Informe de seguimiento II - TFG

Speaker Recognition

Carlos Berea

Índice

1 - Problema a resolver

1.1 - Introducción al problema

En la actualidad en la tecnología se busca que el usuario este lo más cómodo y seguro posible. Esto se puede llegar a conseguir con una buena personalización de la experiencia del usuario. En el caso de la seguridad en la tecnología se ha vuelto muy importante el hecho de conseguir la forma más segura de poder llegar a identificar al usuario. Algunas de las aplicaciones más comunes en las cuales es necesaria la identificación del usuario pueden ser realizar compras online, pedir al teléfono que realice acciones automáticamente con la voz, acceder a un lugar a través de algún tipo de reconocimiento, como por ejemplo facial, por voz o iris, o algo tan simple como acceder a tu teléfono móvil.

Anteriormente todas estas identificaciones de usuarios se hacían a través de una contraseña o un patrón. Pero con el tiempo se han ido descubriendo formas de identificar al usuario de una manera más cómoda y segura, como puede ser el reconocimiento dactilar, facial, de iris o de voz. Estos son mucho más seguros y consiguen una mejor experiencia de usuario en todos los campos de la tecnología.

En este caso nos centraremos en el reconocimiento de voz. El cual podríamos dividir en reconocimiento de lo que se dice o reconocimiento de quien lo dice. Por un lado, está el speech recognition, que es el que se dice, esto tiene muchísimas aplicaciones, como la escritura de texto a través de voz o realizar acciones con el móvil, como por ejemplo el 'Ok Google'. Pr el otro lado, tenemos al speaker recognition que es el que se encarga de saber quién lo dice, que podemos ver en Siri o en acceso a viviendas u oficinas a través de reconocer quién eres.

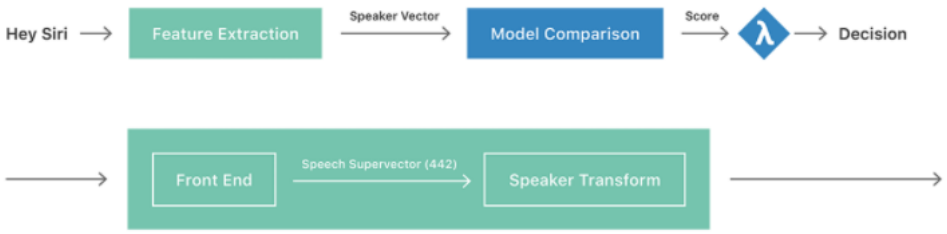
Aquí trataré el quien lo dice ya que actualmente es una de las formas más seguras de reconocer a la persona, debido a que con la otra reconocerías que está haciendo y por ello que acción se debe de realizar, pero no quien lo está diciendo y por lo tanto hay más comodidad a la hora de uso, pero no mayor seguridad. Esto se podría llegar a aplicar en múltiples casos, como por ejemplo reconocer la persona que está hablando en una conversación, poder hacer acciones con el teléfono móvil como Siri o acceder a un sitio a través de una puerta que tenga control de acceso por reconocimiento de voz.

Mas que un problema esto es un tema a tratar e intentar mejorar o replicar, ya que actualmente ya hay tecnologías que realizan el reconocimiento de voz. Ejemplos de estas empresas pueden ser Amazon, Apple, IDR&D [1], entre muchas otras, las cuales ya tienen tecnologías muy avanzadas que con un entrenamiento de apenas entre 3 y 5 frases consiguen hacer un modelo de comparación para conseguir reconocer al hablante.

1.2 - Explicación speaker recognition

Speaker recognition [2], como su nombre lo indica, se encarga de verificar e identificar al hablante de la mejor manera posible a través de la IA (Inteligencia Artificial), esto se consigue a través de comparar la voz entrante con las características de la voz del hablante con el que se ha entrenado el modelo. Esto se puede hacer debido a que cada persona tenemos unas características únicas en la voz que nos hacen diferenciarnos de los demás, y en este caso consigue que el speaker recognition pueda llegar a ser posible.

Normalmente el proceso para poder llegar a tener un speaker recognition se divide en dos fases la de enrollment y recognition. En la fase de enrollment el usuario dice unas frases, estas frases son usadas para crear el modelo de la voz del usuario. En la fase de reconocimiento el sistema compara la voz que se le envía con el modelo entrenado del usuario y decide si aceptarla como buena o no. Este es el caso de Siri, que se puede ver cómo es así en el artículo de Apple de Hey Siri [3] por ejemplo. Aquí podemos apreciar un pequeño diagrama de cómo funciona:



2 – Objetivo

El objetivo principal de este TFG es conseguir un speaker recognition, con código propio o mejorando un código abierto ya realizado como otros speaker verification que utilizan convolutional neural networks [5] el cual sea capaz de reconocer al hablante. Este speaker recognition será entrenado para poder llegar a reconocer al hablante en inglés y con una calidad de audio que no tenga demasiado ruido de fondo, para que sea facil de captar la voz y así ser más sencillo la labor de reconocer quién está hablando en ese momento, esto se puede hacer con datasets como el de Voxceleb [4].

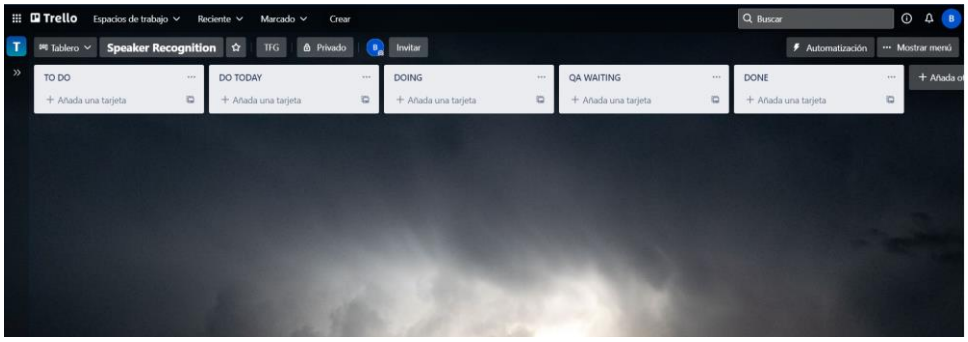
Esto puede llegar a ser complejo por lo tanto antes de llegar al objetivo final de conseguir un speaker recognition como este pasaremos por unos subobjetivos iniciales, más sencillos que nos harán llegar poco a poco a nuestro objetivo final, pero que en caso de no poder llegar a cumplir con el objetivo principal podriamos llegar a quedarnos con alguno de los subobjetivos anteriores y desarrollarlo al máximo. Los sub-objetivos vienen detallados en la siguiente tabla por orden de realización:

Objetivo	Resultado esperado
Speaker Verification	Conseguir entrenar un modelo con un dataset en inglés, con una calidad de audio que no tenga mucho ruido, capaz de verificar que el hablante es el usuario correcto.
Speaker Identification	Conseguir entrenar un modelo con un dataset en inglés, con una calidad de audio que no tenga mucho ruido, capaz de identificar al hablante con audios que sean del mismo dataset que hemos utilizado para entrenarlo.
Speaker Recognition (Objetivo final)	Conseguir entrenar un modelo con un dataset en inglés, con una calidad de audio que no tenga mucho ruido, capaz de identificar al hablante con audios que sean de cualquier dataset.

3 - Metodología

La metodología que voy a utilizar será una metodología Agile en concreto Kanban. Esto me permitirá organizar mis tareas de una forma más ágil y sencilla sin necesidad de un Scrum Master o con los sprints los cuales en este tipo de proyectos tan cortos son inviables ya que no se pueden realizar sprints de 2 semanas. Esto es debido a que las tareas de una semana a otra se pueden torcer y cambiar completamente.

Esta metodología Kanban la utilizare con tarjetas con la plataforma Trello, esta me permite crear diferentes columnas para poder organizar todas las tareas pendientes, tareas para hoy, tareas en curso, QA Waiting y finalizadas. Como la siguiente imagen ejemplo:



En este caso no hay equipo de QA por lo tanto el trabajo de cualificar la calidad del código realizado se hará por parte mía y de mi tutor consensuando si realmente lo que se ha desarrollado

es válido o no. Una tarea no pasará a DONE (Es decir finalizada), hasta que se llegue a tener una revisión y acuerdo entre el tutor y yo sobre que la tarea está bien y por lo tanto finalizada.

Como ya he comentado anteriormente las tareas serán revisadas antes de ser finalizadas, esta revisión con el tutor se llevará a cabo en las reuniones, las cuales se realizarán aproximadamente cada 2 semanas (Las reuniones habituales antes de la entrega de cada informe y una entre medias). A parte de utilizarlas para revisar el trebejo hecho se resolverán dudas y se hablará de cómo organizar las siguientes semanas.

La planificación se llevará a cabo a cuatro semanas vista, esto se hace debido a que es la manera más fácil de organizarse en este tipo de proyectos, en los cuales sabes el objetivo final, pero puede haber muchos problemas por el medio que te hacen cambiar las planificaciones que tenías para hacer de una semana para otra. Por lo tanto, tener una planificación muy larga de más de un mes no sería lógica ya que es muy probable que no se cumpla, mientras que a 5 semanas vista habría menos cambios ya que en la planificación que se va haciendo es más real porque es más reciente. Al ser de 5 en 5 semanas la planificación se irá actualizando cada vez que se vayan a terminar esas semanas o en el caso de saber lo que se va a hacer antes, se actualizará antes. Una vez finalizadas las tareas habrá un apartado de resultados donde explicaré como ha ido esa tarea y si se ha conseguido realizar o no.

En el caso de haber algún contratiempo y no se pueda llegar a los resultados esperados de las tareas de esa semana se realizará un día de estudio de viabilidad, para saber si es correcto seguir por el camino que se estaba tomando o si con ello se va a perder mucho tiempo y hay que cambiar de tarea, en el caso de tener que cambiar de tarea habría que dedicar otro día a estudiar la otra posible vía.

4 - Desarrollo

Fecha	Tareas	Objetivo	Resultados Esperados
11/10 - 17/10	Buscar un dataset que se pueda adecuar a las necesidades del proyecto y buscar códigos ejemplo de speaker verification para ver que tal funcionan.	Speaker Verification	Encontrar un dataset en inglés y con poco ruido de fondo y un código de speaker verification lo suficiente mente bueno y que funcione.

Se ha conseguido llegar al objetivo de conseguir el dataset en inglés y un código de speaker verification, como es el caso del Speaker-verification-pytorch [8].			
18/10-24/10	Entrenar un modelo con el código y el dataset encontrados.	Speaker Verification	Que el código consiga entrenar el modelo con el dataset y no de ningún tipo de problema.
Se ha conseguido, pero no de la manera esperada, debido a problemas con la descarga del dataset VoxCeleb1 [4] a la hora de hacer un cat, el cuál era necesario debido a que el dataset de train venía separado en tres partes, de los archivos aún no se ha llegado al objetivo de conseguir un modelo entrenado a partir del train de VoxCeleb [4] descargado. Pero por otro lado se han encontrado modelos ya entrenados con los pesos con el mismo dataset que he elegido y la parte de test del dataset no ha dado problemas debido a que es a que está en un único archivo .zip.			
25/10-31/10	Mejorar el código o crear uno propio para entrenarlo con el dataset.	Speaker Verification	Obtener unos resultados de acierto en el speaker verification óptimos.
No se ha conseguido, este objetivo no se ha conseguido realizar debido a un bloqueo por culpa de alguno de los códigos a probar debido a que eran demasiado antiguos y por lo tanto daban problemas con las versiones, y después de desbloquearme encontrando otros códigos más recientes empezó a dar problemas a la hora de juntar las diferentes partes del dataset VoxCeleb [4], debido a permisos a la hora de la descarga de los archivos.			
01/11-07/11	Coger otro de los dataset en inglés como por ejemplo uno de Friends [6] y que no tenga mucho ruido para juntarlo con el actual y entrenar un nuevo modelo.	Speaker Verification	Que el código funcione y consiga entrenar el modelo sin problemas.
No se ha conseguido debido a las razones dadas en el apartado anterior, debido a que esta es una continuación del mismo.			
08/11-14/11	Mejorar el código para entrenar con el primer dataset y hacer test con el segundo.	Speaker Verification	Conseguir unos buenos resultados al hacer test con el modelo entrenado con otro dataset diferente.
No se ha conseguido debido a las razones dadas en los apartados anteriores, debido a que esta es una continuación de ellos.			
Fecha	Tareas	Objetivo	Resultados Esperados

15/10 - 21/10	Conseguir entrenar un modelo con el dataset que encontrado para ver cómo funciona y cuánto puede llegar a tardar.	Speaker Verification	Entrenar el modelo en un tiempo que sea óptimo.
No se ha conseguido entrenar de una manera óptima la red neuronal debido a problemas computacionales del ordenador en los cuales en hacer una sola epoch tardaba más de una hora.			
22/11-28/11	Entrenar al speaker verification con los diferentes modelos para la red neuronal y comparar cual da mejores resultados.	Speaker Verification	Conseguir datos realistas para poder hacer una buena comparación entre ellos.
29/11-05/12	Entrenar al speaker verification con los diferentes modelos para la red neuronal y comparar cual da mejores resultados. Lo pongo dos veces seguidas debido a que viendo las previsiones anteriores puedo ver que no va a ser tan sencillo realizar esta tarea y que puede dar problemas.	Speaker Verification	Conseguir datos realistas para poder hacer una buena comparación entre ellos.
Se han podido llegar a comparar los resultados de los diferentes modelos pero los ya pre entrenados, para ver los diferentes resultados que se obtienen con los modelos y loss settings diferentes. Se ha realizado un estudio sobre los diferentes tipos de modelos que hay y mejor combinacion de los mismos siendo el mejor ECAPA-TDNN [13] con AAMSoftmax.			
06/12-12/12	Sacar conclusiones de los datos obtenidos, para ver cuál sería la mejor opción para este problema. Como se realiza en el Benchmark-Analysis-Speaker-Recognition-Techniques [12]	Speaker Verification	Sacar unas buenas conclusiones para poder aplicarlas a mi código.
Como se ha comentado anteriormente las conclusiones que se han podido sacar con la comparación tanto experimental con los modelos preentrenados con sus pesos y otros trabajos y estudios sobre el tema, la mejor combinación sería la ECAPA-TDNN con AAMSoftmax. Con en algunos casos un EER de únicamente un 0,87. Este 0,87 fue conseguido con un ECAPA-TDNN entrenada con el dataset de VoxCeleb 2 y el test se realizó con VoxCeleb 1, ambas en su versión dev.			
13/12-19/12	Intentar llevar a cabo un código propio con las conclusiones que hemos conseguido sacar para poder hacer posteriormente speaker identification con él.	Speaker Identification	Conseguir un código que pueda llegar a hacer speaker identification.

Se ha podido llevar a cabo un código propio basado en el trabajo 'In defence of metric learning for speaker recognition' [14], del cual se ha hecho la versión con ECAPA-TDNN y AAMSoftmax ya que era la mejor combinación. La cual aún no se ha podido probar debido a problemas computacionales.

5- Planificación

Fecha	Tareas	Objetivo	Resultados Esperados
20/12 - 26/12	Se seguirá profundizando en el ECAPA-TDNN, para intentar hacer un entrenamiento del modelo aunquesea de 1 epoch para poder ver los resultados aunque no sea algo óptimo.	Speaker Verification	Entrenar el modelo.
27/12-02/01	Ver la posibilidad en el caso de conseguir resultados con ECAPA-TDNN ver la posibilidad de implementar el MACCIF-TDNN [15], que es una modificación del modelo ECAPA-TDNN	Speaker Verification	Conseguir hacer un código que funcione del MACCIF-TDNN.
03/01-09/01	Seguir con el desarrollo del MACCIF-TDNN, debido a que seguramente lleve bastante tiempo realizarlo.	Speaker Verification	Conseguir datos realistas para poder hacer una buena comparación entre ellos.
10/01-23/01	En estas semanas restantes se realizará el paper para la entrega del proyecto final.	Speaker Verification	Hacer el paper

6- Referencias

[1] - Speaker Recognition Software - IDVoice n.d., IDR&D, acceso 10 de Octubre de 2021, <https://www.idrnd.ai/text-independent-voice-verification>

[2] - Sadaoki Furui 2008, Scholarpedia, acceso 10 de Octubre de 2021, <http://www.scholarpedia.org/article/Speaker_recognition >

[3] - Siri Team 2018, Apple, acceso 10 de Octubre de 2021, <https://machinelearning.apple.com/research/personalized-hey-siri>

[4] - Nagrani, A., Chung, J. S., & Zisserman, A. (2017). Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*.

[5] - Torfi, A., Dawson, J., & Nasrabadi, N. M. (2018, July). Text-independent speaker verification using 3d convolutional neural networks. In *2018 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1-6). IEEE.

[6] - Jinho D. Choi 2016, acceso 10 de Octubre de 2021, < https://github.com/emorynlp/character-mining >

[7] - Yamagishi, Junichi; Veaux, Christophe; MacDonald, Kirsten. (2019). CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92), [sound]. University of Edinburgh. The Centre for Speech Technology Research (CSTR).

[8] - Bernard Deng 2021, acceso 12 de Octubre de 2021, < https://github.com/hiimmuc/Speaker-verification-pytorch#readme >

[9] - Chung, J. S., Huh, J., Mun, S., Lee, M., Heo, H. S., Choe, S., ... & Han, I. (2020). In defence of metric learning for speaker recognition. *arXiv preprint arXiv:2003.11982*.

[10] - Heo, H. S., Lee, B. J., Huh, J., & Chung, J. S. (2020). Clova baseline system for the voxceleb speaker recognition challenge 2020. *arXiv preprint arXiv:2009.14153*.

[11] - Matejka, P., Novotný, O., Plchot, O., Burget, L., Sánchez, M. D., & Cernocký, J. (2017). Analysis of Score Normalization in Multilingual Speaker Recognition. In *INTERSPEECH* (pp. 1567-1571).

[12] - Gianluca Pagliara 2021, 12 de Octubre de 2021, < https://github.com/gianlucapagliara/Benchmark-Analysis-of-Speaker-Recognition-Techniques >

[13] - Desplanques, B., Thienpondt, J., & Demuynck, K. (2020). Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*.

[14] - Chung, J. S., Huh, J., Mun, S., Lee, M., Heo, H. S., Choe, S., ... & Han, I. (2020). In defence of metric learning for speaker recognition. *arXiv preprint arXiv:2003.11982*.

[15] - Wang, F., Song, Z., Jiang, H., & Xu, B. (2021). MACCIF-TDNN: Multi aspect aggregation of channel and context interdependence features in TDNN-based speaker verification. *arXiv preprint arXiv:2107.03104*.