USO DE REDES NEURONALES ARTIFICIALES PARA EL RECONOCIMIENTO DE PALABRAS CLAVE PARA UN ROBOT DE SERVICIO

Universidad Nacional Autónoma de México
Posgrado en Ingeniería Eléctrica
Campo Disciplinario: Procesamiento Digital de Señales

PRESENTA:
Stephany Ortuño Chanelo
Contacto: saney.chanelo@gmail.com

TUTOR:
Dr. Jesús Savage Carmona
Contacto: robotssavage@gmail.com

CONTENIDO

COMPRENSIÓN DEL LENGUAJE NATURAL

El concepto de comprensión del lenguaje natural se refiere a poder realizar una transformación de su representación original a una donde sea posible determinar un conjunto de acciones a realizar (Rich and Knight, 1991).

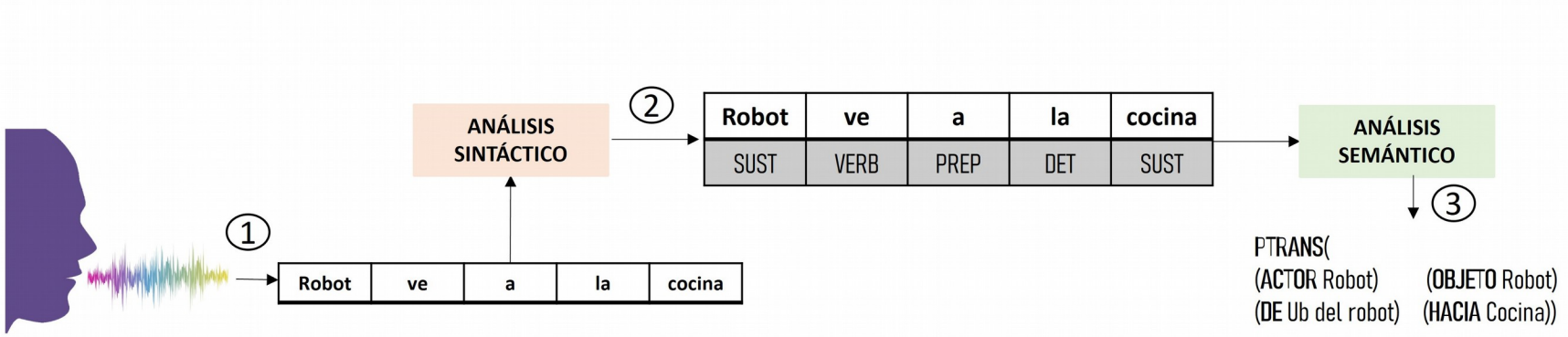


Fig 1. Diagrama del proceso de comprensión del lenguaje natural.

1. Transformación de la señal de entrada a unidades básicas.
2. Análisis sintáctico.
3. Análisis semántico.

RECONOCIMIENTO AUTOMÁTICO DE VOZ

El reconocimiento automático de voz (ASR) es un proceso encargado de decodificar y transcribir el habla. Un sistema ASR típico recibe la entrada a través de un micrófono, lo analiza usando algún patrón, modelo o algoritmo y produce una salida, generalmente en forma de texto.

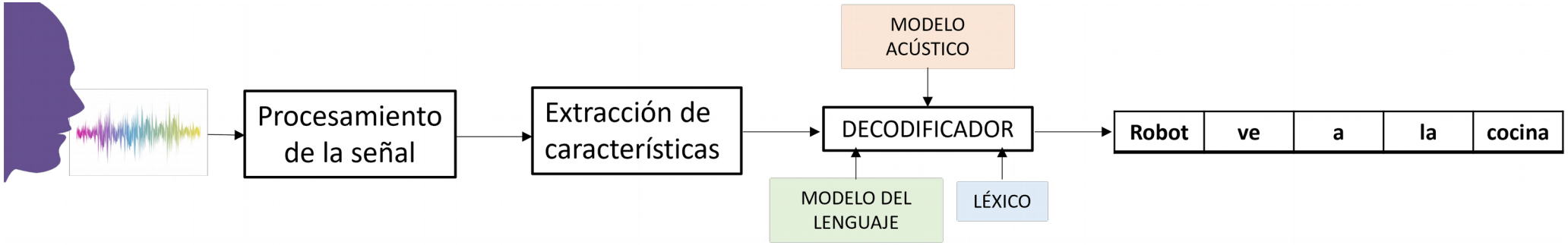


Fig 2. Diagrama de un sistema básico de ASR.

Aproximaciones:

1. Reconocimiento de palabras aisladas.
2. Reconocimiento de palabras clave.
3. Reconocimiento de voz continua.

EL PROBLEMA DEL ASR

A pesar de décadas de investigación en el área, el rendimiento de los sistemas de reconocimiento automático de voz no se acerca a las capacidades humanas. La dificultad en estos sistemas radica en la variabilidad de la señal de voz, ya que las características espectrales y temporales del habla varían dependiendo de una serie de factores (Juang y Rabiner, 2004).

- Fisiológicos: las diferentes dimensiones del tracto vocal cambian las frecuencias de la voz.
- Fenómenos acústicos ambientales: la señal grabada no solo captura la voz sino también múltiples efectos de reverberación y el habla de fondo de múltiples hablantes.
- Variabilidad: estos factores dependen de aspectos relacionados con el locutor como la pronunciación, la duración, etc.
- Continuidad: en el lenguaje natural no existen separadores entre las unidades, equivalentes a los espacios del lenguaje escrito.

La teoría de la **dependencia conceptual (CD)** tiene como premisa que una acción es la base de cualquier proposición. Todas las proposiciones que describen eventos están formadas por conceptualizaciones y un conjunto de roles que dependen de la acción, en general cada primitiva consta de los siguientes elementos: **ACTOR, ACCIÓN, OBJETO, DIRECCI[ON, ESTADO y TIEMPO.**

Schank propone un conjunto finito de acciones primitivas (Savage et al., 2019):

1. **ATRANS:** transferencia de propiedad, posesión o control de un objeto.
2. **PTRANS:** transferencia de la ubicación física de un objeto.
3. **ATTEND:** concentrarse en percibir un estímulo sensorial.
4. **MOVE:** movimiento de una parte del cuerpo.
5. **GRASP:** el actor toma un objeto.

ROBOT, TRAEME UNA MANZANA.

PTRANS ((ACTOR Robot) (OBJECT Manzana) (FROM Cocina)(TO John))

Fig 3. Representación de comando de voz.

JUSTIFICACIÓN

En la actualidad, dentro del ámbito de las interfaces hombre-máquina se busca llegar a un nivel de interacción lo más natural y cercano a lo que está acostumbrado el ser humano. Sin embargo, las técnicas actuales de reconocimiento de voz se encuentran lejos de compararse con la habilidad de un humano. Por esta razón algunas técnicas están inspiradas en la forma en que los humanos realizan esta tarea.

Por ello se propone utilizar redes neuronales artificiales para crear un sistema de reconocimiento automático de voz que permita potenciar las técnicas de procesamiento digital de señales en métodos más eficientes.

OBJETIVO GENERAL

Desarrollar un sistema de reconocimiento de palabras clave utilizando redes neuronales, el cual será utilizado por un robot de servicio autónomo que ejecutará tareas previamente programadas según las ordenes que reciba de manera verbal.

Fig 4. Robot Justina (Biorobotics , UNAM).

PROCESAMIENTO DIGITAL DE LA SEÑAL

Independiente de los métodos utilizados se requiere realizar algún procesamiento a la señal de voz (Rabiner y Schafer, 1979).

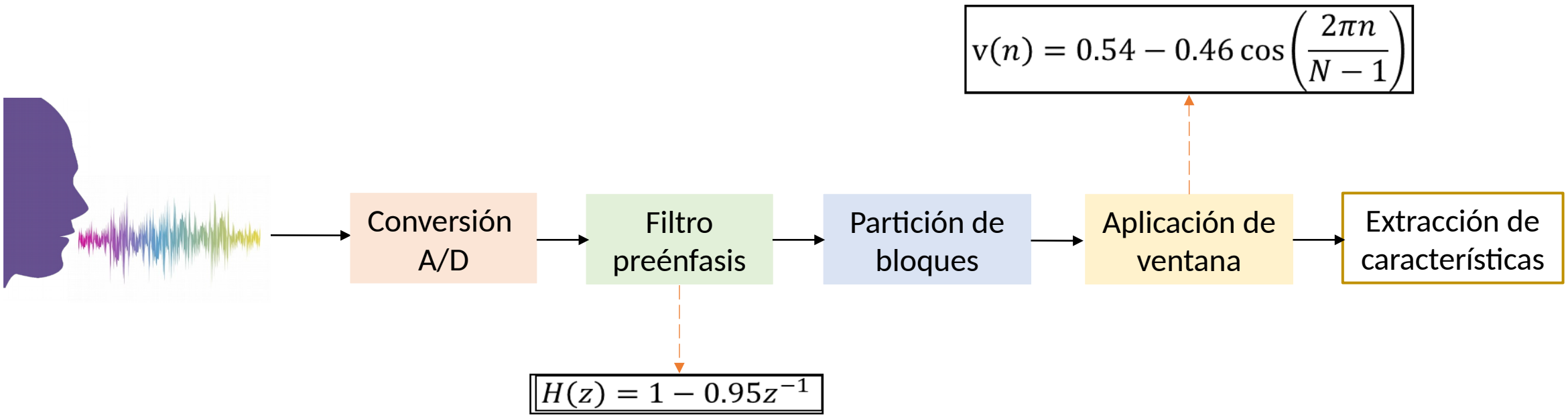


Fig 5. Procesamiento de la señal de voz.

EXTRACCIÓN DE CARACTERÍSTICAS

Su modelo establece que el tracto vocal puede modelarse mediante un filtro digital siendo los parámetros los que determinan la función de transferencia y permite aproximar una señal a partir de señales pasadas (Afzal et al., 2010).

$$y(n) = \sum_{k=1}^{p} a_k y(n-k) + e(n)$$

Para encontrar los coeficientes de filtro que mejor se adapten al segmento actual que se analiza el error cuadrático medio. Tomar la derivada produce un conjunto de M ecuaciones (Bradbury, J., 2000) y se puede utilizar el análisis de autocorrelación para determinar los coeficientes.

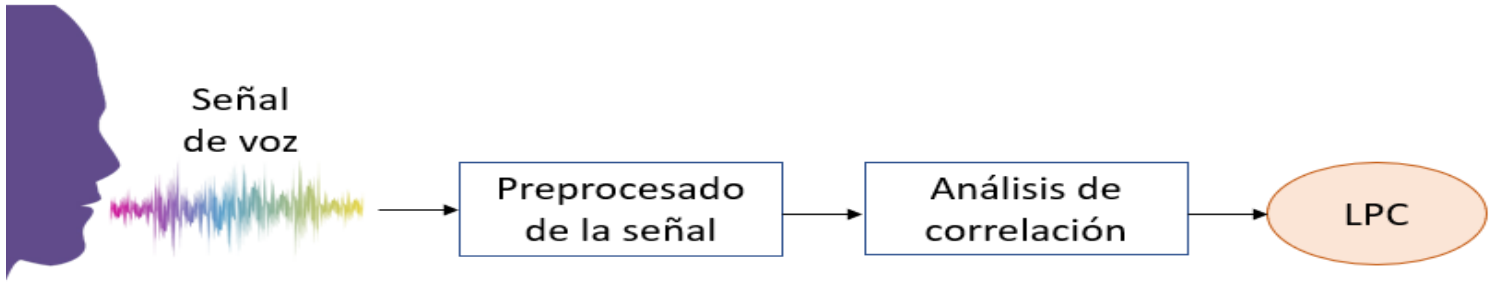


***Fig 6*. Diagrama de bloques del procesador LPC.**

EXTRACCIÓN DE CARACTERÍSTICAS

Se basa en la variación conocida del ancho de banda de frecuencia crítica del oído humano. Los estudios han demostrado que la percepción humana del contenido de la frecuencia del sonido para las señales del habla no sigue una escala lineal (Ramírez et al., 2019).

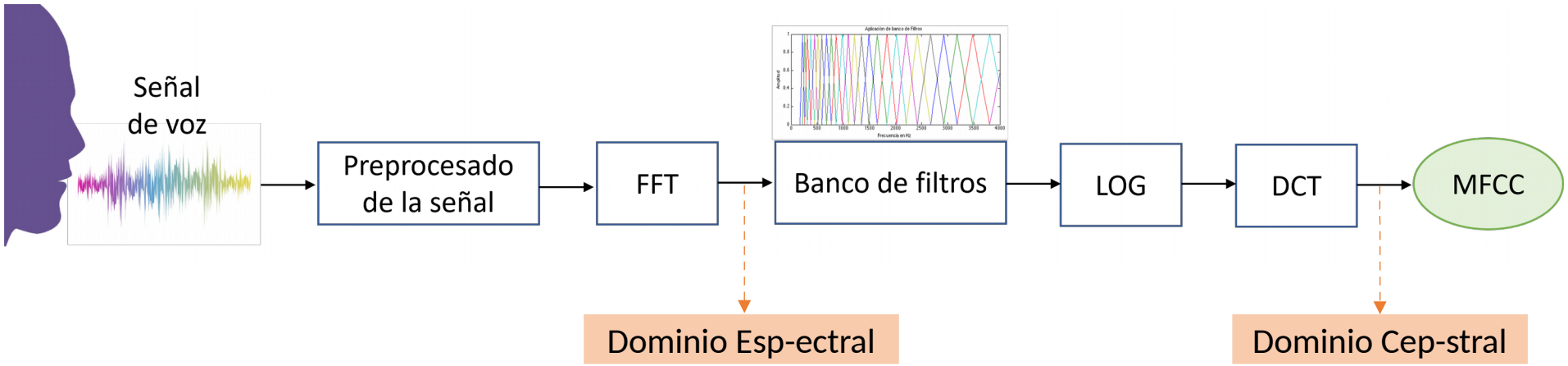
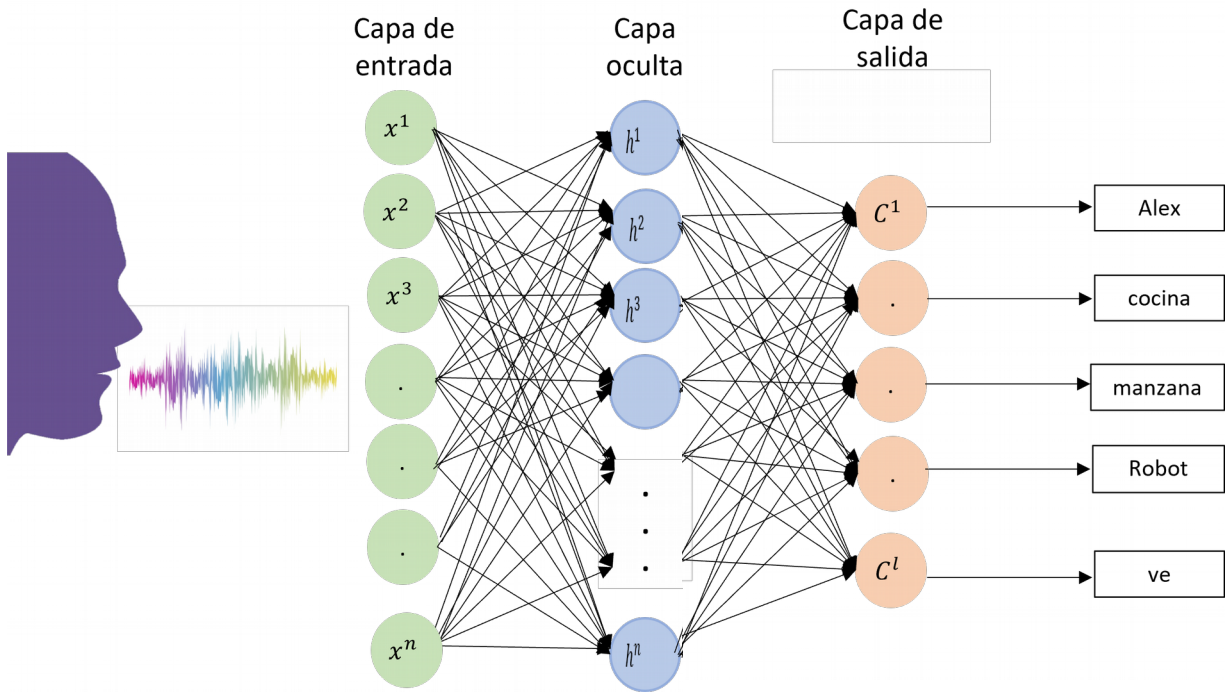


Fig 7. Diagrama de bloques del extractor de características MFCC.

REDES FEEDFORWARD

Agrega una o varias capas ocultas completamente conectadas entre las capas de entrada y salida y transforma la salida de la capa oculta a través de una función de activación.



$$H = \sigma(XW^{(1)} + b^{(1)})$$
$$O = H\,W^{(2)} + b^{(2)}$$

Fig 8. Diagrama básico de la red neuronal para clasificación de señales de voz.

REDES NEURONALES RECURRENTES

A partir de la relación entre las variables ocultas H_t y H_{t-1} de los pasos de tiempo adyacentes, sabemos que estas variables capturaron la información histórica de la secuencia hasta su paso de tiempo actual, al igual que el estado del paso de tiempo actual (Cruz, et al., 2007).

$$H = \varphi(XW_{xh} + H_{t-1}W_{hh} + b_h)$$
$$O = H_t W_{hq} + b_q$$

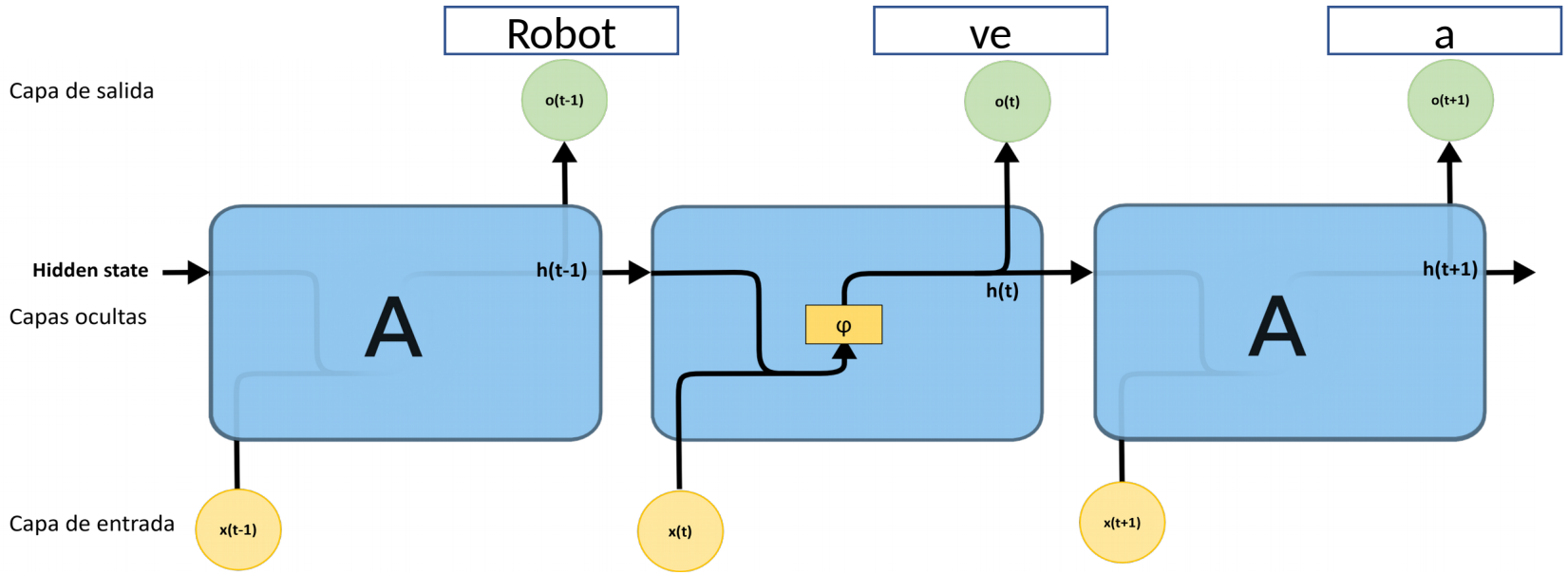


Fig 9. Diagrama de celda recurrente. Imagen modificada de: (Olah, 2015).

LONG-SHORT TIME MEMORY

El diseño de LSTM está inspirado en las puertas lógicas de una computadora. Los LSTM tienen tres tipos de puertas: puertas de entrada, puertas de olvido y puertas de salida que controlan el flujo de información.

Compuertas entrada, olvido y salida

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i)$$
$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f)$$
$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o)$$

Celda de memoria

$$\tilde{C}_t = tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c)$$
$$C_t = F_t \cdot C_{t-1} + I_t \cdot \tilde{C}_t$$

Estado oculto

$$H_t = O_t \cdot tanh(C_t)$$

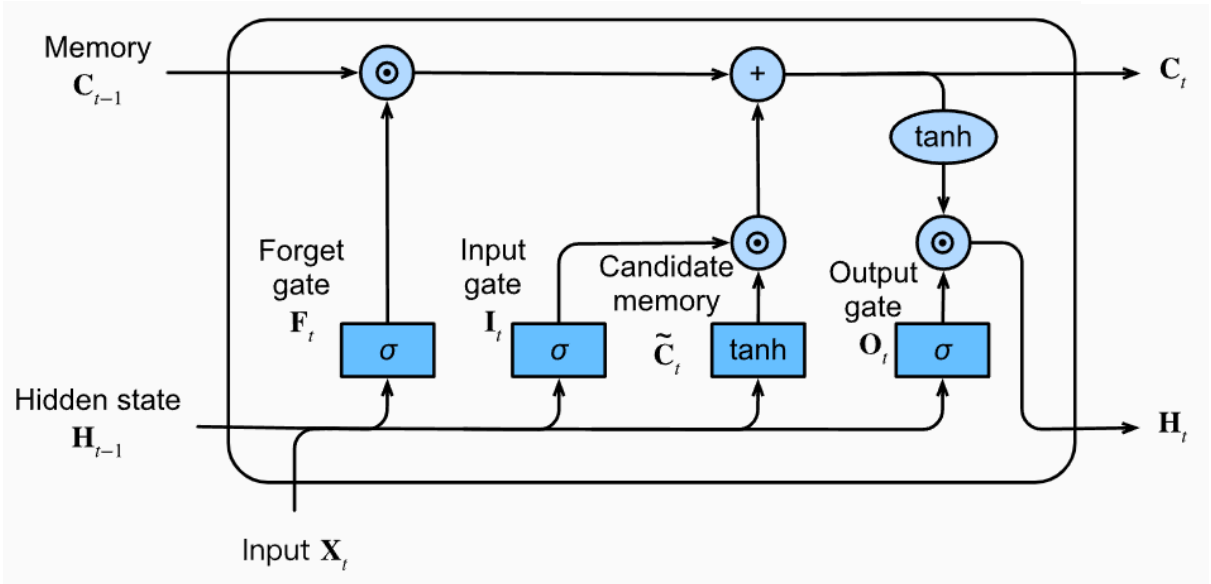


Fig 10. **Diagrama de celda LSTM.** ***Imagen recuperada de: (Zhang, et al., 2020).***

CORPUS

El conjunto de datos consta de 10 palabras grabadas con 40 repeticiones para cada una. Posteriormente este conjunto se dividió en 75% para entrenamiento y 25% para prueba.

PALABRAS AISLADAS

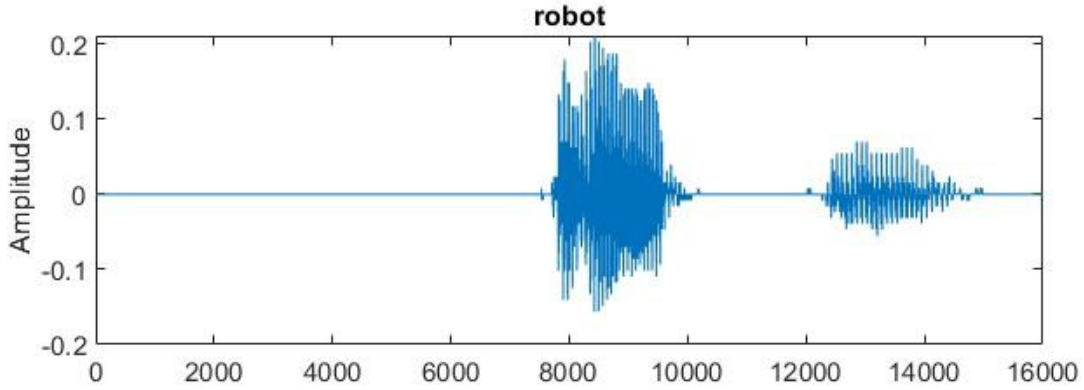


Fig 11. Ejemplo de señal de voz para la palabra aislada "robot".

L = Num. Bloques

13

ETIQUETA

robot

PRUEBAS DE CONCEPTO

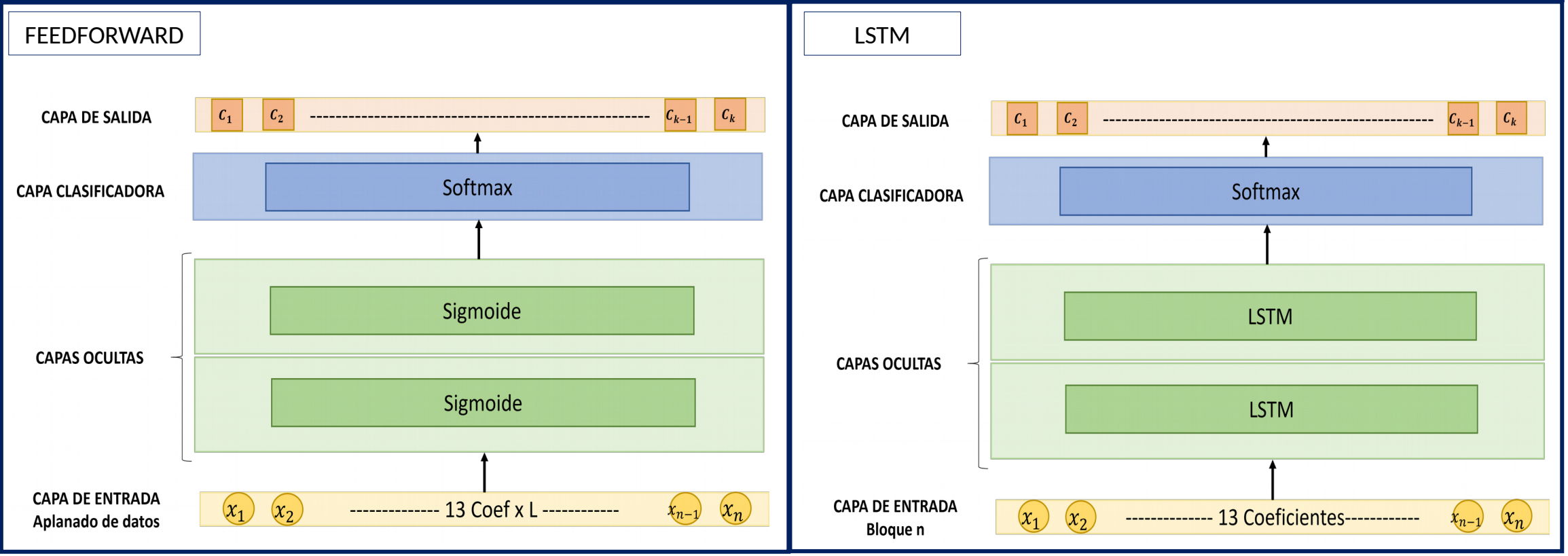


Fig 12. Diagrama básico de la arquitectura de las redes neuronales utilizadas durante las pruebas de concepto.

	FEEDFORWARD	LSTM
LPC	0.61	0.92
MFCC	0.96	1

Tabla I. Resultados de las pruebas de concepto.

COMANDOS DE VOZ

Los comandos se produjeron utilizando el **generador de comandos GPSR oficial** disponible en el repositorio de la ROBOCUP. Por otra parte, las señales de audio se generaron a partir de estos comandos y utilizando el GTTS.

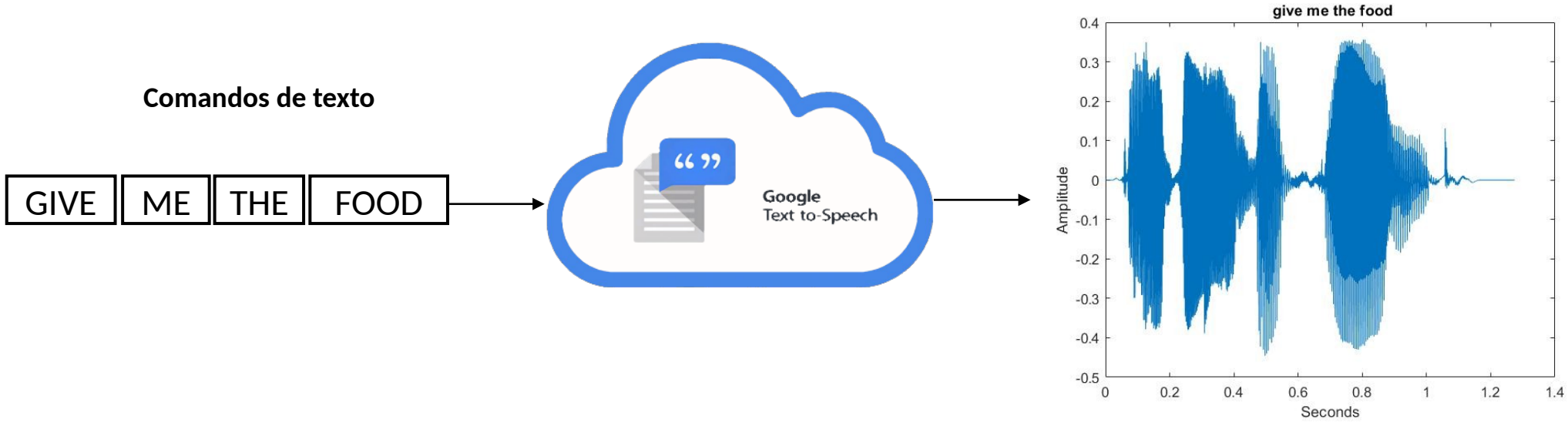


Fig 13. Diagrama de la obtención de señales de audio a partir de comandos de texto.

AUMENTADO DE DATOS

El corpus de entrenamiento consta de 20 comandos con 15 repeticiones para entrenamiento y 5 para prueba.

- Inyección de ruido a la señal.
- Cambio en el tiempo.

GO ~~TO THE~~ KITCHEN

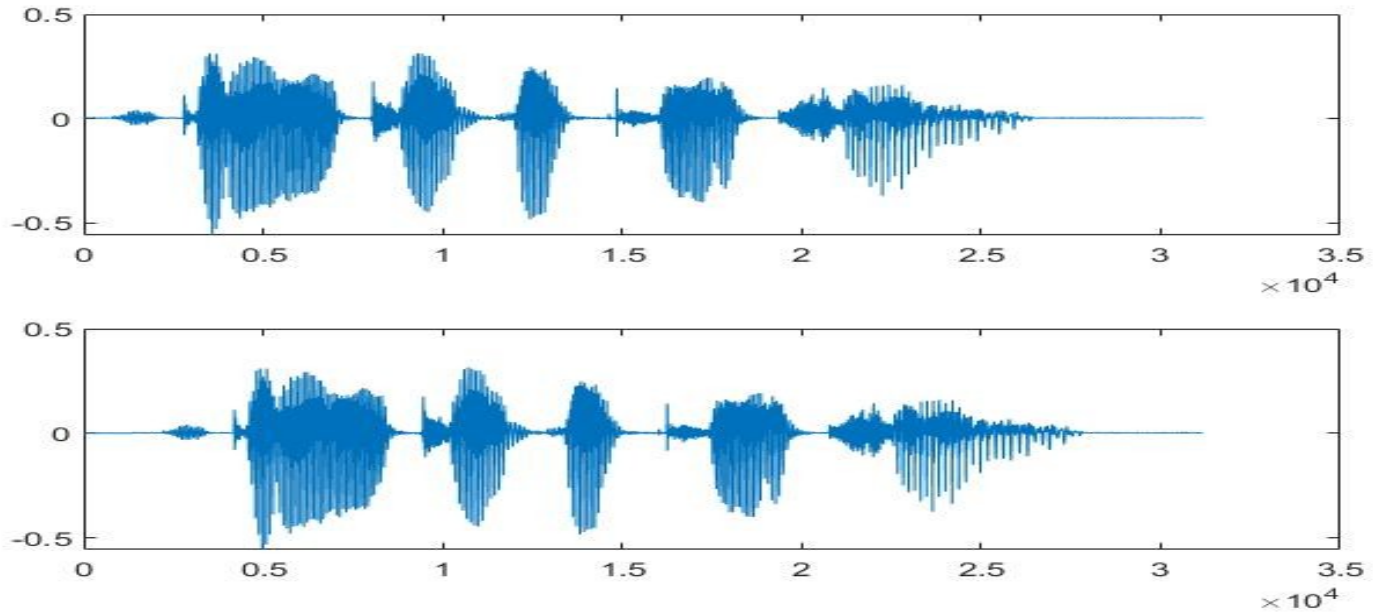


Fig 14. Aumentado de datos por inyección de ruido y corrimiento de la señal.

ETIQUETAS

Los comandos de texto fueron ingresados al modulo Parser Stanza el cual constituye una estructura de árbol de palabras a partir de la oración de entrada, las cuales representan dependencias sintácticas (Standford NPL Group, 2020).

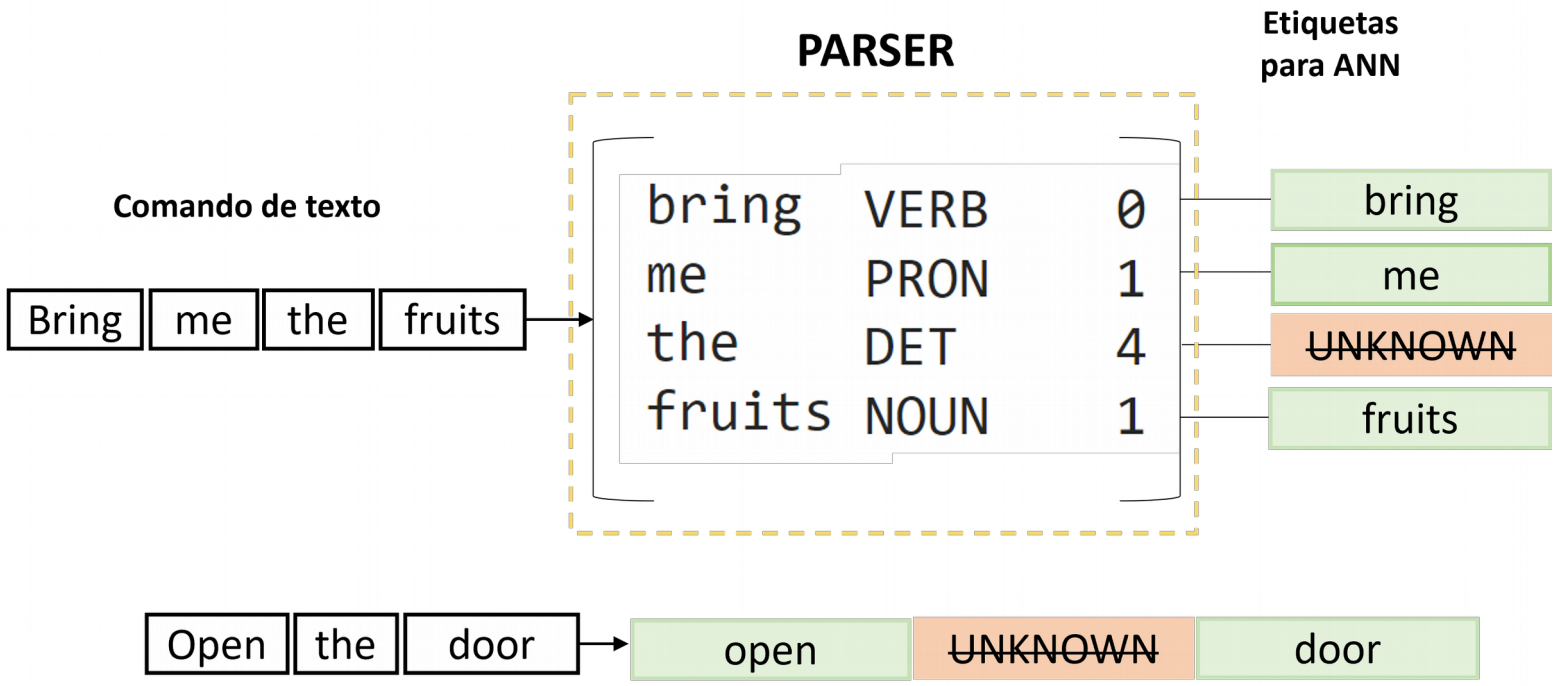
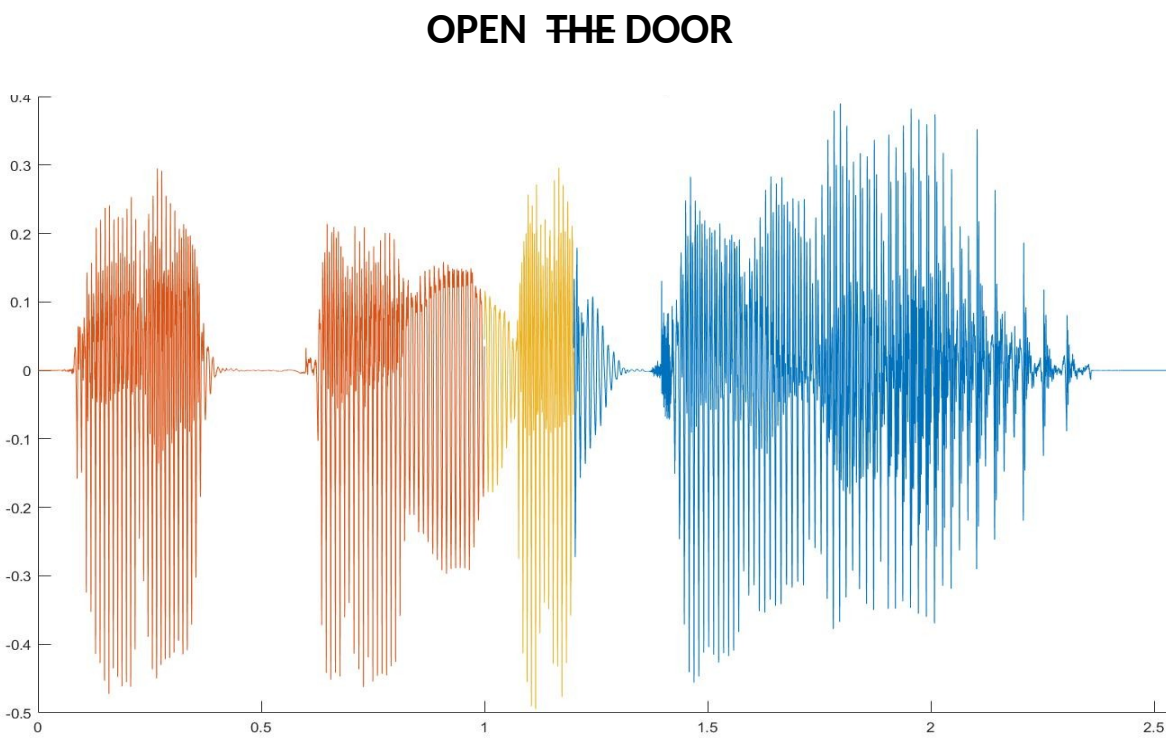


Fig 15. Diagrama de etiquetado de los comandos de texto.

ALINEACIÓN DE ETIQUETAS

Cualquier secuencia de la señal, representada por los vectores de características T_x corresponde a una asignación en la secuencia de etiquetas T_y.



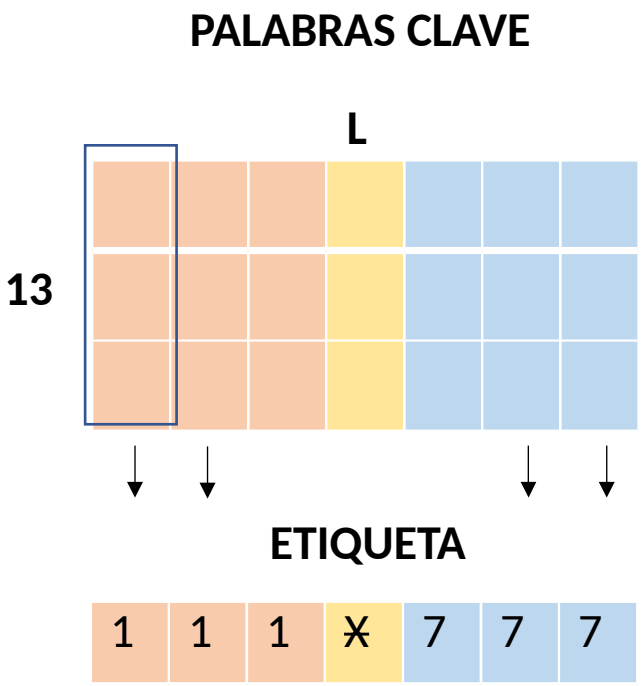


Fig 16. Etiquetado secuencia a secuencia de la señal.

RNN + COEFICIENTES MEL

La probabilidad condicional de las etiquetas en cada paso de tiempo se estima utilizando una RNN, cuyo resultado ingresa a una capa Softmax que predice una distribución de probabilidad sobre el conjunto de símbolos de salida.

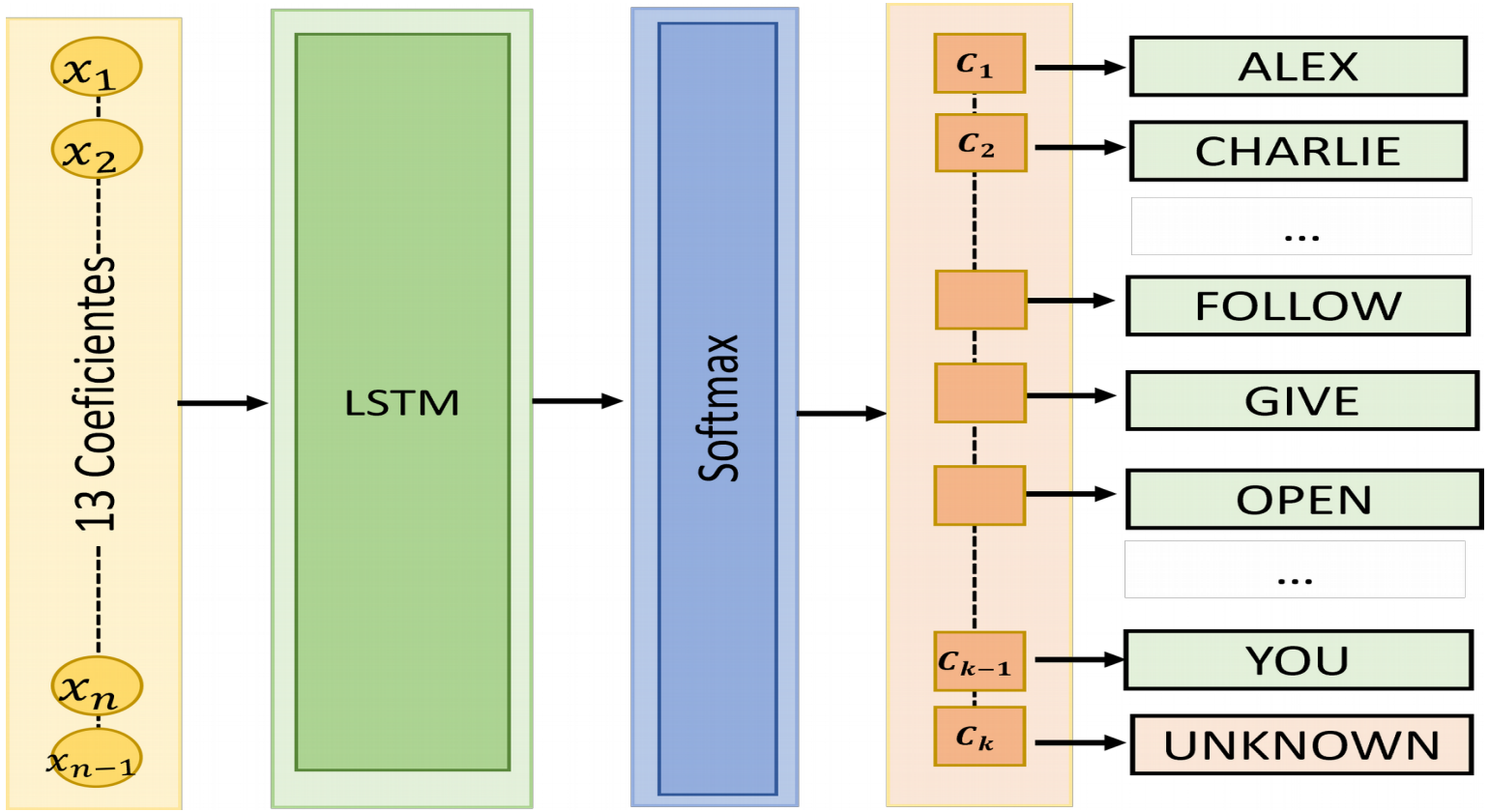


Fig 17. Diagrama básico de detección de palabras clave.

RECUPERACIÓN DE PALABRAS CLAVE

Las alineaciones tienen la misma longitud que la entrada. Para obtener las palabras claves de los comandos se sigue el siguiente proceso:

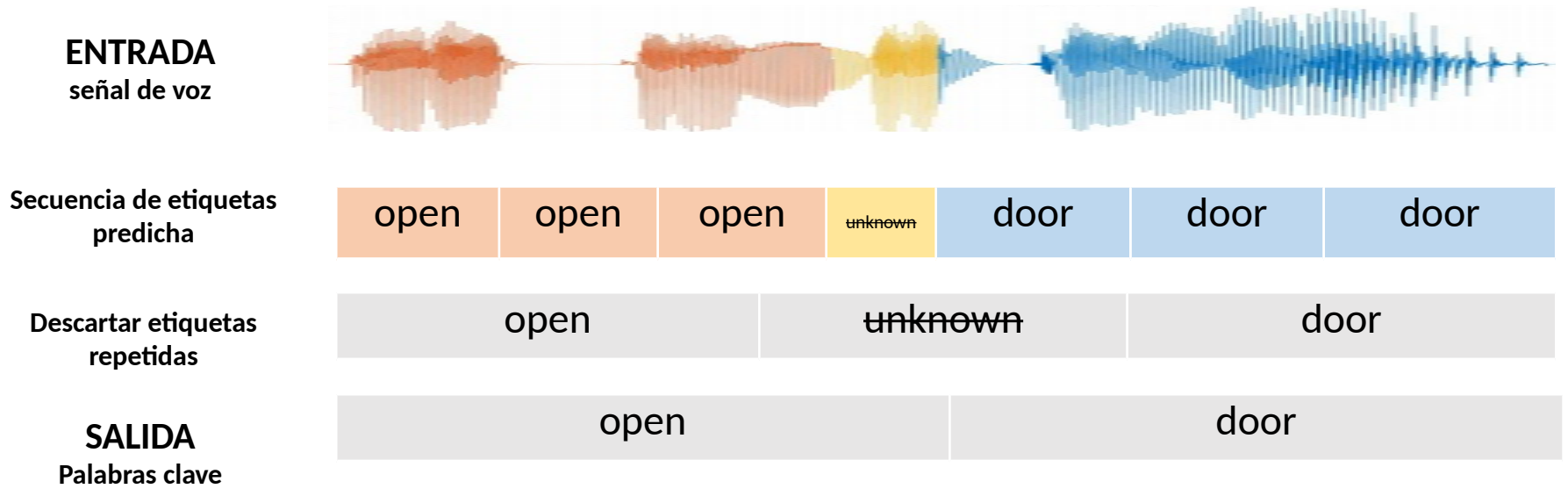


Fig 18. Diagrama del algoritmo de recuperación de palabras clave en un comando.

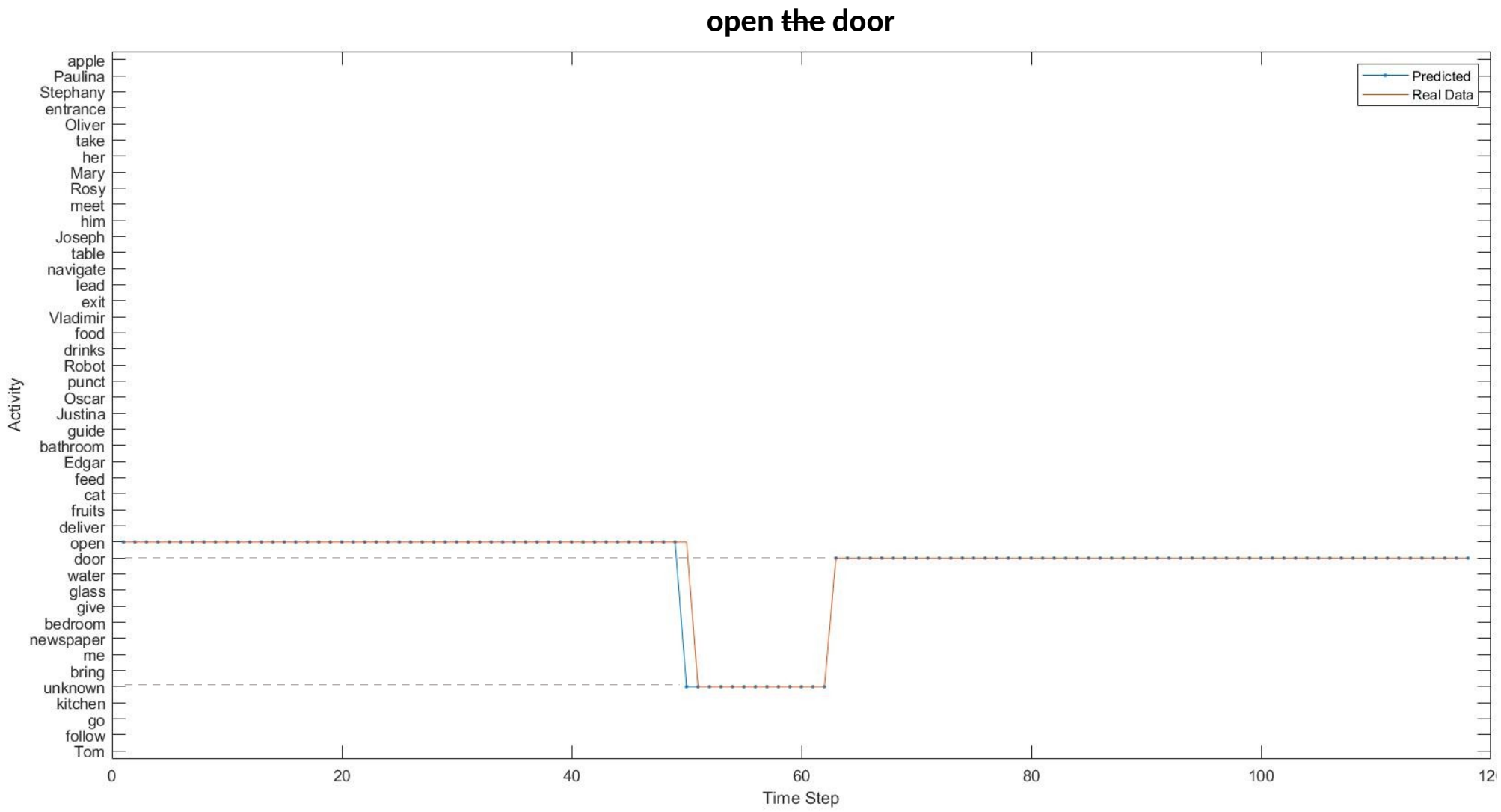


Fig 19. Reconocimiento de voz secuencia a secuencia.

Estas matrices representan los resultados obtenidos al comparar en cada estado de tiempo las salidas predichas del modelo y los comandos de referencia, la cual obtuvo una exactitud general de reconocimiento del 96.12%.

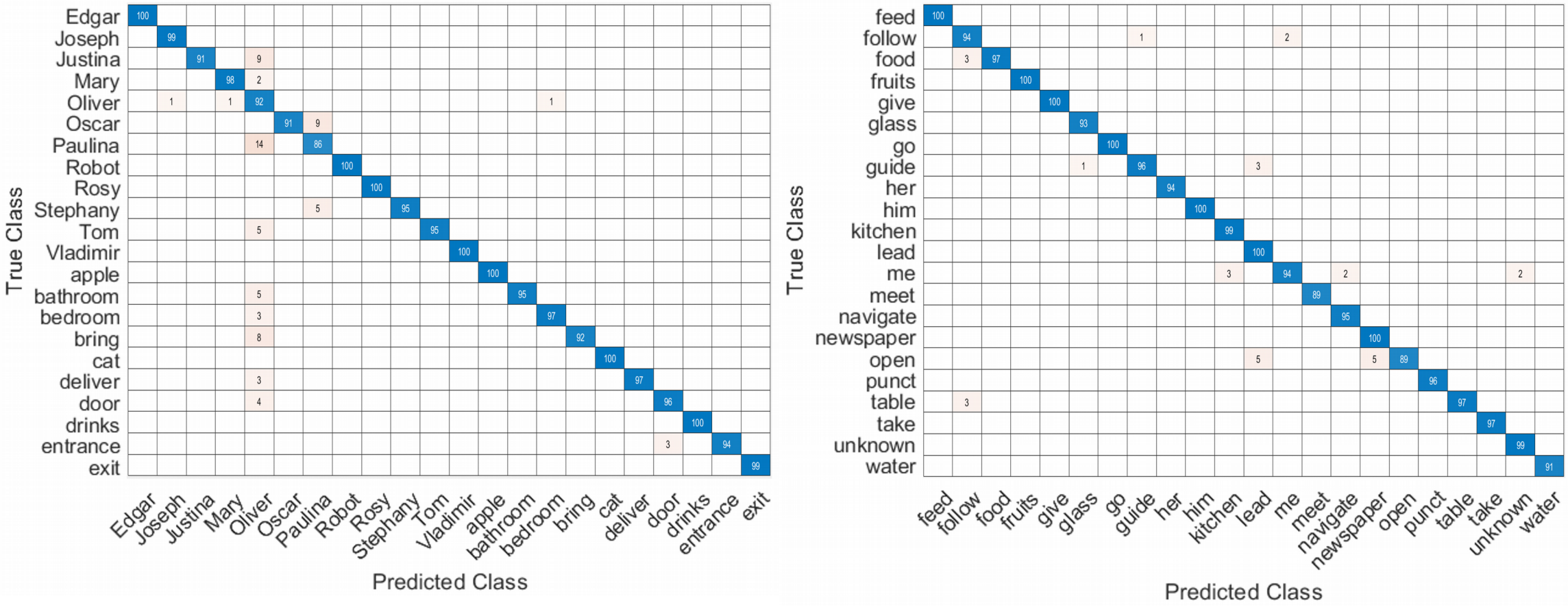


Fig 20. Matrices de confusión de los resultados obtenidos por la el modelo propuesto en cada estado de tiempo.

Robot, go ~~to the~~ kitchen, meet Paulina ~~and~~ deliver ~~an~~ apple ~~to~~ her

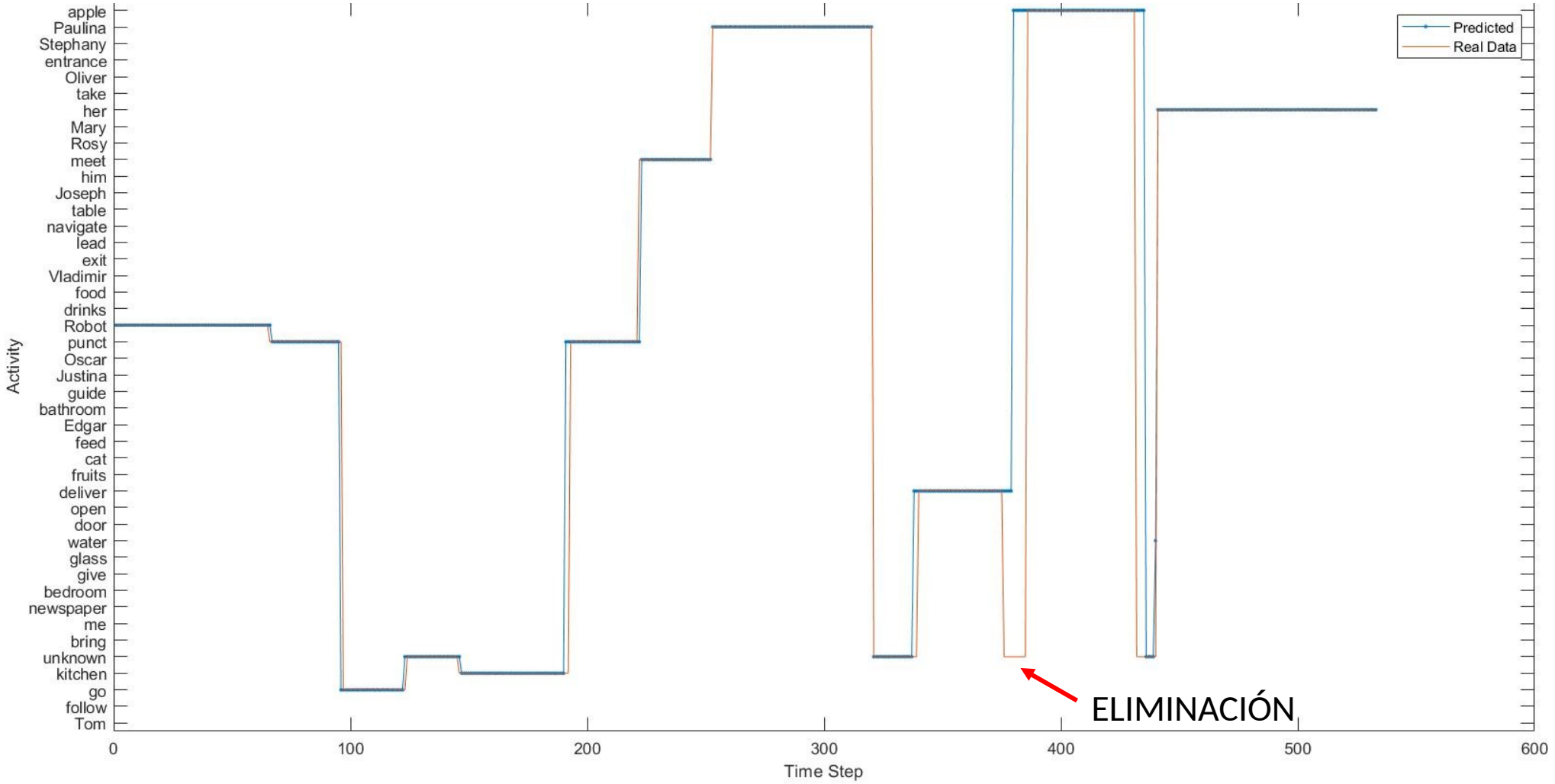


Fig 21. Reconocimiento de voz secuencia a secuencia. Ejemplo de error tipo eliminación.

WORD ERROR RATE (WER)

En ASR, se calcula la tasa de error de palabras (WER) entre la frase generada por el sistema y una frase de referencia *correcta*. Calcula el número mínimo de inserciones, borrados y sustituciones de una palabra por otra, necesarios para transformar una frase en otra.

$$WER = \frac{S + E + I}{N}$$

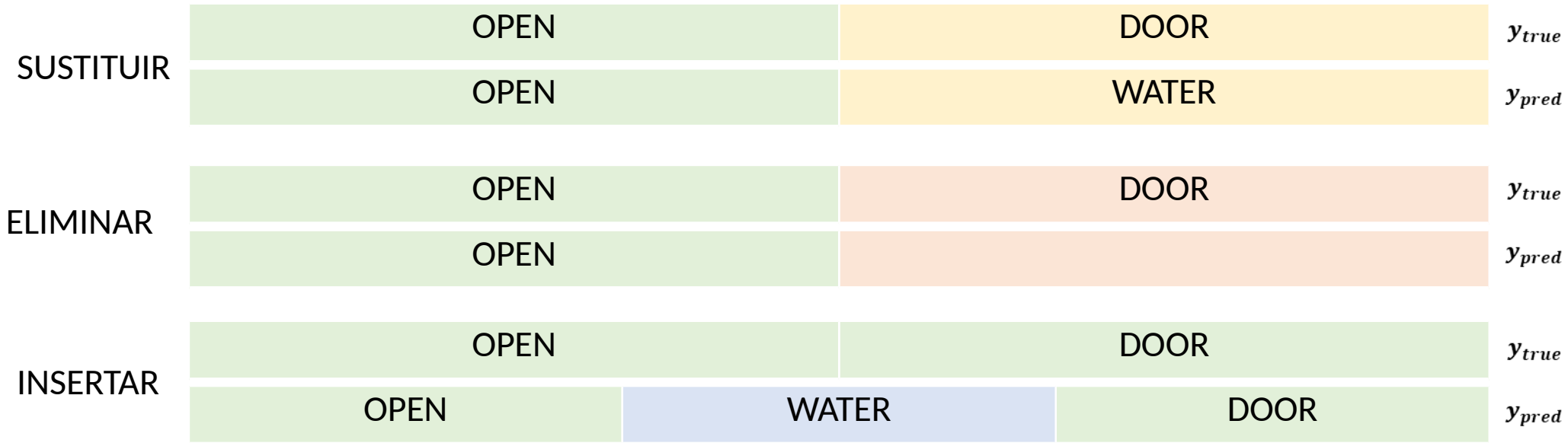


Fig 22. Diagrama con ejemplos de las penalizaciones de la tasa de error de palabras (WER) para un comando.

WER representa una medida más representativa (comparada con la anterior) para la tarea que se está realizando en este proyecto, ya que permite determinar el tipo de error que se está cometiendo.

$$WER = 0.0067$$

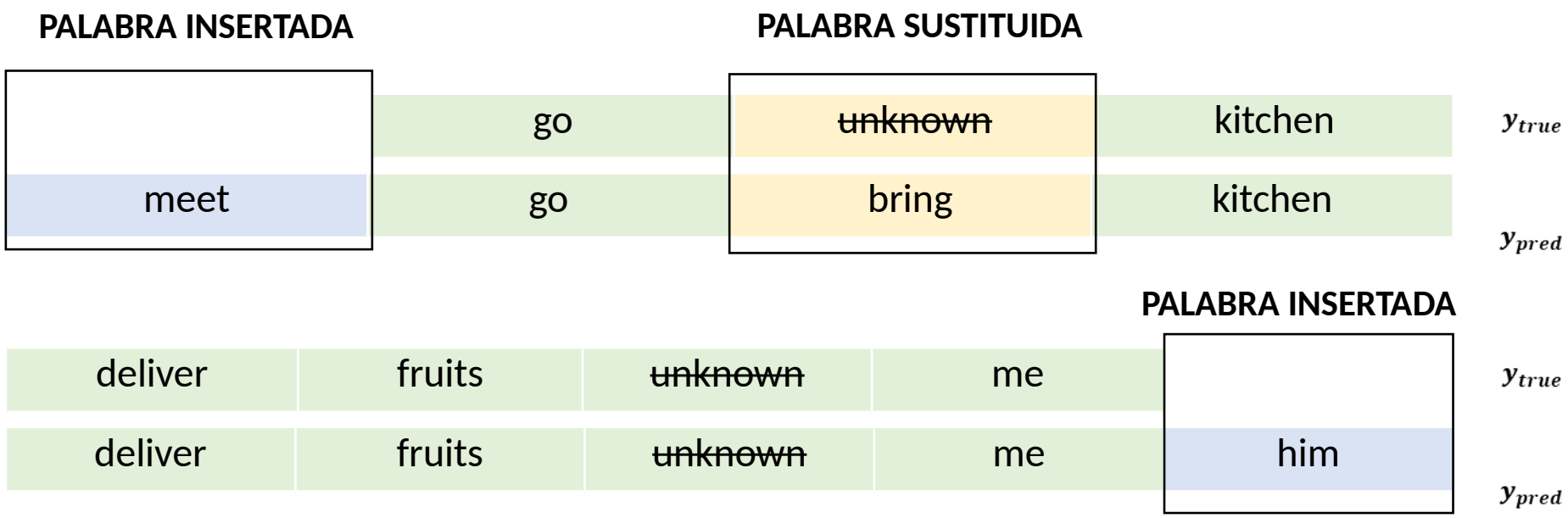
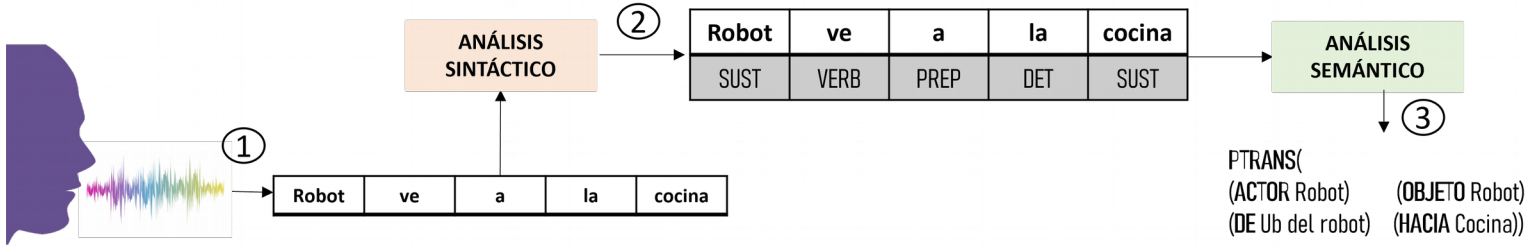


Fig 23. Ejemplos de los tipos de error contenidos en los comandos resultantes.

DEPENDECIA CONCEPTUAL

Finalmente, se realizaron pruebas para generar las estructuras de DC a partir de las palabras claves reconocidas por el sistema, utilizando estas palabras para completar los espacios de la plantilla de la primitiva con los roles de los participantes en la oración. En general el sistema obtuvo un porcentaje de acierto del 85% al momento de transformar las palabras clave.



[Sentence ROBOT, GO TO THE KITCHEN, MEET PAULINA AND DELIVER AN APPLE TO HER]

Robot	robot	NOUN	3	vocative
,	,	PUNCT	1	punct
go	go	VERB	0	root
kitchen	kitchen	NOUN	5	obj
meet	meet	VERB	3	parataxis
Paulina	Paulina	PROPN	5	obj
deliver	deliver	VERB	5	xcomp
apple	apple	NOUN	9	compound
her	she	PRON	7	obj

```
PTRANS((ACTOR Robot) (OBJECT Robot) (FROM Robots place) (TO kitchen))
PTRANS((ACTOR Robot) (OBJECT Robot) (FROM Robots place) (TO Paulinas place))
ATRANS((ACTOR Robot) (OBJECT apple) (FROM kitchen) (TO Paulina))
```

Fig 24. Ejemplos de las primitivas de DC obtenidas a partir de las palabras clave.

CARGA COMPUTACIONAL

El término tiempo real se utiliza ampliamente en muchos contextos, tanto técnicos como convencionales. Sin embargo, el Random House Dictionary of the English Language define como tiempo real al tiempo de respuesta apropiado que requiera el proceso que se está controlando. Específicamente en robótica, el tiempo de respuesta se mide desde el momento en que el usuario da un comando de voz hasta que el robot provee una respuesta (Laplante y Seppo, 2012). Sin embargo, esto involucra procesos adicionales al reconocimiento de voz.

El termino de tiempo real en este proyecto se acotó al tiempo de respuesta desde el reconocimiento de palabras a partir de una señal de voz hasta cuando se obtienen las palabras clave (sin ningún tipo de análisis consecuente).

Proceso	Carga computacional [ms]
Entrenamiento de la red	188,030 ± 98,217
Reconocimiento de voz	935 ± 574
Recuperación de KW	0.193 ± 0.135
Conversión CD	878.500 ± 253

***Tabla II*. Carga computacional de los procesos que componen el reconocimiento de voz.**

CONCLUSIONES

Generar de manera automática los comandos haciendo uso del generador de comandos oficial de la RoboCup, permite obtener una base de datos posible de replicar para futuras pruebas y proyectos.

Con base en las pruebas realizadas se pudo apreciar la importancia de una base de datos de calidad. Las pruebas mostraron que el desempeño del sistema tuvo una mejora proporcional al tamaño y la calidad de los datos contenidos.

Mediante las pruebas iniciales realizadas fue posible determinar una mejora en el reconcomiendo de voz al utilizar MFCC en comparación a los LPC. Por otra parte, fue posible observar que los modelos recurrentes superan significativamente el desempeño de los modelos FeedForward en la tarea de reconocimiento de voz.

El modelo final propuesto (secuencia-secuencia) representa una ventaja para el reconocimiento de palabras clave, con respecto a las arquitecturas mostradas en las pruebas de concepto, ya que permite realizar el reconocimiento a partir de una señal de voz continua.

Las pruebas de CD mostraron que es posible utilizar las palabras clave reconocidas por un ASR para completar los elementos de una primitiva. Lo que permite construir un puente entre el lenguaje natural del humano y el lenguaje basado en reglas utilizado en un robot de servicio.

CONSIDERACIONES A FUTURO

- Se sugiere una mejora en el contenido del corpus; aumentar la cantidad de datos, incluir nuevos hablantes, incluir nuevas técnicas de aumentado de datos, lo cual permitirá aumentar la robustez del sistema de reconocimiento.

- Por otra parte, una de las tareas más laboriosas durante la realización del proyecto consistió en el etiquetado de la señal en cada estado de tiempo, este es un proceso realizado de manera manual por lo que requiere una gran cantidad de tiempo. Se sugiere:

  - Optimizar el proceso de etiquetado.
  - Abordar la tarea del reconocimiento de voz utilizando arquitecturas secuencia a secuencia mas complejas.

REFERENCIAS

- *Afzal, H., Sheeraz, M., and Mark, A. (2010). A novel approach for MFCC feature extraction. In Conference: Signal Processing and Communication Systems (ICSPCS), 2010 4th International Conference on.*
- *Bradbury, J. (2000). Linear Predictive Coding.*
- *Cruz, I., Salazar, S., Rodríguez, A., Grau, R., and García, M. (2007). Redes neuronales recurrentes para el análisis de secuencias. Revista cubana de ciencias informáticas, 1(4).*
- *Laplante, P. A. and Seppo, J. O. (2012). REAL-TIME SYSTEMS DESIGN AND ANALYSIS. A JOHN WILEY and SONS, INC., PUBLICATION, New Jersey.*
- *Olah, C. (2015). Understanding lstm networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs.*
- *Rabiner, L. R., Levinson, S.E., Rosenberg A.E., and Wilpon, J.G. (1979). Speaker Independent Recognition of Isolated Words Using Clustering Techniques. IEEE Trans. Acoustics, Speech and Signal Proc, (ASSP-27).*
- *Ramírez, J. M., Montalvo, A., and Calvo, J. (2019). Evaluaci´on de rasgos acústicos para el reconocimiento automático del habla en escenarios ruidosos usando kaldi. Ingeniería electrónica, automática y comunicaciones, 40(3).*
- *Random-House-Dictionary (1987). Random House Dictrionary of the English Language. 2nd edition.*
- *Rich, E. and Knight, K. (1991). Artificial intelligence. McGraw-Hill, Inc., NJ, USA, second edition.*
- *Savage, J., Rosenblueth, D., Matamoros, M., Negrete, M., Contreras, L., Cruz, J., Martell, R., Estrada, H., and Okada, H. (2019). Semantic reasoning in service robots using expert systems. Robotics and autonomous systems, (114). https://doi.org/10.1016/j.robot.2019.01.007.*
- *Zhang, A., Lipton, Z., Li, M., and Smola, A. (2020). Dive into Deep Learning. https://d2l.ai/index.html.*