

BATS: Bridging Acoustic Transparency in Speech

Diego Quezada
Departamento de Informática
Universidad Técnica Federico Santa María
Valparaíso, Chile
diego.quezadac@sansano.usm.cl

Felipe Cisternas
Departamento de Informática
Universidad Técnica Federico Santa María
Valparaíso, Chile
felipe.cisternasal@sansano.usm.cl

Resumen—El reconocimiento de voz se basa en representaciones de señales acústicas, como espectrogramas y MFCCs. Sin embargo, los modelos actuales son en gran medida opacos en cuanto a cómo toman decisiones en este proceso. La naturaleza física de los datos de entrada en el reconocimiento de voz agrega una capa adicional de complejidad, lo que plantea el desafío de mejorar la transparencia y la comprensión de estos modelos para garantizar un reconocimiento de voz más preciso y confiable.

Index Terms—ASR, XAI, CNN, RNN, Transformers

I. Introducción

En el ámbito del reconocimiento de voz, la selección de un modelo y su arquitectura es fundamental para enfrentar los retos específicos que esta disciplina impone. Modelos de vanguardia como Whisper, que se basan en la arquitectura de Transformers de OpenAI, son altamente eficaces, pero actúan como cajas negras, lo que significa que los procesos de decisión que siguen para convertir señales de audio en palabras no son transparentes. Esta falta de explicabilidad, esencial para generar confianza en los usuarios, hace imperativo el uso de técnicas de eXplainable Artificial Intelligence (XAI) para desentrañar y comprender cómo estos modelos avanzados toman sus decisiones.

La explicabilidad en modelos de reconocimiento de voz no solo mejora la comprensión del proceso de toma de decisiones, sino que también es clave para aumentar su robustez, especialmente cuando dichos modelos se integran como componentes centrales en un software. Esta característica se vuelve esencial para ganar la confianza de los usuarios en el producto final. Sin embargo, el desafío se intensifica debido a la naturaleza física y compleja de los datos de entrada, lo que dificulta proporcionar explicaciones claras y de alto nivel basados en estos datos. Por ello, es crucial mantener un equilibrio entre avanzar en la precisión y eficiencia de los modelos y desarrollar soluciones que mejoren su explicabilidad, asegurando así un balance óptimo entre rendimiento y comprensión del modelo.

Esta investigación tiene como objetivo mejorar la comprensión de los modelos de reconocimiento de voz para que

los usuarios, especialmente aquellos con discapacidades auditivas, puedan confiar en su funcionamiento.

I-A. Trabajo relacionado

Las investigaciones sobre explicabilidad en modelos de reconocimiento de voz son escasas, pero destacan por su aplicación innovadora de métodos de explicabilidad.

En [1] se aborda la tarea reconocimiento de voz utilizando el conjunto de datos CommonVoice [2] y tres sistemas de reconocimiento de voz: Google API, Sphinx y DeepSpeech. Los autores plantean las explicaciones como un subconjunto de fotogramas de audio que son causas mínimas y suficientes de la transcripción. Para esto, adaptan las técnicas de clasificación de imagen SFL [3] y explicaciones composicionales [4] para luego compararlas con las ofrecidas por LIME [5].

En [6] se aborda la tarea de reconocimiento de fonemas utilizando el conjunto de datos TIMIT [7] y el método LIME junto a dos variantes propuestas: LIME-WS y LIME-TS. El conjunto de datos TIMIT segmenta los distintos fonemas en los datos de entrada. Los autores utilizan como representación interpretable un vector que indica la presencia o ausencia de cada segmento de audio. El objetivo es identificar los segmentos más importantes para la predicción del siguiente fonema. Inicialmente, para la generación del vecindario se aplican máscaras aleatorias a los segmentos de TIMIT. Luego, considerando que al analizar un fonema es probable que solo los segmentos cercanos incidan en su identificación, LIME-WS solo aplica máscaras aleatorias a segmentos cercanos al fonema analizado y mantiene los lejanos constantes. Finalmente, en LIME-TS se consideran segmentos creados por un intervalo de tiempo en vez de los definidos por TIMIT y se aplica la misma idea de localidad que en LIME-WS.

En [8] se aborda la tarea de etiquetado de música. Los autores plantean la necesidad de dar explicaciones audibles y proponen audioLIME como una extensión de LIME que utiliza explicaciones basadas en separación de fuentes.

II. Método propuesto

II-A. Marco teórico

El reconocimiento de voz, o más conocido en inglés como speech recognition, es la tarea de asignar una secuencia de palabras a señales acústicas que contienen lenguaje hablado. Implica reconocer las palabras pronunciadas en una grabación de audio y transcribirlas a un formato escrito. El objetivo es transcribir con precisión el discurso en tiempo real o a partir de audio grabado, teniendo en cuenta factores como el acento, la velocidad del habla y el ruido de fondo. Cuando la transcripción se realiza en tiempo real se habla de reconocimiento automático de voz o Automatic Speech Recognition (ASR) en inglés. Considerando $\mathbf{X} = (x^{(1)}, x^{(2)}, \dots, x^{(T)})$ como una secuencia de audio de largo T e $\mathbf{y} = (y_1, y_2, \dots, y_N)$ como una secuencia de palabras de largo N podemos definir la tarea de reconocimiento de voz de manera precisa mediante el siguiente problema de optimización:

$$f^*(\mathbf{X}) = \arg \max_{\mathbf{y}} P^*(\mathbf{y}|\mathbf{X} = \mathbf{X}) \quad (1)$$

Donde P^* es la verdadera distribución de probabilidad condicional que relaciona las entradas \mathbf{X} con las salidas \mathbf{y} [9].

La representación utilizada para \mathbf{X} es de vital importancia para el desempeño de un modelo de reconocimiento de voz. La representación más simple es mediante una serie temporal univariada que modela la amplitud de la señal de audio en el tiempo. Al dividir la señal de audio en pequeñas ventanas de tiempo y calculando el espectro de frecuencia para cada ventana se obtiene un espectrograma: una representación visual de la señal de audio en el tiempo y en el dominio de la frecuencia. A partir del espectrograma se pueden extraer características de audio tales como los coeficientes cepstrales de Mel (MFCCs) que son ampliamente utilizados en la literatura para el reconocimiento de voz.

Para poder evaluar un modelo de ASR se utilizan métricas como:

- Word Error Rate (WER): La métrica Word Error Rate se calcula como el número total de errores, que es la suma de sustituciones, inserciones y eliminaciones de palabras necesarias para cambiar una secuencia de palabras hipotética a una secuencia de referencia, dividido por el número total de palabras en la secuencia de referencia. WER proporciona una medida de cuántas palabras fueron reconocidas incorrectamente en proporción al total de palabras habladas. La fórmula para calcular el Word Error Rate (WER) es:

$$\text{WER} = \frac{S + D + I}{N} \quad (2)$$

Donde:

- S es el número de sustituciones,
- D es el número de eliminaciones,
- I es el número de inserciones,
- N es el número de palabras en la secuencia de referencia.

WER se expresa a menudo como un porcentaje, y cuanto más bajo es el WER, mejor es el rendimiento del sistema de reconocimiento de voz.

- Match Error Rate (MER): La métrica Match Error Rate es similar a WER, pero en lugar de centrarse solo en las palabras, evalúa la precisión en el emparejamiento de cualquier tipo de elementos, como fonemas, letras o incluso palabras en tareas de reconocimiento. La fórmula para calcular MER es:

$$\text{MER} = \frac{S + D + I}{S + D + C} \quad (3)$$

Donde:

- S es el número de sustituciones,
- D es el número de eliminaciones,
- I es el número de inserciones,
- C es el número de aciertos correctos.

MER también se suele expresar en porcentaje y proporciona una medida de cuántos elementos fueron incorrectamente emparejados en relación con el total de elementos que debían emparejarse, cuanto más bajo es MER mejor es el rendimiento del modelo.

- Word Information Lost (WIL): La métrica Word Information Lost es una medida teórica de la información basada en la entropía desde la perspectiva de la información de las palabras que se pierden en la transcripción de la hipótesis comparada con la referencia. La tasa de WIL se puede calcular con la fórmula:

$$\text{WIL} = 1 - \frac{C}{N} + \frac{C}{P} \quad (4)$$

donde:

- C es el número de palabras correctas,
- N es el número de palabras en la referencia,
- P es el número de palabras en la predicción.

La métrica WIL está diseñada para estar acotada entre 0 y 1, proporcionando una forma normalizada de medir la información perdida en la transcripción. Un valor más bajo de WIL indica un mejor rendimiento del sistema de ASR, siendo 0 el puntaje perfecto. Es importante mencionar que, aunque WIL comparte similitudes con WER en cuanto a su uso para evaluar transcripciones de ASR, está diseñada para abordar ciertas limitaciones de WER al proporcionar una medida normalizada que está efectivamente limitada.

- Word Information Preserved (WIP): La métrica Word Information Preserved mide el porcentaje de palabras correctas predichas entre un conjunto de oraciones verdaderas y un conjunto de oraciones

hipotéticas. La fórmula para calcular la tasa de WIP es:

$$wip = \frac{C}{N} + \frac{C}{P} \quad (5)$$

donde:

- C es el número de palabras correctas,
- N es el número de palabras en la referencia,
- P es el número de palabras en la predicción.

Un valor más alto de WIP indica un mejor rendimiento del sistema ASR, siendo 1 la puntuación perfecta.

- Character Error Rate (CER): La métrica Character Error Rate se calcula de manera similar a WER, pero en lugar de considerar las palabras, se enfoca en los caracteres individuales. La fórmula para calcular CER es:

$$CER = \frac{S + D + I}{N} \quad (6)$$

Donde:

- S es el número de sustituciones de caracteres,
- D es el número de eliminaciones de caracteres,
- I es el número de inserciones de caracteres,
- N es el número total de caracteres en la secuencia de referencia.

Al igual que WER, el CER se expresa en porcentaje, y un valor más bajo indica un mejor rendimiento del sistema de reconocimiento.

LIME [5] es un modelo agnóstico que explica de manera local las predicciones de un modelo caja negra f mediante la aproximación de f con un modelo explicable g en una vecindad de la instancia de interés \mathbf{x} . LIME utiliza una representación interpretable $\mathbf{x}' \in \{0, 1\}^{d'}$ de la instancia $\mathbf{x} \in \mathbb{R}^d$. De esta forma, el dominio de la explicación g es $\{0, 1\}^{d'}$ y, por lo tanto, g actúa sobre la presencia o ausencia de componentes interpretables. Para generar la vecindad se utiliza una función $\pi_x(z)$ que mide la distancia entre dos puntos x y z en el espacio de atributos interpretable. Adicionalmente, dado que la idea es aproximar f mediante un modelo interpretable, se considera una penalización Ω que mide la complejidad de g . Finalmente, y considerando una función de pérdida $\mathcal{L}(f, g, \pi_x)$ que mide la imprecisión del modelo g al aproximar f en la vecindad de \mathbf{x} de acuerdo a la función de distancia π , la explicación producida por LIME que asegura interpretabilidad y fidelidad local se obtiene según la siguiente fórmula:

$$\xi(\mathbf{x}) = \arg \min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g). \quad (7)$$

Donde G es el conjunto de todos los modelos interpretables considerados, como por ejemplo modelos lineales o árboles de decisión.

El borrado de representaciones [10] (Representation Erasure) es un método post-hoc y agnóstico al modelo para entender y explicar las decisiones de un modelo de red

neuronal mediante la observación de los efectos que tiene el borrar distintas partes de la representación del modelo. Esto puede incluir dimensiones de vectores de palabras de entrada, unidades ocultas intermedias o palabras de entrada. Se utilizan varias técnicas para analizar los efectos de dicho borrado, como calcular la diferencia relativa en métricas de evaluación o usar aprendizaje por refuerzo para borrar el conjunto mínimo de palabras de entrada con el fin de cambiar la decisión de un modelo neural. Este método ayuda a ofrecer explicaciones claras sobre las decisiones de modelos neuronales y también facilita el análisis de errores en dichos modelos.

II-B. Conjunto de datos

En la presente investigación se utilizará el conjunto de datos CommonVoice [2]. En particular, se utilizarán las grabaciones en inglés asociadas a la versión 11 de CommonVoice disponible en HuggingFace Datasets.

Los conjunto de entrenamiento, validación y prueba consisten en 948736, 16354 y 16354 grabaciones de audio respectivamente. Cada grabación de audio tiene asociada una transcripción. Adicionalmente, cada grabación de audio fue grabada con un sampling rate de 48 kHz.

II-C. Descripción propuesta

La propuesta busca evaluar la explicabilidad de modelos estado del arte en la tarea de reconocimiento de voz mediante métodos post-hoc.

II-C1. SLIME: Para explicar las transcripciones del modelo Whisper, se propone el algoritmo SLIME (Speech LIME) que modela cómo diferentes modificaciones en la señal de audio afectan la calidad de la transcripción.

SLIME fragmenta una grabación de audio en segmentos de 100 milisegundos y considera como representación interpretable un vector booleano que indica la presencia o ausencia de cada segmento. Utilizando esta representación, se genera una vecindad de 100 instancias apagando segmentos de manera aleatoria. Cada instancia de la vecindad fue transcrita por Whisper, y mediante la distancia de Levenshtein entre la transcripción perturbada y la original se cuantificó el impacto de los silencios, proporcionando una base para una comparación cuantitativa entre las transcripciones. Finalmente, se ajustó un modelo explicable de regresión lineal considerando la representación interpretable de los audios como entrada y la distancia de Levenshtein como salida, lo cual permitió evaluar la importancia de los segmentos.

Debido a la alta demanda computacional que implica obtener predicciones de Whisper, se limitó la vecindad exploratoria a 100 combinaciones. Esto estableció un equilibrio entre la profundidad del análisis y las restricciones prácticas.

Los coeficientes resultantes de la regresión lineal ofrecen una interpretación directa de la relevancia de cada segmento de audio. Los segmentos asociados con coeficientes negativos son clave; su presencia es vital para mantener la fidelidad de la transcripción original, mientras que su ausencia es indicativa de un incremento en la distancia de Levenshtein, reflejando transcripciones que se desvían de la original. Esta información es útil para discernir las partes del audio que el modelo considera importantes para una transcripción precisa. El siguiente paso es mejorar la segmentación del audio, buscando métodos que puedan captar con mayor precisión las unidades lingüísticas significativas y mejorar la interpretación de las decisiones del modelo.

En 1 se resume el algoritmo propuesto para obtener las explicaciones de Whisper mediante LIME.

Algorithm 1 SLIME

```

1:  $S \leftarrow \emptyset$ 
2:  $f \leftarrow$  Black-box model
3:  $n \leftarrow$  Number of segments of 100 ms in audio
4:  $x \leftarrow$  Original audio instance
5:  $y \leftarrow f(x)$ 
6:  $x' \leftarrow$  Vector of  $n$  ones
7:  $P \leftarrow$  Generate 100 Binomial perturbations of  $x'$ 
8: for each  $p$  in  $P$  do
9:    $z \leftarrow$  Apply  $p$  to  $x$ 
10:   $d \leftarrow$  Levenshtein distance between  $y$  and  $f(z)$ 
11:   $z' \leftarrow$  Apply random mask to  $x'$ 
12:   $S \leftarrow S \cup (z', d)$ 
13: end for
14:  $g \leftarrow$  Fit linear model to  $S$ 
15: end algorithm

```

II-C2. Borrado de Representaciones: Adicionalmente, analizaremos los cambios en la salida del modelo Whisper al borrar ciertas partes de la representación de la señal de audio a través de MFCCs. Sea f un modelo neuronal de ASR, dado un ejemplo $x \in E$ con una transcripción y , calcularemos el score de una métrica M con la siguiente fórmula:

$$S_M(x, y) = M(y, f(x)) \quad (8)$$

Ahora sea d una dimensión de nuestro vector x la cual queremos explorar, denotaremos $S_M(x, y, \neg d)$ el score cuando la dimensión d es removida; esto quiere decir que todos sus valores son 0. La importancia de la dimensión d denotada por $I(d)$ estará dada por la diferencia relativa entre $S_M(x, y)$ y $S_M(x, y, \neg d)$:

$$I(d) = \frac{1}{|E|} \sum_{x \in E} \frac{S_M(x, y) - S_M(x, y, \neg d)}{S_M(x, y)} \quad (9)$$

Para esto utilizaremos las métricas mencionadas en el marco teórico y calcularemos la importancia de cada dimensión con cada métrica dada la diferencia relativa entre los outputs del modelo.

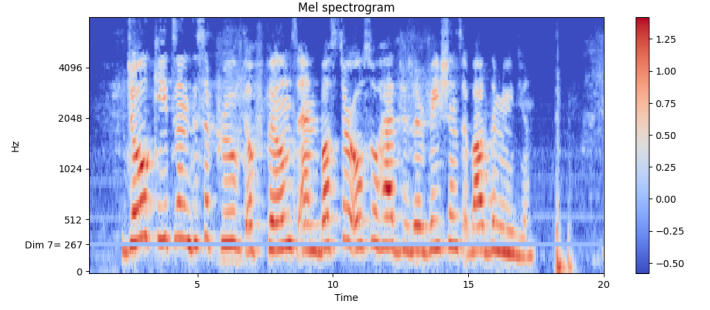


Figura 1. Espectrograma de MEL, al eliminar la dimensión 7 del MFCC, definiendo todos sus valores en 0.

III. Resultados preliminares

III-A. SLIME

Consideremos una grabación de audio x cuyo contenido es they perfectly illuminate the flowerbeds of gardens. Al aplicar SLIME a esta grabación se obtiene la explicación de la Figura 2.

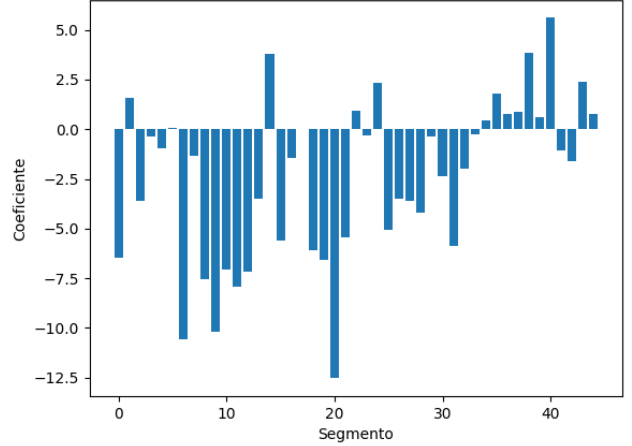


Figura 2. Explicación de SLIME

Como se mencionó anteriormente, un coeficiente negativo indican que un segmento es importante, pues al removerlo la transcripción se desvía de la original considerando la distancia de Levenshtein.

III-B. Borrado de representaciones

En la figura 3 podemos observar el cálculo de importancias para distintas métricas evaluadas en el set de test del dataset Common Voice 11, cada celda muestra la importancia de una dimensión (columna) de los MFCCs en cada métrica (fila) para el modelo entrenado.

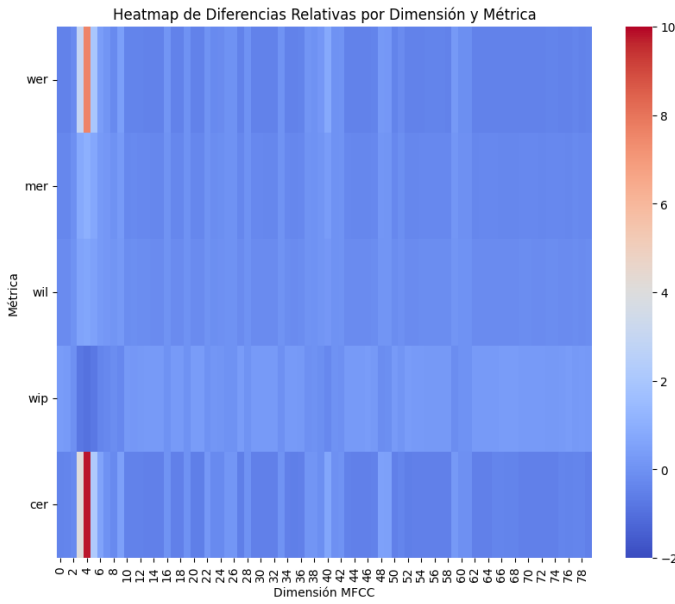


Figura 3. Mapa de calor de importancias $I(d)$ calculado usando la ecuación 9.

En este ejemplo se puede visualizar como la dimensión 4 es la más importante para la métrica WER y CER. Una importancia mayor implica que al remover esa dimensión las predicciones del modelo empeoran con respecto a la métrica evaluada, por ende es una dimensión importante para el modelo.

También podemos observar que hay importancias negativas, esto quiere decir que al eliminar esa dimensión las predicciones del modelo mejoran, lo que es un hallazgo muy interesante, ya que podríamos optimizar al modelo borrando ciertas dimensiones.

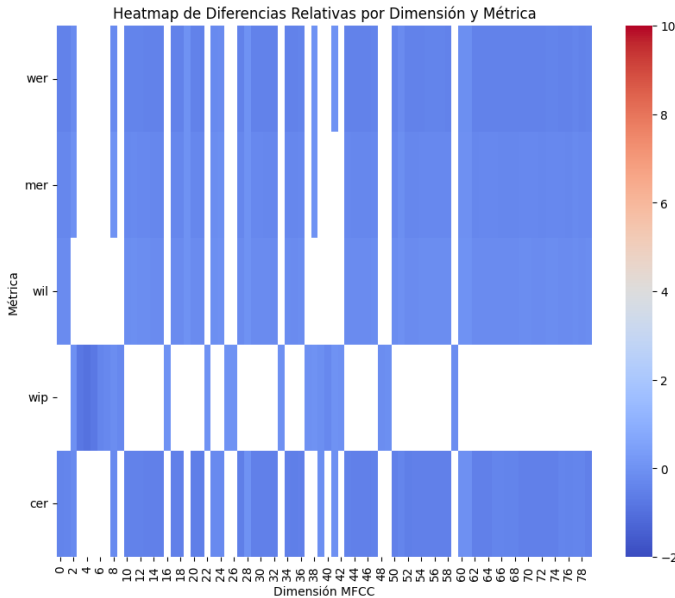


Figura 4. Mapa de calor de importancias $I(d)$ negativas.

Referencias

- [1] X. Wu, P. Bell y A. Rajan, «Explanations for Automatic Speech Recognition,» en ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023, págs. 1-5. doi: 10.1109/ICASSP49357.2023.10094635.
- [2] R. Ardila, M. Branson, K. Davis et al., «Common Voice: A Massively-Multilingual Speech Corpus,» en Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), 2020, págs. 4211-4215.
- [3] Y. Sun, H. Chockler, X. Huang y D. Kroening, Explaining Image Classifiers using Statistical Fault Localization, 2020. arXiv: 1908.02374 [cs.LG].
- [4] H. Chockler, D. Kroening e Y. Sun, Compositional Explanations for Image Classifiers, mar. de 2021.
- [5] M. T. Ribeiro, S. Singh y C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, 2016. arXiv: 1602.04938 [cs.LG].
- [6] X. Wu, P. Bell y A. Rajan, Can We Trust Explainable AI Methods on ASR? An Evaluation on Phoneme Recognition, 2023. arXiv: 2305.18011 [cs.CL].
- [7] J. Garofolo, L. Lamel, W. Fisher et al., «TIMIT Acoustic-phonetic Continuous Speech Corpus,» Linguistic Data Consortium, nov. de 1992.
- [8] V. Haunschmid, E. Manilow y G. Widmer, «audio-LIME: Listenable Explanations Using Source Separation,» CoRR, vol. abs/2008.00582, 2020. arXiv: 2008.00582. dirección: <https://arxiv.org/abs/2008.00582>.
- [9] I. Goodfellow, Y. Bengio y A. Courville, Deep Learning. MIT Press, 2016, [http : / / www . deeplearningbook.org](http://www.deeplearningbook.org).
- [10] J. Li, W. Monroe y D. Jurafsky, Understanding Neural Networks through Representation Erasure, 2017. arXiv: 1612.08220 [cs.CL].