Universidad Autónoma de Ciudad Juárez

Instituto de Ingeniería y Tecnología

Departamento de Ingeniería Eléctrica y Computación

Maestría en Cómputo Aplicado

"Módulo de detección de errores de pronunciación del idioma inglés utilizando una red transformadora integrado en un chatbot"

Tesis para obtener el grado de Maestro en Cómputo Aplicado

Marcos Eduardo Martínez Quezada

"Becado por el Consejo Nacional de Ciencia y Tecnología"

Bajo la dirección de la

Dra. Julia Patricia Sánchez Solís

y la codirección del

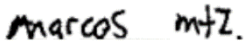
Dr. Rogelio Florencia Juárez

Ciudad Juárez, Chihuahua, octubre 2021

Declaración de originalidad

Por medio de la presente, yo, Marcos Eduardo Martínez Quezada, declaro ser el autor del presente documento de tesis. La redacción es producto original de mi trabajo. No infringe derechos de terceras personas tales como derechos de autos, patentes entre otros. Por lo que me declaro titular del mismo.

Así mismo, declaro que se referencia explícitamente cualquier cita o resumen de otros autores, con sus publicaciones correspondientes, a lo largo del documento. Acepto cualquier reclamación, si se presenta, en cuanto a derechos de autor de parte de terceros, tomando como propia la responsabilidad que esto conlleva.



Ing. Marcos Eduardo Martínez Quezada

Agradecimientos

Agradezco a mis asesores de tesis, Dra. Julia Patricia Sánchez Solís y Dr. Rogelio Florencia Juárez, po el conocimiento brindado a lo largo de la maestría. A todos los profesores que día con día compartieron su tiempo, consejos y sabiduría. A la Universidad Autónoma de Ciudad Juárez por haberme brindado la oportunidad de realizar mis estudios de maestría. Al Consejo Nacional de Ciencia y Tecnología (CONACyT), sin el cual esta investigación no sería posible.

Dedicatoria

Dedico este trabajo principalmente a mi madre por darme su apoyo y motivación diariamente. A mi asesor de tesis, Dr. Rogelio Florencia Juárez por haber dedicado incontables horas a la realización de esta investigación. A mis hermanas por ser una fuente de continuo aprendizaje y mis modelos a seguir. A todos aquellos que estuvieron conmigo durante el proceso de la maestría y el desarrollo de esta tesis.

Índice de contenido

Índice de figuras

Índice de tablas

Resumen

El proceso de globalización en cualquier ámbito ha requerido una mayor capacitación personal. El dominio del idioma inglés es uno de los factores de preparación mayormente demandados por las empresas. Sin embargo, existen personas a las que se les dificulta desarrollar las habilidades requeridas para dominar un nuevo idioma, de las cuales, la pronunciación es una de ellas. Algunos estudiantes que están aprendiendo un segundo idioma no cuentan con un compañero de estudio que les ayude a mejorar la habilidad del habla mediante la práctica.

Existen herramientas computacionales empleadas para ayudar a aprendices de un segundo idioma a mejorar su habilidad de pronunciación. En las últimas décadas fue ampliamente utilizado el algoritmo *Goodness of Pronunciation* (*GOP*), para realizar esta tarea, sin embargo, su funcionamiento se veía mermado al analizar oraciones largas. Posteriormente, la propiedad de memoria de las *Redes Neuronales Recurrentes* (RNN, por sus siglas en inglés) solucionaría parcialmente dicho problema ya que, al igual que el algoritmo *GOP*, se perdía información en secuencias demasiado largas. Con la llegada de la red transformadora se logró procesar secuencias completas con una mejora en el rendimiento, además sin perder información. Este tipo de red fue empleada en un inicio para problemas de texto, sin embargo, comenzó a utilizarse para la detección de voz gracias a su rendimiento. Adicionalmente, se suele utilizar un algoritmo de alineación de secuencias para lograr implementar la detección de errores de pronunciación.

Esta investigación se enfoca en el desarrollo de un sistema de detección de errores de pronunciación empleando técnicas de Inteligencia Artificial con la finalidad de ayudar a los usuarios a mejorar su pronunciación del idioma inglés estadounidense. Principalmente fueron tres las tecnologías implementadas: 1) un modelo reconocedor de voz utilizando una red transformadora, 2) un algoritmo de alineación de cadenas y 3) un chatbot desarrollado con el Lenguaje de Etiquetado de Inteligencia Artificial. La combinación de lo anterior dio como resultado un sistema capaz de mostrar al usuario las palabras en donde se detectaron errores de pronunciación.

I. Planteamiento del problema

En este capítulo se abordan los temas que ayudarán a comprender la temática del problema de investigación. Inicialmente se da una breve introducción sobre el tema y, posteriormente, se presenta la descripción del problema, seguida de los objetivos, la justificación, los alcances y el impacto de esta investigación.

1.1 Introducción

El proceso de globalización que se ha presentado durante los últimos años requiere que las ciudades sean cada vez más competitivas en diferentes ámbitos como el empresarial, educativo o de investigación. Conforme se busca aumentar la competitividad surgen brechas en cuestión de las habilidades, las cuales representan grandes obstáculos en donde el dominio de un segundo idioma, como el inglés es uno de ellos [1].

Existen diversas herramientas que ayudan en la enseñanza de un nuevo idioma, una de ellas son los chatbots. Los entornos en los que estos se pueden utilizar son muy diversos. Pueden ser implementados en comercio electrónico, servicio al cliente e incluso en educación. En esta última área, han sido desarrollados sistemas que permiten incrementar o mejorar las habilidades que se requieren para aprender un idioma. Los sistemas de *Aprendizaje de Idiomas Asistido por Computadora* son utilizados en la enseñanza de un idioma sin la necesidad de un profesor o salón de clases. Estos incluyen lecciones y ejercicios para mejorar las habilidades como la gramática, vocabulario o pronunciación. Los sistemas de práctica de pronunciación son una subárea del mencionado anteriormente y este se centra en aumentar las habilidades de la pronunciación.

Los chatbots utilizados para el aprendizaje de idiomas han sido implementados con buenos resultados. En [2] se aborda un estudio sobre la posibilidad de utilizar chatbots como un medio para aprender idiomas. Se concluyó que estos tienen gran potencial para ser utilizados en este ámbito. Algunas de las ventajas descritas reportan que los estudiantes tienden a sentirse más relajados al hablar con una computadora que con una persona, así como poder repetir el mismo material una y otra vez. El autor y su equipo desarrollaron un chatbot llamado *Gengobot* el cual tiene como objetivo enseñar gramática japonesa a estudiantes de nivel básico. Su funcionamiento se limitaba solo a la interacción mediante la tarea de preguntas y respuestas. Esta información se almacenó en una base de datos para posteriormente utilizar un *framework* de PHP para realizar las consultas. Este chatbot se empleó para la verificación de gramática.

Por otro lado, el *Reconocimiento Automático de Voz* (ASR, por sus siglas en inglés) es utilizado en sistemas de *Práctica de Pronunciación Asistido por Computadora* (CAPT, por sus siglas en inglés) con la finalidad de ayudar a los aprendices a mejorar la pronunciación. Para ello, se emplean modelos de *Inteligencia Artificial* (IA) que detectan los errores. El objetivo de los sistemas CAPT es evaluar la pronunciación, y de ser requerido, dar retroalimentación al usuario [3]. Los sistemas CAPT se basan en modelos ASR para conseguir su objetivo. Existe una variedad de técnicas que se utilizan para el desarrollo de un modelo ASR que van desde modelos estadísticos que modelan el lenguaje en estados de transición para hacer el reconocimiento de voz, hasta modelos que solo necesitan la entrada de la señal de voz para realizar una transcripción.

1.2 Descripción del problema

La pronunciación suele ser la habilidad que causa la mayoría de los problemas durante el proceso de aprendizaje de un segundo idioma. Diversos factores tales como el acento, motivación, edad, personalidad o falta de práctica pueden impedir su desarrollo [4]. Este último elemento puede caracterizarse por la ausencia de un compañero de aprendizaje, timidez al hablar e incluso una retroalimentación incorrecta. La interacción con una persona experta en el idioma que ayude a los estudiantes a corregir los errores es una estrategia que los beneficia. Sin embargo, en ocasiones no se dispone de una persona experta del idioma, por lo que se necesitan otros medios para lograr mejorar la habilidad de pronunciación.

Existen herramientas informáticas para el aprendizaje de idiomas que integran reconocimiento automático de voz en módulos de práctica de pronunciación, tales como los sistemas CAPT, capaces de detectar errores de pronunciación y dar retroalimentación al usuario sobre ellos [5]. Por otro lado, en el trabajo de [6] se menciona que los chatbots pueden ser compañeros apropiados para el aprendizaje. Considerando lo anterior, una integración de ambas tecnologías, ASR y chatbots, podría resultar útil para enfatizar el rubro de pronunciación, emulando la interacción con un profesor que podría brindar retroalimentación del ejercicio. De modo que, permita al usuario practicar tantas veces como lo desee o necesite y de forma gratuita.

1.3 Objetivos

A continuación, se presenta el objetivo general y los objetivos específicos que se plantearon para el desarrollo de esta investigación.

1.3.1 Objetivo general

Implementar un módulo de detección de errores en la pronunciación del idioma inglés estadounidense para su integración en la arquitectura de un chatbot.

1.3.2 Objetivos específicos

- Conformar el conjunto de datos de audio diseñado para evaluar la pronunciación.
- Entrenar un modelo de reconocimiento automático de voz con el conjunto de datos de audio seleccionado haciendo uso de una red Transformadora.
- Implementar un algoritmo de alineación de cadenas para encontrar posibles errores de pronunciación en el idioma inglés.
- Integrar los módulos de reconocimiento de voz y de detección de errores en la arquitectura de un chatbot.

1.4 Justificación

El proceso de aprendizaje de un segundo idioma exige la práctica continua para desarrollar la habilidad del habla. El dominio de una segunda lengua, en este caso el idioma inglés, tiene beneficios laborales en un mundo cada vez más conectado debido a la globalización. Sin embargo, existen diversos factores que podrían comprometer el desarrollo de dicha habilidad. Hay casos en los cuales no se dispone de un compañero de estudio el cual ayude a mejorar la pronunciación. Lo anterior puede ser provocado por horarios pocos flexibles o incluso altos costos de los cursos. Además, el usuario debe tener la habilidad de estructurar sus propias frases que desea practicar y no solo hacerlo con contenido predefinido. Al encontrarse con estos impedimentos, es oportuno la utilización de una entidad virtual que ayude a mejorar la pronunciación del idioma inglés.

Con el desarrollo de este proyecto de investigación se pretende ayudar a los aprendices del idioma inglés, cuyo idioma nativo es el español mexicano, a mejorar su habilidad del habla mediante un sistema de detección de errores de pronunciación. Esto beneficiaría a aquellos estudiantes los cuales no les es posible practicar el idioma debido a la falta de un compañero de estudio o incluso a aquellos que presentan timidez al hablar. Al estar en un ambiente libre de estrés los estudiantes practicarían para mejorar su confianza al hablar, además de recibir retroalimentación.

1.5 Alcances y limitaciones

A continuación, se presentan los alcances y limitaciones del trabajo de investigación.

1.5.1 Alcances

Entre los alcances principales que abarcan este proyecto de investigación se encuentran:

- El sistema desarrollado está pensado para toda aquella persona que busque desarrollar o mejorar la pronunciación del idioma inglés.
- Se permite el uso de oraciones propias al momento de evaluar la pronunciación.
- La entrada del audio de voz se produce en tiempo real.
- La retroalimentación se realiza mediante un chatbot que indica los errores de pronunciación a nivel palabra.
- El sistema desarrollado se puede ejecutar en cualquier sistema operativo en donde se pueda instalar lo mencionado en el punto anterior.

1.5.2 Limitaciones

Durante el desarrollo de la presente investigación se encontraron ciertas limitantes, las cuales se describen a continuación:

- Durante el entrenamiento del modelo ASR se utilizaron audios en inglés nativo y no nativo. Los audios en inglés no nativo pertenecen a personas cuyo idioma materno es el español. Por tal motivo, la exactitud del reconocedor de voz podría verse afectada con usuarios los cuales su idioma nativo no es español.
- El sistema requiere de la instalación del *toolkit ESPnet* y la librería de Python-Levenshtein.
- Solo es posible procesar audios con la extensión .WAV
- Se prefiere que el usuario tenga un nivel intermedio del idioma y conocimiento básico de gramática.
- No se enseña inglés a los usuarios.
- La base de conocimiento del chatbot no está construida para entablar conversaciones de cualquier tema.
- No se considera la acentuación léxica ni el acento tonal al momento de detectar errores en la pronunciación.

1.6 Impacto

Las áreas de impacto al desarrollar este proyecto de investigación y la herramienta resultante se presentan en dos ámbitos: social y tecnológico.

Socialmente ayuda a mejorar la pronunciación mediante la práctica con una entidad virtual sin que ésta pueda verse afectada por cansancio o cualquier otra constante que pudiera interrumpir la actividad de práctica. Así, es posible mantener dicha acción por tiempo indeterminado sin la necesidad de contar con otro ser humano. Así mismo, al momento de elevar la habilidad oral del idioma inglés, es posible obtener mejores oportunidades laborales ya que el mercado laboral lo demanda continuamente.

Por otro lado, tecnológicamente hablando, cuando se trata de crear modelos ASR puede ser un proceso que consume gran cantidad de tiempo, especialmente cuando se utiliza una RNN. En este trabajo de investigación se exploran los beneficios que conllevan implementar una red transformadora como técnica en la elaboración de un ASR. Con este tipo de red el tiempo de entrenamiento se acorta considerablemente debido a que se elimina el proceso de recurrencia lo que ahorra costo computacional. Además, el volumen de datos con los que se alimenta la red es mayor en comparación con las RNN. El modelo ASR en conjunto con un algoritmo de alineación de cadenas logran encontrar errores en la pronunciación en oraciones largas. Otras técnicas pierden rendimiento cuando se analizan secuencias extensas. Adicionalmente, lo anterior se integra en un chatbot para crear una entidad virtual que ayude a mejorar la pronunciación mediante entradas de texto y audio. Hay que tener tres puntos en consideración: 1) se utiliza un chatbot para entablar una conversación escrita y simular una interacción natural con el usuario, 2) se graba un audio de voz de la frase dada como entrada al chatbot y se hace uso del modelo ASR para convertir el audio a texto, 3) se utiliza un algoritmo de alineación de cadenas para encontrar discrepancias entre la entrada escrita dada al chatbot y la transcripción realizada por el modelo ASR.

II. Marco Teórico

Este capítulo abarca las bases teóricas utilizadas durante el desarrollo del proyecto. El *Procesamiento de Lenguaje Natural*, y principalmente el ASR fueron las subdisciplinas de la IA que se implementaron para la realización del proyecto. Así mismo, se efectuó el uso de un chatbot en donde se agruparon las tecnologías antes mencionadas.

2.1 Procesamiento de lenguaje natural

El *Procesamiento de Lenguaje Natural* (NLP, por sus siglas en inglés) es una rama de la IA y la lingüística desarrollada con el propósito de lograr una interacción entre humanos y máquinas, donde estas últimas deben tener la capacidad de entender, interpretar y generar lenguaje humano. Nace bajo el propósito de facilitar al usuario conseguir una comunicación en lenguaje natural. En sus inicios se realizaban tareas simples como la traducción de palabras, posteriormente se encontraron problemáticas que envolvían la forma de estructurar el lenguaje [7].

Un lenguaje se puede definir como un conjunto de símbolos y reglas. Los símbolos son combinados de tal forma que logran convertirse en información. Estos símbolos deben obedecer al conjunto de reglas para que logren tener sentido. Algunos de los campos de investigación relacionados con el NLP son Resumen Automático, Análisis de Discursos, Traducción Automática, Reconocimiento Automático de Voz [8][9][10][11]. El NLP puede clasificarse en dos áreas, el Entendimiento del Lenguaje Natural (NLU, por sus siglas en inglés) y Generación de Lenguaje Natural (NLG, por sus siglas en inglés). El NLU procesa el texto entrante para darle un significado, mientras que el NLG busca comunicar un texto con el sentido apropiado. En otras palabras, NLU procesa entradas y NLG provee salidas.

Hoy en día es común generar grandes cantidades de texto. Ocasionalmente, estos textos podrían contener algún tipo de información útil diferente al de su propósito original. El análisis de texto es uno de los métodos del NLU que ayudan a encontrar posibles elementos que ayuden a esto [12][13][14][15][16]. Para ello se utilizan una serie de pasos con los que se logra realizar el análisis de un texto, en [17] se proponen los siguientes:

a. Análisis léxico: Primeramente, el texto completo es segmentado en palabras, oraciones o párrafos. Cada uno de los segmentos se estructura en un formato que será utilizado en el análisis sintáctico.
b. Análisis sintáctico: Cada elemento es analizado para encontrar la relación que existe con el elemento anterior y posterior.

c. Análisis semántico: Se determinan posibles significados de las palabras en cada segmento basado en la interacción entre ellas.
d. Discurso: El significado de la oración depende de la relación con secuencias previas.
e. Pragmática: Características externas que envuelven aspectos del mundo real. Conocimiento extra que puede cambiar el sentido del texto.

Por otro lado, el NLG se encarga de la generación de frases, sentencias y párrafos. En general, existen tres fases que componen un sistema NLG. Primero, la fase del texto plano cuyo objetivo es seleccionar el contenido apropiado para ser expresado. Se decide la información que deberá ser incluida en el texto generado. Segundo, la fase de planificación de la oración en donde se especifican los límites de la oración y genera orden en los párrafos. Se considera una de las fases más importantes en un sistema NLG debido a que es cuando se da estructura al texto generado dando orden a cada una de las palabras. Finalmente, la fase de realización de la oración en donde se generan párrafos corregidos gramaticalmente y se agrega puntuaciones requeridas.

2.2 Chatbots

Los chatbot son un claro ejemplo de la utilización del NLP y sus dos áreas, el NLU y NLG. Analizan un texto de entrada y con base en ello encuentran una respuesta coherente de salida. Un chatbot es un programa que utiliza IA que pretende simular una conversación al grado de dar la impresión de que se interactúa con un humano [18]. El uso del término chatbot era hasta hace algunos años relacionado a aplicaciones para conversar vía escritura, pero esto cambió con el desarrollo de reconocimiento de voz y su implementación en ellos. El uso del NLP es la principal herramienta para la construcción de los chatbots. En sus inicios, estos eran desarrollados con un funcionamiento básico. Generalmente, contaban con un mecanismo de patrones sencillos y plantillas con respuestas precargadas con el objetivo de simular una conversación [19][20]. Así como existen arquitecturas básicas de chatbots también existen algunas otras más complejas. Hay dos enfoques utilizados al momento de desarrollar un chatbot: coincidencia de patrones y Aprendizaje Automático.

2.2.1 Lenguaje de marcado de Inteligencia Artificial

El *Lenguaje de Marcado de Inteligencia Artificial* (AIML, por sus siglas en inglés) es uno de los métodos de coincidencia de patrones más utilizados y tiene como objetivo describir conocimiento léxico para agentes conversacionales. Tiene una estructura que permite clasificar conocimiento en diferentes categorías [21]. Un chatbot construido bajo esta arquitectura tiene la habilidad de simular una

conversación con un conocimiento precargado. Sin embargo, la interacción depende de la construcción de las etiquetas AIML lo cual podría limitar las entradas y salidas de texto.

Se definen objetos AIML, los cuales son responsables de modelar los patrones de conversación. Cada objeto AIML tiene una etiqueta que corresponde a un comando. Los patrones estructurados en AIML son utilizados como entrada dada por el usuario al chatbot. La forma más básica de construir una etiqueta es:

```
<category>
      <pattern>CUAL ES LA CAPITAL DE MÉXICO</pattern>
            <template>
                  La capital de México es la Ciudad de México
            </template>
</category>
```

En donde, el patrón de texto exacto a utilizar es "`CUAL ES LA CAPITAL DE MÉXICO`" para obtener la respuesta especificada en la etiqueta `<template>.` Al estructurar la etiqueta de esa forma se requiere que la entrada sea estrictamente la especificada en el patrón.

Otra forma de estructurar un patrón es haciendo uso del símbolo "*". Al utilizarlo, no es necesario introducir el patrón exacto. El símbolo acepta cualquier cadena que haya sido introducida en su lugar. Por ejemplo:

```
<category>
      <pattern>COMO PUEDES COMER *</pattern>
            <template>
                  Porque es de mis comidas favoritas
            </template>
</category>
```

2.2.2 Aprendizaje Automático

Los chatbots que son desarrollados utilizando Aprendizaje Automático hacen uso de técnicas de NLP, tanto de NLU y NLG. La gran ventaja de este método es que el chatbot dispone de la habilidad de aprender de las conversaciones. Se considera el contexto con el que se desenvuelve la conversación y no solo la última entrada como el caso de AIML. Típicamente, se necesita un gran conjunto de datos de entrenamiento. Los datos utilizados para el entrenamiento dependerán de la finalidad del chatbot. Las Redes Neuronales sueles ser utilizadas para el entrenamiento de los modelos utilizados por el chatbot [22].

Las RNN se utilizan a menudo en el desarrollo de los chatbots. Este tipo de red toma en cuenta el contexto previo en una conversación, es decir, frases anteriores que se han escrito. Así mismo, las RNN son utilizadas en modelos *Secuencia a Secuencia* en donde se utilizan dos redes de este tipo para crear un *codificador* y *decodificador*. Este tipo de modelo genera secuencias de salida automáticamente considerando la secuencia de entrada. El codificador recibe el texto de entrada y lo procesa de tal forma que su salida será interpretada por el decodificador. De esta forma se puede mantener una conversación con el chatbot.

2.2.3 Chatbots en la educación

Hoy en día es común que los estudiantes reciban tareas en línea. Suelen estar inmersos en un conjunto de herramientas que los acompañan durante el proceso de aprendizaje. El uso de los chatbots puede considerarse una de esas herramientas que ayudan en el área de la educación. Los estudiantes que los utilizan son guiados durante el aprendizaje de algún tema en específico. Se selecciona contenido que estimula al estudiante a seguir utilizándolo [23]. Generalmente la estructura utilizada presenta una serie de ejercicios mostrados en forma de preguntas, en donde el estudiante provee una pregunta que será respondida por el chatbot que además provee retroalimentación [24].

Sin embargo, pueden existir algunas desventajas al momento de utilizar un chatbot para este propósito. Éstas están relacionadas con el método con el cual el chatbot es desarrollado. Al emplear AIML, los temas seleccionados para su construcción serán los únicos que se podrán abordar. Utilizar el método de Aprendizaje Automático, podría no abracar gran conocimiento en un inicio y demoraría en aumentar [25].

En el trabajo de [26] se concluyó que los estudiantes consideran a los chatbots como un servicio atractivo y complaciente. La razón principal se debe a la forma en la que el chatbot presenta la información. El sistema desarrollado recolecta la información de la web y la muestra al usuario sin la necesidad que este deba buscarla por sí mismo.

2.3 Reconocimiento de voz

La voz es la primera forma de comunicación entre las personas. Por esa razón, se han realizado investigaciones que permitan entender la capacidad de la voz humana en un ámbito tecnológico. El reconocimiento de voz es el proceso de convertir lenguaje hablado en texto que la computadora pueda entender. Ha crecido rápidamente en los últimos años debido al beneficio que trae, ya que permite una interacción humano-máquina un tanto más natural [27].

De forma simple, un sistema ASR puede describirse como una serie de muestras X de una señal de audio de voz a la que se le aplica una función f para encontrar la secuencia se palabras W que representen la transcripción de lo que fue dicho. Lo anterior es mostrado en la Ecuación 1.

$$W = f(X) \tag{1}$$

Sin embargo, la función que ayuda a encontrar la transcripción es difícil de realizar. Es necesario crear modelos que puedan producir la dicha secuencia de palabras. Existen una gran variedad de modelos ASR, cada uno de ellos con su respectivo grado de complejidad.

2.4 Modelos de reconocimiento de voz

Un modelo de ASR es el encargado de convertir una señal de audio de voz en una secuencia de palabras. Para ello existen diferentes técnicas, desde modelos estadísticos basados en datos hasta modelos que emplean *Deep Learning*. Todos ellos cumplen con el mismo propósito, la diferencia se concentra en el rendimiento que desempeñan. Los métodos que utilizan *Deep Learning* detectan las palabras con mayor exactitud. Aunque el proceso durante la construcción de un modelo es diferente, todos comparten la tarea de extracción de características del audio.

2.4.1 Características del audio

Un audio como tal no puede ser procesado directamente por una técnica para crear un modelo ASR. Por tal motivo, es necesario convertir la señal de audio de voz en características que podrán ser procesados posteriormente [28]. La técnica de extracción de características más utilizada en la tarea de reconocimiento de voz es conocida como *Mel Frequency Cepstral Coefficients* (MFCC). Estos coeficientes son un representación matemática de los datos de voz [29]. Están basados en el comportamiento del oído humano.

Al audio se le aplica un filtro que elimina altas y bajas frecuencias. Posteriormente, el audio es dividido en segmentos de igual duración. Algunas veces los segmentos se sobreponen para realizar una transición más suave. Después, a cada uno de ellos se le reducen los cambios bruscos en los bordes causados en la segmentación. A continuación, se calcula la transformada rápida de Fourier para extraer la frecuencia de la señal del dominio de tiempo (un espectrograma es la representación visual de esto). Como la señal no sigue un comportamiento lineal, es necesario aplicar otro filtro el cual actuará de forma logarítmica en altas frecuencias y lineal en bajas frecuencias. Finalmente se calcula la transformada discreta del

coseno para obtener como salida los MFCC. Cada uno de los segmentos se convierte en un vector de características [30].

2.4.2 Modelo oculto de Markov

El *Modelo Oculto de Markov* (HMM, por sus siglas en inglés) es una técnica probabilística utilizada para la tarea de reconocimiento de voz. El sistema modelado es procesado con parámatelos ocultos. Se compone de un conjunto de estados los cuales están conectados por transiciones en donde la secuencia de estados está oculta [31].

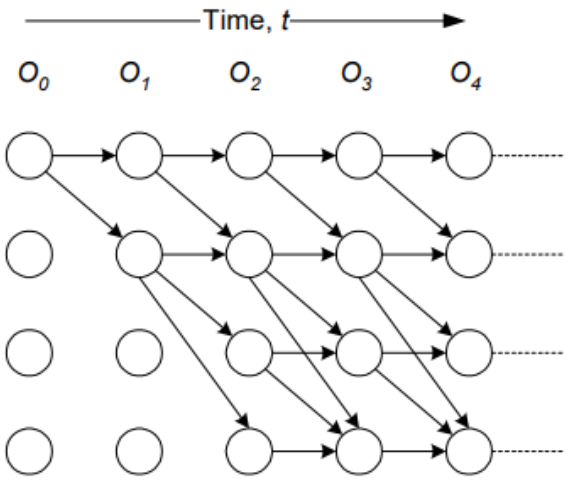
Formalmente, un HMM (Ecuación 2) λ es definido por un conjunto de N estados, M símbolos de observación y tres matrices probabilísticas:

$$\lambda = (\pi, A, B) \tag{2}$$

Donde:
π = Probabilidades del estado inicial
A = Probabilidades del estado de transición
B = Probabilidades de emisión de símbolos

Un HMM puede ser utilizado con unidades lingüísticas como el fonema, palabras o sentencias. El fonema es la unidad lingüística habitualmente más utilizada. Por lo tanto, si se utiliza un conjunto de fonemas bastaría concatenarlos en cierto orden para formar palabras completas. Dicho esto, es posible representar cada estado como un fonema y la transición se realiza en base a probabilidades [32]. En un instante $t + 1$, el modelo visita un estado diferente. Este proceso continua hasta que se cumple el tiempo completo T relacionado con el audio de voz [33]. Cuando el modelo ocupa uno de los estados una observación es emitida.

FIGURA 1. Transición entre estados de un Modelo Oculto de Markov

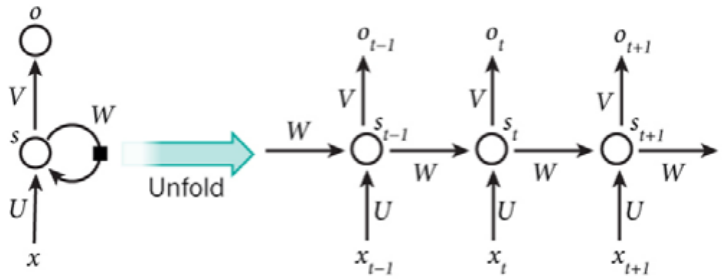


Como se muestra en la Figura 1, cada una de las observaciones es calculada en un periodo de tiempo, en donde cada observación es representada por una unidad de texto. Se puede visualizar todas las posibles rutas entre estados que se pueden tomar [34].

2.4.3 Redes neuronales recurrentes

Con el surgimiento de técnicas de *Deep Learning* y redes neuronales, el rendimiento de los modelos de ASR fue mejorado considerablemente. El HMM tiene una desventaja importante debido a que solo se actualiza de un estado al siguiente, lo que no permite a la red aprender dependencias a largo plazo [35]. Las Redes Neuronales Recurrentes son un tipo de red neuronal que tiene memoria para decidir predicciones futuras. Son llamadas de esa forma debido a que la salida de una neurona se utiliza como una nueva entrada aplicando de esa forma la recurrencia. Por ello, se dice que tiene memoria sobre los datos anteriores. El uso de este tipo de red es apropiado para la tarea de reconocimiento de voz. Esto es debido a que el audio contiene información que no es estática, cada segmento de tiempo contiene diferentes características del audio [36].

FIGURA 2. Red neuronal recurrente.



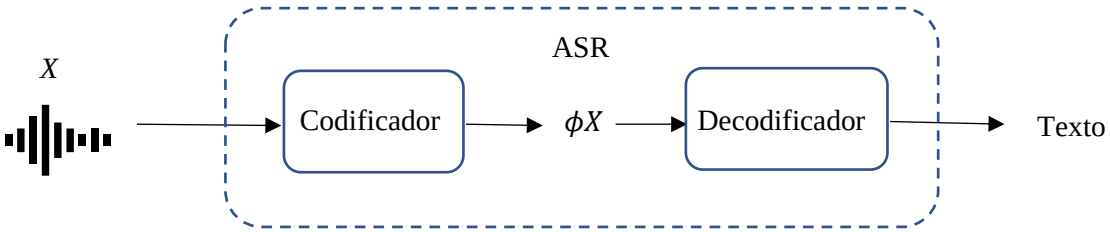
La forma en la que una RNN funciona se muestra en la Figura 2. Un vector X de características del audio alimenta a una neurona recurrente en cada segmento de tiempo. En donde, X_t es una entrada en el tiempo t, X_{t-1} es una entrada anterior y X_{t+1} es una entrada posterior. La neurona tiene dos salidas: S y O. La salida S también va variando conforme el tiempo y esta vuelve a ser una nueva entrada para la neurona. La salida O es la salida en el tiempo t y esta es calculada en conjunto con la memoria almacenada en S [37].

2.4.4 Modelos Extremo a Extremo

Un modelo *Extremo a Extremo* (E2E, por sus siglas en inglés) es un sistema que consigue una secuencia de palabras dada una secuencia de audio de voz. Los modelos ASR E2E se centran en conseguir una transcripción directamente de una señal de audio. La ventaja de utilizar un modelo E2E en comparación con otros antes mencionados es que las transcripciones se realizan con un solo modelo. En un HMM es necesaria la construcción de múltiples modelos por separado [38].

Para la creación de un modelo Extremo a Extremo bastan las características del audio y la transcripción de cada uno de ellos. Las transcripciones pueden ser: caracteres, fonemas o palabras [27]. Entre los diferentes modelos existentes en un enfoque E2E se encuentran los de arquitectura *Codificador* y *Decodificador*. En la Figura 3 se muestra de forma general el funcionamiento de esta arquitectura. El Codificador se alimenta de los vectores de características del audio X y al procesarlos consigue características ricas en contenido lingüístico ϕX. Estas posteriormente pasan al Decodificador para obtener la unidad lingüística utilizada por el modelo.

FIGURA 3. Arquitectura general de un ASR E2E



2.5 Práctica de pronunciación asistida por computadora

Los sistemas de *Práctica de Pronunciación Asistida por Computadora* (CAPT, por sus siglas en inglés) son utilizados para desarrollar la habilidad del habla al momento del aprendizaje de un segundo idioma. Funcionan en conjunto con un sistema ASR que transcribe los audios a texto para posteriormente observar y evaluar la pronunciación del usuario. Si dicha oración contiene algún tipo de error en la

pronunciación, entonces se provee retroalimentación al usuario [39]. Al implementar esta tecnología se facilita el acceso a aprender o mejorar las habilidades de un idioma en el aprendizaje. A comparación de un profesor de idiomas, un sistema CAPT no sufre de cansancio y puede ser utilizado en cualquier momento y cualquier hora del día. Al utilizarlo constantemente se incrementará gradualmente la habilidad además de generar más confianza al momento de hablar.

Los sistemas CAPT se centran en la detección y diagnóstico de errores de pronunciación (MDD, por sus siglas en inglés). En comparación con un sistema ASR, la tarea de MDD presente un mayor grado de dificultad que solo generar texto [40]. Esto debido a que un ASR solo convierte la señal de audio en texto, mientras que la MDD busca encontrar algún tipo de error en la pronunciación en dicha señal de audio.

Normalmente un sistema ASR genera una secuencia de palabras, pero esta tarea podría verse mermada al momento de transcribir una señal de audio de una persona no nativa. Lo anterior se debe a que una persona no nativa no tiene la misma fluidez oral. Por consiguiente, el ASR podría mostrar transcripciones erróneas.

Un sistema CAPT ideal puede ser descrito en cinco fases [41]:

1. Reconocimiento de voz: el sistema ASR convierte la señal de audio de voz en una secuencia de palabras. El sistema ASR puede ser construido con cualquiera de los métodos antes mencionados (HMM, RNN, E2E).
2. Puntuación: se evalúa la calidad de la pronunciación. Los algoritmos *GOP* y de alineación de cadenas pueden ser empleados en esta fase.
3. Detección de error: el sistema localiza los posibles errores en la oración y se los indica al usuario. Si se obtienen puntajes bajos en alguna unidad lingüística, se habrá detectado un error de pronunciación.
4. Diagnóstico de error: el sistema ASR identifica el tipo específico de error realizado por el usuario y sugiere cómo corregirlo.
5. Retroalimentación: se presenta al usuario la información recolectada en los pasos 2, 3 y 4.

III. Trabajos relacionados

En este capítulo se mencionan algunos de los trabajos relacionados con el presente proyecto de investigación. Se presentan diversos trabajos que abordan una problemática similar haciendo uso de técnicas de ASR, así como sistemas de detección de errores de pronunciación. Estos trabajos ayudaron a conseguir un mejor entendimiento del tema y a su vez fueron extraídas las técnicas de mayor utilidad para su utilización en la investigación.

3.1 Revisión de literatura

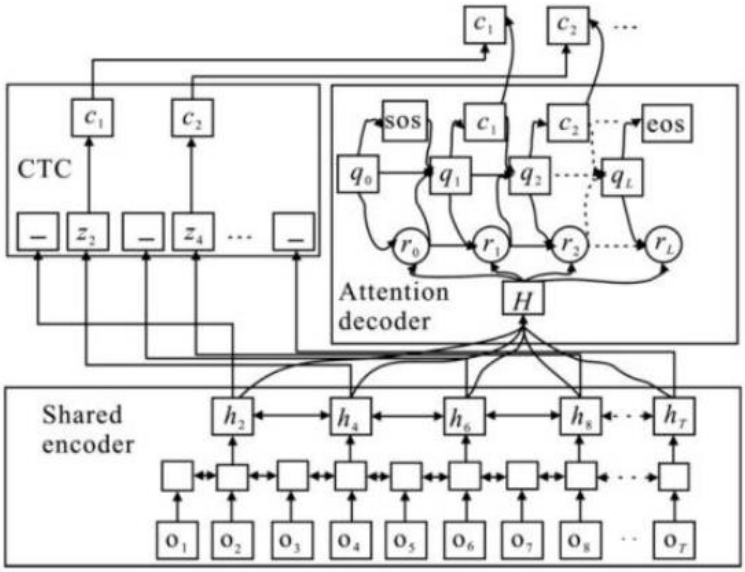
Los sistemas CAPT pueden ser construidos haciendo uso de diferentes técnicas, lo importante es que cumpla con su objetivo, es decir, que logren detectar y diagnosticar un error de pronunciación. Diversas técnicas han sido propuestas para realizar la tarea de MDD. Distintas tecnologías pueden ser utilizadas para la creación de modelos de reconocimiento automático de voz que cumplan el propósito de detectar errores en la pronunciación. Por ejemplo, en [42] se propuso un sistema prototipo para obtener puntajes de la pronunciación de palabras en inglés. Su finalidad principal fue utilizarlo en estudiantes chinos de nivel básico debido al escaso personal de profesores foráneos o la educación desbalanceada entre diferentes regiones. El sistema fue desarrollado utilizando un algoritmo llamado *GOP* [43]. El algoritmo obtiene puntajes de cada uno de los fonemas detectados en la oración y los clasifica aplicando un umbral, los fonemas por debajo del umbral son los mal pronunciados. Otro sistema de práctica de pronunciación que utilizó el *GOP* fue desarrollado en [44]. Se implementó en exámenes de inglés que son requeridos en algunas escuelas chinas. La motivación para utilizar un sistema de este tipo fue porque se dificultaba reclutar expertos calificados para evaluar los exámenes. Además, también se consideró el tiempo que esto implica por lo cual una solución fue implantar un sistema como este.

Es posible crear un sistema ASR E2E para posteriormente utilizarlo en la tarea MDD. En [45] se implementaron Redes Neuronales Convolucionales (CNN, por sus siglas en inglés) en conjunto con RNN's para crear un modelo de reconocimiento de voz Extremo a Extremo. Se utilizó una evaluación de la pronunciación haciendo uso de un algoritmo de alineación de caracteres [46]. Para que la tarea MDD sea realizada debe de considerarse la señal de audio de voz así como la transcripción escrita de dicho audio. Haciendo uso de un algoritmo de alineación de cadenas es posible encontrar las diferencias entre la transcripción conocida y la transcripción realizada de la señal de audio por el sistema ASR. Si cada par de dicha alineación contiene una unidad lingüística (carácter o fonema) no idéntica, significa que un error de pronunciación ha ocurrido.

Dos técnicas comúnmente utilizadas en la creación de un modelo ASR E2E son: la Clasificación Temporal Conexionista (CTC) y el Modelo Secuencia a Secuencia Basado en Atención. Ambos modelos son empleados en conjunto en [47] para crear un modelo ASR que posteriormente se utilizaría en la tarea MDD. El método CTC utiliza una secuencia de caracteres para representar una posible secuencia de palabras. Dentro de todo el conjunto de caracteres se utiliza también la etiqueta en blanco, que funge como un delimitador entre los caracteres al momento de formar varias palabras. Este método busca optimizar la predicción al momento de realizar la secuencia de transcripción. Mientras que el modelo basado en Atención modela la probabilidad de secuencia de salida directamente.

En el proceso de Codificación, la secuencia de vectores $\{O_t \dots O_T\}$ es convertida en una secuencia de características $H = \{h_t \dots h_T\}$. Después en la codificación realizada por la Atención se genera una secuencia de caracteres $\{C_1 \dots C_l\}$. Las etiquetas $< sos >$ y $< eos >$ son utilizadas para representar el inicio y el final de la secuencia, respectivamente. El objetivo de la función CTC en este caso es actuar como un auxiliar. El proceso de lo mencionado es posible visualizarlo en la Figura 4.

FIGURA 4. Arquitectura conjunta de CTC-Atención



Otra de las estructuras E2E utilizadas para el desarrollo de un modelo ASR es la llamada red Transformadora. Este tipo de red se compone de igual forma de un Codificador y un Decodificador. Consigue procesar los datos de entrenamiento durante la creación del modelo en menor tiempo [48]. La ventaja de implementar este tipo de red es el tiempo de computación que emplea y su rendimiento similar o mejor al de las RNN. Este tipo de red también ha sido implementada en tareas de detección de errores de pronunciación. En [49] se utilizó una red Transformadora para la creación de un modelo de reconocimiento de voz que puede detectar errores en los fonemas dado un audio de voz y los fonemas

objetivo. Para un mayor rendimiento al momento de realizar la tarea de MDD, fueron utilizados dos tipos de conjuntos de datos. El primero consta de señales de audio de voz producidos por personas hablando su lengua nativa, en este caso, el inglés. El segundo está formado por personas cuya lengua nativa no es inglés pero lo emplean al momento de producir los audios. Esto con la finalidad de mostrar cómo afecta la variación de los sonidos al pronunciar un fonema en el momento en que se aprende un nuevo lenguaje.

Cualquiera de las técnicas mencionadas puede ser utilizada para el desarrollo de un sistema CAPT. Entre uno de los objetivos de estos sistemas se encuentra el de otorgar al usuario una correcta retroalimentación sobre los errores de pronunciación. Varios sistemas, como los mencionados en [50], proveen diversas formas de emitir la retroalimentación. Algunos presentan el espectrograma de la que el usuario pronunció, mientras que otros muestran de forma visual cómo debe ser pronunciada alguna palabra en la cual se detectó un error. Sin embargo, escasamente cuentan con una entidad virtual con la que se logre mantener una conversación.

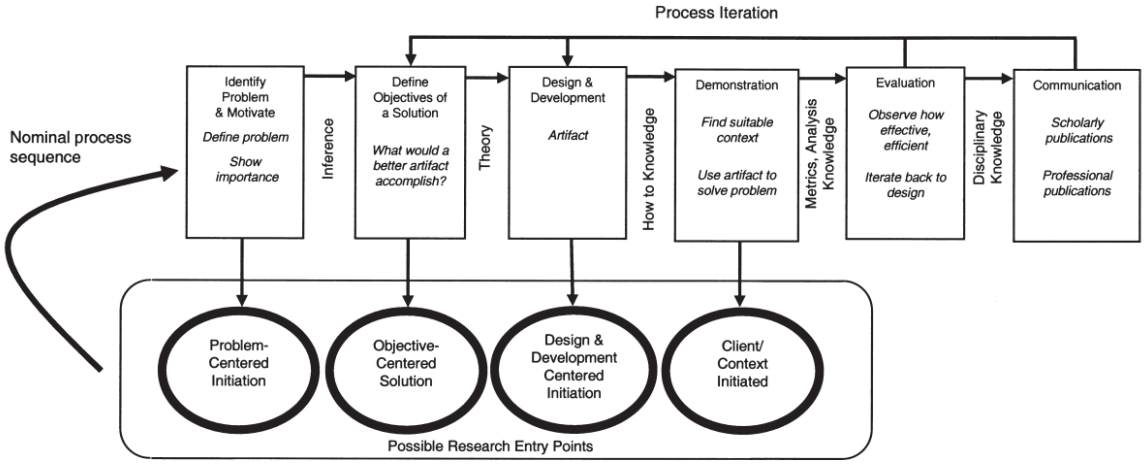
IV. Propuesta de solución

En el presente capítulo es descrita la metodología que se empleó para el desarrollo del proyecto, así como las actividades desarrolladas en cada fase. Se detalla la técnica de modelación del sistema ASR desarrollado, así como todos los factores requeridos para lograrlo. Se explica el algoritmo de alineación de cadenas que ayudó al proceso de detección de errores en la pronunciación. Finalmente, se menciona cómo lo anterior es integrado en la arquitectura de un chatbot.

4.1 Metodología

Se utilizó la *Metodología de Investigación en Ciencias del Diseño* (DCRM, por sus siglas en inglés) [51], la cual consta de seis fases: 1) Identificación del problema y motivación, 2) Definición de los objetivos, 3) Diseño y desarrollo, 4) Demostración, 5) Evaluación y 6) Comunicación. En la Figura 5 es posible visualizar las fases mencionadas.

FIGURA 5. Metodología de Investigación en Ciencias del Diseño.



La fase de *identificación del problema* consiste en definir el asunto especifico que busca ser solucionado, conceptualizarlo ayuda a captar toda su complejidad. Posteriormente, se definen *objetivos* relacionados al problema los cuales deben cumplirse para dar solución a la necesidad. En la fase de *diseño y desarrollo* se crea el artefacto con el cual se busca resolver la problemática. Este puede ser un modelo, un método o una instancia. Los recursos necesarios para pasar de los objetivos al desarrollo del artefacto incluyen el conocimiento de la teoría relacionada a la solución propuesta. Una vez que el artefacto ha sido desarrollado, se procede a *demostrarlo* utilizándolo en varias instancias y observar su comportamiento. Es necesario tener conocimiento sobre el funcionamiento del artefacto aplicado al problema. Durante la *evaluación* se observa y mide el desempeño del artefacto sobre el problema y

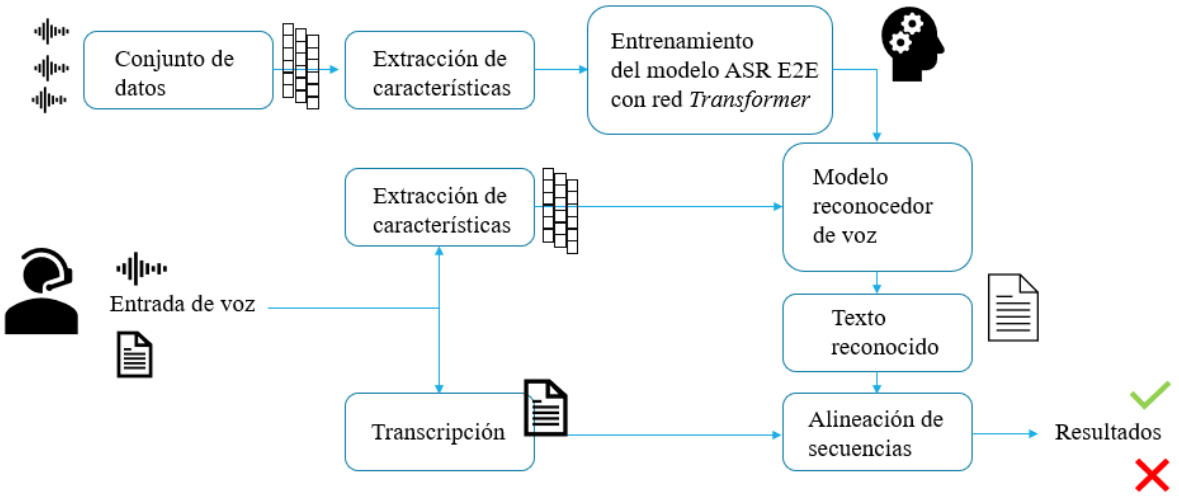
cómo ayuda a solucionarlo. Para ello, es necesario emplear métricas y técnicas de análisis relacionadas a la solución. Al final de esta actividad es posible considerarse regresar a la fase 3 para tratar de mejorar la eficiencia del artefacto. Finalmente, se *comunica* el problema, su importancia, el artefacto, su utilidad y todos los aspectos relevantes encontrados durante la investigación.

Aplicando las fases de la metodología previamente descritas, se logró desarrollar un artefacto que en parte ayuda en la problemática de cometer errores en la pronunciación del idioma inglés. Relacionando la metodología con lo elaborado se realizó lo siguiente:

- Fase 1: La identificación del problema es posible encontrarla en la Sección 1.2.
- Fase 2: Los objetivos son descritos en la Sección 1.3.
- Fase 3: Se creó un sistema para la detección de errores de pronunciación haciendo uso de un sistema ASR E2E empleando una red Transformadora, un algoritmo de alineación de cadenas y un chatbot construido en AIML. Esto es descrito en la sección 4.2.
- Fase 4: Se probó el sistema mientras se desarrollaba hasta conseguir el comportamiento deseado. El proceso se describe en la sección 5.1.
- Fase 5: El ASR se evaluó con una métrica comúnmente utilizada en sistemas de reconocimiento de voz llamada *Tasa de Error de Caracter* (CER, por sus siglas en inglés). Dicha evaluación se presenta en la sección 5.2.
- Fase 6: Se contempla una publicación en una revista de divulgación científica.

La arquitectura de la propuesta de solución se muestra en la Figura 6. Primeramente fue necesario recolectar el conjunto de datos para el entrenamiento del modelo ASR, en este caso señales de audio de voz. Posteriormente, se extraen las características del audio de cada uno de ellos. Estas características del audio son los datos de entrada con los cuales se alimenta la red Transformadora. Una vez finalizado el proceso de entrenamiento, el resultado es el modelo ASR E2E. Con el reconocedor de voz ya en funcionamiento, lo siguiente es hacer uso de él. Para ello el usuario pronuncia un oración, la cual, su transcripción es previamente introducida en el chatbot. El audio por evaluar pasa de nuevo por la extracción de características para posteriormente ser utilizado por el modelo entrenado. Con el algoritmo de alineación de cadenas se compara el texto reconocido por el modelo y la transcripción introducida por el usuario. Si existe alguna diferencia entre los dos textos la interfaz del chatbot indica al usuario que ocurrió en error de pronunciación.

FIGURA 6. Arquitectura del reconocedor de voz para la tarea de detección de errores de pronunciación.



4.2 Modelación del sistema ASR

La modelación constó de una serie de pasos que comprendió desde la recolección de datos para su entrenamiento, hasta la utilización de la técnica para su desarrollo. Se utilizó una herramienta llamada *ESPnet* [52], la cual cuenta con un conjunto de técnicas para el desarrollo del modelo. La red transformadora fue la elegida debido a su rendimiento en comparación con otras técnicas para crear el sistema ASR que ayudó posteriormente a la detección de errores de pronunciación.

4.2.1 Conjunto de datos

El lenguaje nativo o lengua materna (L1) comparte ciertas similitudes con el lenguaje objetivo (L2) durante el proceso de aprendizaje de un segundo idioma. La forma de asimilar los fonemas del L2 está relacionada con los conocimientos obtenidos previamente del L1. Sin embargo, durante el aprendizaje de un segundo idioma es común producir errores de pronunciación debido a inserciones, eliminaciones o modificaciones de fonemas [53]. El modelo de aprendizaje descrito en [54] menciona la relación existente entre el L1 y L2 al agregar o modificar unidades fonéticas durante el proceso de aprendizaje. Debido a lo anterior, se utilizaron dos tipos de conjuntos de datos. El primero con audios en inglés pronunciado por personas nativas y el segundo con audios en inglés pero pronunciado por personas no nativas. Ambos conjuntos de datos se unieron en uno solo para utilizarlos en el entrenamiento. Además, cada uno de los audios cuenta con su correspondiente transcripción. Es común emplear estos dos tipos de conjuntos de datos para crear sistemas de detección de errores de pronunciación [47].

El conjunto de datos de *LibriSpeech* fue el seleccionado como aquel que cuenta solo con audios en inglés producidos por personas nativas. El conjunto completo cuenta con 1,000 horas de muestras de

audios a 16 kHz [55], de los cuales 960 pertenecen al conjunto de entrenamiento y el resto al conjunto de prueba. Por otra parte, el otro conjunto que cuenta con audios en inglés producidos por personas no nativas fue *L2-ARTIC.* Este es frecuentemente utilizado para la creación de sistemas de detección de errores de pronunciación. Cuenta con diversos acentos como el hindú, mandarín, árabe, español, entre otros. Para fines de este proyecto fue utilizado solo el conjunto con el acento en español debido a que el uso del sistema está pensado para usuarios cuyo idioma nativo es español. Utilizar un acento diferente podría comprometer la exactitud del reconocedor de voz al momento de inferir audios en español.

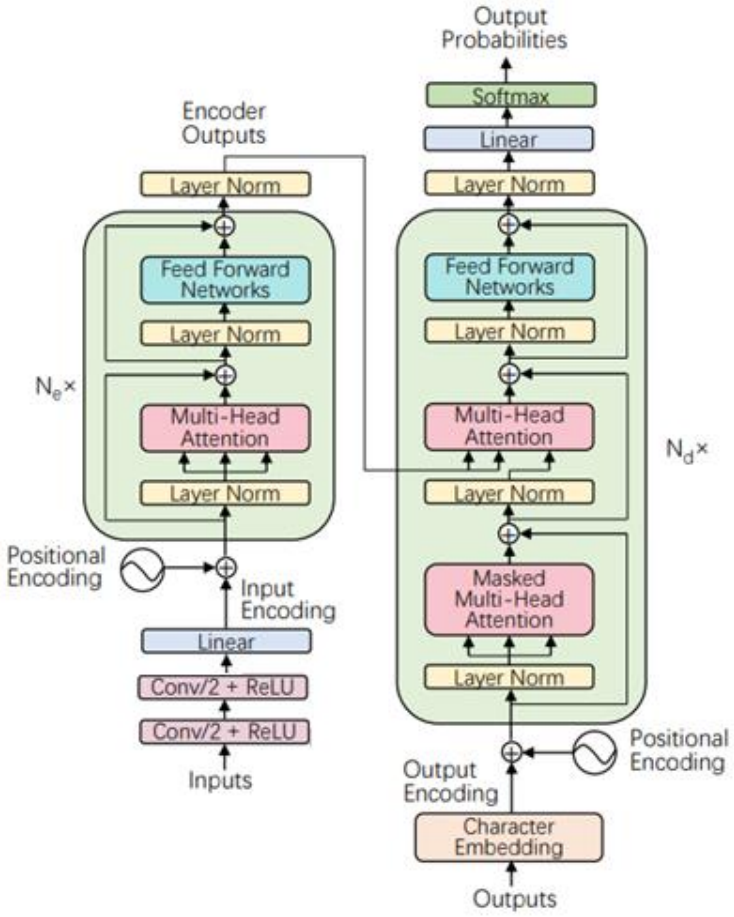
4.2.2 Herramienta ESPnet y red Transformadora

La herramienta *ESPnet* es comúnmente utilizada para la creación de sistemas ASR E2E. Contiene varios conjuntos de datos a su disposición. Así mismo, es posible utilizar diferentes técnicas de modelación para la creación del reconocedor de voz. [52]. Entre las técnicas disponibles para la creación del modelo se encuentran tres tipos de redes neuronales las cuales son: 1) una red convolucional llamada VGG, 2) una red de memoria bidireccional a corto plazo o 3) una red Transformadora. Esta última fue la seleccionada para la creación del modelo utilizado en este proyecto. Esta herramienta también cuenta con una sección de preparación de datos en donde se extraen las características del audio previamente.

La red transformadora fue inicialmente propuesta para la tarea de traducción automática y otros problemas de procesamiento de lenguaje natural [56]. Sin embargo, también es posible aplicarlo a sistemas ASR. Fue propuesto por primera vez para esta tarea en el trabajo desarrollado en [57], llamando a la red *Speech-Transformer*. La estructura de la red se muestra en la Figura 7. Como todos los modelos ASR E2E, su objetivo es transformar una secuencia de características del audio en su correspondiente secuencia de caracteres. Habitualmente se suelen utilizar RNN para la creación de modelos ASR E2E. Desafortunadamente, la naturaleza secuencial de este tipo de red limita la paralelización computacional debido al proceso de recurrencia durante el entrenamiento, lo que podría traducirse a un mayor consumo de tiempo durante este proceso.

La red cuenta con dos partes: un codificador y un decodificador. Cada una de ellas desempeñan sus propios procesos, aunque ambas trabajan en conjunto de inicio a fin. El codificador transforma las secuencias de características del audio $(x_1, \dots, x_T)$ en una representación oculta $\boldsymbol{h} = (h_1, \dots, h_L)$. Después, se suministra $\boldsymbol{h}$ al decodificador para generar una secuencia de salida $(y_1, \dots, y_S)$ caracter por caracter [56]. En cada secuencia de tiempo, el decodificador consume el caracter previo como entrada adicional para emitir el próximo caracter. Ambos, codificador y decodificador, se componen bloques de *Multi-Head Attention,* los cuales calculan las probabilidades de relación que cada segmento de audio tiene con los demás.

FIGURA 7. Arquitectura del modelo Speech-Transformer.



Cada bloque (codificador y decodificador) cuenta con algunos procedimientos y subbloques, los cuales son:

a) *Scaled Dot-Product Attention*

El mecanismo de autoatención relaciona diferentes posiciones de una secuencia de entrada para calcular sus representaciones. Ayuda a entender cómo se relacionan las secuencias unas con otras. Para ello se calcula el *Scaled Dot-Product Attention*, el cual tiene como finalidad encontrar las probabilidades entre cada una de las secuencias. Para ello son necesarias tres entradas, las cuales son: consultas, llaves de dimensión d_k y valores con dimensión d_v. Como se muestra en la mitad izquierda de la Figura 8, se calcula el producto punto entre las consultas y las llaves. Esto da como resultado una matriz que determina el grado de relación que existe entre cada uno de los segmentos. A continuación, los elementos de la matriz son divididos entre $\sqrt{d_k}$ con el objetivo de que la función *softmax* (la cual se utiliza en el siguiente paso) evite las regiones que tienen gradientes muy pequeños. Finalmente, la matriz resultante se multiplica por la matriz de valores. La salida de este proceso se calcula como:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (3)$$

En donde:

- $Q \in \mathbb{R}^{t_q \, x \, d_q}$ son las consultas
- $K \in \mathbb{R}^{t_k \, x \, d_k}$ son las llaves
- $V \in \mathbb{R}^{t_v \, x \, d_v}$ son los valores

b) Multi-Head Attention

Este proceso calcula h la atención de productos escalados. Antes de efectuarlo, hay tres capas lineales que transforman las consultas, las llaves y los valores en representaciones más discriminativas. Estas capas son diferentes para cada una de las cabezas y cuenta con sus propios parámetros. De esta forma, cada una de las cabezas calcula una serie de valores diferentes. Las salidas de las h cabezas se concatenan y se alimenta en otra capa lineal para finalmente obtener una salida de dimensión d_{model}. Este proceso se muestra en la mitad derecha de la Figura 8.

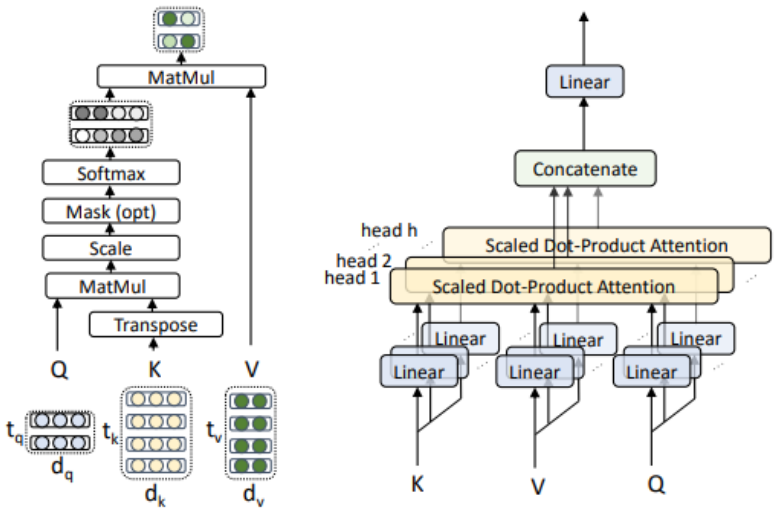
$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \tag{4}$$

$$\text{donde } head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{5}$$

En donde:

- $Q, K \; y \; V$ tienen dimensión d_{model}
- $W_i^Q \in \mathbb{R}^{d_{model} \, x \, d_q}$, $W_i^K \in \mathbb{R}^{d_{model} \, x \, d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \, x \, d_v}$ y $W_i^O \in \mathbb{R}^{hd_v \, x \, d_{model}}$ son matrices de parámetros.

FIGURA 8. (A) Atención de productos escalados. (B) Multi Cabezas de Atención.



c) Feed Forwards Networks

Una red neuronal prealimentada es otra parte importante tanto en el codificador como en el decodificador. Se compone de dos capas lineales con una activación ReLU en medio de ambas. La dimensión de la entrada y la salida es de d_{model}, mientras que la capa interior tiene dimensión d_{ff}. Se define como:

$$FFN(x) = \max(0, xW_1 + b_1)\, W_2 + b_2 \tag{6}$$

En donde:

- $W_1 \in \mathbb{R}^{d_{model}\, x\, d_{ff}}$ y $W_2 \in \mathbb{R}^{d_{ff}\, x\, d_{model}}$ son matrices de parámetros de pesos.
- $b_1 \in \mathbb{R}^{d_{ff}}$ y $b_2 \in \mathbb{R}^{d_{model}}$ son parámetros de sesgo.

El codificador (mitad izquierda de la Figura 7) recibe como entrada los vectores de características de la señal de audio. Estas mismas características pueden ser representadas como espectrogramas con ejes de tiempo y frecuencia. Estas secuencias de características comúnmente son más largas que las secuencias de caracteres. Por lo cual, es posible utilizar redes convolucionales para mitigar el desajuste de longitud entre ambas secuencias. Lo anterior se logra mediante la técnica de Detección de Eventos Acústicos (AED, por sus siglas en inglés) la cual detecta y clasifica eventos acústicos donde hay ausencia de voz [58]. La salida de lo anterior es un vector de características aplanado al cual se le realiza una transformación lineal para obtener los vectores de dimensión d_{model}.

Al no existir recurrencia en el proceso de la red Transformadora como el utilizado en una RNN, es necesario agregar información sobre las posiciones de los segmentos de entrada. Para ellos se emplea una codificación posicional a los vectores de entrada antes de que alimenten al bloque del codificador. Para ellos se emplea los siguiente:

$$PE(pos, 2i + 1) = \cos\left(pos \frac{pos}{10000^{2i/d_{model}}}\right) \tag{7}$$

$$PE(pos, 2i) = \text{sen}\left(pos \frac{pos}{10000^{2i/d_{model}}}\right) \tag{8}$$

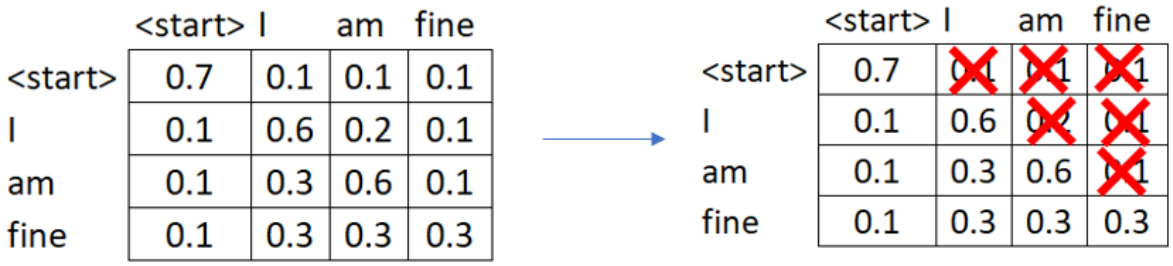
En donde:

- pos representa la posición en la secuencia.
- i representa la dimensión.

Para cada momento impar en la secuencia, se crea un vector haciendo uso de la Ecuación 7. Mientras que para cada momento par en la secuencia, se crea un vector utilizando la Ecuación 8. Debido a que ambos vectores cuentan con una dimensión d_{model}, es posible sumar los vectores de entrada y los vectores posicionales, lo cual es la entrada al codificador. Los vectores resultantes entraran a los N_c bloques del codificador. Posteriormente pasan por los subbloques antes mencionados: MHA y la red neuronal prealimentada. Redes de normalización y de conexión residual son agregadas en cada subbloque para un entrenamiento efectivo.

El decodificador (mitad derecha de la Figura 7) recibe de entrada caracteres en su forma representativa de vectores con una dimensión d_{model}. A estos vectores también se les suma su respectivo vector posicional. Después, estos vectores son alimentados a N_d bloques para obtener las salidas finales. A diferencia del bloque del codificador, el decodificador cuenta con dos subbloques de MHA. El primero, de igual forma recibe como entrada consultas, llaves y valores. La diferencia es que utiliza un proceso extra el cual asegura que las predicciones para la posición j solo puedan depender de las salidas conocidas en posiciones menores que j. Esto evita que en la matriz de atención creada al calcular el producto punto entre las consultas y las llaves, no se tome en cuenta la probabilidad de un segmento con el mismo (Figura 9). La segunda MHA se alimenta de llaves y valores de la salida del codificador y consultas de la salida del subbloque previo. La salida pasa por el último subbloque, otra red neuronal prealimentada. Al igual que en codificador, capas de normalización y conexión residual son agregadas entre los subbloques. Finalmente, las salidas del decodificador son transformadas a probabilidades de clase por una capa lineal y una función softmax. Estas salidas vuelven a ser utilizadas como entrada al decodificador para repetir el proceso y así calcular la probabilidad del siguiente carácter.

FIGURA 9. Matriz atención con probabilidades eliminadas.



4.3 Alineación de secuencias

La alineación de secuencias permite conocer qué tan diferente es una cadena de texto de otra. Esto beneficia en la búsqueda de errores en la pronunciación debido a que se compara el texto inferido por el reconocedor de voz y su respectiva transcripción. Si alguna palabra entre ambas cadenas no coincide, entonces un error en la pronunciación ha ocurrido.

4.3.1 Tasa de error de carácter

La tasa de error de carácter (CER, por sus siglas en inglés) es una métrica utilizada para la evaluación del rendimiento de un reconocedor de voz a nivel carácter. Así mismo, se emplea para conocer un posible error en la pronunciación [47]. Se calcula con la siguiente ecuación:

$$CER = \frac{S+D+I}{N} \tag{9}$$

en donde N es el número total de caracteres, S, D e I corresponden a la cantidad de caracteres que fueron sustituidos, eliminados e insertados, respectivamente. Los valores correspondientes a S, D e I se calculan empleando el algoritmo de la distancia de Levenshtein, el cual compara la cadena de texto obtenida del reconocedor de voz con la cadena de la transcripción canónica.

4.3.2 Distancia de Levenshtein

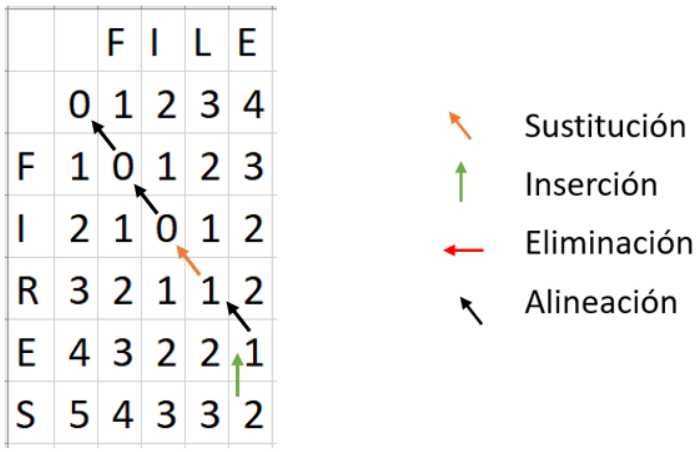
La distancia de Levenshtein es una medida de similitud entre dos cadenas, la cadena original (s) y le cadena objetivo (t). La distancia es el número de eliminaciones, inserciones o sustituciones requeridas para transformar *s* en *t*. Entre más grande es el valor de la distancia, mayor es la diferencia entre ambas cadenas [59]. En el Algoritmo 1 se describe el funcionamiento.

ALGORITMO 1. Distancia de Levenshtein.

```
Entrada: s: cadena original; t: cadena objetivo
Salida:   Matriz[n,m]
//la variable n toma el valor del tamaño de la cadena s
n ← tamaño(s)
//la variable m toma el valor del tamaño de la cadena t
m ← tamaño(t)
//se crea una matriz de dimensión nxm
Matriz[n,m]
//se asignan los valores de 0 a n en las celdas Matriz[i,0]
para i ← 1 hasta n
   Matriz[i,0] ← i
//se asignan los valores de 0 a m en las celdas Matriz[0,i]
para i ← 1 hasta m
   Matriz[0,i] ← i
//en un ciclo para anidado se asigna a Matriz[n,m] el valor mínimo de
//Matriz[i-1,j] + 1, Matriz[1,j-1] + 1 o Matriz[i-1,j-1] + costo
para i ← 1 hasta n
   para j ← 1 hasta m
      //el costo toma el valor de 0 si n y m son diferentes, de
      //lo contrario será 1
      costo ← 0
      si n == m
         costo ← 1
      Matriz[n,m]← min(  Matriz[i-1,j]+1,  Matriz[1,j-1]+1,  Matriz[i-1,j-
1]+costo )
return Matriz[n,m]
```

Como resultado se obtiene una matriz de puntuación de dimensión $n\ x\ m$ la cual realiza un seguimiento del número de ediciones. El valor obtenido en esa posición denota la distancia obtenida entre ambas cadenas. Además, indica la cantidad de movimientos necesarios para transformar *s* en *o*. Para conocer los movimientos realizados se traza un recorrido desde la última celda, hasta la primera posición de la matriz, en donde cada traslado se realiza a la celda mínima entre $Matriz[i-1,\ j]$, $Matriz[1,\ j-1]$ y $Matriz[i-1,\ j-1]$. En la Figura 10 se ejemplifica el uso de la distancia de Levenshtein para conocer los movimientos necesarios para transformar la palabra *file* en *fires*.

FIGURA 10. Distancia de Levenshtein ejemplificada.



4.3.3 Errores de pronunciación

Mejorar la pronunciación al momento de aprender un segundo idioma es una de las tareas que mayor dificultad presenta durante el proceso de aprendizaje. Debido al conocimiento sobre el idioma nativo, se suelen presentar dificultades al reproducir los sonidos fonéticos del idioma objetivo. Por tal motivo, la alteración de estos produce una diferencia a la pronunciación canónica lo que en resumidas cuentas es un error en la pronunciación [60].

La alineación de secuencias es utilizada para encontrar posibles diferencias entre el texto reconocido por el modelo ASR y la transcripción dada. Lo anterior se puede interpretar como la diferencia existente entre la pronunciación del usuario y la pronunciación canónica. En ambos casos una diferencia denotaría un error en la pronunciación. En la Tabla 1 es posible visualizar un ejemplo de dos secuencias en donde la primera es la transcripción y la segunda es el texto reconocido por el modelo ASR. Aplicando el algoritmo de alineación de secuencias se encuentran las diferencias entre ambas cadenas. Las letras sombreadas en amarillo denotan la diferencia existente entre ambas secuencias. Esto a su vez denota un error de pronunciación en la palabra “are” ya que difiere de la inferida por el modelo ASR, la cual es “add”.

TABLA 1. Ejemplo de secuencias con diferencia entre ellas.

H	O	W		A	R	E		Y	O	U
H	O	W		A	D	D		Y	O	U

Los sistemas de detección de errores de pronunciación pueden ser implementados en: 1) nivel segmental y 2) nivel suprasegmental. Los primeros se enfocan en la detección de los errores a nivel fonema o palabra, mientras que los segundos se concentran en la acentuación léxica, acento tonal y entonación [61]. Para fines de este proyecto se utilizó la implementación a nivel segmental enfocado a palabra debido a que el nivel suprasegmental requiere una mayor preparación de los datos de texto (transcripciones) utilizadas para el entrenamiento del modelo ASR.

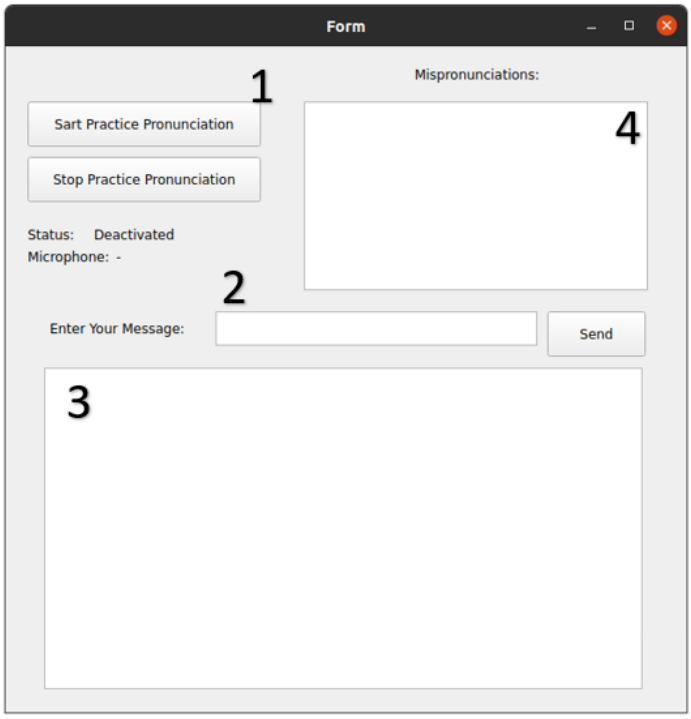
4.4 Chatbot e interfaz gráfica

La función del chatbot en el sistema de detección de errores de pronunciación es la de crear mayor interacción con el usuario al sostener un dialogo con éste. Se decidió utilizar el lenguaje AIML (Sección 2.2.1) debido a su simplicidad ya que el punto fuerte de la investigación se concentró en la creación del modelo de reconocimiento de voz para la tarea de detección de errores de pronunciación. Hay una variedad de temas distribuidos en diferentes archivos AIML los cuales se cargan al sistema uno a uno para posteriormente poder interactuar con su contenido. El usuario introduce un texto que es interpretado por el sistema para encontrar el patrón adecuado y mostrarlo al usuario, de esta forma se va entablando un dialogo.

Cuando el chatbot muestra al usuario su respuesta, a su vez se muestran las palabras en donde ocurrió un error de pronunciación. Una interfaz gráfica ayuda al usuario en este proceso de una manera fluida e intuitiva. La interfaz gráfica (Figura 11) cuenta con:

1. Activación o desactivación de la práctica de pronunciación: Si la práctica de pronunciación esta activa cada vez que el chatbot de una respuesta, a su vez, las palabras mal pronunciadas podrán ser visualizadas, de lo contrario solo se interactuará con el chatbot.
2. Campo de texto para ingresar los mensajes al chatbot.
3. Desarrollo de la conversación con el chatbot.
4. Ventana donde se muestran las palabras mal pronunciadas.

FIGURA 11. Interfaz gráfica del sistema de detección de errores de pronunciación.



Inicialmente la práctica de pronunciación en la interfaz del chatbot se encuentra deshabilitada. Al ser así, el usuario solo mantiene una conversación textual con él. Por otro lado, si la práctica de pronunciación se habilita el usuario debe introducir la transcripción de la oración y posteriormente pronunciarla. A continuación, el modelo ASR realiza la conversión de audio a texto y su salida se compara con la transcripción previamente introducida con la finalidad de encontrar errores en la pronunciación. A su vez, se busca una etiqueta AIML que coincida con la transcripción y de esta forma proveer una respuesta del chatbot hacia el usuario. La salida del chatbot se compone de los posibles errores de pronunciación y la respuesta de la conversación.

V. Resultados y evaluación

En esta sección se presentan los resultados del proceso de evaluación del módulo de detección de errores de pronunciación del idioma en inglés. En la Sección 5.1 se presenta la evaluación de los modelos ASR desarrollados empleando las métricas comúnmente utilizadas en la literatura. La Sección 5.2 presenta los resultados de evaluar el desempeño del módulo de detección de errores con usuarios reales. Por último, en la Sección 5.3 se discuten los resultados obtenidos.

5.1 Evaluación de los modelos ASR

Se desarrollaron y evaluaron cinco modelos ASR. En primera instancia se pensó utilizar el conjunto de audios llamado *Common Voice* de la empresa *Mozilla* [62]. Contiene 2,500 horas de audio en el idioma inglés, recolectados a través de una página web. Debido a que esta página permite a cualquier persona aportar su voz, los audios podrían ser grabados por personas no nativas del lenguaje, por lo que pueden contener múltiples acentos. Cabe mencionar que este trabajo de investigación se centra en personas cuyo idioma nativo es español de México que buscan mejorar la pronunciación del inglés americano. Por tal motivo, solo se utilizó cierto número de horas de este conjunto para realizar pruebas y conocer el funcionamiento del *toolkit* ESPnet antes de proceder al entrenamiento del modelo final. Fueron cuatro los modelos realizados con este conjunto de datos.

Posteriormente, se identificaron los conjuntos de datos de audios *LibriSpeech* [55] y *L2-ARCTIC* [63]. *LibriSpeech* contiene 1,000 horas de audios grabadas por personas cuyo idioma nativo es el inglés. Está conformado por dos conjuntos, entrenamiento y prueba. 960 horas pertenecen al conjunto de entrenamiento y el resto al conjunto de prueba. *L2-ARCTIC* contiene 11.2 horas de audio grabadas por personas de múltiples acentos, incluido el español. De este conjunto de audios se tomaron 4 horas de audios de personas con acento español mexicano y se integraron en los conjuntos de entrenamiento y prueba de *LibriSpeech*, conformando de esta manera el conjunto de audios utilizado en la evaluación del modelo final. La métrica para evaluar el desempeño fue la métrica CER, descrita en la Sección 4.3.1.

Para el desarrollo de los modelos se tomó como base la configuración de la red transformadora mencionada en [57], la cual es empleada para grandes cantidades de datos. El cual está definido por 12 bloques codificadores y 6 bloques decodificadores. Cada modelo se desarrolló variando la cantidad de audios y el número de épocas. Todos los modelos fueron entrenados con la ayuda de una Unidad de Procesamiento de Gráficos (GPU, por sus siglas en inglés) con una memoria de 8 GB.

El primer modelo se entrenó con la cantidad aproximada de 194 horas de audio y 20 épocas. Al finalizar el proceso de entrenamiento, el porcentaje de exactitud no superaba el 8.5% (Figura 12). En la gráfica se observa que en algunas épocas la exactitud del conjunto de entrenamiento tiende a caer, esto probablemente debido a que la cantidad de datos de audio utilizados para entrenar el modelo no fueron suficientes y al realizar la evaluación los porcentajes de exactitud mostraron resultados variados. El porcentaje de la métrica CER superó el 77% lo cual indica que la comparación entre el texto canónico y el texto inferido fueron altamente diferentes (Figura 13).

FIGURA 12. Exactitud del modelo de 194 horas y 20 épocas.

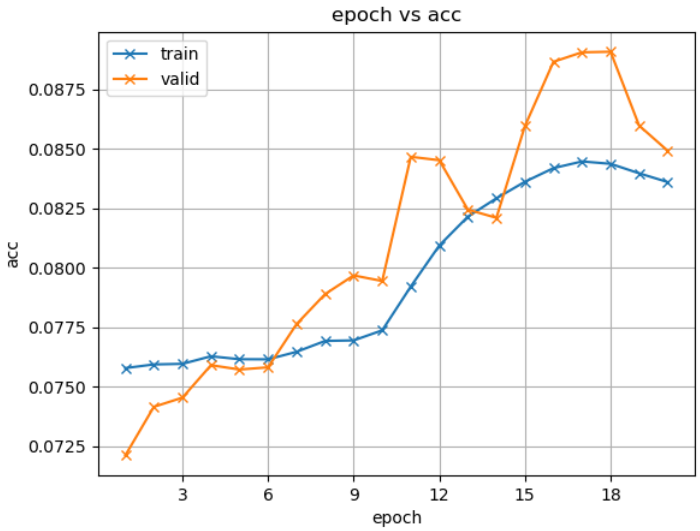
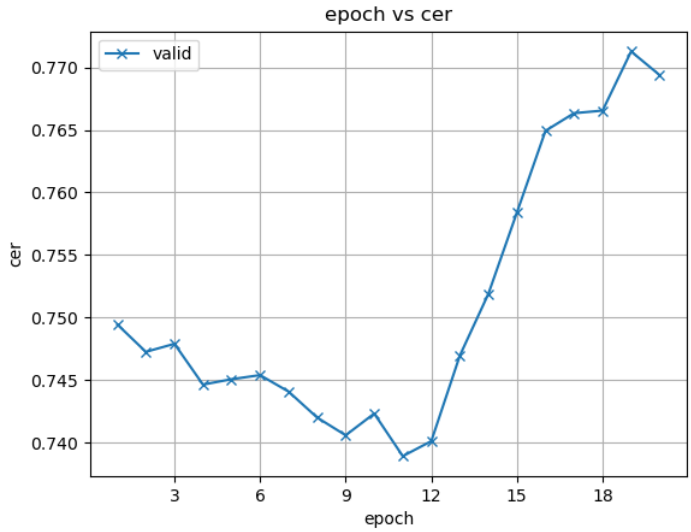


FIGURA 13. Tasa de Error de Carácter del modelo de 194 horas y 20 épocas.



Debido a que los resultados del primer modelo no fueron satisfactorios, se desarrolló un segundo modelo en el que se incrementó el número de horas de audio a 388 pero con las mismas 20 épocas, de las cuales solo se completaron 18. La razón de lo anterior se debió a que los parámetros del modelo se configuraron para detenerse en caso de que el porcentaje de exactitud tendiera a caer en un sobre entrenamiento. La gráfica mostrada en la Figura 14 permite observar el aumento continuo de la exactitud conforme avanzaban las épocas llegando a tener hasta un 65%. Esto sucedió hasta la época 18 en donde la exactitud comenzó a decaer en el conjunto de validación. La métrica CER obtuvo un aproximado del 43%, en comparación con el primero modelo el error disminuyó (Figura 15).

FIGURA 14. Exactitud del modelo de 388 horas y 20 épocas.

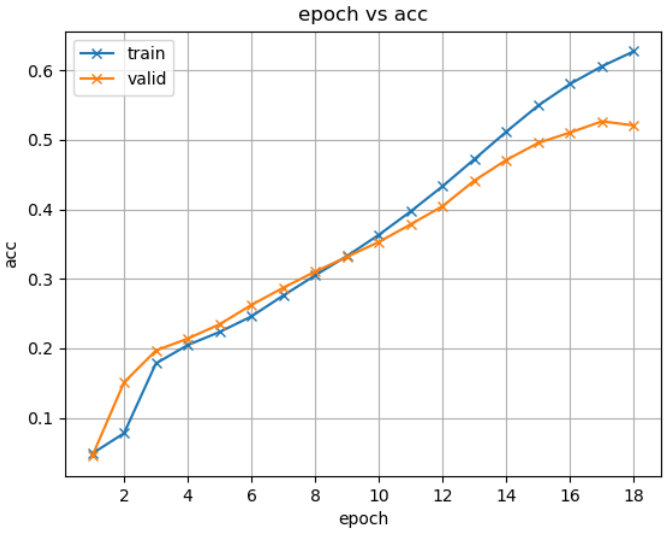
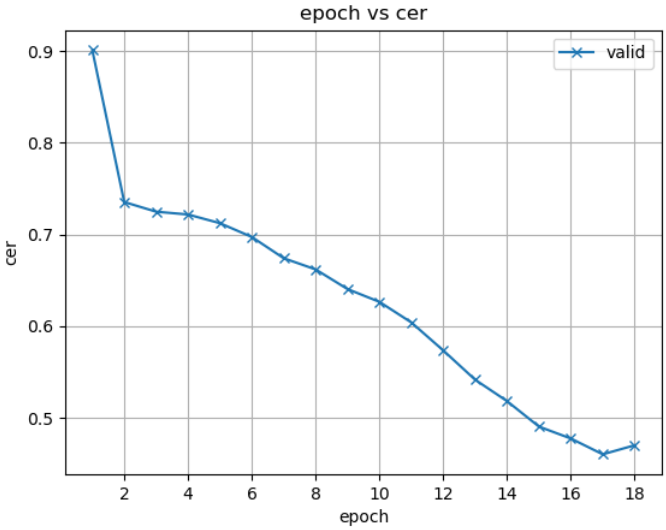


FIGURA 15. Tasa de Error de Carácter del modelo de 388 horas y 20 épocas.



Después de analizar los datos obtenidos por el segundo modelo y observar que se presentó una mejora en su desempeño, se desarrolló el tercer modelo. En este modelo se utilizó de igual forma la cantidad de 388 horas de audio pero se incrementó el número de épocas a 60. Solo se lograron completar 20 debido a la configuración de parámetros para evitar el sobre entrenamiento. El comportamiento de las gráficas es similar al anterior modelo, de hecho, solo fueron dos épocas más de entrenamiento en comparación con el segundo modelo. En la Figura 16 se observa que el porcentaje de exactitud es de 66% mientras que la métrica CER llegó a 44%, similar al modelo anterior. Hasta este momento el tercer modelo no mostró mejora en comparación con el segundo modelo. El número de épocas utilizado en el tercer modelo no impactó en la mejora del modelo.

FIGURA 16. Exactitud del modelo de 388 horas y 60 épocas.

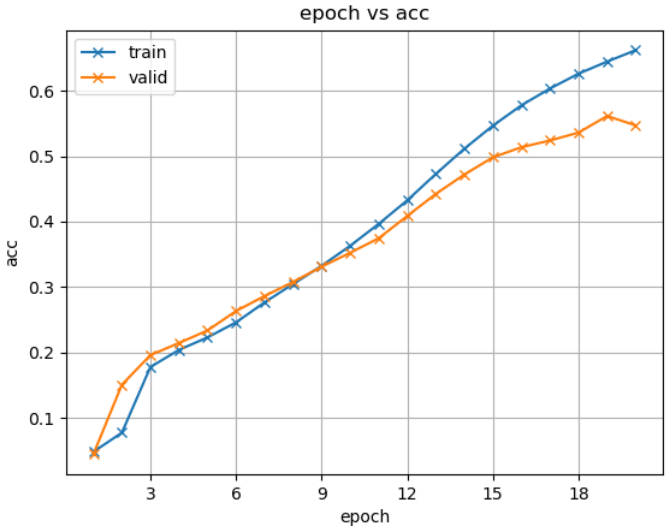
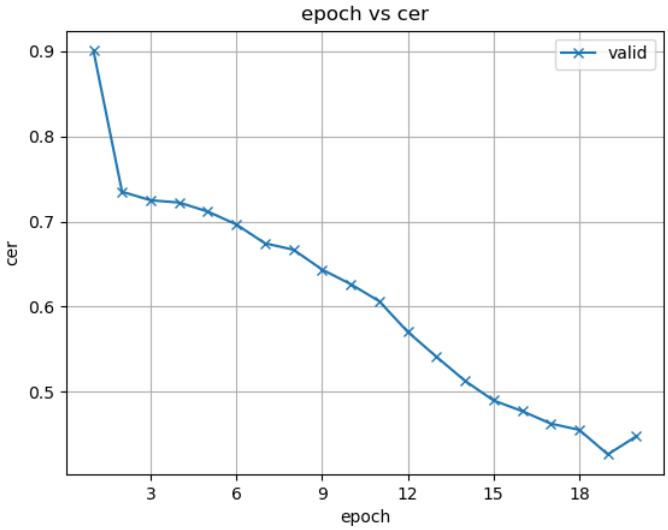


FIGURA 17. Tasa de Error de Carácter del modelo de 388 horas y 20 épocas.



Se decidió crear un cuarto modelo incrementando la cantidad de horas a 766 con 60 épocas. Este modelo se detuvo en la época 14 debido a la misma configuración de parámetros que evitan el sobre entrenamiento. En la gráfica mostrada en la Figura 18 se aprecia que el conjunto de entrenamiento llegó a una exactitud del 88%, mostrando una mejora con respecto al modelo anterior. Mientras tanto, la métrica CER llegó a 31%, lo cual indica que mejoró la transcripción a comparación de los modelos anteriores.

FIGURA 18. Exactitud del modelo de 776 horas y 60 épocas.

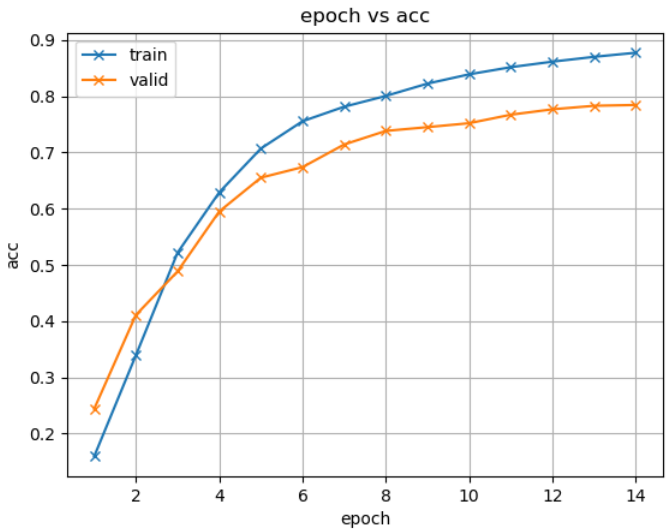
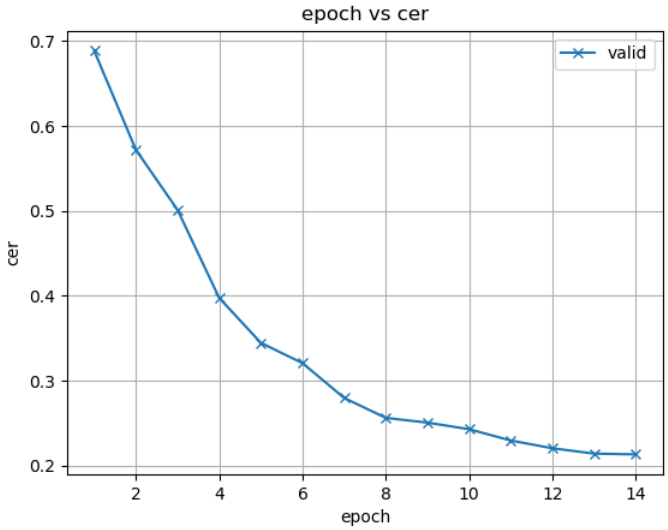


FIGURA 19. Tasa de Error de Carácter del modelo de 776 horas con 60 épocas.



Debido a que se observó que en las épocas posteriores la exactitud seguía incrementando, se decidió desarrollar un quinto modelo en el que se decidió eliminar la configuración de detección temprana con

el fin de evitar que el entrenamiento se detuviera. Este nuevo modelo se entrenó con 1,000 horas de audio del conjunto de datos de *LibriSpeech* y 4 horas del conjunto *L2-ARCTIC* con un total de 100 épocas. Cabe mencionar que en esta ocasión se realizaron las 100 épocas completas. En la Figura 20 se muestra la gráfica relacionada con la exactitud del modelo con cada época realizada. Al final se obtuvo un porcentaje del 90% para el conjunto de entrenamiento y un 86% para el conjunto de validación. Por otro lado, la métrica CER logró un 9.5% como se muestra en la Figura 21.

FIGURA 20. Exactitud del modelo de 1000 horas y 100 épocas.

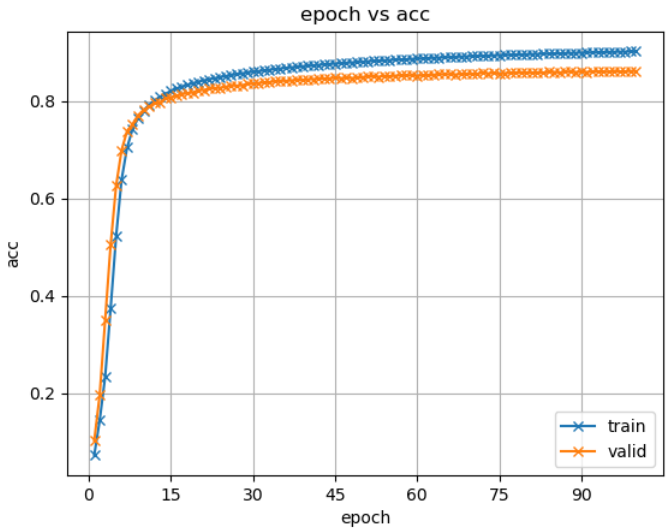
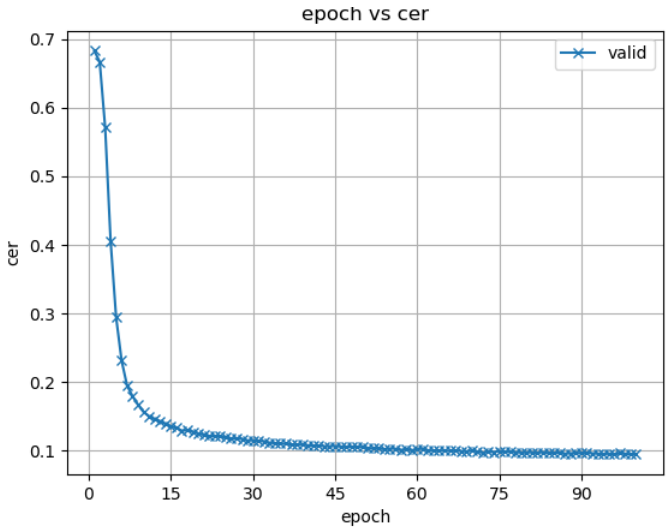


FIGURA 21. Tasa de Error de Carácter del modelo de 1000 horas y 100 épocas.



La Tabla 2 muestra los resultados obtenidos por los modelos desarrollados. Se puede observar que el modelo con menor exactitud y mayor porcentaje de la métrica CER es el modelo uno. Esto claramente se debe a la cantidad de datos utilizados para el entrenamiento del modelo. El modelo cinco presentó el

mejor porcentaje de exactitud y un CER menor a los demás modelos. En base a los resultados obtenidos al comparar los cinco modelos, se determinó que el modelo cinco presentó mejor rendimiento.

TABLA 2. Comparación de modelos.

	Exactitud	CER	Datos utilizados	Época esperada	Época máxima	Tiempo de entrenamiento
Modelo 1	8.50%	77%	194 horas	20	20	5 horas
Modelo 2	65%	43%	388 horas	20	18	4 horas
Modelo 3	66%	43%	388 horas	60	20	5 horas
Modelo 4	88%	31%	776 horas	60	14	3 horas
Modelo 5	**90%**	**9.50%**	**1000 horas**	**100**	**100**	**212 horas**

5.2 Evaluación del modelo ASR con usuarios reales

Después de haber determinado el modelo ASR, se evaluó su desempeño con usuarios reales. Para este fin, se reunió a un grupo de 10 personas con las características mostradas en la Tabla 3. Cada participante realizó las pruebas utilizando este modelo con el fin de medir su rendimiento en base a la métrica CER. Esta métrica se utilizó de manera similar en [64] y [47], con la diferencia de que en lugar de emplear la unidad lingüística a nivel carácter (como en el presente trabajo) se utilizó a nivel fonema. Cabe mencionar que de acuerdo con [65], las unidades lingüísticas comúnmente utilizadas para las evaluaciones de modelos ASR son el fonema y el caracter.

TABLA 3. Características de los participantes.

	Edad	Sexo	Nacionalidad	Nivel de Inglés	Característica importante
Persona 1	27	Masculino	Mexicana	Intermedio	Estudiante
Persona 2	25	Femenino	Mexicana	Alto	Profesora de Inglés
Persona 3	24	Masculino	Mexicana	Intermedio	Estudiante
Persona 4	24	Femenino	Mexicana	Alto	Pronunciación nativa
Persona 5	27	Masculino	Mexicana	Intermedio	Estudiante
Persona 6	35	Masculino	Estadounidense	Alto	Pronunciación nativa
Persona 7	34	Femenino	Mexicana	Intermedio	Estudiante
Persona 8	14	Masculino	Mexicana	Principiante	Estudiante
Persona 9	28	Masculino	Estadounidense	Alto	Pronunciación nativa
Persona 10	30	Femenino	Mexicana	Intermedio	Estudiante

Para realizar las pruebas se seleccionó un conjunto de 10 frases que contuvieran palabras que son comúnmente mal pronunciadas por estudiantes. Estas palabras fueron extraídas del estudio realizado en [66]. Para fines de esta investigación, se eligió un conjunto de diez palabras de dicha investigación con las que se seleccionaron diez oraciones del sitio web de traducción automática *Linguee* de la empresa

DeepL Translator [67]. Las diez oraciones mostradas a continuación fueron las utilizadas para las evaluaciones donde las palabras resaltadas son aquellas que el estudio encontró como difíciles de pronunciar, estas son:

1. *She has **enough** time.*
2. *It is a **pleasure** to meet you.*
3. *The first one is to **measure** the monetary value.*
4. *Their duties **should** be specified.*
5. *But I **think** it is hardly.*
6. *I cannot feed every **beast** of the field.*
7. *Classify **objects** based on their fragility.*
8. *The price per **hour** may vary.*
9. *It is a **slow** swimmer.*
10. *Create **elegant** online albums.*

El usuario tiene la opción de activar o desactivar la evaluación de la pronunciación en el chatbot (Figura 11). Si la opción esta desactivada el usuario solo interactúa de manera textual con él. Para realizar el procedimiento de evaluación del módulo de detección de errores de pronunciación dicha opción permaneció activa.

La evaluación consistió en que cada uno de los participantes escribiera y pronunciara cada una de las oraciones en el orden mostrado. Luego de introducir una oración por teclado, el participante debía pronunciarla. A su vez, la oración se procesaba por el chatbot haciendo uso del lenguaje AIML para encontrar una respuesta adecuada dentro de su base de conocimiento. El modelo ASR convirtió el audio de la pronunciación a texto. Posteriormente, ambas cadenas se compararon para encontrar la cantidad de sustituciones, eliminaciones e inserciones entre los textos. Finalmente, los posibles errores de pronunciación y la respuesta del chatbot eran mostrados al usuario. De esta manera se otorgó al usuario una interacción natural con el chatbot y a la vez la oportunidad de practicar su pronunciación.

En la Tabla 4 se muestran los resultados obtenidos. Las filas representan a los participantes mientras que las columnas las oraciones. Los datos presentados corresponden al porcentaje de CER obtenido por cada uno de los participantes al pronunciar cada una de las oraciones. Los participantes sombreados en color gris obscuro, cuya pronunciación es nativa al inglés, obtuvieron los porcentajes de CER más bajos, es decir, realizaron una buena pronunciación y el proceso de alineación de cadenas encontró pocas

discrepancias. Por otro lado, los participantes sombreados en color gris claro fueron aquellos que obtuvieron altos porcentajes CER, de los cuales, uno de ellos cuenta con un nivel de inglés principiante.

TABLA 4. Resultados de las evaluaciones.

Participante	Oración									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	16%	18%	30%	31%	08%	00%	34%	22%	25%	36%
2	00%	00%	11%	13%	17%	05%	44%	07%	05%	14%
3	00%	07%	09%	13%	00%	11%	07%	15%	10%	39%
4	00%	00%	02%	00%	00%	13%	27%	15%	10%	29%
5	00%	14%	15%	44%	00%	26%	22%	11%	25%	14%
6	00%	00%	07%	00%	08%	00%	10%	07%	00%	07%
7	21%	00%	13%	38%	00%	05%	20%	19%	30%	11%
8	63%	00%	43%	34%	17%	16%	24%	44%	35%	18%
9	00%	04%	30%	19%	00%	03%	22%	15%	20%	32%
10	00%	00%	02%	41%	00%	03%	15%	19%	40%	21%

5.3 Resultado de ejemplo obtenido del proceso de alineación de cadenas

El proceso de alineación de cadenas dentro de la arquitectura del sistema se compone de: 1) la transcripción canónica, 2) la transcripción inferida por el ASR y 3) un algoritmo de alineación de secuencias, en este caso la distancia de Levenshtein. Para ejemplificar el resultado obtenido del proceso de alineación de cadenas se presentan a continuación dos ejemplos utilizando una de las diez frases mencionadas anteriormente. El primer ejemplo considera el resultado obtenido por uno de los participantes que obtuvo los porcentajes de CER más bajos. El segundo ejemplo considera el resultado de uno de los participantes con el CER más alto.

1) Resultado obtenido del participante con el CER más bajo:

Transcripción canónica: *She has enough time (longitud 19)*
Transcripción inferida: *She has enough time (longitud 19)*
Salida: `[ ]`

2) Resultado obtenido del participante con el CER más alto:

Transcripción canónica: *She has enough time (longitud 19)*
Transcripción inferida: *She fast and no attain (longitud 22)*
Salida: `[('replace',4, 4), ('insert',7, 7), ('replace',8, 9), ('replace',10, 11), ('replace',11, 12), ('replace',12, 13),`

```
('replace',13,  14),  ('insert',15,  16),  ('insert',15,  17),
('replace',16, 19), ('replace',17, 20), ('replace',18, 21)]
```

En el primer ejemplo, realizado por el participante 6, ambas secuencias cuentan con la misma longitud y la salida no cuenta con ningún tipo de operación (sustitución, eliminación o inserción) realizada. En el segundo ejemplo, realizado por el participante 8, se aprecia que la salida cuenta con una serie de operaciones, además que la longitud es diferente entre ambas cadenas. Cada una de las operaciones (`('replace',4, 4)`) se interpreta de la siguiente forma:

- El primer argumento corresponde al tipo de operación realizada.
- El segundo argumento indica la posición en primera cadena.
- El tercer argumento indica la posición en segunda cadenas.

La operación se realiza en las posiciones indicadas, por lo que, la salida `('replace',4, 4)` puede leerse como, "se remplazó el caracter de la posición cuarto de la primera cadena por la posición cuatro de la segunda cadena". La longitud de las cadenas comienza en 0 y termina hasta *n*-1. Para ejemplificar lo anterior, tómese en cuenta los resultados mostrados en el caso 2. La palabra *has* sufrió dos operaciones para convertirse en la palabra *fast*:

- Se remplazó el caracter de la posición cuarto de la primera cadena por la posición cuatro de la segunda cadena.
- Se insertó un caracter en la posición siete.

5.4 Discusiones

Los resultados obtenidos muestran un buen funcionamiento del sistema de detección de errores de pronunciación. Esto se confirma con el desempeño de los participantes, ya que los de nivel alto de inglés tendieron a obtener mejores resultados (como se esperaba), mientras que el resto consiguió puntajes elevados de CER. Un factor que influyó en los resultados fueron las pausas y la entonación al momento de realizar las pruebas. La métrica se vio afectada positivamente cuando las personas realizaban la pronunciación con todo de voz alta y pausas entre palabras.

El desempeño obtenido por el modelo ASR cumplió con lo esperado. La exactitud obtenida en el entrenamiento hizo posible que las conversiones de la señal de audio a texto fueran las adecuadas. Así mismo, dichas conversiones variaron según la experiencia en el idioma de cada uno de los participantes, lo que denota que es necesario producir una buena pronunciación para que pueda ser producido el texto

correcto. Al momento de realizar el proceso de alineación de cadenas, el texto producido por el modelo ASR debe realizar una transcripción lo más exacto posible, lo cual se consigue realizando una buena pronunciación teniendo en cuenta las pausas y la entonación.

Por otro lado, se observó que el chatbot y su interfaz gráfica consiguieron una cómoda interacción entre el usuario y el sistema de detección de errores de pronunciación. Debido a que las palabras en donde se presentó un error eran mostradas al usuario por el chatbot, estos tendían modificar la pronunciación de ellas con la intención de mitigar el error. Así mismo, el proceso de dialogo animó a los usuarios a producir sus propias oraciones para mantener la conversación con el chatbot del sistema.

VI. Conclusiones

En este trabajo se presentó un módulo de detección de errores de pronunciación integrado a la arquitectura de un chatbot. El sistema recibe una entrada de audio de la frase en la cual el usuario desea evaluar su pronunciación. Como salida, el chatbot muestra las posibles palabras mal pronunciadas.

Se compone de dos componentes principales: un modelo ASR y un algoritmo de alineación de cadenas. El ASR fue creado con una red transformadora empleando los conjuntos de datos *LibriSpeech* y *L2-ARCTIC* para su entrenamiento. Esta es utilizada para convertir la señal de audio a su respectivo texto para posteriormente detectar palabras mal pronunciadas utilizando el algoritmo de la distancia de Levenshtein.

El modelo ASR alcanzó una precisión del 90% y un CER del 9.50%, mostrando resultados prometedores en la detección de errores de pronunciación. Además, se evaluó utilizando un conjunto de diez usuarios reales. Dicha evaluación presentó mejores resultados en los usuarios que cuentan con una pronunciación nativa. Los usuarios que obtuvieron valores altos de CER son aquellos que cuentan con un nivel de inglés principiante.

Con base en los resultados en las pruebas realizadas, es posible concluir que el módulo de detección de errores de pronunciación logró un buen desempeño. Esto debido a que los participantes con un nivel alto de inglés tienden a obtener mejores resultados con valores CER bajos, mientras que el resto obtiene valores CER altos. Otro factor importante por destacar son las pausas y la entonación realizada por algunos usuarios. Los resultados se ven afectados positivamente cuando los usuarios hablaron con un tono de voz alto y realizaron las pausas pertinentes. Por lo anterior, se puede decir que el módulo de detección de errores de pronunciación es útil para que los usuarios practiquen hasta mejorar progresivamente mientras tienen la posibilidad de interactuar con el chatbot.

Como trabajo futuro, se espera aumentar la base de conocimiento del chatbot para lograr una mayor naturalidad en la interacción con los usuarios. Aplicar una técnica de Procesamiento de Lenguaje Natural para que el chatbot tenga la habilidad de generar mejores respuestas, o en su defecto, crear un modelo Texto a Voz para aumentar aún más el grado de interacción. Así mismo, emplear un conjunto de datos en donde se utilicen fonemas. Esto con las finalidad de realizar detección de errores más precisos, además de tener la capacidad de distinguir acentuación léxica y acento tonal.

Dos artículos de investigación fueron realizados como producto de este proyecto. El primero de ellos titulado *Mispronunciation Detection and Diagnosis Through a Chatbot* [68] fue publicado como capítulo de la colección del libro *Handbook of Research on Natural Language Processing and Smart Service Systems* en el año 2021. Se espera que el segundo trabajo titulado *English mispronunciation detection module using a Transformer network integrated into a chatbot*, el cual ha sido aceptado, sea publicado en el año 2022.

Referencias

[1] K. Cronquist and A. Fiszbein, "El aprendizaje del inglés en América Latina," *El Aprendiz. del inglés en Amércica Lat.*, pp. 1–88, 2017, [Online]. Available: https://www.thedialogue.org/wp-content/uploads/2017/09/El-aprendizaje-del-inglés-en-América-Latina-1.pdf.

[2] N. Haristiani, "Artificial Intelligence (AI) Chatbot as Language Learning Medium: An inquiry," *J. Phys. Conf. Ser.*, vol. 1387, no. 1, 2019, doi: 10.1088/1742-6596/1387/1/012020.

[3] P. H. Su, C. H. Wu, and L. S. Lee, "A recursive dialogue game for personalized computer-aided pronunciation training," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 1, pp. 127–141, 2015, doi: 10.1109/TASLP.2014.2375572.

[4] A. Gilakjani, S. Ahmadi, and M. Ahmadi, "Why is Pronunciation So Difficult to Learn?," *English Lang. Teach.*, vol. 4, no. 3, 2011, doi: 10.5539/elt.v4n3p74.

[5] K. Anwar and R. Husniah, "Evaluating Integrated Task Based Activities and Computer Assisted Language Learning (CALL)," *English Lang. Teach.*, vol. 9, no. 4, p. 119, 2016, doi: 10.5539/elt.v9n4p119.

[6] B. A. Shawar, "Integrating CALL Systems with Chatbots as Conversational Partners," *Comput. y Sist.*, vol. 21, no. 4, pp. 615–626, 2017, doi: 10.13053/CyS-21-4-2868.

[7] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing: An introduction," *J. Am. Med. Informatics Assoc.*, vol. 18, no. 5, pp. 544–551, 2011, doi: 10.1136/amiajnl-2011-000464.

[8] J. J. González-Barbosa, J. Frausto Solís, J. P. Sánchez-Hernández, and J. P. Sanchez-Solís, "Two New Challenging Resources to Evaluate Natural Language Interfaces to Databases Generated Based on Geobase and Geoquery," in *Handbook of Research on Natural Language Processing and Smart Service Systems*, 2021, pp. 70–100.

[9] R. A. Pazos-Rangel, G. Rivera, J. A. Martínez F., J. Gaspar, and R. Florencia-Juárez, "Natural Language Interfaces to Databases: A Survey on Recent Advances," in *Handbook of Research on Natural Language Processing and Smart Service Systems*, 2021, pp. 1–30.

[10] J. A. Porras Medrano, R. Florencia Juárez, G. Rivera Zárate, and V. García Jiménez, "Interfaz de lenguaje natural para consultar cubos multidimensionales utilizando procesamiento analítico en línea," *Res. Comput. Sci.*, vol. 147, no. 6, pp. 153–165, 2018, doi: 10.13053/rcs-147-6-12.

[11] G. Pazos-Rangel, R. A., Florencia-Juarez, R., Paredes-Valverde, M. A., & Rivera, *Handbook of Research on Natural Language Processing and Smart Service Systems*. IGI Global, 2021.

[12] G. Rivera, R. Florencia, V. García, A. Ruiz, and J. P. Sánchez-Solís, "News classification for identifying traffic incident points in a Spanish-speaking country: A real-world case study of class imbalance learning," *Appl. Sci.*, vol. 10, no. 18, 2020, doi: 10.3390/APP10186253.

[13] R. Jiménez, V. García, K. Olmos-Sánchez, A. Ponce, and J. Rodas-Osollo, "Identifying

Suggestions in Airline-User Tweets Using Natural Language Processing and Machine Learning," in *Handbook of Research on Natural Language Processing and Smart Service Systems*, 2021, pp. 481–498.

[14] R. Jiménez, V. García, A. López, A. Mendoza Carreón, and A. Ponce, "Opinion Mining for Instructor Evaluations at the Autonomous University of Ciudad Juarez," in *Handbook of Research on Natural Language Processing and Smart Service Systems*, 2021, pp. 427–444.

[15] A. Requejo Flores, A. Ruiz, R. Mar, and R. Porras, "Location Extraction to Inform a Spanish-Speaking Community About Traffic Incidents," in *Handbook of Research on Natural Language Processing and Smart Service Systems*, 2021, pp. 347–367.

[16] A. Requejo Flores, A. Ruiz, A. López, and R. Porras, "News Classification to Notify About Traffic Incidents in a Mexican City," in *Handbook of Research on Natural Language Processing and Smart Service Systems*, 2021, pp. 227–244.

[17] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural Language Processing: State of The Art, Current Trends and Challenges," no. Figure 1, 2017, [Online]. Available: http://arxiv.org/abs/1708.05148.

[18] Tomáš Zemčík, "A Brief History of Chatbots," in *Perception, Control, Cognition*, 2018, no. October, doi: 10.12783/dtcse/aicae2019/31439.

[19] R. Dale, "The return of the chatbots," *Nat. Lang. Eng.*, vol. 22, no. 5, pp. 811–817, 2016, doi: 10.1017/S1351324916000243.

[20] M. L. Morales-Rodríguez, J. J. González B., R. Florencia Juárez, H. J. Fraire Huacuja, and J. A. Martínez Flores, "Emotional conversational agents in clinical psychology and psychiatry," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6437 LNAI, no. PART 1, pp. 458–466, 2010, doi: 10.1007/978-3-642-16761-4_40.

[21] G. De Gasperis, I. Chiari, and N. Florio, *AIML knowledge base construction from text corpora*, vol. 427, no. June 2014. 2013.

[22] E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," *Mach. Learn. with Appl.*, vol. 2, no. November, p. 100006, 2020, doi: 10.1016/j.mlwa.2020.100006.

[23] O. Villanueva-Mendoza, M. V. González, M. Varela, and L. Zamora, "Chatbot for the Improvement of Conversational Skills of Japanese Language Learners," in *Handbook of Research on Natural Language Processing and Smart Service Systems*, 2021, pp. 101–134.

[24] N. Sandu and E. Gide, "Adoption of AI-chatbots to enhance student learning experience in higher education in india," *2019 18th Int. Conf. Inf. Technol. Based High. Educ. Training, ITHET 2019*, no. January, pp. 1–6, 2019, doi: 10.1109/ITHET46829.2019.8937382.

[25] G. Molnar and Z. Szuts, "The Role of Chatbots in Formal Education," *SISY 2018 - IEEE 16th Int. Symp. Intell. Syst. Informatics, Proc.*, pp. 197–201, 2018, doi: 10.1109/SISY.2018.8524609.

[26] S. Ondas, M. Pleva, and D. Hladek, "How chatbots can be involved in the education process," *ICETA 2019 - 17th IEEE Int. Conf. Emerg. eLearning Technol. Appl. Proc.*, pp. 575–580, 2019,

doi: 10.1109/ICETA48886.2019.9040095.

[27] U. Kamath, J. Liu, and J. Whitaker, *Deep Learning for NLP and Speech Recognition*. Cham, Switzerland: Springer Nature Switzerland AG, 2019.

[28] P. K. Kurzekar, R. R. Deshmukh, V. B. Waghmare, and P. P. Shrishrimal, "A Comparative Study of Feature Extraction Techniques for Speech Recognition System," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 03, no. 12, pp. 18006–18016, 2014, doi: 10.15680/ijirset.2014.0312034.

[29] R. Ranjan and A. Thakur, "Analysis of feature extraction techniques for speech recognition system," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 7C2, pp. 197–200, 2019.

[30] N. Dave, "Feature Extraction Methods LPC , PLP and MFCC In Speech Recognition," *Int. J. Adv. Res. Eng. Technol.*, vol. 1, no. Vi, pp. 1–5, 2013.

[31] H. Z. Muhammad, M. Nasrun, C. Setianingsih, and M. A. Murti, "Speech recognition for English to Indonesian translator using hidden Markov model," *2018 Int. Conf. Signals Syst. ICSigSys 2018 - Proc.*, pp. 255–260, 2018, doi: 10.1109/ICSIGSYS.2018.8372768.

[32] M. Gales and S. Young, "The application of hidden Markov Models in speech recognition," *Found. Trends Signal Process.*, vol. 1, no. 3, pp. 195–304, 2007, doi: 10.1561/2000000004.

[33] S. J. Cox, "Hidden Markov Models for Automatic Speech Recognition: Theory and Application.," *Br. Telecom Technol. J.*, vol. 6, no. 2, pp. 105–115, 1988.

[34] S. J. Melnikoff, S. F. Quigley, and M. J. Russell, "Implementing a hidden Markov model speech recognition system in programmable logic," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2147, no. June 2014, pp. 81–90, 2001, doi: 10.1007/3-540-44687-7_9.

[35] K. Yun, J. Osborne, M. Lee, T. Lu, and E. Chow, "Automatic speech recognition for launch control center communication using recurrent neural networks with data augmentation and custom language model," *arXiv*, 2018, doi: 10.1117/12.2304569.

[36] A. Amberkar, P. Awasarmol, G. Deshmukh, and P. Dave, "Speech Recognition using Recurrent Neural Networks," *Proc. 2018 Int. Conf. Curr. Trends Towar. Converging Technol. ICCTCT 2018*, no. March 2018, pp. 1–4, 2018, doi: 10.1109/ICCTCT.2018.8551185.

[37] L. K.R and S. Elizabeth, "Automatic Speech Recognition using different Neural Network Architectures – A Survey," *Int. J. Comput. Sci. Inf. Technol.*, vol. 7(6), no. 6, pp. 2422–2427, 2016.

[38] D. Wang, X. Wang, and S. Lv, "An overview of end-to-end automatic speech recognition," *Symmetry (Basel).*, vol. 11, no. 8, pp. 1–27, 2019, doi: 10.3390/sym11081018.

[39] H. Franco, L. Neumeyer, M. Ramos, and H. Bratt, "Automatic Detection Of Phone-Level Mispronunciation For Language Learning," *Learn. Proc. Eurospeech 99*, no. June, pp. 851–854, 1999, [Online]. Available: http://leoneu.github.io/pub/eurospeech99.pdf.

[40] K. Li, X. Qian, and H. Meng, "Mispronunciation Detection and Diagnosis in L2 English Speech Using Multidistribution Deep Neural Networks," *IEEE/ACM Trans. Audio Speech Lang.*

Process., vol. 25, no. 1, pp. 193–207, 2017, doi: 10.1109/TASLP.2016.2621675.

[41] A. Neri, C. Cucchiarini, and W. Strik, "Automatic Speech Recognition for second language learning: How and why it actually works," *15th Int. Congr. Phonetic Sci.*, no. May 2014, pp. 1157–1160, 2003, [Online]. Available: http://s3.amazonaws.com/academia.edu.documents/40482248/Automatic_speech_recognition_for_second_20151129-16875-dyzqzg.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1467156207&Signature=P5CNx3XwiEmR7sjgreYoai7sN0g=&response-content-disposition=inline; fi.

[42] H. Wang, J. Xu, H. Ge, and Y. Wang, "Design and implementation of an english pronunciation scoring system for pupils based on DNN-HMM," *Proc. - 10th Int. Conf. Inf. Technol. Med. Educ. ITME 2019*, pp. 348–352, 2019, doi: 10.1109/ITME.2019.00085.

[43] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Commun.*, vol. 30, no. 2, pp. 95–108, 2000, doi: 10.1016/S0167-6393(99)00044-8.

[44] D. Luo, L. Xia, C. Zhang, and L. Wang, "Automatic Pronunciation Evaluation in High-states English Speaking Tests Based on Deep Neural Network Models," *2019 2nd Int. Conf. Artif. Intell. Big Data, ICAIBD 2019*, pp. 124–128, 2019, doi: 10.1109/ICAIBD.2019.8836976.

[45] W. K. Leung, X. Liu, and H. Meng, "CNN-RNN-CTC Based End-to-end Mispronunciation Detection and Diagnosis," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2019-May, pp. 8132–8136, 2019, doi: 10.1109/ICASSP.2019.8682654.

[46] A. Jettakul, C. Thamjarat, K. Liaowongphuthorn, C. Udomcharoenchaikit, P. Vateekul, and P. Boonkwan, "A Comparative Study on Various Deep Learning Techniques for Thai NLP Lexical and Syntactic Tasks on Noisy Data," *Proceeding 2018 15th Int. Jt. Conf. Comput. Sci. Softw. Eng. JCSSE 2018*, pp. 1–6, 2018, doi: 10.1109/JCSSE.2018.8457368.

[47] L. Zhang *et al.*, "End-to-end automatic pronunciation error detection based on improved hybrid ctc/attention architecture," *Sensors (Switzerland)*, vol. 20, no. 7, pp. 1–24, 2020, doi: 10.3390/s20071809.

[48] Y. Zhao, J. Li, X. Wang, and Y. Li, "The Speechtransformer for Large-scale Mandarin Chinese Speech Recognition," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2019-May, pp. 7095–7099, 2019, doi: 10.1109/ICASSP.2019.8682586.

[49] Z. Zhang, Y. Wang, and J. Yang, "Text-conditioned transformer for automatic pronunciation error detection," *arXiv*, 2020.

[50] P. M. Revell-Rogerson, "Computer-Assisted Pronunciation Training (CAPT): Current Issues and Future Directions," *RELC J.*, 2021, doi: 10.1177/0033688220977406.

[51] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2007, doi: 10.2753/MIS0742-1222240302.

[52] S. Watanabe *et al.*, “ESPNet: End-to-end speech processing toolkit,” *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2018-Septe, no. September, pp. 2207–2211, 2018, doi: 10.21437/Interspeech.2018-1456.

[53] A. Sinha, N. Banerjee, A. Sinha, and R. K. Shastri, “Interference of first language in the acquisition of second language,” *J. Psychol. Couns.*, vol. 1, no. 7, pp. 117–122, 2009.

[54] J. E. Flege, “Second Language Speech Learning: Theory, Findings, and Problems,” *Speech Percept. Linguist. Exp. Issues Cross-Language Res.*, no. January 1995, pp. 233–277, 1995.

[55] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2015-Augus, pp. 5206–5210, 2015, doi: 10.1109/ICASSP.2015.7178964.

[56] A. Vaswani *et al.*, “Attention is all you need,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.

[57] L. Dong, S. Xu, and B. Xu, “SPEECH-TRANSFORMER : A NO-RECURRENCE SEQUENCE-TO-SEQUENCE MODEL FOR SPEECH RECOGNITION,” pp. 5884–5888, 2018, [Online]. Available: http://150.162.46.34:8080/icassp2018/ICASSP18_USB/pdfs/0005884.pdf.

[58] M. Espi, M. Fujimoto, K. Kinoshita, and T. Nakatani, “Exploiting spectro-temporal locality in deep learning based acoustic event detection,” *Eurasip J. Audio, Speech, Music Process.*, vol. 2015, no. 1, 2015, doi: 10.1186/s13636-015-0069-2.

[59] R. Haldar and D. Mukhopadhyay, “Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach,” no. January 2011, 2011, [Online]. Available: http://arxiv.org/abs/1101.1232.

[60] A. Name and C. Name, “Experiments of ASR-based mispronunciation detection for children and adult English learners,” no. i, pp. 2–6.

[61] K. Li, S. Mao, X. Li, Z. Wu, and H. Meng, “Automatic lexical stress and pitch accent detection for L2 English speech using multi-distribution deep neural networks,” *Speech Commun.*, vol. 96, no. December 2019, pp. 28–36, 2018, doi: 10.1016/j.specom.2017.11.003.

[62] R. Ardila *et al.*, “Common Voice: A Massively-Multilingual Speech Corpus,” 2019, [Online]. Available: http://arxiv.org/abs/1912.06670.

[63] G. Zhao, S. Sonsaat, A. Silpachai, and I. Lucic, “L2-ARCTIC : A Non-Native English Speech Corpus,” *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, no. September, pp. 2783–2787, 2018.

[64] X. L. and H. M. H.-C. Wai-Kim Leung, “CNN-RNN-CTC BASED END-TO-END MISPRONUNCIATION DETECTION AND DIAGNOSIS,” pp. 8132–8136, 2019.

[65] E. D. Emiru, S. Xiong, Y. Li, and A. Fesseha, “Connectionist Temporal Classification with Attention Model and Phoneme - Based Byte - Pair - Encodings,” 2021.

[66] A. Komariah, “Problems in Pronouncing the English Sounds Faced by the Students of SMPN 2

Halong, Banjar," *J. English Lang. Pedagog.*, vol. 1, no. 2, pp. 1–10, 2019, doi: 10.36597/jelp.v1i2.4127.

[67] A. Schmidhofer and N. Mair, "Machine Translation in Translator Education," vol. 4, no. 2, pp. 163–180, 2018, [Online]. Available: https://doi.org/10.14201/clina201842163180.

[68] M. E. Martinez, F. López-Orozco, K. Olmos-Sánchez, and J. P. Sánchez-Solís, "Mispronunciation Detection and Diagnosis Through a Chatbot," in *Handbook of Research on Natural Language Processing and Smart Service Systems*, 2021, pp. 31–45.