Speech to Command

Eduard Caballero, Iván Castillo, María Aparicio
24552– Inteligencia Artificial II - Grupo J1
Escuela de Ingeniería de Sistemas e Informática

Resumen

La accesibilidad de los dispositivos inteligentes en la actualidad es un factor incidente en la autonomía de tareas y flujo de actividades cotidianas. Existen asistentes por voz que se basan en tecnología con Inteligencia Artificial [1].

El objetivo principal de este proyecto es explorar y aportar a este campo de estudio con el desarrollo de *Speech to Command,* software que permite reconocer comandos en inglés por voz con el propósito de facilitar la interacción con dispositivos inteligentes, utilizando Deep Learning en su núcleo.

Introducción

Las redes neuronales han mostrado ser una herramienta versátil capaz de modelar todos los aspectos acústicos, fonéticos y lingüísticos asociados con la tarea de reconocimiento de voz [1].

Speech to Command es un asistente virtual de voz, basado en Deep Learning, con capacidad de reconocer comandos de voz simples en inglés mediante una interfaz que registra la voz y permite ejecutar las acciones de acuerdo a los comandos detectados en los audios grabados (ver fig. 1). Los resultados obtenidos fueron aceptables en términos de precisión de entrenamiento y testeo.

Figura 1. Entorno gráfico de la implementación.

Proceso y método

Se aplica una metodología que abarca el procesamiento de audios de un dataset (ver fig. 2 y fig. 3) que consta de 65.000 audios, cada uno de un segundo, y dividido en 30 clases [2]. Luego, se construye un dataset con características extraídas de los audios para el entrenamiento de modelos DNN, RNN GRU y LSTM [3-4]. Por último, se evalúan los modelos previamente entrenados mediante dos tipos de pruebas: por pablabras y por frases (ver fig. 4).

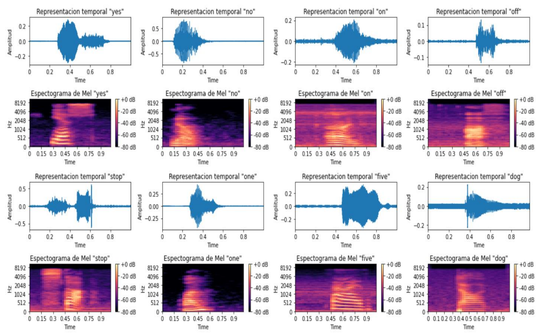

Figura 2. Representación temporal y mel de los audios (clases).

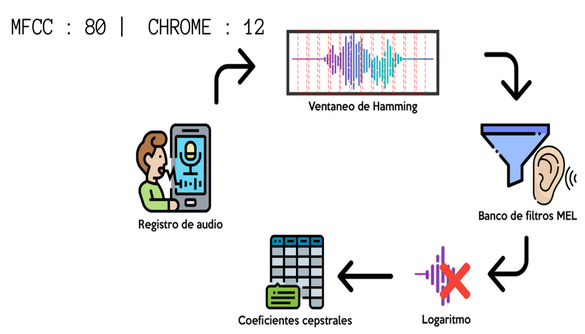

Figura 3. Procesamiento de audios.

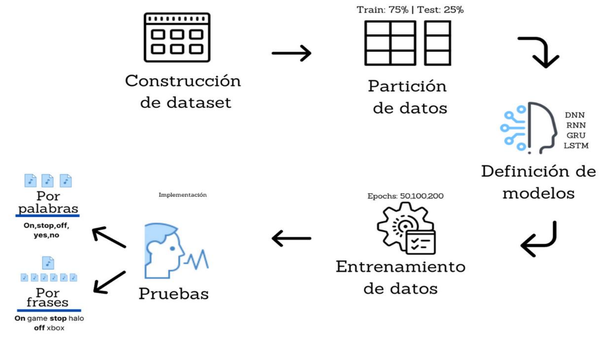

Figura 4. Entrenamiento de modelos y pruebas.

Resultados

Se seleccionaron *on, off, yes, no y stop* como comandos o clases de prueba. Además, se creó una nueva clase llamada *unknown* a partir de la concatenación de otros comandos para los comandos irreconocibles por el sistema de voz.

Los resultados se clasifican por pruebas con la partición del dataset y las pruebas con la implementación desarrollada. Los mejores resultados a partir de pruebas con la partición de test del dataset se obtuvieron con el modelo RNN entrenado con 50 Epochs. Por otro lado, el modelo con mejor rendimiento con la clase *unknouwn* fue GRU entrenado también con 50 Epochs (ver fig. 5).

En cuanto a los mejores resultados a partir de las pruebas con la implementación, que son las pruebas reales hechas sobre la interfaz, el modelo RNN tuvo una precisión de 84% en las pruebas con palabras y el modelo DNN una precisión del 83% para las pruebas con las frases contenidas en la tabla Frases (ver fig. 6).

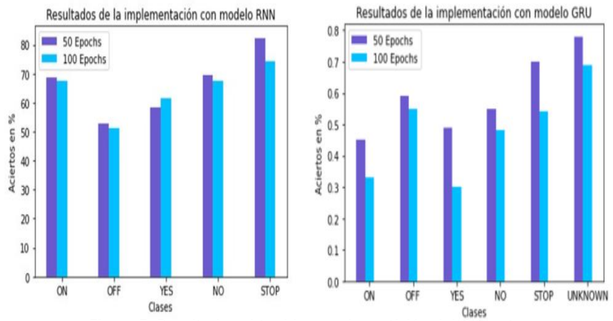

Figura 5. Resultados obtenidos con la partición del dataset.

Análisis de Resultados RNN	Comandos		
	# comandos detectados	# comandos reales	% Acierto
	3	5	60%
	4	5	80%
	5	5	100%
	4	5	80%
	5	5	100%
Totales	21	25	84%

Análisis de Resultados DNN	Comandos		
	# comandos detectados	# comandos reales	% Acierto
	3	2	67%
	2	2	100%
	3	3	100%
	3	2	67%
	1	1	100%
Totales	12	10	83%

Figura 6. Resultados obtenidos con la implementación (pruebas reales).

Conclusiones

- Las características frecuenciales MFCC y Chroma, en conjunto, son una representación que garantiza una predicción aceptable de los comandos de voz.
- La detección de comandos en una frase es un reto de mayor complejidad en comparación a la detención de comandos por palabras, ya que el tratamiento de la frase captada depende de su duración y nivel de potencia para poder identificar cada palabra en esta.
- Una arquitectura sencilla como la DNN implementada puede lograr un rendimiento igual o mayor de óptimo con respecto a las redes neuronales recurrentes.

Trabajo Futuro

Se propone trabajar con comandos tipos números (*one, two, three, four, five*) debido a que las representaciones del audio de los comandos seleccionados (ver figura 4) poseen amplitudes de ondas de sonido similares, provocando posibles sesgos y errores en las predicciones.

Información de contacto

Eduard Alfonso Caballero, Email: eduard.caballero@correo.uis.edu.co
María Camila Aparicio, Email: maria.aparicio2@correo.uis.edu.co
Iván Rodrigo Castillo, Email: ivan.castillo@correo.uis.edu.co

Referencias Bibliográficas

1. Bonafonte, A., (2016). El Deep Learning Revoluciona Las Tecnologías Del Habla | Blog CIT UPC. (2016). Consultado el 6 de Septiembre de 2020, en https://blog.cit.upc.edu/?p=965.
2. TensorFlow Speech Recognition Challenge | Kaggle. (2016). Consultado el 6 de Agosto de 2020, en https://www.kaggle.com/c/tensorflow-speech-recognition-challenge
3. Criptografía práctica. (2013). Consultado el 10 de agosto de 2020, en http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/
4. Jogy, J. (2019). How I Understood: What features to consider while training audio files?. Consultado el 14 de Agosto de 2020, en https://towardsdatascience.com/how-i-understood-what-features-to-consider-while-training-audio-files-eedfb6e9002b