Reportes Técnicos

Proyecto ANII Fondo María Viñas “Herramientas automáticas de observación de aula para el análisis de prácticas docentes en clases a distancia”

2022-2023

Introducción general

El presente documento constituye una compilación de reportes técnicos elaborados en el marco del proyecto “Herramientas automáticas de observación de aula para el análisis de prácticas docentes en clases a distancia”, desarrollado por el Grupo de Procesamiento de Audio de la Facultad de Ingeniería de la Universidad de la República (UdelaR), en estrecha colaboración con Plan Ceibal.

El objetivo central del proyecto es investigar, adaptar y desarrollar técnicas de procesamiento de audio, aprendizaje automático y análisis de señales que permitan generar indicadores objetivos sobre el desarrollo de clases remotas. Estas herramientas buscan complementar y potenciar los procesos de evaluación y monitoreo pedagógico actualmente realizados por observadores humanos, en el contexto de los programas Ceibal en Inglés y Pensamiento Computacional, ambos caracterizados por su formato de enseñanza a distancia mediante videoconferencias entre docentes remotos y grupos de estudiantes en centros educativos.

El conjunto de documentos reunidos aquí describe el recorrido técnico y conceptual de esta línea de trabajo. A lo largo de los distintos reportes se abordan, en orden cronológico y temático, aspectos que van desde la revisión bibliográfica del estado del arte internacional, hasta la implementación de prototipos experimentales para tareas específicas de análisis automático. Cada uno de los informes representa una etapa del avance del proyecto y aporta componentes concretos a un sistema integral de observación automatizada de clases, orientado a combinar la evidencia acústica y lingüística de las grabaciones con los marcos pedagógicos ya utilizados por Ceibal.

El enfoque general combina métodos clásicos de procesamiento digital de señales de audio con técnicas modernas basadas en redes neuronales profundas y representaciones auto-supervisadas. Este equilibrio permite tanto aprovechar la interpretabilidad de los enfoques tradicionales como beneficiarse de la robustez y capacidad de generalización de los modelos de aprendizaje profundo. Asimismo, se busca que las soluciones propuestas sean escalables, transparentes y adaptables a las condiciones particulares de las grabaciones de clase del entorno Ceibal, que presentan desafíos técnicos específicos como ruido de fondo, múltiples hablantes, distintos acentos y variaciones en la calidad de los dispositivos de captura.

Estructura y contenido del documento

El documento reúne varios reportes técnicos breves, organizados temáticamente, que documentan los avances y resultados obtenidos en las distintas etapas del proyecto. Cada sección aborda un problema o módulo de trabajo específico:

1. **Análisis de clases con procesamiento de audio – Revisión bibliográfica**

Este primer informe presenta una revisión sistemática de la literatura internacional sobre procesamiento de audio aplicado al análisis de clases. Se describen distintas líneas de investigación, desde sistemas basados en energía de la señal (como *DART*, 2017) hasta métodos más sofisticados que utilizan características espectrales como MFCC y modelos de *deep learning* para *Classroom Activity Detection (CAD)*. Se destacan los enfoques multimodales que integran audio y video, y se discuten sus implicancias para el caso particular de las clases de Ceibal.

2. Estado del arte de sistemas de diarización de audio

Este segundo reporte profundiza en la tarea de *speaker diarization*, es decir, la identificación automática de “quién habla y cuándo”. Se explican los distintos componentes de un pipeline típico —detección de actividad de voz, embeddings de hablante, clustering y métricas de evaluación (DER, JER, EER)— y se comparan diferentes herramientas disponibles, como *pyannote.audio*, *SpeechBrain*, *UIS-RNN* y *Resemblyzer*. El documento sirve como referencia técnica para el desarrollo posterior de los módulos de detección de hablante en clases.

3. Análisis de pautas de observación de clases

En esta sección se analizan los instrumentos de observación de aula utilizados por los programas **Ceibal en Inglés** y **Pensamiento Computacional**, comparándolos con protocolos internacionales como CLASS y COPUS. Se identifican los principales indicadores pedagógicos susceptibles de automatización, tales como el tiempo de habla del docente, la participación de los estudiantes, los momentos de interacción o la ocurrencia de ruidos de ambiente. Este análisis permite definir criterios de priorización para las etapas posteriores de procesamiento automático.

4. Análisis de clases – Detección del idioma

Se desarrolla un procedimiento para identificar el idioma hablado en distintos tramos de una clase, utilizando el modelo *Whisper* de OpenAI. Este análisis es especialmente relevante para *Ceibal en Inglés*, donde el porcentaje de tiempo de habla en inglés constituye un indicador pedagógico clave. Se detalla el flujo de procesamiento (extracción de audio, segmentación por VAD, análisis probabilístico del idioma) y se presentan resultados preliminares alentadores.

5. Análisis automático de clases grabadas – Características clásicas de audio

Este informe aborda el análisis de parámetros acústicos tradicionales —frecuencia fundamental (f_0) y formantes (F_1, F_2)— como herramientas interpretables para distinguir hablantes y validar el desempeño de modelos automáticos de diarización. Se presentan histogramas, mapas de densidad y ejemplos de clasificación de hablante (docente versus alumno) obtenidos sobre clases reales de Ceibal en Inglés y Pensamiento Computacional, destacando tanto aciertos como limitaciones.

6. Análisis de transcripción de clase – Detección de frases predefinidas

Este módulo se centra en el análisis textual posterior a la transcripción automática de las clases. A partir de las transcripciones generadas por *Whisper*, se implementa un algoritmo de búsqueda de palabras y expresiones características del vocabulario y la gramática esperados según la currícula de Ceibal en Inglés. La herramienta permite identificar el uso efectivo de estructuras lingüísticas clave y generar estadísticas sobre su frecuencia y contexto.

7. Detección de segmentos de audio pregrabados

Aquí se propone un sistema de detección de fragmentos de audio reproducidos durante las clases, como materiales didácticos o ejercicios de *listening*. Basado en técnicas de *audio*

fingerprinting inspiradas en el trabajo de Haitsma y Kalker (2002), el módulo es capaz de reconocer con alta robustez grabaciones conocidas aun en condiciones degradadas o con ruido. Se documentan los parámetros de implementación y los resultados experimentales obtenidos con audios de la plataforma *Little Bridge*.

8. **Análisis automático de clases grabadas – Detección del final de la clase**

Este último informe aborda la detección automática del fin de la clase, un problema práctico que afecta el procesamiento de las grabaciones. Se comparan tres métodos: detección de actividad de voz (*VAD*), identificación del habla del docente mediante *CAD*, y un enfoque final basado en el análisis de la energía acumulada del audio. Este último demostró ser el más confiable, permitiendo excluir de manera automática los tramos irrelevantes posteriores a la finalización de la clase.

Síntesis general y proyección

En conjunto, los informes reunidos en este documento ofrecen una visión completa de los avances técnicos y metodológicos logrados por el Grupo de Procesamiento de Audio en la búsqueda de herramientas que integren la ingeniería de señales con las necesidades del análisis educativo. Cada uno de los módulos presentados constituye una pieza de un ecosistema de procesamiento más amplio, capaz de extraer de las grabaciones de clase información útil para la evaluación pedagógica.

El trabajo evidencia una evolución progresiva: desde el estudio de referencias teóricas y metodológicas, hacia el desarrollo de soluciones aplicadas, validadas sobre datos reales de las clases de Ceibal. A su vez, los resultados obtenidos permiten proyectar nuevas líneas de trabajo, entre ellas la integración de los diferentes módulos en una plataforma unificada de análisis, la incorporación de modelos auto-supervisados de última generación y la definición de métricas específicas para la interpretación educativa de los resultados.

Este documento, por lo tanto, no sólo compila reportes técnicos aislados, sino que representa una síntesis coherente del proceso de investigación aplicada, en la intersección entre la ingeniería, la inteligencia artificial y la educación. Su propósito es dejar constancia del estado actual del trabajo, ofrecer una base sólida para las próximas etapas del proyecto, y servir de referencia para futuras colaboraciones entre la comunidad académica y los programas educativos de Ceibal.

Autores

En cada caso se indican los autores de cada reporte.

Reporte 1

Análisis de clases con procesamiento de audio
Revisión bibliográfica

Autor: Germán Capdehourat

En este informe se presenta una revisión bibliográfica de distintas propuestas de procesamiento de audio aplicadas al análisis de clases.

Existen algunos trabajos cuyo objetivo es brindar indicadores que permiten asistir el análisis del dictado de una clase. Uno de ellos es Decibel Analysis for Research in Teaching (DART) [DART 2017], que en base a la potencia de la señal de audio, estima la ausencia de voz, la presencia de una voz o la de más de una voz hablando simultáneamente. La aplicación indica tener un desempeño de aproximadamente el 90% en dicha estimación.

Otros esfuerzos publicados como [UChile 2021] que proponen clasificar el tiempo de dictado en tres posibles categorías: “Presenting”, “Administration”, y “Guiding”. En este trabajo se realiza una clasificación en base a características más sofisticadas que la potencia de la señal como son las bandas Mel y coeficientes cepstrales de las bandas mel (MFCC). Estas características han demostrado ser muy versátiles y brindar buenos resultados en una gran variedad de aplicaciones reportadas en la literatura, en particular en procesamiento que involucra voz hablada [MFCC 2012].

Algunos artículos buscan objetivos más complejos, que involucran analizar el contenido específico de lo que está diciendo el docente, debiendo abordar temas semánticos de su discurso. Un ejemplo de ello es el trabajo [Questions 2018], donde el objetivo es identificar si el docente realiza preguntas “auténticas”, es decir que no tienen una respuesta predefinida que el docente espera. Otro trabajo de los mismos autores es [AthQues 2018] y en la misma línea [QUESTDET 2020] se enfoca en la detección de preguntas.

Volviendo al problema fundamental de detectar el tipo de actividad que se está dando en clase, el cual varios autores han denominado “Classroom Activity Detection” (CAD), existen diversos artículos recientes usando redes neuronales y aprendizaje profundo. Algunos ejemplos en esta línea son [CAD ICASSP 2019], [CAD AIED 2020], [CAD ICASSP 2020], [CAD AI4G 2020] y [CAD 2021]. También existen enfoques multimodal, integrando también las imágenes del video al audio, como por ejemplo [MULTI 2021].

Si bien estos trabajos abordan el problema de asistir al análisis del dictado de clases, existen algunas características particulares de las clases grabadas por Ceibal que plantean desafíos específicos. Por un lado, las clases que se desea analizar son dictadas a distancia, es decir que los estudiantes están en un salón de clase guiados por un docente ayudante y la clase es dictada por un docente a distancia. Por otro lado, la dinámica de una clase en el ámbito escolar presenta diferencias significativas respecto a clases dictadas en secundaria o de nivel terciario.

Referencias

[MFCC 2012] J. Martinez, H. Perez, E. Escamilla and M. M. Suzuki, "Speaker recognition using Mel frequency Cepstral Coefficients (MFCC) and Vector quantization (VQ) techniques," CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers, pp. 248-251, 2012.

[DART 2017] "Classroom Sound Can Be Used to Classify Teaching Practices in College Science Courses", Owens, Seidel, Wong et al., PNAS. vol. 114 no. 12, p. 3085–3090, 2017.

[UChile 2021] "What Classroom Audio Tells About Teaching: A Cost-effective Approach for Detection of Teaching Practices Using Spectral Audio Features", Danner Schlotterbeck, Pablo Uribe, Roberto Araya, Abelino Jimenez, and Daniela Caballero. In LAK21: 11th International Learning Analytics and Knowledge Conference, Abril 2021.

[UChile2 2021] "TARTA: Teacher Activity Recognizer from Transcriptions and Audio", Schlotterbeck, D., Uribe, P., Jiménez, A., Araya, R., van der Molen Moris, J., Caballero, D.. In: Roll, I., McNamara, D., Sosnovsky, S., Luckin, R., Dimitrova, V. (eds) Artificial Intelligence in Education. AIED 2021. Lecture Notes in Computer Science(), vol 12748. Springer, Cham.

[Questions 2018] "Automatically Measuring Question Authenticity in Real-World Classrooms.", Kelly, Sean, Andrew M. Olney, Patrick Donnelly, Martin Nystrand, and Sidney K. D'Mello. *Educational Researcher* 47, no. 7 (October 2018): 451–64. https://doi.org/10.3102/0013189X18785613.

[AthQues 2018] "An Open Vocabulary Approach for Estimating Teacher Use of Authentic Questions in Classroom Discourse", Cook, Connor; Olney, Andrew M.; Kelly, Sean; D'Mello, Sidney K. International Educational Data Mining Society, Paper presented at the International Conference on Educational Data Mining (EDM) (11th, Raleigh, NC, Jul 16-20, 2018).

[CAD ICASSP 2019] "Deep Learning for Classroom Activity Detection from Audio", R. Cosbey, A. Wusterbarth and B. Hutchinson, ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 3727-3731.

[CAD AIED 2020] Li, H., Wang, Z., Tang, J., Ding, W., Liu, Z. (2020). Siamese Neural Networks for Class Activity Detection. In: Bittencourt, I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds) Artificial Intelligence in Education. AIED 2020. Lecture Notes in Computer Science(), vol 12164. Springer, Cham. (presenta YouTube)

[CAD ICASSP 2020] H. Li et al., "Multimodal Learning for Classroom Activity Detection," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 9234-9238. (presenta YouTube)

[CAD AI4G 2020] "Towards an Audio-based CNN for Classroom Observation on a Smartwatch", I. Zualkernan and M. S. Khan, 2020 IEEE / ITU International Conference on Artificial Intelligence for Good (AI4G), 2020, pp. 224-229.

[CAD 2021] Fine-Grained Classroom Activity Detection from Audio with Neural Networks, Slyman, Eric and Daw, Chris and Skrabut, Morgan and Usenko, Ana and Hutchinson, Brian, arXiv preprint https://arxiv.org/abs/2107.14369, 2021. (código)

[MULTI 2021] "Toward Automated Classroom Observation: Multimodal Machine Learning to Estimate CLASS Positive Climate and Negative Climate", A. Ramakrishnan, B. Zylich, E. Ottmar, J. Locasale-Crouch and J. Whitehill. In IEEE Transactions on Affective Computing, 2021.

[QUESTDET 2020] Neural Multi-task Learning for Teacher Question Detection in Online Classrooms. Huang, G.Y. et al. In: Bittencourt, I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds) Artificial Intelligence in Education. AIED 2020. Lecture Notes in Computer Science, vol 12163. Springer, Cham.

De los trabajos encontrados en la última revisión este resulta el más destacado por varios motivos:

[CAD 2021] Fine-Grained Classroom Activity Detection from Audio with Neural Networks, Slyman, Eric and Daw, Chris and Skrabut, Morgan and Usenko, Ana and Hutchinson, Brian, arXiv preprint https://arxiv.org/abs/2107.14369, 2021. (código)

- La motivación es muy similar a la nuestra, basada en un cambio en las prácticas pedagógicas que implica mayor interacción docente-estudiante. En este contexto, el análisis de clases cobra mayor valor, y hacerlo de forma manual no escala, por eso la necesidad de desarrollar métodos automáticos.
- Al ser el más reciente, cita y construye a partir de los trabajos previos. En particular, la línea de trabajo del [DART 2017], luego mejorada por [CAD ICASSP 2019] y también los trabajos de Li et al. ([CAD AIED 2020] y [CAD ICASSP 2020]) que se enfocan en el mismo problema que nosotros (distinguir al docente de los estudiantes).
- Trabajan con un etiquetado de 9 categorías, pero que puede ser simplificado a 4 o 5 categorías, las cuales son compatibles con trabajos previos:

Table 1: 4-, 5- and 9-way Classroom activity labels.

Activity	4-Way	5-Way	9-Way
instructor announcement	sgl	ist	a
instructor lecture			l
instructor asks question			iq
instructor answers question			ia
students asks question		stu	sq
students answers question			sa
group work	grp		g
silence	sil		s
other	oth		o

Por ejemplo las etiquetas **ist** (*instructor*) y **stu** (*student*), son equivalentes a las etiquetas **p** (*profesor*) y **a** (*alumno*) de nuestro etiquetado.

- Evalúan tres conjuntos de características diferentes:
  - Mel-filterbank: vector de 40 coeficientes mel, características clásicas utilizadas previamente para distintas aplicaciones de audio. Estas características se calculan para ventanas de 500ms, tomadas con un offset de 10ms.
  - OpenSMILE: usan esta herramienta open-source para extraer tres características de la voz: frecuencia fundamental, *loudness* y voicing probability. Estas características se calculan para ventanas de 50ms, tomadas con un offset de 10ms.
  - PASE+ Embeddings: usan la salida de este encoder de propósito general como la representación de las ventanas de 150ms tomadas con un offset de 10ms.

- Evalúan cuatro arquitecturas distintas como clasificadores, todas ellas basadas en redes neuronales:
  - DNN: red neuronal completamente conectada.
  - DTCNN: variante no causal de red convolucional (CNN).
  - GRU: arquitectura de red recurrente (RNN). También dicen haber probado LSTM sin mejores resultados.
  - BiGRU: variante no causal de la anterior, es una GRU bidireccional, haciendo una pasada hacia adelante y otra hacia atrás.
- Tienen el código disponible en un repositorio en github.

Como único punto negativo se destaca lo siguiente:
Trabaja con un conjunto de datos propio (58.7hs de clase anotadas), que aclaran no pueden compartir por restricciones del uso impuestas por la institución.

Reporte 2

Estado del Arte de Sistemas de Diarización de Audio

Autor: Braulio Ríos

1. Introducción

1.1. ¿Qué es diarización?

El objetivo de la diarización es detectar los cambios de locutor en una grabación e identificar qué segmentos de voz corresponden a un mismo locutor (figura 1), respondiendo a la pregunta ¿quién habló cuándo?

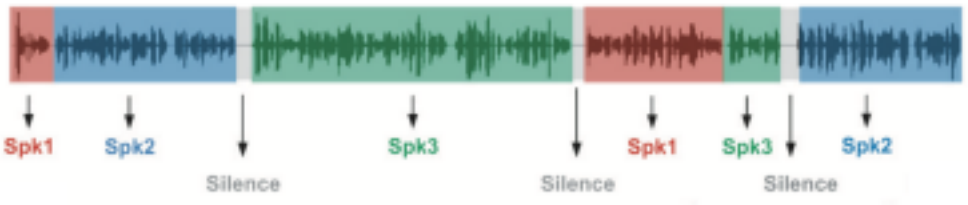


Figura 1: Ejemplo de diarización de audio de 3 hablantes (*speakers*) [Fuente: [1]]

Este problema es distinto al de transcripción, porque se enfoca en la segmentación del audio y la identificación de los distintos hablantes. La transcripción consiste en detectar qué fue lo que se dijo en cada parte y pasarlo a texto.
La diarización se puede usar para complementar la transcripción, de forma de saber además quién dijo cada parte. Sin embargo desde el punto de vista de las características de audio a utilizar, son problemas casi opuestos: para hacer diarización se necesita una representación del audio que permita separar voces distintas sin importar las palabras y sonidos particulares del fragmento; mientras que para la transcripción sucede lo contrario, ya que se necesita una representación que permita identificar bien palabras cualquiera sea la voz particular del hablante.
Más precisamente, esto significa que los vectores de características o *embeddings* que represen tan a cada fragmento de audio para aplicaciones de diarización o identificación de hablante, son muy diferentes en general de los que se utilizan para transcripción de texto, ya que sirven a proble mas distintos. Esta aclaración resulta oportuna porque recientemente han habido grandes avances en modelos de transcripción que utilizan aprendizaje profundo y grandes volúmenes de datos, y que muestran un funcionamiento muy bueno en plataformas de uso popular. Esto puede generar intuitivamente la idea de que la diarización es simplemente un sub-problema de la transcripción que ya debería estar completamente resuelto. Sin embargo, si bien los sistemas de diarización también se han servido del aprendizaje profundo para generar mejores resultados, es necesario entender que los avances en cada uno de estos problemas no se trasladan directamente en general, y que los desafíos que presentan pueden ser muy distintos.

1.2. Aplicaciones, problemas similares y algunos desafíos

La diarización de audio suele utilizarse como preprocesamiento para otras aplicaciones, o como un fin en sí mismo.
Por ejemplo, en el caso del reconocimiento de voz (en inglés *Acoustic Speech Recognition* o ASR) para transcripción de audio a texto como se ha mencionado, se suele hacer una etapa de diarización o detección de cambio de hablante para mejorar la información de contexto al realizar el reconocimiento.
A su vez, el resultado de la diarización puede ser de utilidad en sí mismo para analizar auto máticamente grabaciones telefónicas de atención al cliente, o entrevistas entre paciente y doctor. El caso de uso en que se hará énfasis en este trabajo, es el análisis automático de clases virtuales grabadas, con el fin de tener estimativos de los intervalos en los que habla el docente o los alumnos.
Un sub-problema de la diarización es el de detección de cambio de hablante, que en conjunto con la detección de actividad de voz (VAD o SAD por *Voice/Speech Activity Detector*), permite hacer segmentación de hablantes (*speech segmentation*). Todos estos se pueden considerar pasos previos ya que no requieren identificar si las voces después de cada cambio o en cada segmento ya habían participado previamente en la conversación, sino que basta con delimitar los segmentos de voz que potencialmente pertenecen a distintos hablantes.
Otro que se podría considerar un sub-problema, es el de verificación de hablante o *speaker verification*. En este caso, se requiere verificar si dos segmentos de audio dados pertenecen al mismo hablante. Las representaciones de audio que se usan para resolver este problema permiten distinguir voces entre sí, y por lo tanto pueden usarse como parte de un sistema de diarización.
Un sistema de diarización es básicamente la combinación de las dos tareas anteriores: primero delimitar los segmentos de voz, y segundo saber cuáles pertenecen a hablantes que ya habían participado previamente (el número de participantes no es conocido a priori). Por esta razón, los desafíos de diarización suelen tener distintos *tracks*, donde algunos se enfocan específicamente en resolver uno de los sub-problemas anteriores. Por ejemplo, en el desafío DIHARD 2018 [2] se tiene el primer track donde ya se cuenta con la segmentación de audio y se debe detectar a qué hablante pertenece cada segmento, y el segundo track es para diarización completa. En el desafío VoxSRC-21 [3] a su vez, se tienen los tres primeros tracks para hacer *speaker verification*, y sólo el cuarto es para diarización completa.

1.3. Evaluación: DER, JER, EER

La métrica más utilizada para esta tarea es la *Diarization Error Rate* o DER, definida como ([4], [2]):

$$DER = \frac{\text{False Alarm} + \text{Miss} + \text{Confusion}}{\text{Total}} \quad (1)$$

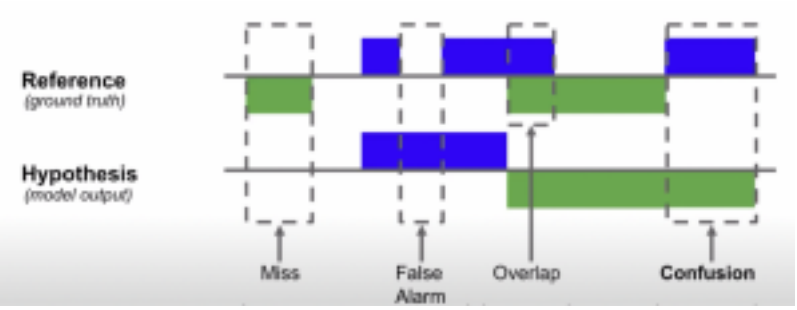


Figura 2: Tipos de error que puede cometer un sistema de diarización, mostrando un ejemplo con 2 hablantes (verde y azul). Imagen: [5]

Los componentes que participan en esta fórmula se pueden ver en la figura 2, donde además se incluye el *overlap*, que usualmente se descarta porque incluir estos segmentos y sumarlos en el numerador puede generar un error mayor a 1 [5]. Además, cuando la segmentación del audio es conocida y se brinda como entrada al sistema de diarización (como es el caso de algunos de los desafíos mencionados), sólo se utiliza *Confusion* para el cálculo del error (ya que *False Alarm* y Miss corresponden a errores de segmentación, específicamente de VAD).

Es importante notar que los *IDs* de hablante en la referencia y los que son detectados por el sistema, en general no coinciden. Por ejemplo, los hablantes etiquetados como 1, 2 y 3 de la referencia podrían coincidir con los que detectó el sistema pero les asignó IDs 2, 4 y 1. Hacer este mapeo no es trivial, ya que se requiere asignar los hablantes cuyos segmentos coincidan en mayor medida, y de forma única. Para eso en general se usa el algoritmo húngaro de asignación óptima. Este no es un problema único de la DER sino de cualquier métrica que se use para comparar las detecciones con las referencias, ya que no se puede conocer el mapeo de IDs a priori.

Como métrica secundaria para diarización suele usarse el *Jaccard Error Rate* o JER, introdu cido en [6] y basado en la popular métrica de intersección sobre la unión (IoU) o *Jaccard index* que se usa en detección de objetos. En este caso, se aplica el mismo principio a los segmentos de audio detectados por el sistema y los de referencia. Luego de hacer la asignación óptima (como se mencionó previamente), a cada ID de la referencia le corresponde un ID detectado, y sobre este par se mide la intersección sobre la unión temporal (IoU), y el error de ese segmento es $JER_i = 1 - IoU_i$. Luego se promedia este valor sobre todos los hablantes de referencia. Vale notar que la intersección de los segmentos, es el intervalo donde coinciden y por lo tanto corresponde a I = Total − False Alarm − Miss (ver figura 2, se descarta el *Overlap*, y en este caso no hay *Confusion* porque se calcula cada IoU entre un sólo hablante detectado y uno de referencia, luego de la asignación óptima mencionada). A su vez la unión corresponde a U = Total, y por lo tanto:

$$JER_{i=1-\frac{I}{U}} = \frac{\text{False Alarm + Miss}}{\text{Total}}$$

Luego el JER total es el promedio de este valor para cada par (hablante detectado, hablante de referencia) según fueron asignados de forma óptima.

Finalmente, vale mencionar también el *Equal Error Rate* o EER, que no aplica directamente a la tarea de diarización sino sólamente al módulo de verificación de hablantes. Puede usarse para evaluar la calidad de los *embeddings* de voz, que son vectores que deberían ser cercanos cuando las voces son similares y distantes en caso contrario. Para medir esto de una forma interpretable, lo que se hace es medir la tasa de *verdaderos positivos* y *falsos positivos* que se logra usando estos vectores con distinto umbral de distancia, para la tarea de clasificación (cuando la distancia entre dos vectores es menor al umbral, se considera que pertenecen al mismo hablante). Ahora bien, para obtener una métrica que no dependa del parámetro de umbral elegido, suele medirse el área bajo la curva ROC, o sencillamente tomar el punto donde la tasa de *verdaderos positivos* y *falsos positivos* coinciden (se debe encontrar el valor de umbral de distancia que genera este punto) y reportar este valor, conocido como EER.

1.4. Algunos recursos disponibles

Para profundizar sobre este tema y otros relacionados al procesamiento automático de voz, se listan algunos recursos que pueden ser de utilidad:

Awesome Diarization [1]: Repositorio GitHub con una gran recopilación de recursos de dia rización (mantenida por el autor de varias de las charlas y sistemas mencionados en este trabajo como [5], [7], [8] y [9]).

SpeechBrain [2]: Biblioteca de software basada en Pytorch y Torchaudio [10], que permite cal cular características de audio y utilizar diversos modelos pre-entrenados para representación de voz.

pyannote.audio [3]: Biblioteca de software muy utilizada para evaluar sistemas de diarización (entre otros), que implementa la asignación óptima de hablantes utilizando el algoritmo húngaro u otros métodos iterativos más rápidos pero menos precisos.

Resemblyzer [4]: Biblioteca con algunos modelos pre-entrenados también para extraer caracte rísticas para reconocimiento de voz.

s4d docs y [5] *pypi* [6]: SIDEKIT for Diarization, es una biblioteca para python, enfocada en proveer herramientas para diarización de audio

Google uis-rnn [7]: Sistema de diarización completamente supervisado para hacer predicción online y no necesita clustering.

VoxCeleb Speaker Recognition Challenge 2021 [8] y 2022 [9]: Últimos desafíos para verificación de hablantes y diarización sobre el dataset VoxCeleb. También hay links a ediciones anteriores de este desafío.

DIHARD Challenge III [10]: Último desafío en enero de 2021, especialmente para casos de uso donde el estado del arte actual no funciona bien. En el link a la tercera edición también se referencian las ediciones anteriores.

[1] https://github.com/wq2012/awesome-diarization
[2] https://speechbrain.github.io
[3] https://pyannote.github.io/pyannote-metrics/reference.html#diarization
[4] https://github.com/resemble-ai/Resemblyzer
[5] https://projets-lium.univ-lemans.fr/s4d/
[6] https://pypi.org/project/s4d/
[7] https://github.com/google/uis-rnn
[8] https://www.robots.ox.ac.uk/~vgg/data/voxceleb/competition2021.html
[9] http://mm.kaist.ac.kr/datasets/voxceleb/voxsrc
[10] https://sat.nist.gov/dihard3

2. Implementaciones de referencia

En la figura 3 se puede ver un *pipeline* de diarización bastante típico, con algunos componentes opcionales y distintas configuraciones posibles.

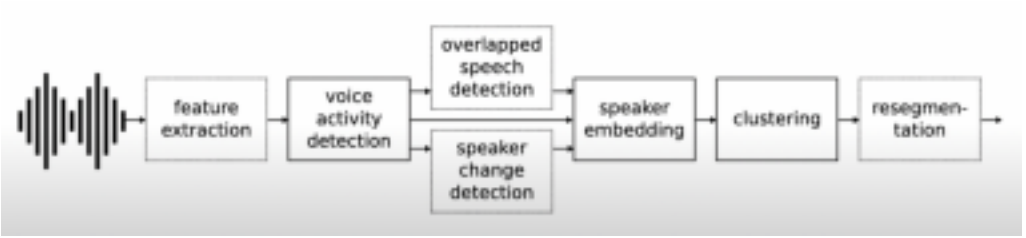


Figura 3: Diagrama (*pipeline*) con los componentes de un sistema de diarización. Imagen: [11]

En las siguientes sub-secciones se detalla la función de cada uno de estos bloques.

2.1. Características de audio

El primer bloque de *feature extraction*, es la extracción de características básicas de audio. En general no se utiliza directamente la forma de onda de la señal, sino que se utiliza alguna representación de tiempo-frecuencia como el espectrograma (módulo de la transformada de Fourier de tiempo corto o STFT), las bandas de frecuencias Mel, o basados en lo que se denomina el *Cepstrum* de la señal, como ser los coeficientes cepstrales (*Mel-Frequency Cepstral Coefficientes* o MFCC), que es una transformada adicional sobre las bandas Mel que permite obtener información de las componentes armónicas de la señal [12]. Las ventanas de tiempo que se utilizan usualmente en esta etapa, tienen una duración de entre 20-30ms. El compromiso en este caso viene dado por el principio de incertidumbre para señales, que contrapone la resolución en frecuencia (para lo que se necesitan ventanas largas de forma de incluir más ciclos de la señal en cualquier frecuencia de interés) con la resolución temporal (que requiere ventanas cortas que reaccionen rápidamente a cualquier cambio transitorio en la señal) (ver [13]).

2.2. Detección de voz, superposición y cambios de hablante

El segundo bloque de la figura 3 (*Voice Activity Detector* o VAD) detecta actividad de voz para evitar procesar los segmentos donde hay otros tipos de sonido como ser ruidos, música, etc. Aunque no es lo más habitual, existen algunos sistemas que no utilizan un módulo especializado de VAD, sino que calculan los *embeddings* para todos los segmentos y eventualmente descartan algunos haciendo algún análisis posterior.
Luego del VAD, existen más variantes en distintos sistemas. Puede existir un módulo de detec ción de *Overlap* (superposición de voces), ya sea para ignorar estos segmentos o para procesarlos de alguna forma particular. También puede existir opcionalmente el módulo de segmentación que detecta los potenciales cambios de hablante (*speaker change detection*). Finalmente, otros siste mas no utilizan ninguno de estos bloques opcionales y en cambio calculan los *speaker embeddings* sobre todos los segmentos donde se detectaron voces, y con ellos es suficiente para detectar los cambios de hablante o la superposición (habitualmente, simplemente se ignora el problema de la superposición).

2.3. Características de voz

El bloque de *speaker embeddings* en la figura 3, transforma las características de audio disponi bles en ese punto, a otra representación vectorial que idealmente sólo debe variar cuando cambia el hablante. Es decir que debería ser invariante a los distintos sonidos del idioma, y permanecer a distancias pequeñas de cualquier otro segmento de voz que pertenezca a la misma persona. En cambio, debería tener una distancia grande con cualquier fragmento pronunciado por otro locu tor, incluso ante la eventualidad de que las palabras pronunciadas sean las mismas. Como ya se mencionó, esta es la gran diferencia fundamental con los sistemas de transcripción de voz a texto, donde en cambio se busca la invarianza ante el cambio de locutor y máxima distinción entre los distintos sonidos del idioma.
Los fragmentos o ventanas móviles sobre los que se calculan estos embeddings, usualmente

duran entre 0.5 y 3 segundos, y existe un compromiso entre usar ventanas cortas para tener una buena resolución temporal al detectar más rápidamente el cambio de hablante, o ventanas más largas para generar un buen contexto de algunos segundos de audio que permitan caracterizar la voz y distinguirla correctamente. Intuitivamente, se puede comparar esto con el tiempo que le lleva a un humano distinguir cierta voz. Es importante entender que para el cálculo de cada embedding, sólo se toma como entrada un cierto fragmento de audio, y no se consideran relevantes en este punto los segmentos anteriores o posteriores. Es decir, sólo se aprovecha la información temporal interna al segmento, que comprende a varias ventanas móviles de *features* de audio provenientes del primer bloque, y son procesadas secuencialmente en orden.

2.4. Clustering

Finalmente, en la etapa de *clustering* se toman todos los vectores de *embeddings* de cada segmento de audio, y se busca agruparlos de forma de generar grupos o *clusters* que contengan a los segmentos de un mismo hablante, ya sean consecutivos en el tiempo o no. Idealmente, el número de clusters debe coincidir con el número de personas que participaron en la conversación. Y el propósito de este paso es justamente encontrar un criterio de cercanía que permita decidir cuáles *embeddings* (y sus correspondientes segmentos de audio) pertenecen a un mismo hablante.
Si los *embeddings* fueran ideales, podría utilizarse simplemente un algoritmo como *k-means* que agrupa en función de la distancia entre vectores, y el sistema debería ser capaz de generar los grupos correctamente. Sin embargo, en ese caso no se estaría utilizando la información de la localización de cada fragmento en el tiempo. No habría especial consideración por dos segmentos consecutivos de audio, por ejemplo. Sin embargo, sería bueno tener cierto sesgo estructural para que dos segmentos consecutivos tengan una probabilidad algo mayor de pertenecer a un mismo grupo (mismo hablante), comparado con segmentos lejanos y aislados que tuvieran la misma distancia vectorial. La razón es que esto es lo que se observa en las conversaciones reales, donde cada hablante suele participar durante varios fragmentos (varios segundos) de corrido, para luego ceder el turno a otra persona. Es una forma de hacer el sistema más robusto ante ruidos espurios, o segmentos con detecciones pobres y dudosas. Para generar esa regularidad, usualmente se utilizan técnicas de clustering espectral, sobre una matriz de afinidad entre *embeddings* que además se regulariza previamente como se puede visualizar en la imagen 4.
La matriz regularizada es mucho más robusta a valores espurios que generarían cambios bruscos de hablante. Los segmentos bien definidos que pertenecen a un mismo hablante, se visualizan como bloques oscuros bien delimitados en la matriz, que indican zonas de alta similitud entre embeddings contiguos.

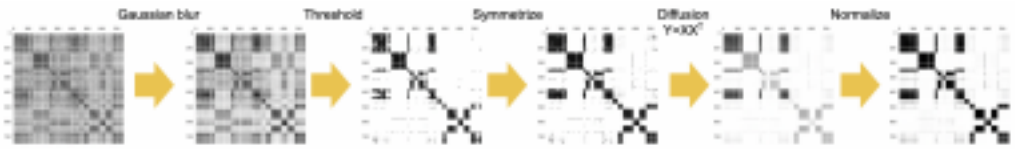


Figura 4: Refinamientos aplicados sobre la matriz de afinidad o similaridad de *embeddings*. Ima gen: [5]

El clustering espectral consiste finalmente en diagonalizar esta matriz utilizando los valores propios. Como la matriz de afinidad es simétrica, dicha descomposición es de la forma:

$$S = P \cdot D \cdot P^T (2)$$

Donde D es una matriz diagonal con los valores propios, y las filas de P (o columnas de P^T) son los vectores propios asociados a esos valores. Si suponemos que $S \in M^{n \times n}$ (donde n es el número de fragmentos de audio sobre los que se calcularon embeddings), la diagonalización

da como resultado tres matrices $n \times n$ también.
Sin embargo, se pueden usar sólo un conjunto $m < n$ de valores propios y eliminar las filas y columnas correspondientes, para obtener $P_r \in M^{n \times m}$, $P^T_r \in M^{m \times n}$ y $D_r \in M^{m \times m}$, de forma que al multiplicarlos, la matriz resultante S_r sigue siendo de dimensión n, y es cada vez más similar a S a medida que $n \longrightarrow m$. Filtrando cierto número m de valores propios altos, se obtiene una aproximación de S (tan buena como se desee), pero más importante, se obtiene una matriz $P_r \in M^{n \times m}$ donde a cada uno de los n segmentos de audio le corresponde una fila de la matriz, con un embedding de dimensión reducida m, que tiene las componentes principales del embedding original. Todo este procedimiento se puede ver como una forma de PCA utilizando la matriz de afinidad como matriz de covarianzas.
Finalmente, se puede utilizar un algoritmo de clustering como *k-means* para agrupar los embed dings de dimensión reducida. Lo esencial de este procedimiento de suavizado y clustering espectral, es que se introduce un sesgo estructural en el sistema, favoreciendo que los embeddings cercanos en el tiempo pertenezcan al mismo clúster (o hablante), y a su vez se eliminan las componentes de los vectores que no aportan demasiado a la separación de las voces que están en juego.

2.5. Otras variantes

El sistema desarrollado por Google llamado *UIS-RNN (Unbounded Interleaved-State Recurrent Neural Network)* [9], no necesita utilizar clustering *offline* (luego de tener todos los embeddings de audio precalculados), sino que permite predecir en tiempo real utilizando un modelo probabilístico que incluye un proceso estocástico de *restaurante chino*, donde las probabilidades de cada segmento de pertenecer a un hablante anterior, dependen de la cantidad de tiempo que dicha persona haya participado en la conversación.
De esta forma, permite hacer diarización *online*, en tiempo real a medida que se obtiene el audio de la conversación. Algunos de los involucrados son los mismos autores del paper [8], y afirman romper el estado del arte en diarización que era marcado por ese sistema. Aunque el código del sistema es libre, los datos de entrenamiento o el sistema pre-entrenado no están disponibles para uso público.

Referencias

[1] Hao Tang, S. M. Chu, M. Hasegawa-Johnson, and T. S. Huang, “Partially Supervised Speaker Clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 959–971, May 2012. [Online]. Available: http://ieeexplore.ieee.org/document/5989833/

[2] Church et al., “DIHARD Challenge,” 2018. [Online]. Available: https://dihardchallenge.github.io/dihard1/overview.html

[3] Brown et al., “VoxCeleb Speaker Recognition Challenge,” 2021. [Online]. Available: https://www.robots.ox.ac.uk/~vgg/data/voxceleb/competition2021.html

[4] H. Bredin, “pyannote.metrics: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems,” in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, Stockholm, Sweden, Aug. 2017. [Online]. Available: http://pyannote.github.io/pyannote-metrics

[5] Q. Wang, “Google’s Diarization System: Speaker Diarization with LSTM,” ICASSP 2018, 2018. [Online]. Available: https://www.youtube.com/watch?v=pjxGPZQeeO4

[6] “The Second DIHARD Diarization Challenge: Dataset, task, and baselines,” Jun. 2019, arXiv:1906.07839 [cs, eess]. [Online]. Available: http://arxiv.org/abs/1906.07839

[7] “Generalized End-to-End Loss for Speaker Verification,” Nov. 2020, arXiv:1710.10467 [cs,

eess, stat]. [Online]. Available: http://arxiv.org/abs/1710.10467

[8] “Speaker Diarization with LSTM,” Jan. 2022, arXiv:1710.10468 [cs, eess, stat]. [Online]. Available: http://arxiv.org/abs/1710.10468

[9] “Fully Supervised Speaker Diarization,” Feb. 2019, arXiv:1810.04719 [cs, eess, stat]. [Online]. Available: http://arxiv.org/abs/1810.04719

[10] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi, “TorchAudio: Building Blocks for Audio and Speech Processing,” arXiv, Tech. Rep. arXiv:2110.15018, Feb. 2022, arXiv:2110.15018 [cs, eess] type: article. [Online]. Available: http://arxiv.org/abs/2110.15018

[11] H. Bredin, “pyannote audio: neural building blocks for speaker diarization,” Apr. 2020. [Online]. Available: https://www.youtube.com/watch?v=37R_R82lfwA

[12] L. R. Rabiner and R. W. Schafer, *Theory and applications of digital speech processing*, 1st ed. Upper Saddle River: Pearson, 2011, oCLC: ocn476834107.

[13] L. Cohen, *Time-frequency analysis*, ser. Prentice Hall signal processing series. Englewood Cliffs, N.J: Prentice Hall PTR, 1995.

Reporte 3

Análisis de pautas de observación de clases

Autor: Germán Capdehourat

El objetivo de este informe es resumir el análisis realizado de las pautas de observación de clases utilizadas en los programas educativos de Ceibal, Pensamiento Computacional y Ceibal en Inglés. En ambos casos, lo que se busca es identificar aquellos puntos más relevantes a la hora de analizar una clase, que a la vez presentan mayor factibilidad de ser abordados mediante algún procesamiento automatizado de las grabaciones de las clases.

Ceibal en inglés

El programa Ceibal en inglés, se trata de una iniciativa que comenzó en Ceibal hace más de diez años para apoyar la enseñanza de inglés en primaria (hoy también extendido a media). La propuesta innovadora contempla profesoras/es remotas/os que pueden estar en Uruguay o en el exterior, y dan su clase a través de equipos de videoconferencia iguales a los instalados en las escuelas. Desde sus comienzos, el programa tuvo un fuerte énfasis en la evaluación de las prácticas pedagógicas y el desarrollo de las actividades en clase. En particular hay un proceso de evaluación y monitoreo de la calidad de las clases (ver Figura 1), mediante observaciones de aula siguiendo una pauta específica (ver Pauta CEI 2022).

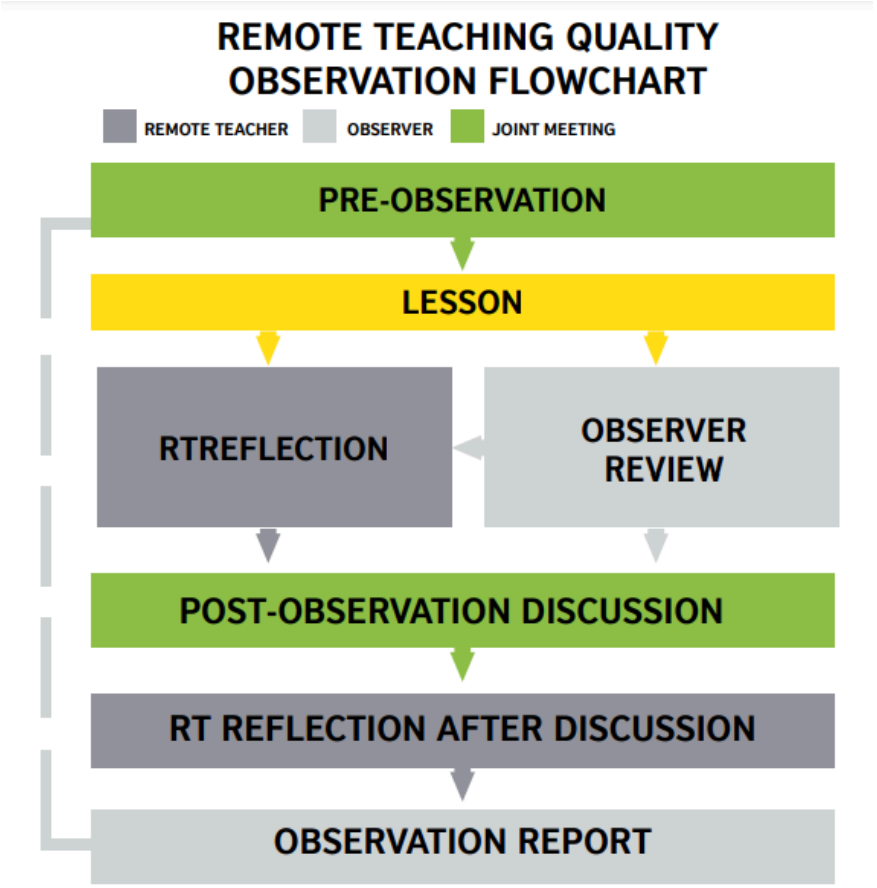


Figura 1: Quality Management of Ceibal en Inglés, Graham Stanley, Gonzalo Negron y David Lind, capítulo de Innovations in education Remote teaching, British Council y Ceibal, 2019.

Como se puede ver en la Figura 1, el proceso tiene varias etapas. En particular existen formularios que se completan previo a la clase (*pre-observation*) y luego de la clase (*post-observation*), así como una reflexión final luego del intercambio con el docente.
En este caso la observación de las clases normalmente se hace en vivo, ya que el observador se conecta a la videollamada o está presente en el aula, pero sin participar. Durante la pandemia se utilizaron también grabaciones, pero ahora con el retorno a la presencialidad plena esto no es lo habitual, salvo para casos puntuales. Esta forma de trabajo lleva a que el observador deba tomar notas durante la clase, de forma de poder registrar los aspectos a tener en cuenta luego para completar la pauta de observación. La práctica habitual de los observadores (denominados *quality managers* en el programa Ceibal en inglés) suele ser transcribir y comentar lo que dice y hace el docente, y también eventualmente los alumnos, así como registrar los tiempos en que suceden las diferentes cosas o comentarios sobre las diapositivas que se están compartiendo.

Pensamiento computacional

En el caso del programa Pensamiento Computacional, se aprovechó la infraestructura de videoconferencia y la experiencia positiva del programa Ceibal en inglés para abordar otra temática, pero replicando el modelo de clases a distancia con docentes remotos. Se trata de un programa educativo más nuevo, por lo que el trabajo en observación de clases en este caso es también más reciente. Su puesta en práctica se vió afectada por la pandemia, ya que los

procesos de evaluación y monitoreo aún no se realizaban de manera estandarizada previo a la pandemia.

En tal sentido, existe una primer instrumento de observación de clases que fue utilizado en 2021 (Instrumento PC 2021). En este caso, como se puede observar en la pauta se trata de un análisis bastante pormenorizado de la actividad durante el desarrollo de la clase. Este instrumento se divide en cuatro grandes áreas, Compromiso, Sensibilidad/Responsabilidad, Aprendizaje Profundo y Aspectos Generales, donde cada una tiene varios puntos que son evaluados por quien realiza el análisis. De la experiencia en el uso del instrumento por parte de los evaluadores expertos se destacan los siguientes puntos:

- La aplicación del instrumento le toma aproximadamente 4hs al observador, por cada hora de clase analizada.

- La práctica habitual es hacerlo mediante grabaciones, lo cual permite ir para adelante y para atrás en la clase, algo necesario al ir avanzando en los distintos puntos del instrumento, que cubren aspectos muy diversos.

Para el 2022, y con la vuelta masiva a la presencialidad sin interrupciones, también evolucionó el proceso de evaluación y monitoreo planteado para este año. En este caso se generó un nuevo instrumento (Instrumento PC 2022) que a diferencia del anterior es más simple y aborda un número menor de puntos. Esto permite que la observación se haga en un tiempo menor, y sea también más fácil de entrenar a observadores para realizar la tarea. La metodología planteada para 2022 implica que a cada docente se lo evalúa primero usando este instrumento. En caso de que la observación arroje resultados insatisfactorios, en ese caso se realizan nuevas instancias de observación, en las que sí se usa la pauta de observación anterior. Por lo tanto, podemos decir que esta etapa funciona a modo de screening primario, ya que al detectar problemas o deficiencias que requieren mayor análisis, luego en la siguiente etapa se hace una observación más detallada con el instrumento previo.

Otras pautas de observación de clases

La observación de clases es un tema de estudio en la educación desde hace muchos años, con una vasta literatura que se refleja en diversos artículos y libros publicados [CLASSOBS 1975, CLASSOBS 1990, CLASSOBS 2004, CLASSOBS 2012, CLASSOBS 2013, CLASSOBS 2014, CLASSOBS 2016, CLASSOBS 2020]. En tal sentido, existen diversas propuestas para estandarizar las pautas de observación de clases [ISI 2009, CLASS 2010, COPUS 2013]. Por ejemplo, el protocolo de observación CLASS (Classroom Assessment Scoring System) contempla clases desde prescolares hasta el último grado de secundaria, en base a cuatro ciclos de observación de 15 minutos (ver referencia por más detalles). Por otro lado, la propuesta de COPUS (Classroom Observation Protocol for Undergraduate STEM) se enfoca específicamente en clases de actividades de CTIM (ciencia, tecnología, ingeniería y matemáticas, conocido también como STEM por sus siglas en inglés). En tal sentido, esta última está enfocada en programas como el de Pensamiento Computacional de Plan Ceibal.

Al igual que para los instrumentos de observación de clases de los programas educativos de Ceibal comentados previamente, en todas estas pautas propuestas en la literatura y utilizadas en distintas partes del mundo, el esfuerzo que se requiere para aplicarlas sigue siendo muy

grande. El trabajo manual y pormenorizado necesario implica una dedicación en cuanto a tiempo y recursos que hace muy difícil su implementación masiva. Por lo tanto, esto lleva a que la solución no escale y se trabaje habitualmente aplicando las pautas en base a muestreos (es decir que se observan solamente algunas clases de cada docente) o con etapas previas de screening para simplificar la tarea. Dada esta situación, es natural que en los últimos años se haya empezado a estudiar la posibilidad de automatizar al menos en forma parcial dicho proceso, mediante el procesamiento de grabaciones de audio y/o video de las clases.

Indicadores más factibles de automatizar

A partir de las pautas de observación, el objetivo es encontrar indicadores que a priori sean más factibles de implementar mediante el análisis computarizado de las grabaciones de las clases. Se busca identificar aquellos que sean viables usando técnicas del estado del arte de procesamiento de audio, mientras que al mismo tiempo generen un aporte significativo en el análisis de la práctica docente y el desarrollo de la clase.

En particular, una de las métricas que está previsto sea posible automatizar en el marco del proyecto es el tiempo que habla el docente durante la clase. Este parámetro puede dar mucha información, considerando las nuevas prácticas pedagógicas que fomentan la comunicación y el intercambio, tanto entre el docente y los estudiantes, así como entre los estudiantes. En el siguiente artículo se discute sobre este indicador, denominado Teacher talking time (TTT) en el contexto de clases como las de Ceibal en Inglés. En este caso un TTT menor podría indicar mayor tiempo de clase disponible para la participación de los estudiantes, lo que implica practicar la oralidad de la lengua extranjera. Algo similar ocurre en clases y experiencias vinculadas a temas de STEAM, como podría ser el caso de Pensamiento Computacional, donde las prácticas pedagógicas más aceptadas en la actualidad valoran mucho el espacio dedicado al intercambio entre los estudiantes y el trabajo en equipo, por ejemplo para discusión y búsqueda de soluciones de forma colectiva.

La identificación de los momentos de clase en los que habla el docente, permitirían no solamente tener este indicador del tiempo total que habla durante toda la clase, sino además detectar en qué momentos hay largas exposiciones donde predomina el docente hablando, y en qué casos ocurre lo opuesto, es decir que hay mayor participación con estudiantes o intercambio con el docente y entre ellos. A continuación se resume una lista de puntos que podrían abordarse con técnicas del estado del arte actual del procesamiento de audio, ordenadas desde mayor nivel de factibilidad a priori y con mayor impacto a la hora de incorporarlas en un proceso de evaluación y monitoreo semi-automatizado:

- Detectar todos los momentos en que habla el docente principal (remoto por videoconferencia). Medir el tiempo total y la distribución de sus participaciones durante la clase.
- Detectar todos los momentos en que hablan los estudiantes. Identificar aquellos casos donde se solapan las voces (ej. respuestas a preguntas del docente).
- Detectar participaciones del docente de aula, en aquellos casos que esto ocurra.
- Identificar en qué lengua se habla. Esto aplica solamente al caso de Ceibal en inglés, y es tanto para el docente como para los estudiantes.
- Detectar problemas técnicos, como saturación, ruidos de micrófono, sonido muy bajo.
- Detectar niveles de ruido ambiente altos, debido a desorden en clase, gritos, conversaciones de fondo, voces mezcladas, ruidos de golpes en mesas y sillas.

- Detectar palabras de uso frecuente[1], como por ejemplo las que usa el docente para responder a los estudiantes (ej. “Excelente”, “Muy bien”, “Perfecto”, “Excellent”, “Very good”, “Well done”).
- Identificar la cantidad de estudiantes distintos que participan en clase, cuantificar la distribución de los tiempos de las participaciones y detectar si existen sesgos (ej. por género).
- Identificar si los estudiantes hablan entre sí o si hablan con el docente.
- Detectar el uso de preguntas por parte del docente, en base al tono de voz.

Conclusiones y próximos pasos

La observación de clases es una herramienta fundamental para la evaluación y monitoreo continuo de los programas educativos y las prácticas docente. Como vimos anteriormente, los instrumentos que se utilizan para la observación de clases requieren un tiempo y trabajo manual pormenorizado, lo que dificulta la escalabilidad para su uso masivo.
Del análisis de las pautas específicas que se usan en Ceibal para los programas Ceibal en Inglés y Pensamiento Computacional, surgen indicadores relevantes que se podrían implementar mediante el procesamiento automatizado del análisis de las clases.
En base a este primer análisis, el objetivo ahora es definir un adecuado etiquetado de los datos disponibles, lo que será el insumo fundamental para las siguientes etapas del proyecto: por un lado el desarrollo de algoritmos y por otro lado su evaluación de desempeño correspondiente.

Reporte 4

Análisis de clases - Detección del idioma

Autor: Emilio Martínez

El objetivo en este caso es detectar el idioma que se habla en clase. Esto es de particular interés para las clases de Ceibal en inglés, donde el objetivo buscado es maximizar el tiempo que se habla en inglés durante la clase, pero suele haber tramos en que se habla en español. Además, el resultado de este análisis se podría combinar con el de la detección de quién está hablando, y de esta forma identificar si los tramos que se habla en español corresponden al docente o a los alumnos.

Procesamiento para detectar el idioma

A partir de la grabación de la clase, el procesamiento para detectar el idioma consiste de los siguientes pasos:

- Conversión del video a audio. Este paso es para utilizar solamente la pista de audio en el procesamiento posterior de la señal.

[1] Para Ceibal en inglés, algunas frases de interés son: “pay attention”, “listen and repeat”, “work in pairs”, “what did we learn?”, “repeat after me”; así como otras usadas habitualmente para dar feedback a los estudiantes: “let´s check”, “correct or incorrect?”, “well done”, “excellent”, “very good”.

- Segmentación del audio según el nivel de potencia de la señal. Esto permite considerar únicamente los tramos donde haya voces hablando. Es lo que se conoce por su sigla en inglés como VAD: Voice Activity Detector.

- Procesamiento mediante whisper de cada tramo de audio. Esta herramienta, habitualmente utilizada para transcripción de audio a texto, incluye también un detector de idioma. Este bloque genera distintas probabilidades para los idiomas posibles en cada tramo del audio.

- En base a las probabilidades obtenidas con whisper, se indica el idioma de cada tramo como aquel más probable según el análisis realizado.

Vale destacar que las probabilidades permiten, además de detectar el idioma más probable, saber en cada caso si estamos ante una situación bastante segura en la detección (ej. la probabilidad de que el idioma sea inglés es mucho mayor a la de español) o si se trata de un caso dudoso (los valores de las probabilidades son similares entre sí). Esto nos permitiría elegir la forma más adecuada para presentar los resultados y analizando distintos casos decidir si conviene dar una respuesta segura o si es preferible indicar que se trata de un tramo dudoso.

Resultados y comentarios finales

A continuación se muestra un ejemplo de los resultados obtenidos. En este caso se puede ver para cada instante de tiempo el idioma detectado en la barra superior, mientras que en la barra inferior se muestra el porcentaje de cada idioma considerando una ventana móvil de 30 segundos. Si bien todavía no se hizo una evaluación profunda del desempeño de la detección de idioma, se puede ver que los resultados son bastante alentadores. Además, este análisis se puede combinar con la detección del hablante y de esa forma analizar por separado el idioma para el docente y los alumnos. También resulta algo sencillo pensar en indicadores globales para todo el desarrollo de la clase, como el porcentaje de cada idioma. Algo adicional para explorar es usar la probabilidad de detección del inglés como indicador de qué tan buena es la pronunciación, algo que podría servir para analizar a los docentes.

Reporte 5

Análisis automático de clases grabadas Ceibal - Características clásicas de audio

Autor: Emilio Martínez

Introducción

En el marco del proyecto FMV de la ANII, se mostró en éste documento un primer análisis de características (temporales, espectrales y de aprendizaje profundo) aplicadas al audio de un grupo reducido de hablantes en una clase. Debido a los prometedores resultados de diarización y detección de hablante obtenidos mediante una arquitectura de aprendizaje automático, que en principio son difíciles de interpretar, se elige tomar un enfoque en paralelo a lo mencionado utilizando herramientas clásicas de procesamiento de señales de audio. Estas herramientas clásicas, de más bajo nivel, permiten visualizar el comportamiento del habla y servirán como respaldo de interpretación para los casos en el que el algoritmo principal de aprendizaje automático falle (como pueden ser, por ejemplo, errores sistemáticos de confusión de hablantes de la clase).

Datos

Para analizar las características del audio de cada hablante se usaron dos clases grabadas por Ceibal y etiquetadas por el Grupo de Procesamiento de Audio de la Facultad de Ingeniería (GPA). Las clases corresponden a los cursos de Ceibal en Inglés (CEI) y Pensamiento Computacional (PC). Para etiquetar estas clases se usaron etiquetas de alto nivel, pero para este análisis se agruparon en las etiquetas “p” y ”a” mencionadas en la sección de Introducción del documento anterior.

Características del audio para la voz hablada

Para el análisis en frecuencia de una señal de audio suele usarse espectrogramas, donde se muestra para cada instante de tiempo las componentes de frecuencia (medida en Hz) que representan la presencia de sonidos graves (frecuencias bajas), medios y agudos (frecuencias altas).

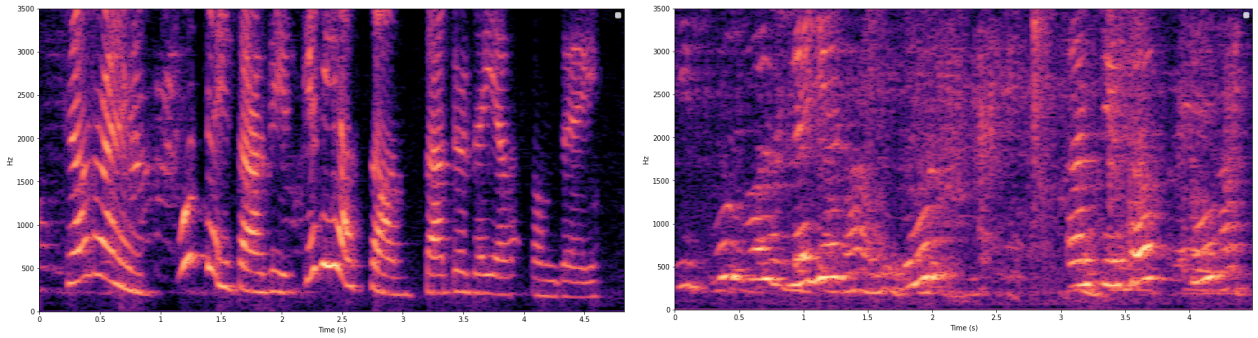


Figura 1: espectrograma de voces (izq. docente, der. alumno)

Para la voz hablada y sonora, la frecuencia predominante, o también llamada frecuencia fundamental o f_0 , suele estar entre 80 Hz y 300 Hz. Considerando que en las grabaciones de las clases analizadas tienen presencia de niños, es normal pensar que el valor de f_0 pueda elevarse por encima del rango mencionado, dependiendo del momento y el ambiente de la clase.

En la Figura 1, se ve que para la voz del docente en cada instante de tiempo, hay repeticiones de la frecuencia fundamental en formas de múltiplos (armónicos), donde no son del todo notorias comparando con la voz del alumno. Este resulta en un espectrograma “menos limpio” que el del docente. Esto se puede deber a factores como pueden ser: distancia del alumno al micrófono (se mezcla la voz con el ruido de ambiente), voces superpuestas con otros alumnos, entre otros factores.

También existen las frecuencias formantes o formantes, que representan cómo es modulada la voz por el tracto vocal con el uso de los fonemas durante el habla. Existen estudios como Vowel Acoustic Space Development in Children: A Synthesis of Acoustic and Anatomic Data que muestran la evolución de los valores típicos de para diferentes edades para las vocales, diferenciando en sexo masculino y femenino, evidenciando una clara separación de éstos valores entre niños, adolecentes y adultos.

Aplicación para distinción de hablante

Presentadas estas características ampliamente utilizadas en el ámbito clásico de procesamiento de señales de voz, se dispone evaluarlas para la clasificación de hablante para las clases de Pensamiento Computacional y Ceibal en Inglés. Para ello, se extraen la frecuencia fundamental (f_0) y las dos primeras formantes (F_1 y F_2). En la Figura 2 se puede apreciar un espectrograma donde se marcan éstas frecuencias en cuestión.

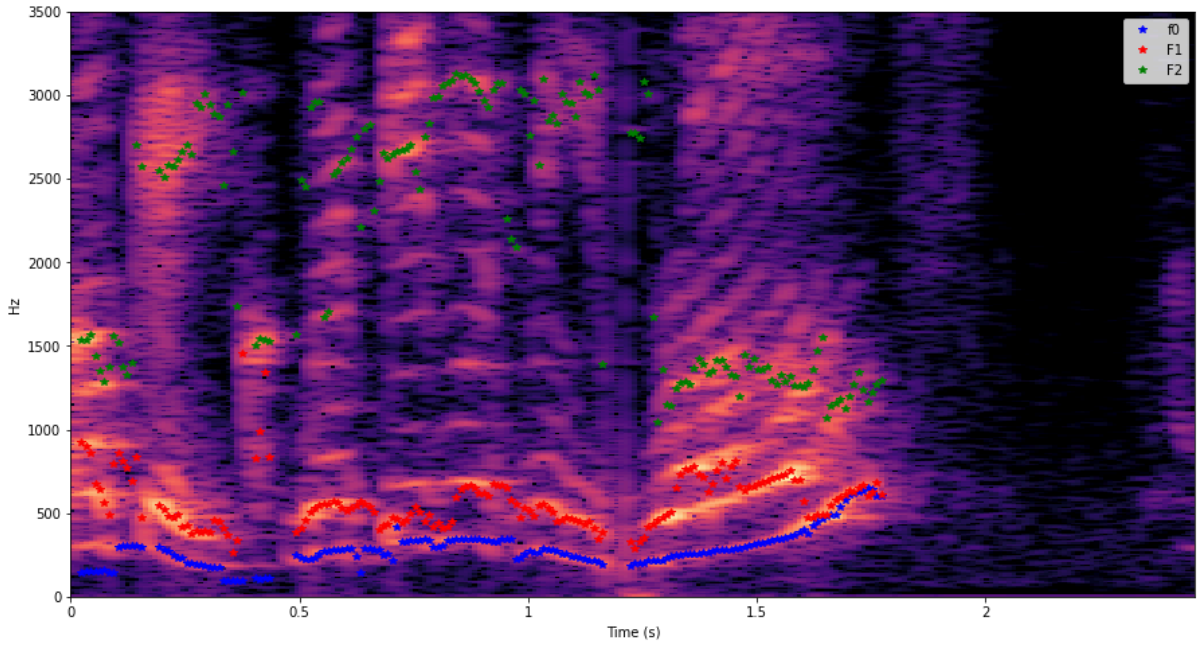


Figura 2: Representación de frecuencia fundamental (azul) y dos primeras formantes (rojo y verde respect.) junto al espectrograma de la voz del docente

Se procede a generar estadísticas con las características extraídas de los segmentos de audios etiquetados como “a” y “p”. Para ello se decide generar un histograma de f_0 y un mapa de densidades para F_1 y F_2 para cada hablante de la clase, tal como se observa en la Figura 3.

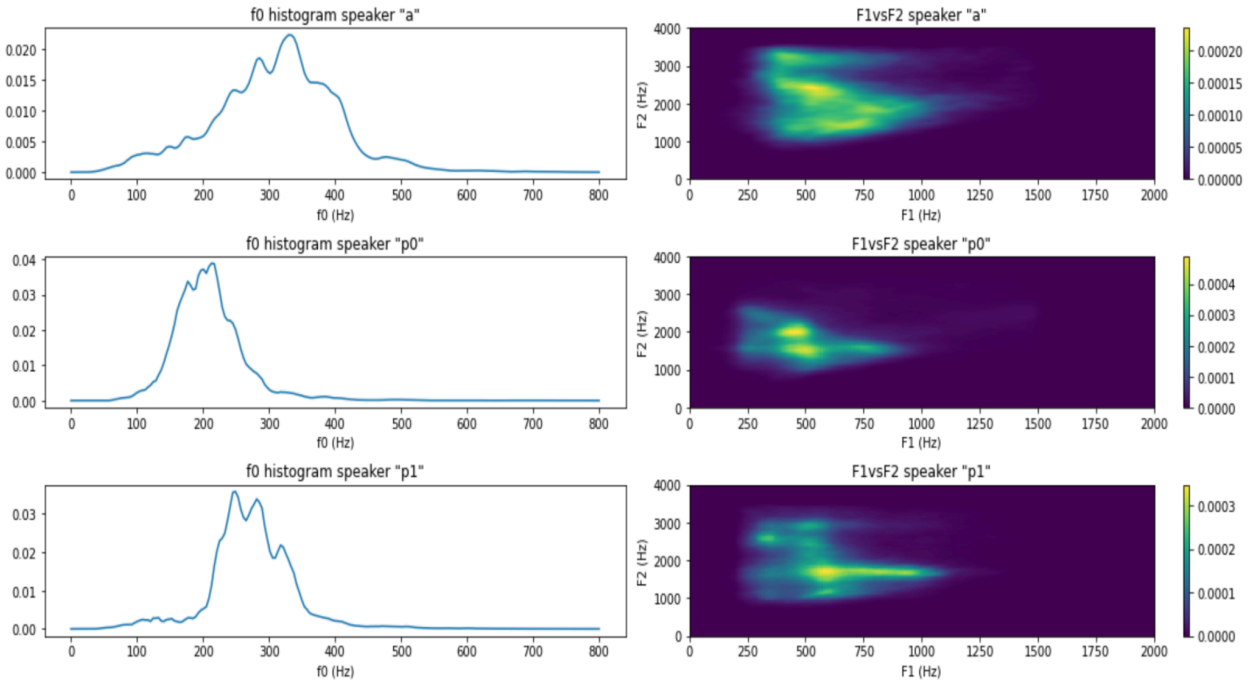


Figura 3: histogramas de f_0 y mapas de densidad de formantes para alumnos de varias clases ("a"), un docente de PC hombre ("p0") y una docente de PC mujer ("p1")

La figura anterior muestra claras distinciones tanto en el histograma como en el mapa de densidad para los diferentes hablantes. El histograma de frecuencia fundamental y mapa de formantes para los alumnos tienen una mayor dispersión en comparación a los demás, y en media tienen un valor de f_0 mayor al de los adultos. También se puede validar que las frecuencias fundamentales del docente masculino son en su mayoría menores a las de la docente femenina. Para las clases de Ceibal en Inglés se pueden ver resultados similares a los presentados de Pensamiento Computacional, se pueden consultar en éste Jupyter Notebook desde Google Colaboratory.

Una vez que se tienen estas estadísticas, un esquema simple de clasificación de hablante es el voto de las características frecuenciales ponderadas por el nivel de la estadística de cada hablante. Es decir, al tener una nueva muestra de audio para identificar si habla el docente o un alumno, se comparan f_0, F_1 y F_2 con cada histograma y mapa de densidad, donde la predicción elegida es la que tiene un mayor "parecido". Las Figuras 4 y 5 muestran un ejemplo de lo anterior clasificando satisfactoriamente la presencia del docente y el alumno respectivamente, marcando las características del audio sobre cada estadística.

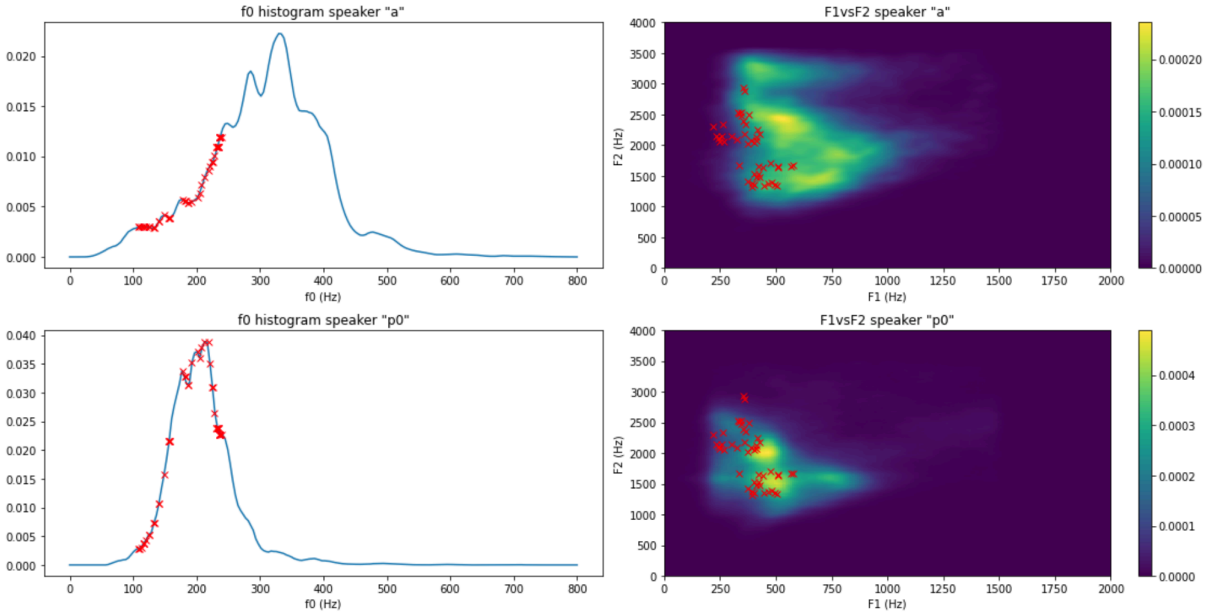


Figura 4: Predicción de docente de PC basada en histograma de frecuencia fundamental y mapa de densidad de formantes

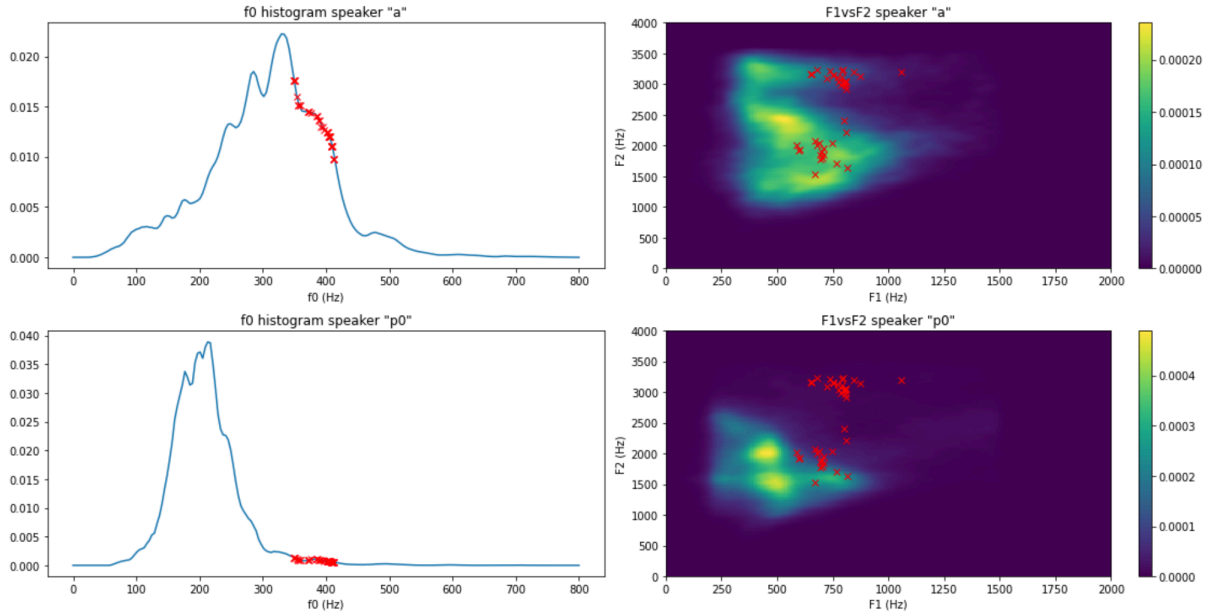


Figura 5: Predicción de alumno de PC basada en histograma de frecuencia fundamental y mapa de densidad de formantes

Resultados y posibles mejoras

En las Figuras 6 a-d se muestran los aciertos y las fallas del esquema de clasificación mediante diferentes clases. Se visualizan estos resultados usando matrices de confusión, donde se comparan las etiquetas reales (posicionadas en las filas) con las etiquetas de la predicción (columnas).

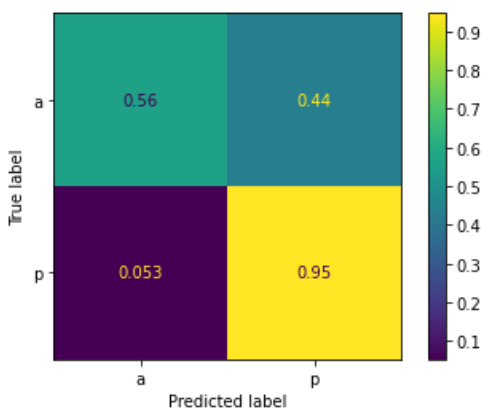


Figura 6.a: Desempeño en clase de PC 1

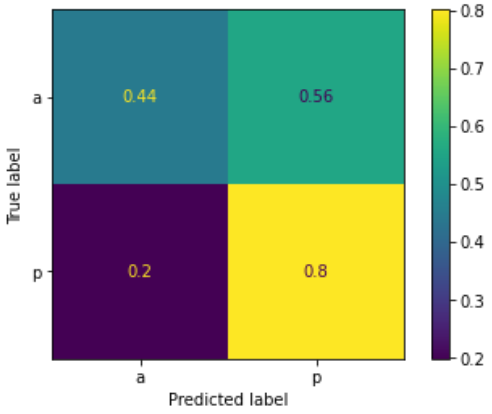
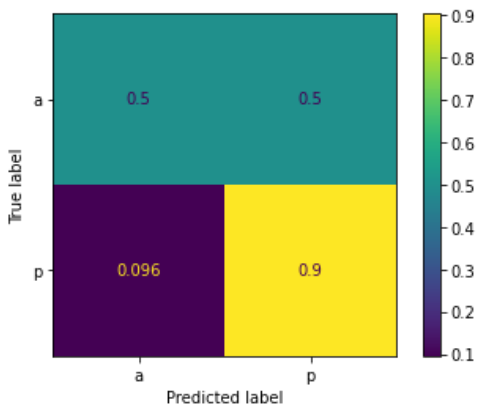


Figura 6.b: Desempeño en clase de PC 2



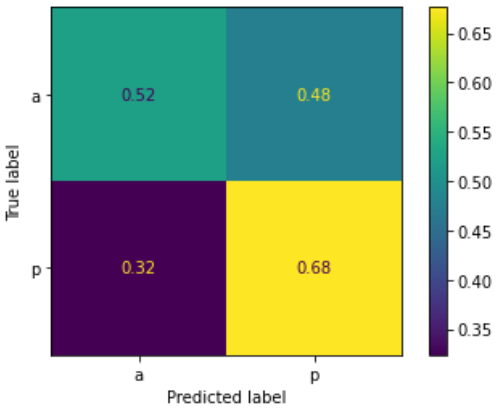


Figura 6.c: Desempeño en clase de CEI 1 *Figura 6.d: Desempeño en clase de CEI 2*

Es fácil ver que la mayoría de las predicciones se corresponden a la del profesor remoto o docente. Una excepción parece ser la clase de Ceibal en Inglés 2 que, además de clasificar mal menos de la mitad de los alumnos (como parece ser el caso de las demás clases), también hay una confusión considerable de la docente hacia los alumnos.

Otra observación es que la clase Pensamiento Computacional 1 muestra la mejor clasificación para el docente. Esto puede deberse a que corresponde a un docente masculino, que se pueden diferenciar fácilmente (tanto en frecuencia fundamental como en formantes) de los alumnos. Aún así, estos valores típicos parecen extenderse más de lo necesario ya que hay una confusión del 44% de los alumnos con el docente.

En base a lo observado surgen los siguientes puntos de mejora en el esquema planteado.

Se obtuvieron las características por el software “WaveSurfer”, que tiene la posibilidad de calcular f_0 , F_1 y F_2 en cada instante de tiempo. Pero en la práctica siempre existe una dependencia de la frecuencia fundamental para la obtención de las formantes que WaveSurfer no tiene en cuenta, y si esto no se tiene en cuenta puede haber un error en el cálculo de las formantes. Idealmente se requiere un cálculo de formantes adaptativo a la frecuencia fundamental (que está en actual desarrollo por el GPA).

Otra mejora puede ser en cuanto a la decisión y qué información da cada característica. Pueden haber regiones de la frecuencia fundamental o de las formantes que correspondan a un sólo tipo de hablante por la naturaleza de la voz hablada. Por ejemplo, si la frecuencia fundamental suele ser muy baja, es más probable que corresponda a un docente masculino, y se puede tener en cuenta ponderando ese voto para la clasificación.

Reporte 6

Análisis de transcripción de clase - Detección de frases predefinidas

Autor: Pablo Cancela

El objetivo en este caso es analizar lo que se habla en clase, para lo cual es necesario una primera etapa de conversión de audio a texto. Luego, una vez que se cuenta con la transcripción de la clase, se analiza el texto para la detección de palabras o frases de interés.

Gramática y vocabulario esperado en Ceibal en inglés

En este caso el análisis está enfocado en clases de Ceibal en inglés, programa que cuenta con una currícula para cada nivel (Level 1, Level 2, Level 3), la cual está acompañada de un vocabulario y gramática predeterminados.

Por ejemplo, si se observa la primera clase del nivel 1, allí se puede ver vocabulario específico para saludos (“Hello”, “thanks”), así como lenguaje esperado en la clase (“listen and repeat”, “look”, “read”, “write”, “work in pairs”, “pay attention”). También hay algunas frases como “How are you?”, “I’m fine, thanks, and you?”, “My name is _____” y “I am _____”. Como se puede ver en los últimos casos, hay frases donde puede haber cosas variables, tales como personajes o lugares.

Procesamiento y análisis de la grabación de la clase

Previo al procesamiento, es necesario definir la lista de palabras o frases que se van a buscar en la clase. Esto sería una entrada adicional, que podría incluso modificarse a gusto según lo que se desea detectar en cada clase específica.

A partir de la grabación de la clase, el procesamiento consiste de dos etapas:

- Transcripción: mediante la aplicación de whisper sobre toda la grabación, se genera un archivo de texto con la transcripción detectada por la herramienta en cada instante de la clase. Es importante notar que el resultado de whisper está separado en distintos intervalos de tiempo, lo cual será aprovechado en la etapa siguiente.

- Análisis del texto: se procesa el texto generado por whisper para cada intervalo de tiempo, analizando la ocurrencia de alguna de las palabras o frases incluidas en la lista de interés. Para ello se recorre el texto caracter a caracter, y usando la distancia de levenshtein (también llamada distancia de edición) se busca la probable ocurrencia de alguna de las frases predeterminadas.

Resultados y comentarios finales

En esta carpeta se muestra un ejemplo de los resultados obtenidos, analizando este video. En particular el archivo output contiene la lista de todas las detecciones, indicando el tiempo de inicio, el tiempo de fin y cuál fue la frase o palabra detectada.

Reporte 7

Detección de segmentos de audio pre-grabados

Autor: Pablo Cancela

Introducción

Algunas actividades que se desarrollan en las clases remotas de Ceibal en Inglés, involucran la reproducción de sonidos pre-grabados.

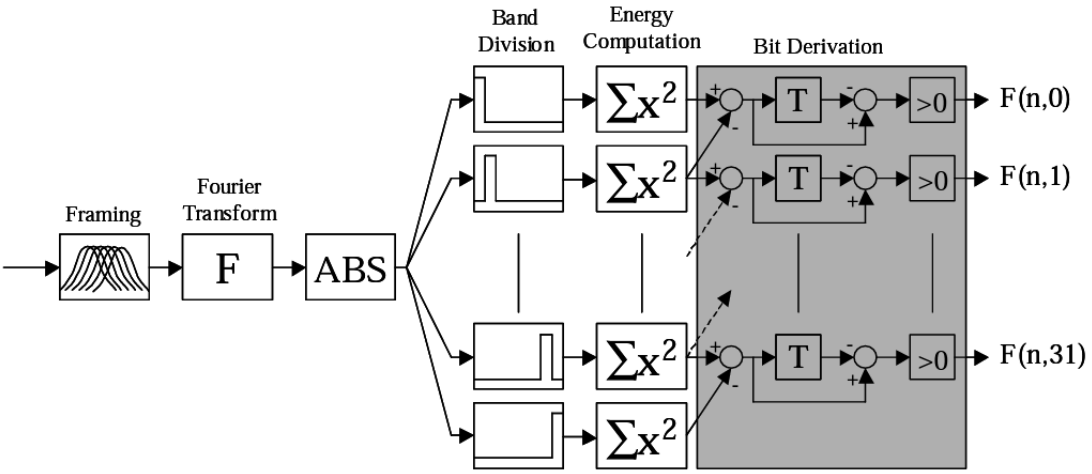
Se pueden diferenciar dos tipos de grabaciones:
1. Sonidos que se reproducen automáticamente al desarrollar una actividad (al abrir Little Bridge, cuando se selecciona “Correct”)
2. Grabaciones de frases o diálogos que son comúnmente utilizadas para ejercitar la capacidad de comprensión de Inglés hablado “Listening”.

La detección de estos audios en las grabaciones de las clases brinda información útil sobre el desarrollo de la clase. En cuanto al primer tipo de grabaciones, éstas pueden informar sobre si se utilizó o no el material didáctico. El segundo tipo de grabaciones permite identificar específicamente qué actividades de “Listening” fueron realizadas.

Dado que la plataforma que se utiliza tiene un conjunto de grabaciones acotado, predefinido, se plantea el desarrollo de un módulo que detecte en las grabaciones de las clases, los tiempos en que se reproducen archivos de audio de una base de datos predefinida.

Técnica utilizada

En este trabajo se utilizará un módulo basado en [1] que permite detectar de manera muy robusta segmentos de audio de al menos 3 segundos y de forma aceptable segmentos de 1 segundo a 3 segundos. La técnica se basa en *Audio Fingerprinting*, en el cual se extrae una huella a partir del audio original, produciendo para cada instante una palabra de 32 bits, con un paso entre instantes de aproximadamente 11,6 milisegundos. La ventana de análisis tiene 370 milisegundos de largo.



Así, la huella de un segmento de audio de D segundos tendrá una huella de largo aproximadamente D/0.0116 s, y con 4 bytes para cada “subhuella”. Los bits de la representación de cada instante se pueden considerar como una umbralización de la derivada discreta, teniendo en cuenta los cambios de energía tanto en el dominio del tiempo como en el dominio de bandas de frecuencias. Los rangos de las bandas frecuencias corresponden a 33 bandas con ancho proporcional a la frecuencia central, cubriendo el rango total de 300Hz a 2000Hz.

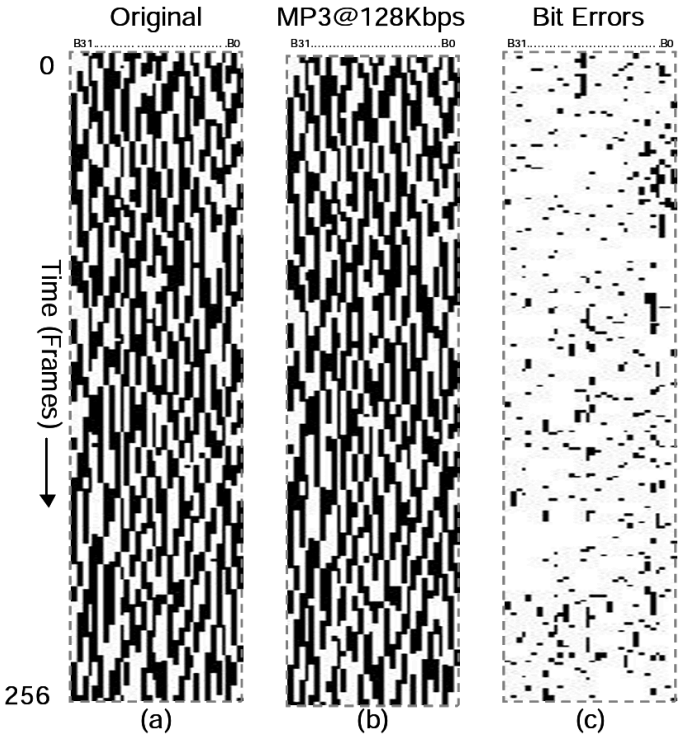
De esta manera se obtiene, para cada instante, una representación 32 bits que tiene la apariencia de tomar valores aleatorios (la probabilidad para cada bit de tomar el valor 0 o 1 es igual y la correlación entre los bits es muy baja). Asimismo, para el cálculo de estas palabras, se utilizan ventanas de análisis sensiblemente más largas que el paso de avance de 11,6 ms por lo que los valores de los bits de estas subhuellas no varían rápidamente sino que lo hacen lentamente (se prenden y apagan muy pocos bits entre dos subhuellas consecutivas).

Estas propiedades hacen que las huellas sean robustas a pequeños desfasajes temporales y asimismo los valores de las subhuellas son bastante estables frente a algunas degradaciones del audio típicas como pueden ser ruido blanco agregado, una ecualización del audio, efectos de una compresión como “mp3” entre otros.

La distancia entre dos huellas que contengan el mismo audio (salvo por distorsiones), se puede calcular como la distancia de hamming promedio para cada pareja de subhuellas considerando que éstas están alineadas. Estadísticamente, si los audios son diferentes, debería en promedio tener 16 bits diferentes (al ser subhuellas de 32 bits y siendo que cada uno de ellos se puede modelar como que toman valores equiprobables) y la distancia debería ser en promedio 0 bits diferentes por subhuella si se trata de exactamente el mismo audio sin distorsión.

Un aspecto importante a resaltar es la estabilidad de los valores que toma una huella distorsionada, que junto con la robustez a grandes variaciones debido a distorsiones del audio hacen que la probabilidad de que al menos una subhuella esté incambiada respecto a la original en un tramo de audio de 3 segundos es muy alta. Simultáneamente el dominio de las huellas es muy grande 2^32, por lo que la probabilidad de que haya dos huellas iguales por

azar es relativamente baja. Por todo esto, es posible implementar un sistema de búsqueda de los audios mediante la búsqueda de un “ancla” que corresponda a una subhuella inalterada que permita establecer el alineamiento para realizar la comparación. En la figura se muestra una huella de 256 subhuellas de largo, en su versión original, luego de comprimir el audio con mp3 y la comparación de la degradación de los bits de la huella. (imagen del artículo original)



La naturaleza de las subhuellas permite además almacenar las huellas de los audios en una tabla de Hash, donde la función de hash puede ser simplemente una parte de la subhuella (por ejemplo los últimos 16 bits). De esta manera es muy rápido encontrar las posibles anclas (coincidencia de subhuellas).

Asumiendo de que la cantidad de audio total que puede corresponder a todos los audios pregrabados en Little bridge sea menor a 10 horas (360000 suhuellas), tomando una tabla de Hash de 16 bits, correspondería en promedio a tener unas 6 subhuellas que pueden ser candidatas. Esto hace la búsqueda súper rápida y dado que 360000 es sensiblemente menor a 2^32, la cantidad de coincidencias de posibles subhuellas que efectivamente coinciden por azar (y sean consideradas “ancla”) es baja, evitando una cantidad excesiva de comparaciones de las huellas completas.

Adaptación al problema

Esta técnica fue principalmente desarrollada para el reconocimiento de música, por lo que tiene algunos problemas cuando se utiliza con voz hablada. El principal problema es que cuando hay una pausa de más de 400ms (por ejemplo entre frases o cuando hay un cambio de hablante en un díálogo), la derivada de la potencia respecto al tiempo y respecto a diferentes bandas es prácticamente aleatoria y cercana a cero generando huellas que no cuentan con información

útil para representar el contenido del audio. Esto hace que para voz hablada, haya una degradación sensible aumentando la distancia de hamming de las huellas en los tramos con silencios de 400ms o más. Esto es posible solucionarlo sustituyendo uno de los bits de la huella por un bit que indique si la potencia en un cierto entorno (promediada en unos 200ms) es sensiblemente más baja que un promedio con una constante mayor (unos 2000ms). Este bit será luego utilizado para ponderar con un factor menor el promedio calculado de la distancia de hamming entre dos huellas.

Los valores de las variables utilizados en el proyecto son diferentes al artículo original:

Frecuencia de muestreo: 8000Hz
Tiempo entre subhuellas: 92 muestras = 10ms
Ancho de ventana de análisis: 1536 muestras = 192ms
Rango de frecuencias: 300Hz a 2000Hz

Se decidió fijar la frecuencia de trabajo en 8000Hz ya que cubre todo el rango de frecuencias.
Se eligió un ancho de ventana de 10 ms para tener más granularidad temporal que resulta necesaria para los archivos de audio de duración muy pequeña (menos de 2 segundos). Por el mismo motivo se eligió una ventana de análisis de ancho sensiblemente menor al artículo original, ya que los fragmentos de audio muy pequeños a ser detectados, quedan “invadidos” por el contenido previo y posterior, degradando el desempeño en estos casos. La naturaleza de las huellas y sus propiedades no cambian sustancialmente al cambiar estas variables, pero incrementan sensiblemente la capacidad de detección de segmentos cortos, que no era parte de lo que se buscaba en el artículo original.

Experimentos

A los efectos de verificar el funcionamiento de esta técnica con los datos disponibles se procedió a implementar la búsqueda de un conjunto reducido de audios de Little Bridge en las clases grabadas disponibles de Ceibal en Inglés.

Cabe destacar que los únicos archivos disponibles para detectar que efectivamente aparecen en las clases disponibles son archivos super cortos (menos de 2 segundos) correspondientes a sonidos que se reproducen automáticamente cuando se inician o completan actividades. De todas formas, la tecnología se muestra que obtiene muy buenos resultados para secuencias de audio largas (3 segundos o más).

Los experimentos fueron realizados con 13 audios de Little Bridge, en 5 clases de Ceibal en Inglés. De acuerdo a lo observado manualmente sólo 2 de los audios en la base de datos aparecen en las 5 clases totalizando 5 apariciones, de las cuales se detectan 4. Tres de las 5 apariciones están claramente solapadas con voces de interacción de la clase, y en particular una de ellas fuertemente solapada, por lo que se considera que si bien esa aparición no fue detectada, la degradación del audio en ese caso, junto con que el audio tiene una duración super corta, hacen extremadamente difícil la detección y se puede considerar como un caso de borde.

Conclusiones y trabajo a futuro.

Se implementó un sistema de búsqueda de fragmentos de audio que es robusto y rápido basado en el trabajo de audio fingerprinting [1]. Se hicieron algunas modificaciones tanto a la definición de la huella (bit de silencio) como el ajuste de los parámetros que definen las huellas para resolver mejor los casos de borde de fragmentos muy cortos (que no es parte del problema planteado en el artículo original). Se obtuvieron buenos resultados para la pequeña cantidad de datos disponible. Se espera recolectar una base de datos razonable para validar mejor el funcionamiento.

Referencias

[1] Jaap Haitsma and Ton Kalker, “A Highly Robust Audio Fingerprinting System”, International Society for Music Information Retrieval Conference, 2002

```
@inproceedings{Haitsma2002AHR,
  title={A Highly Robust Audio Fingerprinting System},
  author={Jaap Haitsma and Ton Kalker},
  booktitle={International Society for Music Information Retrieval
Conference},
  year={2002}
}
```

Reporte 8

Análisis automático de clases grabadas Ceibal - Detección de Final de Clase

Autor: Emilio Martínez

Introducción

En el marco del proyecto Herramientas automáticas de observación de aula para el análisis de prácticas docentes en clases a distancia se dispone de videos de clases remotas por videollamadas, que corresponden a los programas educacionales Ceibal en Inglés (CEI) y Pensamiento Computacional (PC). Muchos de estos videos suelen terminar bastante después de la finalización de la clase, al no cortar la grabación manualmente[2]. De esta manera sigue habiendo actividad de voz por parte de los alumnos, lo que influye en detecciones espurias de los sistemas de detección de hablante idioma, tanto como frases y gramáticas vistos anteriormente.
Es por esto que se investiga diferentes maneras de detectar el fin de la clase (es decir, cuando el docente remoto se retira de la videollamada) y se desarrolla una solución basada en la señal de potencia del audio correspondiente al video de la clase., detallada al final del documento.

Datos

Se usaron un total de 10 clases donde se anotaron manualmente el fin efectivo de la clase. En las grabaciones se encuentran diferentes situaciones:

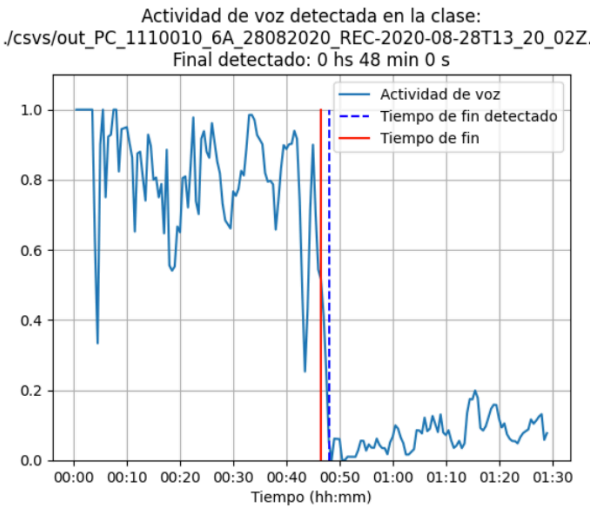
- Video cortado enseguida el docente termina la clase (caso ideal, sin necesidad de detección)
- Video que sigue grabando aún luego de terminada la clase, pero con el audio silenciado en esta última etapa (detección fácil)
- Video con audio aún luego de finalizada la clase (dificultad variada: depende del nivel de murmullo de los estudiantes y cuánto tiempo se prolonga)

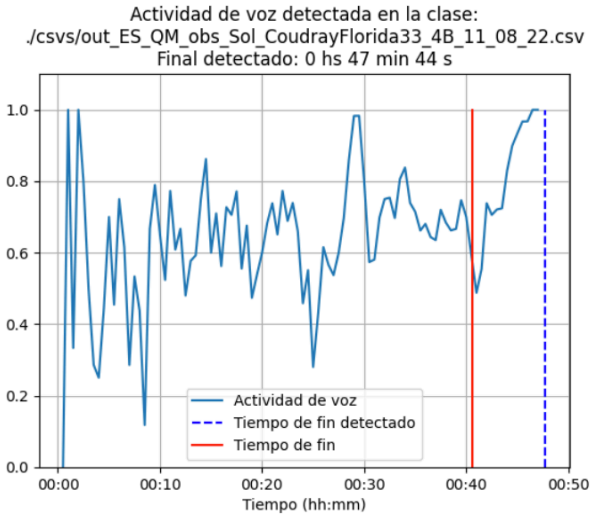
Método de Voice Activity Detection (VAD)

Una manera de identificar el fin de clase es extraer una distribución del uso de la voz durante la clase, por lo que se opta probar con un *Voice Activity Detection* o *VAD*. Hay que tener en cuenta que este método puede llegar a detectar actividades de voz en donde haya una o más

[2] En este ejemplo se tiene un video de 1h 5m de duración, cuando la clase termina efectivamente al minuto 45:30. Por lo que se estaría analizando de manera espuria más de 15 minutos de audio.

personas hablando (tanto de manera contínua como una voz por encima de otra). Las siguientes figuras ilustran dos casos de detección de fin de la clase con este método, junto al final real de la videollamada:

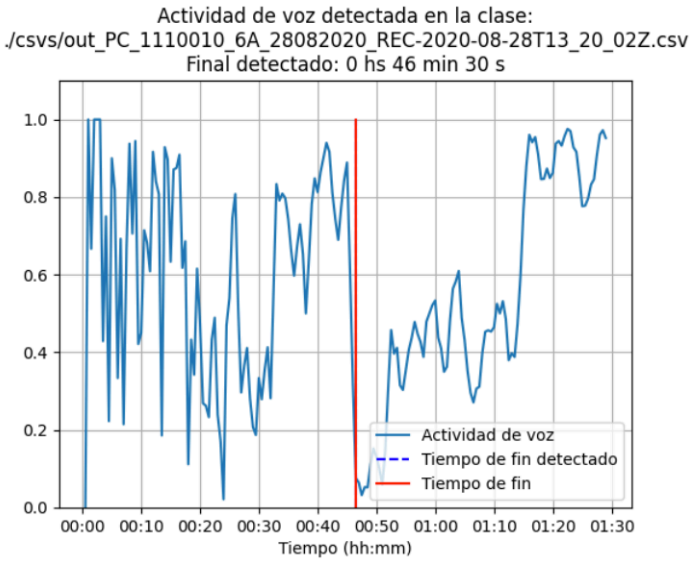


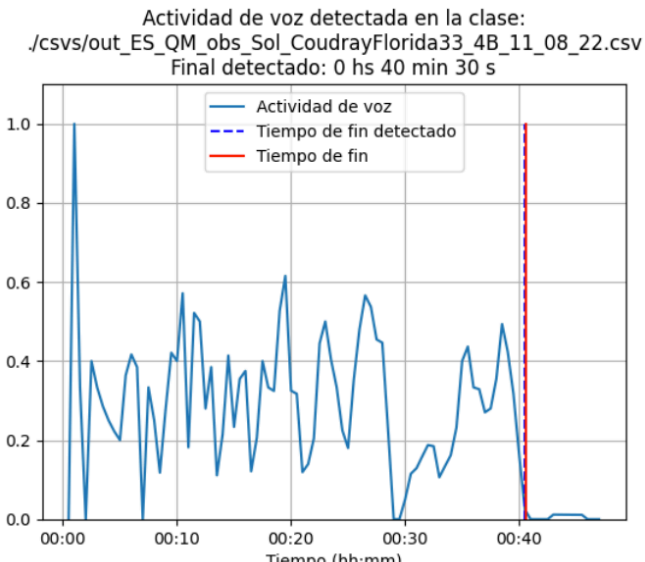


Aquí se toma el fin de clase como el primer instante, luego de pasados los primeros 35 minutos de la grabación, en el que el porcentaje de actividad de voz cae a un 10%. La detección de la figura de la izquierda se encuentra un tiempo de fin cercano al real, mientras que el ejemplo de la derecha difiere de aproximadamente 9 minutos.

Método identificación del habla del profesor remoto con CAD (Classroom Activity Detection)

La idea es idéntica al método con VAD, pero en vez de usar todas las detecciones de voz se elige utilizar únicamente la correspondiente al docente remoto mediante su detección con CAD. Esto tiene cierto sentido ya que no importa los instantes de la grabación cuando ya deja de participar el docente en el aula. Los resultados son los siguientes:

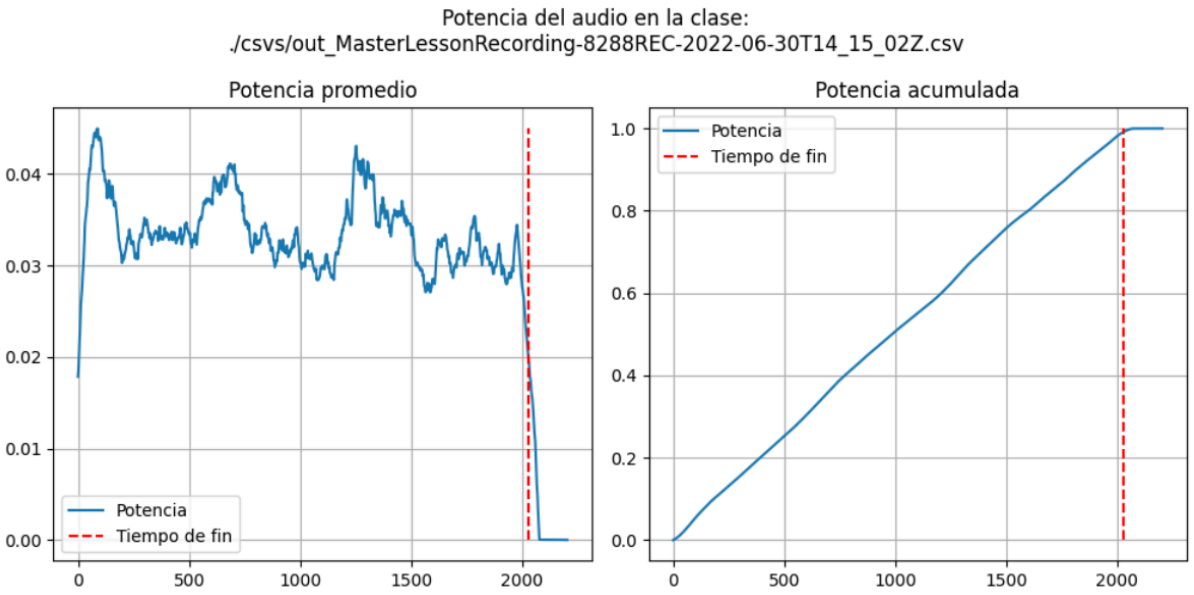


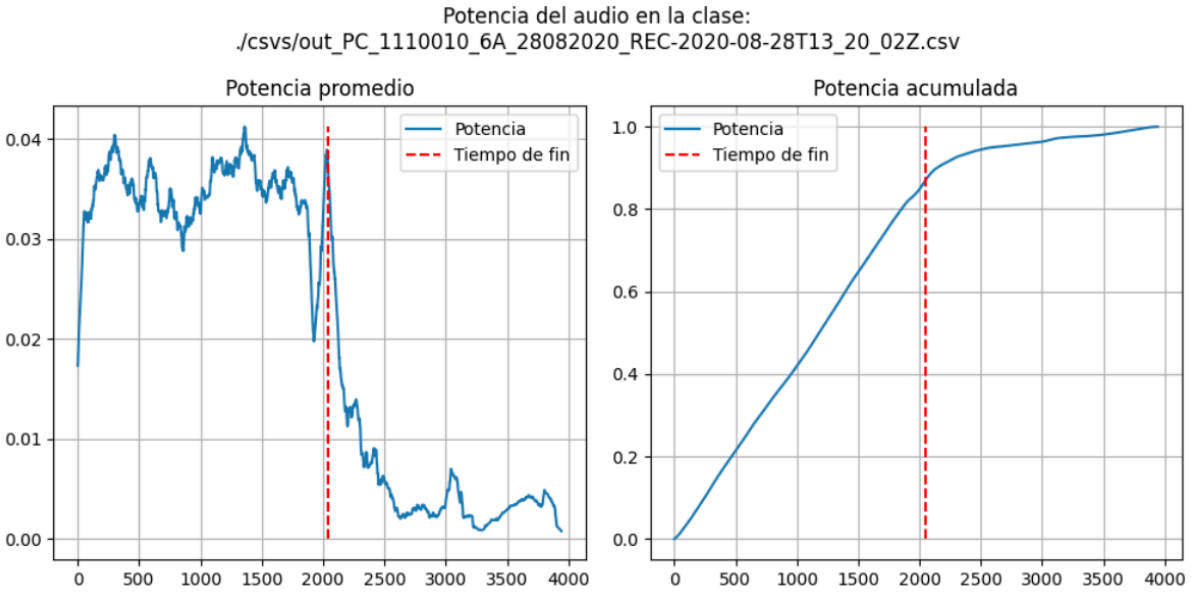


Aquí se tomó el mismo criterio de esperar hasta los primeros 35 minutos de la clase para detectar el final, con el primer instante que baje del 10% de actividad de voz (esta vez, por parte del docente). Aún así, se nota que no es perfecta la detección, ya que se encuentra actividad del docente aún finalizada la clase.

Método basado en Potencia del Audio

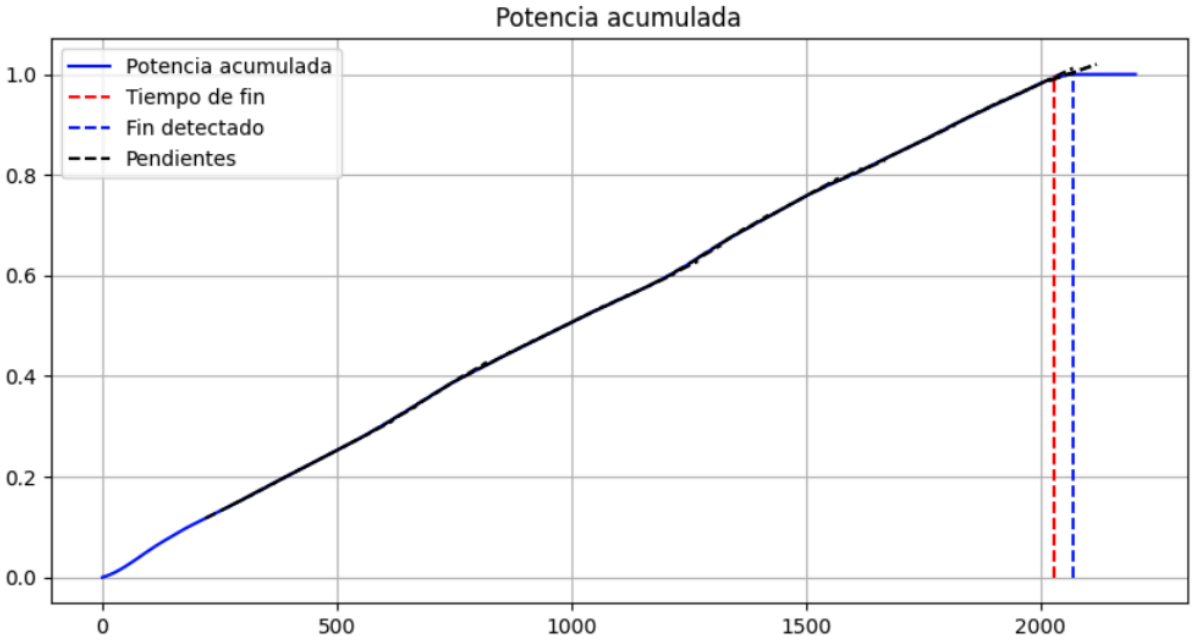
Finalmente se describe el método elegido para la detección del fin de clase, esta vez basado en la potencia de la señal de audio. Para esto se recurre a la biblioteca librosa en lenguaje Python. La siguiente figura muestra cómo es la potencia (izq.) y la potencia acumulada o energía (der.) a lo largo de una clase.

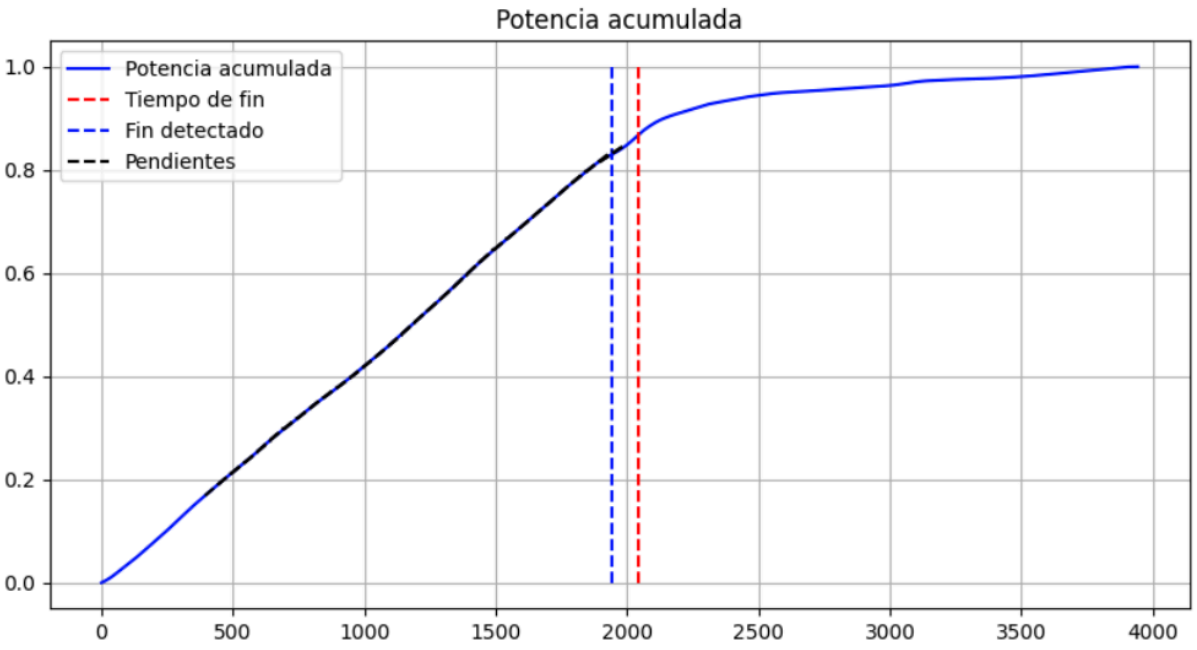




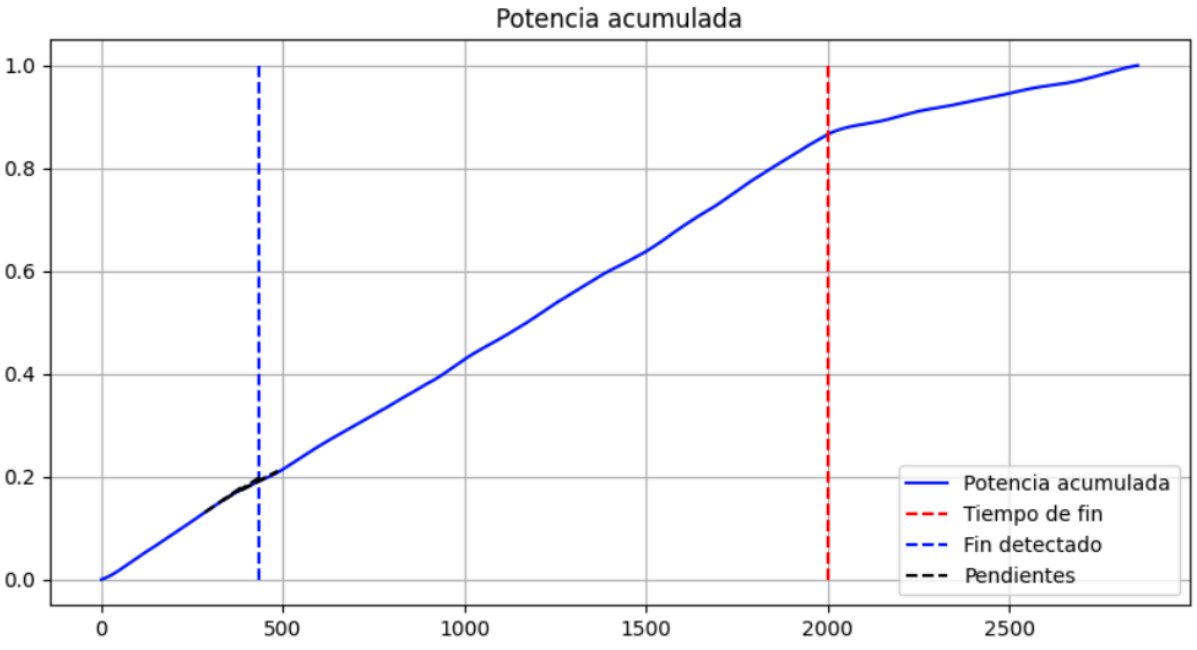
Para la potencia acumulada se puede notar que al cambiar de pendiente coincide con el fin de la clase. Esto se puede deber a que, al finalizar la clase, hay menor potencia media, por lo que empezaría a acumular más lento a partir de ese instante.

Con el fin de detectar este cambio, se calcula la pendiente local a lo largo de la curva de potencia acumulada. Una vez que se obtiene un cambio significativo en la pendiente, respecto a su historial (comparando con la mediana de todas las pendientes anteriores) se considera ese instante como el fin de clase. En la siguiente figura se tiene la detección del fin de clase para las dos clases de la figura anterior. Aquí se nota dónde se detecta el cambio de pendiente, quedando cerca del fin efectivo de la clase.





Un aspecto a considerar es que la detección puede ser temprana respecto al final real de la clase si se tiene un cambio abrupto de pendiente al principio de la clase, lo que implicaría en un descarte de información para el análisis de clase. Un ejemplo se puede ver en la siguiente figura.



Esto se puede solucionar imponiendo que el fin a detectar este cerca de la duración normal de una clase (45 minutos, por ejemplo).

Vale la pena aclarar que este método de detección por pendientes de la energía necesita la configuración de varios parámetros (ventana y salto/hop para calcular la pendiente, margen de cambio de pendiente para evitar detecciones espurias, etc), lo que puede requerir una búsqueda de los mismos para que el método se adecue al comportamiento deseado.