Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Maestría en Explotación de Datos y Descubrimiento de Conocimiento

Sistema de identificación de idioma (LID) para grabaciones de entornos naturales bilingües en comunidades qom

Tesis presentada para optar al Título de Magíster de la Universidad de Buenos Aires

Autor: Lic. Leandro Martín Garber
Director: Dr. Pablo Riera (UBA; UNQ)
Co-Directora: Dra. Florencia Alam (CIIPME-CONICET)

Buenos Aires, Argentina
Fecha de presentación del ejemplar: Noviembre 2022
Fecha de defensa: 19/12/2022
Lugar de trabajo: CIIPME-CONICET

Resumen

Conocer la cantidad de habla que perciben bebés bilingües en cada idioma es fundamental para diseñar programas educativos que contemplen las características lingüísticas propias de este tipo de entornos y promover así mejores posibilidades de aprendizaje. Precisamente, en este trabajo obtengo una medición estimada de la cantidad de habla en qom y español en el entorno del hogar de 8 bebés que viven en contextos rurales indígenas qom a partir del desarrollo de un sistema de identificación de idioma (LID, *spoken Language IDentification*). Dicho sistema es entrenado con un conjunto de grabaciones en entornos naturales en comunidades rurales qom en Argentina.

Este estudio se centra en tres ejes principales: el primero es la descripción de una heurística para codificar los datos de entrenamiento de manera eficiente, el segundo es una comparación de modelos usando técnicas de aprendizaje por transferencia (*transfer learning*) y el tercero es la cantidad de horas de habla en cada lengua para las 8 familias participantes en este experimento.

La arquitectura estudiada es wav2vec 2.0 y se utilizan modelos pre-entrenados a los que se realiza ajuste fino (*fine tuning*). Los modelos son evaluados en su eficacia y capacidad de generalización. Para esto se presentan los resultados al evaluar los mismos con datos fuera de dominio y del mismo dominio. Para el primer caso se consiguió un EER de 0,37, un 21 % mejor que el modelo base. Para el segundo caso el EER es de 0,23, un 8 % mejor que el modelo base. La conclusión es que los modelos wav2vec 2.0 obtienen una eficacia superior y una muy marcada ventaja en capacidad de generalización pero no son tan robustos a la variabilidad de canal y necesitan ajuste fino.

Por último, de las familias participantes se extrajeron 61 horas de habla de las cuales 46.57 (76 %) pudieron ser clasificadas con mayor certeza. El procedimiento revela que el entorno lingüístico de lxs niñxs participantes posee 55 % más habla en qom (28.34 horas) con respecto a español (18.22 horas). Estos resultados son un aporte fundamental a la psicolingüística en tanto que permiten el análisis de grandes corpus de datos de habla en contextos naturales de forma automática, análisis que resultaría muy costoso para ser realizado de forma manual.

Language Identification system for bilingual natural environment recordings in Qom communities

Abstract

Understanding the amount of speech perceived by bilingual babies in each language is essential to design educational programmes that take into account the linguistic characteristics of this type of environments and thus promote better learning possibilities. Precisely, in this paper I obtain an estimated measurement of the amount of speech in Qom and Spanish in the home environment of 8 infants living in rural indigenous Qom contexts by developing a language identification system (LID). This system is trained with a set of recordings in natural environments in rural Qom communities in Argentina.

This study focuses on three main axes: the first one is the description of a heuristic to encode the training data efficiently, the second one is a comparison of models using transfer learning techniques and the third one is the number of hours of speech in each language for the 8 families participating in this experiment.

The architecture studied is wav2vec 2.0 and pre-trained models are used and fine-tuned. The models are evaluated for their efficiency and generalisation capacity. For this purpose, the results are presented when evaluating with out-of-domain and in-domain data. For the first case an EER of 0.37 was achieved, 21 % better than the base model. For the second case the EER is 0.23, 8 % better than the base model. The conclusion is that the wav2vec 2.0 models have a superior efficiency and a very marked advantage in generalisability but are not as robust to channel variability and need fine tuning.

Finally, 61 hours of speech were extracted from the participating families, of which 46.57 (76 %) could be classified with greater certainty. The procedure reveals that the linguistic environment of the participating children has 55 % more speech in Qom (28.34 hours) than in Spanish (18.22 hours). These results are a fundamental contribution to psycholinguistics in that they allow for the analysis of large corpora of speech data in natural contexts automatically, an analysis that would be too costly to be done manually.

Agradecimientos

A mis padres por enseñarme el valor del conocimiento, a mi hermano Flavio por su empatía y apoyo, a mi hermano Sebas por cuidarme aunque llueve o truene. A Miguela por compartir su calidez y sostenerme con su cariño.

A Pablo, mi director, fuente de motivación inagotable. A Flor, mi directora, que me abrió las puertas a mundos desconocidos. A Celia por su generosidad en el liderazgo y su incansable trabajo. A las familias de las comunidades qom y a lxs colaboradorxs bilingües que participan de este estudio.

A lxs profes y estudiantes de Artes Electrónicas de la UNTREF por su sentido de comunidad, respeto y pensamiento crítico.

Valoro sin duda el privilegio de haber podido contar con la calidad humana de quienes me ayudaron, de haber tenido el tiempo y contar con el hardware necesario para llevar esta investigación a cabo. Todo esto gracias a la estructura pública de educación e investigación de este país que da a lugar al derecho al saber y a los esfuerzos individuales y colectivos de las personas que la sostienen.

Índice

1. Introducción

Los sistemas de identificación de idioma en el habla (LID, *spoken Language IDentification*) se ocupan de procesar y analizar una grabación de audio de una persona hablando y asignarle una clase o probabilidad de que corresponda con el idioma utilizado. La mayoría de estos sistemas son entrenados utilizando grabaciones con hablantes monolingües en entornos controlados. Estos sistemas se evalúan en su capacidad para diferenciar entre una variedad de idiomas y/o dialectos. Algunas conferencias notables donde se discute este tipo de tecnología son Interspeech[1] y Odyssey[2]. NIST[3] (National Institute of Standards and Technology) también coordina importantes competencias que cada año establecen el estado del arte en distintos temas vinculados al análisis del habla.

La mayoría de la investigación en este tipo de sistemas se enfocan en sólo 20 de los 7000 idiomas que existen en el mundo. El resto se definen (pobremente) como "idiomas de bajos recursos"(*LRLs, Low-Resource Languages*). Se entienden estas lenguas como: menos estudiadas, que poseen escasez de recursos, menos digitalizadas, de baja densidad y, desde otra óptica, menos privilegiadas e incluso críticas y en peligro de extinción [1].

En este estudio construyo y evalúo un sistema que permite clasificar segmentos de audio en español y en qom a partir de un corpus de audio [2] que consiste en grabaciones de situaciones espontáneas del entorno de niños en comunidades rurales indígenas bilingües en Pampa del Indio, Chaco, Argentina. El idioma qom (o toba) es hablado por 30410 personas a nivel nacional (INDEC, Encuesta Complementaria de Pueblos Indígenas (ECPI) 2004-2005) y puede ser considerado como una LRL.

Es importante abordar toda la cadena de procesos del sistema de LID teniendo en cuenta que uno de los idiomas es una LRL utilizando técnicas específicas que permitan maximizar el rendimiento. En este escrito se describen las mismas, desde la recolección de datos hasta el entrenamiento, estudiando su eficacia y capacidad de generalización.

Uno de los principales objetivos de este proyecto es obtener una estimación de la cantidad de habla en cada lengua que escuchan los niñxs de estas familias. Se trata de una tarea altamente relevante en psicolingüística y educación bilingüe, pero que también exige una gran cantidad de recursos humanos y económicos. Usar las tecnologías del habla para automatizar esta tarea es fundamental para mejorar la investigación en estos campos, caracterizar el entorno lingüístico infantil de manera más precisa y abrir nuevos caminos que surjan de la posibilidad de analizar grandes cantidades de horas de habla que superan la escala humana.

Estudios [3, 4, 5] han mostrado cómo la cantidad de *input* lingüístico está relacionado robustamente con el desarrollo del lenguaje en lxs niñxs monolingües. El bilingüismo y particularmente su relación con el input lingüístico ha sido de considerable interés para

[1] `https://www.isca-speech.org/`
[2] `http://www.speakerodyssey.com/`
[3] `https://www.nist.gov/itl/iad/mig/language-recognition`

investigadores que estudian adquisición temprana de la lengua y asimismo los estudios que atendieron al input de niñxs que crecen en entornos bilingües han mostrado una mayor variabilidad en el mismo con respecto a contextos monolingües [6]. Desde una perspectiva educativa es fundamental medir los factores del entorno que dan forma al desarrollo lingüístico y cognitivo de lxs niñxs bilingües con el objetivo de diseñar programas educativos que contemplen las características lingüísticas propias de estos entornos y promover así mejores posibilidades de aprendizaje.

La literatura en el campo de la tecnología del habla con grabaciones en situaciones espontáneas y para su uso en ciencias sociales es escasa. Es por ello que el proceso de tomas de decisiones dio lugar a preguntas específicas a este tipo de corpus. Codificar los segmentos de audio para armar el conjunto de datos de entrenamiento es una tarea lenta y costosa, ¿cuánto es el mínimo que se puede codificar para obtener resultados aceptables? ¿qué heurísticas se pueden proponer para elegir qué segmentos codificar?, y más relevante aún: ¿puede el modelo resultante generalizar lo suficiente para ser usado en otras comunidades y/o entornos? ¿hasta qué punto? ¿qué estrategias se pueden llevar a cabo para intentar medir el grado de generalización?.

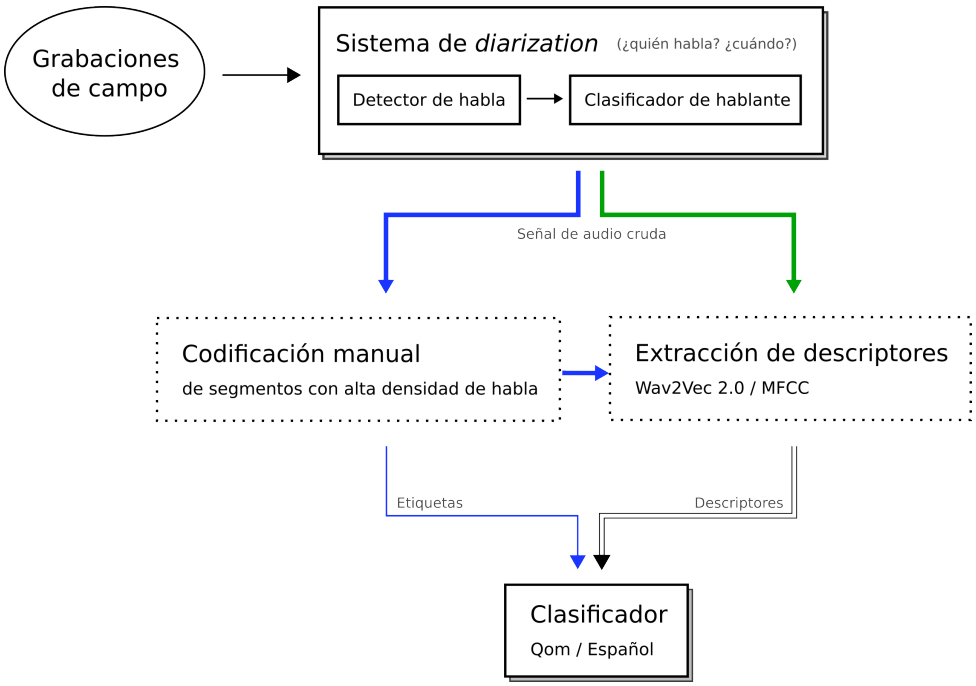

Figura 1: Cadena de procesos del sistema completo. Los módulos con línea punteada son los que profundizo en este estudio. Las líneas marcadas en azul corresponden con el flujo de datos para la etapa de entrenamiento y evaluación, la verdes para la etapa de obtención del resultado final.

En la Figura 1 se puede ver un diseño con los distintos módulos que componen el sistema completo de punta a punta y como se relacionan entre sí. La principal contribución

de este trabajo es el diseño de esta arquitectura y la evaluación de la misma en términos de rendimiento y capacidad de generalización estimada.

En el mismo esquema sintetizo dos etapas marcadas con líneas coloreadas. Las líneas azules responden al flujo correspondiente con la etapa de entrenamiento y evaluación mientras que la línea verde corresponde con la etapa de obtención del resultado final: la cantidad estimada de habla en cada idioma. La línea sin colorear corresponde a ambas etapas.

A partir de las grabaciones de campo (Sección 2.1) se procesa la señal de audio con un modelo pre-entrenado de *diarization* que la segmenta respondiendo las preguntas 'quién habla' y 'cuándo' (Sección 2.2.1). Esto permite extraer los segmentos de habla y desechar secciones donde no hay un hablante (ruido de ambiente, máquinas, animales, etc.). Dependiendo de la etapa, la salida de este módulo es utilizada de manera distinta. En la primera, este proceso es usado para elegir fragmentos con alta densidad de habla para que la **codificación manual** de idioma -necesaria para el conjunto de datos de entrenamiento- sea más eficiente y balanceada (Sección 2.4), los fragmentos elegidos son segmentados y codificados manualmente. En la segunda etapa, es usada directamente para alimentar el módulo de **extracción de descriptores**. Este módulo donde comparo descriptores tradicionales MFCC con los novedosos modelos preentrenados wav2vec 2.0 [7] constituye uno de los principales aportes de la presente tesis, en la sección 2.5 se describe su funcionamiento y su relevancia en esta tarea.

Codificación manual y **extracción de descriptores** están marcados con línea punteada ya que son los que profundizo en este estudio.

Por último proponemos un clasificador sencillo (Seccion 2.6) basado en una red de retardo de tiempo (*TDNN, Time Delay Neural Network*) [8] que nos permitirá evaluar los descriptores y finalmente proveernos de una cantidad estimada de habla en cada idioma para toda la muestra.

El aporte de este trabajo es fundamentalmente: describir el problema, caracterizar los desafíos, sugerir un plan de trabajo para abordar la codificación de audios de estas características y comparar el rendimiento de diversas técnicas específicas en tecnología del habla que son el estado del arte en el campo de LID orientado a LRLs en los últimos años. Las diversas áreas que utilizan audios largos con habla en situaciones espontáneas [9, 10, 11] también encontrarán un aporte si planean utilizar este tipo de tecnología.

En síntesis, el objetivo principal del presente trabajo es:

- Obtener el clasificador de segmentos de audio variable con menor nivel de error para el problema detallado
    - Estudiar distintas arquitecturas y técnicas de redes neuronales artificiales aplicadas a identificación de idioma así como las posibilidades de ingeniería de variables para audios de estas características

- Evaluar la eficacia de los modelos preentrenados wav2vec 2.0
- Observar la capacidad de generalización del modelo: cómo afecta al rendimiento el conjunto de datos de entrenamiento en cantidad y variedad
- Obtener una estimación de la cantidad de habla en cada lengua para el corpus completo

1.1. Declaración ética

Esta investigación es conducida siguiendo los lineamientos éticos especificados en el expediente 5344/99 de CONICET, aprobado y supervisado por el comité de ética. Los padres de los bebés que participan en las grabaciones nos han entregado su consentimiento escrito para su participación y la de sus hijos.

2. Materiales y métodos

2.1. Conjunto de datos

El corpus de datos [2] está compuesto por 164 horas de grabaciones de audio del entorno lingüístico de 8 niñxs (9;26 meses, $\mu = 17.6$, $s = 6.19$) en casas bilingües qom-español de comunidades rurales qom situadas en Pampa del Indio, Chaco, Argentina. Para la recolección de datos se le colocó un chaleco con un micrófono al bebé que permaneció prendido por un promedio de 8 horas sin la presencia de un investigador. Se realizaron 4 grabaciones a 4 familias con 6 meses de diferencia entre cada una (146 horas). En otras 4 familias se grabó una sola toma (18 horas). Los audios fueron segmentados y clasificados por especialistas: primero se segmentaron las emisiones de audio que contenían habla y luego se les asignó la etiqueta Español, Qom o Indefinido. Esta última etiqueta corresponde a emisiones sin contenido lingüístico tales como gritos, llantos, etc.

Familia	**Toma 1**	**Toma 2**	**Toma 3**	**Toma 4**	**Total**
1	334	507	494	275	1610
2	720	526	481	389	2356
3	503	480	578	316	2206
4	720	480	478	345	2579
5	402				402
6	177				177
7	249				249
8	232				232
Total					∼164 horas

Tabla 1: Detalle de las duraciones de los audios por familia en cada toma (en minutos)

Las grabaciones en situaciones espontáneas tienen las siguientes particularidades:

- Son de baja calidad si las comparamos con grabaciones en un estudio controlado con micrófonos óptimamente posicionados o llamadas telefónicas
- Tienen un alto nivel de ruido ya que sucede todo tipo de eventos durante los diálogos: ruido de viento, máquinas, animales, música, televisión, etc.
- Muchas grabaciones muestran solapamiento entre hablantes
- Pueden poseer largos segmentos temporales sin habla

Además de estas características, la tarea es particularmente desafiante dada la manera en que los hablantes utilizan ambas lenguas: un mismo hablante puede a veces usar una lengua, a veces otra, a veces en una misma emisión mezclar ambas y en algunas ocasiones no hay una diferencia prosódica o acústica clara que marque la alternancia de código. Además, los hablantes son de distintas edades y frecuentemente se incluyen niñxs con baja madurez vocálica. Siendo qom un LRL sumado a todas estas particularidades, hacen a esta tarea un desafío no trivial y que no está ampliamente discutido en la literatura de LID.

2.2. Preprocesamiento y limpieza

Los archivos de audio originales no pueden ser utilizados en su formato original. En esta sección detallo los procesos necesarios para extraer los segmentos que contienen habla, limpiarlos, optimizarlos y hacerlos compatibles con el input de los modelos experimentales.

Estas tareas comprenden:

1. Extraer los segmentos que contengan habla
2. Remover segmentos con demasiado ruido
3. Cortar los segmentos mayores a 3 segundos por una limitación de hardware
4. Convertirlos al formato que recibe el modelo

2.2.1. Extracción de habla

Utilicé un modelo pre-entrenado [12, 13] de *diarization* que separa diferentes interlocutores en una grabación de audio y responde a las preguntas de 'quién habla' y 'cuando'.

Este modelo está basado en una arquitectura de redes neuronales artificiales llamada SincNet [14] y fue entrenado con ∼237 horas de audio en situaciones espontáneas que cubren un gran rango de entornos, condiciones e idiomas que se asemejan a nuestro conjunto de datos.

Como muestra la Figura 2, la red es entrenada a partir de secuencias de audio de duración fija que son procesadas por SincNet que aprende una batería de filtros específicos

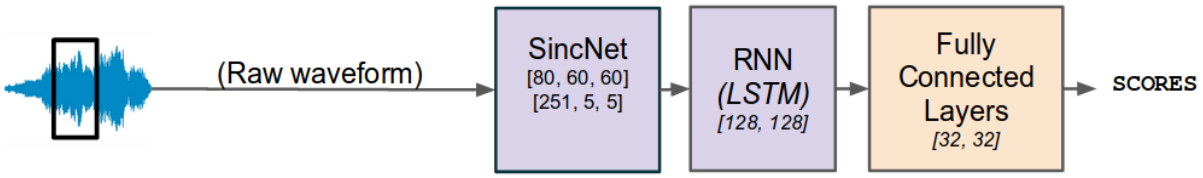

Figura 2: Diagrama de la arquitectura para el modelo utilizado de *diarization*

para esta tarea. Esta representación de bajo nivel es utilizada como input en 2 capas recurrentes tipo *LSTM* (*Long short-term memory*) bidireccionales y luego 3 capas densas que terminan en una función de activación sigmoidea que devuelve una puntuación para cada clase: *hombre, mujer, niñx principal, niñx secundario, habla*. Se trata de un problema de clasificación multiclase donde cada segmento de audio podría resultar en más de una etiqueta.

Al igual que este trabajo, lxs autorxs de este modelo comparan su eficacia usando dos conjuntos de evaluación: uno con datos que comparten el dominio de los datos de entrenamiento y otro con datos nuevos, nunca vistos por el modelo (*hold-out*). Encuentran un 10 % de mejora cuando se trata de datos cuyo dominio estaba en el set de entrenamiento.

El modelo mostró buenos resultados seleccionando los segmentos de habla pero no distingue los de habla electrónica (radio, TV, tablet, etc). Sería interesante entrenar al modelo agregando una nueva clase de habla electrónica para poder estudiar por separado lo que es habla emitida por el entorno afectivo del niñx y los medios de comunicación.

Utilizo este sistema con dos finalidades: como heurística para codificar los datos de entrenamiento de manera eficiente (Sección 2.4.3) y para extraer todo el habla del corpus completo con el fin de clasificarlo.

2.2.2. Limpieza y optimización

Por limitaciones de hardware, las muestras de habla que pude procesar tienen un máximo de 3 segundos. Las muestras de al menos 5 segundos fueron recortadas en segmentos de 3 segundos para aprovechar al máximo el corpus.

Escuché atentamente y de manera aleatoria el 20 % del corpus y encontré que las muestras menores a 1.5 segundos solían ser de menor calidad que el resto, habiendo una gran cantidad de las mismas que sólo le aportaban ruido al conjunto de datos. Dado que ya contaba con un corpus reducido, eliminar todas las muestras cortas implicaba una reducción considerable del tamaño de la muestra es por ello que decidí cargarlas en el software AudioStellar [15] (Figura 3). El mismo extrajo descriptores *MFCC* de estos archivos de audio y mediante una cadena de procesos de reducción de dimensionalidad generó una visualización 2D. Además este software realiza un análisis de agrupamientos en baja dimensión proponiendo grupos por densidad. El procedimiento entonces consistió en escuchar 3 muestras aleatorias de cada grupo, si estas eran de baja calidad, entonces

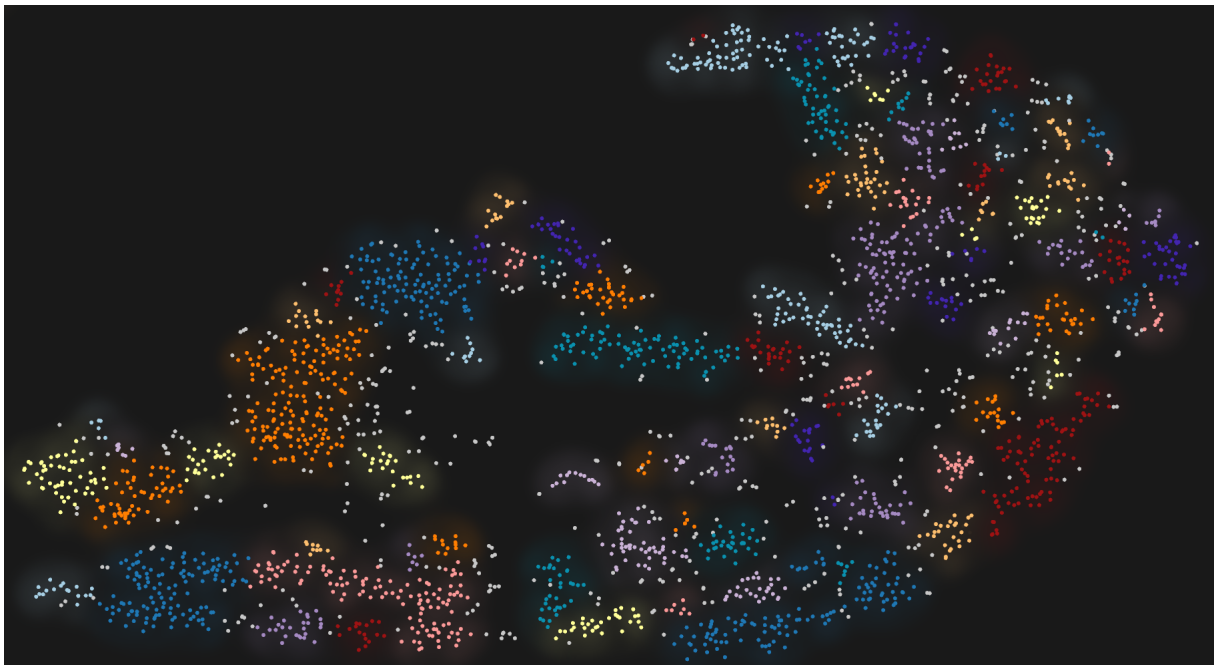

Figura 3: Mapa de sonidos del corpus generado por AudioStellar [15]. Cada punto es una muestra del corpus cuya duración es menor a 1.5 segundos. Puntos cercanos corresponden a muestras similares tímbricamente (usando descriptores *MFCC*)

escuchaba otras 3. Si estas últimas tampoco tenían las calidad suficiente entonces eliminaba ese grupo del corpus. Alrededor del 40 % de muestras menores a 1.5 segundos fueron eliminadas de esta manera.

Finalmente, todos los segmentos fueron convertidos a archivos WAV mono de 16000hz de frecuencia de muestreo.

2.3. Conjunto de entrenamiento y validación

Para evaluar los modelos resultantes de cada experimento hice uso de la técnica de validación cruzada *k-folding*. La misma consiste en construir k particiones (*folds*) que son submuestreos del conjunto total y entrenar cada modelo k veces usando *k-1* particiones para el entrenamiento y ese 1 restante para la evaluación. Se trata de una técnica robusta que permite analizar si el rendimiento es consistente para los distintos conjuntos de entrenamiento y sacar estadísticos que permiten estimar mejor el desempeño del clasificador en la población general.

Esto es de suma relevancia para este trabajo ya que uno de los objetivos es el de estimar la capacidad de generalización de los modelos. Tomé dos conjuntos de particiones:

1. **Estratificado**. Son 5 particiones sampleadas del conjunto de datos de tal manera que cada una contenga una cantidad balanceada de: segmentos de audio en idioma qom y español, tipo de hablante (mujer, hombre y niñx) y duración del segmento. Cada partición es de ∼23 minutos, lo que resulta que en cada entrenamiento se utili-

zarán ∼92 minutos como conjunto de entrenamiento y ∼23 minutos como conjunto de evaluación.

Este conjunto nos permite evaluar los modelos en el hipotético caso de que nuestro muestreo sea representativo de la población general. En la Sección 2.4.3 se describe la estrategia utilizada para acercarnos a este supuesto.

2. **Por familia**. Son 7 particiones que corresponden a las grabaciones de cada familia y tienen duraciones variables. El escenario que se evalúa al usar este corte es un poco más realista que el anterior para interpretar el poder de generalización del modelo dado que me aseguro que el mismo no tenga ninguna información sobre el conjunto de evaluación: elimino la posibilidad de que conceptualice y use información del canal en vez de la lengua en sí para clasificar [4].

   La variabilidad de canal es uno de los retos más importantes en las aplicaciones de voz dado que las características que nos interesan para clasificar y la información del canal se encuentran juntas y vinculadas en el espectro. Conceptualmente, queremos que el modelo aprenda una representación que desenrede esta información para poder separarla.

   Como puede observarse en la Figura 4, la cantidad de habla grabada de la familia #4 está enormemente desbalanceada. Es por esto que esta familia nunca es usada como conjunto de evaluación para disminuir el sesgo causado por entrenar con una cantidad muy baja de datos.

Pretendo que la comparación entre estos dos conjuntos eche algo de luz acerca de la capacidad de generalización de los modelos y de cuenta de la relevancia de codificar los datos utilizando la heurística propuesta en la Sección 2.4.3.

2.4. Codificación

Esta tarea comprendió escuchar los audios, segmentar donde hay habla y finalmente codificar el segmento utilizando las etiquetas **Qom** o **Español** dependiendo de si la emisión esta principalmente en un idioma o el otro. El equipo de investigación interdisciplinario en CIIPME-CONICET ha hecho esta codificación manualmente y la misma requirió un primer trabajo de segmentación y codificación por lenguas y luego un trabajo con un hablante nativo al que se le consultaban las dudas de codificación.

Todo esto demanda una gran cantidad de tiempo y recursos humanos y económicos. Este proceso aplicado sobre el corpus de audio completo sería una tarea imposible y es precisamente este problema el que buscamos resolver en la presente tesis.

[4] Por ejemplo, que la reverberación de un recinto donde siempre se habla español sirva como sesgo para clasificar, ya que en la población podría existir un recinto similar donde se hable qom

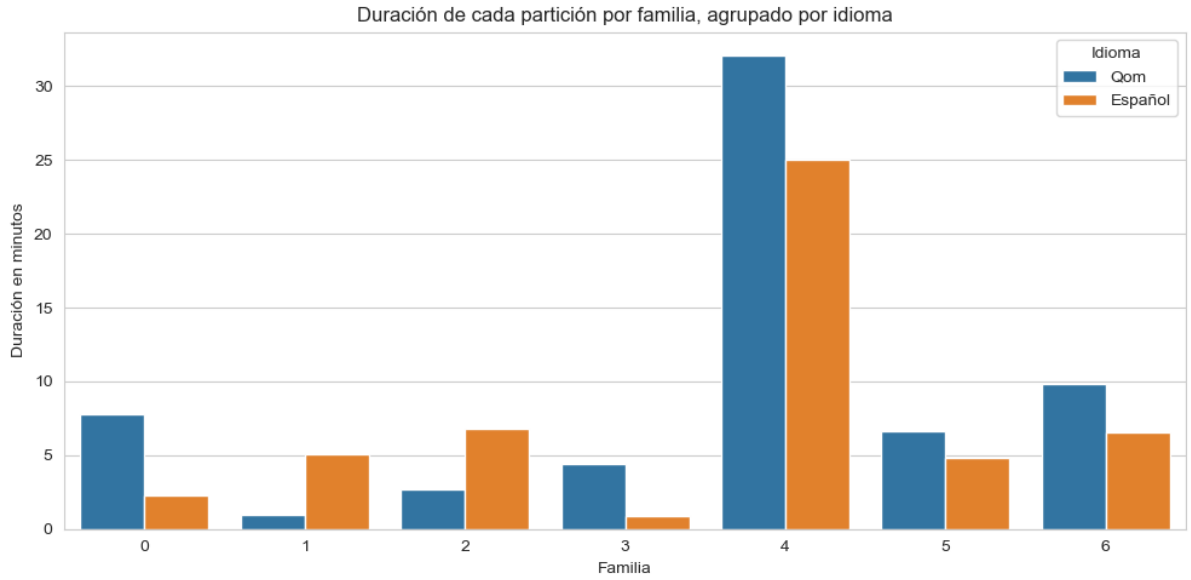

Figura 4: Cantidad de tiempo por idioma para el conjunto de particiones **por familia**.

Los datos con los que contaba antes de comenzar este trabajo eran 10 minutos en formato CHA y 8 minutos en formato EAF. A los primeros era necesario corregirlos utilizando un sistema de detección de habla. El resto de los datos de entrenamiento fueron codificados durante el desarrollo de este estudio, primero muestreando una hora aleatoriamente y luego optimicé esta tarea tomando muestras con alta densidad de habla. Estos procesos se detallan a continuación.

2.4.1. Formato CHA

Al momento de iniciar esta tesis, contábamos con 10 minutos de habla que se encontraba transcrita y codificada por hablante, lengua y otras variables lingüísticas relevantes tales como a quién estaba dirigida el habla (a un niñx o a un adulto). Para almacenar esta información se había utilizado el software CLAN [16] que cuenta con el formato de archivo CHA. El problema principal que detectamos con este software es que sólo se marca el tiempo al final de cada emisión, esto hace que no sepamos cuando comienza la misma ya que si segmentamos sólo usando estas marcas, el clip de audio resultante contendrá ruido entre emisiones.

Para solucionar este problema, primero segmentamos usando las marcas de tiempo y luego procesamos cada clip utilizando el algoritmo To-Combo-SAD [17, 18]. Se trata de una técnica de detección de habla (*SAD, Speech Activity Detector*) que devuelve los instantes de tiempo donde comienza y termina la emisión de un hablante en una señal sonora. Utiliza un combinación lineal (componentes principales) de 5 descriptores cuya distribución muestra una evidente bimodalidad, en donde las ventanas de habla y no habla se muestran separadas (Fig. 5). Luego se ajusta un modelo de mezcla gaussiana (*GMM,*

Gaussian Mixture Model) para obtener ambas medias y clasificar las ventanas de audio. Posee un parámetro α que permite configurar la proporción de falsos positivos, el mismo fue ajustado manualmente tomando muestras y escuchando el resultado.

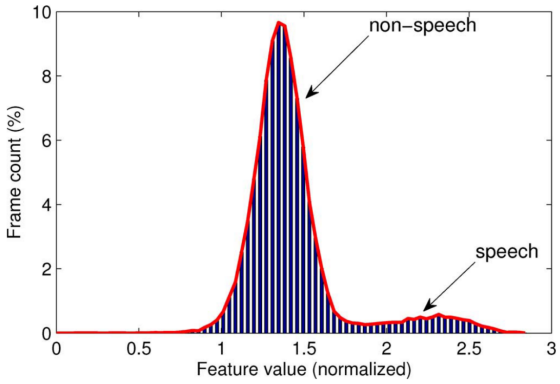

Figura 5: Distribución del descriptor combinado para una señal de audio arbitraria [17].

El procesamiento de los archivos CHA fueron hechos con una librería en Python que programé ad-hoc [19].

2.4.2. Formato EAF

Contábamos además con 8 minutos de habla codificada de igual manera pero utilizando el software ELAN [20]. El mismo posee un sistema de segmentación más avanzado y una librería de Python [21] que permite interpretar los archivos y conseguir un clip de audio por cada emisión de una manera sencilla.

Este formato es el que usamos de ahora en adelante para las tareas que requieran segmentación.

2.4.3. Optimización de la codificación de datos de entrenamiento

En un primer momento seleccionamos de forma manual una hora de audio en la que detectamos que había habla y la codificamos. Sin embargo, de esta manera sólo logramos extraer alrededor de 10 minutos de habla por cada segmento elegido ya que el resto era silencio, habla electrónica o ruido. Notamos además que el conjunto de entrenamiento resultaba muy desbalanceado con gran cantidad de habla de niñxs y mujeres y muy baja cantidad de habla de hombres por cada lengua.

Es por ello que opté por un sistema automático que detecte segmentos de alta densidad de habla por hablante. Este sistema permitiría una codificación más eficaz para lograr extraer más habla y balancear en mayor medida los distintos hablantes.

Desarrollé un algoritmo que tomara el output del sistema de *diarization* (Sección 2.2.1) y ordene, por hablante, ventanas de 5 minutos de audio por densidad de habla. De esta manera pudimos seleccionar para codificar 12 segmentos de 5 minutos, entre todo el

corpus de audio, con gran densidad de habla y poder elegir entre hablantes masculinos y femeninos para balancear el conjunto de entrenamiento.

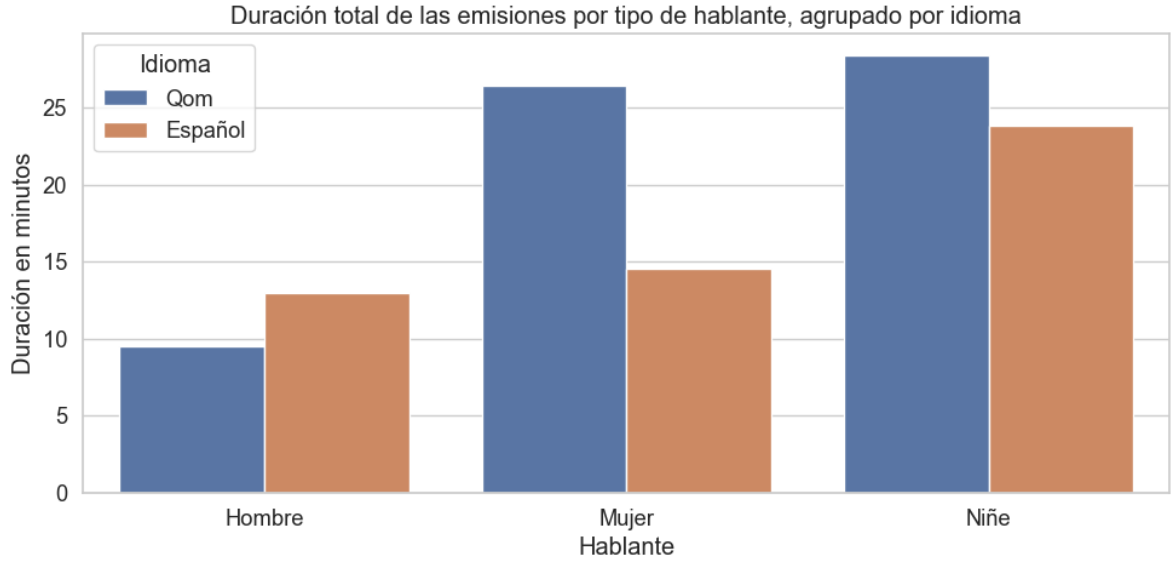

Figura 6: Duración total de las emisiones por tipo de hablante, agrupado por idioma. Los totales son 64 minutos de habla en qom y 51 en español.

En la Figura 6 puede observarse la distribución del conjunto de datos final. De los 64 minutos de habla en qom y 51 en español, existe un fuerte desbalance de idioma cuando el hablante es mujer, donde el habla en qom es 70 % mayor al habla en español. Por otro lado 45 % del habla total es producida por niñxs. El habla masculina es la más difícil de encontrar en las grabaciones y es por eso que representa la menor parte aunque bastante balanceada, esto es gracias a la heurística propuesta.

2.5. Extracción de descriptores

En este estudio se compara la eficacia de distintos descriptores obtenidos mediante modelos preentrenados de la arquitectura wav2vec 2.0 [7] y descriptores base MFCC (*Mel Frequency Cepstral Coefficients*) con delta y delta-delta.

Cuando comencé con este trabajo en el 2019 los modelos más avanzados de LID proponían el uso de *bottleneck features* (BNF, descriptores de cuello de botella) [22, 23, 24] que se basaban en entrenar un clasificador de fonemas con una lengua de altos recursos (en general inglés) y luego extraer las salidas de una capa intermedia. Con la aparición de los mecanismos de atención [25] llegaron nuevos estudios que utilizaban bloques *transformer* para obtener modelos de lenguaje utilizando una pérdida contrastiva (*contrastive loss*) primero en modelos de lenguaje escrito y luego en modelos de habla como wav2vec 2.0. La eficacia de este tipo de bloques es el estado del arte en las arquitecturas modernas de procesamiento natural del lenguaje (NLP, *Natural Language Processing*) debido a que

pueden captar dependencias de plazos más largos que lo que conseguían las unidades recursivas como *LSTM* o *GRU*. Además son paralelizables, lo que permite procesar un flujo mayor de datos en cada pasada a costa de mayor consumo de memoria.

La hipótesis de estos modelos es que la red capta relaciones a largo plazo y logra construir conocimiento de las características del habla que no es específica a ninguna tarea en particular. De esta manera, los mismos descriptores se podrían usar para reconocimiento automático del habla (*ASR, Automatic Speech Recognition*), clasificación del hablante, clasificación de emociones, identificación de idioma, entre otros.

Utilizo la implementación en PyTorch a través de la libreria 🤗HuggingFace [26]. Dado que el idioma qom es un LRL, mi hipótesis es que una herramienta de aprendizaje por transferencia de estas características aprovechan mejor la poca cantidad de datos que se pudieron recolectar y, aunque el modelo preentrenado no cuente con información del qom, sí tiene información de habla en español y fonemas pertenecientes a otras lenguas que podrían ser similares.

2.5.1. Wav2Vec 2.0

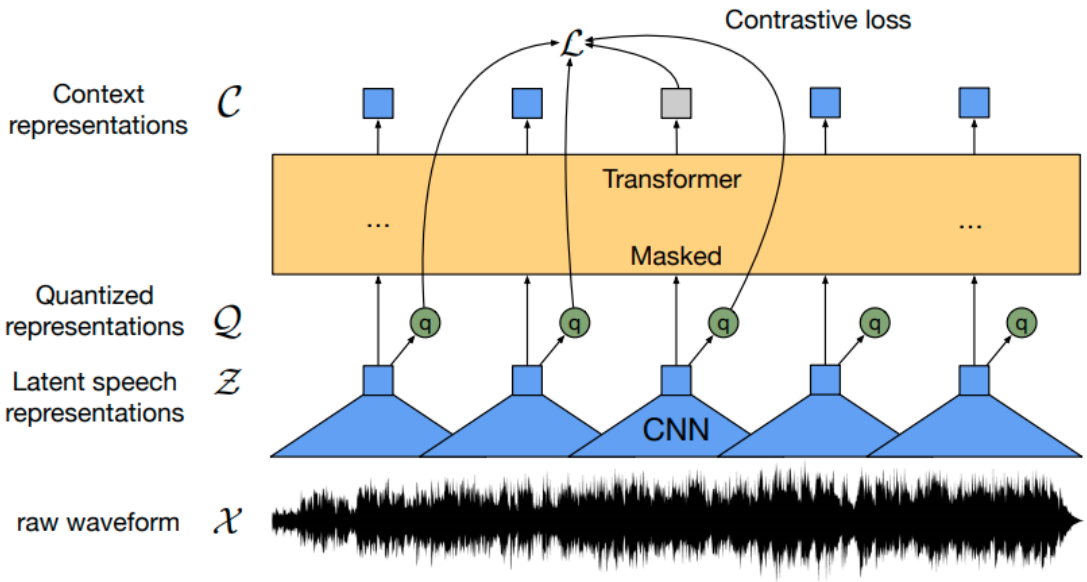

Figura 7: Diagrama de la arquitectura wav2vec 2.0 [7]

En la figura 7 se puede observar la arquitectura de wav2vec 2.0. El input del modelo es la señal cruda en el dominio del tiempo de cada emisión. La misma puede ser de largo variable y la frecuencia de muestreo debe coincidir con los datos de preentrenamiento, esta es 16000Hz. Los datos deben tener media 0 y varianza 1. Otra de las diferencias que se pueden encontrar cuando comencé esta investigación es que la práctica habitual era utilizar descriptores MFCC o similares como input de las redes. En estos últimos años es más común utilizar un codificador de descriptores que toma el audio crudo y consta de una

serie de capas convolucionales que extraerán descriptores relevantes para la tarea. En esta arquitectura se utilizan 7 bloques de convoluciones temporales con 512 canales en cada uno, el tamaño del kernel y el *stride* son pequeños y van decreciendo con la profundidad de la red. Antes de entrar al bloque de *transformers* y dada su arquitectura paralela, es necesario que se codifique información posicional sobre los fonemas: el orden y la posición de los mismos es esencial en las lenguas ya que definen la gramática y la semántica de las emisiones. La propuesta en [26] para ASR es utilizar una técnica que modula los vectores de palabras utilizando funciones trigonométricas y que consigue embeber información de posicionamiento absoluto en cada palabra. En el caso del audio, la literatura sugiere hacer uso de un codificador y decodificador convolucional con un kernel lo suficientemente grande a diferencia del codificador de descriptores utilizado anteriormente. En este caso se trata de una capa convolucional temporal de 16 grupos con tamaño de kernel 128 y stride de 1. Con esta estrategia es que se consigue embeber información de posicionamiento relativo a cada fonema. Esta idea es similar al concepto de *senone* [27] donde cada vector no tiene sólo información sobre el fono, sino que se lo aumenta con información contextual que refiere a fonos pasados y futuros. La hipótesis en este punto es que los senones resultantes de cada uno de los idiomas serán distintos y por lo tanto extraer conocimiento de la dependencia entre los mismos es relevante para nuestro problema de clasificación.

Wav2Vec 2.0 propone dos configuraciones distintas para los bloques de *transformers*, el `BASE` y el `LARGE`. En este experimento uso los modelos preentrenados *VoxPopuli* y *XLSR* que utilizan la configuración `LARGE`. Esta está conformada por 24 bloques de *transformers* de dimensión 1024 para los proyectores KVQ (*Key, Value, Query*) y 4096 para la capa densa. La salida de la red es una serie de vectores llamados representaciones de contexto que son las que uso como input para el clasificador. En la misma figura 7 se observa también la formalización de un espacio Q que corresponde con las representaciones cuantizadas. La misma sólo se usa durante el preentrenamiento y es necesaria para la estrategia auto-supervisada (*self-supervised*) de la arquitectura. El espacio Z que resulta del codificador de descriptores es discretizado en vectores representativos (conceptualmente similares a los centroides que se podrían obtener con k-medias [28]) que se almacenan en un libro de códigos (*codebook*) utilizando *Gumbel softmax*. Para el preentrenamiento auto-supervisado se enmascaran (ocultan) ciertas secciones de la emisión y utilizando la función de pérdida contrastiva se fuerza a la red a reconstruir lo enmascarado logrando aprender un modelo de habla sin necesidad de tener etiquetas de clase.

2.5.2. Ajuste fino (*Fine tuning*)

Lxs autorxs de wav2vec2 sugieren realizar un ajuste fino de los modelos preentrenados para conseguir que se adapten al problema en cuestión utilizando todo el bagaje de conocimiento adquirido con el conjunto de datos original con el que fueron entrenados (*transfer learning*).

En este caso, el procedimiento consiste no solo en entrenar el módulo de clasificación que añado a la arquitectura (ver Sección 2.6) sino también las capas de *transformer* del modelo original.

En la Sección 2.7 comparo los resultados entre los modelos con y sin ajuste fino.

2.5.3. Modelos preentrenados

Los dos siguientes modelos preentrenados son los que uso en este estudio para comparar su efectividad y capacidad de generalización. Ambos corresponden a la arquitectura `LARGE` de wav2vec2 y son entrenados con los conjuntos de datos que detallo en esta sección.

Vale la pena notar que en ambos casos no se incluye el idioma qom y, si bien sí incluyen español, éste no es la variante hablada en el Chaco. Además, hay una diferencia de canal muy importante: los entornos donde fueron grabados son totalmente diferentes y los micrófonos utilizados son distintos. La variabilidad de canal es uno de los retos más importantes en las aplicaciones de voz dado que las características que nos interesan para clasificar y la información del canal se encuentran juntas y vinculadas en el espectro. Conceptualmente, queremos que el modelo aprenda una representación que desenrede esta información para poder separarla.

VoxPopuli

El modelo preentrenado VoxPopuli fue lanzado en el 2021 por Facebook y utiliza el conjunto de datos con el mismo nombre [29]. Cuenta con 100k horas de habla en 23 idiomas: Búlgaro, Checo, Croata, Danés, Holandés, Inglés, Estonio, Finlandés, Francés, Alemán, Griego, Húngaro, Italiano, Latvio, Lituano, Maltés, Polaco, Portugués, Rumano, Eslovaco, Esloveno, Español y Sueco. Estas grabaciones provienen del parlamento europeo entre los años 2009 y 2020. Las grabaciones originales fueron segmentadas en clips de 15-30 segundos de duración usando un detector de actividad vocal basado en energía (*VAD, voice activity detector*).

XLSR-53

El modelo preentrenado XLSR-53 fue lanzado en el 2020 por Facebook [30]. Está compuesto de 3 conjuntos de datos:

- CommonVoice [31]. Es una iniciativa de Mozilla que emplea el *crowdsourcing* para la recolección y validación de los datos. Los colaboradores utilizan la web o la aplicación para iPhone y leen oraciones que aparecen en pantalla. Son 2k horas de habla en 38 idiomas, aunque para el entrenamiento del modelo se utilizaron 11 idiomas: Español, Francés, Italiano, Kyrgyz, Holandés, Ruso, Sueco, Turco, Tatar y Chino.

- BABEL [32]. Son conversaciones telefónicas de IARPA en distintos idiomas de los que se incluyó Bengalí, Cantonés, Georgiano, Haití, Kurmanji, Pashto, Tamil, Turco, Tokpisin y Vietnamita.
- Multilingual LibriSpeech (MLS) [33]. Es un corpus de audio derivado de la lectura de audiolibros de LibriVox y consiste en 8 idiomas: Holandés, Inglés, Francés, Alemán, Italiano, Polaco, Portugués y Español. Son alrededor de 50k horas de habla, donde 44k son en inglés.

Se combinan estos 3 conjuntos de datos para conseguir 56k horas de habla en 53 idiomas.

2.6. Clasificador

Dado de que uno de los principales objetivos en este trabajo es poder comparar la efectividad de los descriptores obtenidos a partir de modelos wav2vec 2.0, el módulo de clasificación es bastante mínimo. Se trata de una pequeña red de retardo de tiempo (*TDNN, Time Delay Neural Network*) [8, 34] con sólo dos capas, de contexto [1,3] y *stride* 1, se puede ver una descripción de la arquitectura en la Tabla 2. El input de esta red de clasificación son los descriptores obtenidos, en el caso de los modelos wav2vec 2.0 se trata de una matriz de 1024 filas x 194 columnas, estas últimas corresponden con 3 segundos de audio. La primera capa de la red de clasificación reduce la dimensionalidad a 64x194 y la segunda a 32x192, luego una operación de *average pooling* obtiene un vector de 32 posiciones que se conecta directamente con una capa densa con una función de activación sigmoidea que asegura una salida entre 0 y 1 que es usada para clasificar. En el caso del modelo base con descriptores MFCC + delta + delta-delta la única diferencia es que el input es de 60x94 entonces la primera capa reduce la dimensionalidad a 32x94 y la segunda a 16x92. Esta TDNN es implementada en PyTorch como una capa convolucional 1D de kernel 1 para la primera capa y 3 para la segunda. Cada capa está seguida de *Batch Normalization*, una función de activación ReLU y *Dropout* de 0.1. La función de pérdida es una entropía cruzada binaria entre las probabilidades objetivo resultantes, sin regularización. El optimizador es Adam y el factor de aprendizaje un hiperparámetro que resultó entre 10^{-5} y 10^{-3} dependiendo del experimento.

2.7. Experimentos

Los experimentos efectuados tienen como objetivo evaluar la efectividad y la capacidad de generalización del sistema propuesto. Para ello se han entrenado diez modelos utilizando dos estrategias distintas para separar los datos de entrenamiento y prueba usando validación cruzada *K-Fold*: 5 particiones estratificadas y 7 particiones por familia (ver

Descriptores	Capa	Canales	Kernel	Entrada	Salida
wav2vec 2.0	Convolucional 1D	64	1024x1	1024x194	64x194
	Convolucional 1D	32	64x3	64x194	32x194
	Average pooling	-	-	32x192	32x1
	Densa	1	-	32x1	1x1
MFCC	Convolucional 1D	32	60x1	60x94	32x94
	Convolucional 1D	16	64x3	32x94	16x94
	Average pooling	-	-	16x92	16x1
	Densa	1	-	16x1	1x1

Tabla 2: Descripción de la arquitectura del clasificador

Sección 2.3). La estrategia estratificada nos informará el rendimiento del modelo si contamos con un muestreo representativo de la población, en la práctica esto no es posible pero el objetivo de la heurística de codificación de datos que usé en este trabajo fue diseñada para tender a eso (ver Sección 2.4). Por otro lado, las particiones por familia informarán el rendimiento cuando el dominio de los datos de evaluación no ha sido vistos por la red al momento de entrenar y de esta manera poder evaluar su capacidad de generalización.

Por cada uno de los grupos de particiones (estratificada y por familia), los modelos a comparar son:

- Modelo base (*baseline*)
- Wav2Vec2 preentrenado con VoxPopuli, con y sin ajuste fino
- Wav2Vec2 preentrenado con XLSR-53, con y sin ajuste fino

2.7.1. Métrica EER (*Equal Error Rate*)

La métrica que utilizo para medir la eficacia de los modelos es la más comúnmente usada en literatura *LID* y de datos biométricos en general, llamada *EER, Equal Error Rate*. La *EER* corresponde con el punto de una curva *ROC* donde la proporción de falsos positivos y falsos negativos es equivalente, en otras palabras, el umbral de aceptación donde el costo por clasificar incorrectamente es igual para una clase que para el otra. En este problema tiene sentido ya que el costo por clasificar incorrectamente cada clase es el mismo.

Cuanto más bajo el valor de *EER*, más alta es la eficacia del sistema.

2.7.2. Aumento de datos (*data augmentation*)

Esta técnica se basa en generar sintéticamente más datos de los que tenemos. Se clona una proporción de los datos originales de nuestro corpus y se los procesa aplicando una perturbación que puede mejorar la capacidad de generalización del modelo.

En este trabajo usé la librería nlpaug [35] que posee una sección dedicada el audio. Los procesos utilizados fueron los siguientes:

- ***Pitch shifting***. Se transforma la señal original para que la misma suene más agudo o más grave.
- **Velocidad**. Se aumenta o disminuye la velocidad de la señal. Esto también causa que la misma cambie en altura.
- **Desplazamiento**. Se desplaza la señal original hacia la izquierda o la derecha.
- **Perturbación del largo del tracto vocal, (*VTLP*)**. Se trata de un proceso en el dominio de la frecuencia que utiliza filtros Mel y los corrompe para modificar la voz del hablante [36].

Todos los modelos fueron entrenados con un porcentaje de aumento de datos: [0, 0.1, 0.2, 0.4]. Las muestras elegidas para aumentar fueron clonadas 4 veces y a cada una se le aplica un proceso de los detallados.

2.7.3. Tasa de aprendizaje

Todos los modelos usan el optimizador Adam con los parámetros por defecto de la librería de PyTorch excepto la tasa de aprendizaje (*learning rate*). La misma es considerada un hiperparámetro que se optimizó entre 0.01 y 0.000001.

2.8. Hardware utilizado

- CPU: Intel(R) Core(TM) i7-7700 @ 3.60GHz
- GPU: GeForce RTX 2080 Ti 11GB GDDR6
- Memoria RAM: 32gb

2.9. Software utilizado

- Lenguaje: Python
- Librerías: PyTorch, Tensorflow, Numpy, Matplotlib, Pandas, scikit-learn, librosa, seaborn, nlpaug

3. Resultados y discusión

3.1. Modelos de evaluación

En la Tabla 3 pueden observarse los resultados de los experimentos. Para los conjuntos de particiones estratificadas el **modelo base** muestra un muy buen desempeño que los modelos con ajuste fino sólo pueden superar marginalmente tanto cuando tomamos la media como cuando tomamos el desvío estandar y los máximos alcanzados. Inesperadamente los modelos wav2vec 2.0 sin ajuste fino se muestran sustancialmente peor, es aparente que la gran diferencia de canal y la ausencia del idioma qom son un factor determinante para que no pueda conseguir descriptores robustos y separables. En este escenario es **VoxPopuli con ajuste fino** el modelo que mejor rendimiento y mejor varianza presenta si bien **XLSR-53 con ajuste fino** resulta también muy parecido. Si bien este último resultado era el más esperado, la hipótesis inicial era que se separe aún más del **Modelo base**.

	EER		
	Media	Desvío estandar	Rango
K-Fold estratificado			
Modelo base	0.25	0.05	[0.16,0.30]
VoxPopuli sin ajuste fino	0.28	0.05	[0.22, 0.33]
VoxPopuli con ajuste fino	**0.23**	0.03	[0.19, 0.27]
XLSR-53 sin ajuste fino	0.27	**0.02**	[0.25, 0.30]
XLSR-53 con ajuste fino	0.24	0.04	**[0.18, 0.27]**
K-Fold por familia			
Modelo base	0.45	0.11	[0.30, 0.57]
VoxPopuli sin ajuste fino	0.42	0.08	[0.39, 0.52]
VoxPopuli con ajuste fino	**0.37**	**0.07**	**[0.26, 0.46]**
XLSR-53 sin ajuste fino	0.43	0.09	[0.35, 0.59]
XLSR-53 con ajuste fino	**0.37**	**0.07**	[0.28, 0.47]

Tabla 3: Resultados de los experimentos.

Al momento de tomar el conjunto de particiones por familia es cuando los modelos wav2vec 2.0 se separan definitivamente del **modelo base** consiguiendo resultados 21 % mejores. Se puede observar un comportamiento similar que en los experimentos anteriores donde los modelos wav2vec 2.0 con ajuste fino se muestran con mejor desempeño. En este caso sí se puede observar una ventaja marginal de los modelos sin ajuste fino sobre el **modelo base** con mejores media y variabilidad.

Podemos verificar entonces, que los descriptores generados por los modelos wav2vec 2.0 generalizan mejor que los tradicionales *MFCC* y consiguen mejor rendimiento y certeza aunque es sumamente importante realizar el ajuste fino en el caso de contar con datos de otros dominios. En otras palabras, los resultados indican que si bien los modelos sin duda han aprendido características acústicas y fonológicas también tienen un fuerte sesgo por el canal: los micrófonos utilizados, el recinto, la distorsión, etc. que puede mejorar haciendo ajuste fino de los mismos. Además se puede ver que en este problema los conjunto de entrenamiento VoxPopuli y XLSR-53 se comportan de manera similar siendo que el

primero contiene 100 mil horas de habla y el segundo 56 mil horas si bien este último concentra más idiomas y más diferencia de canal (detalle en Sección 2.5.3).

En cuanto al aumento de datos, el único modelo que se vio beneficiado por el mismo es el **modelo base** para el conjunto estratificado. El resto, todos presentaban menor rendimiento, incluso al 0.1. Mi hipótesis era que todos iban a presentar una mejoría y no queda claro si la estrategia de aumento fue equivocada o que simplemente la técnica no da resultados positivos con estos datos que contienen alto nivel de ruido.

La tasa de aprendizaje óptima resultó de 0.0001 para el **modelo base** y el modelo sin ajuste fino y de 0.00001 para los que sí se realizó el ajuste. Está claro que al aplicar esta técnica es necesario utilizar una tasa más baja y un mayor número de épocas.

El tamaño de lote (*batch size*) se configuró en 16 para los modelos con ajuste fino y en 64 para el resto de los modelos. Esto es debido a una limitación de hardware por falta de memoria. En estudios de ablación confirmé que hay un impacto negativo cuando reduje el tamaño de lote para los modelos sin ajuste fino.

3.2. Modelo final

Para obtener una medición del habla en cada idioma de los datos de evaluación no codificados, utilizo el modelo **VoxPopuli con ajuste fino** entrenado con la totalidad de datos de entrenamiento. Los hiperparámetros elegidos corresponden con los del **K-Fold estratificado** ya que los datos de entrenamiento contienen grabaciones de cada una de las familias de los datos de evaluación. El error esperado es el que figura en la Tabla 3, alrededor del 23 % con un desvio del 3 %.

Como marca la Figura 1, los datos son primero procesados por el sistema de *diarization* donde primero extraigo los segmentos de habla. Seleccioné aquellos segmentos mayores a 2 segundos y donde el hablante no es el niñx principal. Los segmentos mayores a 5 segundos fueron recortados en segmentos más cortos de a 3 segundos (ver Sección 2.2.2).

En la Figura 8 podemos observar un histograma y una estimación de densidad sobre los puntajes (*scores*) arrojados por la salida del modelo cuando el mismo es alimentado con los datos de evaluación no codificados. Podemos observar dos picos en cada uno de sus extremos que corresponden a los datos clasificados con mayor certeza, los cercanos a 0 son de habla en español y los cercanos a 1 son de habla en qom. Se puede apreciar que hay más cantidad de datos certeros en el pico derecho (qom) en comparación con el pico izquierdo (español). A partir de la estimación de densidad el mismo es 70 % mayor.

Para dar un estimación en horas de la cantidad de habla en cada idioma es menester elegir un umbral donde los segmentos puntuados sobre ese umbral serán considerados habla en qom y los que se encuentren por debajo, habla en español. Esta elección no es de ninguna manera trivial o automática ya que hay elegir un criterio de cómo hacerlo y la misma modifica la estimación final. En principio lo ideal es elegir el punto de corte

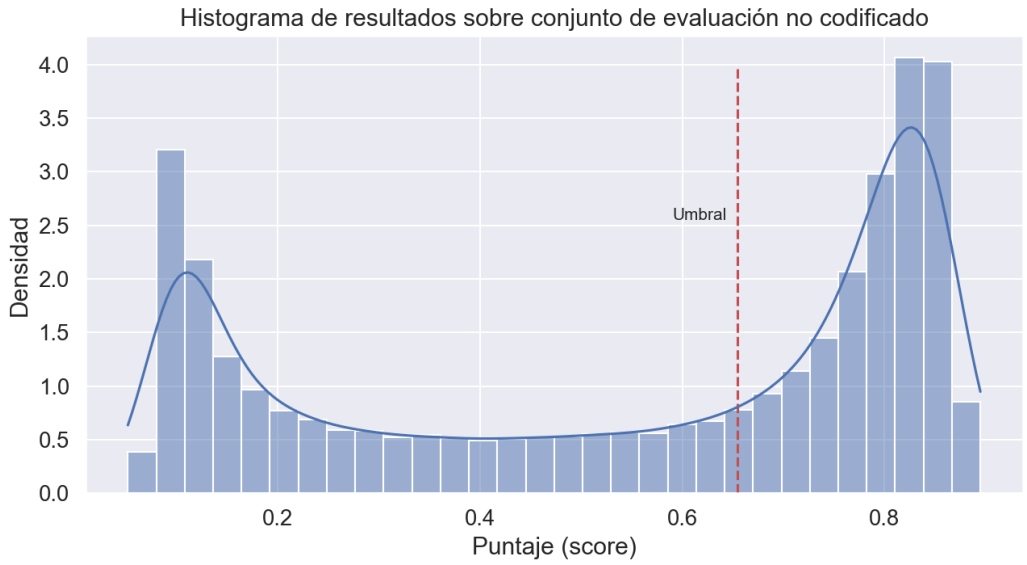

Figura 8: Distribución de la salida del modelo en los datos finales de evaluación. La línea punteada muestra el umbral elegido.

correspondiente al EER que es la métrica principal que uso en este trabajo y donde el costo por clasificar incorrectamente un idioma o el otro es equivalente. Siguiendo este esquema, el umbral elegido es la media de los umbrales resultantes del modelo elegido cuando fue evaluado anteriormente. El mismo es 0.655 y puede apreciarse en la Figura 8. El resultado al aplicar este umbral está expresado en la Tabla 4. Puede leerse que sobre el total de 61.21 horas, la cantidad total de habla de cada lengua es muy similar con una leve diferencia a favor del qom. Recordemos que los datos de evaluación contienen habla electrónica procedente de la televisión y la radio que se escucha de forma casi constante en las casas y sólo se emite en español.

	Español	**Qom**	Total
Umbral	30.39 (49.7 %)	30.82 (50.3 %)	61.21
Mayor certeza	18.22 (39 %)	28.34 (61 %)	46.57 (76 %)

Tabla 4: Cantidad de horas de habla en cada idioma para los dos niveles de umbral propuestos

Otra aproximación a este problema es medir sólo aquellos segmentos donde el modelo tiene mayor certeza. Para eso tomé los segmentos puntuados entre 0 y 0.3 como español y entre 0.7 y 1.0 como qom. Este dato también los expresé en la segunda línea de la Tabla 4 y corresponden a 76 % del corpus de evaluación, en otras palabras, 3 cuartas partes del habla sin codificar pueden ser clasificadas con un nivel de certeza más aceptable y en este caso el idioma qom supera al español en un 55 %. Este resultado puede interpretarse de dos maneras, o bien que efectivamente hay mayor cantidad de habla en qom en el corpus de datos o que el modelo tiene mayor certeza al clasificar este idioma. Dado que el modelo

wav2vec 2.0 elegido no fue preentrenado con habla en qom y que por otro lado existe una gran cantidad de habla electrónica en español, me inclino a pensar que el corpus contiene una porción significativamente más grande de habla de la comunidad en idioma qom.

3.3. Otros experimentos

Durante el desarrollo de este estudio también se llevaron a cabo varios experimentos que finalmente no sirvieron directamente al mismo. En un primer lugar, como comencé este proyecto en 2019 todavía no habían aparecido modelos preentrenados de las características del wav2vec 2.0. En ese momento todavía aparecían trabajos que comparaban entre modelos basados en *i-vectors* [37], modelos de aprendizaje profundo y fusiones entre ambos. Comencé algunos experimentos con *i-vectors* usando primero la librería Kaldi y luego la librería Bob para Python. Si bien pude obtener algunos resultados el marco teórico de *i-vectors* es abrumadoramente distinto y teniendo el objetivo de especializarme en redes neuronales profundas no tenía sentido hacer divergir mis estudios. Me enfoqué entonces en estudiar distintas arquitecturas de redes para diseñar el clasificador, en particular desarrollé múltiples modelos con capas densas, convolucionales y recurrentes donde los hiperparámetros utilizados eran la cantidad, los tipos y el tamaño de las capas, los optimizadores, los reguladores y los parámetros de los descriptores de entrada MFCC. De modelo base utilizaba un SVM. Con el conjunto de datos que tenía en ese momento -que no era la versión final- las redes con sólo capas convolucionales se mostraban más eficientes, veloces y con menos *overfitting* que las demás arquitecturas. Como comento en la Sección 2.5, en ese momento la literatura sugería utilizar *bottleneck features* (BNF, descriptores de cuello de botella) que se basaban en entrenar un clasificador de fonemas con una lengua de altos recursos (en general inglés) y luego extraer las salidas de una capa intermedia. Es por eso que desarrollé también algunos ensayos utilizando los descriptores generados por el modelo BUT/Phonexia [24]. La eficacia de estos descriptores no era provechosa para este problema, posiblemente por la diferencia de canal e idioma. En esta misma línea también también probé un clasificador universal de fonos llamado Allosaurus [38] que tampoco dio buenos resultados.

Durante este momento evaluaba los modelos que iba generando usando cinco semillas aleatorias que producían cinco subconjuntos de entrenamiento y evaluación. Más adelante, a partir de discutir esto con mi director, surgió la idea de crear los conjuntos estratificado y por familia, y *k-folding* para evaluar correctamente.

Poco después apareció wav2vec 2.0, dejé de usar Keras que es la librería con la que había desarrollado todo lo anterior y comencé a estudiar PyTorch. Para este momento en la literatura ya había varios extractores de descriptores [39, 40, 41], sin embargo wav2vec 2.0 era el primero en utilizar eficazmente capas *transformer* y ponía a disposición su librería y modelos pre-entrenados con miles de horas de habla en distintas lenguas.

Las primeras pruebas consistieron en utilizar los modelos preentrenados sin ajuste fino y elegir qué capas intermedias o mezclas de las mismas eran relevantes para esta tarea [42] utilizando un hiperparámetro. Se trató de un proceso cualitativo donde obtenía las variables latentes, las visualizaba con t-SNE coloreando los puntos por idioma o por hablante y luego escuchaba. Este experimento sugirió que el sistema era mejor en la tarea de separar hablantes que en la de identificar idiomas ya que los mismos aparecían cercanos en el espacio latente. Finalmente, dado que el ajuste fino terminó siendo la técnica más relevante para que el sistema funcione, el proceso de seleccionar capas intermedias fue descartado.

Con respecto al ajuste fino, la tasa de aprendizaje resultó un hiperparámetro fundamental a optimizar. Comencé probando con tasas en el orden de 10^{-2} hasta 10^{-5} y cuando los experimentos no daban los resultados esperados y discusiones mediante con distintos profesionales, el problema es que tenía que utilizar tasas aún más bajas con 10^{-6} siendo la óptima encontrada. Es aparente que es necesario utilizar tasas más bajas que lo común para conseguir que el ajuste fino de buenos resultados.

Otras pruebas que no quedaron especificadas en este trabajo incluyen la búsqueda de un módulo de clasificación. Comencé utilizando redes más grandes con capas densas pero las mismas resultaban en un *overfitting* difícil de controlar. Finalmente llegamos al clasificador actual que es más liviano y sencillo para que la comparación de modelos sea más justa y adecuada.

Por último el aumento de datos *(data augmentation)* es una técnica con la que también logré hacer distintas pruebas pero nunca conseguí resultados notables. Las mismas consistieron en probar los distintos tipos enumerados en la Sección 2.7.2 por separado pero tampoco pude llegar a buen puerto. Incluso tomé muestras de los audios procesados de esta manera y los escuché por separado y si bien a mi oído humano le hacían sentido, el sistema no mejoraba e incluso empeoraba. Mi hipótesis es que los datos son demasiado ruidosos y estos procesos sólo suman más ruido, aunque también quedaron espacios de hiperparámetros por explorar como para dar un diagnóstico más contundente.

4. Conclusiones

El primer eje de este trabajo es el diseño de una heurística para codificar los datos de entrenamiento de manera eficiente. El esquema propuesto (Sección 2.4.3) logra reducir sustancialmente los recursos humanos y económicos necesarios para efectuar esta tarea y es una herramienta clave para realizar otras tareas similares en el futuro. Es posible mejorarla si implementamos la clasificación de habla electrónica en el sistema de diarization ya que es muy común que los segmentos sugeridos contengan solamente un dispositivo electrónico encendido como una TV o una radio.

El segundo eje es la comparación de modelos con arquitectura wav2vec 2.0 y el modelo

base. Vimos como los modelos wav2vec 2.0 obtienen una eficacia superior y una muy marcada ventaja en capacidad de generalización cuando se realiza ajuste fino. Dada la gran cantidad de horas de habla en español con que fueron entrenados, esperaría que la clasificación en este idioma sea superior. Incluso en la Figura 8 se puede observar que el qom es clasificado con una certeza superior. Mi hipótesis es que factores como la variabilidad de canal y el dialecto están quitando robustez al sistema por lo que sería interesante reentrenar estos modelos con datos situados no codificados y luego también hacer ajuste fino con la tarea de clasificación.

El tercer eje es obtener una estimación de la cantidad de horas de habla en cada lengua. El trabajo revela que el entorno lingüístico de lxs niñxs participantes posee 55 % más de habla en qom con respecto a español cuando se toma la medida de mayor certeza. Este es un dato cuantitativo altamente relevante ya que se trata de niños que por su temprana edad todavía no han ingresado el sistema escolar. En el contexto escolar el español ha sido siempre la lengua mayormente empleada, aún cuando se trata de comunidades indígenas bilingües. Conocer el grado de bilingüismo de lxs niñxs que ingresan a la escuela resulta fundamental para poder diseñar mejores propuestas educativas. El equipo de investigación que dirige Celia Rosemberg y en el que se enmarca esta tesis, lleva a cabo desde hace varios años numerosas intervenciones, producción de material, capacitación docente y propuestas educativas así como evaluaciones de comprensión y producción de lenguaje. Los resultados de la presente tesis, tanto en relación con la cantidad de habla que los niños escuchan en cada lengua, como con la posibilidad de poder analizar de forma automática datos de habla en entornos naturales se inscriben en esta línea de investigación que no sólo da cuenta de las características de los entornos si no que a partir de esas características busca proponer mejores herramientas educativas.

4.1. Trabajo futuro

En la Sección 3.1 mostré que en este problema los modelos con ajuste fino basados en wav2vec 2.0 generalizan mejor hacia fuera del dominio cuando son comparados con el modelo base y sólo obtienen una ganancia marginal cuando los datos de entrenamiento y evaluación pertenecen al mismo dominio e incluso cuando los mismos no cuentan con el proceso de ajuste fino. Mi hipótesis es que los datos de preentrenamiento son tan diferentes en canal que sólo se puede aprovechar un poco de lo aprendido por transferencia. Sería interesante aprovechar la gran cantidad de horas en el corpus sin codificar para hacer un ajuste fino del modelo usando el mismo tipo de entrenamiento que el modelo original, esto es, sin modificar la tarea a la clasificación. Esto podría resultar en un aprendizaje a partir del dominio real en que será usado el modelo y podría verse una mejora substancial.

En este sentido, también podrían compararse no solo wav2vec 2.0 sino otros modelos y arquitecturas que existen de tecnología del habla usando herramientas nuevas como s3prl

[43] que mediante una interfaz unificada permite usar variedad de modelos sin que haga falta programar módulos personalizados para cada instancia.

Por otro lado, las grabaciones del corpus poseen muchas horas de habla electrónica (radio, TV, tablet, etc). Sería interesante reajustar el modelo de *diarization* y sumarle esa categoría. Esto permitirá hacer análisis por separado para cada tipo de habla -electrónica y humana- ya que tiene características distintas. Esto también haría más eficiente la selección de segmentos de alta densidad de habla ya que permitiría elegir aquellos sin habla electrónica.

La posibilidad de analizar de forma automática grandes corpus de audios en los hogares de niños qom resulta indispensable a la hora de conocer en mayor profundidad los contextos de desarrollo de estos niños. La investigación psicolingüística mostró que la cantidad del input lingüístico en las experiencias tempranas impactan en el desarrollo del lenguaje y de la alfabetización [44], es por ello que el estudio del habla al que están expuestos los niños en contextos naturales es fundamental. Sin embargo, el análisis del input de forma manual de una lengua indígena, que requiere el trabajo de investigadores y hablantes de la lengua, demanda una gran cantidad de tiempo y recursos humanos y económicos que imposibilitan el trabajo con corpus de datos extensos que permitirían tener un panorama "real" de la cantidad de habla en cada lengua. Poder utilizar esta herramienta en este y otros corpus, aportará un gran conocimiento en el área no solo de la psicolingüística sino también de la educación. En efecto, el conocimiento de las características del input lingüístico al que lxs niñxs estuvieron expuestxs en sus experiencias tempranas es necesario para diseñar programas educativos bilingües que se adecuen a las necesidades y realidades de dichos contextos y puedan promover el bilingüísmo, contribuyendo así al mantenimiento y revitalización de la lengua [45].

En esta línea, hay una variedad de estudios que esta primera aproximación hace posible llevar a cabo. Por un lado realizar un análisis longitudinal para examinar si hay diferencias en el entorno lingüístico desde etapas previas al ingreso a la escolaridad hasta avanzadxs en la misma. Por otro lado, estudiar qué hablantes producen cada lengua, cómo es la composición y las interacciones, tanto entre niñxs como en adultos, examinar cómo las experiencias lingüísticas van variando entre comunidades y familias. Por último, si bien este sistema fue utilizado para clasificar lenguas, con mínimos cambios se podría utilizar la misma arquitectura para clasificar otras variables: madurez vocálica, emociones, habla dirigida al niñx, habla entre adultxs, etc.

4.2. Palabras finales

En este escrito he logrado volcar una descripción del trabajo completo que llevé a cabo para estudiar este problema. Se trató de un desafío complejo por diversos motivos. Primero claro está por la naturaleza de los datos: grabaciones con un nivel de ruido excesivo y

con dinámicas de gran variabilidad, pero por otro lado por la dificultad de obtener esas grabaciones, codificarlas y procesarlas. Se trata de un trabajo que fue hecho a lo largo de 3 años (pandemia mediante) y cuyo campo de estudio evoluciona constantemente: las estrategias con las que se encara este tipo de proyecto de aprendizaje automático se modificaron significativamente desde que comencé hasta ahora, esto resultó en que mi plan tuviera que modificarse sustancialmente en la mitad del desarrollo o el mismo no hubiese resultado relevante. Algunas de estas cavilaciones aparecen desplegadas a lo largo de la Sección 2 de este escrito.

Me quedo sin duda con un enorme bagaje de aprendizaje en esta disciplina donde lo complejo del trabajo también motivó el análisis profundo del problema y llevó a tomar varias decisiones que en un *problema de juguete* no hubieran aparecido. En este sentido también rescato aquellos caminos que finalmente se mostraron errados por lo que hubo que volver todo para atrás y seguir adelante, pero que si no los hubiera tomado sólo me hubiesen quedado dudas de que hubiera sido.

Finalmente sólo me queda la motivación de seguir aprendiendo las millones de cosas que descubrí que no sabía, usando el conocimiento que alguna vez pensé que no era capaz de adquirir y que espero poder transmitir y facilitar con la integridad y espíritu crítico que en esta disciplina que está transformando nuestros futuros son fundamentales de mantener.

Referencias

[1] Christopher Cieri, Mike Maxwell, Stephanie Strassel y Jennifer Tracey. "Selection Criteria for Low Resource Language Programs". En: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), mayo de 2016, págs. 4543-4549. URL: https://aclanthology.org/L16-1720 (visitado 26-05-2022).

[2] Celia Rosemberg, Florencia Alam, Alejandra Stein, Maia Migdalek, Alejandra Menti y Gladys Ojea. *El entorno lingüístico de niños pequeños argentinos*. CONICET. 2015.

[3] Nereyda Hurtado, Virginia A. Marchman y Anne Fernald. "Does input influence uptake? Links between maternal talk, processing speed and vocabulary size in Spanish-learning children". eng. En: *Developmental Science* 11.6 (nov. de 2008), F31-39. ISSN: 1467-7687. DOI: 10.1111/j.1467-7687.2008.00768.x.

[4] Poliana Gonçalves Barbosa, Cláudia Cardoso-Martins y Catharine H. Echols. "Child-directed speech and its impact on early vocabulary acquisition: Evidence from Brazilian Portuguese". En: *Psychology & Neuroscience* 9.3 (2016), págs. 326-339. ISSN: 1983-3288(Electronic),1984-3054(Print). DOI: 10.1037/pne0000058.

[5] Marisa Casillas, Andrei Amatuni, Amanda Seidl, Melanie Soderstrom, Anne Warlaumont y Elika Bergelson. "What do Babies Hear? Analyses of Child- and Adult-Directed Speech". En: ago. de 2017, págs. 2093-2097. DOI: 10.21437/Interspeech.2017-1409.

[6] Theres Gruter y Paradis Johanne. *Input and Experience in Bilingual Development*. English. John Benjamins Publishing Company. ISBN: 978-90-272-4406-2.

[7] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed y Michael Auli. "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations". En: *arXiv:2006.11477 [cs, eess]* (sep. de 2020). arXiv: 2006.11477. URL: http://arxiv.org/abs/2006.11477 (visitado 06-10-2020).

[8] Vijayaditya Peddinti, Daniel Povey y Sanjeev Khudanpur. "A time delay neural network architecture for efficient modeling of long temporal contexts". en. En: *Interspeech 2015*. ISCA, sep. de 2015, págs. 3214-3218. DOI: 10.21437/Interspeech.2015-647. URL: https://www.isca-speech.org/archive/interspeech_2015/peddinti15b_interspeech.html (visitado 03-08-2022).

[9] Mark VanDam, D. Kimbrough Oller, Sophie E. Ambrose, Sharmistha Gray, Jeffrey A. Richards, Dongxin Xu, Jill Gilkerson, Noah H. Silbert y Mary Pat Moeller. "Automated Vocal Analysis of Children With Hearing Loss and Their Typical and

Atypical Peers". eng. En: *Ear and Hearing* 36.4 (ago. de 2015), e146-152. ISSN: 1538-4667. DOI: 10.1097/AUD.0000000000000138.

[10] Melinda Caskey y Betty Vohr. "Assessing language and language environment of high-risk infants and children: a new approach". eng. En: *Acta Paediatrica (Oslo, Norway: 1992)* 102.5 (mayo de 2013), págs. 451-461. ISSN: 1651-2227. DOI: 10.1111/apa.12195.

[11] Steven F. Warren, Jill Gilkerson, Jeffrey A. Richards, D. Kimbrough Oller, Dongxin Xu, Umit Yapanel y Sharmistha Gray. "What automated vocal analysis reveals about the vocal production and language learning environment of young children with autism". eng. En: *Journal of Autism and Developmental Disorders* 40.5 (mayo de 2010), págs. 555-569. ISSN: 1573-3432. DOI: 10.1007/s10803-009-0902-5.

[12] Marvin Lavechin, Ruben Bousbib, Hervé Bredin, Emmanuel Dupoux y Alejandrina Cristia. "An open-source voice type classifier for child-centered daylong recordings". En: arXiv:2005.12656 [eess]. arXiv, 2020. DOI: 10.48550/arXiv.2005.12656. URL: http://arxiv.org/abs/2005.12656 (visitado 03-08-2022).

[13] Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz y Marie-Philippe Gill. "pyannote.audio: neural building blocks for speaker diarization". En: *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing*. arXiv:1911.01255 [cs, eess]. Barcelona, Spain: arXiv, mayo de 2020. DOI: 10.48550/arXiv.1911.01255. URL: http://arxiv.org/abs/1911.01255 (visitado 03-08-2022).

[14] Mirco Ravanelli y Yoshua Bengio. "Speaker Recognition from Raw Waveform with SincNet". En: *arXiv:1808.00158 [cs, eess]* (ago. de 2019). arXiv: 1808.00158. URL: http://arxiv.org/abs/1808.00158 (visitado 11-02-2020).

[15] Leandro Garber, Tomás Ciccola y Juan Cruz Amusategui. "AudioStellar, an open source corpus-based musical instrument for latent sound structure discovery and sonic experimentation". en. En: Santiago de Chile, 2020, pág. 6.

[16] Brian MacWhinney. "CLAN Manual". En: *The CHILDES Project: Tools for Analyzing Talk. 3rdEdition. Mahwah, NJ: Lawrence Erlbaum Associates* (2000). Publisher: TalkBank. DOI: 10.21415/T5G10R. URL: https://talkbank.org/manuals/CLAN.pdf (visitado 14-10-2020).

[17] Seyed Omid Sadjadi y John H. L. Hansen. "Unsupervised Speech Activity Detection Using Voicing Measures and Perceptual Spectral Flux". en. En: *IEEE Signal Processing Letters* 20.3 (mar. de 2013), págs. 197-200. ISSN: 1070-9908, 1558-2361. DOI: 10.1109/LSP.2013.2237903. URL: http://ieeexplore.ieee.org/document/6403507/ (visitado 11-07-2018).

[18] Ali Ziaei, Lakshmish Kaushik, Abhijeet Sangwan, John Hansen y Doug Oard. *Speech Activity Detection for NASA Apollo Space Missions: Challenges and Solutions*. Journal Abbreviation: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH Publication Title: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. Sep. de 2014. DOI: 10.21437/Interspeech.2014-369.

[19] Leandro Garber. *CHAFile*. Feb. de 2021. DOI: 10.5281/zenodo.4557861. URL: https://zenodo.org/record/4557861 (visitado 05-08-2022).

[20] *ELAN (Version 6.0) [Computer software]*. (2020). Nijmegen: Max Planck Institute for Psycholinguistics, The Language Archive. Nijmegen: Max Planck Institute for Psycholinguistics, 2020. URL: https://archive.mpi.nl/tla/elan (visitado 12-05-2021).

[21] Mart Lubbers y Francisco Torreira. *pympi-ling: a Python module for processing ELANs EAF and {Praat}s TextGrid annotation files*. 2013. URL: https://pypi.python.org/pypi/pympi-ling.

[22] Pavel Matejka, Le Zhang, Tim Ng, Sri Harish Mallidi, Ondrej Glembek, Jeff Ma y Bing Zhang. "Neural Network Bottleneck Features for Language Identification". en. En: jun. de 2014, pág. 6.

[23] Radek Fér, Pavel Matějka, František Grézl, Oldřich Plchot, Karel Veselý y Jan Honza Černocký. "Multilingually trained bottleneck features in spoken language recognition". en. En: *Computer Speech & Language* 46 (nov. de 2017), págs. 252-267. ISSN: 08852308. DOI: 10.1016/j.csl.2017.06.008.

[24] Anna Silnova, Pavel Matejka, Ondrej Glembek, Oldrich Plchot, Ondrej Novotny, František Grézl, Petr Schwarz, Lukas Burget y Jan Cernocky. "BUT/Phonexia Bottleneck Feature Extractor". En: jun. de 2018, págs. 283-287. DOI: 10.21437/Odyssey.2018-40.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser e Illia Polosukhin. "Attention Is All You Need". En: *arXiv:1706.03762 [cs]* (dic. de 2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762 (visitado 24-11-2021).

[26] Thomas Wolf y col. "Transformers: State-of-the-Art Natural Language Processing". En: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, oct. de 2020, págs. 38-45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6 (visitado 03-08-2022).

[27] M.Y. Hwang y X. Huang. "Subphonetic modeling with Markov states-Senone". En: *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. ISSN: 1520-6149. Mar. de 1992, 33-36 vol.1. DOI: 10.1109/ICASSP.1992.225979.

[28] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov y Abdelrahman Mohamed. "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units". En: *arXiv:2106.07447 [cs, eess]* (jun. de 2021). arXiv: 2106.07447. URL: http://arxiv.org/abs/2106.07447 (visitado 24-11-2021).

[29] Changhan Wang, Morgane Rivière, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino y Emmanuel Dupoux. "VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation". En: *arXiv:2101.00390 [cs, eess]* (ene. de 2021). arXiv: 2101.00390. URL: http://arxiv.org/abs/2101.00390 (visitado 12-05-2021).

[30] Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed y Michael Auli. *Unsupervised Cross-lingual Representation Learning for Speech Recognition*. Number: arXiv:2006.13979 arXiv:2006.13979 [cs, eess]. Dic. de 2020. DOI: 10.48550/arXiv.2006.13979. URL: http://arxiv.org/abs/2006.13979 (visitado 20-06-2022).

[31] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers y Gregor Weber. *Common Voice: A Massively-Multilingual Speech Corpus*. Number: arXiv:1912.06670 arXiv:1912.06670 [cs]. Mar. de 2020. DOI: 10.48550/arXiv.1912.06670. URL: http://arxiv.org/abs/1912.06670 (visitado 20-06-2022).

[32] M. J. F. Gales, K. M. Knill, A. Ragni y S. P. Rath. *Speech recognition and keyword spotting for low-resource languages : Babel project research at CUED*. en. Proceedings Paper. Conference Name: Fourth International Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU-2014) Meeting Name: Fourth International Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU-2014) Pages: 16-23 Place: St. Petersburg, Russia Publisher: International Speech Communication Association (ISCA). Mayo de 2014. URL: https://www.isca-speech.org/archive/sltu_2014/sl14_016.html (visitado 03-08-2022).

[33] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve y Ronan Collobert. "MLS: A Large-Scale Multilingual Dataset for Speech Research". En: *Interspeech 2020*. arXiv:2012.03411 [cs, eess]. Oct. de 2020, págs. 2757-2761. DOI: 10.21437/Interspeech.2020-2826. URL: http://arxiv.org/abs/2012.03411 (visitado 03-08-2022).

[34] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey y Sanjeev Khudanpur. "X-Vectors: Robust DNN Embeddings for Speaker Recognition". en. En: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, AB: IEEE, abr. de 2018, págs. 5329-5333. ISBN: 978-1-5386-4658-8. DOI: 10.1109/ICASSP.2018.8461375. URL: https://ieeexplore.ieee.org/document/8461375/ (visitado 25-05-2020).

[35] Edward Ma. *NLP Augmentation*. 2019. URL: https://github.com/makcedward/nlpaug.

[36] Navdeep Jaitly y Geoffrey E Hinton. "Vocal Tract Length Perturbation (VTLP) improves speech recognition". en. En: (), pág. 5.

[37] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel y Pierre Ouellet. "Front-End Factor Analysis for Speaker Verification". en. En: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4 (mayo de 2011), págs. 788-798. ISSN: 1558-7916, 1558-7924. DOI: 10.1109/TASL.2010.2064307. URL: http://ieeexplore.ieee.org/document/5545402/ (visitado 14-05-2018).

[38] Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anastasopoulos, David R. Mortensen, Graham Neubig, Alan W. Black y Florian Metze. "Universal Phone Recognition with a Multilingual Allophone System". en. En: *arXiv:2002.11800 [cs, eess]* (feb. de 2020). arXiv: 2002.11800. URL: http://arxiv.org/abs/2002.11800 (visitado 02-06-2020).

[39] Shaoshi Ling y Yuzong Liu. "DeCoAR 2.0: Deep Contextualized Acoustic Representations with Vector Quantization". En: *arXiv:2012.06659 [cs, eess]* (dic. de 2020). arXiv: 2012.06659. URL: http://arxiv.org/abs/2012.06659 (visitado 22-03-2021).

[40] Shaoshi Ling, Julian Salazar, Yuzong Liu y Katrin Kirchhoff. "BERTphone: Phonetically-aware Encoder Representations for Utterance-level Speaker and Language Recognition". en. En: *Odyssey 2020 The Speaker and Language Recognition Workshop*. ISCA, nov. de 2020, págs. 9-16. DOI: 10.21437/Odyssey.2020-2. URL: http://www.isca-speech.org/archive/Odyssey_2020/abstracts/93.html (visitado 22-03-2021).

[41] Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal y Yoshua Bengio. "Multi-task self-supervised learning for Robust Speech Recognition". En: *arXiv:2001.09239 [cs, eess]* (abr. de 2020). arXiv: 2001.09239. URL: http://arxiv.org/abs/2001.09239 (visitado 19-11-2020).

[42] Leonardo Pepino, Pablo Riera y Luciana Ferrer. "Emotion Recognition from Speech Using Wav2vec 2.0 Embeddings". En: *arXiv:2104.03502 [cs, eess]* (abr. de 2021). arXiv: 2104.03502. URL: http://arxiv.org/abs/2104.03502 (visitado 09-11-2021).

[43] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko-tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed y Hung-yi Lee. "SUPERB: Speech Processing Universal PERformance Benchmark". En: *Proc. Interspeech 2021*. 2021, págs. 1194-1198. DOI: 10.21437/Interspeech.2021-1775.

[44] E Hoff. "How social contexts support and shape language development ". en. En: *Developmental Review* 26.1 (mar. de 2006), págs. 55-88. ISSN: 02732297. DOI: 10.1016/j.dr.2005.11.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S0273229705000316 (visitado 06-10-2022).

[45] Zurer Pearson y Luiz Amaral. "Interactions between input factors in bilingual language acquisition". En: *Input and Experience in Bilingual Development*. Ene. de 2014, págs. 99-117. DOI: 10.1075/tilar.13.06pea.