UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE MÁSTER

Implementación y evaluación de un sistema QbE-STD (Query-by-Example Spoken Term Detection)

Máster Universitario en Investigación e Innovación en TIC (i^2TIC)

Autora: Cabello Aguilar, María

Tutor: Torre Toledano, Doroteo
Departamento de Tecnología Electrónica y de las Comunicaciones

FECHA: Septiembre 2018

Implementación y evaluación de un sistema QbE-STD (Query-by-Example Spoken Term Detection)

Autora: Cabello Aguilar, María

Tutor: Torre Toledano, Doroteo

Grupo Audias -Audio, Data Intelligence and Speech
Dpto. Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Septiembre 2018

Resumen

Con el fin de extraer información y reconocer palabras clave en los ficheros de audio presentes en medios de comunicación e Internet, surgen los sistemas QbE-STD (Query-by-Example Spoken Term Detecion).

Los sistemas QbE-STD tratan, por un lado de buscar un ejemplo de un objeto o parte de él en otro objeto (QbE), y por otro de encontrar palabras o secuencias de ellas en archivos de audio (STD).

En este Trabajo Fin de Máster se ha desarrollado un sistema QbE-STD independiente del idioma cuya entrada o query está basada en términos hablados, lo que permite a un usuario realizar una búsqueda en un repositorio de audio emitiendo con su voz el término a buscar.

Como técnica de representación del habla se han empleado los llamados posteriorgramas fonéticos, obtenidos mediante los decodificadores fonéticos desarrollados por la Universidad de Tecnología de Brno (BUT).

Para la detección de los términos de búsqueda en los repositorios de audio se ha utilizado el algoritmo Subsequence Dynamic Time Warping (S-DTW).

Además de desarrollar un sistema QbE-STD que sirva como punto de partida para futuras vías de trabajo del grupo AUDIAS[1], se han incluido distintas técnicas y aportaciones con el objetivo de intentar mejorar los resultados obtenidos. Entre estas técnicas se encuentra la selección de unidades fonéticas o la fusión de idiomas.

Para el desarrollo de la solución y la realización de las pruebas se han utilizado los audios pertenecientes a las evaluaciones Albayzin 2016 y 2018 Search on Speech.

Los resultados obtenidos se han podido contrastar con otros sistemas publicados, ya que para el cálculo de la precisión se ha empleado un procedimiento de evaluación oficial propuesto por el instituto de tecnología NIST y ampliamente utilizado.

Los valores de precisión alcanzados demuestran que mediante el sistema básico se obtienen unos resultados competitivos y semejantes a los de otras implementaciones de este tipo.

Palabras clave

Query-by-example spoken term detection, búsqueda de términos hablados, posteriorgramas fonéticos, reconocedores BUT, subsequence DTW, evaluación Albayzin 2016, evaluación Albayzin 2018, reconocimiento fonético, métodos independientes del lenguaje.

[1] http://audias.ii.uam.es/

Abstract

In order to extract information and recognize key words in the audio files belonging to media and Internet, QbE-STD (Query-by-Example Spoken Term Detection) systems are developed.

QbE-STD systems have as purpose, on the one hand, to search for an example of an object or part of it in another object (QbE), and on the other, to find words or sequences of them in audio files (STD).

In this Master Thesis, a language-independent QbE-STD system has been developed, whose input or query is based on spoken terms, which allows an user to perform a search in an audio repository by saying the search term with his/her own voice.

As a technique of speech representation, phonetic posteriorgrams have been used, obtained through the phonetic decoders developed by the Brno University of Technology (BUT).

The Subsequence Dynamic Time Warping (S-DTW) algorithm has been used to detect the search terms in the audio repositories.

In addition to developing a QbE-STD system that will be used as a first point for future investigation of AUDIAS[2] group, different techniques and contributions have been included in order to try to improve the achieved results. Among these techniques, the phonetic units selection or the languages fusion have been implemented.

In the development and test phases, the audios belonging to the Albayzin 2016 and 2018 Search on Speech evaluation have been used.

The achieved results have been compared with other published systems, because of the use of an official evaluation procedure proposed by NIST technology has been implemented to obtain accuracy.

The precision values obtained show that competitive results have been achieved through the basic system, and these are similar to those of other implementations of this type.

Keywords

Query-by-example spoken term detection, spoken terms search, phonetic posteriorgramas, BUT recognizers, subsequence DTW, Albayzin 2016 evaluation, Albayzin 2018 evaluation, phonetic recognition, independent-language methods.

[2] http://audias.ii.uam.es/

Agradecimientos

A mi compañero de vida, Raúl. Gracias por no soltarme nunca la mano a lo largo de este emocionante viaje. Gracias por tu paciencia, tus palabras llenas de calma y, sobre todo, gracias por creer siempre en mí.

A mis padres y hermano, que por un lado estaban deseando que esto terminara, pero por otro, saben que yo no podría vivir sin un poquito de acción. Todo lo que soy es gracias a vosotros.

A mi pequeño Iván, por haber aprendido antes de tiempo que hay cosas que requieren mucho esfuerzo, y por asumir que a veces hay que cambiar los divertidos juegos contigo por un ordenador.

A mi tutor Doroteo, por saber manejar a la perfección una situación como la mía y acompañarme en cada paso que he dado.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1 Motivación

El contenido multimedia crece constantemente, y cada instante se crea nueva información. Estos contenidos son muy valiosos, por lo que se hace necesario contar con herramientas que permitan realizar búsquedas automáticas en las grandes bases de datos heterogéneas de imágenes, audio y vídeo [1] [2].

Con el fin de realizar búsquedas en información de audio, el reconocimiento de palabras clave se ha convertido en un campo cada vez más extendido. Pese a que el acceso y la búsqueda en estos ficheros de audio es una tarea compleja, existen distintas aproximaciones [3], tales como Query-by-Example (QbE), que consiste en encontrar un fragmento de audio o query en un repositorio; y Spoken Term Detection (STD) o detección de términos hablados.

QbE-STD aúna estas 2 técnicas, ofreciendo la posibilidad al usuario de realizar una búsqueda de términos hablados en un repositorio o documento de audio.

En la actualidad existen evaluaciones como NIST OpenKWS (NIST Open Keyword Search Evaluation) o Albayzin Search on Speech (en la que la Universidad participa activamente), donde una de las tareas que se suele englobar es QbE-STD.

Además, una de las líneas de investigación del grupo AUDIAS es la detección de términos hablados, por lo que es un aspecto de especial motivación contar con un sistema QbE-STD.

El objetivo fundamental de este Trabajo Fin de Máster es implementar un sistema QbE-STD que ofrezca los mejores resultados posibles, y que constituya el punto de partida para posteriores mejoras y evoluciones, pudiendo así participar en las actuales y próximas ediciones de las evaluaciones mencionadas.

Para acometer este objetivo principal, se realizó un trabajo previo en la asignatura Iniciación a la Investigación y la Innovación englobada en el Máster Universitario i^{2}TIC (Investigación e Innovación en Tecnologías de la Información y la Comunicación) de la Escuela Politécnica Superior. En este trabajo se analizaron los sistemas QbE-STD existentes, las fases que componen cada uno de ellos y los métodos y técnicas aplicadas en su desarrollo.

En este Trabajo Fin de Máster se emplean técnicas de procesamiento de audio y voz. Para ello, han sido indispensables las nociones aprendidas en la asignatura de Biometría cursada en el Máster.

1.2 Objetivos

Para lograr cumplir el objetivo principal, que es el desarrollo del sistema QbE-STD, es necesario definir una serie de objetivos parciales:

- Implementar en Matlab un sistema QbE-STD fácilmente configurable y escalable de cara a futuras evoluciones del mismo, consiguiendo así que sea una vía de trabajo dentro del grupo AUDIAS.

- Evaluar los resultados obtenidos con un procedimiento de evaluación oficial que permita contrastar, de forma objetiva, los logros conseguidos con los alcanzados por otros sistemas QbE-STD.

- Incluir, siempre que sea posible, aportaciones propias al sistema, consiguiendo así una posible diferenciación respecto a los ya desarrollados.

1.3 Organización de la memoria

La memoria de este Trabajo Fin de Máster consta de los siguientes capítulos:

- **Capítulo 1: Introducción.**

  En este capítulo se describen los aspectos fundamentales que han motivado este Trabajo Fin de Máster, los objetivos del mismo y la relación que existe con alguna de las asignaturas del Máster Universitario i^{2}TIC.

- **Capítulo 2: Estado del arte**.

  En este capítulo se detallan los conceptos teóricos necesarios para comprender qué es un sistema QbE-STD, cuáles son las fases que lo componen, los métodos existentes, las técnicas que se emplean, así como la terminología usada a lo largo de todo el documento.

- **Capítulo 3: Diseño y desarrollo.**

  En este capítulo se detallan todos los elementos que forman parte del entorno experimental del sistema implementado, describiendo la base de datos de partida, la tecnología de reconocimiento empleada, así como el procedimiento para evaluar la precisión de los experimentos. Por otro lado, se explican todas las fases de las que consta el sistema QbE-STD desarrollado.

- **Capítulo 4: Pruebas y resultados.**

  En este capítulo se muestran los distintos tipos de experimentos que se han realizado para parametrizar el sistema desarrollado y conseguir así los mejores resultados posibles. Además, se comparan los resultados obtenidos con algunos que ya han sido publicados en evaluaciones oficiales.

- **Capítulo 5: Conclusiones y trabajos futuros.**

  En este capítulo se plasman las conclusiones finales a las que se llega tras completar este trabajo, así como futuras líneas de trabajo e investigación que se proponen.

Capítulo 2

Estado del arte

2.1 Introducción a los sistemas QbE-STD

Para realizar la búsqueda de información de audio, existen distintas aproximaciones [3]: Spoken Document Retrieval (SDR), que consiste en la recuperación de una lista de documentos de audio; Keyword Spotting (KWS), que trata la búsqueda de palabras clave; Spoken Term Detection (STD) o detección de términos hablados; Query-by-Example (QbE), que consiste en encontrar un fragmento de audio o query; etc.

Este Trabajo Fin de Máster se ha centrado en el desarrollo de un sistema QbE-STD, sistemas que aúnan las técnicas QbE y STD.

QbE se emplea para buscar un ejemplo de un objeto o de una parte de él, en otro objeto.

Debido a la gran capacidad de almacenamiento digital que existe hoy en día, se aplican técnicas QbE en varios ámbitos tales como la música [4]. En esta última década se han desarrollado multitud de sistemas denominados Music Information Retrieval (MIR), en los que la query puede ser un ejemplo de una canción, o bien un fragmento de la misma cantado (Query by Singing) o tarareado (Query by Humming) por el usuario.

Se ha utilizado también la técnica QbE en la medición de similitud de imágenes en términos de color, textura y/o forma [5].

Otro ámbito del uso de QbE es la búsqueda de movimientos que identifiquen y extraigan aquellos que son propios de un individuo [6].

STD tiene como fin encontrar de forma rápida y precisa palabras o secuencias de palabras en archivos de audio [7], donde la entrada del sistema es una consulta en formato texto o audio.

Generalmente la entrada suele estar basada en texto. En caso de que la entrada sea en formato audio, alguna de las etapas que intervienen en el proceso varían.

Las técnicas **QbE-STD** plantean, por tanto, un escenario en el cual el usuario ha encontrado información de su interés en un repositorio o documento de audio y desea buscar información semejante en el mismo. La query de la búsqueda consiste en un fragmento del audio del repositorio o bien una o varias grabaciones del usuario pronunciando el término de interés.

2.2 Descripción de un sistema QbE-STD

En la siguiente figura se representa el esquema típico de un sistema QbE-STD [8].

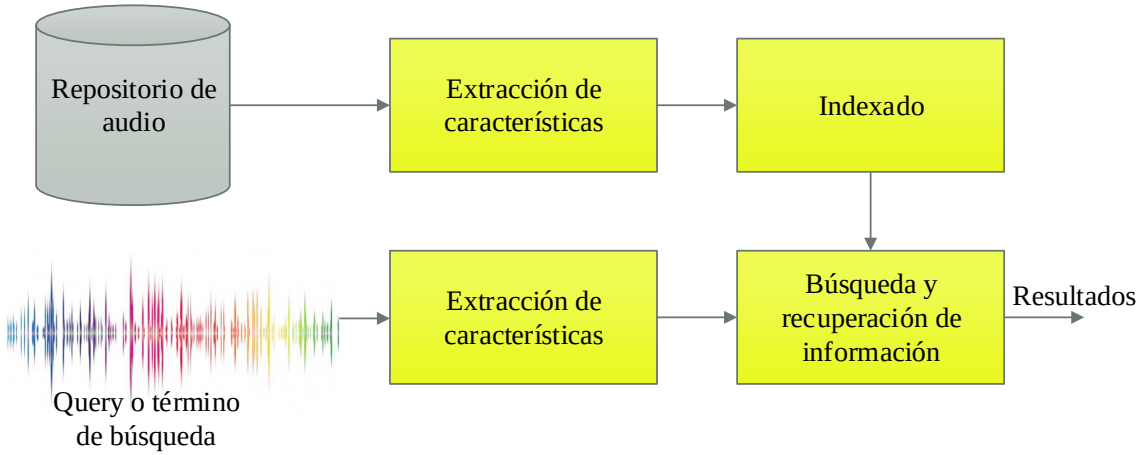

Figura 1 - Diagrama de bloques de un sistema QbE-STD

Los elementos/fases principales que aparecen son los siguientes:

- **Query/término de búsqueda.**

  Secuencia de audio que contiene la palabra o palabras que se quieren buscar.

- **Repositorio de audio.**

  Base de datos que contiene el audio en el que se quiere detectar las ocurrencias del término o términos que componen la query.

- **Extracción de características.**

  Tanto la query como el repositorio se procesan para extraer sus características relevantes.

  En los sistemas en los que la query está basada en texto, se suele realizar la decodificación en lattices (grafos que representan secuencias de palabra/subpalabra más probables), empleando para ello un sistema de reconocimiento automático de voz (Automatic Speech Recognition, ASR) [3].

  Esta fase se explica en más profundidad en la sección 2.2.2.

- **Indexado.**

  Fase que permite organizar los datos que se han extraído del conjunto total de datos sobre el que buscar, de manera que las búsquedas resulten eficientes.

- **Búsqueda y recuperación de información.**

  Búsqueda de palabra o palabras de interés en el índice, que en este caso es el conjunto de características extraídas del repositorio.

  Esta fase se explica en más profundidad en la sección 2.2.3

- **Resultados.**

  Hipótesis de ocurrencias de la query en el repositorio junto con un índice temporal y una medida de confianza para cada una de ellas, determinando así cuáles son válidas por su fiabilidad.

2.2.1 Dependencia del idioma

De cara a la elección de los sistemas QbE-STD, uno de los factores más importantes que deben tenerse en cuenta es el idioma de la query y/o del repositorio.

Una solución QbE-STD puede ser dependiente o independiente del idioma, en función de si el conocimiento del mismo es necesario o posible de obtener tanto para el repositorio como para la query. La independencia del idioma es necesaria en aquellos escenarios en los que no se disponga de suficientes recursos para entrenar un sistema de reconocimiento de voz, o en una situación multiidioma [1]. Por ejemplo, el método STD basado en texto asume que se disponen de los recursos necesarios sobre el idioma, pudiéndose emplear gran cantidad de transcripciones, diccionarios de pronunciaciones, etc [7].

En función de la cantidad de recursos que se disponen sobre el idioma, se puede hacer la siguiente clasificación [1]:

- **Técnicas ASR.**

  Se emplea un reconocedor de voz entrenado en el idioma de interés.

- **Sistemas con pocos recursos.**

  Se emplea algún modelado que no requiere demasiado conocimiento sobre los datos a procesar.

- **Sistemas con cero recursos.**

  No se utiliza ningún modelo, recurso o información sobre el idioma para realizar la búsqueda.

Hay muchas investigaciones centradas en la dependencia con los idiomas y los métodos QbE-STD empleados en función de esta dependencia [1] [7] [9] [10] [11]. En los trabajos referenciados se destaca el reto importante en los sistemas independientes del idioma (métodos cross-lingual), que es mejorar las tasas de error obtenidas.

2.2.2 Extracción de características

La extracción de características es una tarea de mucha importancia, ya que la eficiencia de esta fase influye en el comportamiento final de las siguientes fases, y por tanto, del sistema completo [12].

Las características de un audio tratan de mantener la información relevante para la aplicación y descartar la no relevante. En los sistemas QbE-STD interesan las características que permitan mantener el contenido lingüístico.

Estas características se extraen tanto del repositorio o base de datos como de la query del usuario, y la elección de cuáles usar en un sistema QbE-STD sigue siendo a día de hoy un problema presente en la investigación [1].

La característica más usual en este tipo de sistemas, al igual que en muchas otras tareas de reconocimiento de voz, consiste en extraer los coeficientes **Mel Frequency Cepstral Coefficients (MFCC)**, que se basan en la percepción auditiva humana.

En la Figura 2 se puede ver el diagrama de bloques del cálculo de los coeficientes MFCC.

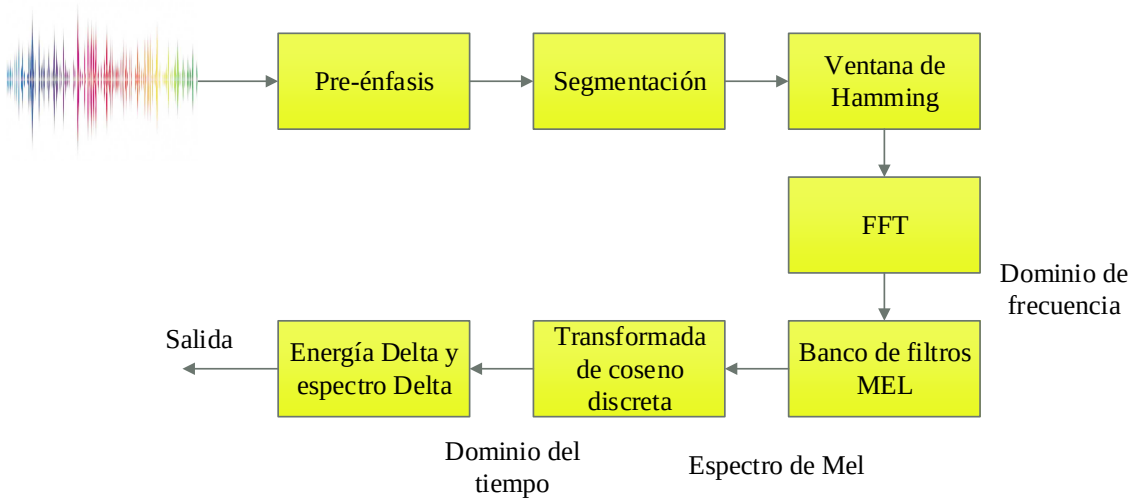

Figura 2 - Diagrama de bloques para el cálculo de MFCC

A continuación, se explican brevemente cada una de las etapas [12]:

1. **Pre-énfasis.**

   Se aplica un filtro que enfatiza las frecuencias altas, es decir, se incrementa la energía de las frecuencias más altas.

2. **Segmentación.**

   Conversión de la señal analógica en señal digital con tamaños del frame entre 20 a 40 milisegundos.

3. **Ventana de Hamming.**

   Se emplea para evitar las discontinuidades al principio y final de los bloques de la señal analizados.

4. **Transformada rápida de Fourier (Fast Fourier Transform, FFT).**

   Utilizada para convertir cada frame de un número determinado de muestras, del dominio del tiempo al dominio de la frecuencia.

5. **Banco de filtros Mel.**

   Una serie de filtros triangulares se emplean para obtener una salida que siga la escala Mel.

6. **Transformada de coseno discreta.**

   Proceso para convertir el espectro en escala Mel al dominio del tiempo. El resultado de esta conversión son los llamados MFCC, cuyo conjunto de valores forman los vectores acústicos.

7. **Energía y parámetros Delta.**

   A los vectores acústicos se les añade la energía y características relacionadas con los cambios en las características cepstrales a lo largo del tiempo.

Además de los MFCC, en las distintas investigaciones se han empleado características como los **posteriorgramas fonéticos**.

Una de las características más comúnmente utilizadas en la técnica DTW son estos posteriorgramas fonéticos, mediante los cuales se representa la probabilidad a posteriori de cada clase o unidad fonética (distintas para cada idioma) por cada frame temporal del audio en cuestión [2][9]. Para cada frame se obtienen, por tanto, vectores de dimensión igual al número de unidades fonéticas.

Los posteriorgramas fonéticos suelen estar representados por 3 estados: principio, centro y fin del fonema. La probabilidad de cada unidad fonética en cada frame se calcula como la suma de las probabilidades de los 3 estados para dicha unidad y frame [13].

Los posteriorgramas pueden calcularse empleando para ello un reconocedor fonético completo (modelo acústico y modelo del lenguaje) que genere lattices, obteniendo dichos posteriorgramas directamente a partir de los lattices. También se pueden calcular a partir de reconocedores fonéticos con modelo de lenguaje sencillos, por ejemplo bigramas, que generen directamente dichos posteriorgramas.

Los posteriorgramas pueden ser obtenidos mediante distintas herramientas basadas en redes neuronales profundas (DNN) [10]: por ejemplo con un reconocedor fonético de KALDI o BUT (Brno University of Technology).

Está ampliamente demostrado que los posteriorgramas son unas buenas características a emplear, ya que llevan embebida información lingüística a múltiples escalas temporales (sílabas, prosodia) [14].

En la Figura 3 se puede ver una representación de posteriorgramas para diferentes unidades fonéticas a lo largo de los distintos frames de un audio.

Además de la representación mediante MFCC y posteriorgramas, existen otras como los espectrogramas, representando la energía de distintos rangos de frecuencia; las características de cuello de botella, obtenidas a partir de Perceptrones Multicapa (Multi Layer Perceptron, MLP) donde una de las capas ocultas es típicamente muy pequeña y se emplea para la obtención de las características; los coeficientes Perceptual Linear Prediction (PLP), derivados de los coeficientes de precisión lineal que incluyen conocimientos acerca de la percepción auditiva humana; etc.

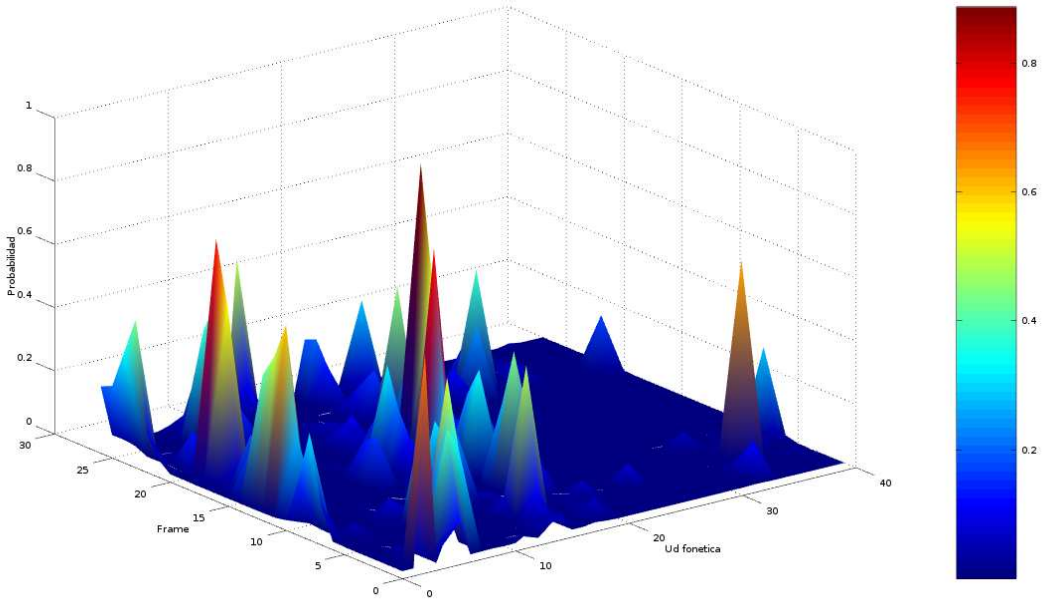

Figura 3 - Representación de posteriorgramas

En el enfoque con cero recursos se suelen emplear características acústicas extraídas de la forma de onda, tales como MFCC, mientras que en los escenarios con pocos recursos suelen utilizarse representaciones mediante posteriorgramas fonéticos [2].

Existen distintas líneas de trabajo en relación a la **extracción de características**: usar un conjunto completo de características frente a seleccionar las más relevantes en función de distintos criterios.

Para poder seleccionar características se hace imprescindible establecer una medición de relevancia de cada una de ellas, de manera que se escogen las más relevantes y se descartan el resto.

Generalmente, los resultados que se obtienen al reducir el conjunto de características mejoran notablemente frente a los conseguidos al emplear el conjunto total de las mismas.

El uso de un menor conjunto de datos implica una mejora en la eficiencia computacional. No obstante, actualmente los equipos informáticos son muy potentes y hacen posible usar tanta información como se tenga disponible sin suponer un problema de eficiencia.

A continuación, se explican brevemente algunos métodos de selección de características [1]:

- **Técnica de correlación.**
  Consiste en asignar un valor de relevancia a cada característica en función del camino o caminos de mejor alineamiento posible obtenidos mediante la técnica Dynamic Time Warping (DTW), que se explica en la sección 2.2.3 .
  Mediante los coeficientes de correlación de Pearson se obtiene la contribución de cada característica a una función de coste. Se establece que aquellas características que más contribuyen a una determinada función de coste son no relevantes, mientras que las que menos contribuyen se definen como las más relevantes.

- **Técnica de valle profundo.**
  Relacionada también con la técnica DTW, se basa en 2 principios: uno es el hecho de que cuanto menor sea el coste de las regiones vecinas del camino óptimo, mejor; el otro consiste en la afirmación de que cuanto más profundo sea el valle (relativo a un gráfico coste por cada frame) alrededor del camino óptimo, mejor.
  En la Figura 4 se puede ver cómo cuanto mejor sea el alineamiento, el coste en las zonas vecinas es menor (regiones con colores más blancos).
  En la Figura 5 se muestra el coste en cada frame de la zona marcada con un rectángulo azul en la Figura 4, y se puede comprobar que cuanto mejor sea el alineamiento, más profundo es el valle en la gráfica del coste.

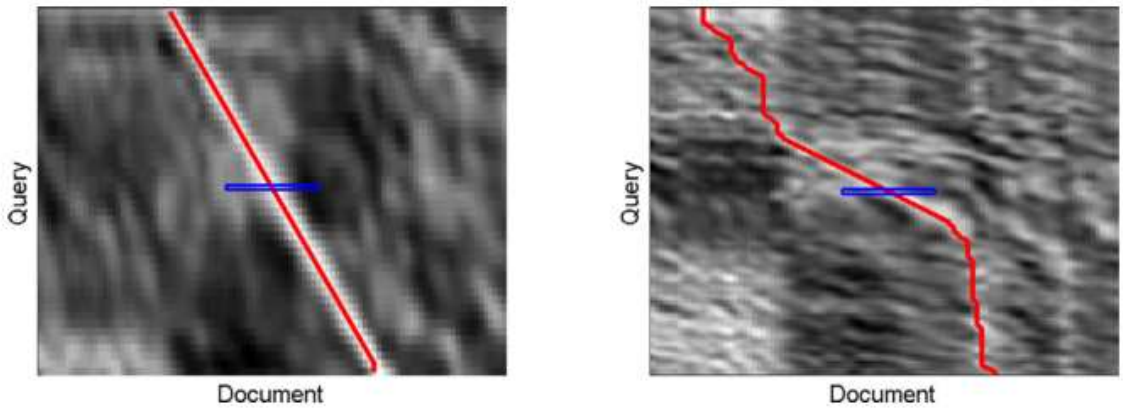

Figura 4- Alineamiento de una query que aparece exactamente en el repositorio (izquierda) y la misma query grabada por otro locutor (derecha) [1]

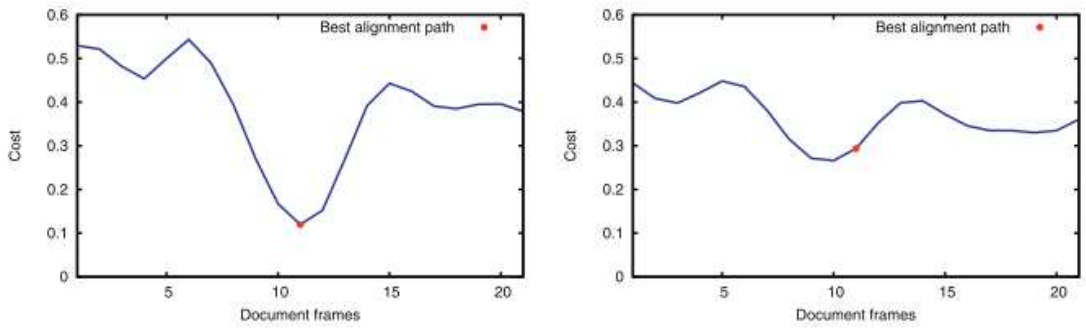

Figura 5 - Coste cuando la query aparece exactamente en el repositorio (izquierda) y cuando la misma query es grabada por otro locutor (derecha) [1]

Otro enfoque distinto relacionado con la selección de características es el de reducir la dimensionalidad del conjunto total de características, empleando para ellos métodos como Principal Component Analysis (PCA).

En la Figura 6 se observa cómo varía la precisión del sistema en función del método de selección de características y el número de características escogidas.

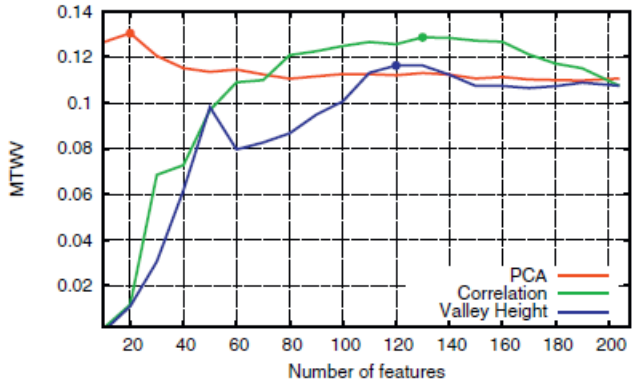

Figura 6 - Precisión usando selección de características [1]

2.2.3 Búsqueda y recuperación de información

Los sistemas QbE-STD se pueden clasificar en 2 categorías distintas en función de las características extraídas y de los métodos empleados para la detección de la query:

- **Acoustic Keyword Spotting (AKWS).**

  Este método consiste en construir un modelo de la query (keyword model) para obtener su transcripción fonética mediante técnicas basadas en redes neuronales y modelos ocultos de Markov [15]. Posteriormente, se emplea log-likelihood entre el modelo de la query y el modelo del repositorio de búsqueda.

  Por lo tanto, este método se emplea con mejores resultados en sistemas supervisados, donde se dispone de transcripción fonética del repositorio y se puede llevar a cabo un entrenamiento del modelo [16].

- **Pattern/template-matching.**

  Este método consiste en calcular la similitud a nivel de características entre la query y los segmentos de audio del repositorio, y suele emplearse para ello la técnica DTW.

DTW permite encontrar un alineamiento óptimo entre 2 secuencias dependientes del tiempo bajo ciertas restricciones [6]. Estas secuencias suelen estar formadas por características muestreadas en puntos equidistantes en el tiempo.

Se establece una medida de coste o distancia local (distancia euclídea, distancia coseno, producto escalar, distancia Kullback-Leibler, coeficientes de correlación de Pearson, etc.), que tendrá un valor pequeño si las 2 secuencias son semejantes. De esta manera, se construye una matriz de coste, siendo el objetivo encontrar un alineamiento entre las secuencias con el menor coste posible.

Pueden existir varios alineamientos con la misma distancia o coste, por lo que el llamado camino óptimo no tiene por qué ser único.

Para poder determinar el camino óptimo se emplean técnicas de programación dinámica, donde surge el concepto de matriz de coste acumulado *D*. Por lo tanto, estableciéndose un conjunto de condiciones iniciales, se obtiene la matriz de coste acumulado, y a partir de ella

la distancia DTW, siendo ésta el coste total del mejor alineamiento posible (con menor coste).

En la Figura 7 se observa una matriz de coste y su correspondiente matriz de coste acumulado, donde el camino óptimo aparece resaltado en blanco.

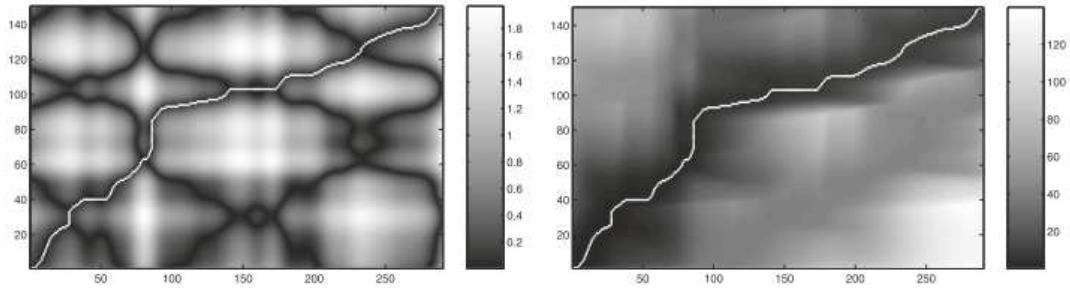

Figura 7 - Matriz de coste (izquierda) y matriz de coste acumulado (derecha) [6]

Con el objetivo de acelerar los cálculos y minimizar el coste computacional, se pueden llevar a cabo distintas acciones: modificar las condiciones del tamaño del paso (frame), añadir pesos locales para favorecer el alineamiento horizontal/vertical/diagonal, incluir constantes globales como las regiones Sakoe-Chiba o Itakura [6], aplicar reducción de dimensionalidad, etc.

Además, existen otras variaciones de esta técnica:

- **DTW multiescala (MsDTW).**

  La estrategia consiste en encontrar de forma recursiva un alineamiento óptimo en una resolución mayor, y posteriormente, proyectarlo en un espacio de resolución menor con el fin de refinarlo.

  En la Figura 8 se observa el camino óptimo en un determinado nivel (a), y el camino óptimo con respecto a la región obtenida de proyectar la región (a) en un nivel con mayor resolución.

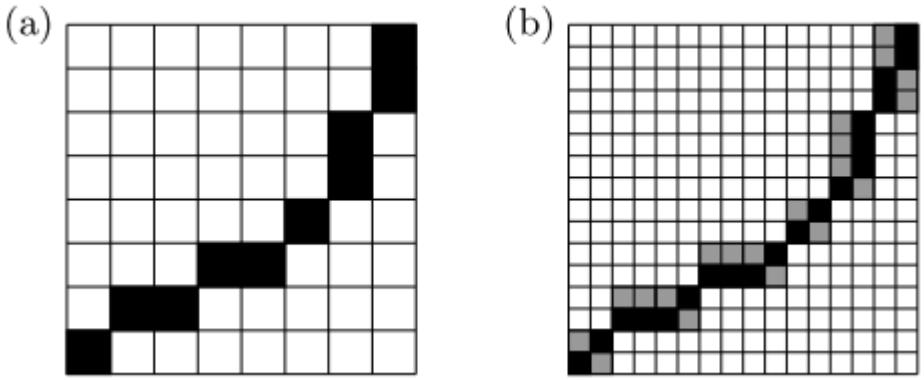

Figura 8 – Aplicación de técnica MsDTW [6]

- **DTW de subsecuencias (S-DTW).**

  Empleado en los casos en los que las secuencias a comparar tienen longitudes muy diferentes. En estos casos, el objetivo es encontrar un fragmento de la secuencia de mayor longitud que se alinee con la secuencia de menor longitud.

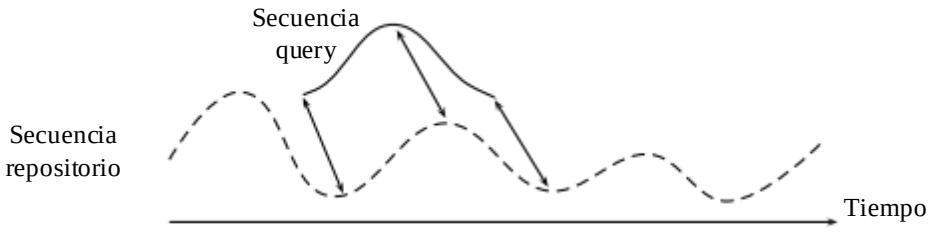

Figura 9 - Alineamiento entre secuencia de una query con una subsecuencua de un repositorio [6]

Como acción complementaria, es posible incorporar un detector de actividad de voz (Voice Activity Detector, VAD), de manera que se eliminen los silencios al principio y al final de la secuencia. Esto ha permitido conseguir una mejora en las prestaciones en torno a un 10% [16].

DTW es cuasi-óptimo en situaciones con cero o pocos recursos, así como en escenarios multiidioma [2] [16].

2.2.4 Calibración

Los sistemas explicados anteriormente tienen como objetivo obtener un conjunto de hipótesis sobre la detección de la query en el repositorio de búsqueda; teniendo asociado cada una de estas hipótesis un score o medida de confianza/verosimilitud.

Existen multitud de trabajos que incorporan un proceso de calibración en los sistemas QbE-STD. Hay transformaciones del score que se basan en enfoques comúnmente empleados en reconocimiento de hablantes, tales como la fusión y adaptación discriminativa para obtener un conjunto completo de scores [17]. Otros métodos emplean suavizado para obtener una distribución puramente Gaussiana [1] [2] [18], normalizaciones en función de la longitud del documento [15], z-norm para obtener una misma distribución de los scores [1], normalización m-norm y fusión de subsistemas basados en regresión logística [16] [19] o normalización q-norm dependiente de la query [20].

2.3 Fusión de sistemas QbE-STD

La fusión de subsistemas es una técnica común en las tareas de procesamiento de voz que ha aportado importantes resultados, concretamente en el área de los sistemas QbE-STD. La fusión consiste en combinar los scores obtenidos de cada uno de los subsistemas de cara a obtener una única salida.

Una de las soluciones desarrolladas es fusionar múltiples sistemas DTW aplicando regresión logística [10].

Se pueden encontrar trabajos de investigación donde se ha llevado a cabo la fusión de multitud de subsistemas AKWS y DTW [15] [16] que maximizan la entropía cruzada, alinean temporalmente los resultados de cada subsistema y realizan una combinación lineal de los mismos.

En la Figura 10 se muestra el diagrama de bloques de un sistema donde se fusionan subsistemas DTW y AKWS.

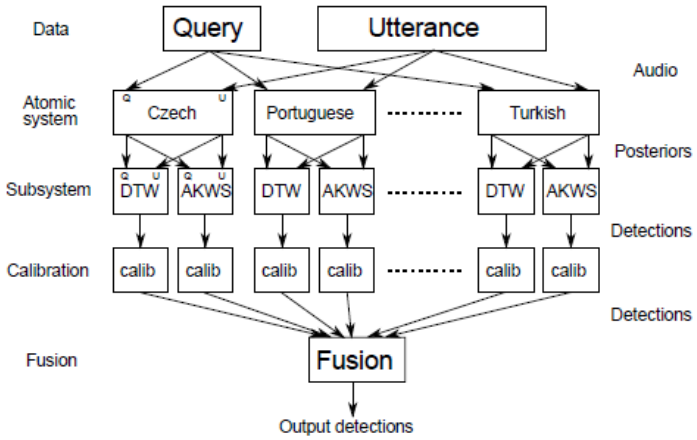

Figura 10 - Diagrama de bloques de un sistema QbE-STD con fusión DTW y AKWS [16]

Otras investigaciones realizan una fusión en el sentido de que emplean la estimación de reconocedores en varios idiomas [17] [19] [21].

Un enfoque diferente de fusión consiste en emplear un único sistema pero varios ejemplos de una misma query, de manera que se obtienen distintos scores para cada una de ellas. Posteriormente se fusionan los mismos con el fin de obtener un único score por query [9] [18].

Para llevar a cabo un proceso de fusión es necesario realizar primero algún tipo de calibración como los explicados en la sección 2.2.4.

Además, los parámetros involucrados en la fusión de los sistemas deben ser optimizados de cara a obtener los mejores resultados posibles.

2.4 Precisión de sistemas QbE-STD

La métrica de evaluación más extendida en estos sistemas es **ATWV** (Actual Term Weighted Value) propuesta por NIST (National Institute of Standard and Technology) de Estados Unidos. Esta métrica integra la tasa de no detecciones y las falsas alarmas de cada query en una única medida que, posteriormente, se promedia para todas las queries.

$$ATWV(\theta) = 1 - average\{P_{Miss}(term, \theta) + \beta \cdot P_{FA}(term, \theta)\} \quad (1)$$

Donde $P_{Miss}(term, \theta)$ es la probabilidad de que una ocurrencia no se haya detectado dado el umbral θ, $P_{FA}(term, \theta)$ es la probabilidad de insertar falsas ocurrencias/alarmas dado el umbral θ, y

$$\beta = \frac{C}{V}(PR_{term}^{-1} - 1) \tag{2}$$

Donde PR_{term} es la probabilidad del término, que suele fijarse a 10^{-4}; y $\frac{C}{V} = 0.1$, que es un factor de peso para dar más o menos relevancia a las falsas alarmas o a las ocurrencias no encontradas.

Por lo tanto, cuanto más próximo a 1 sea el valor de ATWV, mejor será la precisión del sistema.

Por otro lado, la métrica **MTWV** (Maximum Term Wighted Value) se define como el valor máximo ATWV que puede ser obtenido por un sistema teniendo en cuenta todos los posibles valores de θ empleados, y seleccionando el θ óptimo.

En la Tabla 1 se muestran algunos datos de estas métricas obtenidos por distintos sistemas QbE-STD, así como la evaluación en la que se llevaron a cabo y el conjunto de datos que se emplearon:

SISTEMA	ATWV	MTWV
DNN + posteriorgrama [3] (Albayzin 2014, test data)	0,2881	0,2894
SGMM+posteriorgrama [3] (Albayzin 2014, test data)	0,5167	0,5167
Posteriorgrama + DTW + Selección [2] (Albayzin 2014, test data)	0,2020	0,2140
Posteriorgrama + DTW + Selección [18] (Albayzin 2016, dev data)	0,2180	0,2180
Gaussian posteriorgrama + DTW [18] (Albayzin 2016, dev data)	0,1757	0,1831
Feat extraction + DTW [17] (MediaEval 2013, test data)	0,3989	0,3994

Tabla 1 - Resultados de sistemas QbE-STD

Como puede comprobarse en la Tabla 2, la precisión conseguida mediante la fusión de varios subsistemas es mayor.

	ATWV	MTWV
Posterior + DTW [13] Fusión (MediaEval 2014, test data)	0,5841	0,6096
Posterior + DTW **Relevance-feedback** **Fusión [20]** (MediaEval 2012, test data)	0,7430	-
AKWS + DTW posterior **Fusión [16]** (MediaEval 2013, test data)	0,3751	0,3776
Posterior + DTW **Gaussian posterior + DTW** **Acoustic feat + DTW** **Fusión [18]** (Albayzin 2016, dev data)	0,2750	0,2800

Tabla 2 - Resultados de sistemas QbE-STD con fusión de subsistemas

También son ampliamente utilizadas las **curvas DET** (Detection Error Tradeoff), que se muestran en la Figura 11, y que evalúan las prestaciones de un sistema QbE-STD en varias ratios de no detecciones y falsas alarmas.

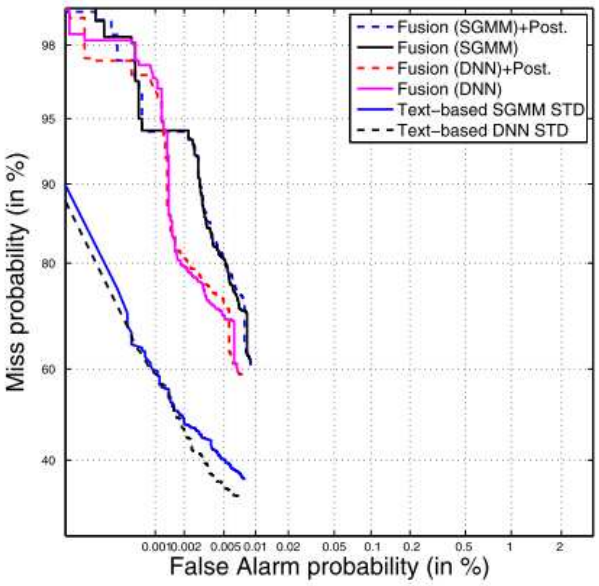

Figura 11 – Ejemplo de curvas DET [3]

Capítulo 3

Diseño y desarrollo

3.1 Elección del sistema a desarrollar

De todos los sistemas analizados en la revisión del estado del arte, se ha decidido desarrollar un sistema QbE-STD basado en **posteriorgramas** como característica y **DTW** como método de búsqueda. Los motivos por los que se ha decidido implementar este tipo de solución son los siguientes:

- Se ha comprobado que el uso de posteriorgramas como característica de representación del audio ofrece muchas posibilidades de cara a futuros trabajos de investigación.
- Para la descomposición en posteriorgramas se cuenta con el ampliamente utilizado reconocedor de BUT, que dispone de modelos acústicos del checo, inglés, húngaro y ruso; y que ofrece una buena precisión en sus resultados.
- Se busca un escenario de pocos o cero recursos donde no se dispone de conocimiento previo del idioma de la query y el repositorio. La técnica DTW ofrece buenos resultados en estos casos concretos.
- La longitud de la query y el repositorio de búsqueda será previsiblemente muy diferente, por lo que se opta por escoger la variación S-DTW.

3.2 Entorno experimental

El desarrollo de este Trabajo Fin de Máster se ha implementado en un entorno con Sistema Operativo Linux.

La ejecución de los experimentos se ha llevado a cabo de manera remota en un servidor de la Universidad que cuenta con 12 núcleos de cálculo en paralelo, y que está asociado a un servidor de discos en red con redundancia y varios TB de espacio.

3.2.1 Base de datos

En todo sistema QbE-STD es necesario contar con al menos una query y un repositorio en el que buscar dicha query.

En este desarrollo se han empleado los datos de la evaluación ***Albayzin 2016***, cuyo workshop fue parte del congreso IberSpeech 2016[3]; y los correspondientes a la evaluación ***Albayzin 2018*** que actualmente se encuentra en curso.

- En la evaluación Albayzin 2016 se contó con 2 bases de datos [22]: MAVIR (conjuntos *train*, *dev* y *test*), extraída del workshop MAVIR[4] en 2006, 2007 y 2008 y en idioma español; y EPIC (sólo conjunto de *test*), extraída de conversaciones en español grabadas en el Parlamento Europeo en el año 2004.

  Todos los audios que intervienen en la evaluación tienen el siguiente formato: PCM, 16 KHz, un único canal y 16 bits por muestra.

  El conjunto de datos utilizado durante toda la fase de desarrollo ha sido el conjunto ***dev*** de la base de datos MAVIR. Este set de datos está formado por 2 ficheros de audio con una duración en total de 1 hora, que componen el repositorio de búsqueda. Se dispone, además, de 102 audios que corresponden a las queries con los términos a buscar.

  En la fase de pruebas, se han empleado también los audios del conjunto de ***test*** de MAVIR. Este set de datos está formado 3 audios con una duración total de 2 horas, y 106 queries.

- En la evaluación Albayzin 2018 [23] se dispone de 3 bases de datos: MAVIR (conjuntos *train*, *dev* y *test*), usada previamente en la evaluación Albayzin 2016; COREMAH (sólo conjunto de *test*), que contiene audios de personas nativas y no nativas hablando en español en un entorno universitario, grabadas en 2014 y 2015; y RTVE (conjuntos *train*, *dev* y *test*), que consiste en programas grabados de Radio Televisión Española entre 2015 y 2018.

  El formato de los audios de las bases de datos MAVIR y COREMAH es PCM, 16 KHz, un único canal y 16 bits por muestra. El formato de los audios de RTVE es AAC, stereo, 44.1KHz y una tasa de bit variada.

  El conjunto de datos ***dev*** de RTVE se ha incorporado en la fase de pruebas. Este set está compuesto por 15 horas en total, que corresponden a 12 repositorios y 103 queries.

Para la implementación del sistema QbE-STD no se han empleado datos de ***train*** o entrenamiento ya que se ha planteado un escenario en el que no se tiene conocimiento previo del idioma, es decir, un escenario de cero recursos donde no se lleva a cabo ningún tipo de modelado.

[3] https://iberspeech2016.inesc-id.pt/

[4] http://www.mavir.net

3.2.2 Reconocedor fonético BUT

Con el fin de decodificar los audios tanto de la query como del documento, se han utilizado los reconocedores fonéticos desarrollados por la Universidad de Tecnología de Brno (BUT)[5] [24]. Estos reconocedores, cuyo principal fin es la investigación, han sido utilizados con éxito en distintas tareas tales como identificación del idioma [25], indexado y búsqueda de grabaciones de audio y búsqueda de palabras clave [26].

Los reconocedores BUT pueden ser usados tanto en entorno Windows como Linux. Para la compilación de los reconocedores se necesitan tener instaladas las librerías ATLAS (Automatically Tuned Linear Algebra Software)[6] y BLAS (Basic Linear Algebra Software)[7].

Existen reconocedores BUT en los siguientes idiomas: inglés (EN), checo (CZ), húngaro (HU) y ruso (RU). Todos ellos han sido empleados en la fase de pruebas del sistema QbE-STD desarrollado.

Los reconocedores de los idiomas checo, húngaro y ruso se han entrenado con las bases de datos SpeechDat-E (8 KHz), mientras que el sistema de idioma inglés se ha entrenado con la base de datos TIMIT (16 KHz). Debido a esta diferencia en la frecuencia de muestreo, los experimentos llevados a cabo con el idioma inglés parten de audios con frecuencia de muestreo de 16 KHz, mientras que los experimentos en checo, húngaro o ruso implican una conversión de los audios a 8 KHz mediante la herramienta Sox[8].

Por otro lado, cada uno de los idiomas está representado por un número de unidades *U*:

IDIOMA	UNIDADES	UNIDADES NO FONÉTICAS
EN	39	1
CZ	45	3
HU	61	3
RU	52	3

Tabla 3 - Número total de unidades y unidades no fonéticas de cada idioma.

En la Tabla 3 se muestra el número total de unidades de cada idioma, teniendo en cuenta que algunas de ellas son unidades no fonéticas correspondientes a ruidos intermitentes, ruidos que no proceden del habla y pausas cortas.

[5] http://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context

[6] http://math-atlas.sourceforge.net/

[7] http://www.netlib.org/blas/

[8] http://sox.sourceforge.net/

Cada unidad se representa por un modelo de 3 estados: inicio, centro y fin. Como para cada estado se calcula una probabilidad a posteriori, el reconocedor ofrece 3 probabilidades para cada unidad fonética, una correspondiente a cada estado.

Además, los posteriorgramas se extraen cada 10 ms, dando lugar así a una señal de *T* frames.

Teniendo en cuenta todo esto, la matriz de posteriorgramas para un audio determinado tiene un tamaño (*T*, 3***U*).

3.2.3 HTK

HTK es un conjunto de herramientas que permite generar modelos ocultos de Markov (Hidden Markov Models, HMMs) [27], y fue diseñada en primera instancia para construir soluciones de procesamiento del habla basadas en HMMs.

HList es un programa que forma parte de HTK, mediante el cual se puede convertir en texto plano el contenido de una fuente de datos escrita en alguno de los formatos soportados por HTK.

El reconocedor BUT ofrece como resultado un fichero de posteriorgramas en formato HTK, por lo que es necesario el uso de HList.

3.2.4 NIST

Tal y como se ha comentado en la sección 2.4, se emplea el sistema de evaluación propuesto por NIST, en el que la métrica principal propuesta es ATWV.

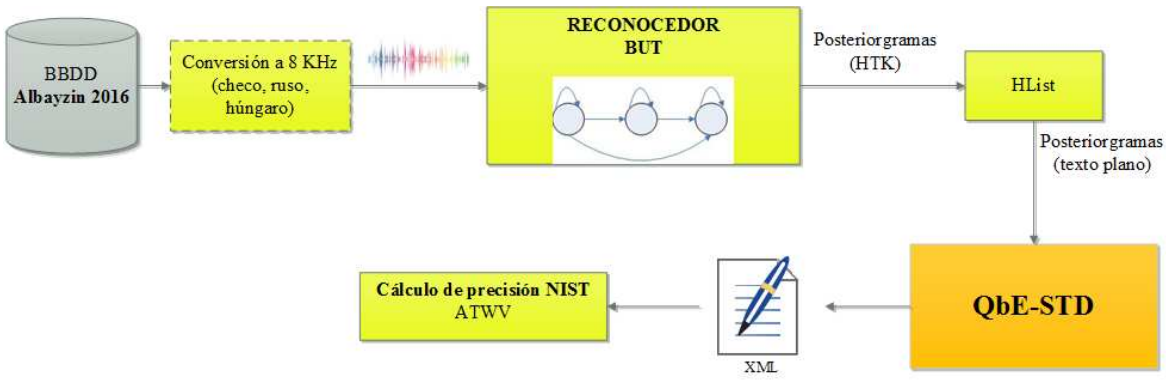

Figura 12 - Esquema completo del entorno del sistema QbE-STD desarrollado

En el conjunto de documentos que se proporcionan en las evaluaciones ***Albayzin 2016 y 2018 Search on Speech***, se entrega también un script para evaluar los resultados de los sistemas. Para poder usarlo, las detecciones deben escribirse en un fichero con formato XML y estructurado de una manera concreta. En la Figura 13 se puede ver un ejemplo de un fichero

XML en el que aparecen las ocurrencias que se han encontrado de 3 queries en un repositorio.

```
<stdlist termlist_filename=
"/opt/Experimentos/MC/Evaluaciones/development/scoring/QbESTD/mcabello/results_EN_06-Aug-2018_154250.xml" indexing_time=
"1.000" language="spanish" index_size="1" system_id="fake">
<detected_termlist termid="DEV-0002" term_search_time="0.5" oov_term_count="1">
</detected_termlist>
<detected_termlist termid="DEV-0003" term_search_time="0.3" oov_term_count="1">
<term file="mavir03" channel="1" tbeg="1195.690" dur="0.280" score="-0.03393" decision="YES"/>
</detected_termlist>
<detected_termlist termid="DEV-0007" term_search_time="0.5" oov_term_count="1">
<term file="mavir03" channel="1" tbeg="352.050" dur="0.150" score="-0.08310" decision="YES"/>
<term file="mavir03" channel="1" tbeg="1576.960" dur="0.420" score="-0.09187" decision="YES"/>
<term file="mavir03" channel="1" tbeg="1854.330" dur="0.400" score="-0.09371" decision="YES"/>
</detected_termlist>
</stdlist>
```

Figura 13 - Ejemplo de fichero XML de salida

A continuación, se indica el significado de los campos más importantes que se pueden encontrar en estos ficheros de resultados:

- **`termlist_filename`**: nombre del fichero XML y ruta completa del mismo.
- **`termid`**: nombre del audio de la query.
- **`term_search_time`**: duración de la query en segundos.
- **`file`**: nombre del audio del documento o repositorio.
- **`tbeg`**: instante en el que tiene su inicio la ocurrencia de la query en el repositorio, indicado en segundos.
- **`dur`**: duración de la ocurrencia encontrada, indicada en segundos.
- **`score`**: puntuación de la ocurrencia, fiabilidad de la misma.
- **`decision`**: campo en el que se indica si la ocurrencia se da por válida (YES) o no (NO), bajo ciertos criterios del desarrollador.

Al ejecutar el script proporcionado para la evaluación, se obtienen una serie de ficheros de resultados entre los que destacan los siguientes:

- **`score.occ.txt`**: fichero resumen en el que aparecen los valores de ATWV, MTWV, falsas alarmas, ocurrencias no detectadas, score óptimo, etc,
- **`score.ali.txt`**: listado detallado de todas las ocurrencias; las detectadas correctamente, las falsas alarmas y las que no se han detectado.
- **`score.det.png`**: gráfico con curva DET del experimento en cuestión.

3.3 Desarrollo del sistema QbE-STD

Se ha desarrollado un sistema en Matlab que permite, en la medida de lo posible, la máxima automatización. Para ello, todos los parámetros de relevancia para el algoritmo se introducen como parámetros de entrada:

- Listado de queries a buscar.
- Listado de documentos en los que buscar las ocurrencias de las queries.
- Idioma del reconocedor con el que se quiere llevar a cabo el experimento (CZ/HU/RU/EN).
- Parámetros numéricos relacionados con el algoritmo DTW de búsqueda de la información, τ, τ_2 y neigh, y explicados en la sección 3.3.2.
- Indicador de selección de unidades fonéticas (yes/no), explicado en la sección 3.3.3.
- Indicador de fusión (yes/no), explicado en la sección 3.3.4.

3.3.1 Preprocesado de posteriorgramas

Como se ha explicado en la sección 3.2.2, a la salida del reconocedor BUT se obtiene una matriz de probabilidades a posteriori de tamaño (*T*, 3**U*), donde *T* es el número de frames de 10 ms, *U* es el número de unidades fonéticas del idioma del reconocedor, y cada una de estas unidades está representada por 3 estados.

A continuación se muestra la estructura de la matriz de posteriorgramas que se obtiene del reconocedor, tras aplicarle el formato de texto plano mediante el programa HList.

$$P = \begin{pmatrix} p_{1,1,t_1} & p_{1,2,t_1} & p_{1,3,t_1} & \cdots & p_{U,1,t_1} & p_{U,2,t_1} & p_{U,3,t_1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ p_{1,1,T} & p_{1,2,T} & p_{1,3,T} & \cdots & p_{U,1,T} & p_{U,2,T} & p_{U,3,T} \end{pmatrix}$$

La probabilidad a posteriori de cada unidad fonética *u* en cada frame *t* se calcula como la suma de los posteriors de sus estados *s* [13]:

$$p_{u,t} = \sum_{s=1}^{3} p_{u,s,t} \qquad \textbf{(3)}$$

De esta manera, el nuevo tamaño de la matriz de posteriorgramas es (*T*, *U*).

$$P = \begin{pmatrix} p_{1,t1} & \cdots & p_{U,t1} \\ \vdots & \ddots & \vdots \\ p_{1,T} & \cdots & p_{U,T} \end{pmatrix}$$

3.3.2 Algoritmo de búsqueda DTW

Con el fin de comparar las secuencias de la query y el repositorio buscando un alineamiento entre ellas que indique una ocurrencia, se emplea el método S-DTW, es decir, la versión DTW de subsecuencia [6], mencionada en la sección 2.2.3.

En la Figura 1 se muestran los pasos que se han seguido para acometer esta tarea.

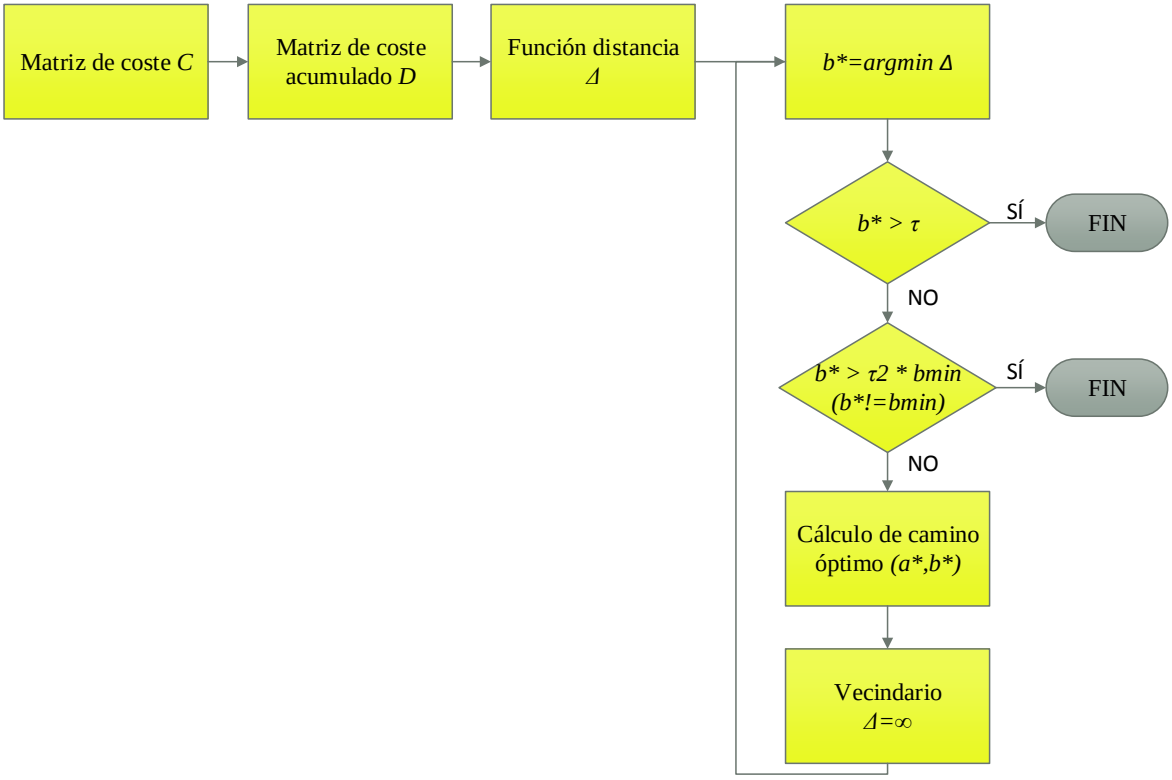

Figura 14 - Diagrama de bloques del algoritmo S-DTW implementado

- **Matriz de coste.**

  Para comparar las 2 secuencias de audio (query *X* y repositorio *Y*) se necesita una medida de coste local, denominada también medida de distancia local.

  Se define la query como $X(1:n) \coloneqq (x_1, \dots, x_n)\ para\ n \in [1:N]$ y el repositorio $Y(1:m) \coloneqq (y_1, \dots, y_m)\ para\ m \in [1:M]$.

  El coste será pequeño si las secuencias son similares entre ellas, y será más alto si no lo son.

  La matriz de coste es el resultado de calcular el coste entre todos los elementos de la matriz de posteriorgramas de la query y del repositorio.

  En este sistema QbE-STD se ha escogido como medida de coste el coeficiente de correlación de Pearson *r*, ya que ofrece mejores prestaciones que otras métricas [1] [2]. Este coeficiente es independiente de la escala de medida de las variables, ofreciendo información sobre el grado de relación de dos variables cuantitativas.

$$r(x_n, y_m) = \frac{U(x_n \cdot y_m) - \|x_n\|\|y_m\|}{\sqrt{(U\|x_n^2\| - \|x_n\|^2)(U\|y_m^2\| - \|y_m\|^2)}} \quad \textbf{(4)}$$

Como en el coeficiente de correlación de Pearson los valores altos corresponden a costes bajos y viceversa, se aplica la siguiente transformación para mapear el resultado en el rango [0,1]:

$$c(x_n, y_m) = \frac{1 - r(x_n, y_m)}{2} \quad \textbf{(5)}$$

Por lo tanto, en función del valor del coeficiente de correlación de Pearson *r*, el coste *c* puede tener uno de los siguientes valores:

- $r = -1 \rightarrow c = 1.$
- $r = 0 \rightarrow c = 0{,}5.$
- $r = 1 \rightarrow c = 0.$

Además de este mapeo, se ha testeado la precisión del sistema realizando una modificación en el cálculo del coste de la siguiente manera:

- $r \leq 0 \rightarrow r = 0$
- $c(x_n, y_m) = 1 - r(x_n, y_m)$

(6)

Es decir, para todos los valores de coeficiente de correlación menor o igual que 0, el coste será máximo, potenciando así que la diferencia entre las secuencias que se alinean y las que no, sea más clara.

- **Matriz de coste acumulado.**

La matriz de coste acumulado *D* en el caso del método S-DTW se construye de la siguiente manera:

$$D(n,1) = \sum_{k=1}^{n} c(x_k, y_1)\ para\ n \in [1:N]$$
$$D(1,m) = c(x_1, y_m)\ para\ m \in [1:M] \quad \textbf{(7)}$$
$$D(n,m) = min\{D(n-1,m-1), D(n-1,m), D(n,m-1)\} + c(x_n, y_m)$$
$$para\ el\ resto\ de\ los\ casos$$

Se define además la matriz de coste acumulado extendida como la matriz resultante de añadirle una columna a la matriz de coste acumulado $D(n,0) \coloneqq \infty\ para\ n \in [0:N]$; y una fila $D(0,m) \coloneqq 0\ para\ m \in [0:M]$.

En la Figura 15 se puede observar tanto la matriz de coste como la matriz de coste acumulado en un caso en el que existe una ocurrencia de la query en el repositorio. Las regiones más oscuras corresponden a zonas donde los costes son menores, es decir, donde existe una mayor correlación o alineamiento entre query y

repositorio. El camino óptimo de la ocurrencia se detecta con claridad en la franja diagonal que comienza en torno al frame 200 del eje de abscisas.

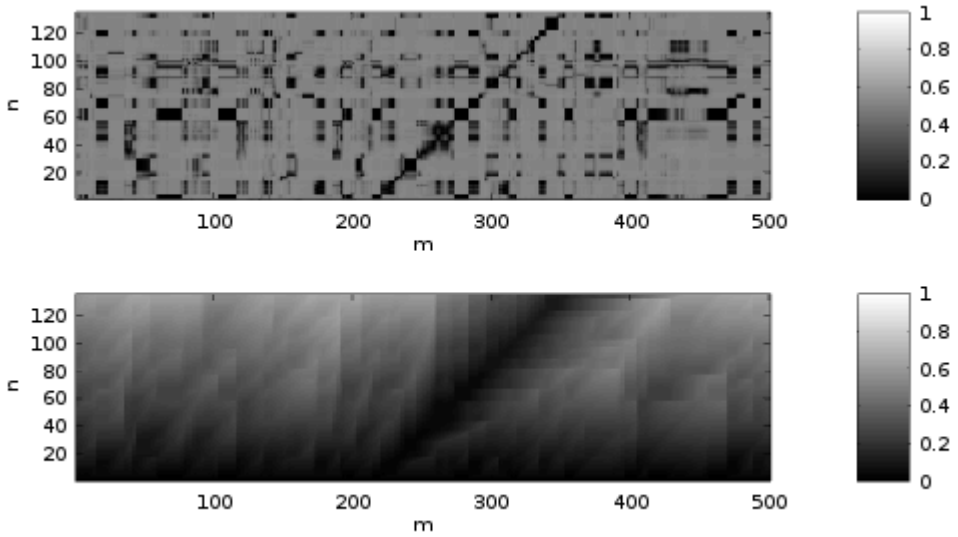

Figura 15 – Matriz de coste (arriba) usando Pearson. Matriz de coste acumulado (abajo)

En la Figura 16 se muestra la matriz de coste y de coste acumulado para la misma query y documento de la Figura 15. Sin embargo, para el cálculo de la matriz de coste se ha empleado la variación del coeficiente de correlación de Pearson mencionada en el punto anterior. Se observa que, efectivamente, con este cálculo las variaciones de los costes son mucho más bruscas; y la matriz de coste acumulado presenta mayores costes acumulados en zonas de no ocurrencia.

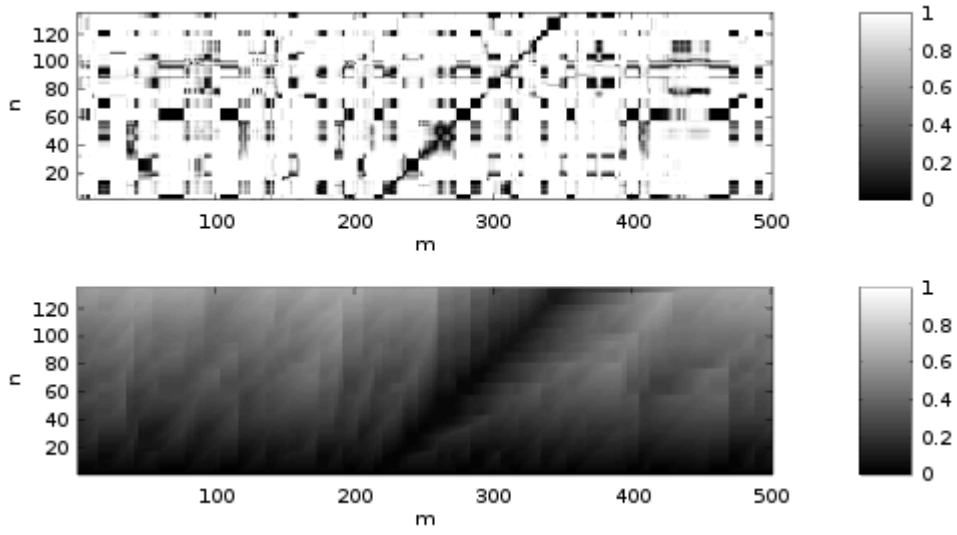

Figura 16 - Matriz de coste (arriba) usando Pearson con la modificación. Matriz de coste acumulado (abajo)

- **Función distancia Δ.**

  La función distancia corresponde en realidad a la última fila de la matriz de coste acumulado, es decir, $D(N, m)$, y se utiliza para calcular los valores mínimos a partir de los cuales se calculan los caminos de cada una de las ocurrencias.

  En la Figura 17 se muestra la función distancia del mismo ejemplo mostrado en las anteriores figuras. Se puede apreciar el mínimo global b^*, denominado también *bmin* y marcado con una línea roja, en la región de menor coste de la matriz de coste acumulado (mostrada abajo).

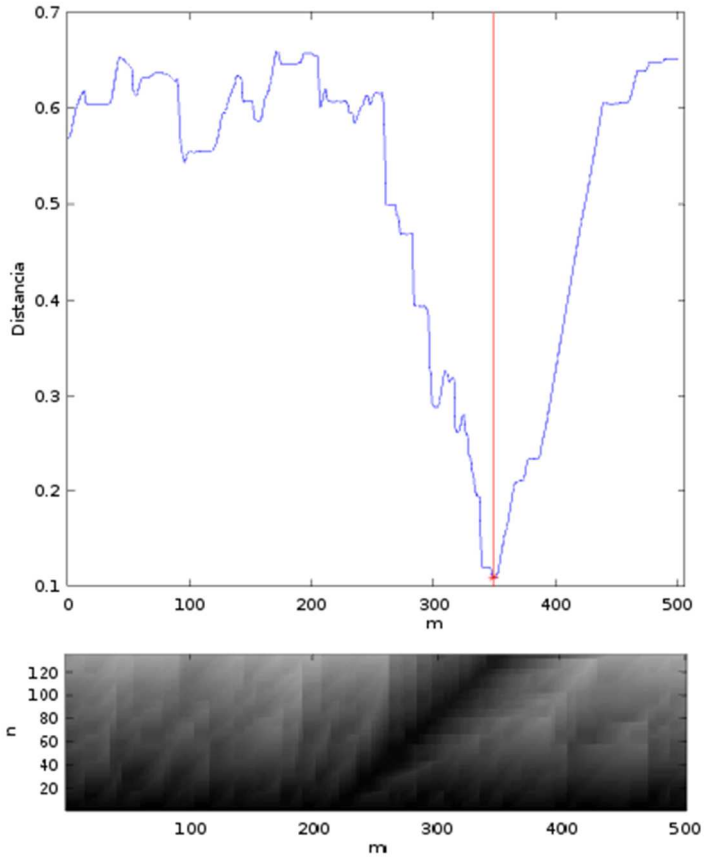

Figura 17 – Función distancia (arriba) de la matriz de coste acumulado (abajo)

- **Características de *b**.**

  Primeramente, es necesario aclarar que todos los mínimos de la función distancia, tanto globales como locales, se denominan *b**. Sin embargo, únicamente el mínimo global es denominado *bmin*.

$$b^* \coloneqq argmin\, D(N, b)|_{b\in[1:M]} \quad \textbf{(8)}$$

  El punto *b**, independientemente de si corresponde a un mínimo global o local, debe ser siempre menor que un umbral τ específico. El valor de dicho umbral es configurable y permite descartar los *b** o, por el contrario, continuar con el siguiente paso del algoritmo.

Por otro lado, como no tiene por qué existir un único camino óptimo, se van calculando de forma iterativa los valores *b**, tal y como se muestra en la Figura 14.

Todos los *b**, excepto el correspondiente al mínimo global *bmin*, deben cumplir una segunda condición para emplearse en el cálculo de un camino óptimo. Esta condición está relacionada con la variación máxima de *b** respecto a *bmin*. Es decir, los mínimos locales pueden ser un máximo de τ_2 veces más grandes que el mínimo global *bmin*.

Con estas 2 condiciones se intenta descartar falsos positivos, al evitar detectar como ocurrencias caminos óptimos con probables costes elevados.

- **Cálculo del camino óptimo.**

Para cada valor *b** que cumple con la o las condiciones anteriormente descritas, se procede a calcular el camino óptimo $p = (p_1, \dots, p_L)$. Es decir, se quiere encontrar una secuencia (*a**, *b**), donde *a** ϵ [1:*M*] y *b** ϵ [1:*M*] con el menor coste posible.

Para ello, se comienza a calcular el camino en orden inverso, es decir, se comienza por el punto $p_L = b^*$. El cálculo finaliza cuando se alcanza *n=1*, es decir, la primera fila de la matriz *D*:

$$p_{l-1} \coloneqq argmin\{D(n-1, m-1), D(n-1, m), D(n, m-1)\} \quad \textbf{(9)}$$

En la Figura 18 se muestra en color rojo el camino óptimo calculado a partir del mínimo global correspondiente a la Figura 17.

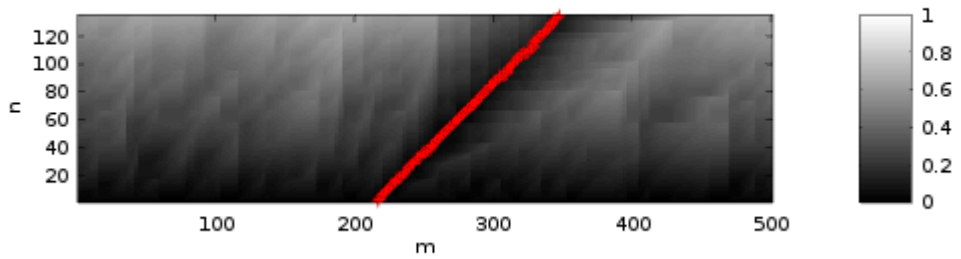

Figura 18 – Camino óptimo

Para cada camino encontrado se calcula el instante de inicio de la ocurrencia, el instante de fin y un score o medida de confianza para dicho camino.

- **Vecindario**

Por último, una vez calculado un camino óptimo, se desea evitar encontrar otros caminos demasiado próximos. Por ejemplo, en este sistema se descartan todos los caminos cuyo mínimo *b** se encuentre a menos de 500 ms (50 frames de 10 ms)

de cualquiera de los ya calculados para determinada query y repositorio. Esto es así ya que es poco probable que una persona repita la misma palabra en una conversación en menos de 500 ms.

Para descartar estos caminos próximos, se asigna un valor ∞ a los puntos de la función distancia próximos a b^*:

$$\Delta(b) \coloneqq \infty \; para \; todo \; b \; \in [b^* - t, b^* + t] \tag{10}$$

Donde t=50, que corresponde al número de frames en los que no queremos buscar ninguna ocurrencia.

De este modo, a la hora de calcular un b^*, este valor no se encontrará en el vecindario de ninguno de los b^* de los caminos óptimos previamente calculados.

Llegado este punto, se procede a buscar el siguiente b^* hasta que cualquiera de ellos deje de cumplir alguna de las condiciones explicadas, momento en el cual finalizaría el proceso de búsqueda.

3.3.3 Selección de unidades fonéticas

Además del sistema QbE-STD básico, se ha incluido una selección de unidades fonéticas [2]. El objetivo es implicar en el proceso de búsqueda únicamente a aquellas unidades fonéticas más relevantes con el fin de mejorar el proceso de búsqueda de ocurrencias.

Se asume que las unidades fonéticas que más contribuyen en el coste de un camino óptimo son las menos relevantes; mientras que aquellas que implican un menor coste son las más relevantes. Por lo tanto, se desea conocer cuál es el coste aportado a un camino concreto por cada una de ellas de manera individual.

Dada una query *X* y un repositorio *Y*, asumiendo que la query está presente en el repositorio, se calcula el camino óptimo *P(X,Y)* de longitud *K*. Para cada uno de los elementos (x_n, y_m) del camino, se calcula la contribución de cada unidad fonética *u* al coste o correlación de Pearson:

$$r(x_n, y_m, u) = \frac{Ux_{n,u}y_{m,u} - \frac{1}{U}\,\|x_n\|\|y_m\|}{\sqrt{(U\|x_n^2\| - \|x_n\|^2)(U\|y_m^2\| - \|y_m\|^2)}} \tag{11}$$

Tras esto, se define la contribución de cada unidad fonética como:

$$c(x_n, y_m, u) = \frac{1 - r(x_n, y_m, u)}{2} \tag{12}$$

Así, se puede obtener la relevancia *R* de una unidad fonética como:

$$R(P(x,y),u) = \frac{1}{K}\sum_{k=1}^{K} c(x_k, y_k, u) \tag{13}$$

Para realizar una buena estimación, dado un conjunto de queries y sus ocurrencias en ciertos repositorios, las relevancias de las distintas unidades fonéticas de cada una de las ocurrencias se suman.

Mediante todos estos cálculos se pueden ordenar las unidades fonéticas por orden de relevancia, teniendo en cuenta que las que tengan un menor valor *R* serán las más relevantes.

A partir de este punto, es decisión del desarrollador emplear más o menos unidades fonéticas en el proceso.

3.3.4 Fusión

En la sección 2.3 se ha comentado que la forma más habitual de fusionar sistemas QbE-STD consiste en la fusión de las ocurrencias con los scores obtenidos por cada uno de los subsistemas utilizados, ya sean distintos tipos de subsistemas o el mismo subsistema con distintos idiomas.

Sin embargo, en este Trabajo Fin de Máster se ha realizado un nuevo tipo de fusión con el objetivo de comprobar qué precisión se alcanza.

El método empleado consiste en, dada una query *X*, un repositorio *Y*, y un conjunto de idiomas, calcular la matriz de coste acumulada *D* para cada uno de los idiomas y combinarlas, obteniendo así una única matriz en la que realizar la búsqueda del camino óptimo.

Como cada idioma tiene características diferentes en cuanto a la distribución de los valores de los costes, las matrices de coste acumulado de los distintos idiomas no pueden sumarse tal cual. Por ello, a la matriz de coste acumulado de cada idioma se le aplica una normalización *z-norm*, mediante la cual se normalizan tanto la media como la desviación típica de dicha matriz.

Una vez hecho esto, se suman las matrices de coste acumulado de los idiomas implicados y se normaliza a escala [0-1].

Finalmente, se procede con el algoritmo de búsqueda para encontrar el camino óptimo.

Capítulo 4

Pruebas y resultados

A continuación se explican las distintas pruebas que se han realizado con los datos de las evaluaciones ***Albayzin 2016 y 2018 Search on Speech***, así como la precisión obtenida en cada una de ellas.

Se han llevado a cabo más de 55 experimentos diferentes que suponen aproximadamente 50 horas de ejecución del algoritmo, en los que han ido variando los parámetros configurables (τ y τ_2), el idioma (CZ/EN/HU/RU), las unidades fonéticas empleadas, etc.

En todos los experimentos, salvo que se especifique lo contrario, se ha utilizado el conjunto completo *dev* de Albayzin 2016. Los experimentos realizados con el conjunto de *test* de Albayzin 2016 y el conjunto *dev* de Albayzin 2018 aparecen debidamente indicados.

4.1 Variación en la matriz de coste

Tal y como se ha mencionado en la sección 3.3.2, se ha llevado a cabo una modificación en el mapeo del coste con los correspondientes coeficientes de correlación de Pearson, posibilitando que la diferencia entre las secuencias que se alinean y las que no, sea mayor.

A pesar de que gráficamente sí se observa la diferencia entre ambas versiones de la matriz de coste (ver Figura 15 y Figura 16), a la hora de realizar los experimentos no se aprecia diferencia notable en los resultados de los mismos. Esto se puede comprobar en la Tabla 4.

Tipología de matriz de coste	ATWV	MTWV	Score
mapping [-1,0,1] → [1,0.5,0]	0.1770	0.1823	-0.0821
mapping [< 0,1] → [-1,0]	0.1770	0.1823	-0.0776

Tabla 4 – Valores de precisión en función de la versión de la matriz de coste, conjunto dev Albayzin 2016

4.2 Parametrización de τ

Como se ha explicado en la sección 3.3.2, si los valores del mínimo global o los mínimos locales b^* superan el umbral τ, se finaliza el proceso de búsqueda. Es decir, este parámetro cuyo rango es [0,1], impone el valor máximo que puede tener el primer elemento del camino óptimo.

En la Figura 19 se muestran ejemplos, para cada uno de los idiomas, de la variación en las ratios de falsas alarmas (*False Alarm*) y ocurrencias no detectadas (*Missing*) en función de distintos valores de τ. En todos estos experimentos se ha mantenido constante el valor τ_2.

Se observa que los valores altos de τ implican una situación poco restrictiva, dando lugar a más falsas alarmas pero menos ocurrencias sin detectar. Sin embargo, los valores bajos de τ implican una situación más restrictiva, dando lugar a más ocurrencias sin detectar pero menos falsas alarmas.

En la Tabla 5 se puede comprobar cuáles son los valores de ATWV para cada uno de los experimentos mostrados en la Figura 19. Como la métrica ATWV se calcula en función de las falsas alarmas y las ocurrencias no detectadas (ver en las ecuaciones **(1** y **(2**), se obtienen mejores valores con valores menores de τ.

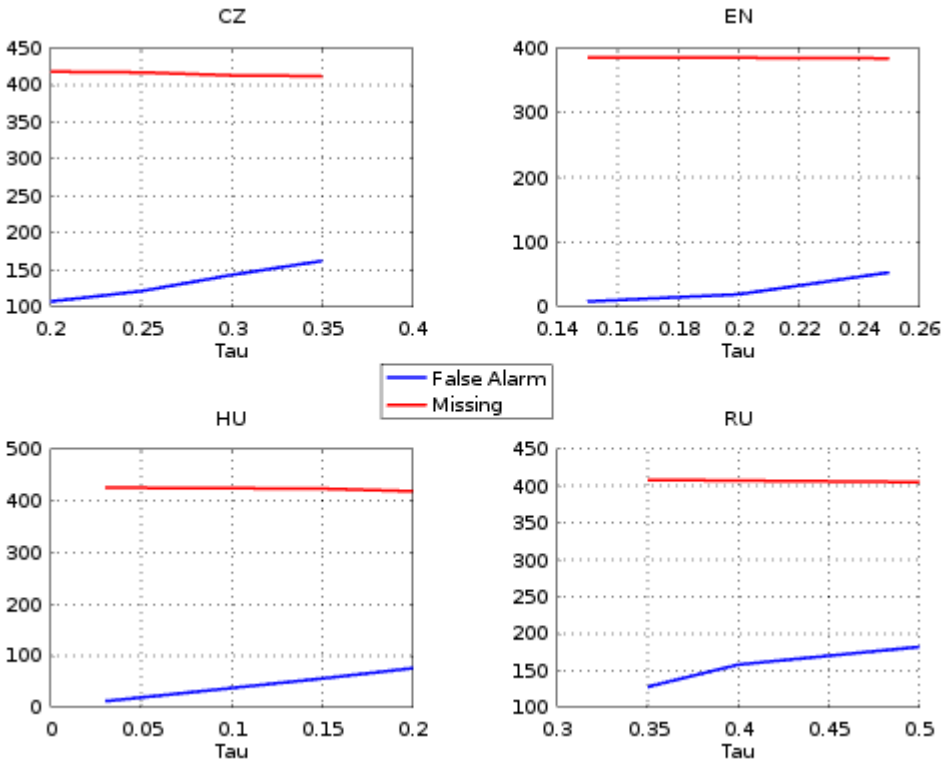

Figura 19 – Variación de falsas alarmas y no detecciones en función de τ, conjunto dev Albayzin 2016

TAU	CZ ATWV	EN ATWV	HU ATWV	RU ATWV
0.03	-	-	-0.0417	-
0.15	-	0.1770	-0.1213	-
0.20	-0.2475	0.1562	-0.1448	-
0.25	-0.2979	0.0788	-	-
0.30	-0.3496	-	-	-
0.35	-0.3887	-	-	-0.2462
0.40	-	-	-	-0.2978
0.50	-	-	-	-0.3556

Tabla 5 – ATWV de cada idioma en función de τ, conjunto dev Albayzin 2016

4.3 Parametrización de τ_2

Tal y como se ha comentado en la sección 3.3.2, los mínimos locales obtenidos en la función distancia Δ deben ser, como máximo, τ_2 veces mayores que el mínimo global *bmin*. Este umbral se establece porque, al poder existir más de un camino óptimo de una query en un documento, se necesita establecer un límite para el cálculo de los sucesivos caminos.

En la Figura 20 se observa la variación en la tasa de falsa alarma y no detecciones en función de τ_2 para todos los idiomas.

Al igual que ocurre con el parámetro τ, cuanto mayor es el valor de τ_2, más ocurrencias falsas se detectan. Sin embargo, el número de ocurrencias sin detectar permanece prácticamente constante. Esto nos hace pensar que hay casos que no se están pudiendo detectar con el algoritmo, independientemente de los valores τ y τ_2 que se parametricen.

En la Tabla 6 se muestran los valores de ATWV en función de τ_2. Al permanecer prácticamente constante la tasa de no detecciones, los valores ATWV son mejores cuanto menor es el valor de τ_2, es decir, cuanto menor es la ratio de falsas alarmas. Añadir que no se han probado los mismos valores de τ_2 para todos los idiomas ya que se ha ido parametrizando en función de los resultados: tasa de falsa alarma, no detecciones, detecciones correctas, ATWV, MTWV; por lo que en algunos casos era más interesante una variación mayor y en otros probar con valores más próximos.

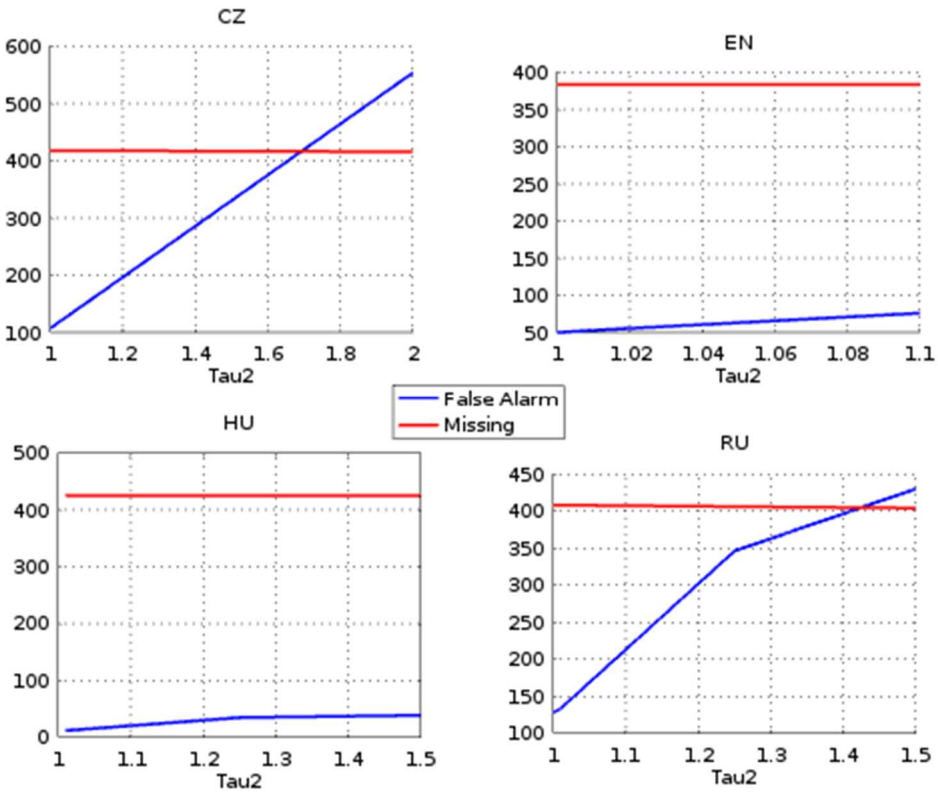

Figura 20 - Variación en falsas alarmas y no detecciones en función de τ_2, conjunto dev Albayzin 2016

τ_2	CZ ATWV	EN ATWV	HU ATWV	RU ATWV
1	-0.2475	0.0830	-	-0,2462
1.01	-	0.0788	-0.0417	-0.2587
1.1	-	0.0331	-	-
1.25	-	-	-0.1294	-0.7596
1.5	-	-	-0.1419	-0.9882
2	-1.2720	-	-	-

Tabla 6 – Valores de ATWV en función de τ_2, conjunto dev Albayzin 2016

En la Tabla 7 aparece la combinación de τ-τ_2 que mejor precisión ha dado lugar en cada uno de los idiomas.

IDIOMA	τ	τ_2
CZ	0.20	1
EN	0.15	1.01
HU	0.03	1.01
RU	0.25	1.01

Tabla 7 – Mejor combinación τ-τ_2

El sistema QbE-STD implementado es una solución **independiente del lenguaje**. Es por ello que se puede emplear para búsquedas en cualquier idioma de query y documento, no teniendo por qué ser en ninguno de los idiomas en los que han entrenado los reconocedores BUT.

Por otro lado, existen idiomas que, por sus características fonéticas, son mejor o peor reconocidos en función del idioma del reconocedor.

Todo esto implica que para un mismo audio, en función del idioma del reconocedor que se haya usado, los rangos de variación de los costes son diferentes, y del mismo modo, los valores τ y τ_2.

4.4 Selección de unidades fonéticas

Una vez escogida la mejor parametrización en cada uno de los idiomas, se proceden a realizar los experimentos teniendo en cuenta la relevancia de las unidades fonéticas. Las unidades de cada idioma se ordenan en función de su relevancia según el procedimiento descrito en la sección 3.3.3. Posteriormente, se procede a ejecutar el experimento con la mejor parametrización encontrada, empleando para ello un número determinado de características.

En la Tabla 8 se observa la precisión obtenida, en términos de ATWV, para cada idioma en función del número de unidades relevantes utilizadas. Al lado del idioma aparece, entre paréntesis, el número de unidades fonéticas que tiene cada idioma en total.

IDIOMA	10 uds	20 uds	30 uds	40 uds	50 uds	Todas las uds
CZ (45)	-0.4475	-0.3115	-0.3191	-0.3140	-	**-0.2475**
EN (39)	0.1421	0.1729	0.1721	-	-	**0.1770**
HU (61)	-0.1154	-0.0792	-0.0667	-0.0667	-0.0625	**-0.0417**
RU (52)	-0.4275	-0.2699	-0.2042	-0.2030	-	**-0.0255**

Tabla 8 – ATWV en función de las unidades fonéticas seleccionadas, conjunto dev Albayzin 2016

Como se puede observar, los mejores resultados (resaltados en negrita) se alcanzan empleando en el experimento todas las unidades fonéticas, es decir, contando con toda la información posible que nos ofrece el reconocedor.

4.5 Resultados destacables

Se muestran ahora información sobre los resultados del experimento que mejor precisión ha alcanzado, correspondiente al idioma inglés (τ=0.15, τ_2=1.01).

IDIOMA	PFA	PMISS	ATWV	MTWV	SCORE
EN	0.00002	0.801	**0.1770**	**0.1823**	-0.0821

Tabla 9 – Resultados del experimento con mejor precisión (EN), conjunto dev Albayzin 2016

Donde *PFA* corresponde con la probabilidad de falsa alarma; *PMISS* es la probabilidad de no detección; ATWV y MTWV las métricas ya explicadas; y por último *SCORE* que es el valor de confianza de la ocurrencia.

Si se comparan estos resultados con los obtenidos por el sistema QbE-STD desarrollado por la Universidad de Vigo (el sistema ganador de la evaluación Albayzin 2016 en QbE-STD), evaluando el mismo conjunto de datos ***dev***, se puede concluir que el sistema desarrollado ha alcanzado unos niveles competitivos:

IDIOMA	PFA	PMISS	ATWV	MTWV
EN	0.00004	0.745	**0.2189**	0.2180

Tabla 10 – Resultados obtenidos por GTM-UVigo, conjunto dev Albayzin 2016 [10]

Con los scores obtenidos en estas ocurrencias se ha representado la curva DET de la Figura 21.

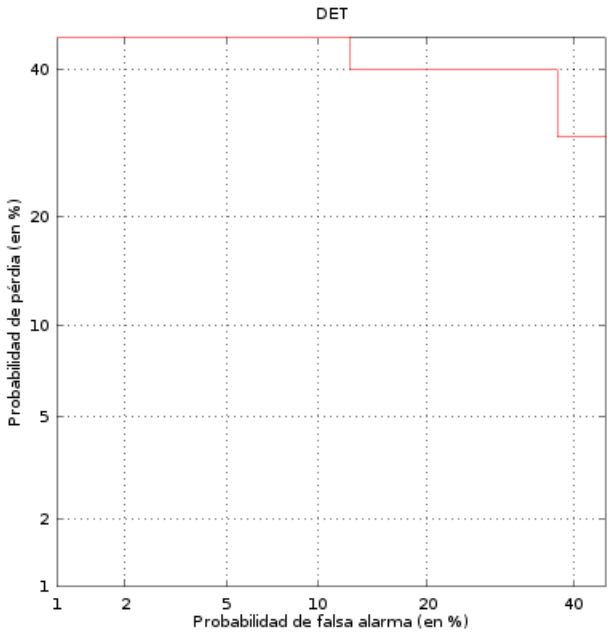

Figura 21 – Curva DET del experimento con mejor precisión (EN), conjunto dev Albayzin 2016

Llegado este punto, se evalúa el sistema correspondiente a este mejor experimento empleando el conjunto de ***test de Albayzin 2016***.

En la Tabla 11 se muestran los datos de precisión obtenidos. La probabilidad de falsa alarma es menor y la probabilidad de no detección ha aumentado un 5%. Teniendo en cuenta esto y el peso que tienen ambas probabilidades en la métrica ATWV (ver fórmula **(1)**), se explica esta disminución en la precisión.

IDIOMA	PFA	PMISS	ATWV	MTWV	SCORE
EN	0.00001	0.840	**0.1550**	0.1550	-0.1286

Tabla 11 – Resultados del experimento con mejor precisión, conjunto test Albayzin 2016

La curva DET de este experimento se muestra en la Figura 22, donde se puede concluir que, pese a que el valor ATWV es menor, las ocurrencias detectadas tienen un menor índice de falsa alarma.

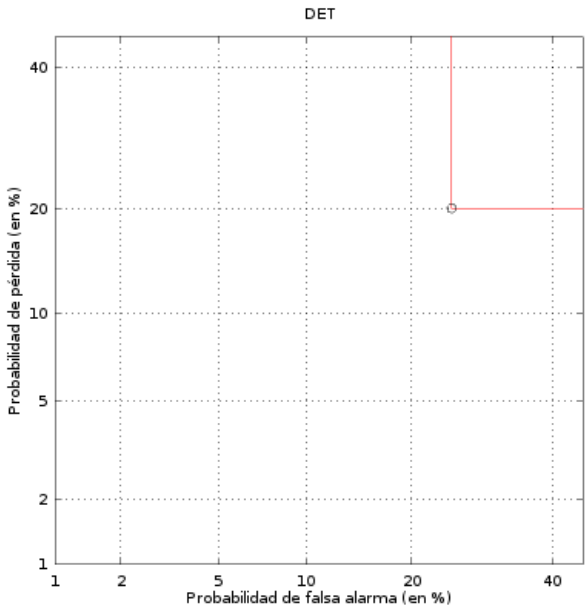

Figura 22 - Curva DET del experimento con mejor precisión (EN), conjunto test

Con el objetivo de comprobar si el sistema desarrollado es válido para la evaluación ***Albayzin 2018***, se han realizado 2 últimas pruebas replicando la parametrización usada en los experimentos de la evaluación Albayzin 2016. En ellos se han considerado las 103 queries y 2 de los 12 repositorios del conjunto ***dev***. A fecha de escritura de este documento todavía no se habían liberado los datos de test de la evaluación ***Albayzin 2018***.

En la Tabla 12 aparecen los valores de las métricas de estos 2 experimentos. El hecho de haber usado únicamente 2 de los 12 repositorios implica que estos valores no sean realmente representativos.

IDIOMA	REPOSITORIO	PFA	PMISS	ATWV	MTWV	SCORE
EN	millenium-20170522	0.00000	0.750	**0.2500**	0.2500	-0.0711
EN	LN24H-20160112	0.00001	0.794	**0.1981**	0.1981	-0.0604

Tabla 12 - Resultados del experimento con mejor precisión para 2 repositorios, conjunto dev Albayzin 2018

4.6 Fusión de idiomas

Empleando la fusión de idiomas explicada en la sección 3.3.4 no se ha conseguido mejorar el resultado del mejor experimento con un idioma aislado (EN).

Se han fusionado los idiomas mostrados en la Tabla 13, donde el mejor de los resultados aparece marcado en negrita. No obstante, este valor de ATWV es más bajo que el

experimento usando el idioma inglés, lo que no es de extrañar dada la gran diferencia en rendimiento entre el sistema del idioma inglés y el restox.

IDIOMA	ATWV	MTWV
EN + RU	**0.0168**	0.0651
EN+ RU + HU	0.0102	0.0887
EN+ RU + HU + CZ	-0.0582	0.0729

Tabla 13 – Métricas obtenidas mediante la fusión de idiomas, conjunto dev Albayzin 2016

Capítulo 5

Conclusiones y trabajos futuros

5.1 Conclusiones

A lo largo de este Trabajo Fin de Máster se ha conseguido implementar un sistema QbE-STD que, mediante el uso de los posteriorgramas fonéticos como característica de representación del habla, y del algoritmo DTW como método de búsqueda, sirve como punto de partida para futuras líneas de trabajo del grupo AUDIAS.

En el desarrollo del código se ha respetado la estructura de ficheros y directorios propuesta por la evaluación Albayzin Search on Speech, con el fin de que se pueda reutilizar fácilmente en próximas evaluaciones.

Este sistema se ha preparado para que sus resultados se puedan evaluar mediante las métricas más utilizadas: ATWV y MTWV. De este modo, la precisión alcanzada en cada experimento se puede comparar directamente con la de otras implementaciones desarrolladas por distintos equipos de investigación.

Se ha comprobado que los mejores resultados se obtienen utilizando el reconocedor de BUT entrenado en inglés. Hay que remarcar que todos los audios que han participado en el proceso de experimentación de este trabajo son en idioma castellano, por lo que la situación podría ser diferente en caso de evaluar audios en otros idiomas.

Se han incluido aportaciones propias al desarrollo básico, como son la fusión de idiomas y la selección de características, así como el estudio del comportamiento en función de los parámetros de ajuste y del tipo de matriz de coste.

Los valores de precisión alcanzados se pueden considerar competitivos, ya que se aproximan a los de los mejores sistemas presentados a las evaluaciones más recientes en este ámbito.

Por último, al comprobar el correcto funcionamiento del sistema y su suficiente grado de fiabilidad, se va a proceder a participar en la evaluación en curso ***Albayzin 2018***. Esto supondrá el primer sistema QbE-STD que participará por parte de la Universidad Autónoma de Madrid en dicha evaluación.

5.2 Trabajos futuros

Tras el desarrollo del sistema QbE-STD se abren nuevas vías de investigación y trabajo para el grupo AUDIAS con el fin de evolucionarlo y mejorarlo. Para ello, será interesante seguir optimizando los parámetros mediante la realización de pruebas con bases de datos y queries de distinta procedencia y/o distintos idiomas.

Por otro lado, al comprobar que la fusión desarrollada no ha ofrecido buenos resultados, será conveniente sustituir esta fusión por una fase de calibración y posterior fusión mediante regresión logística, empleando para ello las herramientas Bosaris[9] y/o FoCal[10].

[9] https://sites.google.com/site/bosaristoolkit/

[10] https://sites.google.com/site/nikobrummer/focal

Bibliografía

[1] P. Lopez-Otero, L. Docio-Fernandez, and C. Garcia-Mateo, “Finding relevant features for zero-resource query-by-example search on speech,” Speech Commun., vol. 84, pp. 24–35, 2016.

[2] P. Lopez-Otero, L. Docio-Fernandez, and C. Garcia-Mateo, “Phonetic unit selection for cross-lingual query-by-example spoken term detection,” 2015 IEEE Work. Autom. Speech Recognit. Understanding, ASRU 2015 - Proc., pp. 223–229, 2016.

[3] J. Tejedor, D. Torre-Toledano, P. Lopez-Otero, L. Docio-Fernandez, and C. Garcia-Mateo, “Comparison of ALBAYZIN Query-by-example Spoken Term Detection 2012 and 2014 Evaluations,” EURASIP J. Audio, Speech, Music Process., pp. 1–19, 2016.

[4] N. Borjian, “A Survey on Query-by-Example based Music Information Retrieval,” vol. 158, no. 8, pp. 31–34, 2017.

[5] B. Günsel and A. Murat Tekalp, “Shape similarity matching for query-by-example,” *Pattern Recognit.*, vol. 31, no. 7, pp. 931–944, 1998.

[6] M. Müller, *Information retrieval for music and motion*. 2007.

[7] J. Tejedor, M. Fapšo, I. Szöke, J. “Honza” Černocký, and F. Grézl, “Comparison of methods for language-dependent and language-independent query-by-example spoken term detection,” *ACM Trans. Inf. Syst.*, vol. 30, no. 3, pp. 1–34, 2012.

[8] G. Mantena, S. Achanta, and K. Prahallad, “Query-by-Example Spoken Term Detection using Frequency Domain Linear Prediction and Non-Segmental Dynamic Time Warping,” *Ieee/Acm Trans. Audio*, vol. 22, no. 5, pp. 946–955, 2014.

[9] T. J. Hazen, W. Shen, and C. White, “Query-By-Example Spoken Term Detection Using Phonetic Posteriorgram Templates,” pp. 421–426, 2009.

[10] P. Lopez-Otero, L. Docio-Fernandez, and C. Garcia-Mateo, “GTM-UVigo Systems for the Query-by-Example Search on Speech Task at MediaEval 2015,” Vigo, 2015.

[11] I. Szoke *et al.*, “Query by example search on speech at MediaEval 2015,” *CEUR Workshop Proc.*, vol. 1436, 2015.

[12] L. Muda, M. Begam, and I. Elamvazuthi, “Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques,” vol. 2, no. 3, pp. 138–143, 2010.

[13] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, “GTTS-Ehu systems for QUESST at MediaEval 2014,” *CEUR Workshop Proc.*, vol. 1263, no. 3, 2014.

[14] M. Cernak, A. Asaei, and H. Bourlard, “On structured sparsity of phonological posteriors for linguistic parsing,” *Speech Commun.*, vol. 84, pp. 36–45, 2016.

[15] A. Abad, R. F. Astudillo, and I. Trancoso, “The L2F spoken web search system for mediaeval 2013,” *CEUR Workshop Proc.*, vol. 1043, 2013.

[16] I. Szöke, L. Burget, F. Grézl, J. “Honza” Černocký, and L. Ondel, “CALIBRATION AND FUSION OF QUERY-BY-EXAMPLE SYSTEMS - BUT SWS 2013, BUT Speech @ FIT , Brno University of Technology , Czech Republic,” pp. 7899–7903, 2014.

[17] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, “GTTS systems for the SWS task at MediaEval 2013,” *CEUR Workshop Proc.*, vol. 1043, no. 4, pp. 2–3, 2013.

[18] P. Lopez-Otero, L. Docio-Fernandez, and C. Garcia-Mateo, “GTM-UVigo systems for Albayzin 2016 Search on Speech Evaluation,” no. November, pp. 306–314, 2016.

[19] I. Szöke, M. Skácel, and L. Burget, “But quesst 2014 system description,” *CEUR Workshop Proc.*, vol. 1263, 2014.

[20] G. G. Florian Metze, Xavier Anguera, Etienne Barnard, Marelie Davel, “The spoken web search task at MediaEval 2012,” pp. 8121–8125, 2013.

[21] Varona, M. Penagarikano, L. J. Rodriguez-Fuentes, and G. Bordel, “On the use of lattices of Time-Synchronous Cross-Decoder Phone Co-occurrences in a SVM-phonotactic language recognition system,” *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, pp. 2901–2904, 2011.

[22] J. Tejedor and D. T. Toledano, “The ALBAYZIN 2016 Search on Speech Evaluation Plan,” pp. 1–11, 2016 (available on: https://iberspeech2016.inesc-id.pt/index.php/albayzin-evaluation/, Accessed: 6 Sept. 2018).

[23] J. Tejedor and D. T. Toledano, “The ALBAYZIN 2018 Search on Speech Evaluation Plan,” pp. 1–13, 2018 (available on: http://iberspeech2018.talp.cat/index.php/albayzin-evaluation-challenges/search-on-speech-evaluation/, Accessed: 6 Sept. 2018).

[24] P. Schwarz, “Phoneme recognition based on long temporal context,” *PhD Thesis, Brno Univ. Technol.*, 2008.

[25] P. Matejka, P. Schwarz, J. Cernocky, P. Chytil, "Phonotactic Language Identification using High Quality Phoneme Recognition", in Proc. Eurospeech2005, Sep, 2005.

[26] I. Szoke, P. Schwarz, L. Burget, M. Fapso, M. Karafiat, J. Cernocky, P. Matejka, "Comparison of Keyword Spotting Approaches for Informal Continuous Speech", in Proc. Eurospeech2005, Sep, 2005.

[27] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Lui, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.4)*, Cambridge, UK, 2006.