UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

SISTEMA MULTIMODAL PARA EL RECONOCIMIENTO DE PERSONAS EN GRABACIONES DE TV

Trabajo de Fin de Grado

Presentado en la facultad de

Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona

Universidad Politécnica de Cataluña

por

Carla Cortillas Liesa

En cumplimiento parcial

de los requerimientos del grado de

INGENIERÍA DE SISTEMAS AUDIOVISUALES

Supervisor: Javier Hernando

Barcelona, Septiembe 2016

Abstract:

The Project described in this document falls within the topic of person recognition in TV recordings by mean of multimodal systems.

It has been developed as collaboration with image and audio processing groups in the signal theory department in UPC. Thus, it is a project about the development of a person recognition system in TV Broadcast videos implemented for the participation in the Whorkshop Mediaeval 2016.

The aim of the competition consist in find the names of the people that appears and talks in each shot of different videos of a given database. This discovery should be done in totally unsupervised manner using only the information in each shot as image, audio or text.

For this purpose, it has been proposed to implement the fusion of three monomodal algorithms. These technologies will process the information present in the text, image and audio in independent manner. Information after monomodal technologies will be fused with objective of create e a multimodal algorithm able to tag the shots in the database.

Finally, this thesis is centered in the monomodal audio algorithm development for which it has been proposed a tracking system based in i-vectors.

Resumen:

El proyecto descrito en este documento se enmarca en el tópico del descubrimiento de personas en vídeos de televisión mediante sistemas multimodales.

Forma parte de una colaboración con los grupos de procesado de audio e imagen del departamento de Teoría de señal. Así pues, se trata del desarrollo de un sistema de descubrimiento de personas en un entorno de televisión broadcast para la participación en un Workshop denominado Mediaeval 2016.

El reto de la competición consiste en encontrar los nombres de las personas que aparecen hablando en cada escena de los diferentes shows de un conjunto de videos. Dicho descubrimiento debe realizarse de manera totalmente no supervisada, utilizando únicamente la información presente en cada escena, como la imagen, el audio o el texto.

Para ello, se propone utilizar la fusión de tres algoritmos monomodales que procesen de manera independiente el texto, la imagen y el audio, para lograr un sistema multimodal capaz de etiquetar cada escena dada una base de datos.

Finalmente, el presente proyecto se centrará en el desarrollo del algoritmo de audio a nivel monomodal, para el cual se ha propuesto la implementación de un sistema de tracking basado en i-vectors.

Revisión y aprobación:

Revision	Date	Purpose
0	15/09/2016	Revisión del documento
1	27/09/2016	Aprobación del documento

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Carla Cortillas Liesa	c.carla.cl@gmail.com
Javier Hernando	javier.hernando@upc.edu
Miquel India	miquelindia90@gmail.com

Written by:		Reviewed and approved by:	
Date	14/09/2016	Date	27/09/2016
Name	Carla Cortillas	Name	Javier Hernando
Position	Autor del proyecto	Position	Supervisor del proyecto

Tabla de contenido:

Abstract: 1
Resumen: 2
Revisión y aprobación: 3
Tabla de contenido: 4
Lista de Figuras 6
Lista de Tablas: 7
1. Introduction 8
1.1. Presentación y motivación: 8
1.2. Definición y objetivos: 8
1.3. Sistemas multimodales para la detección de personas 9
1.3.1. Tecnologías monomodales 9
1.3.2. Algoritmos de fusión: 10
2. Estado del arte 12
2.1. Sistemas de reconocimiento del locutor: 12
2.2. Seguimiento del locutor: 12
2.2.1. Detector de actividad. 13
2.2.2. Extracción de características. 13
2.2.3. Entrenamiento y clasificación: 15
3. Metodología: 18
3.1. Desarrollo del algoritmo monomodal de audio. 18
3.1.1. Segmentación: 20
3.1.2. Extracción de i-vectors: 21
3.1.3. Scoring: 23
3.1.4. Decisión final: 25
3.1.5. Evaluación del algoritmo monomodal de audio 25
3.2. Sistema completo: 26
3.2.1. ILP: 27
3.2.2. Asignación Texto – Voz: 29
3.2.3. Evaluación del sistema completo: 30
4. Resultados 32
4.1. Resultados del sistema de seguimiento de audio: 32
4.1.1. Base de datos de Desarrollo: 32
4.1.2. Base de datos de test. 35

4.2. Resultados sistema completo: 36
5. Conclusiones: 37
Bibliografía: 38

Lista de Figuras

Figura 1. Diagrama de bloques del sistema completo……9
Figura 2. Ejemplo de salida de un sistema de detección del hablante……12
Figura 3 .Diagrama de bloques de un sistema de seguimiento del hablante……13
Figura 4. Diagrama de bloques del proceso de extracción de los MFCC……..14
Figura 5. Proceso de segmentación para la extracción de los MFCC……14
Figura 6. Banco de filtros Mel……15
Figura 7. Diagrama redes neuronales……...16
Figura 8. Diagrama de bloques del sistema de seguimiento propuesto……19
Figura 9. Ejemplo asignación de targets……..24
Figura 10. Ejemplo fichero pre-scoring……24
Figura 11. Ejemplo de fichero de salida del proceso de scoring……..24
Figura 12. Ejemplo de la tarea de detección multimodal de personas……...26
Figura 13. Representación temporal de la salida de los algoritmos monomodales…....27
Figura 14. Diagrama de bloques algoritmo multimodal Mediaeval 2015……28
Figura 15. Ejemplo fusión Imagen- Voz……....28
Figura 16. Ejemplo matriz de pesos……..29
Figura 17. Sistema de asignación directa Texto-Voz……..29
Figura 18. Curvas DET base de datos de desarrollo……34
Figura 19. Curvas Precisión-Recall base de datos de desarrollo……34

Lista de Tablas:

Tabla 1: Resultados del sistema de seguimiento según el primer experimento……34
Tabla 2: Resultados del sistema de seguimiento según el segundo experimento……34
Tabla 3: Resultados del sistema de seguimiento según el segundo experimento……34
Tabla 4: Tabla resumen de resultados……..34
Tabla 5: Comparación sistemas de segmentación……..36
Tabla 6: Comparación experimentos para la competición……37
Tabla 7: Comparación algoritmos de fusión……37

1. Introduction

1.1. Presentación y motivación:

El trabajo descrito en este documento ha sido desarrollado en el departamento de Teoría de Señal de la UPC, en colaboración con los grupos de procesado de imagen y más concretamente de audio. Así pues, el proyecto surge con la participación de dichos grupos en el workshop Mediaeval 2016.

Se trata de una competición entre diferentes grupos de investigación a nivel internacional que tiene como temática el descubrimiento de personas en vídeos de televisión broadcast.

En los últimos años, hemos sido participes del interés de la población en buscar información sobre otras personas. Sin embargo, podemos observar que la mayoría de la información que podemos encontrar proviene de fuentes de texto, ya que la falta de tecnología destinada a indexar bases de datos de vídeos hace de ello una tarea difícil.

Además, con el notable incremento de archivos multimedia en los últimos tiempos aumenta la necesidad de disponer de herramientas capaces de determinar la identidad y localización de las diferentes personas que aparecen durante los vídeos, así como de aplicaciones que nos permitan buscar dentro de esas bases de datos.

1.2. Definición y objetivos:

La participación en la competición requiere del desarrollo de un sistema capaz de descubrir a las personas que aparecen y hablan al mismo tiempo en las diferentes escenas dada una base de datos de videos.

De esta manera, el sistema a desarrollar deberá ser capaz de etiquetar las diferentes escenas con el nombre de la persona que aparece hablando en cada una de ellas. El reto está en que dicho etiquetado deberá obtenerse de manera totalmente no supervisada, teniendo en cuenta únicamente la información que puede extraerse de cada escena (imágenes, audio, texto…).

Para ello, se propone utilizar un algoritmo multimodal que fusione tres tecnologías monomodales que procesen por separado la información procedente del audio, video y texto.

Así pues, cada una de las tres tecnologías monomodales deberá procesar la información correspondiente presente en cada escena para identificar quien habla (audio) y aparece (imagen) en cada escena así como extraer el nombre de la persona en cuestión (texto). El algoritmo multimodal deberá etiquetar las escenas a partir de la información procedente de las fuentes monomodales. Finalmente, el fichero de etiquetas se entregará a la organización de la competición para ser evaluado con las métricas de evaluación del concurso.

Así pues, será necesario analizar y conocer dichas métricas para adecuar el sistema al modelo de evaluación y optimizar los resultados, como veremos en apartados posteriores.

La participación en este proyecto se trata de la continuación de un sistema anterior, utilizado para la participación en el Mediaeval 2015, para el cual se propone mejorar los resultados introduciendo diferentes mejoras a nivel monomodal i/o multimodal. Dado que

la colaboración se desarrollara en el departamento de procesado de audio, el proyecto se centrará en desarrollar un nuevo algoritmo monomodal de audio utilizando técnicas de reconocimiento del locutor.

Para ello, deberemos conocer el sistema desarrollado para la competición el año pasado así como analizar las diferentes alternativas de mejora para decidir finalmente cual se adecua más a nuestro trabajo, tal como se explica brevemente en los siguientes apartados.

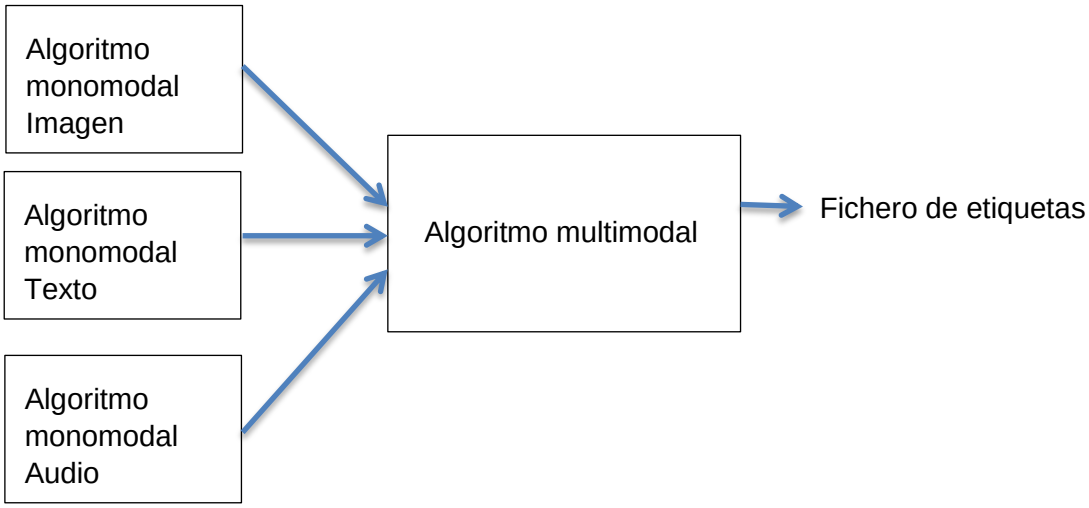

Figura 1: Diagrama de bloques del sistema completo

1.3. Sistemas multimodales para la detección de personas.

Los sistemas multimodales para el descubrimiento de personas tienen como objetivo indexar bases de datos de videos de TV con el nombre de las personas que en ellos aparecen a partir de la combinación de un conjunto de tecnologías monomodales. Al igual que en cualquier sistema de reconocimiento de personas, el primer paso será extraer la mayor cantidad de información posible procedente del vídeo, en este caso, dicha extracción se realiza por medio de las técnicas monomodales, entendidas como herramientas capaces de extraer información específica de un determinado tipo a partir de un fichero dado. Dicha información será procesada o bien de manera conjunta (Ej: Tracking multimodal) para obtener un etiquetado a nivel multimodal o bien individualmente para obtener diferentes identificaciones a nivel monomodal que serán procesadas mediante un algoritmo de fusión para obtener el etiquetado final. Como se ha mencionado en apartados anteriores, en nuestro caso, las tecnologías monomodales a utilizar se basarán en la extracción de la información procedente de las imágenes, del audio y del texto de los vídeos. A continuación, se explicará brevemente el objetivo de estas técnicas monomodales y los diferentes algoritmos que se pueden utilizar así como diferentes algoritmos de fusión.

1.3.1. Tecnologías monomodales

Imagen:

El algoritmo monomodal basado en la información de imagen, tiene como objetivo identificar quién aparece en cada escena. Además, dado que el reto de la competición consiste en identificar la persona que aparece a la vez que habla en cada escena,

podrían incorporarse técnicas para identificar si la persona que aparece está hablando basándose en el movimiento de los labios. En este caso, podrían utilizarse, entre otros, los siguientes algoritmos:

- Sistemas de detección de caras: Definen el área rectangular donde están posicionadas las caras.
- Tracking de caras: Consiste en llevar a cabo el seguimiento de las caras detectadas con los sistemas de detección anteriores a lo largo del vídeo.
- Diarización de caras: Define una segmentación homogénea con su correspondiente clustering de caras.

El sistema de detección desarrollado el año pasado para el Mediaeval 2015 incorporaba la diarización de caras como tecnología monomodal de imagen. Para el nuevo sistema a desarrollar, el grupo de procesado de imagen ha propuesto sustituirla por un sistema basado en tracking de caras.

Texto: El algoritmo de texto tiene como objetivo procesar la información procedente de subtítulos, rótulos, etc… para detectar los nombres que aparecen en ellos. Para ello, existen diferentes técnicas tales como las que se citan a continuación:

- Optical Character Recognition (OCR): Se utiliza para convertir el texto procedente de la imagen en un formato que pueda ser reconocido y manipulado por un ordenador (Ej:ASCII).
- Named Entity Detection (NED): Trata de clasificar las palabras del texto en categorías predefinidas, como nombres o ubicaciones.
- Automatic Speech Recognition (ASR): Se utiliza para obtener transcripciones del lenguaje hablado a texto legible. Puede ser útil cuando se trabaja con bases de datos de videos en los que no se dispone de información en formato texto (rótulos, subtítulos…) ya que permite extraer información acerca de quién va a aparecer o hablar en la escena (Por ejemplo si un presentador da paso a la persona que hablará a continuación).

Dada la dificultad que supone la incorporación de un ASR a este tipo de sistemas, el algoritmo monomodal de texto de nuestro sistema de reconocimiento consiste en una combinación de OCR+NED.

Audio: Existen multitud de técnicas utilizadas para el reconocimiento del hablante en ficheros de audio. Para ello, la voz debe ser modelada pudiéndose utilizar técnicas convencionales cómo las basadas en GMMs o más innovadoras cómo técnicas basadas en modelado de la voz utilizando i-vectors. Para clasificar los modelos de cada segmento, pueden utilizarse técnicas como la Diarización, técnica utilizada en el sistema del año pasado, o el tracking, técnica propuesta como nueva tecnología monomodal de audio en nuestro sistema de reconocimiento de personas. Dado que el trabajo a desarrollar en este proyecto está directamente relacionado con dicha tecnología monomodal de audio, dichas técnicas serán explicadas en detalle en los siguientes apartados.

1.3.2. Algoritmos de fusión:

Una vez obtenidas las indexaciones del vídeo en términos de audio, imagen y texto procedentes de las técnicas monomodales explicadas anteriormente, es necesario establecer ciertas restricciones para combinar las salidas de dichas fuentes y obtener la correcta segmentación con sus correspondientes etiquetas nominales. Acorde con el reto de la competición, las restricciones del algoritmo de fusión deberán tener en cuenta

cuestiones como si la persona que aparece en pantalla está hablando o en caso de que, según la indexación de caras, aparezca más de una cara en pantalla asignar el nombre, procedente de la indexación de texto, a la cara correcta o asociar la cara correcta a la voz detectada…

Para ello, para el sistema anterior utilizado para el Mediaeval 2015 se propuso utilizar un sistema ILP, cuyas restricciones se establecen según los solapes de los ficheros de salida de las fuentes monomodales, tal y cómo se explica en apartados posteriores.

Otras opciones podrían ser la implementación de un tracker multimodal, que funcionaría en base a vectores de características conjuntas de audio e imagen o sistemas de asignación directa en los que se asume por ejemplo que en las escenas en las que se detecta una cara y un nombre el nombre pertenece a esa cara y por tanto se lleva a cabo la asignación cara/texto.

Dado que muchos de los sistemas actuales para el reconocimiento de personas toman ventaja únicamente de sistemas de diarización del hablante junto a sistemas OCR+NED, en este proyecto se ha realizado la prueba de utilizar la asignación directa entre texto y voz, con objetivo de proponer alternativas de mejora al sistema ILP, asumiendo que cuando se detecta a la vez un nombre y un hablante el nombre que aparece en pantalla corresponde a la persona que está hablando y por tanto se realiza la asignación (speak

er/texto). Podrán compararse los resultados en el apartado correspondiente.

En los siguientes apartados, se pasará a explicar en detalle la tecnología utilizada para desarrollar el tracking de audio, el método propuesto, las métricas de evaluación y los resultados obtenidos.

2. Estado del arte

2.1. Sistemas de reconocimiento del locutor:

Podemos entender los sistemas de reconocimiento del hablante como una modalidad biométrica que parte de la voz de una persona para identificarla o confirmar su identidad.

Durante más de seis décadas, los científicos han estudiado la habilidad del oído humano para reconocer y discriminar voces [1]. Con la llegada de los ordenadores digitales, la investigación en el reconocimiento del hablante dio un giro hacia el desarrollo de técnicas automáticas de reconocimiento del hablante que dio lugar al descubrimiento de sistemas simples capaces de superar al oído humano [2]. En las últimas décadas, los investigadores han desarrollado algoritmos más sofisticados que han incrementado su rendimiento a la hora de ser evaluados en entornos más realistas, y actualmente estos sistemas son utilizados en numerosas aplicaciones de telecomunicaciones tales como la identificación por voz o la indexación de bases de datos de vídeos, tal y como se estudia en el presente proyecto.

Nuestro sistema de reconocimiento tiene como objetivo indexar un fichero de audio determinando quien habla en cada momento tal y como se muestra en la siguiente figura.

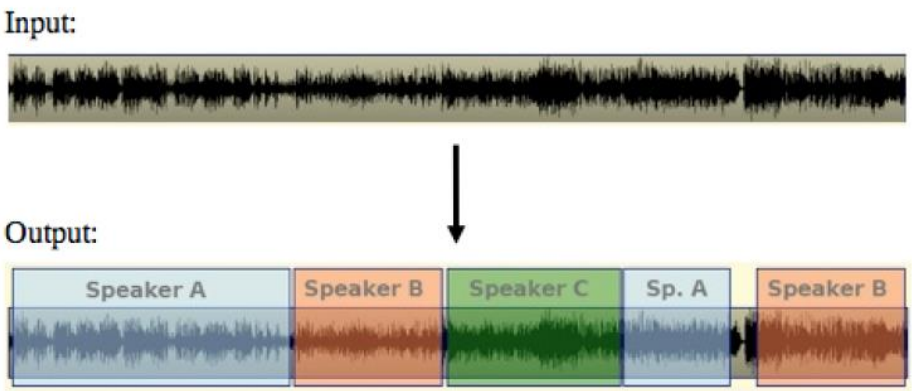

Figura 2. Ejemplo de salida de un sistema de detección del hablante.

Para ello, utilizaremos un sistema de reconocimiento basado en técnicas de seguimiento, tal y como se explica en el siguiente apartado.

2.2. Seguimiento del locutor:

En esta sección presentaremos el diagrama de bloques típico de un sistema de reconocimiento del locutor basado en técnicas de tracking. Posteriormente pasaremos a definir con más detalle cada uno de los bloques.

Así pues, la figura 3 muestra un diagrama de bloques de un sistema de seguimiento de locutor:

En un proceso de tracking de audio para el reconocimiento del locutor, partimos por un lado de los targets, aquellos locutores que debemos identificar a lo largo de la grabación, y por otro lado de los segmentos de análisis. El objetivo es encontrar los segmentos del fichero de entrada en los que aparece cada target.

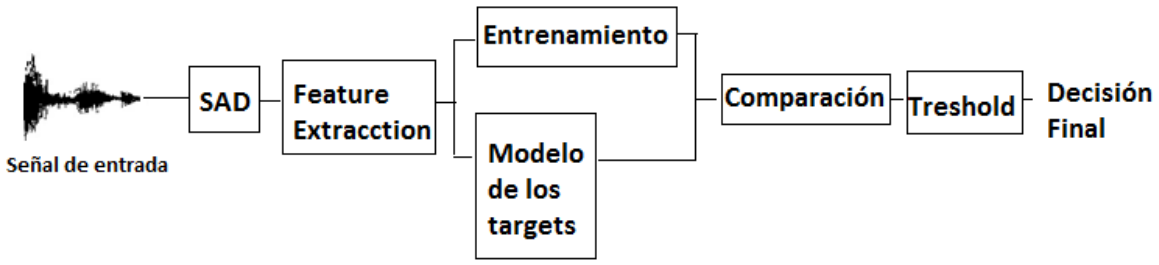

Figura 3 .Diagrama de bloques de un sistema de seguimiento del hablante

Para ello, la señal de voz se divide en diferentes segmentos en cada uno de los cuales se deberá determinar si uno de los targets está presente. Para ello, la señal de entrada se procesa de la manera que se describe a continuación y una vez obtenidos los modelos que parametrizan la voz tanto de los targets como del resto de segmentos, se deberá establecer una técnica de comparación de modelos, conocida como técnica de scoring, para determinar según el resultado de dicha técnica si en el segmento que se analiza se ha detectado algún target.

A continuación, pasaremos a definir los diferentes bloques del diagrama anterior.

2.2.1. Detector de actividad.

Cómo podemos observar, antes de comenzar con el proceso de tracking, nos encontramos un paso previo de procesado de la señal de entrada. Se suele incorporar un Sistema de Detección de Actividad (SAD) o Sistema de Detección de Voz (VAD) para descartar aquellos segmentos de silencio del fichero de audio. Se trata de sistemas destinados a la detección de presencia o ausencia del habla humana. Existen varias técnicas a utilizar dependiendo de la naturaleza de la señal. Si asumimos que los segmentos de no habla contienen silencio o ruido y el SNR de la señal es alto podemos utilizar SADs *basados en la energía* de la señal, obteniendo resultados eficientes con un bajo coste computacional. Estos sistemas comparan la energía a corto plazo con un threshold predefinido para decidir si el segmento es habla, si está por encima del umbral, o no, si está por debajo. Suelen incorporar técnicas de preprocesado para aumentar previamente el SNR de la señal.

Si el SNR de la señal es bajo, es preferible utilizar SADs *basados en modelos* ya que a pesar de tener un mayor coste computacional ofrecen mayores prestaciones ante señales de peor calidad, aunque su rendimiento dependerá del entrenamiento de los modelos.

Ambos métodos pueden ser combinados formando un sistema hibrido [3].

2.2.2. Extracción de características.

Una vez descartada la información innecesaria para el desarrollo del tracker, pasaremos a parametrizar la voz. Para ello, nos interesará extraer características que minimicen la variabilidad inter-locutor (variabilidad que se produce en un mismo hablante a causa del estado de ánimo, la entonación, el ambiente…) y maximicen la variabilidad entre-locutor (entre diferentes hablantes) y que sean robustas al ruido.

En el caso de los sistemas de reconocimiento del hablante se extraen las características espectrales de la voz ya que se relacionan con el tracto vocal, que es característico de cada persona. Las más utilizadas son los coeficientes MFCC (Mel-Frequency-Cesptrum-Coefficients), que serán definidas a continuación. Sin embargo existen otras variantes tales como los Linear Frequency Cepstral Coefficients (LFCC), Perceptual Linear Predictive (PLP), Linear Predictive Coding (LPC) entre otros.

Mel-Frequency-Cesptrum-Coefficients (MFCC)

Se definen como un conjunto de coeficientes que representan el espectro de la señal de voz con la peculiaridad de que tratan de aproximar la resolución espectral a la respuesta del oído humano, es decir, mayor facilidad para distinguir tonos a bajas frecuencias que a altas. Para ello, se llevan a cabo las siguientes etapas:

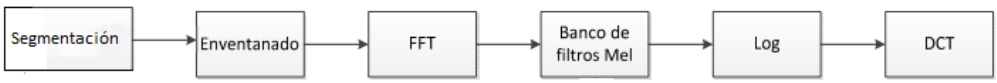

Figura 4. Diagrama de bloques del proceso de extracción de los MFCC.

Segmentación: La señal de entrada se divide en tramas de entre 20 y 30 milisegundos, que serán procesadas de manera individual, para evitar la variabilidad de la voz y conseguir tramas en las que pueda considerarse estacionaria, con un solapamiento del 50% (10-15 segundos) con objetivo de no perder la información de la transición entre tramas, tal y como se muestra en la siguiente figura.

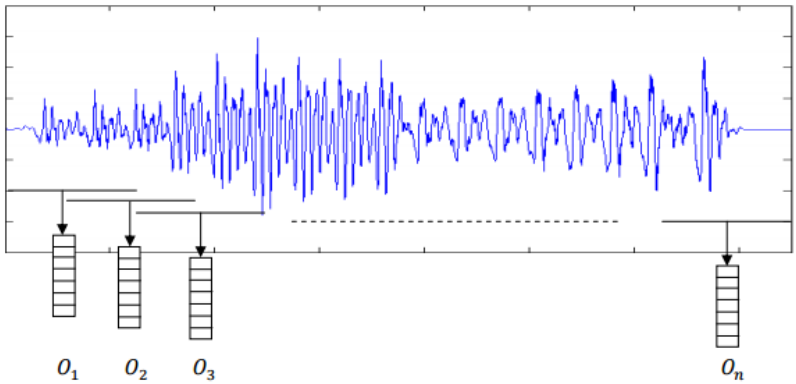

Figura 5. Proceso de segmentación para la extracción de los MFCC

Enventanado: Una vez segmentada la señal, las tramas serán enventanadas para obtener posteriormente un correcto análisis frecuencial. La ventana más utilizada es la de Hamming.

FFT: Como se ha comentado, posteriormente al enventanado se procede al análisis frecuencial de la señal. Cómo es habitual, dicho análisis se lleva a cabo mediante el cálculo de su DFT (Discret Fourier Transform) o bien de su versión rápida FFT (Fast Fourier Transform). Normalmente, sólo se utiliza la amplitud del espectro obtenido.

Banco de filtros Mel: Tal y como ha sido mencionado, la peculiaridad de los MFCC reside en su adaptación a la respuesta frecuencial del oído humano. Para ello, la FFT se somete a una transformación no lineal mediante un banco de filtros denominados filtros Mel (Figura 7).

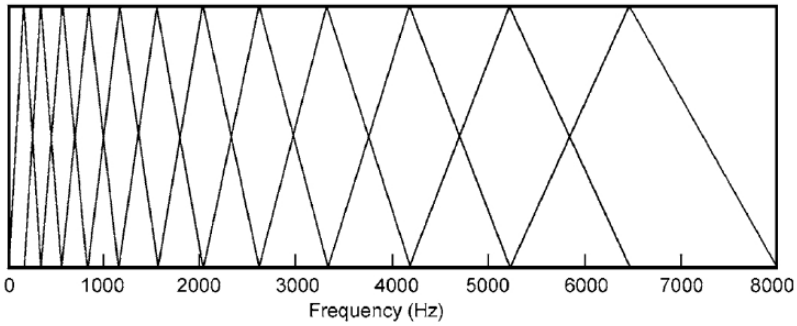

Figura 6. Banco de filtros Mel

El resultado en la nueva escala representa un comportamiento lineal en el rango de frecuencias de 0 Hz a 1kHz y logarítmico en frecuencias superiores.

Log y DCT: Finalmente, a la salida se los filtros se le aplica el logaritmo natural seguido de la Transformada Discreta Coseno (DCT) con objetivo de compactar la energía en el menor número de coeficientes posible. De esta manera, el vector de características es generado con los primeros 12-20 coeficientes.

Además de los coeficientes definidos anteriormente, algunos sistemas incorporan al vector de características coeficientes adicionales conocidos como delta-cepstrum, que tratan de analizar la información transicional entre los segmentos mediante el cálculo de la primera y segunda derivada temporal de los coeficientes cepstrales.

Para acabar, también pueden aplicarse técnicas de normalización de las características antes de pasar a la etapa de training. Estas técnicas tienen como objetivo compensar el efecto de variabilidad indeseada que produce el canal sobre la señal de voz. Para ello, se utilizan técnicas tales como la normalización por media cepstral (CMN) o el filtrado RASTA (RelAtiveSpecTrAl).

2.2.3. Entrenamiento y clasificación:

El objetivo de la etapa de entrenamiento, consiste en crear un modelo del locutor a partir del vector de características procedente de la etapa anterior. Para ello, la técnica de modelado mediante GMMs (Gaussian Mixture Model) ha sido durante muchos años la metodología de referencia en los sistemas de reconocimiento del hablante. Se trata de un modelo probabilístico que asume que los datos son generados por una mezcla de un número finito de distribuciones gausianas de parámetros desconocidos, cuya estimación se lleva a cabo mediante el algoritmo de Expectation-Maximization (EM). Suele utilizarse junto a lo que se conoce como Universal Background Model (UBM), un modelo que representa la distribución independiente de locutor de todos los vectores de características, es decir, modela las características comunes a todos los locutores. El entrenamiento se realiza a partir de una gran cantidad de audio procedente de un gran número de locutores y de diversas condiciones acústicas. En estos sistemas, cuando un locutor es detectado, los parámetros del UBM se adaptan a la distribución de características del mismo, de forma que el modelo universal o UBM adaptado se convierte en modelo del locutor.

Sin embargo, actualmente, los sistemas basados en lo que se conoce como i-vectors, se han convertido en el estado del arte en los sistemas de reconocimiento del locutor. Es por esto, que ha sido la técnica utilizada para desarrollar nuestro sistema de tracking.

Dicha técnica consiste en extraer modelos de locutor denominados vectores de identidad o i-vectors que serán comparados mediante técnicas de scoring con objetivo de determinar, en la etapa de decisión final, si ambos pertenecen al mismo hablante. Las bases de dicha técnica serán explicadas en el siguiente apartado.

Otras técnicas utilizadas actualmente en sistemas de reconocimiento del locutor serán explicadas a continuación:

Cuantización Vectorial:

Consiste en la representación de las características extraídas en el apartado anteriorior como un espacio vectorial, el cual cuenta con un conjunto infinito de patrones posibles (espacio de características). En este espacio se pretende asignar un conjunto de patrones desconocidos (test) a un conjunto finito de patrones de referencia; de manera que al vector a reconocer se le asigna un vector patrón cuya distancia a él sea mínima.

El espacio representativo quedará dividido en zonas o regiones, donde al vector representativo de esa región se denominará “codeword” (centroide), de forma que los vectores que caigan en dicha región se asignarán a dicho centroide. El conjunto de todos los centroides se denomina “codebook” (muestrario). Los vectores de test serán asignados al centroide más cercano, mientras que cada una de las regiones podrían corresponderse con cada uno de los locutores a detectar.

Redes neuronales:

El vector de coeficientes cepstrales se multiplica por los respectivos pesos según la siguiente figura, y se suman los resultados. Finalmente, se evalúa en la función de activación, del tipo sigmoidal, escalón u otra. En este caso, la salida puede ser 1 o -1, simbolizando la pertenencia o no a alguna clase.

El entrenamiento de esta clasificador responde a la existencia de una serie de ejemplares de entrenamiento, los cuales pasan sucesivamente por esta red neuronal, en este caso forzando uno de los dos valores en la salida, lo que trae consigo que el error sea asumido por la modificación de los valores de los pesos. En el test, el vector de características de la voz desconocida es sometido al paso de esta red ya entrenada (con todos los pesos determinados) y su salida se asociará con alguna de las clases, que se corresponderán con cada uno de los locutores a detectar. [4]

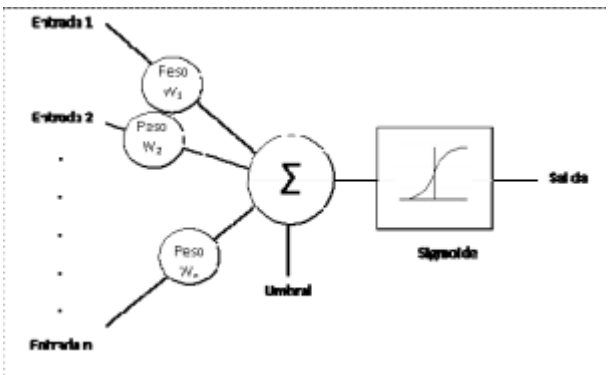

Figura 7: Diagrama redes neuronales

Máquinas de soporte vectorial:

El objetivo de esta técnica es transformar los vectores de entrada, en este caso los vectores de características, en vectores de mayor dimensión con los que el problema pueda solucionarse linealmente. Esta frontera de decisión lineal corresponde a un hiper plano separador de dos clase. Para ello, se elige el hiper plano que mejor separa ambas clases. La etapa de test verifica a qué lado de la línea de separación están los vectores de características de la voz desconocida. A un lado están los vectores que caracterizan al individuo y al otro el resto de la población. [5]

3. Metodología:

Tal y como ha sido mencionado, el presente proyecto abarca el desarrollo de un tracker de audio que será posteriormente integrado, como algoritmo monomodal de audio, en un sistema multimodal para el reconocimiento de personas. Así pues, los siguientes apartados se centrarán en explicar el desarrollo e implementación de dicho algoritmo, así como las técnicas utilizadas para su evaluación.

Además, se explicará el desarrollo de los distintos algoritmos de fusión probados, ILP, asignación directa Voz-Texto, y un nuevo sistema basado en la métrica de evaluación del Mediaeval. Finalmente, se explicarán las técnicas de evaluación utilizadas para determinar el rendimiento del sistema completo.

3.1. Desarrollo del algoritmo monomodal de audio.

Según lo expuesto en los apartados anteriores, para el nuevo sistema multimodal de reconocimiento de personas a desarrollar, el departamento de audio ha decidido integrar un nuevo algoritmo monomodal basado en técnicas de tracking como alternativa al sistema de diarización anterior.

Para ello, se ha propuesto desarrollar un sistema de tracking basado en i-vectors siguiendo las bases teóricas expuestas a continuación.

Investigaciones recientes en el ámbito de reconocimiento del locutor se han centrado en la utilización de dicha tecnología, motivadas por las técnicas JFA (Join Factor Analysis). El principal objetivo del JFA consiste en encontrar dos subespacios que representen por separado la variabilidad del hablante y del canal. Sin embargo, diferentes experimentos dejaron ver que la técnica sólo es capaz de separar parcialmente ambas variabilidades, dado que se descubrió que el espacio del canal contiene información útil para distinguir entre hablantes. Es por esto que posteriormente se propuso utilizar un único espacio capaz de modelar conjuntamente la variabilidad del hablante y del canal, conocido como espacio de variabilidad total.

De esta manera, el principal fundamento de esta nueva técnica, parte de lo que se conoce como super-vector GMM. Un supervector representa de forma compacta la información de locutor presente en un GMM a partir de la concatenación de los vectores de medias, de dimensión 1xd, de las K Gaussianas de un GMM, obteniéndose así un vector de dimensión 1xKd. Puede entenderse como una técnica de compresión que reduce la dimensionalidad del super vector asumiendo que un super-vector GMM dependiente del hablante y del canal puede ser representado de la siguiente manera:

$$\mathbf{s = m + Tw}$$

Donde **m** representa un super-vector independiente del canal y del hablante, por ejemplo, un supervector UBM. **T** es una matriz de bajo rango denominada matriz de variabilidad total, que representa una base del espacio de variabilidad total reducido. Finalmente, **w** es lo que se conoce como identity vector o i-vector, cuya dimensión es mucho menor a la del super-vector GMM reteniendo la información relevante. Así, el i-vector de un hablante, representa las coordenadas de dicho locutor en el espacio de variabilidad total.

Como se ha mencionado, el objetivo del tracker consiste en detectar un conjunto de targets a lo largo de un fichero de audio. Para ello, el primer paso será extraer un modelo

de cada uno de los targets, esto es, el i-vector de cada target. Seguidamente, segmentaremos dicho fichero para extraer luego el i-vector de cada segmento. Finalmente, compararemos el i-vector de cada uno de los segmentos con el i-vector de cada uno de los targets, mediante la técnica que se explica a continuación, para determinar si en el segmento de análisis aparece uno de los targets.

Así pues, a continuación se explicarán la técnica de scoring utilizada en nuestro sistema de seguimiento para comparar i-vectors, el score similitud coseno. Se trata de un método sencillo a la par que eficaz para determinar la similitud entre dos i-vectors. Se define como:

$$Score(iv1, iv2) = \frac{(iv1)^t \cdot iv2}{\|iv1\| \cdot \|iv2\|}$$

Se evalúa el valor del coseno del ángulo comprendido entre ambos vectores, de manera que si ambos apuntan a un mismo lugar, esto es, máxima similitud, el valor del score será 1. Del mismo modo, el valor de la puntuación decrecerá a menor similitud de i-vectors.

Dicho esto, nuevo sistema basará su funcionamiento en la comparación de modelos extraídos de diferentes segmentos de voz. En nuestro caso, dichos modelos serán los i-vectors extraídos de cada segmento.

A continuación, se presentará el funcionamiento de dicho algoritmo así como la implementación y desarrollo de cada una de las etapas.

Dicho algoritmo tiene como objetivo detectar una serie de hablantes, a los que denominaremos Targets, a lo largo de un fichero de audio. Para ello, el primer paso será extraer el i-vector de cada uno de los Targets. Seguidamente, segmentaremos el fichero completo con objetivo de extraer el i-vector de cada segmento. Una vez obtenidos los i-vectors de los targets así como los i-vectors del fichero completo (uno cada segmento) dichos vectores de características serán comparados utilizando la distancia coseno con el fin de obtener un score que determine la similitud entre ambos segmentos a partir del cual decidir si en ambos segmentos habla el mismo locutor. O lo que es lo mismo, comprobar en cada segmento si aparece alguno de los targets, tal y como se muestra en la siguiente figura.

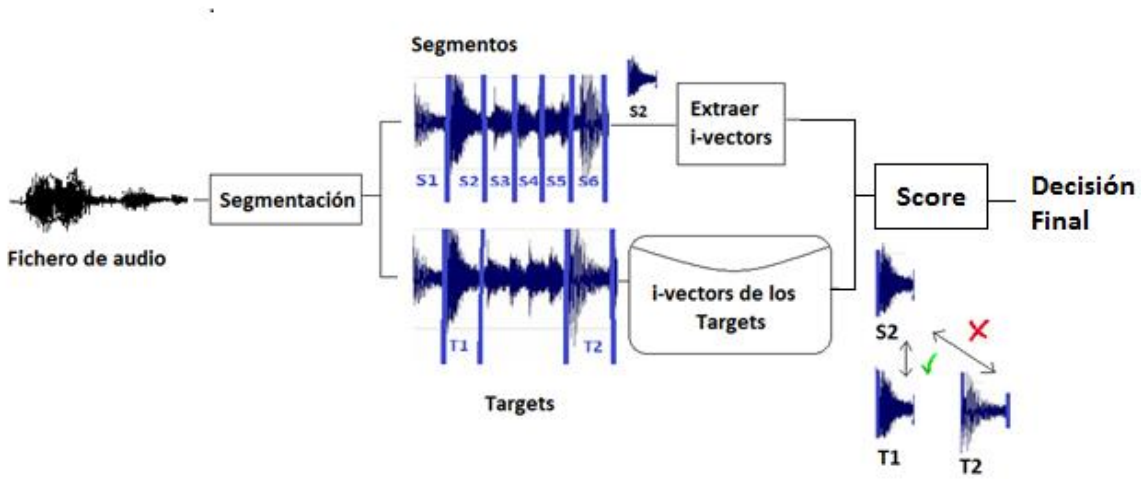

Figura 8. Diagrama de bloques del sistema de seguimiento propuesto

A continuación, se pasará a detallar la implementación y desarrollo de las diferentes etapas del algoritmo.

3.1.1. Segmentación:

Podemos dividir la etapa de segmentación por un lado en la segmentación del fichero completo y por otro lado en la obtención de los segmentos denominados como Targets, es decir los segmentos que pertenecen a los locutores a detectar durante la señal de audio.

Segmentación de los locutores a detectar.

Para obtener un modelo de cada uno de los locutores que buscamos, es decir, los targets, necesitaremos obtener un segmento de audio en el que aparezca cada uno de ellos. Para ello, se han realizado diferentes pruebas utilizando diferentes métodos para obtener un segmento adecuado para entrenar cada target. Por un lado, podemos seleccionar los segmentos más adecuados por medio de un análisis del fichero de audio, es decir, escuchando el audio y eligiendo aquellos segmentos en los que podemos escuchar la voz de cada locutor con mayor claridad. Se trata de un método fiable ya que nos permite obtener un modelo adecuado para cada target. Sin embargo, únicamente podremos aplicar este procedimiento en periodos de prueba ya que procesar un elevado número de shows supondría un elevado trabajo de análisis. Como alternativa, se pueden obtener los segmentos pertenecientes a cada target en base a un fichero .rttm, en el que aparecen los diferentes locutores que aparecen hablando a lo largo del video junto con el intervalo de tiempo correspondiente. Se trata de un método menos fiable ya que puede ser susceptible de elegir segmentos en los que la voz del locutor no pueda apreciarse con claridad, de manera que obtendremos un modelo menos aproximado. En nuestro caso, este método ha sido utilizado en periodos de prueba. Respecto al método a utilizar en el caso real, para la obtención de los targets en los videos de la base de datos del Mediaeval 2016, ninguno de los métodos anteriores pueden ser utilizados, dado que no disponemos de los ficheros .rttm ni ninguna otra información previa. Por lo tanto, obtendremos la segmentación a partir del fichero de salida de la tecnología monomodal de texto en el que se indican los intervalos de tiempo en los que aparece cada nombre. Supondremos que en el momento que aparece un nombre en pantalla habla dicho locutor,de manera que elegiremos para cada target, el segmento de video en el que aparece su nombre. El resultado será un segmento de audio .wav partir del cual obtener el i-vector del target. Se trata de un método menos fiable que los anteriores, ya que arrasta los errores de la técnica monomodal de texto y puede dar lugar a segmentos de video demasiado cortos para entrenar los targets.

Como puede verse en el apartado de resultados, el rendimiento del tracker depende en gran medida del entrenamiento de los targets, concretamente de la duración y fiabilidad de los segmentos escogidos para obtener los modelos.

Segmentación de la señal completa.

Tal y como se ha explicado en el estado del arte, como paso previo en cualquier sistema de tracking, es conveniente utilizar un detector de voz, SAD o VAD. En nuestro caso, antes de proceder a la segmentación del audio, procesaremos el fichero mediante un SAD implementado en el departamento de audio con objetivo de descartar los segmentos de silencio. Se trata de un SAD basado en la energía de la señal, cuyo funcionamiento ha sido explicado en un apartado previo, implementado en un script de

Matlab con el que dado un fichero de audio obtenemos un fichero .uem que muestra únicamente los intervalos de tiempo en los que detecta voz.

Una vez descartados los segmentos de silencio, procederemos a la segmentación de la voz. Para ello, se realizaran segmentos de igual duración. También en este apartado se han realizado diferentes pruebas experimentales utilizando diferentes duraciones de segmentos, con y sin solape entre tramas. En nuestro caso, se ha observado que el caso óptimo se obtiene utilizando segmentos de 3 segundos, tal y como se puede observar en el apartado de resultados. Finalmente, obtenemos un conjunto de segmentos con igual formato que el formato de entrada (.wav).

Para llevar a cabo la segmentación del audio, se han implementado diferentes funciones en Python.

3.1.2. Extracción de i-vectors:

Una vez obtenidos los diferentes segmentos, deberá obtenerse el i-vector que modele cada uno de ellos, para ser posteriormente comparados.

Para ello, deberán extraerse las características frecuenciales de los segmentos, en nuestro caso, hemos utilizado los coeficientes MFCC, explicados anteriormente, que serán utilizados para el entrenamiento de la matriz de variabilidad total y del super-vector independiente del canal y del hablante, en nuestro caso un UBM entrenado a partir de diferentes segmentos de audio en los que aparecen diferentes locutores en diferentes situaciones y ambientes. Finalmente a partir dichas matrices se extraerán los i-vectors de cada vector.

Así pues, el desarrollo de nuestro sistema se ha llevado a cabo en base a un software denominado ALIZE que incorpora diferentes herramientas para la obtención de modelos del locutor, entre ellos los i-vectors. Se trata de una plataforma libre implementada en C++ destinada al reconocimiento del hablante que consta por un por lado de la librería ALIZE que incluye las funciones requeridas para trabajar con mezclas de gausianas (GMM) así como funciones de entrada/salida para ficheros de diferentes formatos y por otro lado de un toolkit denominado LIA_RAL separado a su vez en diversos componentes destinados a diferentes finalidades tales como la diarización, normalización de características, extracción de modelos…

A continuación, se explicará la implementación de las diferentes etapas necesarias para obtener el i-vector de los segmentos.

Extracción MFCC:

Para obtener los coeficientes Mel Cepstrum de cada segmento utilizaremos un software destinado al reconocimiento de la voz denominado HTK (Hidden Markov Model Toolkit). Se trata de un conjunto de módulos y herramientas disponibles en código C que proporcionan facilidades para el análisis de la voz. Entre ellos, la herramienta HCopy, que nos servirá para obtener un fichero de características a partir de un fichero de entrada y un fichero de configuración. En nuestro caso, los ficheros de entrada serán los diferentes segmentos en formato .wav y obtendremos un fichero de características en formato .prm que será construido según los parámetros del fichero de configuración.

HCopy –C config fichero de entrada.wav fichero de salida.prm

Así pues, el archivo de configuración nos permitirá informar de la frecuencia de muestreo del fichero de entrada así como elegir tanto la duración de las tramas en las que se

dividirá la señal previamente a la extracción de características, tal y como se ha explicado en apartados anteriores, como el número de coeficientes que queremos obtener. Siguiendo las bases teóricas explicadas anteriormente, elegiremos tramas de 30 milisegundos y vectores de características de 20 coeficientes.

Normalización de características:

Una vez obtenidos los ficheros de características de cada segmento, dichas características serán normalizadas con objetivo de compensar la variabilidad de la voz a causa del canal. Para ello, hemos utilizado la herramienta *NormFeat* de LIA_RAL que convierte los ficheros de características obtenidos en el apartado anterior en ficheros de características normalizadas a partir de los cuales se generan ficheros de etiquetas que indican los intervalos de tiempo de habla. Ficheros que serán utilizados para entrenar la matriz de variabilidad total y el UBM, tal y como puede leerse a continuación.

Generación del espacio de variabilidad total:

Entrenamiento del Universal Backgroud Model (UBM):

Para obtener el UBM necesario para la extracción de los i-vectors, nuestro sistema utiliza la función *TrainWorld* del paquete LIA_RAL. El objetivo de esta etapa reside en crear un modelo que debe representar todo el espacio de posibles alternativas al locutor en cuestión. De esta manera, dicha función auna las voces de diferentes hablantes con objetivo de entrenar un único modelo GMM (Gaussian Mixture Model). Así pues, los parámetros de las gaussianas generadoras del modelo se estimarán mediante un algoritmo de Expectation- Maximization (EM). Se trata de un algoritmo iterativo que tiene como objetivo la estimación de parámetros en modelos probabilísticos, en ente caso un GMM. En cada iteración se llevan a cabo dos etapas diferenciadas, por un lado la etapa de Expectatión, en la que desarrolla una estimación de los parámetros del modelo a partir de los datos de entrada, y por otro lado la etapa Maximization, en la que se busca un nuevo conjunto de parámetros que maximice la función de probabilidad obtenida en la etapa de expectation. Así pues, en cada iteración, los parámetros del modelo son reestimados de forma que aumente la verosimilitud con que este modelo puede generar los datos.

Dicho esto, la función *TrainWorld* trabaja junto con un fichero de configuración en el que deberán indicarse los diferentes parámetros de configuración del algoritmo tales como los datos de entrada con los que entrenar el modelo, el número de gaussianas del GMM, o el número de iteraciones del algoritmo.

Según el funcionamiento de nuestro sistema, por cada fichero de audio de entrada se entrenará un modelo UBM a partir del cual se obtendrán los i-vectors de cada segmento, de manera que los datos utilizados para obtener el modelo serán los diferentes segmentos extraídos de la segmentación del fichero de entrada, dado que de esta manera, obtenemos una serie de segmentos en los que aparecen diferentes hablantes en diferentes ambientes que nos sirven para obtener un modelo independiente del locutor y el canal. En cuanto a la elección del número de gaussianas a utilizar y el número de iteraciones del algoritmo EM, nuestro sistema se basa en los ficheros de configuración obtenidos de la web oficial de ALIZE, con los que se considera que se obtiene un rendimiento aproximado al estado del arte actual. En este caso, nuestro algoritmo genera un GMM de 32 gaussianas en 5 iteraciones.

Finalmente, a partir de esta configuración, la función, *TrainWorld* procesa cada uno de los segmentos de entrada, más concretamente el fichero de etiquetas generado en la etapa anterior para cada segmento, calcula la media y la covarianza global de dichos segmentos a partir de las cuales se inicializa el modelo y posteriormente se entrenan los parámetros en un proceso de 5 iteraciones buscando la máxima probabilidad del modelo. Finalmente se obtiene un fichero de salida *world.gmm* que representa el UBM que será posteriormente utilizado para la extracción de los i-vectors.

Entrenamiento de la matriz de variabilidad total:

El siguiente paso necesario para generar los i-vectors será entrenar la matriz de variabilidad total. Se trata de un proceso similar al entrenamiento del UBM explicado en el apartado anterior. En este caso, utilizaremos el binario *TotalVariability* del paquete LIA_SpkDet del LIA_RAL. Al igual que en los casos anteriores, esta herramienta trabaja con un fichero de configuración y un fichero .ndx a partir del cual informar del conjunto de datos utilizado para entrenar la matriz. Igual que en el apartado anterior, el conjunto de datos a utilizar estará formado por los diferentes segmentos en los que se divide el fichero de audio que se está procesando. Así pues, para comenzar, el binario procesa los datos de entrada, es decir, los ficheros de etiquetas previamente generados para cada segmento. Una vez procesados los datos, el programa inicializa una matriz aleatoria. A partir de este punto se inicia un proceso iterativo para estimar la matriz de variabilidad total de nuevo mediante el algoritmo Expectation-Maximization, de manera que en cada iteración se actualiza la matriz hasta obtener el resultado final, obteniendo así el fichero de salida denominado TV.matx. Igual que en el caso anterior, el número de iteraciones, el rango de la matriz y el resto de parámetros se informan desde el fichero de configuración. En nuestro caso, hemos entrenado una matriz de rango 10 en un proceso de 10 iteraciones.

Obtención i-vector:

Una vez obtenido el UBM y la Matriz de Variabilidad Total necesarios para obtener los i-vectors según la expresión del apartado 2.1.1.3, nuestro sistema deberá procesar los segmentos del fichero de audio que se está tratando con objetivo de obtener un i-vector de cada uno de ellos. Para ello, nuestro sistema sistema utiliza el programa *IvExtractor* del paquete LIA_SpkDet de la librería LIA_RAL, que cargará los ficheros de salida procedentes de las etapas anteriores, *world.gmm* (UBM) y *TV.matx* (Matriz de Variabilidad Total) y procesará los datos de entrada, que en este caso serán una vez más los diferentes segmentos en los que inicialmente se divide el fichero de audio de entrada así como los segmento que representan cada Target, para cada uno de los cuales se obtendrá un fichero de salida con extensión .y (“*nombre_del_segmento”.y)* que representa el i-vector del segmento en cuestión. Una vez obtenidos los i-vectors, deberemos distinguir entre Targets y segmentos de análisis, y obtener un score que determine la similitud entre ambos. Para ello, la siguiente etapa de nuestro algoritmo consistirá en un proceso de scoring, tal y como se explica en el siguiente apartado.

3.1.3. Scoring:

El proceso de scoring tiene como objetivo obtener una medida de la comparación entre los i-vectors. Tal y como se ha explicado en apartados anteriores, el objetivo de nuestro proceso de scoring será obtener una medida de la similitud entre los segmentos que representan cada uno de los targets con cada uno de los segmentos de análisis en los que se ha dividido el fichero de audio a partir de sus i-vectors, obtenidos en el bloque

anterior. Para ello, de las técnicas de scoring analizadas en el apartado 2.1.1.4 hemos decidido incorporar a nuestro sistema la similitud coseno, ya que dada su simplicidad se trata de un proceso rápido de bajo coste computacional a la par que ofrece resultados óptimos. Para el proceso de scoring, nuestro sistema integra la herramienta *IvTest* del mismo toolkit que las utilizadas en las etapas anteriores. En este caso, la herramienta trabaja con dos ficheros de entrada .ndx. Por un lado, *trainModel.ndx* en el que se debe indicar la etiqueta de cada target junto al nombre del fichero que representa a dicho target siguiendo el siguiente formato a modo de ejemplo:

spk0 130614FR52100_B_MPG_T0

spk1 130614FR52100_B_MPG_T1

spk2 130614FR52100_B_MPG_T2

spk3 130614FR52100_B_MPG_T3

spk4 130614FR52100_B_MPG_T4

Figura 9: Ejemplo asignación de targets

Por otro lado, el fichero *ivTest_plda_target-seg.ndx* en el que se debe indicar los segmentos a comparar siguiendo el siguiente formato a modo de ejemplo:

130614FR52200_B_MPG_S0 spk0 spk1 spk2 spk3 spk4 spk5 spk6 spk7

130614FR52200_B_MPG_S1 spk0 spk1 spk2 spk3 spk4 spk5 spk6 spk7

130614FR52200_B_MPG_S2 spk0 spk1 spk2 spk3 spk4 spk5 spk6 spk7

130614FR52200_B_MPG_S3 spk0 spk1 spk2 spk3 spk4 spk5 spk6 spk7

Figura 10: Ejemplo fichero pre-scoring

En la primera columna podemos ver el segmento de análisis que queremos clasificar mientras que el resto indican cada uno de los targets con los que se comparará el segmento de análisis de la primera columna mediante la técnica score coseno, cuyas bases teóricas han sido explicadas en apartados anteriores.

A partir de estos dos ficheros y de los i-vectors obtenidos en el paso anterior, el programa devuelve un fichero .txt en el que se muestran los scores resultantes de comparar el i-vector de cada segmento de análisis con en i-vector de cada uno de los targets, tal y como puede verse en el siguiente ejemplo:

M spk0 0 130614FR52100_B_MPG_S0 -0.304711

M spk1 1 130614FR52100_B_MPG_S0 0.0418345

M spk2 0 130614FR52100_B_MPG_S0 -0.109001

M spk3 1 130614FR52100_B_MPG_S0 0.41681

M spk4 0 130614FR52100_B_MPG_S0 -0.00334372

M spk5 0 130614FR52100_B_MPG_S0 -0.267411

Figura 11: Ejemplo de fichero de salida del proceso de scoring

Dicho fichero de texto, será utilizado en la siguiente y última etapa para clasificar cada uno de los segmentos de análisis según su semejanza con cada uno de los targets, tal y como se explica a continuación.

3.1.4. Decisión final:

La finalidad de esta última etapa de nuestro sistema de tracking consiste en generar un fichero de salida indicando cual es el target que aparece en cada uno de los segmentos en los que se ha dividido el audio. Para ello, se ha implementado un script en python que a partir del fichero de salida del proceso de scoring, elija para cada segmento el target para el que se ha obtenido un mayor score. Finalmente, se determinará que puede considerarse que en dicho segmento habla dicho target siempre y cuando el score entre ambos supere un umbral prefijado.

Dicho umbral ha sido fijado de manera experimental, a partir de los resultados obtenidos en diferentes pruebas. Tal y como podrá verse en el apartado de resultados, se ha definido como umbral óptimo un valor de 0.5.

De esta manera, se obtiene un fichero de salida en el que cada intervalo de video en el que una persona habla está etiquetado con el nombre del locutor que en el aparece, cumpliendo así el objetivo de la técnica monomodal de audio, a integrar en el sistema completo posteriormente explicado.

3.1.5. Evaluación del algoritmo monomodal de audio.

Para evaluar el comportamiento del sistema de tracking implementado como tecnología monomodal de audio a integrar en el nuevo sistema multimodal para el Mediaeval 2016, se ha propuesto plantear el problema como si se tratara de un sistema de verificación. Es decir, para cada segmento, se verificará, a partir de un fichero de referencia si en dicho segmento aparece o no el locutor que le ha asignado el tracker. Para ello, necesitaremos un fichero de referencia en el que aparezca un etiquetado de quien aparece realmente en cada momento. Partiendo de esta información, se ha implementado un script en Python que contabiliza los segmentos acertados y los erróneos.

Puede entenderse como un clasificador que clasifica los diferentes segmentos de manera correcta (el locutor que ha asignado el tracker coincide con el locutor del fichero de referencia) o errónea (el locutor que ha asignado el tracker no coincide con el locutor del fichero de referencia). Así pues, para determinas la precisión y probabilidad de pérdida de nuestro clasificador, se han establecido las siguientes métricas:

Precisión-Recall

En adelante, hablaremos de segmentos detectados o positivos (P) como aquellos cuyo score (distancia coseno con el target más semejante) está por encima del umbral de decisión y de segmentos rechazados o negativos (N) como aquellos cuyo score está por debajo del umbral.

Por un lado, se define como Precisión la fracción de segmentos detectados acertados (TP), de total de segmentos detectados, de manera que evalúa la precisión o probabilidad de acierto del sistema. Por otro lado, se define como Recall o Sensibilidad como la fracción de segmentos detectados acertados respecto al total de segmentos. Denominaremos grupo de segmentos total como Verdadero o True (T). Se relaciona con la probabilidad de pérdida del sistema. Así pues, a mayor umbral de clasificación,

obtendremos un menor número de muestras detectadas, con lo que aumentará la precisión de nuestro sistema a costa de una disminución del recall, ya que se perderán más segmentos. Por el contrario, un umbral menor dará lugar a un mayor recall, ya que el sistema sufrirá menos pérdidas, pero obtendremos menor precisión.

Con objetivo de solventar el compromiso entre precisión y recall, se calculan las curvas precisión recall, en las que se representa la precisión en función del recall para diferentes umbrales. Dichas curvas, permiten observar cual es el umbral óptimo que permite equilibrar ambos parámetros, teniendo en cuenta que los dos deben tender a la unidad.

$$Precision = \frac{TP}{P} \qquad Recall = \frac{TP}{T}$$

Curvas DET:

En el caso de las curvas DET (Detection Error Trade-off), el objetivo es evaluar las tasas de error del clasificador, en cuanto a Falsa alarma (segmento detectado erróneo,FP) o Faslo rechazo (segmento rechazado correcto, FN).

Esta vez, se representa la probabilidad de falsa alarma, obtenida como la fracción de muestras detectadas que son falsas, es decir erróneas, en función de la probabilidad de falso rechazo, que se calcula como la fracción de segmentos rechazados que son verdaderos, es decir, correctos respecto al total de segmentos rechazados.

En el apartado de resultados podrán observarse dichas curvas como resultado de diferentes pruebas de tracker.

$$Falsa\ Alarma = \frac{FP}{P} \qquad Falso\ rechazo = \frac{FN}{N}$$

3.2. Sistema completo:

Tal y como ha sido comentado, el presente proyecto constituye un bloque del sistema final a implementar. Así pues, ha sido integrado como tecnología monomodal de audio, junto con las tecnologías monomodales de imagen y texto en el sistema multimodal final para el reconocimiento de personas.

Siguiendo los objetivos de la competición, los diferentes shows de la base de datos, serán divididos en shows de aproximadamente 3 segundos, y el objetivo del sistema será etiquetar cada shot con el nombre de la persona que habla y aparece en pantalla. Como ha sido dicho, la tarea debe realizarse de manera totalmente no supervisada, sin ninguna información a priori de las personas que aparecen en los videos. Por lo tanto, el sistema deberá ser capaz de explotar la información que aparece en pantalla para identificar a los personajes.

Figura 12: Ejemplo de la tarea de detección multimodal de personas

Para ello, se necesitarán las tres tecnologías multimodales (Audio, Texto e imagen) explicadas anteriormente de las cuales obtener indexaciones monomodades. Los ficheros de etiquetas a nivel nomonodal de los que se pueden extraer las identidades de caras, el texto o el locutor que aparece en cada segmento de vídeo o shot deberán ser procesados para obtener un etiquetado global que determine finalmente quién es la persona que habla y aparece en cada shot. Para ello, se requiere de un algoritmo de fusión que procese los ficheros de etiquetas procedentes de las tres tecnologías monomodales con objetivo de obtener el etiquetado final. A modo de ejemplo, la siguiente figura muestra la representación temporal del resultado del etiquetado de las tres tecnologías monomodales.

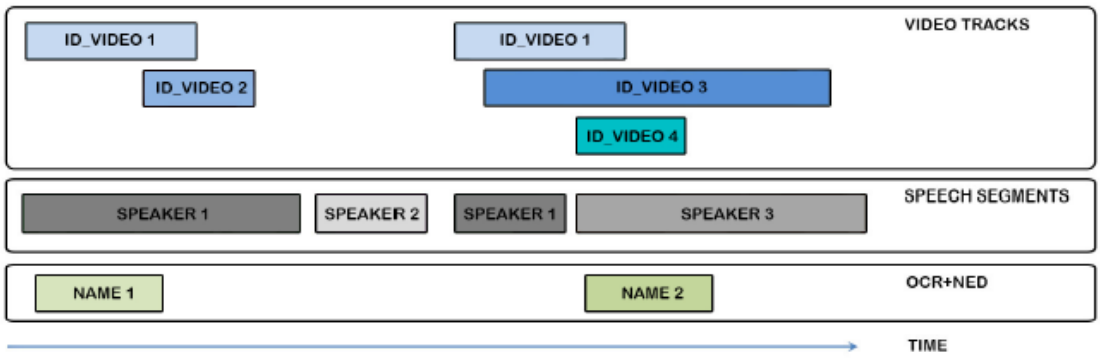

Figura 13: Representación temporal de la salida de los algoritmos monomodales.

Como puede verse, contrariamente al etiquetado de texto y voz, en el etiquetado de vídeo, pueden solaparse dos o más tracks, es decir, pueden aparecer varias identidades de caras en un mismo segmento, ya que en los videos suele aparecer más de una persona en pantalla por lo que no siempre estará claro a cuál de las identidades pertenece la voz o el nombre que aparece en pantalla. Además, dado que objetivo del sistema es determinar quién habla y aparece al mismo tiempo, debe tenerse en cuenta que puede escucharse la voz de una persona que no es la que aparece en pantalla o un nombre que no pertenezca a ninguna de ellas, de manera que el algoritmo de fusión deberá tener en cuenta todas estas restricciones.

A continuación, pasara e explicarse el algoritmo multimodal utilizado el año pasado así como una nueva alternativa de fusión que fue implementada al principio del proyecto como opción de mejora multimodal, ya que en el apartado de resultados podrá verse la comparación entre ambos sistemas. Sin embargo, para el nuevo sistema a presentar en Mediaeval 2016 se ha implementado un nuevo algoritmo basado en la técnica de evaluación de la competición, con objetivo de adaptar el sistema y optimizar los resultados. Así pues, el nuevo sistema será expuesto una vez explicada la métrica de evaluación.

3.2.1. ILP:

Dicho algoritmo propone obtener el etiquetado final en tres pasos que consisten en dos fusiones por pares de fuentes y el etiquetado final, tal y como puede verse en la siguiente figura:

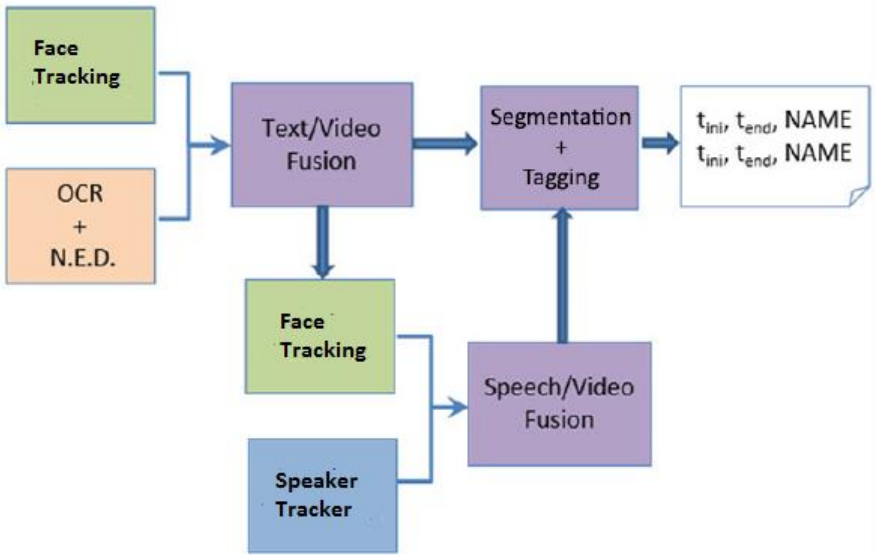

Figura 14: Diagrama de bloques algoritmo multimodal Mediaeval 2016

Como puede observarse, el primer paso del algoritmo consiste en la fusión de la información procedente de las fuentes de imagen y texto a partir del cual obtendremos los nombres de las caras que aparecen en cada segmento, descartando aquellas caras a las que no se les asigne ningún nombre. Seguidamente, se lleva a cabo la fusión Audio/Imagen, a partir de la cual se asocia a cada cara su voz .Finalmente, se lleva a cabo el proceso de etiquetado en el que a cada par cara-voz se le asigna el nombre procedente del paso1, obteniendo como resultado el fichero final en el que aparecen los nombres de la persona que habla y aparece en cada intervalo de vídeo.

Para realizar dichas fusiones, el sistema evalúa los intervalos de solape de los tracks procedentes (Imagen/Texto e Imagen/Voz) de cada par de fuentes.

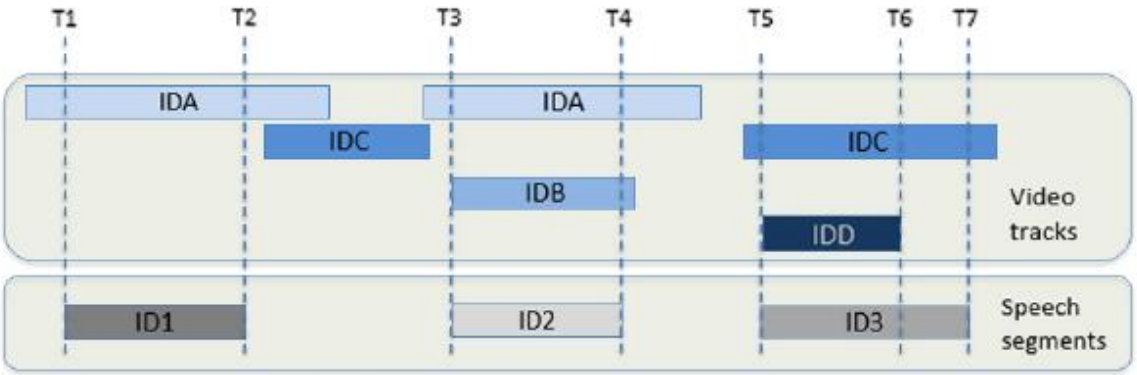

Figura 15: Ejemplo fusión Imagen- Voz

A partir de dichos solapes, el algoritmo genera una matriz de pesos que en el caso de la fusión Audio/Vídeo tendrá como columnas las identidades de video y como filas las identidades de voz y tendrá como valores los pesos de la asignación ID-vídeo/ID-voz calculados como el tiempo de solape entre ambas IDs entre el total de IDs de Vídeo que se solapan con la misma ID de audio tal y como se muestra a continuación:

$$Pesos = \begin{bmatrix} p(ID\ vídeo_1, ID\ voz_1) & \cdots & p(ID\ video_n, ID\ voz_1) \\ \vdots & \ddots & \vdots \\ p(ID\ vídeo_1, ID\ voz_n) & \cdots & p(ID\ vídeo_n, ID\ voz_n) \end{bmatrix}$$

Figura 16: Ejemplo matriz de pesos.

Dónde:

$$\mathrm{p(ID\ vídeo_1, ID\ voz_1)} = \frac{\mathrm{T_{solape\ ID\ Vídeo1-ID\ Voz1}}}{\mathrm{número\ de\ IDs\ de\ vídeo\ que\ se\ solapan\ con\ ID\ voz1}}$$

Finalmente, a partir de la matriz de pesos se realiza la asignación cara-voz asignando a cada cara la voz para la cual se ha obtenido un mayor peso. El proceso se realiza de la misma forma para la fusión Imagen/Texto.

A continuación se explicará la alternativa de algoritmo monomodal propuesta en este proyecto. Se trata de un sistema más simple basado en la asignación directa entre texto y voz.

3.2.2. Asignación Texto – Voz:

Según una observación de los vídeos de la base de datos, podemos comprobar que en la mayoría de los casos, cuando aparece un texto en pantalla, el nombre que en el aparece corresponde a la persona que está hablando en ese momento. Así pues, la propuesta de este sistema de fusión consiste en asumir que cada vez que aparece un nombre en pantalla a la vez que habla un locutor, dicho nombre pertenece a la persona que está hablando, de manera que se producirá la asignación directa entre el locutor y el nombre. Para ello, se ha implementado por medio de un script en Python un sistema que procese tanto el fichero de salida de la fuente de audio como el fichero procedente de la fuente de texto. Partiendo de esta información, se analizan los solapes de ambos ficheros de etiquetas, de manera que cuando se detecta un solape entre los intervalos del etiquetado de texto con los de voz, se le asigna al speaker detectado en dicho intervalo de tiempo el nombre correspondiente al texto detectado en dicho intervalo. Dada la asignación, cada vez que dicho locutor vuelva a aparecer en el vídeo, es identificado con el nombre correspondiente.

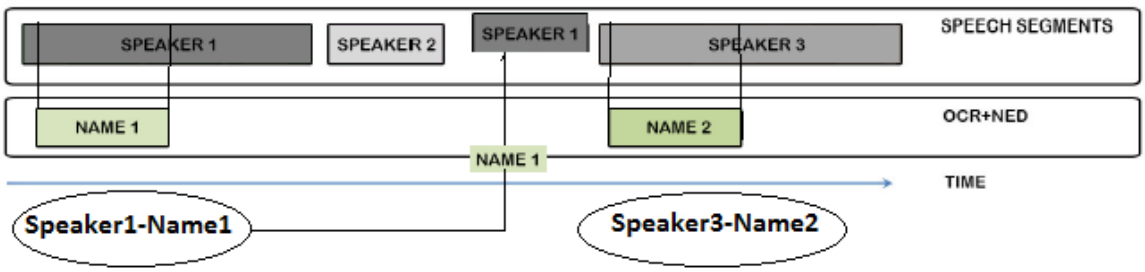

Figura 17: Sistema de asignación directa Texto-Voz

3.2.3. Evaluación del sistema completo:

El contenido de este apartado se centra en analizar las métricas de evaluación utilizadas para determinar el rendimiento del sistema completo de reconocimiento. Se trata de las métricas establecidas en las bases de la competición para evaluar los sistemas de reconocimiento de personas de los participantes y proclamar un ganador. Como en cualquier sistema de detección, se evalúa tanto el acierto o precisión del sistema como las pérdidas. Para ello, los diferentes shows de la base de datos de la competición son divididos en shots de aproximadamente 3 segundos y se deberá comprobar si para cada shot si la persona asignada por el sistema multimodal en dicho shot coincide con la persona en la que realmente aparece en dicho shot. Para ello, Mediaeval proporciona un fichero de referencia en el que aparece que persona o personas aparecen en cada shot, que servirá para contabilizar los shots acertados o fallados. A partir de esta información se calcula la métrica de la competición denominada MAP, Mean Average Precision.

Mean Average Precision (MAP):

Se trata de una evaluación a nivel de Query, entendiendo el conjunto de *Querys* como el conjunto de personas a identificar dentro de la base de datos, es decir, todas las personas que aparecen a la vez que hablan en los vídeos.

Consiste en calcular para cada query, es decir para cada persona presente en la base de datos de la competición (Ej: Joan Laporta), lo que se conoce como Average Precision, AP.

Así pues, para calcular el Average Precision de una query, buscaremos el conjunto de shots en los que el sistema ha detectado a la persona en cuestión. Una vez detectados, dichos shots serán ordenados en función de un parámetro de confianza que acompaña a cada detección. El objetivo de tal parámetro reside en determinar la fiabilidad de que en en un shot aparezca la persona que se ha detectado, o dicho de otra forma, la seguridad de que el sistema haya acertado en cada shot. Dicho parámetro puede extraerse a partir de las tecnologías monomodales o bien obtener una confianza multimodal. En el caso de audio, se ha utilizado el score de la distancia coseno como parámetro de confianza a nivel monomodal, ya que un mayor score entre el segmento de análisis y el target detectado, significa que ambos segmentos son muy semejantes con lo que existe una mayor probabilidad de acierto. Una vez ordenados los shots, se calculará el Average Precision de un query como el sumatorio en función de k, donde k va de 1 hasta el número de detecciones de la query en cuestión, de la precisión en la posición k, es decir, teniendo en cuenta k detecciones, entre k. Solamente contribuirán al sumatorio las precisiones en las que la detección k es correcta, tal y como indica la siguiente expresión:

$$AP(q) = \sum_{k=1}^{N} \frac{p(k) x\, R}{k}$$

Donde N representa el número de shots en los que el sistema ha detectado la query y P(k) es la precisión obtenida en la posición k, es decir teniendo en cuenta las k primeras detecciones según la ordenación a partir de las confianzas, R toma valor 1 cuando el shot k es correcto y 0 en el caso contrario, de manera que si las detecciones erróneas queden al final del sumatorio no penalizarán al resultado, ya que se calculará las diferentes precisión teniendo en cuenta que los k shots detectados son correctos.

Así pues, dado el análisis de las métricas de evaluación de la competición, podemos concluir que para optimizar los resultados de nuestro sistema según las bases de la

competición, es importante obtener confianzas fiables que dejen los posibles errores al final, para que no repercutan en el resultado.

Finalmente, se calcula el Mean Average Precision como el promedio de los Average Precision calculados para cada query, siguiendo la siguiente expresión:

$$MAP = \frac{1}{|Q|}\sum_{q\epsilon Q} AP(q)$$

Donde Q representa el conjunto de querys o personas en la base de datos.

El siguiente apartado muestra los resultados de la realización de diferentes pruebas en términos de las métricas explicadas.

Tal y como se ha mencionado, se ha integrado al sistema final un algoritmo de fusión que explote la ventaja de los scores, de manera que pretende ordenar correctamente las detecciones con objetivo de optimizar resultados. Así pues, dicha técnica consiste en analizar el etiquetado de voz y el de caras a nivel de shot procedente de las tecnologías monomodales. Ambos ficheros de etiquetas están indexados con el nombre de la persona detectada en cada shot. Para realizar la fusión de ambos etiquetados se distinguen los siguientes casos:

- Caso 1: En un mismo shot coincide la identidad asignada por el algoritmo de audio y de imagen, de forma que se asigna la identidad a dicho shot con un score calculado como el promedio del score procedente del fichero de audio y el de voz.
- Caso 2: La identidad asignada a un shot por ambos algoritmos no coincide. En este caso, se le asignan al mismo shot las dos identidades asignándoles a cada una a la mitad des score procedente de la tecnología monomodal.
- Caso 3: Un shot solo tiene asignada la identidad procedente de una de las dos tecnologías monomodales. Se le asigna dicho identidad con la mitad del score.

De esta manera, se consigue una manera de determinar la confianza de cada asignación.

4. Resultados

Es el momento de analizar el rendimiento de los sistemas implementados según las métricas previamente explicadas. Para ello, en este apartado se describe el análisis de los resultados obtenidos tanto de la tecnología monomodal de audio, es decir, el sistema de tracking, como del sistema completo, de manera que se compararán los resultados obtenidos de los dos algoritmos de fusión anteriormente descritos. Así pues, dicho análisis será separado en dos partes, de forma que por un lado se explicarán los resultados procedentes de diferentes experimentos probados en el tracker de audio y por otro lado se analizará la comparación de los dos algortimos de fusión (ILP y Asignación directa Texto-Vídeo) según los resultados obtenidos de la evaluación del sistema completo integrando uno u otro algoritmo.

A continuación, pasarán a detallarse los diferentes experimentos que se han realizado tanto como para ajustar los parámetros del tracker, (el umbral, la duración de los segmentos…) como para evaluar el rendimiento final, y se presentarán los resultados obtenidos.

4.1. Resultados del sistema de seguimiento de audio:

Con objetivo de determinar el rendimiento del sistema de tracking implementado, se han realizado diferentes pruebas y experimento utilizando vídeos de diferentes bases de datos, que serán explicados a continuación. Sin embargo, antes de probar el sistema final, será necesario fijar ciertos parámetros tales como el umbral de decisión o la duración de los segmentos. Para ello, se deberán realizar varias pruebas a partir de una base de datos de training con objetivo de determinar la configuración que ofrece mejores resultados.

4.1.1. Base de datos de Desarrollo:

Para realizar una primera evaluación del tracker así como fijar los parámetros previamente comentados, se ha elegido utilizar como base de datos de training una base de datos sencilla denominada Ágora, la cual se compone de un conjunto de videos de un programa de telenoticias de la televisión catalana. Así pues en los diferentes vídeos aparecen diferentes locutores tales como el presentador o las personas entrevistadas que nuestro sistema de tracking deberá reconocer.

Tal y como se ha expuesto anteriormente, el rendimiento del tracker implementado depende en gran medida del entrenamiento de los targets, concretamente de los segmentos utilizados en el proceso de entrenamiento. Así pues, con objetivo de reflejar dicha influencia centraremos el análisis en la comparación de las técnicas de segmentación de targets explicadas en el apartado 3.1.1.1.

Segmentación de locutores a detectar a partir de ficheros escuchados:

El primer experimento consiste en probar el tracker seleccionando manualmente los segmentos con los que se entrenarán los targets. Para ello, se ha escogido un vídeo de la base de datos Ágora y se han extraído segmentos en los que se ha escuchado hablar a cada target, un segmento por target, a partir del cual deberán detectarse los demás segmentos en los que vuelve a aparecer. Con objetivo de determinar el impacto de la duración de los segmentos de entrenamiento en el rendimiento del sistema fina, se han escogido segmentos de 3, 5 y 10 segundos obteniendo los resultados que pueden verse a continuación:

Segmentos de 3 segundos

Th	0.3	0.4	0.5	0.6	0.7	0.8
Precision	0,48923077	0,53544776	0,60740741	0,7250996	0,79487179	0,86206897
Recall	0,95495495	0,86186186	0,73873874	0,54654655	0,37237237	0,22522523
FR	0,510769231	0,464552239	0,392592593	0,2749003	0,20512820	0,13793103
FA	0,159574468	0,221153846	0,256637168	0,3062880	0,35544217	0,3926940

Tabla 1: Resultados del sistema de seguimiento según el primer experimento

En la tabla se muestran los resultados de probar el tracker en las condiciones descritas anteriormente, entrenamiento de los targets a partir de segmentos de 3 segundos obtenidos manualmente, para un barrido de diferentes umbrales de decisión. Para la segmentación del resto de la señal, se han utilizado segmentos de 3 segundos.

Finalmente, a partir del barrido de umbrales, podemos determinar que los resultados óptimos se obtienen con un umbral de T=0.5 por lo que a partir de este punto se calcularán las métricas de evaluación en base a dicho umbral.

Segmentos de 5 segundos:

A continuación, se muestra la tabla de resultados obtenidos siguiendo el mismo procedimiento pero escogiendo segmentos de 5 segundos:

Precision	0,85641026
Recall	0,9092559
FA	0,1435897
FR	0,3144654

Tabla 2: Resultados del sistema de seguimiento según el segundo experimento

Segmentos de 10 Segundos:

Precision	0,87716263
Recall	0,91516245
FA	0,122837
FR	0,283132

Tabla 3: Resultados del sistema de seguimiento según el segundo experimento

En este caso, hemos utilizado segmentos de 10 segundos, de manera que podemos observar que ha mayor duración de los segmentos utilizados se obtienen mejores resultados, ya que disponemos de más información para generar un modelo del locutor fiable, tal y cómo muestra la siguiente tabla resumen:

Duración (Seg)	3	5	10
Precision	0,60740741	0,85641026	0,87716263
Recall	0,73873874	0,9092559	0,91516245
FR	0,3925925	0,1435897	0,1228373
FA	0,2566371	0,3144654	0,283132

Tabla 4: Tabla resumen de resultados

Segmentación de los targets a partir de ficheros de referencia:

En el siguiente experimento, se han extraído como segmentos de entrenamiento de los targets los segmentos etiquetados a partir de un fichero de referencia .rttm. Se trata de un fichero en el que los diferentes intervalos de un fichero de audio aparecen etiquetados con la etiqueta del locutor que aparece en cada intervalo, de manera que podemos extraer información sobre quién habla en cada momento. Igual que en el apartado anterior, de han realizado diferentes pruebas en base a la duración de los segmentos utilizados. Esta vez, se han utilizado para la evaluación 10 vídeos de la base de datos. A continuación, se muestran las curvas precisión-recall y curvas DET obtenidas para segmentos de 3,5 y 10 segundos.

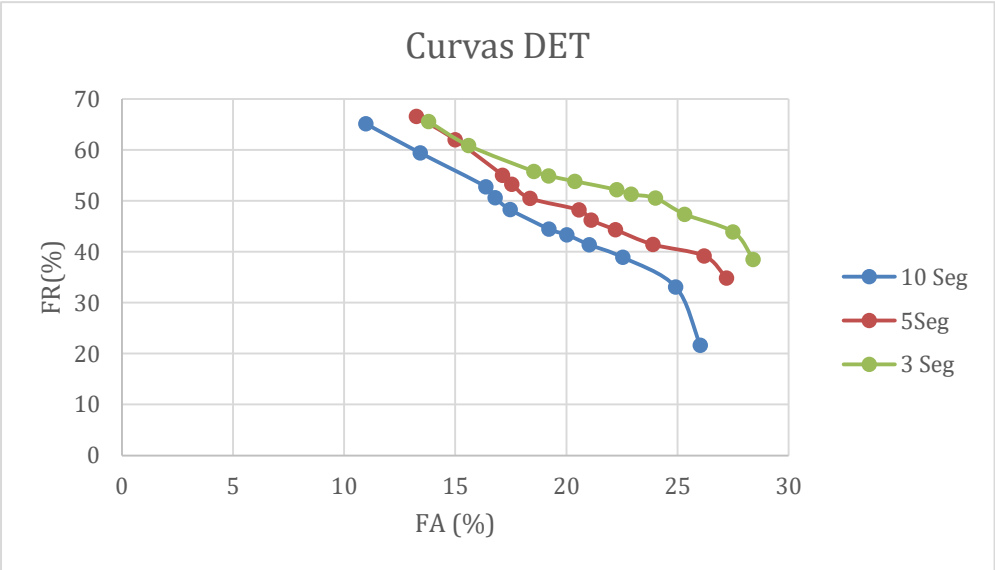

Figura 18: Curvas DET base de datos de desarrollo

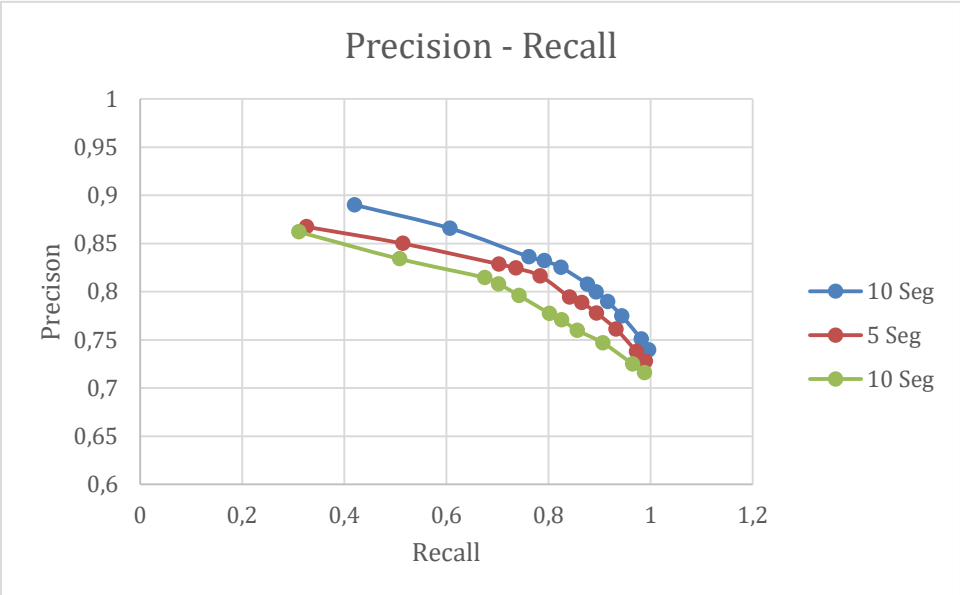

Figura 19: Curvas Precisión-Recall base de datos de desarrollo

Una vez más, podemos comprobar que los resultados óptimos se producen al utilizar segmentos de 10 segundos.

Finalmente, se adjunta una tabla resumen que compara los mejores casos de los algoritmos anteriores, es decir, segmentos de 10 segundos y umbral de 0.5:

	Segmentación a partir de ficheros escuchados	Segmentación a partir de ficheros .rttm
Precision	0,87716263	0,8251993
Recall	0,91516245	0,82503888
FA	0,122837	0,482555019
FR	0,283132	0,17496112

Tabla 5: Comparación sistemas de segmentación

Segmentación de los targets partiendo del sistema monomodal de texto:

En el caso del sistema de tracking a integrar en el sistema de reconocimiento final, ninguna de las opciones anteriores podrá ser utilizada, ya que no es viable etiquetar manualmente el total de vídeos de la base de datos de test y no disponemos de ficheros de referencia ni ninguna otra información a priori. Así pues, la propuesta de tracker final cosiste en entrenar los targets partiendo de la segmentación procedente el optical carácter recognition, en la que aparecen los nombres que aparecen en cada intervalo de vídeo. De esta forma, elegiremos para cada target el segmento de vídeo en el que aparece su nombre. Este último experimento ha sido probado con la base de datos de test, por lo que será explicado en el siguiente apartado.

4.1.2. Base de datos de test.

En este apartado se describirá el rendimiento del tracker configurado para ser integrado en el sistema final, es decir, el tracker que utiliza la información del texto para entrenar los targets. En este caso, el tracker se ha evaluado con la base de datos de la competición, Mediaeval 2016. Se trata de un conjunto de 865 videos en diferentes idiomas ya que proceden da la televisión, francesa y catalana.

En este caso, dada la importancia de la duración de los segmentos de entrenamiento de los targets para obtener un buen resultado final, la salida del algoritmo de texto, OCR, ha sido procesada con objetivo de no limitar al sistema a extraer únicamente el segmento en el que aparece el texto. Así pues, se han integrado varias alternativas que permiten resegmentar el fichero procedente del OCR teniendo en cuenta que en la mayoría de los casos el locutor habla segundos antes y segundos después de que aparezca el texto de manera que se pueden utilizar segmentos más largos para entrenar los targets. Es por esto que se ha realizado una primera prueba con la salida original del OCR y una segunda prueba con la salida del OCR procesada, con los segmentos más largos. El procesado consiste en utilizar los resultados procedentes de un proceso de diarización, proceso mediante el cual se obtiene una indexación de audio con la identidad de la persona que habla en cada segmento, de manera que si un segmento de texto coincide con un segmento de la diarización, de le asigna a dicho nombre el segmento completo de la diarización, obteniendo un segmento más largo para entrenar al locutor.

La siguiente tabla, muestra una comparativa de ambos experimentos junto con el baseline de la competición:

	MAP
Baseline	11,60%
OCR	16,00%
OCR-Procesado	33,40%

Tabla 6: Comparación experimentos para la competición

Como podemos observar, en ambos casos se ha conseguido superar el baseline.

4.2. Resultados sistema completo:

Finalmente, pasaremos a analizar los resultados del sistema monomodal final. Como ha sido dicho, en este apartado podrá verse la comparativa de resultados finales obtenidos mediante los algoritmos de fusión previamente explicados. Para ello, la siguiente tabla muestra el resultado de probar ambos sistemas esta vez con la base de datos del concurso del año pasado, Mediaeval 2015, ya que son los datos de los que se disponía al principio del proyecto:

	MAP	Precision	Recall
ILP	30.10 %	0.349519136684	0.313088443584
Text-Speaker (Asignacion directa)	38.29%	0.473454925983	0.383847963537

Tabla 7: Comparación algoritmos de fusión.

A pesar de mejorar ligeramente los resultados, ambos sistemas ofrecen un rendimiento notablemente bajo, por lo que finalmente se optó por la implementación de un nuevo algoritmo de fusión con el que obtener mejores resultados.

Para acabar, el resultado del sistema completo final con las nuevas técnicas monomodales, y el nuevo algoritmo multimodal, testeado con la base de datos de la competición ,se ha conseguido alcanzar un MAP= 60 %, obteniendo un segundo puesto en la competición.

5. Conclusiones:

Para acabar, del desarrollo del Proyecto anteriormente descrito puede extraerse que es posible obtener buenos resultados en un sistema de reconocimiento de locutor utilizando un sistema simple tal y como el que ha sido implementado. De esta manera, se demuestra el alto rendimiento de los sistemas basados en i-vectors, a pesar de que se necesitan segmentos largos del locutor para entrenar un modelo fiable. Además, dicha técnica abre distintos frentes de mejora como utilizar otras técnicas de scoring como podría ser un sistema PLDA (Probabilistic Linear Discriminant Analysis), tecnología de compensación de variabilidad que nos permite obtener un score entre dos modelos de locutor. Otras opciones de mejora podrían encontrarse en el entrenamiento del espacio de variabilidad total, en cuanto a la variación del número de gaussianas utilizadas, el número de iteraciones del entrenamiento o los datos de entrada.

En cuanto a la segmentación de la señal, podría incorporarse un sistema segmentador que fragmente la señal en segmentos en los que habla un único locutor, como paso previo a ser comparados, con objetivo de evital analizar segmentos en los q aparece más de un locutor.

En cuanto al sistema multimodal completo, se ha podido comprobar un aumento de resultados como consecuencia de la adaptación del sistema a la métrica de evaluación de la competición, doblando los resultados del sistema del año pasado.

Bibliografía:

[1] F. McGehee, "The Reliability of the Identification of the Human Voice," J General Psychology 17, 249 (1937)
[2] W D. Voiers, "Perceptual Bases ofSpeaker Identity," J Acoust. Soc. Am. 36, 1065 (1964).
[3] Xavier Anguera, Mateu Aguil_o, Chuck Wooters, Climent Nadeu, and Javier Hernando. Hybrid speech/non-speech detector applied to speaker diarization ofmeetings. 2006.
[4] Identificación biométrica de locutores para el ámbito forense.2008.
[5] Reconocimiento de comandos por voz con máquinas de soporte vectorial a través de bandas espectrales. 2007.