PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

RADIO MOSCÚ, ANÁLISIS Y MEJORAMIENTO DE AUDIO PARA TRANSCRIPCIÓN E INDEXACIÓN DE DATOS

DANIEL IGNACIO NAVARRO BILBAO
FABIÁN ESTEBAN VERGARA LOBOS

INFORME FINAL DE PROYECTO PARA
OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO CIVIL EN INFORMÁTICA

DICIEMBRE, 2018

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

RADIO MOSCÚ, ANÁLISIS Y MEJORAMIENTO DE AUDIO PARA TRANSCRIPCIÓN E INDEXACIÓN DE DATOS

DANIEL IGNACIO NAVARRO BILBAO
FABIÁN ESTEBAN VERGARA LOBOS

Profesor Guía: Héctor Allende Cid
Profesor Co-referente: Francisco Cifuentes Silva

Carrera: **Ingeniería Civil Informática**

DICIEMBRE, 2018

Dedicado a todas las personas
que me hicieron feliz en este proyecto personal,
a los días de lluvia del sur,
las semanas de sol entre cerros,
los meses de buenas amistades en los estudios
y los años viviendo con una magnífica familia.

Daniel Navarro Bilbao

A mi madre, mi mama
,mi padre y todas las personas
que me dieron su apoyo durante este proceso.

Fabián Vergara Lobos

Índice

Resumen

La colección de audios de la Radio Moscú presentes en la Biblioteca del Congreso Nacional de Chile son elementos históricos con una gran fuente de información. Estos audios contienen una variedad de distorsiones que no permiten su correcta comprensión, por otro lado, la biblioteca solo almacena estas cintas en bruto y no se cuenta con herramientas que permitan un análisis exhaustivo. Esta investigación se centró en analizar técnicas de tratamiento de audio, para luego implementar una serie de servicios, tales como la eliminación de ruidos, la transcripción de estos audios y la indexación de documentos con el propósito de poder estudiar y acceder a esta serie radial.

Palabras Clave: Eliminación de ruido, transcripción, indexación, Biblioteca del Congreso Nacional, Radio Moscú.

Abstract

The National Library of the Chilean Congress keeps the audios of the Radio Moscu, this collections has a very important piece of history with a lot of information. This audios has distortions that make difficult to understand and know what it says. On the other side this library just keeps this collection and doesn't have the tools to make something with this audios, like doing a exhaustives analysis. In this research we analyzed some techniques to make an speech enhancement and then implement services to make this audios more comprensive, transcribe it and realize an indexation of documents. All this with the purpose to be able to study and share this radio show.

Palabras Clave: Speech enhancement, transcription, indexation, National Library of the Chilean Congress, Radio Moscú.

Lista de Figuras

Lista de Tablas

1. Introducción

El problema surge dado una colección de audios que se encuentran dentro de la Biblioteca del Congreso Nacional (BCN) [1], estos conservan un breve lapso de la historia de nuestro país a través del programa radial *Escucha Chile*, emitido por Radio Moscú. Esta colección contiene una recopilación de los programas transmitidos por la emisora donde están registradas las voces de los locutores que apuntaban a una audiencia compuesta principalmente por chilenos radicados en el extranjero y por supuesto a personas en Chile.

Estos audios tienen origen en la donación de las cintas magnetofónicas que hizo el señor Dante Melgarejo a la BCN [2]. En estas pistas se abarca el período comprendido entre los años 1975 y 1990, por lo que relata y recoge testimonios de quienes vivieron durante este periodo histórico de Chile. Las grabaciones tienen ruido, interferencia y una baja calidad en comparación a lo percibido en una radio convencional, además actualmente estas grabaciones han sido escasamente estudiadas y pobremente analizadas.

En este contexto, dado lo relevante de este material desde el punto de vista histórico, resulta de interés para la BCN contar con estas grabaciones en formato texto, tanto para hacerlos públicos a la ciudadanía como también para posteriormente construir nuevos productos y servicios de interés ciudadano y parlamentario. En ese sentido, el presente trabajo pretende desarrollar a través de procesamiento automatizado, la transformación de pistas de audio con altos niveles de ruido e interferencia, en texto que resulte fidedigno al contenido, permitiendo posteriormente la indexación y consultas sobre los datos obtenidos.

En este documento se da a conocer el contexto del problema, una explicación de los métodos a utilizar en cada una de las etapas del proyecto y el modelo de la solución propuesta. También se incluyen experimentos, sus resultados y un análisis de los datos obtenidos. Finalmente se ofrecen proyecciones a futuro con una serie de sugerencias y observaciones para poder continuar con el proyecto de mejor manera y por último se entregan las conclusiones obtenidas.

2. Definición de Objetivos

Considerando las necesidades de la BCN, es posible señalar que los objetivos que se han propuestos son:

2.1. Objetivo General

Realizar un procesamiento de las grabaciones de la Radio Moscú que permita el mejoramiento del audio, reconocimiento del habla e indexación de palabras.

2.2. Objetivos Específicos

- Mejorar la calidad del audio de las grabaciones, ya que actualmente cuentan con grandes distorsiones y ruidos que entorpecen la interpretación de los diálogos.
- Realizar una transcripción automática de la serie de audios ya mencionada de forma tal que permita una mayor comprensión y documentación de los diálogos que contienen las pistas.
- Generar un diccionario de palabras a partir de la transcripción ya obtenida y de esta manera, realizar una serie de registros con la finalidad de hacer un análisis de los temas que se tratan en estas grabaciones.

3. Contextualización del Problema e Investigaciones Previas

Resulta fundamental conocer y tener nociones de investigaciones previas que tengan relación con este proyecto, ya que de esta manera se pueden obtener directrices de cómo desarrollar y materializar este trabajo, es por esto que se expondrán diversos estudios de variados investigadores para conocer y saber la manera en que se pueden cumplir los aspectos más relevantes de este proyecto, mejorar el audio y realizar una transcripción de este.

3.1. Mejoramiento del Audio

El mejoramiento del habla (*Speech Enhancement*) tiene la finalidad de hacer más legible las pistas de sonido mejorando la calidad del audio suprimiendo otros sonidos indeseables que son considerados ruido. Este problema ha sido una necesidad que ha abarcado una gran cantidad de estudios y técnicas desde los años 70's, principalmente debido a la variedad de problemas que se solucionan al obtener unos buenos resultados. Desde los primeros intentos que se centran en técnicas como el filtrado *Wiener* [3] en 1979, pasando por décadas posteriores, hasta llegar a técnicas más sofisticadas con el aprovechamiento del aprendizaje de las Redes Neuronales y el *Deep learning*.

La mala calidad de audio de las grabaciones obtenidas de la Radio Moscú, en donde existe un ruido constante principalmente debido a las capacidades de radiotransmisión de la época, y las típicas interferencias que ocurren al sintonizar señales de radio analógicas. Es por ello que resulta fundamental realizar una eficiente eliminación de ruido de las grabaciones no solo por el hecho de ser uno los objetivos del presente proyecto, sino que también permitirá que se puedan realizar los posteriores trabajos como la correcta transcripción, indexación y análisis de las grabaciones.

Para realizar esta tarea nos centraremos en analizar principalmente dos métodos, debido a que estos cuentan con su repositorio en *GitHub*[1][2] en el cual se facilita los métodos ya codificados descritos en sus respectivas investigaciones, además de los *datasets* que se utilizan. Uno de ellos es *Wavenet* el cual es una Red Neuronal para la sintetización del habla y *SEGAN* el que mediante Redes Antagónicas Generativas *GAN* logra el mejoramiento de los

[1] `https://github.com/drethage/speech-denoising-wavenet`
[2] `https://github.com/santi-pdp/segan`

audios en cuestión.

Wavenet es una Red Neuronal Convolucional (CNN) desarrollada por *Google DeepMind* en 2016 capaz de sintetizar audio en bruto, puede generar voces humanas que suenan reales, además tiene el potencial de imitar cualquier tipo de sonido, es decir, copiar el sonido de voces humanas o hasta música. Es debido a que la eliminación de ruido comparte propiedades con la síntesis de habla, que se adopta *Wavenet* para el mejoramiento del habla [4].

Por otro lado, *SEGAN* (*Speech Enhancement Generative Adversarial Network*) [5] es un método para el mejoramiento del habla que hace uso de Redes Generativas Antagónicas. Esta red, por sus siglas en inglés "GAN", es un método que utiliza una red generativa "G" que crea muestras imitando la distribución de datos originales "Z". Por otro lado, existe otra red "D" que discrimina entre las muestras que son reales, es decir, las que provienen de "Z" y las falsas, que han sido generadas por "G". El componente antagónico viene dado ya que "G" intenta engañar a "D", ajustando sus parámetros para que "D" clasifique como reales los datos provenientes de "G". En el caso del mejoramiento del habla, la red "G" es quien realiza la mejora, la entrada es la señal de voz con ruido y su salida es la versión mejorada. Una característica importante de "G" es su estructura *end-to-end*, lo que permite procesar audio puro en una frecuencia de 16kHz, eliminando cualquier transformación intermedia para extraer características acústicas. La red "D" se encarga de transmitir la información que considera verdadera y falsa, por esta razón, "G" puede ajustar ligeramente su onda de salida, eliminando el ruido de las señales, ya que estas son consideradas como falsas.

Una reciente investigación [6] resulta aplicable para esta problemática, se trata de una técnica que emplea *deep learning* de manera *end-to-end* (SPDWDFL). Con este método se contempla que el audio de entrada sea una pista con interferencia o elementos que entorpezcan la claridad de la voz. En esta investigación se utiliza una red entrenada completamente convolucional (*fully-convolutional context aggregation network*) y además usando una pérdida profunda de características, la cual se fundamenta en una comparación de las activaciones de características internas en una red distinta para luego así detectar entornos ruidosos que dificulten o se sobrepongan a las voces presentes.

3.2. Reconocimiento Automático del Habla

El reconocimiento automático del habla (RAH), o por sus siglas en inglés (ASR), es un proceso que ha sido ampliamente investigado en los últimos 20 años, desarrollando así, un gran repertorio de distintas técnicas capaces de interpretar las ondas sonoras del habla humana y transcribirlas a texto. La mayoría de estas tecnologías hacen uso de distintos tipos de técnicas de *Deep Learning*. También existen actualmente algunos productos capaces de reconocer automáticamente el habla, no obstante, algunas de estas soluciones resultan pagadas y además son incapaces de interpretar dialectos y acentos del habla hispana, lo cual resulta sumamente importante, ya que para este proyecto es necesario contar con una herramienta que interprete obligatoriamente el lenguaje español. Además considerando nuestra problemática y la estructura de las grabaciones de la Radio Moscú, será necesario contar con un reconocedor de habla *(Speech Recognition)* basado en habla continua *(Continuous Speech)* [7].

Se procedió a estudiar trabajos de esta índole de diversos investigadores, los cuales serán detallados a continuación. También se darán a conocer detalles de las distintas propuestas de los investigadores, así como de los productos que actualmente ofrece el mercado. Ninguna de las herramientas investigadas proponen u otorgan un porcentaje de éxito del 100 % de reconocimiento del habla, sin embargo, se conocen técnicas empleadas para perfeccionar y obtener un buen porcentaje de acierto.

Los sistemas de RAH básicamente buscan resolver un problema estadístico en el que se desea conocer la señal interpretada de la onda acústica entrante. Para comprender la manera en que un RAH funciona se incluye la figura 1 la cual expone, a modo general, la manera en que estos sistemas trabajan y sus principales componentes.

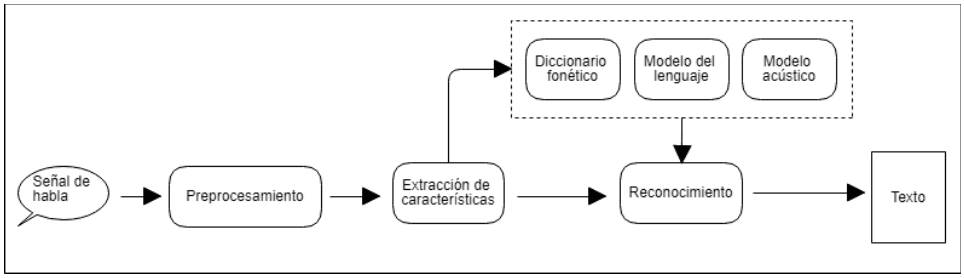

Figura 1: Estructura Clásica de un Sistema de RAH

Explicando brevemente la manera en que un sistema de RAH funciona resulta relevante señalar que la entrada (o *input*), es decir, la señal de habla (o *Speech Signal*) serán las ondas sonoras del hablante. Luego en el preprocesamiento (o *Pre-processing*) se comienzan a realizar adecuaciones respecto al sonido en bruto para poder trabajar sobre este. Estos procesos son principalmente el aislamiento de ruidos de fondo o algunos tipos de interferencia. En la extracción de características (o *Feature Extraction*) se obtienen datos como la modulación, pronunciación e identificación del hablante, así como también la duración o longitud de la onda entrante. El diccionario fonético es el vocabulario de palabras junto con la división fonética de cada una de ellas. El modelo del lenguaje aporta la distribución de probabilidad de la palabra y el orden de la misma en el sistema. El modelo acústico representa la distribución de probabilidad de los fonemas en la señal de audio [8]. El reconocimiento (o *Recogniser*)es el proceso en el cual se procede a evaluar y juzgar las ondas sonoras entrantes para asociarlos a una combinación de palabras acorde a los conocimientos que tenga el sistema (diccionario fonético, modelo del lenguaje y modelo acústico). Finalmente la salida (o *output*) es el resultado que arroja el sistema, es decir, la transcripción del audio ingresado a texto.

La gran diferencia entre los diversos métodos empleados para el RAH radican en la manera en que se implementa el sistema de reconocimiento, los cuales principalmente son mediante el uso de técnicas de *Deep Learning* [9], y la manera en que se conforman el diccionario fonético, el modelo del lenguaje y el modelo acústico, ya que, es finalmente este trío el que más repercute en el entrenamiento y finalmente en el índice de acierto del sistema RAH.

En el ámbito de las investigaciones de RAH mediante el empleo de *Deep Learning* se destaca el uso de: Redes Neuronales Recurrentes *(Recurrent Neural Networks (RNNs))* [10], Redes Neuronales Convolucionales *(Convolutional Neural Networks (CNNs))* [11] y Redes Neuronales Profundas *(Deep Neural Networks (DNNs))* [12]. Todos estos trabajos afrontan de distinta forma la manera en que se reconoce el habla y han demostrado ser métodos complejos, con una gran cantidad de horas empleadas en desarrollo y hasta 2.000 horas en la etapa de entrenamiento [12], estos son puntos negativos que hacen sugerir utilizar productos ya elaborados.

Los servicios que hoy en día ofrecen un reconocimiento automático del habla son variados, tanto en la forma en que lo realizan como en sus opciones

y limitantes. Se procederá a detallar aspectos relevantes de los principales productos de RAH.

- **IBM Watson Speech to Text**[3]: Ofrece RAH en habla hispana y demuestra tener un buen comportamiento en diálogos sencillos y además es capaz de identificar a distintos participantes.
- **Amazon Transcribe**[4]: Soporta varios idiomas, incluido el español. Es capaz de realizar marcas temporales, lo que facilita localizar de manera simple la palabra en la grabación junto a su respectivo posición el texto (una vez transcrito). Próximamente contará con la opción de identificar a los hablantes. Además se señala la posibilidad de poder trabajar con audios de baja calidad y también es capaz de detectar pausas y puntuación de manera automática.
- **API Speech de Google Cloud**[5]: Soporta el español (incluso el español chileno). Es capaz de detectar palabras claves y funcionar en entornos ruidosos.
- **Bing Speech API**[6]: Este producto de *Microsoft* es capaz de funcionar con el idioma español y además cuenta con un servicio extra, en versión preliminar, para autenticar a los hablantes (*Speaker Recognition API*).
- **CMU Sphinx**[7]: Este sistema es de código abierto por lo que no tiene precios asociados y su repositorio de *GitHub*[8] es público. Posee una comunidad activa y además es capaz de funcionar en variados idiomas incluyendo el español. *CMU Sphinx* está diseñado para plataformas de bajos recursos.

Los anteriormente mencionados son algunos de los servicios más populares en este rubro, exceptuando *CMU Sphinx* (que es de código abierto), todos se deben adquirir por medio de suscripciones, aunque también cuentan con servicios de prueba. A continuación se incluye la tabla 1, la cual es una comparativa de las características más relevantes entre los productos anteriormente mencionados. No se incluirá *CMU Sphinx*, ya que actualmente no es una herramienta que instantáneamente se podría utilizar.

* El servicio de identificación de hablantes es un servicio extra.

[3] `https://www.ibm.com/watson/services/speech-to-text/`
[4] `https://aws.amazon.com/es/transcribe/`
[5] `https://cloud.google.com/speech/?hl=es`
[6] `https://azure.microsoft.com/es-es/services/cognitive-services/speech/`
[7] `https://cmusphinx.github.io`
[8] `https://github.com/cmusphinx`

	Identificación de hablantes	Soporta entornos ruidosos	Servicio de paga	Servicio gratuito
IBM Watson Speech to Text	Sí	No especifica	Desde 0,02USD por minuto	100 minutos por mes
Amazon Transcribe	Próximamente	Sí	0,0004USD por segundo	60 minutos por mes
API Speech Google Cloud	No	Sí	0,006USD cada 15 segundos (máximo 1 millón de minutos)	60 minutos por mes
Bing Speech API	Próximamente*	No especifica	4USD por 1.000 transacciones	5.000 transacciones por mes

Tabla 1: Comparación de Servicios de RAH

3.3. Indexación de Documentos

El último objetivo encomendado en este trabajo consiste en generar estadísticas y consultas en los textos una vez que estén transcritos. Se procedió a investigar la manera en que librerías o grandes coleccionadores de documentos de texto trabajan y principalmente se estudió la manera en que estas afrontan búsquedas de frases o palabras claves. De esta manera se descubrió *Invenio*[9], un software dedicado al rubro de los repositorios bibliográficos en línea que actualmente sirve de plataforma en importantes bibliotecas o importantes referentes bibliográficos del mundo, como lo es CERN[10](*European Organization for Nuclear Research*). Considerando las investigaciones que esta organización ha realizado [13] es posible señalar que *Solr*[11] ha demostrado ser altamente eficiente en lo que respecta a búsquedas de palabras en entornos donde se tienen más de 656.000 registros, superando a otros productos de similar índole como lo es *Xapian*[12]. Por otro lado, el servicio de *Amazon Transcribe* puede ser asociado con otro servicio de esta plataforma, *Amazon Elasticsearch Service*[13], el cual suple la necesidad de búsqueda de texto completo.

[9] `http://invenio-software.org/`
[10] `http://cds.cern.ch/`
[11] `http://lucene.apache.org/solr/`
[12] `https://xapian.org/`
[13] `https://aws.amazon.com/es/elasticsearch-service/`

4. Modelo de Solución Propuesta

En la figura 2 se expone el modelo de solución propuesto para el presente trabajo, englobando todos los objetivos del proyecto de manera secuencial y lógica para el tratamiento del problema.

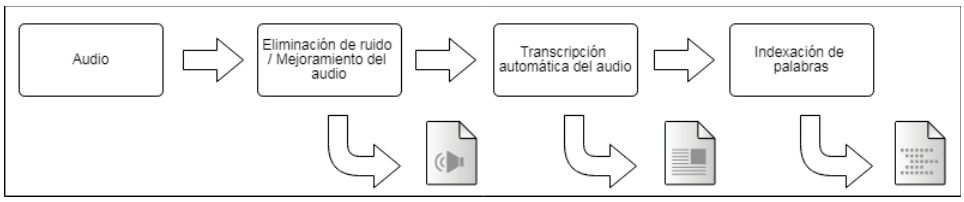

Figura 2: Modelo de Solución Propuesta

Para facilitar la comprensión de las etapas propuestas en la figura 2, se procede a detallar cada una de ellas:

1. **Audio:** El elemento de entrada al sistema será una de las pistas de grabación (en bruto) de la Radio Moscú.

2. **Mejoramiento del audio:** En esta etapa se procederá a tratar cada una de las pistas acústicas de la Radio Moscú con la intención de poder aislar los ruidos e interferencias que afecten la comprensión de los diálogos. Para luego de esta manera obtener un archivo de audio con una buena calidad y que se entendible.

3. **Transcripción automática del audio:** Una vez obtenido un audio ya refinado, este será procesado por una herramienta de RAH, de manera tal que se obtenga un documento de texto con los diálogos transcritos.

4. **Indexación de palabras:** Finalmente, dado que ya se cuenta con una transcripción de los audios, se realizará un indexado de palabras, para así facilitar búsquedas de palabras en específico.

4.1. Principales Técnicas y Herramientas

Considerando lo investigado es posible señalar que ya se cuentan con algunas nociones e intereses de utilizar algunas herramientas o tecnologías en las diversas etapas que componen este proyecto. A continuación se expondrán las técnicas, herramientas o tecnologías tentativas a utilizar en cada una de las fases del desarrollo.

Mejoramiento del audio	Wavenet
	SEGAN
Transcripción	Amazon Transcribe
	API Speech de Google Cloud
Indexación de palabras	Invenio
	Solr
	Amazon Elasticsearch

Tabla 2: Herramientas, Técnicas y Tecnologías a Utilizar

Dado que es necesario evaluar la efectividad de los métodos propuestos, es que será necesario buscar métricas que permitan medir el impacto que tiene en los otros procesos. Es por ello, que se utilizará el reconocimiento automático del habla o transcripciones obtenidas como medidor de calidad del audio. Contrastando las transcripciones obtenidas con los audios en bruto y las que se le hayan realizado una eliminación de ruido.

Se espera que las métricas enfocadas en las transcripciones puedan medir el impacto con respecto a la eliminación del ruido. Esta forma de determinar la calidad de los audios se hace debido a la falta de un conjunto de datos compuesto por audios limpios que permita compararlos con los resultados obtenidos en esta etapa.

5. Transcripciones con Audio Sin Procesar

Resulta relevante conocer el comportamiento del reconocimiento automático del habla, valiéndonos de los servicios proporcionados por las APIs ya antes mencionadas. Es por ello, que es necesario realizar un experimento preliminar que permita analizar y comparar las capacidades de cada una de las APIs, permitiendo obtener los fundamentos necesarios para escoger la API que más se adecúe al proyecto. Se procederá a ejecutar los servicios de reconocimiento automático y transcripción del habla con fragmentos de distintos audios, posteriormente se contrastarán los resultados obtenidos.

El experimento consiste en tomar 12 muestras distintas de los audios de Radio Moscú con una duración de 2 minutos cada uno, además se añaden audios que no se relacionan con la Radio Moscú pero permiten evaluar a los servicios con audios que cuentan con una mejor calidad, cada uno tiene una duración de 1 minuto. Ninguno de los audios anteriormente mencionados cuentan con un preprocesamiento de eliminación de ruido, por lo que nos permite evaluar a los servicios en condiciones más extremas de las que se pretende ejecutar el proyecto. Para realizar este experimento solo se utilizarán las APIs *Speech de Google Cloud* y *Amazon Transcribe*, debido a que se estimó que las otras APIs como *IBM Watson* no cumplían con los requerimientos mínimos en cuanto a calidad para ser usadas.

Se utilizaron las APIs de los servicios otorgados por *Google Cloud* y *Amazon*, esto proporcionó una gran ventaja debido a que permite prescindir en esta primera etapa de un mayor poder de procesamiento. Para que las APIs pudieran procesar los audios fue necesario cambiar el formato de .mp3 a .flac en el caso de Google y .wav para Amazon, debido a que este era el formato permitido por estos servicios, además fue necesario almacenarlos en sus respectivos storage. Ambas funcionaban bajo la lógica de API REST por lo que mediante consultas permiten obtener resultados a los pocos minutos en el caso de estas muestras.

5.1. Métodos Empleados Para la Obtención de Métricas

Para poder conocer la efectividad de las transcripciones obtenidas de las APIs se investigaron diversos métodos para evaluar la similitud entre textos. Para obtener las mediciones se realizaron parejas compuestas por la transcripción manual con la transcripción automática de Google o Amazon de cada una de las respectivas pistas utilizadas en la muestra. Aplicando

los métodos de: la distancia de Levenshtein y el porcentaje de similitud semántica se obtuvo valores que simbolizan la similitud entre la transcripción esperada (transcripción manual) y la transcripción de *Speech de Google Cloud* y *Amazon Transcribe*.

5.1.1. Distancia de Levenshtein

La distancia de edición, distancia entre palabras o también llamada distancia de Levenshtein es un algoritmo que retorna la cantidad mínima de operaciones necesarias para modificar una cadena de texto, de manera tal que resulte idéntica a otra. Estas operaciones pueden ser sustituir, añadir o eliminar un carácter. Gracias al valor que este algoritmo retorna es posible conocer la similitud o cercanía entre cadenas, ya que a medida que este valor esté más cercano a 0 es posible señalar que existe una mayor similitud entre los textos.

Por ejemplo, la distancia de Levenshtein entre las cadenas "Copa" y "Mundo" resulta ser de 5, es decir, aplicando 5 operaciones de edición "Copa" se transformaría a "Mundo", estos mismos 5 pasos serían necesarios para hacer que "Mundo" sea igual a "Copa". Para ilustrar este ejemplo se incluye la siguiente tabla, la cual detalla y expone el funcionamiento de la distancia de Levenshtein.

		C	O	P	A
	0	1	2	3	4
M	1	1	2	3	4
U	2	2	2	3	4
N	3	3	3	3	4
D	4	4	4	4	4
O	5	5	4	4	5

Tabla 3: Ejemplo Distancia de Levenshtein

En la celda 1,1 tenemos las cadenas "C" y "M" las cuales no son iguales, entonces se debe aplicar una operación, que en este caso sería sustituir. Luego en la celda 1,2 se tienen "C" y "MU" y es necesario aplicar dos operaciones. Es así como este algoritmo trabaja, y una vez obtenido todos los pasos necesarios para hacer que las cadenas sean idénticas, se tiene la distancia entre ellas.

En el caso de esta investigación, la prueba realizada con este método consistió en comparar la transcripción manual con la automática.

5.1.2. Porcentaje de Similitud Semántica

Tal como su nombre lo menciona, este método también mide la similitud entre cadenas de texto, este es calculado gracias a la librería de Python *SpaCY*[14]. Por medio de esta librería es posible vectorizar las palabras se genera un conjunto de palabras relacionadas semánticamente para luego así evaluar la similitud entre las oraciones que se estén comparando. La explicación matemática se refleja en la siguiente ecuación:

$$similitud = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{1}$$

En la que tanto "A" como "B" son vectores.

Con esta métrica es posible señalar que mientras el resultado sea más próximo a 100 %, los textos comparados poseen una gran similitud semántica y por lo tanto ambos tratan de lo mismo. En caso contrario, si el resultado es una cifra cercana al 0 %, entonces se trataría de un par de textos que semánticamente no tienen nada en común.

5.1.3. Tasa de Error por Palabra (WER)

Word error rate o en español tasa de error por palabra es una extensión de lo que es la distancia de Levenshtein. Ésta es utilizada para medir al desempeño del reconocimiento automático del habla y se calcula de la siguiente manera:

$$WER = \frac{S + D + I}{N} \tag{2}$$

- "S" es el número de sustituciones
- "D" es el número de eliminaciones
- "I" es el número de agregaciones
- "N" se calcula de la siguiente forma:

$$N = S + D + C \tag{3}$$

- "C" es el número de palabras correctas

[14] `https://spacy.io/`

El resultado de WER nos da nociones para comprender que tan distintos son dos textos, en la que a mayor porcentaje, mayor será la diferencia entre ambos.

5.2. Resultados Obtenidos

Como se mencionó anteriormente, existen dos grupos de muestras en el experimento, una de ellas fue medir la similitud entre un audio transcrito manualmente con uno transcrito automáticamente, con un audio sin mayores complicaciones en cuanto a la calidad y el entendimiento de los diálogos. La segunda serie de muestras consistió en conocer el grado de éxito de la transcripción automática, mediante la similitud con el texto que se esperaba obtener (transcripción manual) con las pistas del programa radial Escucha Chile de la Radio Moscú.

Para poder evaluar de mejor manera el desempeño de ambas herramientas de transcripción, los textos fueron preprocesados de manera tal que estén normalizados para así poder juzgar con una mayor igualdad de condiciones los textos. Además se considerará el tiempo que cada una de las plataformas tarda desde que se recibe la petición de transcribir hasta que se retorna el texto correspondiente a la transcripción del audio.

5.2.1. Resultados en Audios de Prueba

Las tablas 4 y 5 exponen los resultados obtenidos empleando la dinámica anteriormente señalada en una colección de 4 pistas de audio, en condiciones normales y sin mayores complicaciones para poder captar los diálogos en cada una de las grabaciones. Además se muestran datos relevantes como el promedio que nos permite delinear un limite base de aceptación, la varianza la cual muestra que tan dispersos se encuentran los datos y la covarianza para analizar la relación entre las métricas utilizadas.

Es posible afirmar que las APIs tuvieron un comportamiento similar, en el que los porcentajes de similitud por cada pista estuvieron muy cercanos entre si. Podemos apreciar que en promedio el servicio de Amazon muestra un mejor rendimiento y hay mayores varianzas en los resultados entregados por la API de Google, principalmente debido a que estos muestran una mayor inestabilidad en sus resultados, también podemos ver que la covarianza es negativa, esto es debido a que la similitud y la distancia se encuentran inversamente correlacionados, es decir, si el porcentaje de similitud crece

la distancia disminuye. Finalmente el servicio entregado por Amazon es la opción más conveniente para el reconocimiento y transcripción automática, principalmente debido a que muestra resultados con mayor similitud y una menor dispersión entre los resultados que la API de Google.

Track	Similitud	Distancia	Tiempo de ejecución
1	97,5 %	104	2:14
2	95,6 %	163	2:21
3	98,2 %	51	2:22
4	96,1 %	239	3:02
Promedio	96,9 %	139,25	2:30
Varianza	1,5	6514,9	6.319×10^{-8}
Covarianza*	-61,6		

Tabla 4: Resultados Amazon Audios de Prueba

* Considerando similitud-distancia

Track	Similitud	Distancia	Tiempo de ejecución
1	98,0 %	101	0:16
2	91,8 %	373	0:14
3	99,6 %	16	0:18
4	94,7 %	234	0:13
Promedio	96 %	181	0:15
Varianza	12.1	24432,7	6.5×10^{-10}
Covarianza*	-407,3		

Tabla 5: Resultados Google Audios de Prueba

* Considerando similitud-distancia

Para facilitar la interpretación de estos datos, se incluyen las siguientes gráficas, en estas es posible exponer lo parejo que resultó el funcionamiento de *Speech de Google Cloud* y *Amazon Transcribe*. En la primera gráfica se puede observar que los valores estuvieron muy parejos y es posible percibir que el promedio entre los valores de similitud semántica de ambas APIs fue de 96.45 %, el cual es un valor aceptable y que vislumbra que efectivamente hubo similitud entre la transcripción obtenida por las APIs y la transcripción manual, en otras palabras, ambos textos tratan de lo mismo. En la segunda gráfica se aprecian los valores de la distancia, en esta es posible señalar que

de igual manera existieron valores relativamente parejos entre ambas APIs y se destaca, además, que el promedio ponderado resultó ser aproximadamente 160.

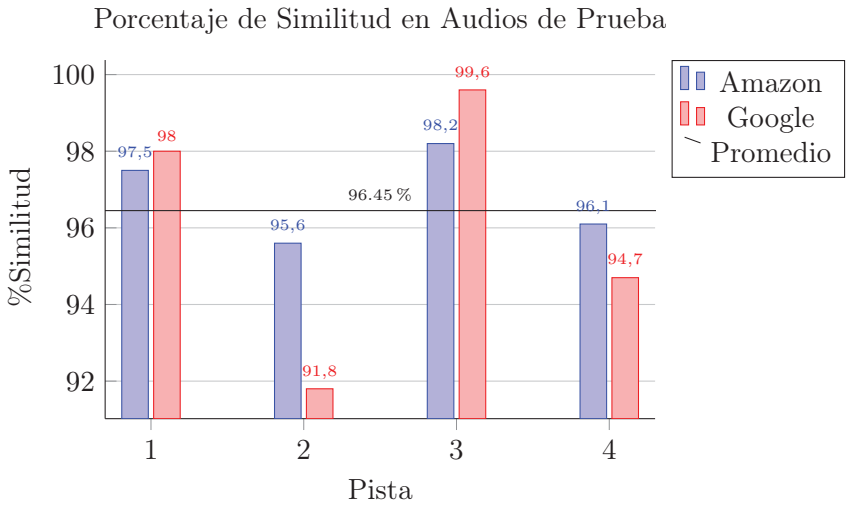

Figura 3: Porcentaje de Similitud en Audios de Prueba

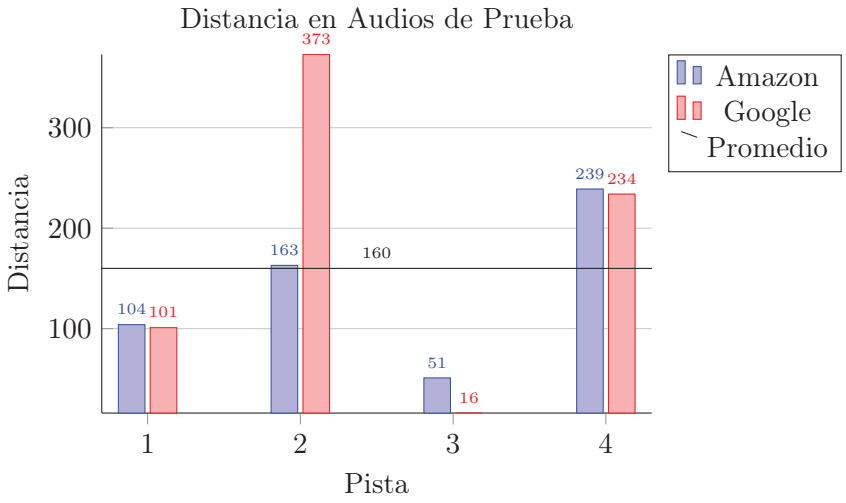

Figura 4: Distancia en Audios de Prueba

5.2.2. Resultados en Pistas de la Radio Moscú

Las tablas 6 y 7 contienen los resultados de similitud entre los servicios de Amazon y Google. A partir de estas tablas podemos apreciar que *Amazon Transcribe* tuvo un mejor comportamiento en lo que es transcribir audios con las características de las pistas de la Radio Moscú. De igual forma que en el caso anterior, se muestran valores significativos para el análisis de los resultados obtenidos.

En este caso la tendencia fue que la API de Amazon tuvo constantemente mayor efectividad en lo que es porcentaje de similitud y una menor distancia, estos valores son positivos y nos da a entender que a pesar de que existe un mejor desempeño del servicio *Amazon Transcribe* por sobre *Speech de Google Cloud*, si resulta necesario aplicar ciertas técnicas capaces de aislar el ruido y mejorar de alguna manera la calidad de estas pistas para hacerlas más sencillas de entender.

Track	Similitud	Distancia	Tiempo de ejecución
1	94,3 %	907	4:02
2	99,8 %	67	3:22
3	90,0 %	602	5:02
4	96,4 %	540	4:42
5	96,8 %	499	5:23
6	93,9 %	600	5:03
7	98,3 %	349	4:23
8	98,5 %	293	5:02
9	96,2 %	407	4:22
10	96,1 %	266	4:22
11	98,9 %	158	4:02
12	96,3 %	407	4:22
Promedio	96,3 %	424,6	4:17
Varianza	7,1	51050,8	2.634×10^{-7}
Covarianza*	-400,6		

Tabla 6: Resultados Amazon Pistas Radio Moscú

* Considerando similitud-distancia

Track	Similitud	Distancia	Tiempo de ejecución
1	92,7 %	1001	1:17
2	98,9 %	970	0:28
3	74,4 %	1251	1:57
4	98,6 %	586	1:25
5	77,1 %	1580	2:06
6	93,4 %	663	1:14
7	98,2 %	474	0:47
8	98,3 %	679	1:02
9	92,5 %	884	0:57
10	97,6 %	373	0:53
11	94,9 %	1092	1:17
12	93,7 %	706	1:13
Promedio	92,5 %	854,8	1:13
Varianza	67,9	119443,5	1.022×10^{-7}
Covarianza*	-2056,8		

Tabla 7: Resultados Google Pistas Radio Moscú

* Considerando similitud-distancia

Para poder representar estos valores, a continuación se mostrarán un par de gráficos, los cuales exponen una comparación entre las APIs y su desempeño en cuanto al porcentaje de similitud y distancia. En el primer gráfico se aprecia el liderazgo en efectividad que Amazon demostró, registrando una gran cantidad de pistas en las que se tuvo una efectividad por sobre la media ponderada de ambas APIs, la que tuvo un valor de 94,4 %. Por otro lado, en la segunda gráfica se ve que *Amazon Transcribe* fue de igual manera, más exitosa en acertar al texto esperado, ya que, en general las transcripciones de esta API tuvieron valores por bajo el promedio, que en esta ocasión fue de 639,75.

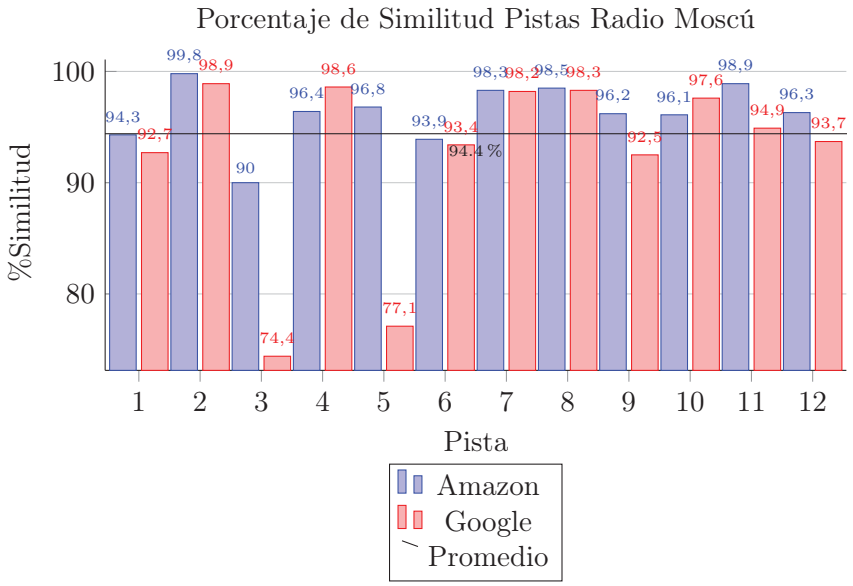

Figura 5: Porcentaje de Similitud Pistas Radio Moscú

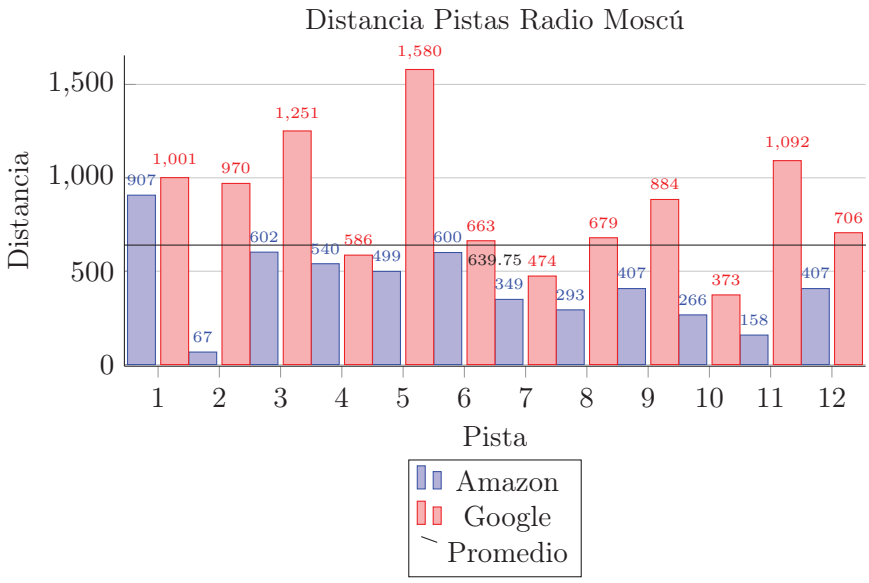

Figura 6: Distancia Pistas Radio Moscú

5.3. Conclusiones del Experimento

Dado los resultados obtenidos en el experimento preliminar para el reconocimiento y transcripción automática de texto, es posible llegar a ciertas conclusiones en cuanto a la prueba realizada. Se demuestra lo postulado inicialmente, que es necesario realizar un mejoramiento del audio o eliminación de ruido previo a la transcripción automática del audio, debido principalmente a que esto permitiría obtener mejores resultados que los obtenidos con el conjunto de muestras de Radio Moscú. Esto se determina, ya que los resultados obtenidos para el conjunto de pistas de los audios que eran considerados de una mejor calidad, tuvieron mejores resultados tanto en el porcentaje de similitud como en la distancia en comparación a los audios de Radio Moscú. Teniendo en el primer conjunto de muestras un porcentaje del 96,45 % y 160 de distancia en promedio, en cambio para el caso del conjunto de audios de la Radio Moscú se obtuvo un 94,4 % y una distancia 639.75 en promedio entre las dos APIs. Por lo que se concluye que los servicios actúan de una mejor forma con audios que se encuentran con una mejor calidad de voz y menor cantidad de ruido.

6. Pruebas de Mejoramiento del Audio

6.1. Desarrollo

Para realizar el mejoramiento del habla en de las pistas de audio en este experimento, se utiliza Wavenet principalmente debido a que es el más versátil para su funcionamiento en distintos tipos de hardware y se descarta SEGAN y SPDWDFL. Debido a que una de las grandes limitantes para realizar el experimento fue la capacidad de procesamiento de las máquinas, es por ello, que el modelo se ajustó a una menor capacidad de computo en desmedro de obtener mejores resultados. Para los entrenamientos se empleó la misma data que fue utilizada en los experimentos y evaluaciones realizas en la investigación de Wavenet.

Los entrenamientos se realizaron en la plataforma *Online Google Colab*, la cual cuenta con una GPU Tesla K80, aproximadamente 13GB de memoria RAM y una CPU Intel Xeon 2.20GHz.

El objetivo es medir la calidad del habla con el audio mejorado mediante el reconocimiento automático del habla. Para ello, se realizan transcripciones a los audios y se contrastan con las transcripciones realizadas a las mismas pistas sin que se les haya realizado una eliminación del ruido. Finalmente se analizan las métricas para observar el impacto que produjo el experimento.

6.2. Comparación de Resultados

A continuación se muestran las tablas 8 y 9 las cuales representan la distancia de Levenshtein y el porcentaje de similitud semántica de cada una de las 12 pistas de audios utilizadas en este experimento. Para poder exponer de manera más cómoda estos datos, las celdas de color verde simbolizan una mejoría con respecto a lo que fue la transcripción sin aplicar ninguna técnica de mejoramiento de audio.

En este primer experimento no se aprecia una mejoría uniforme en los audios, es decir, se observan leves mejoramientos en algunos audios mientras que otros se ve un aumento del ruido en la señal, sin embargo, otorga directrices en la manera en que se debe entrenar la máquina para así obtener mejores modelos y enfrentar de mejor manera los distintos factores que dificultan y entorpecen la claridad de los audios.

Track	Similitud	Distancia	Tiempo de ejecución
1	94,8 %	643	4:03
2	97,6 %	275	3:41
3	88,8 %	905	3:41
4	94,2 %	656	8:41
5	93,6 %	770	8:41
6	88,6 %	668	5:12
7	96 %	418	4:01
8	96,2 %	522	4:01
9	96,8 %	472	4:21
10	93 %	384	4:02
11	98,1 %	234	4:02
12	95,4 %	409	4:01
Promedio	94,4 %	529,7	4:52
Varianza	9,5 %	40953,0	1.60×10^{-6}
Covarianza*	-436,1		

Tabla 8: Resultados Mejoramiento del Audio Amazon

* Considerando similitud-distancia

En la tabla 8 se ve reflejado que para la pistas 1 y 9 se registró una mejora gracias al uso de las técnicas empleadas para el mejoramiento del audio, en el resto de la colección de audios se empeoraron las cifras de similitud semántica y distancia Levenshtein. También resulta posible apreciar que aumentó el tiempo empleado en la ejecución y que el promedio de las similitudes y distancias empeoraron.

Track	Similitud	Distancia	Tiempo de ejecución
1	96,1	993	0:54
2	98,2	1072	0:30
3	91	1166	0:54
4	94,6	1073	0:52
5	86,4	1519	0:51
6	91,4	978	1:13
7	95,7	839	0:45
8	96,9	926	0:49
9	92,2	1051	0:52
10	92,5	537	0:50
11	95,8	788	0:27
12	89	1068	0:53
Promedio	93,3	1000,8	0:49
Varianza	12,3	55080,9	1.86×10^{-8}
Covarianza*	-387,8		

Tabla 9: Resultados Mejoramiento del Audio Google

* Considerando similitud-distancia

Con la API de Google por medio de esta técnica empleada se mejoró el desempeño en 5 de las colecciones de audio (pista 1, 3, 5 y 11) y además se mejoró el promedio de similitud.

A continuación se muestran los gráficos que permiten visualizar los diferentes datos obtenidos, comparando cada uno de los resultados en cada métricas. De esta forma se puede observar las diferentes variaciones que se tienen con las muestras.

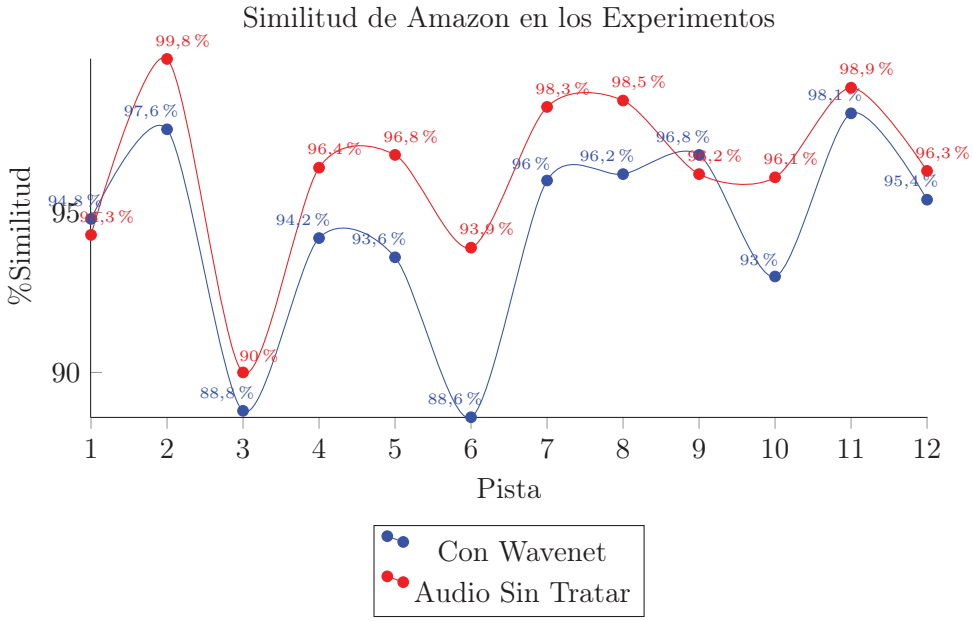

Figura 7: Similitud de Amazon en los Experimentos

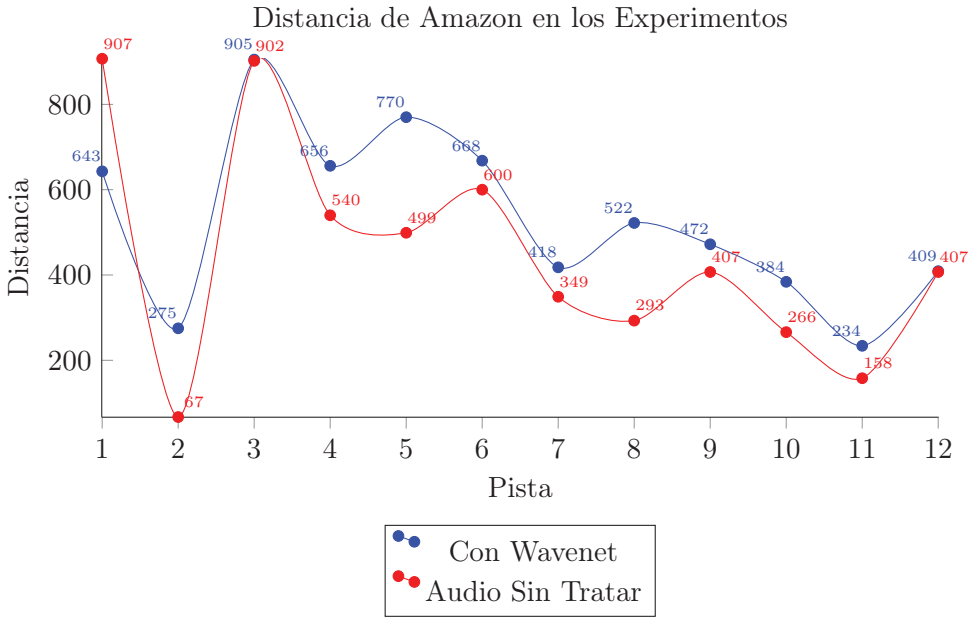

Figura 8: Distancia de Amazon en los Experimentos

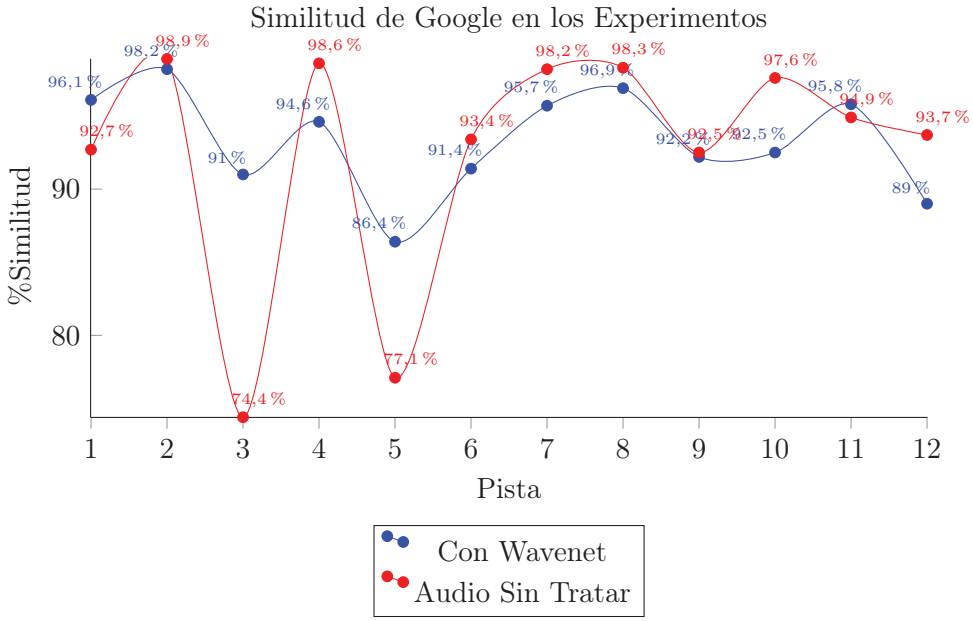

Figura 9: Similitud de Google en los Experimentos

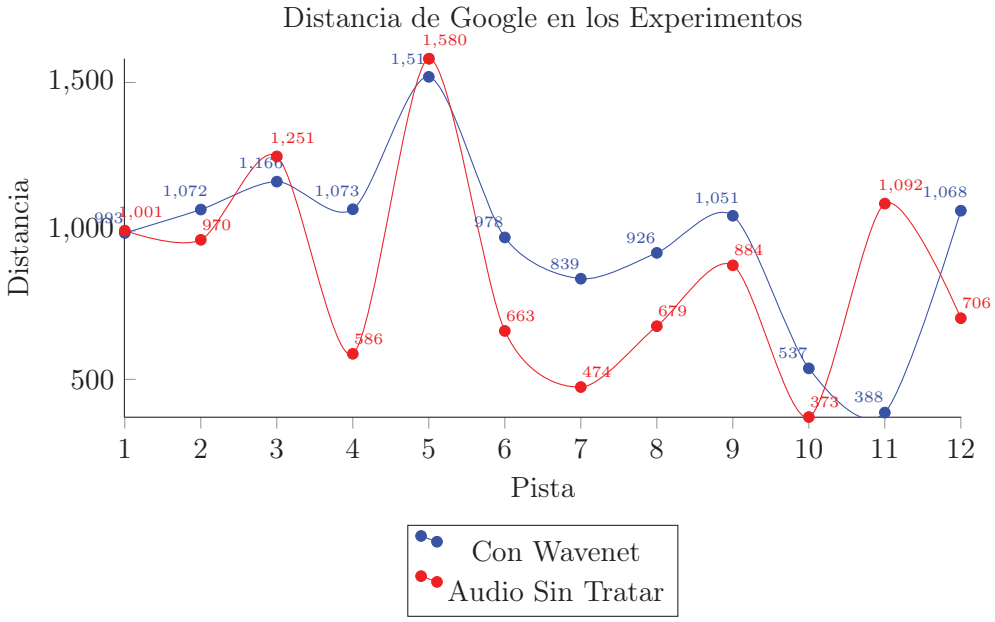

Figura 10: Distancia de Google en los Experimentos

6.3. Problemas Detectados

A continuación, se explican los diferentes problemas identificados durante el primer experimento de eliminación del ruido. Estos errores nos permiten entender la complejidad del problema que abarcan estas pistas de audio y nos posibilitan a encontrar soluciones.

Una gran limitante que ha sido anteriormente mencionada, es la falta de hardware idóneo para ejecutar los métodos propuestos. El hardware de acceso libre de Google Colab no fue suficiente para ejecutar los modelos propuestos. Debido a este problema no hemos sido capaces de implementar las otras soluciones aparte de Wavenet, además este último ha tenido que ser adaptado a las capacidades ya antes mencionadas. Es por ello, que tanto el modelo de SEGAN como SPDWDFL quedan finalmente descartados.

La colección de datos para el entrenamiento del modelo esta enfocada a ruidos de ambiente, es decir, esta muestra conversaciones con diferentes ruidos de fondo como restaurantes, vía publica, otras conversaciones de fondo, trafico, entre otros. Dada esta razón se estudia la creación de un nuevo dataset que cumpla con las necesidades de nuestro problema, ya que la Radio Moscú al ser un programa radial muestra un espectro de ruidos diferentes a los de ambientes más cotidianos. El estudió de diferentes señales de ruido coloreado puede satisfacer el problema dada su similitud en cuanto al sonido.

6.4. Ruido Coloreado en la Data de Entrenamiento

En la literatura aplicar distintos matices de ruido es un proceso recurrente para mejorar el rendimiento tanto para modelos de mejoramiento de audio, como para modelos encargados del reconocimiento automático del habla [14, 15, 16]. En esta ocasión se reconstruyeron los datos de entrenamiento agregando tres tipos de ruido, los cuales fueron: Ruido blanco, Ruido rosa y SSN (*speech-shaped noise*).

El ruido blanco, rosa y SSN aportan distintas distorsiones en la colección de audios de entrenamiento, buscando simular los ruidos presentes en algunos de los audios que se encuentran en la Radio Moscú. Es por esto que se realizó una serie de experimentos aplicando cada uno de estos tres tipos de ruido, para conocer el comportamiento que tenía el modelo para realizar un mejoramiento del audio más eficiente. Para poder facilitar el entendimiento

de la teoría de los ruidos, se incluye una breve descripción de los ruidos utilizados:

- **Ruido Blanco:** Es una señal, la cual tiene una cantidad de energía monótona en todas las bandas de frecuencia.
- **Ruido Rosa:** Es una variante de lo que es el ruido blanco, con la diferencia que este tipo de ruido remueve una mayor cantidad de energía a medida que la frecuencia aumenta.
- **SSN:** O *Speech shaped noise* se define como un ruido azaroso que tiene una duración similar a la prolongación de la voz contenida en el audio.

Se ejecutaron tres experimentos con cada uno de los ruidos mencionados anteriormente. A continuación se muestran los resultados obtenidos en específico de cada tipo de ruido empleado.

6.4.1. Experimento Ruido Blanco

Para mostrar los resultados obtenidos al aplicar ruido blanco en los datos de entrenamientos se expone las siguientes tablas en razón a los resultados obtenidos en la API de Amazon y Google.

AMAZON			
Track	Distancia	WER	Tiempo de ejecución
1	801	68,692 %	0:02:41
2	216	27,338 %	0:03:41
3	1151	92,079 %	0:02:41
4	768	70,175 %	0:03:41
5	767	76,423 %	0:06:01
6	739	81,429 %	0:04:21
7	431	46,975 %	0:04:01
8	602	57,71 %	0:03:22
9	476	50,588 %	0:04:02
10	395	59,487 %	0:04:01
11	314	35,178 %	0:03:42
12	476	56,744 %	0:03:21
Promedio	594,7	0,6	0:03:48
Varianza	68153,5	0,0388	3.63×10^{-7}

Tabla 10: Experimento Amazon con Ruido Blanco

GOOGLE			
Track	Distancia	WER	Tiempo de ejecución
1	1040	83,178 %	0:01:24
2	867	52,878 %	0:00:37
3	1099	90,099 %	0:01:08
4	1003	76,316 %	0:00:54
5	1458	92,276 %	0:01:09
6	947	80,476 %	0:01:38
7	691	50,178 %	0:00:47
8	887	61,660 %	0:00:54
9	962	68,235 %	0:01:06
10	449	46,154 %	0:00:58
11	775	53,755, %	0:00:35
12	1004	80,465 %	0:01:01
Promedio	931,8	0,7	0:01:01
Varianza	59077,1	0,0265	4.37×10^{-8}

Tabla 11: Experimento Google con Ruido Blanco

6.4.2. Experimento Ruido Rosa

Para mostrar los resultados obtenidos al aplicar ruido rosa en los datos de entrenamientos se expone las siguientes tablas en razón a los resultados obtenidos en la API de Amazon y Google.

AMAZON			
Track	Distancia	WER	Tiempo de ejecución
1	767	70,561 %	0:02:41
2	117	17,266 %	0:03:41
3	1071	90,594 %	0:03:02
4	763	68,421 %	0:03:43
5	648	72,764 %	0:06:01
6	645	72,764 %	0:04:21
7	377	42,705 %	0:04:21
8	501	48,221 %	0:03:41
9	448	49,804 %	0:04:02
10	277	47,692 %	0:04:01
11	251	30,040 %	0:03:41
12	383	54,884 %	0:03:42
Promedio	520,7	0,6	0:03:55
Varianza	72204,1	0,0422	3.23×10^{-7}

Tabla 12: Experimento Amazon con Ruido Rosa

GOOGLE			
Track	Distancia	WER	Tiempo de ejecución
1	973	76,168 %	0:02:08
2	908	54,317, %	0:00:33
3	1049	86,634 %	0:01:40
4	950	71,930 %	0:00:51
5	1471	93,089 %	0:01:01
6	1011	85,238 %	0:01:20
7	730	51,601 %	0:00:43
8	1036	74,308 %	0:00:57
9	1068	77,65 %	0:00:59
10	413	44,103 %	0:00:45
11	703	49,802 %	0:00:38
12	910	74,419 %	0:01:12
Promedio	935,2	0,7	0:01:04
Varianza	64119,4	0,0258	1.02×10^{-7}

Tabla 13: Experimento Google con Ruido Rosa

6.4.3. Experimento SSN

Para mostrar los resultados obtenidos al aplicar ruido SSN en los datos de entrenamientos se expone las siguientes tablas en razón a los resultados obtenidos en la API de Amazon y Google.

AMAZON			
Track	Distancia	WER	Tiempo de ejecución
1	943	81,308 %	0:02:42
2	381	41,007 %	0:04:02
3	1098	92,574 %	0:03:01
4	670	67,982 %	0:04:01
5	594	70,732 %	0:05:41
6	666	78,095 %	0:04:21
7	372	43,060 %	0:04:01
8	482	48,221 %	0:04:01
9	404	45,882 %	0:04:22
10	322	51,80 %	0:03:41
11	164	23,320 %	0:03:41
12	422	53,953 %	0:03:41
Promedio	543,2	0,6	0:03:56
Varianza	71594,0	0,0401	2.63×10^{-7}

Tabla 14: Experimento Amazon con SSN

GOOGLE			
Track	Distancia	WER	Tiempo de ejecución
1	1183	93,458 %	0:00:58
2	1047	62,950 %	0:00:47
3	1109	91,584 %	0:01:18
4	1091	82,895 %	0:01:15
5	1461	91,870 %	0:01:04
6	1030	86,667 %	0:01:33
7	796	56,228 %	0:00:52
8	962	65,613 %	0:01:03
9	1032	73,333 %	0:00:51
10	514	53,846 %	0:00:46
11	743	51,383 %	0:00:32
12	976	80,000 %	0:00:45
Promedio	995,3	0,7	0:00:59
Varianza	55993,2	0,0246	3.85×10^{-8}

Tabla 15: Experimento Google con SSN

6.4.4. Resultados Generales

Evaluando estos tres tipos de ruidos aplicados a la data de entrenamiento se pudo apreciar el comportamiento del modelo registrando los resultados que se muestran a continuación.

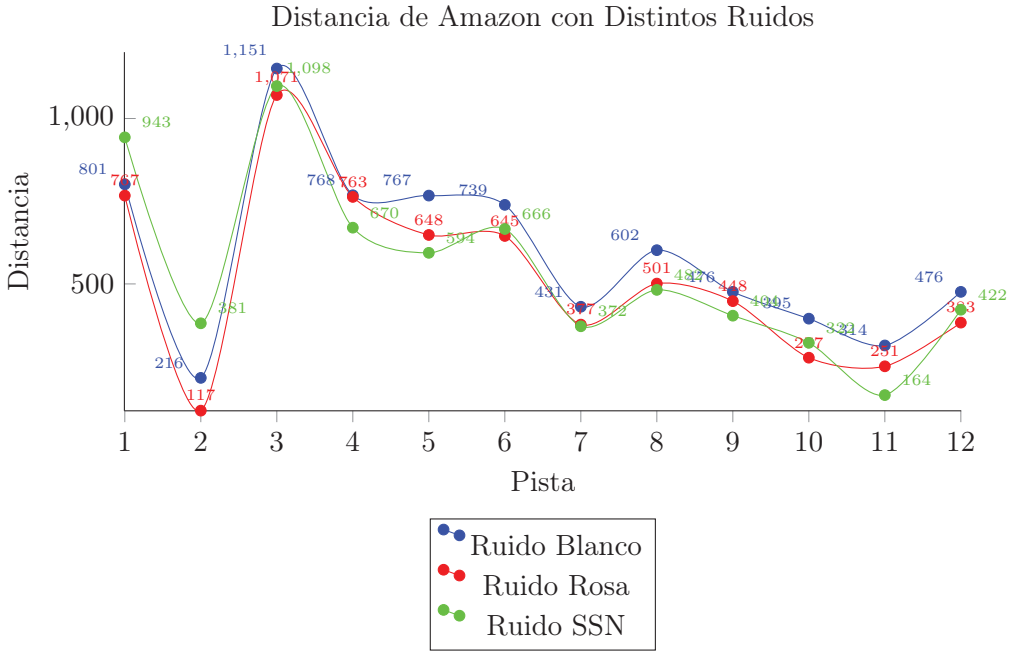

Figura 11: Distancia de Amazon con Distintos Ruidos

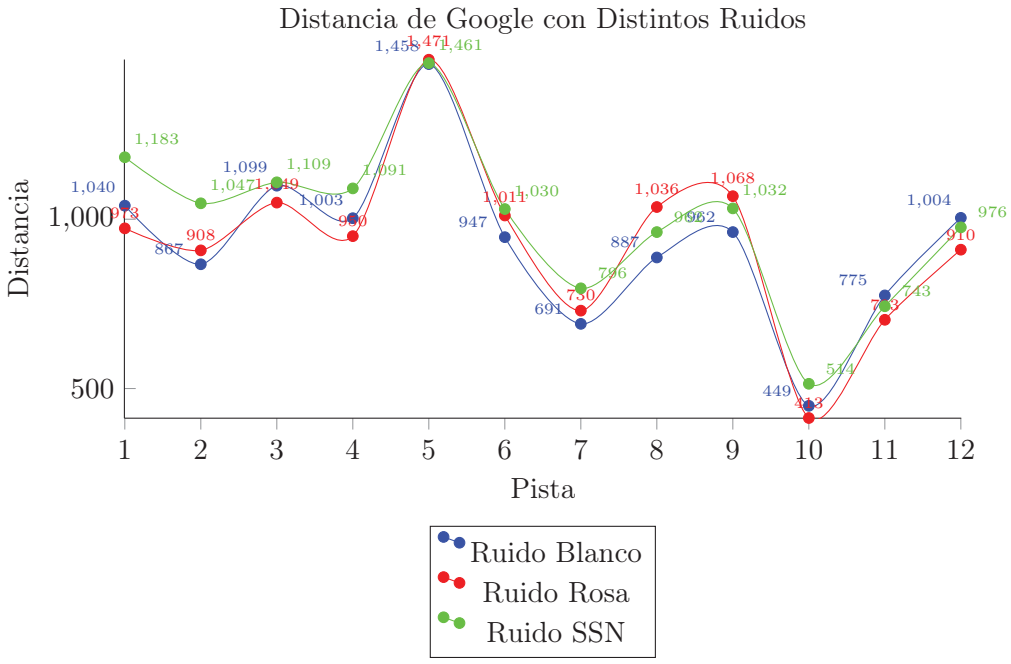

Figura 12: Distancia de Google con Distintos Ruidos

6.5. Conclusiones del Experimento

El problema que destaca y que fue encontrado en los experimentos, es que un solo espectro de ruido no entregaba buenos resultados. Esto es debido a que una sola pista de audio presenta una gran variedad de ruidos por lo que no hay una homogeneidad de distorsiones en las pistas. Esto quiere decir que en un audio podemos encontrar diferentes tipos de interferencias que transcurren de forma secuencial. Es debido a ello, que se le dificulta al modelo eliminar el ruido en su totalidad en cada una de las pistas de audio.

Considerando los resultados de todos los experimentos es posible señalar que el método no logró lo esperado, si bien se modificaron los datos de entrenamiento agregando distintos tipos de ruido, como es posible apreciar en la gráfica 13, en la que se agrupó los mejores resultados de lo que fue el *denoising* aplicando ruido rosa y el mejor modelo obtenido desde Wavenet con la data sin tratar. Los nuevos modelos entrenados no superan en promedio a las transcripciones realizadas previamente a la etapa de *denoising*.

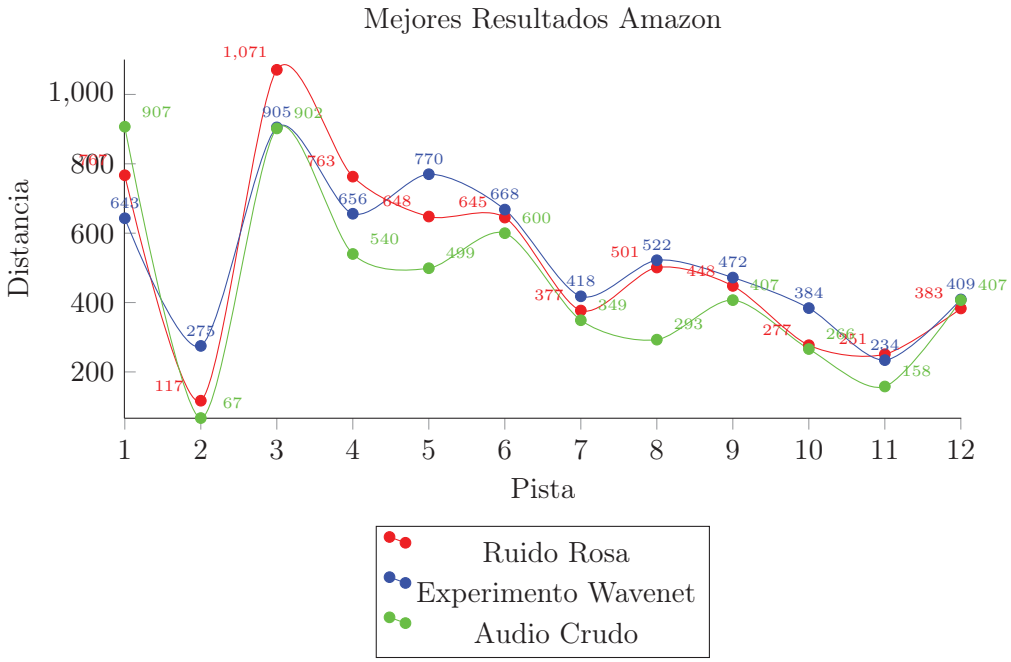

Figura 13: Mejores Resultados Amazon

7. Indexación de Documentos Transcritos

La indexación de palabras es la etapa final del proyecto y quizás la que puede generar mayor utilidad para el usuario final. Esto es debido a que se pueden realizar diferentes estudios con los datos generados, realizando análisis cualitativos de los documentos y llegando a un mayor entendimiento de estos, además poder realizar búsquedas sobre el universo de textos ayuda una eficiente manipulación.

En la primera aproximación hacia la indexación se propuso crear una bolsa de palabras con los textos transcritos que permitiera realizar operaciones básicas como extraer todos los sustantivos, pronombres, entre otros tipos de palabras, además saber el número de veces que aparece determinada palabra. Si bien, esta bolsa de palabras ayudó a comprender como funciona la manipulación de datos, no cumple con los requerimientos mínimos para lograr una indexación de palabras del nivel deseado.

Otra alternativa que se exploró fue la implementación de un índice invertido, puesto que se investigó que es una forma eficiente de realizar consultas en una indexación de datos. Se desarrollo un índice invertido que posteriormente fue alimentado con los documentos generados y se lograron buenos resultados. Sin embargo, debido a problemas de eficiencia y complejidad en el desarrollo se optó por buscar una mejor solución.

Entre las herramientas de indexación de datos que fueron investigadas, destaca Solr sobre las otras, debido a su facilidad de uso y versatilidad para trabajar en conjunto con otras herramientas de desarrollo. Además, cabe mencionar que esta herramienta hace uso del índice invertido para realizar eficientes consultas. Por lo dicho anteriormente, se decidió utilizar Solr por sobre las otras soluciones ya mencionadas.

Luego de haber indexado los datos utilizando Solr se propuso el desarrollo de una aplicación web como interfaz para el usuario. La aplicación web permite al usuario acceder de forma flexible y sencilla a los registros de los documentos, agregando diferentes funcionalidades para generar consultas y manipular la data.

8. Trabajo a Futuro

A partir de esta investigación es posible dilucidar y comprender aspectos a mejorar y considerar para perfeccionar lo que fue aislar el ruido presente en las pistas de la Radio Moscú, ya que éste fue el aspecto que mayor dificultades y complicaciones presentó. Para poder afrontar una problemática como lo es aplicar técnicas de *denoising* en audios tan complejos y, a ratos, difíciles de comprender como lo son los de la Radio Moscú, es posible que implementando un modelo a medida capacitado para aislar los ruidos sea posible obtener un sistema mejor adaptado para afrontar los ruidos e interferencias presentes en este programa radial.

Con un buen hardware, principalmente compuesto por una potente tarjeta gráfica, se debería esperar un mejor comportamiento en lo que es computar y resolver aislar los ruidos presentes de un audio. En ocasiones ciertas investigaciones estudiadas señalaban utilizar ordenadores que superaban los 5GB de memoria de video y una cantidad por sobre los 12GB de memoria RAM. Esto considerando que gran parte de los estudios visualizados utilizaron GPU.

Se sugiere que para realizar un mejoramiento de audio exitoso se utilice un dataset de entrenamiento que cuente con características similares a lo que son los del programa Escucha Chile, ya que en esta ocasión, los audios utilizados en el entrenamiento fueron unos que contemplaban ruidos de calle, bares y muchedumbre, muy distintos a los ruidos presentes en la Radio Moscú. También el uso de ruidos generados como lo son el ruido rosa, blanco, SSN, entre otros. Podría hacer surgir un modelo mejor capacitado para aislar ruidos. También se sugiere que al utilizar algún modelo, hacer uso de una data de entrenamiento con audios en español, como lo son las de este programa radial, ya que esto podría ser beneficioso y así obtener mejores resultados.

En cuanto a la indexación de datos es posible extender esta etapa, más allá de las funcionalidades ya entregadas, agregando nuevas funcionalidades a los documentos como el análisis semántico. El análisis semántico entregaría al momento de realizar consultas resultados más complejos y detallados, permitiendo exprimir aún más los datos.

9. Conclusiones

Considerando lo complejo que resulta elaborar un sistema de reconocimiento automático del habla, se optó por utilizar un servicio ya elaborado para así obtener más fácilmente la transcripción de estos audios. Durante la investigación se emplearon principalmente las APIs de Amazon y Google, ambas tuvieron un desempeño diferente al transcribir los audios de la Radio Moscú. La tendencia fue que en general Amazon Transcribe tenía una mayor precisión en acertar lo que se estaba hablando. Por lo tanto se señala que es preferible hacer uso de la API de Amazon por sobre la de Google.

Los diferentes experimentos realizados durante la etapa de *denoising* no lograron cumplir con el objetivo de eliminación del ruido, principalmente debido a problemas y limitaciones que ya fueron descritas en capítulos anteriores. A pesar de no haber logrado aislar los ruidos de las voces de los locutores, es posible señalar que se cuentan con nociones de como se podría sobrellevar esta adversidad mediante las sugerencias descritas en el capítulo 9. Lo anterior es debido a que el modelo utilizado basado en Wavenet, no arrojó buenos resultados por una mala calibración y limitantes técnicas.

En cuanto a la indexación, si bien se nombraron variadas alternativas, a medida que se fue concretando la investigación y proyecto se decidió desechar las anteriores y optar exclusivamente por Solr. Para facilitar la manipulación de los datos haciendo uso de Solr, se desarrolló una aplicación web que permite navegar y realizar una variedad de consultas por todos los documentos indexados.

Referencias

[1] "Biblioteca del Congreso Nacional de Chile." https://www.bcn.cl/noticias/escucha-chile-comunicacion-politica-y-solidaridad-1973-1990, visitado el 28/02/2018.

[2] B. del Congreso Nacional de Chile, *Escucha Chile, Comunicación, política y solidaridad 1973-1990*. Biblioteca del Congreso Nacional de Chile, 2015.

[3] J. Soo Lim, "Enhancement and bandwidth compression of noisy speech by estimation of speech and its model parameters.," 08 2005.

[4] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," 06 2017.

[5] S. Pascual, A. Bonafonte, and J. Serrà, "Segan: Speech enhancement generative adversarial network," pp. 3642–3646, 08 2017.

[6] F. G. Germain, Q. Chen, and V. Koltun, "Speech Denoising with Deep Feature Losses," 06 2018.

[7] K. Santosh, W. Bharti, and P. Yannawar, "A review on speech recognition technique," vol. 10, 11 2010.

[8] J. David Celis Nuñez, R. Andres Llanos Castro, B. Medina Delgado, S. Sepulveda, and S. Alexander Castro Casadiego, "Modelo acústico y de lenguaje del idioma español para el dialecto cucuteño, orientado al reconocimiento automático del habla," vol. 22, p. 362, 09 2017.

[9] Z. Zhang, J. Geiger, J. Pohjalainen, A. Mousa, and B. Schuller, "Deep learning for environmentally robust speech recognition: An overview of recent developments," 05 2017.

[10] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," 07 2015.

[11] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," pp. 410–414, 09 2016.

[12] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," 12 2011.

[13] P. O. Glauner, J. Iwaszkiewicz, J.-Y. LeMeur, and T. Simko, "Use of solr and xapian in the invenio document repository software," *CoRR*, vol. abs/1310.0250, 2013.

[14] P. Papadopoulos, A. Tsiartas, J. Gibson, and S. Narayanan, "A supervised signal-to-noise ratio estimation of speech signals," 05 2014.

[15] Y. Hu and P. C. Loizou, "A generalized subspace approach for enhancing speech corrupted by colored noise," *Speech and Audio Processing, IEEE Transactions on*, 08 2003.

[16] M. Mathe, S. Nandyala, and T. Kishore Kumar, "Speech enhancement using kalman filter for white, random and color noise," 03 2012.