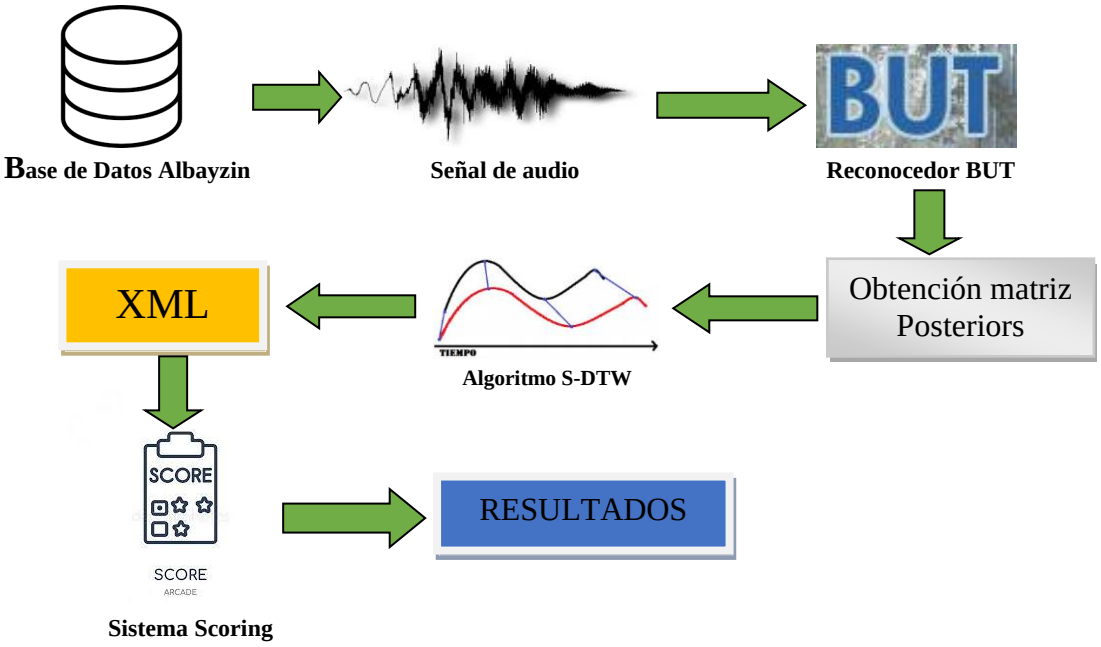


Figura 13: Esquema de Diseño

En resumen, la información se obtiene a partir de la base de datos Albayzin, de donde se obtienen las palabras, que serán las consultas de nuestro sistema, y los repositorios o audios grandes donde buscaremos dichas palabras.

Se obtiene la señal de audio, a partir de la cual obtendremos sus características (posteriorgramas fonéticos) gracias al reconocedor BUT (Brno University of Technology). Una vez obtenidas dichas probabilidades de fonemas, se crearán las matrices de posteriors de cada uno de los audios a comparar. Estas matrices serán la entrada del algoritmo S-DTW.

Posteriormente se generará un archivo XML donde se puede ver de todas las querys el momento en el que se han encontrado, el tiempo en el que coinciden, la puntuación… Dicho archivo XML será la entrada al sistema scoring que nos dará un resultado final con el que sacaremos conclusiones.

3.2 Origen de la información

Lo información relativa a esta sección se obtiene de la referencia [16].

Para desarrollar este trabajo, todos los audios (consultas y repositorios) empleados, han sido utilizados anteriormente en la evaluación ALBAYZIN. En dicha evaluación, se han utilizado tres bases de datos: MAVIR, COREMAH y una base de datos que contiene programas de Radio Televisión Española (RTVE). En este trabajo, sólo se ha extraído información de la base de datos MAVIR.

Dentro de dicha base de datos podemos encontrar datos de desarrollo, datos de prueba y datos de entrenamiento. En este caso, sólo se ha trabajado con los datos de desarrollo y los datos de prueba.

3.2.1 Datos de Desarrollo

El principal objetivo de estos datos, es ajustar los parámetros del sistema, para posteriormente con otros datos, intentar optimizar los resultados finales del sistema.

Estos datos presentan para la base de datos MAVIR, en torno a 375 términos diferentes con una longitud de entre 5 y 27 letras, para la tarea de detección de término hablado (STD). De estos 375, 100 servirán como consultas para el sistema de detección de términos hablados de consulta por ejemplo (QbE STD).

Los datos de desarrollo presentan por otro lado, dos archivos de audio que suman aproximadamente una hora de duración.

3.2.2 Datos de Prueba

En los datos de prueba, los audios grandes o repositorios, duran en torno a dos horas, y son tres archivos de audio.

Para la tarea de detección del término hablado, hay aproximadamente 200 palabras diferentes, cada una de ellas formada por entre 4 a 28 fonemas. De esas 200 palabras, alrededor de 100 pueden ser usadas como consultas para el sistema de detección de términos hablados de consulta por ejemplo.

Una vez obtenida la información de la base de datos se procederá al pre-procesamiento de cada una de las señales. En este caso, como ya se ha dicho anteriormente, se ha decidido obtener como características, los posteriorgramas fonéticos o probabilidades a posteriori de los fonemas, a partir del reconocedor BUT (Brno University of Technology). A continuación, se entra más en detalle.

3.3 Esquema Sistema QbE STD (Query-by-Example Spoken Term Detection)

El sistema QbE STD presenta dos etapas fundamentales: la extracción de características, que se lleva a cabo gracias al reconocedor BUT, y el algoritmo de reconocimiento para la detección de consultas. Dichas etapas se explicarán en los siguientes apartados.

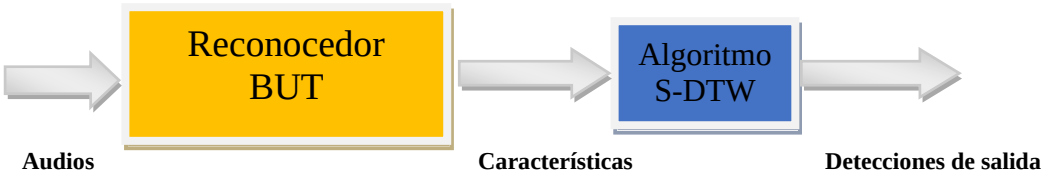


Figura 14: Esquema QbE STD

3.4 Reconocedor BUT (Brno University of Technology)

La siguiente información está basada en la referencia [18].

Una vez se han extraído los audios de la base de datos MAVIR, el objetivo es el pre-procesamiento de la señal, es decir, entre otras cosas, extraer las características de cada uno de los audios. Esto se ha llevado a cabo gracias al reconocedor BUT, que fue desarrollado por la Facultad de Tecnología de la Información de la Universidad Tecnológica de Brno. Dicho reconocedor tuvo éxito en las tareas de identificación de idiomas, indexación y búsqueda de registros de audio, y en la detección de palabras clave.

Hay reconocedor BUT para el idioma inglés, el checo, el húngaro y el ruso. En este trabajo sólo se ha usado el reconocedor BUT en inglés. Cada uno de estos idiomas está representado en el reconocedor con el siguiente número de fonemas: para el checo hay 45 fonemas, para el húngaro 61 fonemas, para el ruso 52 fonemas, mientras que para el inglés se dispone de 39 fonemas. En cada uno de los idiomas podemos encontrarnos con términos no fonéticos. En el caso del inglés disponemos de un término no fonético.

Como se ha mencionado anteriormente, disponemos de tres estados para cada uno de los fonemas (principio, medio y fin del fonema). Para cada fonema habrá tres probabilidades. Es por ello que finalmente se dispondrá de una matriz donde para cada unidad temporal habrá 120 valores (40 fonemas x 3 estados).

Cabe destacar, que el reconocedor BUT para el inglés se ha diseñado a través del corpus TIMIT (Texas Instruments / Massachusetts Institute of Technology). Este conjunto de datos, presenta grabaciones de 630 hablantes de 8 dialectos principales del inglés americano. Incluye transcripciones fonéticas, ortográficas y de palabras alineadas en el tiempo. El diseño de este conjunto de datos, fue desarrollado por el Instituto de Tecnología de Massachusetts, SRI (Stanford Research Institute) Internacional, y Texas Instruments. El resto de idiomas han sido diseñados a partir del conjunto de datos SpeechDat-E.

Algunos de los comandos que se pueden utilizar en el terminal, son los siguientes:

- Para configurar el sistema de reconocimiento:

```
phnrec -c
PHN_CZ_SPDAT_LCRC_N1500|PHN_HU_SPDAT_LCRC_N1500|PHN_RU_SPDAT_LCRC_N1500|
PHN_EN_TIMIT_LCRC_N500
```

- Establecer el formato de entrada:

```
phnrec -c PHN_EN_TIMIT_LCRC_N500 -w alaw|lin16
```

- Establecer archivos de entrada y salida:

```
phnrec -c PHN_EN_TIMIT_LCRC_N500 -i input.raw -o output.rec
```

- Generar archivo HTK con los posteriors:
  phnrec -c PHN_EN_TIMIT_LCRC_N500 test.raw -t post -o output.htk [17]

Una vez obtenido el archivo HTK, se procede a obtener la matriz de posteriors, que será la entrada al algoritmo de reconocimiento.

Como algoritmo de reconocimiento o comparación, ha sido desarrollado el S-DTW en Matlab. Tras esto, se ha construido la salida en un archivo con formato XML, que ha servido como entrada para el sistema scoring, cuya salida se intentará optimizar ajustando ciertos parámetros para que los resultados se asemejen a los de evaluaciones anteriores.

Todo esto se explicará en el apartado de Desarrollo.

4 Desarrollo

Se ha desarrollado en Matlab un algoritmo S-DTW (Subsequence – Dynamic Time Warping) con el que iremos buscando cada una de las consultas en el repositorio y obteniendo datos como el tiempo en el que se encuentra la query en el audio grande, el tiempo en el que coinciden y la puntuación.

Como se ha explicado en secciones anteriores, una vez obtenido este conjunto de datos, se ha generado un archivo XML para cada audio en el que se han buscado las querys. Dicho archivo es la entrada para el sistema de puntuación, con el que obtendremos la salida final con datos como el MTWV (Maximum Term Weighted Vaue) o el ATWV (Actual Term Weighted Vaue).

4.1 Obtención de Matriz de Posteriors

La matriz de posteriors o matriz de probabilidades a posteriori se ha obtenido a partir de un script de Matlab [19].

A partir de este script, se obtiene la matriz de posteriors a partir del archivo HTK obtenido con el reconocedor BUT.

La matriz de posteriors, estará formada por tantas columnas como fonemas por 3 estados haya para el idioma escogido en el reconocedor de BUT, y tantas filas como unidades temporales de 10 milisegundos tenga la señal escogida. En este caso, al haber escogido el idioma inglés, siempre tendremos 120 columnas, ya que en el reconocedor BUT para el inglés se dispone de 40 fonemas.

La matriz de posteriors quedaría de la siguiente manera:

P1,1,t0	P1,2,t0	P1,3,t0	...	P40,1,t0	P40,2,t0	P40,3,t0
...	...	...	...	...	...	...
P1,1,T	P1,2,T	P1,3,T	...	P40,1,T	P40,2,T	P40,3,T

Una matriz de *T x 3N*, donde *T* es el número de unidades temporales que tiene la señal (de 10 milisegundos cada unidad) y *N* es el número de fonemas para el idioma elegido.

La primera fila se corresponde con el primer instante del audio (t0) y va del primer fonema al fonema 40 (que son los 40 que hay en inglés). La última fila se corresponde con el último instante de la señal de audio (*T*). Las tres primeras columnas son las probabilidades del primer fonema para cada uno de los 3 estados en todos los instantes de tiempo de la señal. Las 3 últimas columnas son las probabilidades para el último fonema en todos los instantes de tiempo.

Por lo tanto, la probabilidad de cada fonema en cada instante de tiempo será la media de las 3 probabilidades de los estados para cada instante de tiempo con lo que quedaría una matriz de *T x N*.

A continuación, se procede al desarrollo del algoritmo S-DTW.

4.2 Algoritmo S-DTW (Subsequence – Dynamic Time Warping)

La información correspondiente a este apartado, se corresponde con la referencia [8].

Como ya se ha descrito en apartados anteriores, el algoritmo S-DTW es una modificación del algoritmo DTW (Dynamic Time Warping) y su principal objetivo es encontrar la señal de menor longitud en la señal de mayor longitud, y se utiliza cuando las señales tienen longitudes muy distintas. A continuación, se explica más detalladamente los pasos de dicho algoritmo.

La siguiente figura, muestra el esquema del algoritmo S-DTW implementado:

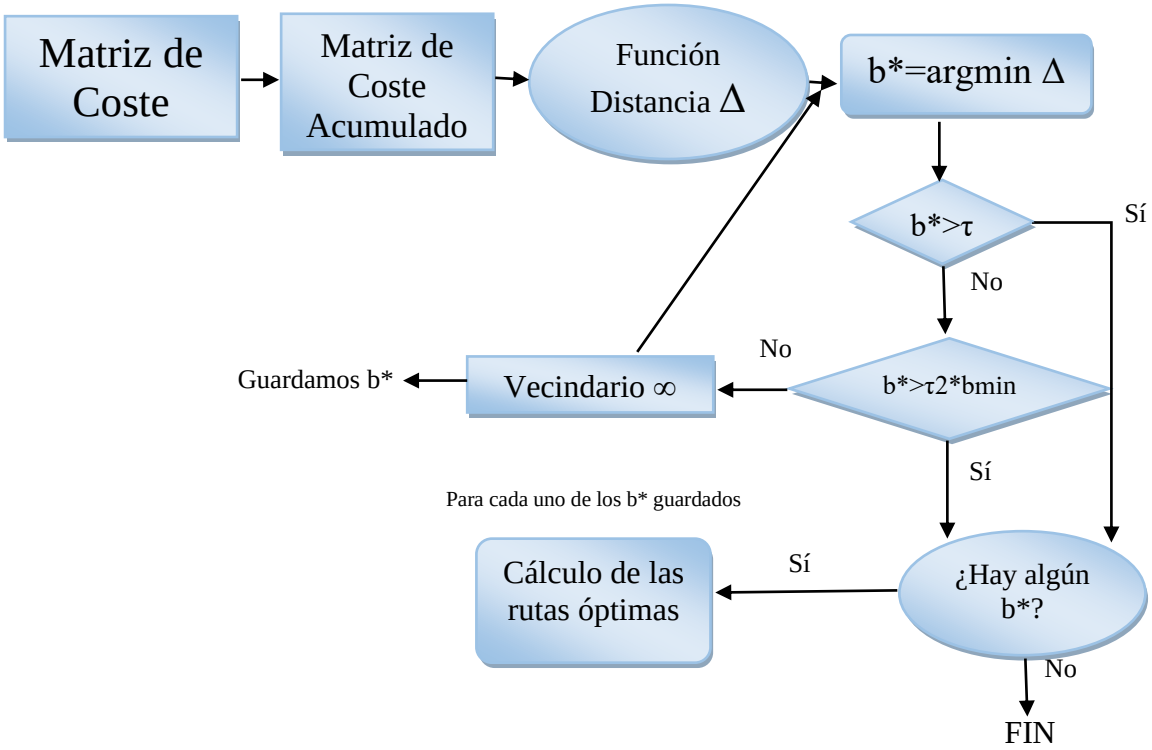


Figura 15: Esquema Algoritmo S-DTW

4.2.1 Matriz de Coste

Se va a definir la consulta que vamos a buscar en el repositorio como X_n, que va de 1 hasta N (1:N), mientras que se va a definir el repositorio como Y_m que va de 1 hasta M (1:M). Por otro lado, U representará el número de las unidades fonéticas (en este caso 40).

El primer paso de este algoritmo es calcular la matriz de coste. Dicha matriz, almacena la similitud entre cada par de la query y del repositorio. Es el coste entre todos los elementos de la matriz de posteriors de la consulta y del repositorio.

Se ha empleado el coeficiente de correlación Pearson (*r*) para construir la matriz de coste. Se puede definir dicho coeficiente, como un índice para medir el grado de relación de dos variables cuando éstas son cuantitativas y continuas [13].

El coeficiente de correlación de Pearson de cada par, se ha calculado con la siguiente fórmula:

$$r(x_n, y_m) = \frac{U(x_n \cdot y_m) - ||x_m||||y_m||}{\sqrt{U||x_n^2|| - ||x_n||^2)(U||y_m^2|| - ||y_m||^2)}}$$

En Matlab, dicho coeficiente se puede sacar gracias a la función "corrcoef", con la que se obtiene la matriz de correlación de cada par. Cogiendo de dicha matriz la posición (1,2) o (2,1) se obtiene el coeficiente de correlación.

El coeficiente de Pearson se mapea entre los valores [0,1]. Para ello, se utiliza la matriz de coste que se calcula con la siguiente fórmula que depende del coeficiente de correlación:

$$c(x_n, y_m) = \frac{1 - r(x_n, y_m)}{2}$$

Atendiendo a la siguiente fórmula, cabe destacar que el coste será máximo (1) cuando el coefieciente de Pearson sea -1, será 0.5 cuando r sea igual a 0, y será 0 cuando el coefieciente sea igual a 1.

Cuanto menor sea el coste y mayor el coeficiente de Pearson, más similar será el par que estamos comparando.

A continuación se muestra un ejemplo de la matriz de coste:

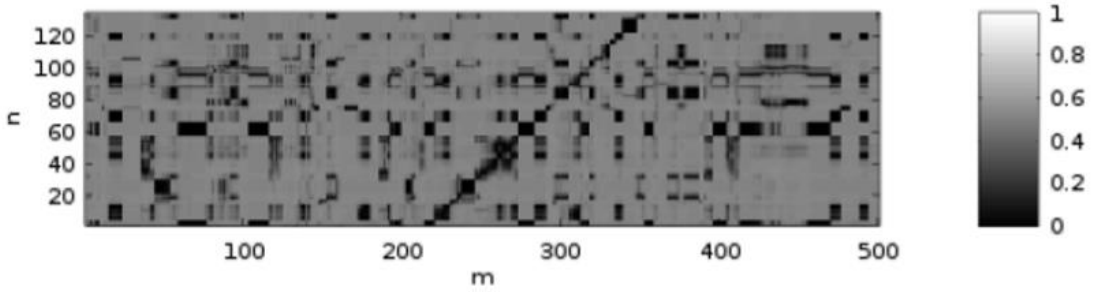


Figura 16: Matriz de Coste con coeficiente de correlación de Pearson

Como se puede observar, la matriz de coste representa una relación o la similitud que hay entre la query o consulta, y el repositorio. Atendiendo a la figura, hay que destacar que cuanto menor sea el coste (zonas más oscuras), existe una mayor similitud entre la query a buscar y el repositorio. Según la figura, parece que se ha encontrado la query en el repositorio, y va desde m = 200 a m = 350 aproximadamente.

4.2.2 Matriz de Coste Acumulado

En el caso del algortimo S-DTW, una vez calculada la matriz de coste, el siguiente paso es el cálculo de la matriz de coste acumulado. Se calcula a partir de la matriz de coste con la siguiente fórmula:

$$\mathrm{D}_{n,m} = \begin{cases} c(\mathrm{x}_n, \mathrm{y}_m) & \text{if} \quad n = 0 \\ c(\mathrm{x}_n, \mathrm{y}_m) + \mathrm{D}_{n-1,0} & \text{if} \quad n > 0,\ m = 0 \\ c(\mathrm{x}_n, \mathrm{y}_m) + \mathrm{D}^*(n, m) & \text{else,} \end{cases}$$

donde $D^*(n,m) = min(D_{n-1,m}, D_{n-1,m-1}, D_{n,m-1})$.

En la siguiente figura se muestra un ejemplo de la matriz de coste acumulado.

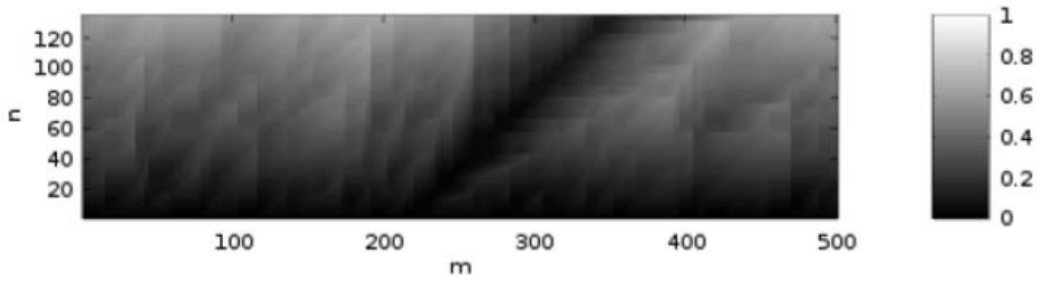


Figura 17: Matriz de Coste Acumulado con coeficiente de correlación de Pearson

Atendiendo a la figura anterior, se puede observar que al irse acumulando el coeficiente de correlación de Pearson, las primeras filas están más oscuras y las últimas están más claras. En dicha figura, también se puede ver con claridad donde se encuentra la consulta en el repositorio.

4.2.3 Función Distancia

El siguiente paso en el algoritmo S-DTW es el cálculo de la función distancia (Δ). Dicha función se corresponde con la última fila de la matriz de coste acumulado.

A lo largo de esta función, se buscarán los mínimos (b^*), que serán los valores iniciales de las rutas óptimas de las consultas u ocurrencias a buscar en el repositorio. Primero se buscará el mínimo global (b_{min}) de la función, y posteriormente se buscarán los mínimos locales que cumplan también con los parámetros establecidos.

Haciendo referencia a las figuras anteriores, podemos concluir que el mínimo global de la función distancia estaría en torno a m = 350 y sería el primer punto inicial del que partiríamos.

4.2.4 Parámetros tau

Una vez se ha encontrado un mínimo de la función distancia, ya sea mínimo global o mínimo local, hay que comprobar si dicho punto cumple con dos condiciones.

El mínimo encontrado, primero tendrá que ser superior a un umbral τ configurable, y segundo, tendrá que ser superior a τ_2 (segundo umbral configurable) x b_{min} (mínimo global). Si alguna de esas dos condiciones no se cumple, dicho mínimo no será uno de los puntos de partida de las rutas óptimas, y se acabará la búsqueda.

El objetivo es ajustar ambos umbrales para obtener los resultados más óptimos posibles.

En conclusión, para que el mínimo encontrado (b^*) sea uno de los puntos donde comienza una de las ocurrencias detectadas se tiene que cumplir que:

$$b^* > \tau,\ b^* > \tau_2 \, x \, b_{min}$$

4.2.5 Vecindario ∞

Una vez se ha encontrado un mínimo b^* que cumple con las condiciones establecidas (taus del apartado anterior), se pretende evitar encontrarnos con ocurrencias o rutas óptimas muy próximas a las ya encontradas.

Por ello, la idea es descartar las rutas, cuyo punto de inicio se encuentre a una distancia muy próxima, ya que es muy poco probable que se diga la misma palabra de forma consecutiva. Haciendo infinito los puntos próximos al mínimo local se consigue evitar esto, ya que esos puntos nunca podrán ser ya un mínimo de la función distancia.

En el caso de este sistema, se ha elegido hacer infinito los puntos que se encuentren a una distancia menor a 500 ms (50 frames de la unidad temporal que son 10 ms) del mínimo encontrado, ya que la probabilidad de que se diga una palabra dos veces en un intervalo menor a 500 milisegundos es mínima.

Todo este procedimiento, y el anterior explicado en la sección 4.2.4, se repetirá hasta que alguno de los mínimos encontrados no cumpla con alguna de las condiciones establecidas para los taus.

4.2.6 Cálculo de las rutas óptimas

Con cada uno de los mínimos *(b^*)*, que cumpla con las condiciones anteriormente descritas, se procede a encontrar su ruta óptima *(a^*, b^*)*. Se halla de la siguiente manera:

Supongamos que nuestra ruta óptima final es $p = (p_{1, \ldots, p_l})$. Comenzando por $p_1 = b^*$, los siguientes puntos se calculan con la siguiente fórmula:

$$p_{l-1} = argmin(D(n-1, m-1), D(n-1, m), D(n, m-1))$$

En la siguiente figura, se muestra una ruta óptima sobre la matriz de coste acumulado.

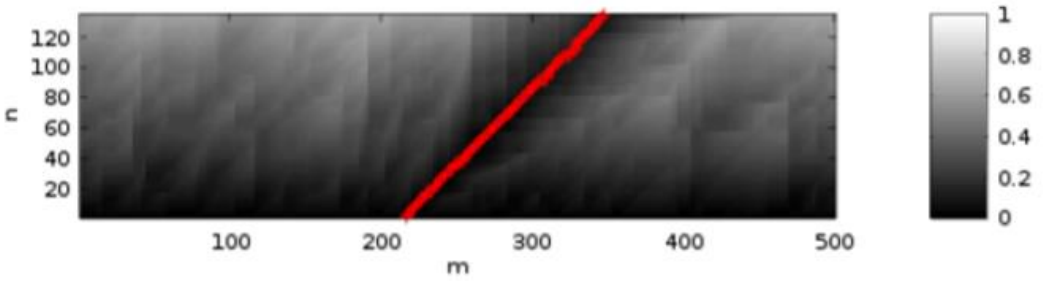


Figura 18: Camino óptimo sobre matriz de coste acumulado

4.3 Generar salida XML

La siguiente etapa, a continuación del algoritmo de reconocimiento, consiste en construir la salida XML que sigue el formato indicado en la evaluación NIST (National Institute of Standards and Technology). A partir de la información que devuelve el algoritmo, se generará una única salida para todas las querys sobre un repositorio determinado. El algoritmo devuelve información como el punto donde se encuentra la ocurrencia, el tiempo que dura la coincidencia entre la consulta y el repositorio, la puntuación de dicha detección…

Se genera una lista con todas las consultas detectadas en el repositorio. A continuación, se muestra un ejemplo de dos detecciones en el archivo XML y se explica cada uno de los campos:

```
<detected_termlist termid="TEST-0000" term_search_time="0.67" oov_term_count="1">
<term file="mavir04" channel="1" tbeg="1604.26" dur="0.59" score="14.7055" decision="YES"/>
</detected_termlist>

<detected_termlist termid="TEST-0001" term_search_time="0.27" oov_term_count="1">
<term file="mavir04" channel="1" tbeg="2038.66" dur="0.24" score="11.6326" decision="YES"/>
</detected_termlist>
```

- **termid:** Es el nombre de la query que vamos a buscar en el repositorio.
- **term_serach_time:** Es el tiempo en segundos que dura la consulta.
- **file:** Nombre del audio donde vamos a realizar la búsqueda.
- **tbeg:** Es el tiempo en el que empieza la ocurrencia, es decir, el tiempo en el que se empieza a detectar la consulta en el repositorio.
- **dur:** Es la duración de la ocurrencia, es decir, el tiempo en el que coinciden la consulta y el audio en el que se busca.
- **score:** Es la puntuación de la detección. Es la inversa del coste medio del camino óptimo encontrado para la consulta
- **decision:** Sirve para señalar si la detección es correcta.

La siguiente etapa del sistema, será el sistema scoring, que se llevará a cabo cuando se haya generado el archivo XML repitiendo esto para todas las consultas encontradas en el repositorio. A continuación, se explica más detalladamente el funcionamiento de dicho sistema.

4.4 Sistema de puntuación y evaluación

La información correspondiente a esta sección se basa en [20].

Para llevar a cabo la evaluación y el sistema de puntuación del sistema, se han utilizado las métricas que plantea la evaluación de NIST (National Institute of Standards and Technology) [20]. Dicha evaluación plantea dos métricas importantes: ATWV (Actual Term Weighted Vaue) y MTWV (Maximum Term Weighted Vaue).

El TWV (Término de valor ponderado), se calcula como 1 menos la suma de la probabilidad por término de la ocurrencia no detectada y de la probabilidad por término de las falsas alarmas. Su fórmula para calcularlo es la siguiente:

$$TWV(\theta) = 1 - \left[P_{Miss}(\theta) + \beta \cdot P_{FA}(\theta)\right]$$

donde θ es el umbral que determina las palabras que son puntuadas; β es un parámetro que depende del coste de una detección incorrecta, del valor de una detección correcta y de la probabilidad previa de una palabra clave; P_{MISS} es la probabilidad de las ocurrencias no detectadas; y P_{FA} es la probabilidad de las falsas alarmas.

Cabe destacar, que nos referimos a ocurrencias no detectadas, a aquellas consultas que se encuentran en el repositorio y no son detectadas por el algoritmo de reconocimiento, mientras que nos referimos a falsas alarmas, a aquellas consultas que no se encuentran en el repositorio pero que por equivocación detecta el algoritmo.

En el caso de ATWV, es el TWV (Term Weighted Vaue) promedio de todas las detecciones que toman como decisión un "SÍ", es decir, para aquellas detecciones que decidimos que pueden ser válidas. Se podría decir que la fórmula del ATWV es la descrita anteriormente.

Por otro lado, el MTWV es el máximo valor de ATWV para un sistema dentro de todos los posibles valores de θ.

La entrada al sistema de puntuación, será el archivo XML generado como se ha explicado en la sección anterior.

Este sistema nos dará como salida varios archivos, entre los que destaca el archivo score.occ. En dicho documento, se muestra un resumen con la información de la búsqueda de cada una de las querys en el repositorio. Aparecen dos tablas: en la primera podemos observar consulta a consulta las ocurrencias detectadas correctamente, las falsas alarmas, las ocurrencias no detectadas, y sus porcentajes. En la segunda se puede ver un resumen del total de consultas, con valores como el ATWV, el MTWV, la probabilidad media de las

falsas alarmas, la probabilidad media de las ocurrencias no detectadas, y la puntuación media.

A continuación, se muestra un ejemplo de ambas tablas:

TermID	Text	Search Time	ALL Ref	Corr	FA	Miss	Occ. Value	P(FA)	P(Mis)
TEST-0000	TEST-0000	0.67	6	1	0	5	0.167	0.00000	0.833
TEST-0001	TEST-0001	0.27	4	1	0	3	0.250	0.00000	0.750
TEST-0004	TEST-0004	0.36	16	0	0	16	0.000	0.00000	1.000
TEST-0005	TEST-0005	0.42	1	1	0	0	1.000	0.00000	0.000
TEST-0006	TEST-0006	0.46	2	1	0	1	0.500	0.00000	0.500
TEST-0009	TEST-0009	0.84	2	0	0	2	0.000	0.00000	1.000
TEST-0011	TEST-0011	0.68	8	1	0	7	0.125	0.00000	0.875
TEST-0013	TEST-0013	0.57	11	0	0	11	0.000	0.00000	1.000
TEST-0016	TEST-0016	0.57	5	1	0	4	0.200	0.00000	0.800

Figura 19: Tabla score.occ consulta a consulta

DET Curve Analysis Summary

Description	Weighted Max Value	Weighted A. Value	Decision P(Fa)	Decision P(Miss)	Decision Score
ALL	0.1074	0.0724	0.00002	0.875	6.876500e+00

Figura 20: Tabla socre.occ resumen total

5 Integración, pruebas y resultados

Después de que se haya llevado a cabo el desarrollo del sistema, se intenta optimizar su resultado. Para ello, se han realizado pruebas con todas las consultas de la base de datos MAVIR de la parte development, que se han buscado en el repositorio mavir03; y las consultas de la parte test, que se han buscado en el audio mavir04. Se ha intentado ajustar los taus para obtener los mejores resultados.

Se van buscando mediante el algoritmo de reconocimiento todas las consultas una a una en el repositorio elegido, y se va generando el archivo de salida XML que posteriormente servirá como entrada para el sistema de puntuación que es el que nos dará los resultados finales. La salida del algoritmo para cada par consulta/repositorio es diferente, y pueden ocurrir distintas cosas:

- Que la consulta no sea detectada por el algoritmo en el audio. En este caso, en el archivo XML se indicará de la siguiente manera:

```
<detected_termlist termid="TEST-0137" term_search_time="0.47" oov_term_count="1">
</detected_termlist>
```

- Que la consulta sea detectada una vez:

```
<detected_termlist termid="TEST-0140" term_search_time="0.67" oov_term_count="1">
<term file="mavir04" channel="1" tbeg="876.86" dur="0.61" score="15.1893" decision="YES"/>
</detected_termlist>
```

- Que la consulta sea detectada varias veces:

```
<detected_termlist termid="TEST-0128" term_search_time="0.39" oov_term_count="1">
<term file="mavir04" channel="1" tbeg="1820.06" dur="0.27" score="3.9427" decision="NO"/>
<term file="mavir04" channel="1" tbeg="452.07" dur="0.24" score="5.1550" decision="YES"/>
</detected_termlist>
```

Una vez se construye todo el archivo XML con todos los pares query/repositorio, se genera la salida final del sistema, y podremos obtener diferentes conclusiones.

Para optimizar los resultados, por un lado, se han ajustado los taus para intentar descartar falsas alarmas y ocurrencias no detectadas; y, por otro lado, una vez el algoritmo de reconocimiento devuelve la puntuación de la ocurrencia, se ha decidido si dicha ocurrencia es utilizada en la evaluación del sistema de puntación. Para decidir si va a influir en la evaluación, se escribe un “YES” o un “NO” en el parámetro “decision” de cada consulta en el archivo XML.

En este trabajo, primero se ha probado con un par de taus para la parte development, y posteriormente se han intentado ajustar los taus en la parte test para intentar optimizar los resultados.

Como se ha dicho con anterioridad, se han hecho pruebas con todas las consultas, y gracias al sistema de puntuación se han obtenido distintos resultados para los distintos valores de los parámetros.

Observando las funciones distancia con la base de datos development, decidí probar con los parámetros τ_1=5 y τ_2=1.20 ya que, tras la primera observación de dichas funciones, parecía que los resultados iban a ser óptimos. Posteriormente, hice la prueba para todos los datos de la base de datos test. Sin embargo, debido a la mala puntuación obtenida en los parámetros ATWV (Actual Term Weighted Vaue) y MTWV (Maximum Term Weighted Vaue), intenté ajustar los parámetros, obteniendo un mejor MTWV, pero peor ATWV. Conseguí optimizar el valor de ATWV utilizando un umbral score igual a 5. De esta manera, descartaba las ocurrencias cuyo score fuera inferior a este umbral.

A continuación, se muestra una tabla para resumir los resultados obtenidos en función de los distintos parámetros para la parte test:

RESULTADOS (TEST COMPLETO)							
τ_1	τ_2	umbral score = 5	ATWV	MTWV	P(Fa)	P(Miss)	SCORE
5	1.20	NO	0.0332	0.0615	0.00003	0.911	8.9799
15	1.01	NO	0.0095	0.1074	0.00002	0.875	6.8765
15	1.01	SÍ	0.0724	0.1074	0.00002	0.875	6.8765

Tabla 1: Resultados

En esta tabla, se recogen los resultados obtenidos a partir de las pruebas realizadas con toda la base de datos variando los valores de los taus y utilizando o no un umbral score para decidir si contar con ciertas consultas en el resultado final. Se recogen los valores de ATWV; MTWV; P(Fa), que es la probabilidad media de falsas alarmas; P(Miss), que es la probabilidad media de las ocurrencias no detectadas y el score o puntuación media.

Atendiendo a los resultados de esta tabla, podemos concluir que escribiendo un "NO" en el parámetro "decision" de las querys que no superen un score específico (en mi caso elegí 5 como umbral score), se obtienen los mejores resultados de ATWV, y que dicho parámetro no influye en los demás valores. La probabilidad de falsas alarmas y la probabilidad de ocurrencias no detectadas, no varían mucho en función de los parámetros, y se mantienen más o menos constantes.

Una manera de sacar más conclusiones, sería haciendo pruebas utilizando por separado cada uno de los taus, y así ver cómo afecta a los resultados la variación de cada uno de ellos.

Los mejores resultados obtenidos, han sido con los parámetros τ_1 = 15, τ_2 = 1.01 y utilizando el umbral score, con los que se han obtenido los siguientes valores: ATWV = 0.0724, MTWV = 0.1074, P(Fa) = 0.00002, P(Miss) = 0.875 y SCORE = 6.8765.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

A partir de la realización de este trabajo, se pueden obtener varias conclusiones atendiendo entre otras cosas a los resultados y a la utilización que pueden tener este tipo de reconocedores en la vida cotidiana.

Lo primero a destacar, es que hoy en día numerosas aplicaciones utilizan el reconocimiento de voz para mejorar su rendimiento, por lo que estos trabajos relacionados con el reconocimiento y sus posteriores mejoras y avances van cobrando cada vez más importancia.

En este tipo de disciplinas como es el reconocimiento del habla, no siempre se obtienen unos resultados óptimos, ya que siempre suele haber un porcentaje de error y de desacierto.

En este trabajo, se ha logrado el objetivo principal, que era el desarrollo de un sistema de búsqueda de palabras clave en voz mediante ejemplos, basado en las evaluaciones anteriores de ALBAYZIN. Se ha desarrollado un sistema QbE STD (Query-by-Example Spoken Term Detection), cuya extracción de características se ha basado en los posteriorgramas fonéticos, y el algoritmo de reconocimiento implementado ha sido el S-DTW (Subsequence - Dynamic Time Warping).

El sistema, ha conseguido su mayor finalidad, que es la de encontrar consultas de voz en un repositorio o audio grande. Se ha intentado ajustar algunos parámetros que influyen en los resultados de ATWV (Actual Term Weighted Vaue) y MTWV (Maximum Term Weighted Vaue), que son las métricas escogidas para la evaluación, para intentar optimizar el sistema. Aunque, es verdad, que dichos parámetros se podrían ajustar más y seguramente obtener unos resultados más precisos y óptimos.

6.2 Trabajo futuro

Debido a, como ya se ha mencionado varias veces a lo largo de este trabajo, la importancia del reconocimiento de voz en la vida cotidiana, va a "obligar" que siga habiendo numerosas mejoras e investigaciones relacionadas con esta área.

En lo que a este sistema se refiere, pienso que en un futuro se pueden ajustar más los parámetros para intentar optimizarlo más, así como probar con nuevas bases de datos e intentar maximizar los valores de ATWV y MTWV. También, en un tiempo cercano, pienso que se podría probar a reconocer frases a través de dicho sistema.

Atendiendo al desarrollo del sistema, se podría probar a, en lugar de hacer la media de las 3 probabilidades de cada uno de los fonemas a la hora de sacar la matriz de posteriors, dejar las 3 probabilidades por separado y dejar la matriz de coste y todo lo posterior en función de 120 probabilidades en lugar de 40, y ver si esto mejora el rendimiento del sistema.

Referencias

[1] Raquel Fernández Serrano, Pau Pérez Pascual, Jesús Torres Ferrando, Mª Teresa Lorente Martínez, Concepción Monteagudo Córdoba; "Búsqueda Inteligente de Información Multimedia", Trabajo de Universidad Politécnica de Valencia, curso 2002-2003.

[2] "Sistemas de reconocimiento de voz, ¿la correcta evolución a la hora de crear contenidos?". En:marketing.com, Disponible en: https://www.marketingdirecto.com/digital-general/digital/sistemas-reconocimiento-voz-correcta-evolucion-hora-crear-contenidos, Publicado: septiembre 2017. Consultado: abril 2020.

[3] Coelho, Fabián (s.f). "Fonema (qué es, explicación y ejemplos)". En: Diccionariodedudas.com. Disponible en: https://www.diccionariodedudas.com/fonema/. Consultado: abril 2020

[4] José M. Martínez (17 de mayo 2009), "Internet nació de un proyecto militar de Estados Unidos en la Guerra Fría", RTVE, Noticias, Ciencias y Tecnología.

[5] "Historia de los sistemas de reconocimiento de voz", En: timetoast.com, Disponible en: https://www.timetoast.com/timelines/historia-del-reconocimiento-de-la-voz, Consultado: abril 2020.

[6] Deiby Erasmo Obando Portilla, Guillermo Daniel Ortega Galeano; "Sistema interactivo de reconocimiento de fonemas para la interpretación y traducción a lengua de señas"; Trabajo de Grado, Universidad de Nariño (Colombia), año 2011.

[7] Dr. Carlos Alejandro de Luna Ortega, Dr. Miguel Mora González, Dr. Julio César Martínez-Romo, Dr. Francisco Javier Luna Rosas, "Reconocedor de Palabras con el uso de Regresión Lineal y Coeficiente Muestral", Trabajo de Universidad de Guadalajara, Centro Universitario de los Lagos, Universidad Politécnica de Aguascalientes, Instituto Tecnológico de Aguascalientes, año 2012.

[8] María Cabello, Doroteo Torre Toledano, Javier Tejedor, "A Language-independent approach for the Query-by-Example Spoken Term Detection task of the Search on Speech ALBAYZIN 2018 evaluation", AUDIAS Universidad Autónoma de Madrid, Universidad San Pablo-CEU. En: ResearchGate. Publicado: noviembre 2018.

[9] "Dynamic Time Warping", En: godieboy.com, Disponible en: https://godieboy.com/2012/09/dynamic-time-warping.html, Publicado: 18 septiembre 2012, Consultado: abril 2020.

[10] "Introducción a la distorsión de tiempo dinámico", Capítulo 23, "Learning algorithm eBook", En: riptutorial.com, Disponible en: https://riptutorial.com/es/algorithm/example/24981/introduccion-a-la-distorsion-de-tiempo-dinamico.

[11] Gloria Inés Álvarez, “Bases Formales de la Computación: Sesión 3. Modelos Ocultos de Markov”, Trabajo Departamento de Ciencias e Ingeniería de la Computación Pontificia Universidad Javeriana Cali, año 2008.

[12] Pablo Aguilera Bonet, “Reconocimiento de Voz usando HTK”, Proyecto fin de carrera, Departamento de Teoría de la señal y comunicaciones, Universidad de Sevilla.

[13] “Red Neuronal Artificial”, En: https://es.wikipedia.org/wiki/. Disponible en: https://es.wikipedia.org/wiki/Red_neuronal_artificial. Consultado: abril 2020.
“Coeficiente de correlación de Pearson”. En: En: https://es.wikipedia.org/wiki/. Disponible en: https://es.wikipedia.org/wiki/Coeficiente_de_correlaci%C3%B3n_de_Pearson. Consultado: abril 2020.

[14] Diego Calvo, “Definición de red neuronal artificial”. En: diegocalvo.es/. Disponible en: http://www.diegocalvo.es/definicion-de-red-neuronal/. Publicado en: julio 2017, Consultado: abril 2020

[15] Javier Tejedor, Doroteo Torre Toledano, Paula López-Otero, Laura Docio-Fernández, Jorge Proença, Fernando Perdigão, Fernando García-Granada, Emilio Sanchis, Anna Pomopili y Alberto Abad, “ALBAYZIN Query-by-Example Spoken Term Detection 2016 Evaluation”, EURASIP Journal on Audio, Speech, and Music Processing. En: “Springer Open”, Artículo número 2, abril 2018.

[16] Javier Tejedor, Doroteo Torre Toledano, “The ALBAYCIN 2018 Search on Speech Evaluation Plan”. Universidad San Pablo-CEU, AUDIAS Universidad Autónoma de Madrid. EURASIP Journal on Audio, Speech, and Music Processing. En: ResearchGate. Publicado: año 2019.

[17] Sentencia para obtener posteriorgramas fonéticos en archivo htk. En: groups.google.com. Disponible en: https://groups.google.com/forum/#!msg/phnrec/l_WAYh2Hdd8/ySExSe0Tng4J. Consultado: Finales 2018.

[18] Petr Schwarz , Pavel Matejka, Lukas Burget, Ondrej Glembek, Faculty of Information Technology of Brno University of Technology, “Phoneme recognizer based on long temporal context”. En: BUT Speech@FIT, Disponible en: https://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context. Publicado en: 07 de agosto 2009

[19] Disponible en: http://read.pudn.com/downloads162/sourcecode/math/737106/READHTK.M__.htm, Autor: Mike Brookes, 1997. Consultado: 2019.

[20] “OpenKWS13 Keyword Search Evaluation Plan”. NIST Open Keyword Search 2013 Evaluation (OpenKWS13). National Institute of Standards and Technology (NIST), Washington DC, USA. Publicado en: Julio 2013

Glosario

QbE STD	Query-by-Example Spoken Term Detection
TWV	Term Weighted Value
ATWV	Actual Term Weighted Value
MTWV	Maximum Term Weighted Value
DTW	Dynamic Time Warping
S-DTW	Subsequence - Dynamic Time Warping
HMM	Hidden Markov Model
MFCC	Mel Frequency Cepstral Coefficients
DCT	Discrete Cosine Transform
LPC	Linear predictive coding
ANN	Artificial Neural Network
DNN	Deep Neural Network
NIST	National Institute of Standards and Technology
BUT	Brno University of Technology
GMT	Group of Multimedia Technology
TIMIT	Texas Instruments / Massachusetts Institute of Technology
IBM	International Business Machines Corporation