UNIVERSIDAD CENTROAMERICANA

JOSÉ SIMEÓN CAÑAS

TRANSFER LEARNING PARA LA TRANSCRIPCIÓN DE CONFERENCIAS DE ÁREAS STEM

TRABAJO DE GRADUACIÓN PREPARADO PARA LA

FACULTAD DE INGENIERÍA Y ARQUITECTURA

PARA OPTAR AL GRADO DE

INGENIERO INFORMÁTICO

POR:

FERNANDO RENÉ CÁRCAMO AGUILAR

JOSUE DAVID HURTADO ARGUETA

RAFAEL ALEJANDRO MELARA RAMIREZ

FERNANDO ADONAY ROSA CARDOZA

OCTUBRE 2024,

ANTIGUO CUSCATLAN, EL SALVADOR, C.A.

RECTOR

MARIO ERNESTO CORNEJO MENA, S. J.

SECRETARIA GENERAL

LIDIA GABRIELA BOLAÑOS TEODORO

DECANO DE LA FACULTAD DE INGENIERÍA Y ARQUITECTURA

CARLOS ERNESTO RIVAS CERNA

DIRECTOR DE LA CARRERA DE INGENIERÍA INFORMÁTICA

JOSÉ ENMANUEL AMAYA ARAUJO

DIRECTORA DEL TRABAJO

SILVIA CAROLINA ORTIZ CUÉLLAR

LECTOR

RONALDO ARMANDO CANIZALES TURCIOS

AGRADECIMIENTOS

Mostramos nuestra gratitud a Silvia Ortiz por su guía y dirigir este trabajo de investigación, por su apoyo en establecer el plan de trabajo que dieron lugar a los pasos que tomamos para completar este trabajo y por su docencia durante la carrera. A Ronaldo Canizales por sus aportes y comentarios a este trabajo, por su apoyo educativo en esta área que en un inicio era desconocida para nosotros y también por su excelente docencia durante los primeros años de la carrera, que son en nuestra opinión, los más importantes. Al IDHUCA por inspirar este trabajo de investigación y demostrar interés en el uso de tecnologías novedosas para el cumplimiento de su misión, reconociendo también su aporte humanitario a la sociedad salvadoreña y a Mauricio Erazo por proponer este tema. Por último, agradecemos a todos los docentes que nos dieron los conocimientos y herramientas en temas de ciencias de la computación, programación, ingeniería de software y matemáticas que, sin ellos, este trabajo no hubiera sido posible.

DEDICATORIA

Gracias a Dios por su guía, fortaleza y bendición de poder estudiar una carrera que me apasiona tanto. A mi madre por todo su apoyo hasta donde se lo pudo permitir durante toda mi vida como estudiante, por su constante apoyo y sacrificio para que yo ahora pueda gozar de los frutos de su trabajo y por todas las herramientas brindadas para defenderme en la vida. A mi padre por ser la inspiración para elegir esta carrera que, desde pequeño, me tuvo en contacto con la computación, por las largas escuchas donde le contaba sobre los conocimientos que iba adquiriendo y por ser un incondicional apoyo para todas las cosas que necesité durante estos 5 años. A mis hermanos que nunca dudaron en reforzar los conocimientos en los momentos que los necesité y que hicieron mi paso por la universidad mucho más cómodo, en especial a mi hermana Josseline, por su constante cuidado y por su tremendo apoyo en los días más oscuros que simplemente las palabras no son suficientes para poder describir. A la Lic. Adriana Cárcamo por su increíble apoyo a nivel personal que hizo constantemente superarme en todas las áreas de mi vida durante estos últimos 2 años, por los que faltan y por brindarme el empuje que necesitaba después del periodo de cuarentena. A todos esos amigos que he hecho durante este tiempo y a los que ya no están, por las risas, por la ayuda, por las conversaciones, por las anécdotas, por los abrazos, por los retos, por los días de trabajo, las noches de diversión y por todo lo que hizo que estos años hayan sido lo más amenos posible. Y, por último, a Sheicko que me ha acompañado durante las noches de desvelo del tercer ciclo, bachillerato, universidad y mi carrera profesional y que, como no, está aquí a mi lado mientras escribo esta dedicatoria. A todos ustedes: ¡mil gracias y un abrazo mi gente!

~ David Hurtado

DEDICATORIA

El camino de un universitario no es nada fácil; sin embargo, al llegar al punto de escribir mi dedicatoria en mi tesis, solo me queda agradecer. Comienzo por agradecerle a Dios por darme la fortaleza, la sabiduría, la resiliencia y demás cualidades que me ayudaron a llegar a donde estoy ahora. También me agradezco a mí mismo por siempre creer en mí, por nunca darme por vencido, por más difícil que fuese la situación universitaria a la que me enfrentara. En mi memoria quedan recuerdos increíblemente gratificantes, así como momentos muy oscuros, pero al final luché, luché y luché, y puedo decir que lo logré. Sin embargo, claramente jamás estuve solo; creo que, de haberlo estado, no estaría donde estoy. Jamás dejaré de estar agradecido con mis padres, que fueron mi principal apoyo en toda mi carrera universitaria. Agradezco a mi madre por levantarse a las 3:30 am todos los días y prepararme el desayuno para que yo comiera en la universidad. Esos pancitos con huevos hechos con amor eran un banquete para mí. Mi padre, quien jamás fue un obstáculo, alguien que, en lugar de decepcionarse al momento de contarle que saqué una mala nota, me decía: "¡Hay que buscar cuál fue el error y mejorar!" Esas palabras tan alentadoras levantaban los ánimos cuando estaba triste, así que por eso y más, gracias, padres.

Pero esto no acaba aquí. Agradezco a todo el personal docente de la UCA por quienes pasó mi formación como estudiante de ingeniería informática, por siempre demostrar entusiasmo por enseñar a sus alumnos, por jamás negarse a responder alguna duda y por todo el conocimiento transmitido a través de sus clases. Además, no sé qué hubiese sido de mí de no haber encontrado personas buenas en mi camino. Hice amigos increíbles a lo largo de mi carrera, personas que hasta el día de hoy siguen siendo parte de mi vida. Gracias por enseñarme, gracias por responder mis dudas, gracias por haber comido junto a mí en los almuerzos, gracias por las risas, gracias por todo, mis queridos amigos. También cabe mencionar que hubo personas que llegaron a mi vida y se fueron, pero cada uno dejó una enseñanza; unos de buena manera y otros tal vez de una no tan buena, pero al final me quedo con lo bueno y agradezco todo lo que hicieron por mí en su momento.

A mis amigos fuera de la universidad, gracias por motivarme. Gracias a mi mejor amigo Brayan por esas veces que yo me desvelaba estudiando y él me ayudaba a manejar desde San Vicente (mi ciudad natal) hasta mi universidad porque yo estaba cansado y no podía manejar así.

Estoy seguro de que podría llenar páginas y páginas agradeciendo. Las palabras sobran al momento de escribir esto, pero no me queda más que decir nuevamente, ¡Gracias a todos!

~ Fernando Carcamo

DEDICATORIA

Primero, quiero agradecer a Dios con enorme gratitud por darme la fuerza necesaria, la sabiduría y la inteligencia diaria para desarrollarme académicamente. Pero, sobre todo, le agradezco por cuidar mi salud, permitiéndome cumplir con esta etapa profesional.

A mi madre, Vilma Sonia Melara, quien siempre confió en mí y me brindó su apoyo incondicional durante toda la carrera. A mis hermanos, Salvador Melara, a quien admiro por enseñarme el valor de la disciplina y el trabajo; he aprendido mucho de él. A mi hermana, Margarita Melara, a quien considero una persona aplicada e inteligente; le agradezco por ser un apoyo en mi etapa de crecimiento. Y a mi hermano, Rodolfo José Melara, a quien agradezco por asesorarme y aconsejarme durante mi etapa como estudiante universitario.

También agradezco a mis sobrinos, quienes me llenan de alegría al ser un ejemplo y una referencia para ellos en esta etapa de mi vida. Mi familia ha mostrado su apoyo en los momentos difíciles y me motivaron a terminar mi carrera.

Es para mí una gran satisfacción dedicarles a ellos mi logro profesional, ya que, con mucho trabajo y esfuerzo, he podido concluir mis estudios.

Finalmente, agradezco a todos mis profesores, quienes compartieron sus conocimientos, me quedo satisfecho por la calidad de ellos a la hora de impartir sus clases, su comprensión al momento de solventar las dudas y preguntas, y a mis compañeros de la carrera. Nunca podré olvidar todas las alegrías, risas y tristezas que pasamos en cada ciclo. Gracias infinitas a todos.

~Rafael Melara

DEDICATORIA

Con inmensa gratitud y profundo cariño, dedico este proyecto de graduación a las personas que han sido pilares fundamentales en este viaje académico y personal. A mis padres Joaquín y Virna, por su amor incondicional y apoyo inquebrantable. Gracias por enseñarme la importancia del esfuerzo, la perseverancia y los valores que me han guiado hasta aquí. Sus sacrificios y constantes palabras de aliento han sido la base sobre la que he construido mis sueños. A mi hermana, mi compañera obligada de vida y confidente. Gracias por estar siempre a mi lado, por tus consejos y por compartir cada risa y lágrima. Tu presencia ha sido una fuente constante de fortaleza y motivación.

A mis docentes de la universidad, quienes, con su dedicación y pasión por la enseñanza, me han inspirado y desafiado a ser la mejor versión de mí mismo. Gracias por compartir su vasto conocimiento y por guiarnos con paciencia y sabiduría a lo largo de estos años. A mis compañeros de clase, en especial aquellos guerreros que han estado conmigo en trabajos y proyectos grupales, quienes me han acompañado en aventuras y desafíos. Juntos hemos enfrentado exámenes, trabajos y largas horas de estudio, creando lazos que perdurarán más allá de las aulas. A mis mejores amigos Daniel y Ademir, por ser el refugio en los momentos de estrés y la alegría en los momentos de celebración. Su apoyo constante y su confianza en mí han sido cruciales para llegar a esta meta. Aunque no pudimos finalizar esta tarea juntos, su amistad es invaluable y sé que lograrán grandes cosas.

A mi equipo de tesis, por su colaboración, dedicación y esfuerzo. Este trabajo es el resultado de nuestro compromiso y trabajo en equipo, y no podría haberlo logrado sin ustedes. Y a todos aquellos que, de una u otra manera, han contribuido a mi formación y crecimiento. Sus palabras, gestos y acciones han dejado una huella imborrable en mi vida.

Esta tesis es tanto mía como de ustedes. Gracias por ser parte de este viaje y por ayudarme a alcanzar esta meta.

Con todo mi cariño y gratitud,

~ Fernando Rosa

RESUMEN

En este trabajo se presenta un caso de aplicación de la técnica *transfer learning*, una de las muchas técnicas dentro de *machine learning*, utilizadas para entrenar modelos de inteligencia artificial. En el presente se aborda la situación de cómo obtener transcripciones de ponencias relacionadas a temas dentro del área STEM, donde por ponencia se refiere a clases, charlas o conferencias.

Primeramente, se da comienzo abordando investigaciones de apoyo que demuestran la versatilidad de la inteligencia artificial y también pasando por todos los avances que se han hecho en este campo.

En el siguiente capítulo, tanto en tecnologías donde se explica que es una red neuronal, algunos tipos que existen y para que pueden ser utilizadas y así como las áreas, técnicas desarrolladas y los resultados que se pueden conseguir con estas, así es como se aborda el área *del deep learning*, subrama del machine learning, la técnica de transfer learning y el concepto de "*fine tuned*".

En el capítulo 3 se desarrolla el detalle de cómo se eligió un modelo de reconocimiento de voz para generar transcripciones que ha sido recientemente desarrollado junto con las técnicas y tecnologías más novedosas, se explica el funcionamiento de este, y el proceso por el cual se ha entrenado con la técnica de transfer learning con unos datos de conferencias educativas en temas del área STEM que dan como resultado un modelo superior al base para esta tarea en este contexto específico, donde para comprobar su superioridad, se hace uso de las métricas popularmente más utilizadas.

Finalizando con el capítulo 4, donde orgullosamente, mostramos los resultados de este trabajo y lo que se infiere a partir de los resultados de las mediciones del rendimiento de ambos modelos

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

SIGLAS

ASR:	Automatic Speech Recognition
BERT:	Bidirectional Encoder Representations from Transformers
CAQDAS:	Computer Assisted Qualitative Data Analysis Software
CER:	Character Error Rate (Tasa de Error de Caracteres)
CMU Sphinx:	Carnegie Mellon University Sphinx
CNN:	Convolutional Neural Networks
CONIA:	Congreso de Ingeniería y Arquitectura
Conv1D:	Convolutional of 1 Dimension
ELU:	Exponential Linear Unit
FSF:	Free Software Foundation
GeLU:	Gaussian Error Lineal Unit
GPS:	Global Positioning System
GPT:	Generative pre-trained transformer
HMM:	Hidden Markov Models
IA:	Inteligencia Artificial
IDHUCA:	Instituto de derechos humanos UCA
ILSVRC:	ImageNet Large Scale Visual Recognition Challenge
KER:	Keyword Error Rate
LESHO:	Ley de Lengua de Señas de Honduras
MER:	Match Error Rate
ML:	Machine Learning
MLP:	Multilayer Perceptron
PLN:	Procesamiento del Lenguaje Natural
ReLU:	Rectified Linear Unit
RNN:	Recurrent neural network
STEM:	Science Technology Engineering Mathematics
SVM:	Support vector machine
T5:	Text-to-Text Transfer Transformer
UCA:	Universidad Centroamericana José Simeón Cañas
WER:	Word Error Rate
WIL:	Word Information Lost
WIP:	Word Information Preserved

CAPÍTULO 1. INTRODUCCIÓN

1.1. Problemática

Antes de los inicios de la computación, existían diversas actividades, tareas e incluso puestos laborales realizados por humanos los cuales eran procesos puramente mecánicos, a medida la computación avanzaba, las computadoras se volvieron herramientas imprescindibles para la realización de muchas tareas que redujeron las horas destinadas a varios procesos que pudieron ser automatizados. Sin embargo, muchos problemas quedaban fuera del alcance de lo que un ordenador podía, resolver puesto que el procesamiento de la información para completar ciertas tareas se escapaba de la potencia que los computadores tenían en ese momento además de la falta de los modelos matemáticos necesarios para poder resolver tareas de esa complejidad y también de la abstracción y formato en los cuales manejar los datos para que pudieran ser interpretados por una computadora.

Sin embargo, las matemáticas, la computación y la neurociencia siguió avanzando y se desarrollaron técnicas para procesar información de manera similar a como lo hacemos los humanos, con la llegada de la inteligencia artificial y los modelos de *machine learning*, que permitieron crear sistemas capaces de aprender a partir de unos datos de ejemplo en vez de desarrollar sistemas específicamente para la tarea dada, donde la complejidad es tal que no se conocía un algoritmo capaz de resolverla. Dichos problemas van desde la interpretación del audio, usada para traducir, transcribir y detección de elementos importantes, usado, el reconocimiento de imágenes en miles de dispositivos como medida de seguridad y autenticación biométrica y el procesamiento y detección de patrones en una enorme cantidad de datos.

Con esta rama nueva de la computación en marcha, vino con ella inconvenientes y problemas a resolver, como por ejemplo la recolección de datos que debían ser usados por estos sistemas para aprender, sin embargo, con el tiempo se desarrollaron técnicas que daban solución a estos. *Transfer learning* fue una de esas técnicas creadas, que permitió reutilizar varios de los sistemas desarrollados y entrenados para poder dar solución a casos más específicos de la tarea que ya cumplían, permitiendo así agilizar el entrenamiento de un sistema necesario para resolver una tarea que el original no podía completar.

Esto dio la versatilidad necesaria para que miles de organizaciones, compañías, proyectos y personas, vieran en el horizonte, la posibilidad de integrar y usar estas herramientas para sus actividades y automatizar y eficientizar aquellas tareas que en un principio parecían a las que no había escapatoria. Es así como el Instituto de Derechos Humanos de la Universidad Centroamericana (IDHUCA), en su misión humanitaria para la sociedad salvadoreña, en la que ofrece apoyo y asesoría legal a las víctimas de violaciones de

derechos humanos, empezó a barajar la posibilidad de utilizar dichas herramientas para dar solución algunos de sus problemas administrativos, como, por ejemplo, almacenar los relatos de los hechos de las personas a las que ayudan en formato de texto con rigurosidad, para luego ser consultado por los interesados.

Este proyecto, inspirado por el IDHUCA, da solución a otro problema: a lo largo de la carrera existen varias actividades en las que hay un ponente explicando un tema a un público como clases, charlas, conferencias, laboratorios y una larga lista, actividades en las cuales una transcripción resultaría útil para los estudiantes, en especial para aquellos con problemas de visión y audición y por supuesto para todos aquellos que por diversas razones no pueden asistir a dichas actividades, ya sea por temas laborales, personales o problemas como la distancia, el tiempo o las condiciones del camino que impiden movilizarse, también resultaría útil para ponentes, con la transcripción pueden llevar un historial útil para el análisis de su lenguaje para transmitir el conocimiento y también para la detección de ambigüedades o directos errores a la hora de explicar un tema, por estas razones, una transcripción resultaría útil en cualquier ponencia para todos los interesados en ella.

Específicamente en el caso de la Universidad Centroamericana José Simeón Cañas (UCA), cada año se celebra el Congreso de Ingeniería y Arquitectura (CONIA), un congreso que da lugar a diversas conferencias en el área de ingeniería y arquitectura, el ejemplo ideal para el problema planteado es así que, para dar solución a este problema en nuestra universidad, se entrenará un modelo de machine learning que permita transcribir la voz de los ponentes a texto, para que sirva del uso que tanto estudiantes como ponentes consideren apropiado.

Y también, se considera este proyecto como la demostración de que estas tecnologías están al alcance más que nunca, esperando que los que nos inspiraron lo tomen en consideración para sus futuros proyectos de mejora en sus procesos administrativos para que sigan apoyando a los salvadoreños y salvadoreñas que lo necesitan, y sigan inspirando más profesionales para que desde sus áreas construyan un futuro mejor.

1.2. Objetivos

1.2.1. Objetivo General

Entrenar un modelo de reconocimiento del habla mediante la técnica de transfer learning que permita obtener transcripciones de conferencias de temas relacionados al área ciencias, tecnología, ingeniería y matemáticas (por sus siglas STEM en inglés) que sea superior al modelo base.

1.2.2. Objetivos específicos

- Elegir un modelo de reconocimiento del habla como base para ser entrenado con datos nuevos.
- Explicar su funcionamiento y arquitectura para poder realizar un entrenamiento por transfer learning.
- Elegir un dataset para el entrenamiento del modelo que se apegue al contexto de la problemática a resolver.
- Medir la precisión del modelo entrenado para el idioma español utilizando las métricas más utilizadas para comprobar la efectividad de modelos de reconocimiento del habla.
- Comparar el modelo entrenado con el modelo base transcribiendo conferencias del CONIA para demostrar superioridad.

1.3. Alcance del software

El resultado de esta investigación es un modelo de reconocimiento del habla entrenado con datos de conferencias educativas sobre áreas STEM en español, dicho modelo no es completamente nuevo, sino que está entrenado sobre un modelo base entrenado con datos generales para poder reconocer y transcribir discursos y oraciones en el idioma español el cual ha sido reforzado con la técnica de transfer learning, además de los archivos auxiliares que sirvieron para el proceso de entrenamiento, como los datos utilizados para entrenar, y los de comprobación del modelo, como los audios utilizados para medir su efectividad.
Esto constituye un repositorio (ver anexo A) donde podrá encontrar lo siguiente:

- El programa utilizado para entrenar el modelo.
- Los programas utilizados para obtener las transcripciones del modelo tanto base como el entrenado.
- El dataset utilizado para entrenar al modelo, ver anexo B y la sección 3.2 para más detalles
- Los datos de prueba de archivos audio obtenidos de conferencias del CONIA 2023 que sirvieron para medir la efectividad del modelo entrenado.
- Las transcripciones generadas por el modelo base, el modelo entrenado y manualmente que servirán como referencia para comprobar la efectividad.
- El programa utilizado para normalizar los datos de las transcripciones generadas, para conocer esta normalización vea la sección 3.6.2

1.4. Antecedentes

1.4.1. Investigaciones Internacionales

Propuesta de traductor de voz a texto en tiempo real a lengua de señas basado en inteligencia artificial para los idiomas oficiales de Sudáfrica para la era COVID-19

Madahana, Khoza-Shangase, Moroe, Mayombo, Nyandoro y Ekoru (2022) realizaron una investigación innovadora en la Universidad de Witwatersrand, Johannesburgo, Sudáfrica, centrada en el desarrollo de una solución de traducción en tiempo real que utiliza inteligencia artificial (IA) para convertir el habla en texto y lenguaje de señas. Esta investigación es crucial dado que la Organización Mundial de la Salud proyecta que para 2050, aproximadamente 2.5 mil millones de personas a nivel mundial experimentarán algún grado de pérdida auditiva.

Este estudio tiene como objetivo realizar una revisión exploratoria sobre la aplicación de inteligencia artificial (IA) para la traducción de habla a texto en tiempo real a lenguaje de señas y, en consecuencia, proponer una solución de traducción en tiempo real basada en IA para los idiomas sudafricanos de habla a texto a lenguaje de señas.

La búsqueda de publicaciones revisadas por pares sobre la traducción en tiempo real de texto a lenguaje de señas basada en inteligencia artificial para personas con discapacidad auditiva entre 2019 y 2021 se realizó en bases de datos bibliográficas electrónicas como ScienceDirect, PubMed, Scopus, MEDLINE y ProQuest. Esta revisión se llevó a cabo como paso previo al desarrollo de un traductor sudafricano en tiempo real.

El estudio demostró una brecha significativa en la utilización clínica y la investigación de tecnologías avanzadas en el continente africano, particularmente en el contexto de la pandemia de COVID-19, que ha exacerbado los desafíos para las personas con discapacidad auditiva. La investigación llevada a cabo por Madahana (2022) no sólo arroja luz sobre estas cuestiones críticas, sino que también pavimenta el camino para futuras innovaciones que podrían transformar la forma en que las tecnologías de asistencia mejoran la vida de las personas en todo el mundo.

Algoritmo de voz a texto para personas como herramienta educativa personas con sordera

David Amador (2020) Explica un algoritmo de voz a texto como herramienta educativa para personas con sordera, trabajo de investigación realizado en la Universidad Tecnológica Centroamericana de Honduras en la buscaba apoyar a la población con problemas de sordera que carecen de apoyo en área de educación,

cabe mencionar que la población de Honduras con problemas auditivos tienen su propio lenguaje de señas conocido como LESHO (Lenguaje de señas de Honduras) estas personas no contaban con el acceso a escuelas especiales a lo que se optó a crear un algoritmo para apoyarlas en el sistema educativo.

La investigación tiene como objetivo crear un algoritmo traductor de voz a texto en la estructura gramatical utilizada por los sordos, el cual sirva como herramienta de apoyo en el área Educativa; con la utilización de un micrófono y una computadora.

Desarrollo de una herramienta para la conversión de voz de texto

En el proyecto llevado a cabo por Moya (2018) en la Universidad Politécnica de Madrid se realizó una aplicación funcional, capaz de convertir un audio de conversación en texto mediante su transcripción, es decir, un proceso speech-to-text.

En el proyecto se buscó crear un audio de conversación en texto accediendo a distintos repositorios de audios en español en inglés, se emplearon para entrenar un conjunto de algoritmos que permite la transcripción de voz a texto y se explora el uso de redes neuronales. Entrenado el algoritmo y aplicando *Machine Learning* con el uso de la plataforma *TensorFlow* y con Python como lenguaje de programación, empleando Colab, un servicio de cloud que permite el uso gratuito de los recursos de Google debido a la gran capacidad de cómputo que se necesita para entrenar un modelo

Transfer learning en la detección de cáncer en la piel

La investigación realizada en la universidad internacional de La Rioja por Martínez (2021) llevo un análisis de la utilidad y la fiabilidad del Transfer Learning en la detección de distintos tipos de cáncer en la piel, se desarrolló un modelo que se alimenta de varios modelos pre-entrenados de un dataset de imágenes de lunares y manchas que ha sido previamente clasificado, tratando de comprobar si verdaderamente esta técnica se puede plantear como una herramienta sanitaria que pueda ayudar a los médicos.

Para desarrollar el modelo se utilizó el lenguaje programación Python, donde también se utilizaron diferentes librerías para el manejo de datos, previo al desarrollarse modelo se buscaron imágenes de personas sin violentar sus datos personales, y realizar un análisis a partir de lunares o manchas en la piel.

Testeando ATLAS.ti con OpenAI: hacia un nuevo paradigma para el análisis cualitativo de entrevistas con Inteligencia artificial

El avance de las tecnologías de la información y la comunicación ha propiciado la aparición de programas conocidos como software de análisis de datos de computación asistida (por sus siglas en inglés: CAQDAS). Estos programas, como ATLAS.ti (Paulus y Lester, 2015; Paulus, 2018), NVivo (Niedbalski y Ślęzak, 2017) o MAXQDA (Schultheiß y Lewandowski, 2020; 2021), entre otros, facilitan la organización, sistematización, procesamiento y análisis de los datos utilizados en investigaciones. Este tipo de herramientas asisten a los investigadores en la mejora del rigor y la eficacia de su estudio, como por ejemplo en el análisis de entrevistas, donde se compara el proceso manual y haciendo uso de inteligencia artificial en la figura 1.1.

Un programa que facilita el almacenamiento, gestión, consulta y análisis de datos no estructurados. Las funcionalidades de ATLAS.ti incluyen la capacidad de codificar documentos de texto (.doc, .txt, .pdf, entre otros), archivos audiovisuales (.wav, .mp3, .avi, .mp4, entre otros), fotografías e incluso datos de Twitter u otras aplicaciones de terceros.

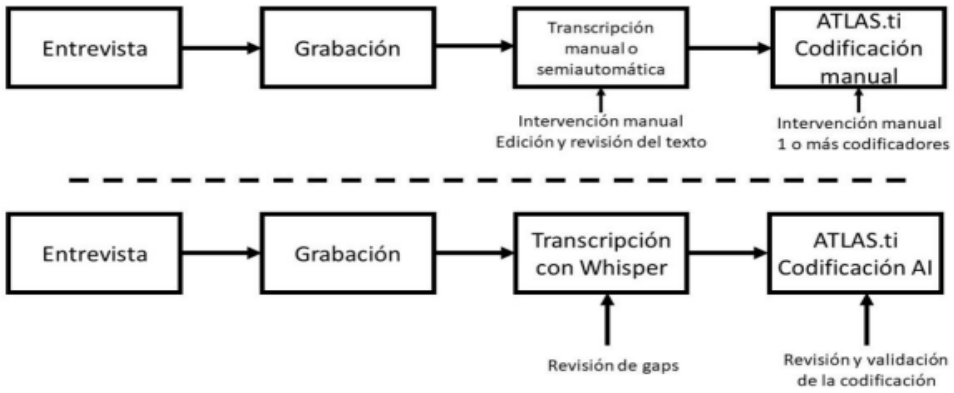


Figura 1.1: Comparación de flujos de trabajo de análisis de entrevistas de forma manual o con herramientas soportadas en AI.

Clasificación del cáncer de piel a nivel dermatólogo con redes neuronales profundas

Un ejemplo destacado del uso de deep learning es el estudio realizado por Esteva (2017), donde se desarrolló un modelo de deep learning para la clasificación de lesiones cutáneas utilizando imágenes dermatológicas. Este modelo, basado en redes neuronales convolucionales (por sus siglas en inglés: CNN), fue entrenado con más de 129,000 imágenes de piel, logrando una precisión comparable a la de dermatólogos certificados en la identificación de cáncer de piel. La investigación demuestra cómo el deep learning puede ser utilizado para mejorar el diagnóstico médico y potencialmente salvar vidas mediante la detección temprana de enfermedades.

Un análisis empírico de la tasa de error de palabra y tasa de error de palabra clave

(Park, 2008) hace un análisis de dos métricas utilizadas para medir el rendimiento de un sistema de reconocimiento de voz automática (por sus siglas en inglés: ASR) que son la tasa de error de palabra (por sus siglas en inglés: WER) y la tasa de error de palabra clave (por sus siglas en inglés: KER) y determinar cuál era la más adecuada para calibrar un sistema ASR, para ello utilizaron 100 llamadas de un *call center*, determinando que generalmente el WER es más que suficiente para el análisis de efectividad, especialmente para los casos que se mantienen por debajo del 25%.

1.4.2. Investigaciones nacionales

Compilación de investigaciones de tecnología 2017 sobre extracción de conocimiento a partir de texto

La Universidad tecnológica de El Salvador junto con Flores (2018) realizó una investigación sobre la extracción de conocimiento a partir de texto con el fin de aplicar *data mining*, a partir de un conjunto grande de textos, obtener información que permita obtener datos útiles para generar conceptos, ontologías y definiciones

Las pruebas realizadas fueron preprocesamiento de texto aplicación de los modelos sobre cada una de las partes de artículos, uso de nubes de palabras y otros métodos gráficos para determinar si el modelo es capaz de extraer información útil y de calidad para los textos extraídos de la red social Twitter, se hizo una limpieza de contenido los mejores resultados se obtuvieron con el modelo Word2Vec, se pudo extraer información útil de propuestas y noticias.

CAPÍTULO 2. MARCO TEÓRICO

Desde el nacimiento de la concepción de la idea del aprendizaje profundo a través de las redes neuronales artificiales surgidas por el alto desempeño y evolución de la inteligencia artificial (IA), para dar paso luego a temas más complejos como el machine learning, el transfer learning y el fine tuning, comenzamos definiendo que es la inteligencia artificial según la interpretación de uno de los precursores del computador moderno IBM nos dice que la IA es la tecnología que permite que las computadoras simulen la inteligencia humana y las capacidades humanas de resolución de problemas.

2.1. Inteligencia artificial

Los proyectos mencionados anteriormente en los antecedentes, a pesar de dar solución a problemas muy distintos entre sí, comparten algo en común: el acercamiento utilizado utiliza técnicas, metodologías, tecnologías y arquitecturas que fueron desarrolladas gracias a los avances de una rama de la computación: la Inteligencia Artificial o IA.

Por sí sola o combinada con otras tecnologías como, sensores, geolocalización, robótica, entre otros, la IA puede realizar tareas que de otro modo requerirían inteligencia o intervención humana. Los asistentes digitales, la guía por Sistema de Posicionamiento Global (Global Positioning System, GPS), los vehículos autónomos y las herramientas de inteligencia artificial generativa como ChatGPT (GPT significando Transformer Generativo Pre Entrenado o Generative Pre Trained Transformer) de Open AI son solo algunos ejemplos de inteligencia artificial en las noticias diarias y en nuestra vida cotidiana (IBM), pero también cabe destacar casos tan específicos y diversos como los mencionados en los antecedentes en las investigaciones tituladas" Compilación de investigaciones de tecnología 2017 extracción de conocimiento a partir de texto" y " Propuesta de traductor de voz a texto en tiempo real a lengua de señas basado en inteligencia artificial para los idiomas oficiales de Sudáfrica para la era COVID-19".

Desde la década de 1980, la aparición de los sistemas, que utilizan reglas predefinidas para imitar la toma de decisiones humanas, revitalizó el interés en la IA. Sin embargo, fue el avance en el aprendizaje automático y el poder de procesamiento de las computadoras en la década de 2010 lo que verdaderamente transformó el campo. Tecnologías como el deep learning, impulsadas por redes neuronales artificiales, permitieron a los sistemas de IA superar a los humanos en tareas complejas como el reconocimiento de imágenes y el análisis de datos masivos (Goodfellow, Bengio, & Courville, 2016).

2.2. Machine learning

El aprendizaje automático (Machine Learning, ML) es una rama de la inteligencia artificial, que se centra en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender y hacer predicciones o decisiones basadas en datos. A diferencia de los sistemas tradicionales programados explícitamente, los sistemas de ML mejoran su rendimiento en tareas específicas a medida que se exponen a más datos (Mitchell, 1997).

El concepto de aprendizaje automático tiene sus raíces en los primeros días de la informática y la estadística. En la década de 1950, Arthur Samuel, un pionero en el campo, desarrolló uno de los primeros programas de juego de damas que podía aprender de sus propias jugadas. Este evento marcó uno de los primeros ejemplos prácticos de una máquina "aprendiendo" de la experiencia (Samuel, 1959).

En las décadas siguientes, el campo del ML evolucionó significativamente con el desarrollo de algoritmos más sofisticados y el aumento de la capacidad de procesamiento de las computadoras. En la década de 1990, el enfoque estadístico se volvió dominante, y técnicas como las máquinas de soporte vectorial (por sus siglas en inglés: SVM) y los modelos de Markov ocultos (por sus siglas en inglés: HMM) se hicieron populares. La última década ha visto una explosión en el uso del Deep Learning, impulsado por redes neuronales artificiales más profundas y poderosas, lo que ha llevado a avances impresionantes en áreas como el reconocimiento de voz, la visión por computadora y la traducción automática (Goodfellow, Bengio, & Courville, 2016).

El campo del ML abarca una amplia gama de algoritmos, cada uno adecuado para diferentes tipos de problemas. LeCun, Bengio, & Hinton (2015) mencionan estos como algunos de los algoritmos más influyentes incluyen:

Regresión Lineal y Logística: Utilizados principalmente para problemas de predicción y clasificación binaria, respectivamente. Estos modelos son fáciles de interpretar y eficientes para datos linealmente separables.

Árboles de Decisión y Random Forests: Son utilizados tanto para problemas de clasificación como de regresión. Los árboles de decisión dividen repetidamente los datos en subconjuntos homogéneos basados en la variable que proporciona la mejor separación.

Máquinas de Soporte Vectorial (SVM): Estas son eficaces para problemas de clasificación y regresión en espacios de alta dimensionalidad. SVM encuentra el hiperplano que mejor separa las clases de datos.

Redes Neuronales: Modelos inspirados en el cerebro humano, capaces de aprender representaciones complejas de los datos. Las redes neuronales profundas, en particular, han revolucionado campos como la visión por computadora y el procesamiento del lenguaje natural mediante el uso de múltiples capas de neuronas interconectadas.

Una investigación de Hannun (2014) exploró el uso de redes neuronales profundas para el reconocimiento de voz. El estudio utilizó una arquitectura de red neuronal recurrente (por sus siglas en inglés: RNN) entrenada en un gran conjunto de datos de voz, logrando resultados que superaron significativamente los métodos tradicionales de la época.

2.3. Deep Learning

Deep Learning es una rama del machine learning que se centra en la creación y arquitectura de redes neuronales artificiales con múltiples capas para modelar y comprender patrones y estructuras en grandes cantidades de datos para entrenar modelos que realicen tareas como las que hacemos los humanos. Este enfoque ha ganado prominencia en la última década debido a su capacidad para mejorar significativamente el rendimiento en tareas complejas como el reconocimiento de voz, la visión por computadora y el procesamiento del lenguaje natural y con diversas aplicaciones en las áreas como la medicina en el caso "Clasificación del cáncer de piel a nivel dermatólogo con redes neuronales profundas".

El concepto de redes neuronales se remonta a la década de 1940, cuando Warren McCulloch y Walter Pitts propusieron el primer modelo matemático de neuronas artificiales. Sin embargo, no fue hasta los años 80 y 90 que los avances en algoritmos como el *backpropagation* permitieron entrenar redes neuronales más profundas. La verdadera revolución del deep learning comenzó en la década de 2010, impulsada por el aumento de la capacidad de cómputo y la disponibilidad de grandes volúmenes de datos etiquetados (Goodfellow, Bengio, & Courville, 2016).

Las redes neuronales profundas están compuestas por varias capas de neuronas, donde cada capa transforma la entrada de manera no lineal para aprender características jerárquicas de los datos, dichas arquitecturas son mencionadas en la sección 2.6.

A pesar de sus éxitos, el deep learning enfrenta varios desafíos. Entre ellos se encuentran la necesidad de grandes cantidades de datos etiquetados, el alto costo computacional y la interpretabilidad de los modelos. Investigaciones actuales se enfocan en mejorar la eficiencia de los algoritmos, desarrollar técnicas de aprendizaje no supervisado y semisupervisado, y crear modelos más interpretables (Zhang, Bengio, Hardt, Recht, & Vinyals, 2021).

2.4. Whisper

Para dar solución a la problemática, haremos uso de la técnica de transfer learning, explicada en la sección 2.7, que hace uso de un modelo base ya entrenado para poder entrenarlo con un conjunto de datos nuevos para reforzar su conocimiento antes adquirido, mejorando su rendimiento para entradas que compartan un contexto similar al de los datos del segundo entrenamiento, es por ello que es importante conocer su funcionamiento. En este caso, se hará uso de Whisper como modelo base, para conocer los detalles del proceso por el cual se eligió el mencionado revisar la sección 3.1.

Whisper es un modelo de transcripción de voz a texto desarrollado por OpenAI, específicamente una red neuronal transformer. Utilizando un enfoque de deep learning, Whisper está diseñado para manejar múltiples idiomas y variaciones en el habla, proporcionando una solución robusta y precisa para la transcripción automática. La accesibilidad del código abierto de Whisper permite a los desarrolladores adaptar y mejorar el modelo según sus necesidades específicas, facilitando aplicaciones innovadoras en reconocimiento de voz (Radford, 2023).

Whisper ha sido utilizado en diversas aplicaciones de transcripción de voz a texto. Un estudio reciente hecho por Lopeza (2023) mostró cómo Whisper puede ser aplicado para transcribir entrevistas multilingües con alta precisión. La capacidad de Whisper para manejar múltiples acentos y dialectos lo convierte en una herramienta valiosa en investigaciones lingüísticas y en la creación de subtítulos automáticos para contenido multimedia (Radford, 2023).

Whisper es un modelo *open source* que fue entrenado con una técnica que ha sido infravalorada según (Radford, 2023) para los sistemas de reconocimiento de voz, la cual es conocida como supervisión débil, que demostró la capacidad de obtener resultados comparables a los de otros modelos como Wav2Vec desarrollado por Meta y tener una mayor robustez, demostrando que modelos utilizados con esta técnica

son capaces de desenvolverse bien en contextos nuevos para el modelo, eliminando la necesidad de *fine-tuning* para conseguir resultados de alta calidad

2.4.1. Arquitectura

Whisper utiliza una arquitectura que está desacoplada del modelo para poder ser implementada con facilidad en entornos nuevos, esto logrado con una red neuronal *transformer*, es un tipo de red neuronal que solventa los problemas de las redes neuronales recurrentes y convolucionales según (Vaswani, 2017) siendo entrenadas más rápido gracias a que se basan únicamente en mecanismos de atención y que procesan varios datos en paralelo y no de forma secuencial, estos son procesos en los cuales se les asigna una importancia mayor a ciertos datos que a otros de la información que se está procesando, así como lo hacen nuestros sistemas biológicos, que se enfoca en la información relevante, por ejemplo, ciertas palabras en una frase serán más importantes para predecir la siguiente mientras se pueden ignorar las demás (Chaudhari, Polatkan, Ramanath, & Mithal, 2019).

Debido al procesamiento en paralelo, es necesaria mantener información sobre la posición relativa o absoluta de las palabras, por ello se hace uso de un proceso de codificación posicional, en el caso de Whisper este proceso está basado en funciones sinusoidales.

Como se explica en la figura 2.1, Whisper hace uso de capa convolucional de 1 dimensión (abreviadas Conv1D) debido a que se está procesando audio, como explica Di (2018) es una capa donde se le aplica una operación de convolución a los datos que son divididos en bloques, y una función de activación GELU (Gaussian Error Linear Unit) que supera a RELU (Hendrycks & Gimpel, 2016)

Estos datos son en realidad el audio representado en un espectrograma mel**,** el cual es una representación de la frecuencia del audio en el tiempo, pero que no está medida en hertz como un espectrograma común, sino que está medido en mels, esto es usado en procesamiento de audio debido a que el oído humano es más sensible a cambios en bajas frecuencias que en las altas frecuencias, esta sensibilidad decrece logarítmicamente (Introduction To Audio Data - Hugging Face Audio Course, s. f.), el espectrograma mel se deshace de ello haciendo que todas las diferentes frecuencias sean iguales.

Luego los datos empiezan a viajar por la red, pasando primero por los bloques codificadores que les pasan los datos generados a los bloques decodificadores para poder generar una salida (Cho, K., Courville, A., & Bengio, Y.,2015).

Los bloques codificadores están compuestos por una red perceptrón multicapa (por sus siglas en inglés: MLP) que es una de las redes más sencillas que existe, compuestas por una capa de entrada, una de salida y una o más capas ocultas, con la característica de que los nodos de las capas adyacentes están totalmente conectadas, y compuestas por un mecanismo de *self-attention* que significa que se trabaja con una única secuencia de una entrada. Los datos generados luego son enviados a los decodificadores compuestos por los mismos elementos, pero con el agregado de que cuentan también con un mecanismo de *cross-attention*, lo que significa que trabajan con los datos generados de secuencias de otras entradas, esto es importante para contextualizar y poder generar los tokens y predecir los siguientes.

Este proceso es repetido hasta terminar con el audio, descrito en la siguiente figura 2.1.

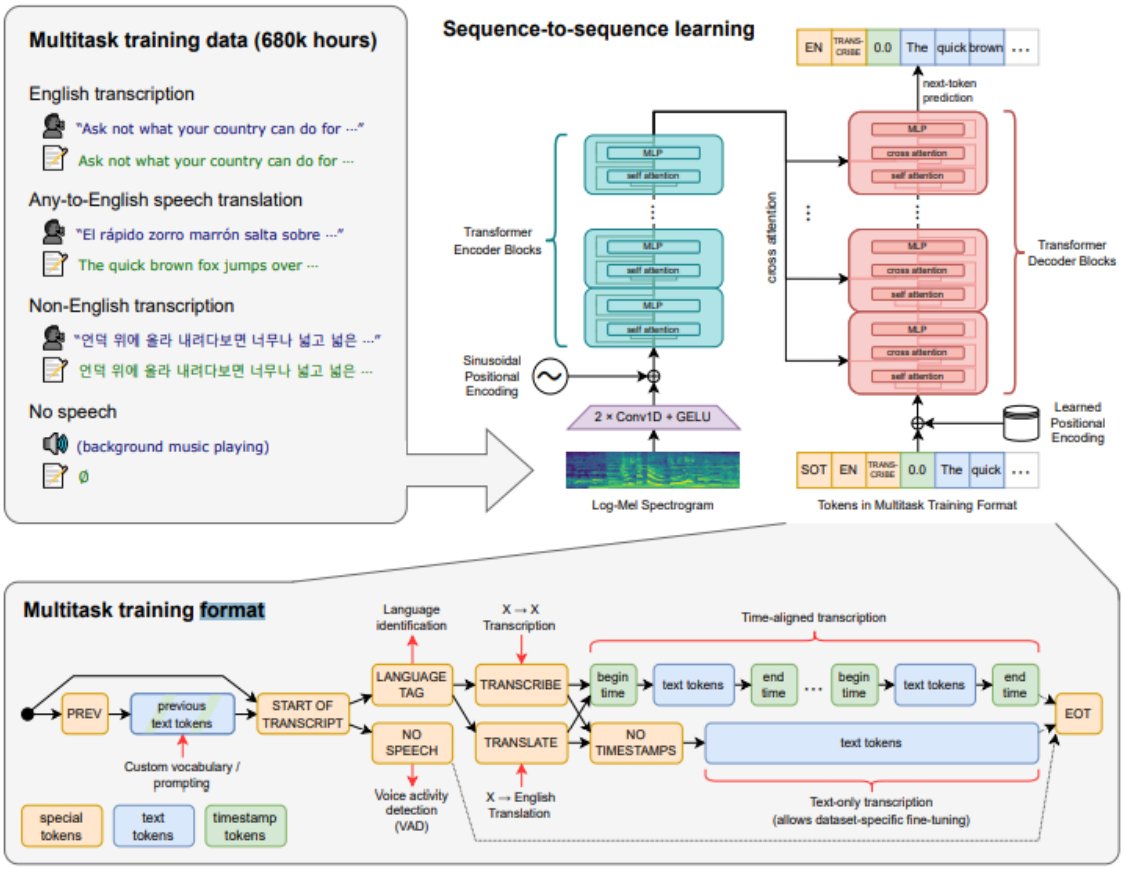


Figura 2.1: Arquitectura y proceso general de Whisper, OpenAI (2022)

2.4.2 Entrenamiento de la red neuronal

OpenAI (2022) explica, una vez la arquitectura está completa, la red puede empezar a procesar datos y adquirir conocimiento, OpenAI siguió el siguiente proceso para entrenar la red:

- Eligieron varios datasets y procedieron a formatearlos y limpiarlos de acuerdo con sus criterios.
- Ingresaron los datos limpios a la red.
- Experimentaron con el modelo.
- Obtuvieron las métricas de su rendimiento.

Usualmente, la preparación del dataset para entrenar el modelo es un proceso en el que se dedica una cantidad de tiempo y esfuerzo considerable, puesto que es crucial para obtener un modelo que cumpla la tarea deseada con precisión, no obstante, Whisper utilizó un acercamiento simple, utilizó las transcripciones de audios que se encontraban en internet, esto hizo que el dataset fuera diverso en calidad de audio y transcripciones.

La diversidad en la calidad de los audios es útil para obtener modelos más robustos, sin embargo, las transcripciones de baja calidad no son deseadas, por ello, la mayor parte del trabajo en este proceso fue eliminar dichas transcripciones. Algunas de estas transcripciones fueron generadas por otros modelos de sistemas de reconocimiento del habla (por sus siglas en inglés: ASR), combinar transcripciones hechas por sistemas y hechas por humanos es perjudicial según (Ghorbani, 2021) por lo que fue necesario eliminarlas, esto se logró a base de inspeccionar las transcripciones con algunos filtros y reglas que permiten detectar que una transcripción fue generada gracias a las limitaciones de los sistemas, como por ejemplo la falta de signos de puntuación, el formato del texto y aspectos estéticos como el uso de mayúsculas. También, se utilizó un sistema de reconocimiento de idioma para asegurar que las transcripciones estaban escritas en el idioma que supuestamente correspondía al audio, aquellas que no concordaban fueron eliminadas (excepto aquellas que transcribían de un idioma cualquiera al inglés, esto fue así debido a que Whisper también es capaz de traducir de varios idiomas al inglés). Por último, se dividieron los audios en segmentos de 30 segundos y se entrenó un modelo preliminar y se calculó su precisión, los resultados fueron inspeccionados manualmente y esto reveló muchas transcripciones incompletas y otras generadas que la primera revisión lo logró detectar.

Con el dataset limpio, se entrenó una familia de modelos de 5 integrantes, uno cada vez más robusto que el anterior, esto se hizo con el algoritmo AdamW, una de las utilidades de PyTorch, altamente utilizadas en redes que cumplen con las características de Whisper.

2.4.3. Resultados de Whisper

El resultado es una familia de modelos entrenada con una técnica infravalorada hasta ahora, que se basa únicamente en utilizar un dataset diverso con datos de calidad permite obtener un sistema capaz de resolver la tarea deseada con alta precisión, sin el uso de las técnicas más utilizadas en estos problemas y obtener modelos robustos.

2.5 Métricas de efectividad

Para comprobar la efectividad de un sistema ASR se debe medir la precisión con la cual reconoce las palabras, para ello existen varias métricas que se basan en medir la cantidad de errores que comete al procesar el audio, los errores que puede cometer un sistema ASR se clasifican en 3:

- Error por sustitución, es decir, aquellas palabras o caracteres que fueron interpretadas por una distinta en la transcripción referencia
- Error por inserción, es decir, aquellas palabras o caracteres que fueron agregadas y no existen en la transcripción referencia
- Error por eliminación, es decir, aquellas palabras o caracteres que fueron eliminadas y no fueron recogidas por la transcripción referencia

Tomando como base las métricas analizadas en el estudio mencionado en los antecedentes "Un análisis empírico de la tasa de error de palabra y tasa de error de palabra clave", se determinó que la tasa de error de palabras (por sus siglas en inglés: WER) sería la métrica principal para medir la efectividad del modelo entrenado, no obstante, hay más métricas que utilizaremos como apoyo extra para medir y comparar el rendimiento del modelo.

Las métricas utilizadas son:

- **Word Error Rate (WER):** La tasa de error de palabra mide el porcentaje de palabras que fueron incorrectamente predichas, donde 0 es un puntaje perfecto, como se muestra en la ecuación 2.1.

$$WER = \frac{S + I + E}{N} \quad (Ec. 2.1)$$

- **Character Error Rate (CER):** La tasa de error de carácter mide el porcentaje de caracteres que fueron incorrectamente predichas, donde 0 es un puntaje perfecto, como se muestra en la ecuación 2.2.

$$CER = \frac{S + I + E}{N} \ (Ec. 2.2)$$

- **Match Error Rate (MER):** La tasa de error de coincidencia mide el porcentaje de palabras que fueron incorrectamente predichas e insertadas, donde 0 es un puntaje perfecto, como se muestra en la ecuación 2.3.

$$MER = \frac{S + I + E}{N + I} \ (Ec. 2.3)$$

- **Word Information Lost (WIL):** La información perdida de palabra mide el porcentaje de palabras que fueron incorrectamente predichas entre unas oraciones de referencia y otras oraciones generadas, donde 0 es un puntaje perfecto, como se muestra en la ecuación 2.4.

$$WIL = 1 - \frac{C}{N} + \frac{C}{P} \ (Ec. 2.4)$$

- **Word Information Preserved (WIP):** La información preservada de palabra mide el porcentaje de palabras que fueron correctamente predichas entre unas oraciones de referencia y otras oraciones generadas, donde 1 es un puntaje perfecto, como se muestra en la ecuación 2.5.

$$WIP = \frac{C}{N} * \frac{C}{P} \ (Ec. 2.5)$$

Donde:

- S es la cantidad de sustituciones
- I es la cantidad de inserciones
- E es la cantidad de eliminaciones
- C es la cantidad de palabras correctas
- N es la cantidad de palabras en la transcripción de referencia
- P es la cantidad de palabras en la transcripción generada

Notar que, para calcular este valor, es necesario contar con la transcripción referencia y con la transcripción generada por el sistema. La transcripción referencia se suele recomendar que sea obtenida manualmente por un humano, aunque algunos datasets cuentan con transcripciones obtenidas por sistemas.

2.6 Redes Neuronales

Como se mencionó anteriormente, *Whisper* es una red neuronal *transformer*, que estas son uno de los avances más importantes dentro del campo de *Machine Learning* (ML), resultado del mejoramiento de varias arquitecturas que surgieron a medida que avanzaban, pasando primero por convolucionales a recurrentes y que junto con el uso y avances de otras tecnologías y mecanismos, como lo son los mecanismos de atención y las mejoras que se hicieron en las funciones de activación, desembocaron en la creación de las redes neuronales transformer.

Según el artículo de IBM, las redes neuronales, también conocidas como redes neuronales artificiales (por sus siglas en inglés: ANN) son un subconjunto de machine learning y constituyen el eje de los algoritmos de deep learning. Su nombre y estructura se inspiran en el cerebro humano, e imitan la forma en la que las neuronas biológicas se señalan entre sí.

Las redes neuronales están compuestas por capas de nodos, incluyendo una capa de entrada, una o más capas intermedias y una capa de salida. Cada nodo, o neurona artificial, se conecta a otro y posee un peso y un umbral asignados. Si la salida de un nodo excede el umbral establecido, este nodo se activa y transmite información a la siguiente capa de la red. De lo contrario, no se transfiere información a la siguiente capa.

Las redes neuronales se basan en el entrenamiento con datos para aprender y mejorar su precisión con el tiempo. Una vez que estos algoritmos de aprendizaje se afinan con precisión, se convierten en herramientas informáticas y de inteligencia artificial muy poderosas, permitiéndonos clasificar y agrupar datos rápidamente. Tareas como el reconocimiento de voz o de imágenes pueden completarse en minutos, en contraste con las horas que tomaría a expertos humanos hacerlo manualmente. Un ejemplo destacado de red neuronal es el algoritmo de búsqueda de Google o el caso mencionado en los antecedentes en el estudio titulado “Clasificación del cáncer de piel a nivel dermatólogo con redes neuronales profundas”.

Existen distintos tipos de arquitecturas de redes neuronales, entre las cuales tenemos:

Redes Neuronales Convolucionales (CNN): Especializadas en procesamiento de datos con estructura de grilla, como imágenes. Utilizan capas de convolución para detectar características locales como bordes y texturas (LeCun, Bengio, & Hinton, 2015).

Redes Neuronales Recurrentes (RNN): Adecuadas para secuencias de datos, como series temporales o texto, ya que pueden mantener información a lo largo de las secuencias mediante sus estados internos (Hochreiter & Schmidhuber, 1997).

¿Cómo funcionan las redes neuronales?

Piense en cada nodo individual como su propio modelo de regresión lineal, formado por datos de entrada, ponderaciones, un sesgo (o umbral) y una salida.

Una vez que se determina una capa de entrada, se asignan ponderaciones. Estas ponderaciones permiten determinar la importancia de cualquier variable, donde las más grandes contribuyen más significativamente a la salida en comparación con otras entradas. A continuación, todas las entradas se multiplican por sus respectivas ponderaciones y se suman. A continuación, la salida se pasa a través de una función de activación, que determina la salida. Si la salida supera un determinado umbral, activa el nodo y pasa los datos a la siguiente capa de la red. Como resultado, la salida de un nodo se convierte en la entrada del nodo siguiente. Este proceso de pasar datos de una capa a la siguiente define esta red neuronal como una red de propagación hacia delante.

2.6.1 Redes neuronales convolucionales

Las Redes Neuronales Convolucionales o CNN son un tipo especializado de redes neuronales diseñadas para procesar y analizar datos con una estructura de cuadrícula, como las imágenes, fueron de los grandes avances que abrieron el camino a otras arquitecturas del Deep learning. Las CNN son especialmente eficaces en tareas de reconocimiento y clasificación de imágenes, debido a su capacidad para capturar y aprender características espaciales y patrones jerárquicos presentes en los datos visuales (LeCun, 2015).

Un estudio notable realizado por Krizhevsky (2012) presentó AlexNet, una CNN que ganó la competencia del reto de reconocimiento visual a larga escala ImageNet (por sus siglas en inglés: ILSVRC) en 2012. AlexNet superó significativamente los modelos anteriores en la tarea de clasificación de imágenes, demostrando la eficacia de las redes neuronales profundas con múltiples capas convolucionales. Este trabajo marcó un hito en el campo de la visión por computadora y fomentó un creciente interés en el desarrollo de arquitecturas de CNN más profundas y complejas.

2.6.2 Redes neuronales recurrentes

Las Redes Neuronales Recurrentes (RNN) son una clase de redes neuronales diseñadas para manejar datos secuenciales, presentando una mejora a sus predecesoras las CNN. A diferencia de las redes neuronales

tradicionales, las RNN tienen conexiones recurrentes que permiten mantener un estado interno y recordar información a lo largo de una secuencia de datos. Esto las hace ideales para tareas que requieren el manejo del contexto y el orden de los datos, como el procesamiento del lenguaje natural, la traducción automática y el análisis de series temporales (Goodfellow, Bengio, & Courville, 2016).

Las RNN y sus variantes se utilizan en diversas aplicaciones donde los datos secuenciales son fundamentales:

Procesamiento del Lenguaje Natural (PLN): En tareas como el análisis de sentimientos, la generación de texto y la traducción automática, las RNN pueden capturar el contexto y las dependencias entre palabras. Por ejemplo, el modelo seq2seq (sequence-to-sequence) utiliza dos RNN (un codificador y un decodificador) para traducir frases de un idioma a otro (Sutskever, Vinyals, & Le, 2014).

Reconocimiento de voz: Las RNN son efectivas para convertir secuencias de audio en texto. Los modelos como Deep Speech de Baidu han utilizado RNN para mejorar significativamente la precisión del reconocimiento de voz (Hannun, 2014).

2.6.3. Redes neuronales transformer

Una red neuronal de tipo transformer es un tipo de red que solventa los problemas de las redes neuronales recurrentes y convolucionales según (Vaswani, 2017) siendo entrenadas más rápido gracias a que se basan únicamente en mecanismos de atención y que procesan varios datos en paralelo y no de forma secuencial, estos son procesos en los cuales se les asigna una importancia mayor a ciertos datos que a otros de la información que se está procesando, así como lo hacen nuestros sistemas biológicos, que se enfoca en la información relevante, por ejemplo, ciertas palabras en una frase serán más importantes para predecir la siguiente mientras se pueden ignorar las demás (Chaudhari, Polatkan, Ramanath, & Mithal, 2019).

Debido al procesamiento en paralelo, es necesario mantener información sobre la posición relativa o absoluta de las palabras, por ello se hace uso de un proceso de codificación posicional.

2.6.4 La Función de Activación

Según Bhardwaj (Bhardwaj, 2018) la función de activación en cada neurona artificial decide si las señales entrantes han alcanzado el umbral y deben emitir señales para el siguiente nivel. Es crucial establecer la función de activación correcta debido al problema de desaparición del gradiente

Otra característica importante de una función de activación es que debe ser diferenciable. La red aprende de los errores que se calculan en la capa de salida. Se necesita una función de activación diferenciable para realizar la optimización de la propagación hacia atrás mientras se propaga hacia atrás en la red para calcular gradientes de error (pérdida) con respecto a los pesos, y luego optimizar los pesos en consecuencia, utilizando el descenso de gradiente o cualquier otra técnica de optimización para reducir el error. A lo largo del desarrollo y avances de las redes neuronales, se han usado varias funciones con distintas propiedades, actualmente las funciones de activación que son más utilizadas son las siguientes:

- **Rectified Linear Unit (ReLU, o Unidad lineal rectificada):** Una de las más utilizadas durante mucho tiempo, devolviendo x para valores positivos, 0 en caso contrario, representada por la ecuación 2.6.

$$\sigma(x) = (0, x) \ \ (Ec. 2.6)$$

- **GeLU**

La función de activación Gaussian Error Lineal Unit (GELU, o Unidad lineal de error gaussiano) es una función utilizada en redes neuronales, especialmente en modelos de deep learning y es la utilizada por Whisper. Fue propuesta por Hendrycks y Gimpel en 2016 y ha ganado popularidad debido a su efectividad en varias tareas en PLN y visión por computadora definida como la ecuación 2.7.

$$GELU(x) \ = \ 0.5x \ ((1 \ + tanh \ (\sqrt{\frac{2}{\pi}}(x + 0.044715x^3))) \ (EC. 2.7)$$

Seleccionar la función de activación más adecuada es un factor crítico en la efectividad de los modelos de aprendizaje profundo, ya que influye en su capacidad de aprendizaje, estabilidad y eficiencia computacional. En los últimos años, GELU se ha convertido en un método dominante, superando a funciones tradicionales como ReLU en diversas aplicaciones. El estudio llevado a cabo por Minhyeok Lee (Lee, 2023) presenta una investigación matemática rigurosa de la función de activación GELU, explorando en detalle sus propiedades de diferenciabilidad, acotación, estacionariedad y suavidad. Para los numerosos conjuntos de datos evaluados, el GELU superó la precisión del ELU y ReLU, lo que lo convierte en una alternativa viable a las no lineales anteriores.

En comparación con otras funciones de activación como ReLU y tanh, GELU ha demostrado mejorar el rendimiento de modelos en diversas tareas, especialmente en el procesamiento de lenguaje natural. Por

ejemplo, se utiliza en la arquitectura de modelos BERT (Bidirectional Encoder Representations from Transformers) desarrollados por Google, superando a ReLU en muchos aspectos.

2.7 Transfer Learnig

Transfer Learning, o aprendizaje por transferencia, es una técnica del aprendizaje automático (machine learning) que se centra en utilizar un modelo pre-entrenado en una tarea para aplicarlo a una nueva tarea relacionada. Este enfoque es particularmente útil cuando se dispone de grandes cantidades de datos en una tarea, pero se tienen pocos datos en la tarea objetivo. Al aprovechar el conocimiento adquirido en la tarea original, los modelos pueden aprender de manera más eficiente y con mejores resultados en la nueva tarea (Pan & Yang, 2010) ejemplificado en el estudio mencionado en los antecedentes titulado "Transfer learning en la detección de cáncer en la piel".

El concepto de aprendizaje por transferencia ha evolucionado significativamente a lo largo de las últimas décadas. En sus inicios, la transferencia de conocimiento entre tareas se realizaba principalmente a través de técnicas de aprendizaje multitarea, donde múltiples tareas se entrenaban simultáneamente para mejorar el rendimiento general. Sin embargo, el verdadero potencial del Transfer Learning se materializó con el avance de las redes neuronales profundas y la disponibilidad de grandes volúmenes de datos.

El advenimiento de modelos pre-entrenados de gran escala, como las redes neuronales para visión por computadora y los modelos de lenguaje profundo como BERT y GPT en procesamiento del lenguaje natural, ha permitido una transferencia de conocimiento más efectiva y eficiente. Estos modelos pueden ser reutilizados y ajustados para nuevas tareas, mejorando significativamente el rendimiento en situaciones con datos limitados (Weiss, Khoshgoftaar, & Wang, 2016).

A pesar de sus ventajas, el aprendizaje por transferencia enfrenta varios desafíos. Entre ellos se encuentran la selección del modelo base adecuado, el ajuste fino para evitar el *overfitting* y la adaptación a diferentes dominios de datos. El fine tuned puede llevar a que el modelo se sobreajuste a los datos de la tarea objetivo, especialmente si estos datos son limitados. Además, la adaptación de dominio puede ser compleja cuando las diferencias entre las distribuciones de datos de las tareas fuente y objetivo son significativas.

La investigación futura en Transfer Learning se centra en desarrollar métodos más robustos para el ajuste y la adaptación del modelo, así como en explorar su uso en tareas no supervisadas y semi supervisadas. También se está investigando cómo mejorar la interpretabilidad de los modelos transferidos y cómo

garantizar que la transferencia de conocimiento no introduzca sesgos o errores sistemáticos en la tarea objetivo (Torrey & Shavlik, 2010).

Un uso relevante en el campo de la transcripción de voz a texto es el estudio realizado por Kahn. (2020), donde se utilizó Transfer Learning para mejorar la precisión de la transcripción automática en diferentes idiomas. En este estudio, se emplearon modelos de redes neuronales recurrentes pre-entrenados en grandes corpus de datos de voz en inglés y se ajustaron para realizar transcripciones precisas en idiomas con menos recursos de datos. La investigación demostró que los modelos pre-entrenados pueden ser ajustados para lograr una alta precisión en la transcripción de voz a texto en varios idiomas, reduciendo significativamente el tiempo y los recursos necesarios para entrenar modelos desde cero.

2.8. Fine-tuning

El fine-tuning, o ajuste fino, como explica Howard & Ruder (2018) es una técnica dentro del campo del aprendizaje profundo que consiste en ajustar un modelo pre-entrenado para una tarea específica utilizando un conjunto de datos más pequeño y específico de esa tarea. Esta técnica es fundamental en el transfer learning y permite aprovechar el conocimiento adquirido por el modelo en una tarea anterior para mejorar su rendimiento en una nueva tarea, a menudo con menos datos disponibles.

Zeyer (2019) muestra que el fine-tuning se ha aplicado con éxito en diversas áreas, mostrando mejoras significativas en el rendimiento de los modelos en múltiples dominios:

Visión por Computadora: En la clasificación de imágenes, detección de objetos y segmentación semántica, el fine-tuning de modelos pre-entrenados en conjuntos de datos como ImageNet ha permitido obtener resultados de alta precisión incluso con conjuntos de datos específicos y limitados. Por ejemplo, un modelo pre-entrenado en ImageNet puede ajustarse finamente para clasificar imágenes de diferentes especies de plantas o animales en un conjunto de datos más pequeño y especializado.

Procesamiento del Lenguaje Natural: El fine-tuning ha revolucionado el procesamiento del lenguaje natural con la introducción de modelos pre-entrenados como BERT y GPT-3. Estos modelos, después de ser ajustados con datos específicos de tareas como el análisis de sentimientos, la clasificación de textos o la respuesta a preguntas, han demostrado un rendimiento superior en comparación con los modelos entrenados desde cero.

Reconocimiento de Voz: En el reconocimiento automático de voz, los modelos de deep learning pre entrenados en grandes corpus de datos de voz pueden ser ajustados finamente para mejorar la precisión en la transcripción de voz a texto en diferentes idiomas o dialectos, o para adaptarse a entornos acústicos específicos.

2.9. Open Source

El software de código abierto (Open Source) se refiere a programas cuyo código fuente está disponible públicamente, permitiendo a los usuarios y desarrolladores ver, modificar y distribuir el software libremente. Este enfoque promueve la colaboración, la transparencia y la innovación, facilitando el desarrollo de tecnología avanzada en diversas áreas, incluido el deep learning y el machine learning en general (Raymond, 2001).

En el contexto del deep learning, proyectos como TensorFlow, PyTorch y Keras han sido cruciales, proporcionando plataformas robustas para el desarrollo y la implementación de modelos de aprendizaje automático (Fitzgerald, 2000).

Este proyecto se ha centrado en utilizar herramientas open source o proyectos que hayan hecho uso de estas, debido a que el enfoque de código abierto ofrece numerosos beneficios que han impulsado su adopción generalizada en la comunidad deep learning:

Colaboración y Comunidad: El software de código abierto fomenta la colaboración entre desarrolladores de todo el mundo. Esto facilita la creación de soluciones innovadoras y la rápida resolución de problemas, ya que una comunidad global puede contribuir al desarrollo y mejora del software.

Transparencia y Seguridad: Al tener el código fuente disponible públicamente, el software de código abierto permite una mayor transparencia y seguridad. Los errores y vulnerabilidades pueden ser identificados y corregidos rápidamente por la comunidad, mejorando la confiabilidad del software.

Flexibilidad y Personalización: Los usuarios pueden modificar el software de acuerdo con sus necesidades específicas, añadiendo o eliminando funcionalidades según sea necesario. Esto es especialmente valioso en el desarrollo de modelos de aprendizaje profundo, donde los investigadores pueden experimentar con diferentes arquitecturas y técnicas.

Reducción de Costos: El software de código abierto suele estar disponible de forma gratuita, lo que reduce significativamente los costos de desarrollo y permite a las organizaciones invertir recursos en otros aspectos críticos de sus proyectos (Wheeler, 2007).

Aplicaciones de Open Source en Machine Learning.

Para Chollet (2015) el software de código abierto ha sido un facilitador crucial en el campo del aprendizaje automático y profundo, permitiendo a los investigadores y desarrolladores construir y compartir modelos avanzados de manera eficiente, creando herramientas como:

TensorFlow: Desarrollado por Google, TensorFlow es una biblioteca de aprendizaje automático de código abierto que ha sido ampliamente adoptada para el desarrollo de modelos de aprendizaje profundo. Proporciona una plataforma flexible y escalable para la investigación y producción, permitiendo a los usuarios implementar modelos complejos de manera eficiente.

PyTorch: Desarrollado por Facebook, PyTorch es otra biblioteca popular de aprendizaje profundo que se destaca por su facilidad de uso y capacidad para realizar cálculos en tiempo real. PyTorch ha ganado popularidad en la comunidad de investigación debido a su interfaz intuitiva y su compatibilidad con la construcción de redes neuronales dinámicas.

CAPÍTULO 3. METODOLOGÍA

En el presente capítulo se explica a detalle las decisiones tomadas para el desarrollo de la solución propuesta en el trabajo de investigación, que comprende desde la selección del modelo a través de un proceso de depuración, donde se realizan diversas pruebas con distintas herramientas para seleccionar la que mayor ventajas presente pese a las limitantes encontradas y luego el proceso de entrenamiento, siguiendo un acercamiento similar al que utilizó OpenAI para entrenar a Whisper: la búsqueda de datasets que mejor se acoplen a la idea del proyecto, formatearlos de tal manera que puedan ser procesados por el modelo, y pasando luego a la fase de experimentación y medición para medir la efectividad del modelo, en el que se espera ver una mejora luego del entrenamiento por transfer learning.

3.1. Selección de modelo

Inicialmente, se hizo una investigación sobre aplicaciones y proyectos que hacen uso de algún sistema de reconocimiento de voz o que directamente su propósito fuera directamente la transcripción de voz a texto, durante este periodo, encontramos una librería que marcó nuestro punto de partida para formar los criterios que utilizaremos para evaluar y descartar a los candidatos, dicha librería es SpeechRecognition (ver anexo B) desarrollada por Anthony Zhang, la cual tiene como propósito facilitar el poder consumir varios modelos y APIs de reconocimiento de habla, la lista se encuentra en el anexo B la cual provee de 13 candidatos.

Otro candidato que se agregó a la lista fue DeepSpeech, un proyecto open source desarrollado por Mozilla, con este último agregado, fueron en total 14 candidatos entre los cuales elegimos uno para ser usado en nuestra investigación.

Los criterios utilizados para elegir al modelo fueron los siguientes:

- Proyecto de código con acceso al público, puesto que buscábamos entender su funcionamiento
- Que su uso fuese gratuito
- Facilidad para entrenar al modelo
- Uso de tecnologías y métodos eficientes
- Acceso a detalles sobre su creación, desarrollo y entrenamiento

Con estos criterios, se descartaron los candidatos que fallaban en alguno de ellos. Aquellos modelos que son ofrecidos como servicios fueron los primeros en ser descartados, como por ejemplo Google Cloud

Speech API, Azure AI Speech y IBM Speech to Text, estos siguen un sistema de créditos para los cuales se cobra por cada minuto de transcripción realizadas, además que, al ser software propietario, es imposible tener acceso a los detalles de su funcionamiento.

Otros como Tensorflow y Wit.ai eran más complicados de usar que otros candidatos como Whisper, DeepSpeech y Google Speech Recognition, por lo que fueron descartados, además fueron descartados aquellos que están deprecados como Snowboy Hotword Detection y Microsoft Bing Voice Recognition. Los últimos candidatos fueron:

- Whisper
- DeepSpeech
- Google Speech Recognition
- Vosk
- CMUSphinx

De estos Vosk y CMUSphinx fueron descartados ya que no se contaban con los detalles de su desarrollo o funcionamiento. Por ello los candidatos finales fueron Whisper, Google Speech Recognition y DeepSpeech, con estos se hicieron pruebas preliminares detalladas en la sección 3.1.1.

Los resultados son similares entre los 3 modelos, sin embargo, se tuvo que descartar Google Speech Recognition por la falta de información que había, por lo que la decisión final fue tomada al estudiar las técnicas, métodos y herramientas utilizadas para el desarrollo de Whisper y DeepSpeech.

3.1.1 Pruebas preeliminares

Para el desarrollo de estas pruebas se utilizó el lenguaje Python y el editor de texto Visual Studio Code. La documentación de SpeechRecognition provee de algunos scripts utilizando los modelos a los que da soporte, haciendo uso como entrada el micrófono, utilizando estos scripts se hicieron pruebas con Google Speech Recognition, consiguiendo así poder transcribir lo que hablábamos por el micrófono. También se hicieron pruebas con Whisper, utilizando la documentación oficial, en las que se hicieron pruebas con un script que permitió transcribir audios en formato wav y formato mp3, Whisper también permite utilizar la línea de comandos para transcribir audios que nos permite hacerlo de manera mucho más sencilla con un solo comando, asimismo, se hicieron pruebas con DeepSpeech usando un script que hace uso del micrófono.

Las pruebas fueron ejecutadas en una computadora con las siguientes especificaciones:

- **Procesador**: AMD Ryzen 7 3700X 8-Core Processor 3.60 GHz
- **RAM**: 32.0 GB
- **GPU**: NVIDIA GeForce RTX 3060
- **Almacenamiento**: 1.5 TB
- **Sistema operativo**: Windows 64-bit, x64-based processor

Todas estas pruebas demostraron resultados similares en la precisión de la transcripción, la velocidad en la que se completaba esta misma y en sus limitaciones, como, por ejemplo, que ciertas palabras o acentos comunes de algunas regiones de El Salvador no eran reconocibles para ninguno de ellos. No obstante, de entre todas estas pruebas, Whisper resaltaba al ser el que más nos permitía configurar ciertos parámetros, por ejemplo, el idioma en el que se va a transcribir y la potencia del modelo (puesto que Whisper tiene 5 modelos, cada uno más potente que el anterior), también Whisper genera no solo una transcripción, sino que también información que puede resultar útil según el caso de aplicación, como por ejemplo una transcripción acompañada de las marcas de tiempo.

3.2. Selección de dataset

En busca de cumplir nuestro objetivo, se realizó una investigación sobre los posibles candidatos al dataset que contuviera la información y el contexto más similar al caso de uso de esta investigación. Existen varias plataformas para encontrar datasets, entre ellas Kaggle, Openslr y Hugging Faces, lugares donde se encuentran datasets para entrenar diversidad de modelos para resolver una tarea específica, para el propósito de esta investigación, el criterio para elegir el dataset es:

- Contar con audio/transcripción es indispensable
- Enfocado en el lenguaje español
- Cantidad de horas de audio
- Calidad del audio
- Diversidad de dialectos

3.3. Entrenamiento del modelo

Para empezar a entrenar el modelo, se tiene que respetar un formato definido por el modelo para aceptar el dataset, en el caso de Whisper, estos audios tienen que estar en formato wav y la duración de los audios no puede ser mayor a 30 segundos, afortunadamente el dataset elegido ya cumple este formato.

Para ingresar el dataset al modelo, se necesita un archivo csv con las rutas a los audios para que puedan ser procesados y la transcripción de dicho audio para poco a poco que el modelo vaya aprendiendo a predecir e interpretar las palabras.

Durante este periodo, surgieron varias limitaciones, la memoria juega un papel muy importante para realizar una tarea de este estilo, el equipo usado no podía soportar procesar el dataset por completo, al no poder realizar mejoras en el hardware, se optó por solucionar el problema realizando entrenamientos parciales de 25% del dataset.

3.4 Experimentos y mediciones

Para medir la efectividad del modelo entrenado, se comparará las métricas mencionadas en la sección 2.5 del modelo pre y post entrenamiento, para ello se transcribieron manualmente dos conferencias distintas del Congreso de Ingeniería y Arquitectura (CONIA) 2023, que es aproximadamente una hora de muestra para la realización de este experimento.

Para ello fue necesario normalizar los audios en el formato que Whisper espera, mencionado anteriormente en la sección 2.4, en total resultaron 120 audios de 30 segundos en formato .wav, estos se pueden encontrar en el repositorio del anexo A.

El experimento consistirá en transcribir todos los audios con ambos modelos, obtener las métricas mencionadas en la sección 2.5 para cada transcripción y obtener la media para cada modelo, se espera ver una mejora con respecto al modelo base.

3.5. Normalización de datos

Ambos modelos generan transcripciones con signos de puntuación como puntos, comas y signos de interrogación, estos deben ser eliminados puesto que su presencia en una transcripción pero no en la otra afectaría el cálculo de las métricas cuando en realidad no son considerados errores, además ambos modelos también generan transcripciones con palabras correctamente acentuadas, no obstante, para evitar afectar el

cálculo debido a cualquier error humano cometido al transcribir las conferencias con palabras incorrectamente acentuadas, se decidió eliminar todos los acentos.

CAPÍTULO 4. RESULTADOS

4.1. Modelo elegido

Las pruebas preeliminares demostraron resultados similares en la precisión de la transcripción, la velocidad en la que se completaba esta misma y en sus limitaciones, como, por ejemplo, que ciertas palabras o acentos comunes de algunas regiones de El Salvador no eran reconocibles para ninguno de ellos. No obstante, de entre todas estas pruebas, Whisper resaltaba al ser el que más nos permitía configurar ciertos parámetros, por ejemplo, el idioma en el que se va a transcribir y la potencia del modelo (puesto que Whisper tiene 5 modelos, cada uno más potente que el anterior), también Whisper genera no solo una transcripción, sino que también información que puede resultar útil según el caso de aplicación, como por ejemplo una transcripción acompañada de las marcas de tiempo.

DeepSpeech fue desarrollado usando redes neuronales recurrentes, mientras que Whisper hace uso de técnicas que han mejorado el desarrollo de los sistemas de reconocimiento de voz, como las redes neuronales transformers, que se explican en la sección 2.3.3 y también una función de activación superior: GELU, explicada en la sección 2.6.4. Por esta mejora en las tecnologías y técnicas utilizadas además de ser una propuesta novedosa, se optó por usar Whisper para el desarrollo de este trabajo de investigación, así Whisper responde satisfactoriamente a los criterios de la sección 3.1.

Whisper debido a que su formato open source permite la versatilidad necesaria para poder exponer el funcionamiento de un modelo de machine learning de reconocimiento de voz y ofreciendo las ventajas mencionadas en la sección 2.7. Es una red entrenada por OpenAI, una compañía de investigación y desarrollo de inteligencias artificiales, con productos lanzados como ChatGPT, DALL-E y su red neuronal llamada Whisper para reconocimiento de voz lanzada en septiembre de 2022. Esta red neuronal se caracteriza por ser open source, licenciada bajo los permisos de la Licencia MIT lo cual permite a cualquier persona en el mundo acceder al código fuente de la red neuronal para su uso privado, modificación, distribución y uso comercial, además de permitir enviar mejoras en el código que deben pasar por un proceso de aprobaciones para poder ser agregados al código fuente. Whisper además cuenta con modelos pre entrenados que varían en robustez dependiendo de la precisión deseada, por supuesto, a cambio de un mayor requerimiento de procesamiento.

4.2. Dataset Elegido

Durante este periodo, se encontró un dataset precisamente apegado a estos criterios, dicho dataset se encuentra en Openslr y es generado a partir de varias charlas TEDx en español, en total 11,000 audios fueron generados. Las charlas TEDx son un espacio donde expertos de todo el mundo comparten ideas frente a un público, estos eventos son organizados por TED, una organización sin ánimo de lucro que organiza eventos en pro de la educación. Estas charlas son similares a las presentadas a las del CONIA porque siguen el mismo formato: un experto expone un tema ante un público, por ello mismo y además de cumplir con los criterios mencionados en la sección 3.2, con 24 horas de audio, consideradas suficiente para nuestro propósito., se hizo uso este dataset para este proyecto. Dicho dataset se puede encontrar en el anexo B.

4.3. Resultados del entrenamiento

Este proceso tomó aproximadamente 14 en horas en realizarse, ejecutado en el mismo equipo en el cual se ejecutaron las pruebas preliminares de la sección 3.1.1 que dio como resultado 4 carpetas que tienen codificada la información de lo aprendido para realizar la transcripción, estos son unos archivos .pt y .pth que son interpretados por PyTorch y son los que procesan los archivos de audio y obtener la transcripción.

Debido a que el entrenamiento se tuvo que hacer en 4 partes es la razón por la cual se generan 4 carpetas, esto afecta el resultado del modelo, ya que esto hace que el modelo entrenado genere 4 transcripciones similares, cada una generada con lo aprendido en cada cuarto del dataset, con lo cual genera una transcripción promedio que es la considerada como el resultado final.

Es importante mencionar que este modelo cuenta con las mismas limitantes que el modelo padre: solo puede procesar audios de 30 segundos como máximo, que deben estar en formato .wav

4.4. Resultados del experimento

Los resultados de las métricas se muestran en la tabla 4.1 para cada modelo, como se puede apreciar en la tabla 4.1 el modelo entrenado resulta tener una mejora que varía entre 11% y el 17%, cometiendo menos errores que el modelo base de Whisper sin haber sido entrenado, la comparación se hace más evidente en la figura 4.1, donde se muestra en azul el desempeño de Whisper y en rojo el modelo entrenado, notar que Whisper comete más errores en cada una.

Tabla 4.1. Resultados de las métricas calculadas

	Whisper	**Entrenado**
WIP	63.77	81.17
WIL	36.23	18.83
WER	27.43	12.42
MER	26.65	12.15
CER	16.84	5.49

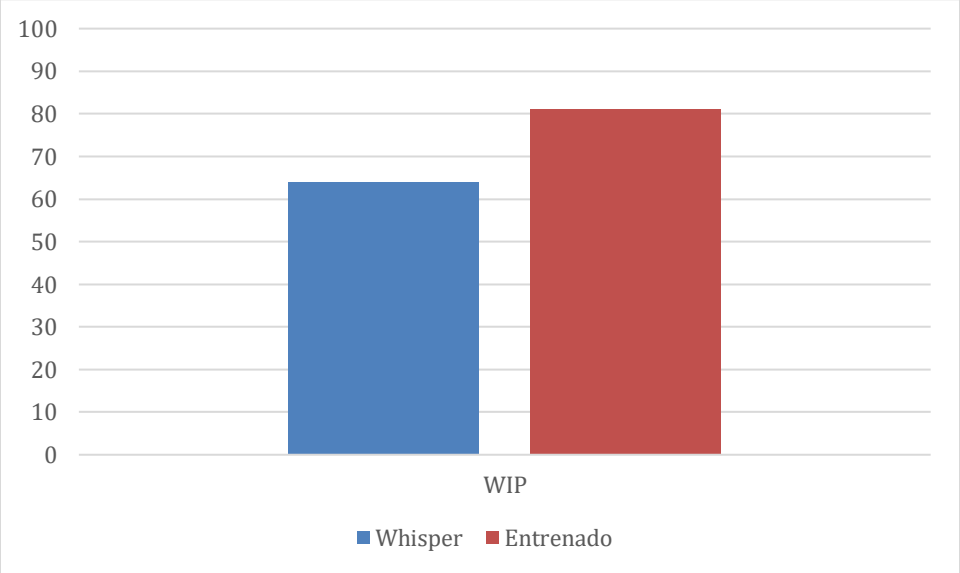


Figura 4.1. Gráfico de columnas comparando la métrica WIP, donde se busca sea el máximo posible. Calculado usando la biblioteca jiwer

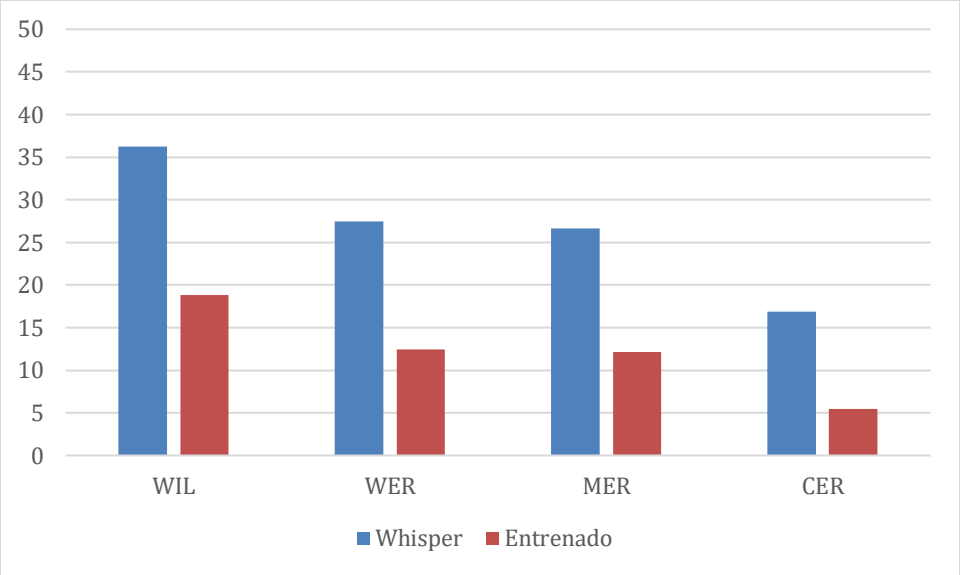


Figura 4.2. Gráfico de columnas agrupadas por las métricas WIL, WER, MER y CER, donde se busca sea el mínimo posible. Calculado usando la biblioteca jiwer

Para calcular estos valores se hace uso de las transcripciones generadas y las de referencia, y se procede a contar los errores, a manera de ejemplificar, se muestra un ejemplo en la figura 4.2:

```
sentence 51
REF: que iba a andar yo pensando en mire voy a **** *** uti tengo que aprender si tips estas tecnologias
HYP:   y van a   dar yo pensando  y mire voy a util voy   a tengo que aprender si  tex estas tecnologias
       S   S       S             S              I   I   S                         S
```

Figura 4.3: Fragmento de la comparación entre una transcripción generada y su referencia, mostrando los errores de inserción y sustitución en ese caso.

Los resultados son de acuerdo con lo esperado, una mejora en cuanto a la cantidad de errores que cometió el modelo entrenado con respecto a su padre en el contexto de charlas STEM.

CAPÍTULO 5. CONCLUSIONES

5.1. Conclusiones

I. Los modelos de Automatic Speech Recognition (ASR, Reconocimiento de Voz Automático en español) han demostrado ser capaces de interpretar, transcribir e incluso traducir con efectividad un discurso oral, si bien aún queda trabajo a futuro para refinar y perfeccionar, actualmente ya se encuentran en un estado capaz de poder resolver y responder a los distintos problemas y casos de uso en los que sea necesarios, siempre y cuando se cuente con los datos necesarios para adecuar al modelo al contexto deseado.

II. La técnica *transfer learning* demostró ser totalmente efectiva, reduciendo los recursos necesarios para obtener un modelo capaz de resolver la tarea que se le encomienda incluso mejor que el modelo con el cual se partió como mostraron los resultados de la sección 4.4.

III. La comunidad ha publicado una basta cantidad de datasets con variedad de fuentes y contextos que pueden ser útiles para el entrenamiento de un modelo como el utilizado para esta investigación, lo cual ahorra considerablemente tiempo y esfuerzo que pueden ser dedicados para dicho proceso y no para generar un dataset, que es un proceso con un alto componente manual y susceptible a errores humanos.

IV. Existen diversidad de arquitecturas para los distintos modelos de machine learning, cada una acoplándose a la tarea que realizan, sin embargo, existen métricas y medidas con las cuales determinar la superioridad de una que de otra para lo que se requiera, como los componentes que hicieron que se eligiera Whisper como el modelo utilizado mencionados en la sección 4.1.

V. El modelo entrenado demostró ser superior en el caso de aplicación con un dataset de considerable menor duración con respecto al base, demostrando así que no solo no es necesaria un entrenamiento de larga duración, sino que incluso puede ser preferible para evitar "*overfitting*", pues obtuvo mejores resultados, extendemos el uso del modelo al lector para el caso que cree conveniente que puede ser encontrado en el repositorio del anexo A.

VI. Se mostró el proceso y los recursos con los cuales se puede entrenar un modelo de machine learning para resolver una tarea en especifica, demostrando así la factibilidad y los resultados de dedicar recursos para desarrollar un sistema que obtenga resultados para tareas que pueden ser automatizadas.

5.2. Recomendaciones

I. En caso de necesitar un sistema de ASR, se recomienda comprobar la efectividad de un modelo ya entrenado para el caso de uso, pues este puede resultar suficiente para la tarea que se desea que resuelva, en vez de destinar recursos para el entrenamiento de un modelo que no es necesario.

II. Durante nuestra investigación, no encontramos un dataset que incluyera o se centrara en el español salvadoreño, por ello, dejamos como trabajo futuro a investigadores que consideren la creación de este y la utilización que se le puede dar en los distintos ámbitos y actividades que pueden dar solución en El Salvador.

III. Si se va a entrenar un modelo de ASR, es preferible que el equipo utilizado tenga la potencia más alta que los recursos que se disponga puedan proveer, idealmente, se recomienda utilizar hardware más potente que el utilizado en este proyecto (ver anexo A).

IV. Se recomienda comprobar la efectividad del modelo en un contexto distinto al planteado, queda como trabajo futuro comprobar si el modelo responde con la misma mejora en contexto que no tengan que ver con áreas STEM.

V. En caso de no disponer de un dataset que se adecue el caso de uso, no descartar la idea de crearlo manualmente, que si bien es un proceso que requiere de uso de información y tiempo de creación, su uso para desarrollar un modelo entrenado capaz de resolver las tareas que hace un humano es digno de considerar.

VI. Queda como trabajo futuro integrar el modelo a una versión utilizable en una aplicación con interfaz gráfica, así como también un mecanismo para poder aceptar más formatos de archivos de audio y poder enviar los archivos en fragmentos de 30 segundos para poder obtener la transcripción de archivos de mayor duración

GLOSARIO

Atención: En redes neuronales, se le conoce como atención a la capacidad de un nodo de una red de poder enfocarse y retener información de un dato o conjunto de ellos.

Data mining: También conocida como minería de datos, son todas aquellas técnicas y tecnologías que permiten determinar patrones a partir de un conjunto de datos.

Espectrograma: Es la representación visual de un audio en frecuencia sobre el tiempo

Inteligencia artificial: Conocida así a todo sistema que realiza una función compleja similar a como las que realizan los humanos

Robustez: En machine learning, se dice que un sistema es robusto cuando tiene la capacidad de conseguir resultados precisos para datos que escapan al dominio de los de su entrenamiento

Software: Todo componente de computación intangible, por ejemplo: Sistema operativo, aplicación, programa, etc.

Transcripción: Resultado de escribir las palabras dichas en un audio

Overfitting: Fenómeno que sucede al entrenar tanto un modelo de machine learning que se ajusta tan bien a los datos del entrenamiento que pierda su capacidad de dar resultados con datos nuevos.

REFERENCIAS

Amador, 2020, Algoritmo de traducción de voz a texto como herramienta educativa para personas con sordera. https://repositorio.unitec.edu/bitstream/handle/123456789/12035/Algoritmo%20de%20traducción%20de%20voz%20a%20texto%20como%20herramienta%20educativa%20para%20personas%20con%20sordera.pdf?sequence=1&isAllowed=y

Chaudhari, S., Polatkan, G., Ramanath, R., & Mithal, V. (2019). *An attentive survey of attention models*. Recuperado de: https://arxiv.org/abs/1904.02874

Cho, K., Courville, A., & Bengio, Y. (2015). Describing multimedia content using attention-based encoder-decoder networks. IEEE Transactions on Multimedia, 17(11), 1875-1886.

Chollet, F. (2015). Keras: The Python Deep Learning library. Retrieved from https://keras.io

Connor, J. T., Martin, R. D., & Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. IEEE Transactions on Neural Networks, 5(2), 240-254.

Di, W., Bhardwaj, A., & Wei, J. (2018). Deep Learning Essentials. Packt Publishing.

Flores Cortez, O. O., & Cortez Reyes, R. A. (2018). Extracción de conocimiento a partir de texto. Universidad Tecnológica de El Salvador. Recuperado de: https://www.utec.edu.sv/vips/uploads/investigaciones/investigacion78.pdf

Firmansyah, M. H., Paul, A., Bhattacharya, D., & Urfa, G. M. (2020). AI based Embedded Speech to Text Using Deepspeech. arXiv preprint arXiv:2002.12830.

Fitzgerald, B. (2000). The transformation of open source software. MIS Quarterly, 24(3), 587-598.

Ghorbani, B., Firat, O., Freitag, M., Bapna, A., Krikun, M., Garcia, X., ... & Cherry, C. (2021). Scaling laws for neural machine translation. arXiv preprint arXiv:2109.07740.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 27.

Guyon, I., & Pereira, F. (1995, August). Design of a linguistic postprocessor using variable memory length Markov models. In Proceedings of 3rd International Conference on Document Analysis and Recognition (Vol. 1, pp. 454-457). IEEE.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., ... & Ng, A. Y. (2014). Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567.

Hannun, Case, Casper., Catanzaro, B., Diamos, G., Elsen, E., ... & Ng, A. Y. (2014). Deep Speech: Scaling up end-to-end speech recognition.

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 328-339

Introduction to audio data - Hugging Face Audio Course. (s. f.). https://huggingface.co/learn/audio-course/chapter1/audio_data

Izaurieta, F., & Saavedra, C. (2000). Redes neuronales artificiales. Departamento de Física, Universidad de Concepción Chile.

Julius. (2020). Retrieved 10 November 2020, from https://julius.osdn.jp/en_index.php

Kahn, J., Lee, A., Hori, T., & Keshet, J. (2020). Self-training for end-to-end speech recognition. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

Kaldi. (2020). Retrieved 10 November 2020, from https://kaldi asr.org/doc/files.html

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

Lee, M. (2023). GELU activation function in deep learning: a comprehensive mathematical analysis and performance. arXiv preprint arXiv:2305.12073.

Lopezosa, C., Codina, L., & Boté-Vericad, J. J. (2023). Testeando ATLAS. ti con OpenAI: hacia un nuevo paradigma para el análisis cualitativo de entrevistas con inteligencia artificial.

Madahana, M., Khoza-Shangase, K., Moroe, N., Mayombo, D., Nyandoro, O., & Ekoru, J. (2022). A proposed artificial intelligence-based real-time speech-to-text to sign language translator for South African official languages for the COVID-19 era and beyond: In pursuit of solutions for the hearing impaired. South African Journal of Communication Disorders, 69(2), a915. https://doi.org/10.4102/sajcd.v69i2.915

Martínez González, C. (2021). Transfer Learning para la detección de cáncer de piel. UNIR, Universidad Internacional de La Rioja. Recuperado de: https://reunir.unir.net/bitstream/handle/123456789/12326/Martínez%20González%2c%20Carlos.pdf?sequence=1&isAllowed=y

Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

Moya Córdoba, D. (2021). Desarrollo de una herramienta para la conversión de voz a texto. Universidad Politécnica de Madrid. Recuperado de: https://oa.upm.es/66298/1/TFG_DIEGO_MOYA_CORDOBA.pd

OpenAI. (2022). Whisper: OpenAI's speech recognition model. Recuperado de: https://cdn.openai.com/papers/whisper.pdf

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345-1359.

Park, Y., Patwardhan, S., Visweswariah, K., & Gates, S. C. (2008, September). An empirical analysis of word error rate and keyword error rate. In *Interspeech* (Vol. 2008, pp. 2070-2073).

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... & Silovsky, J. (2011). The Kaldi speech recognition toolkit. In IEEE 2011 workshop on automatic speech recognition and understanding (No. CONF). IEEE Signal Processing Society.

Radford, A., Narasimhan, K., & Salimans, T. (2023). Whisper: A general-purpose speech recognition model. OpenAI

Raymond, E. S. (2001). The cathedral & the bazaar: Musings on Linux and open source by an accidental revolutionary. O'Reilly Media.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 3(3), 210-229.

Shmyrev, N. (2020). About CMUSphinx. Retrieved 10 November 2020, from https://cmusphinx.github.io/wiki/about/

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems, 27, 3104-3112.

Torrey, L., & Shavlik, J. (2010). Transfer learning. In Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques (pp. 242-264). IGI Global.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need*. Recuperado de: https://arxiv.org/pdf/1706.03762

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big Data, 3(1), 1-40.

Wheeler, D. A. (2007). Why open source software / free software (OSS/FS)? Look at the numbers! Recuperado de http://www.dwheeler.com/oss_fs_why.html

Zeyer, A., Irie, K., Schlüter, R., & Ney, H. (2019). Improved training of end-to-end attention models for speech recognition. Proc. Interspeech 2019, 22-26.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. Communications of the ACM, 64(3), 107-115.

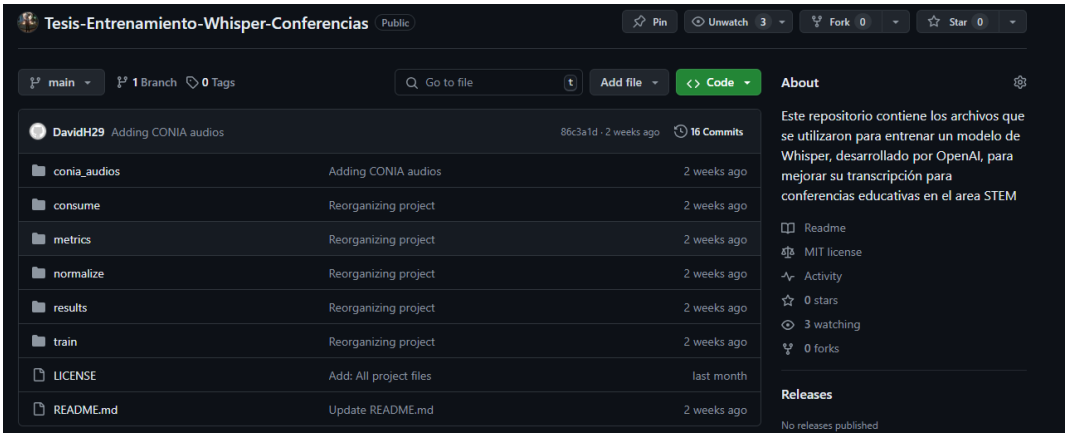
Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2020). Learning transferable architectures for scalable image recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8697-8706

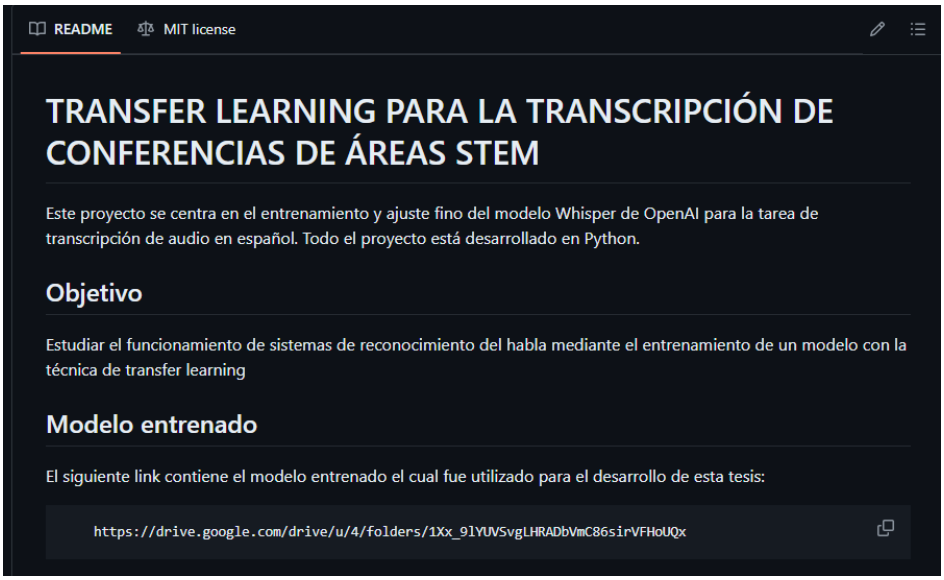
ANEXO A

ENLACE AL REPOSITORIO

En este se encuentran todos los archivos utilizados para el entrenamiento y el cálculo de métricas, también se encuentra un manual de uso en el archivo READ.me

https://github.com/DavidH29/Tesis-Entrenamiento-Whisper-Conferencias





ANEXO B

ENLACE AL DATASET ELEGIDO

Creado por Carlos Hernández, perteneciente a la Universidad Nacional Autónoma de Mexico en la Ciudad de México, creado en 2019, creado a partir de múltiples conferencias TEDx impartidas por hispanohablantes
https://www.openslr.org/67/

ANEXO C

LISTA DE MODELOS INTEGRADOS EN SPEECHRECOGNITION

La versión utilizada de la librería es V3.10.4, la cual integra los siguientes modelos:

- CMU Sphinx
- Google Speech Recognition
- Google Cloud Speech API
- Wit.ai
- Microsoft Azure Speech
- Microsoft Bing Voice Recognition
- Houndify API
- IBM Speech to Text
- Snowboy Hotword Detection
- Tensorflow
- Vosk API
- OpenAI whisper
- Whisper API

Para más información: https://pypi.org/project/SpeechRecognition/