Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Inclusión léxica y sintáctica en modelos de etiquetado offline de transiciones de turno

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Jonathan Scherman

Director: Pablo Brusco

Buenos Aires, 2024

INCLUSIÓN LÉXICA Y SINTÁCTICA EN MODELOS DE ETIQUETADO OFFLINE DE TRANSICIONES DE TURNO

En una conversación hablada entre dos personas, existe una noción implícita de manejo de turnos que permite un intercambio fluido de mensajes. Esto sucede a través de distintos tipos de señales (acústicas, prosódicas, léxicas e incluso gestuales) que los hablantes producen y que, de manera combinada, afectan la manera en la que se desarrollarán los turnos de la conversación. Por ejemplo, se ha estudiado cómo el uso de palabras de relleno (tales como "em..." o "este...") suele indicar que el hablante tiene la intención de mantener el turno.

Entender cuál es la dinámica del manejo de turnos y construir sistemas que puedan identificar y clasificar distintas transiciones de manera automática ha ganado mucha relevancia en las últimas décadas debido a las múltiples aplicaciones prácticas en el mundo real, siendo los sistemas de diálogo hablado (como Alexa o Siri) y el análisis de grandes volúmenes de datos sus principales aplicaciones. En particular, varios estudios han mostrado que, en adición a las características acústico-prosódicas, las características léxicas y sintácticas de la conversación contienen información relevante para entender la mecánica del manejo de turnos.

En esta tesis, nos enfocamos en el desarrollo de una herramienta de etiquetado *offline* de transiciones de turno en conversaciones completas de tipo humano-humano, para lo cual construimos distintas variantes de modelos multimodales sobre un corpus compuesto por conversaciones diádicas orientadas a tareas, en inglés y español. Partimos de un modelo preexistente basado en redes neuronales recurrentes bidireccionales, que extendemos con atributos léxicos y sintácticos. Además, consideramos modelos alternativos basados en arquitecturas multiescala. Para cada uno de los modelos desarrollados, evaluamos la capacidad de generalización en conversaciones tanto en el mismo idioma en el que fueron entrenados como en otros.

Los resultados obtenidos muestran que la inclusión de estos atributos otorgó mejoras para algunas de las variantes propuestas de hasta un 15 % relativo en el subconjunto de desarrollo y un 9 % relativo en el subconjunto de control al ser evaluados en conversaciones nuevas del mismo idioma, mientras que observamos disminuciones en el rendimiento al variar entre idiomas. Por su parte, no pudimos evidenciar mejoras en la propuesta multiescala, aunque creemos que tiene potencial para hacerlo.

Palabras claves: Manejo de Turnos, Diálogo Hablado, Inglés, Español, Aprendizaje Automático, Redes Neuronales, BERT, Atributos Léxicos, Atributos Sintácticos.

LEXICAL AND SYNTACTIC INCLUSION IN OFFLINE TURN-TAKING LABELING MODELS

During dyadic spoken conversations, there is an implicit notion of turn-taking that allows for a smooth exchange of messages between both speakers. This happens through different kinds of signals (acoustic, prosodic, lexical, and even gestural) that speakers produce and that, when combined, affect the way the turns of the conversation develop. For example, it is well known that the use of filler words (such as "um..." or "like...") often indicates that the speaker intends to hold the floor.

Understanding the dynamics of turn-taking and building systems that can automatically identify and classify different transitions has gained much relevance in recent decades due to the numerous practical real-world applications, being spoken dialogue systems (such as Alexa or Siri) and the analysis of large volumes of data its main applications. In particular, several studies have shown that, in addition to acoustic-prosodic cues, the lexical and syntactic characteristics of the speakers hold relevant information for understanding the mechanics of turn-taking.

In this thesis, we focus on the development of an *offline* turn-taking transition labeling tool for processing human-human conversations end-to-end, for which we built different variants of multimodal models using a task-oriented corpus of dyadic conversations in english and spanish. We start from a pre-existing bidirectional recurrent neural network model, which we later extend by including lexical and syntactic features. Additionally, we consider alternative models based on multiscale architectures. For each of the developed models, we evaluate the generalization capabilities in conversations from the same language they were trained on and in others as well.

The obtained results show that the inclusion of these features yields improvements of up to 15 % relative in the development subset and 9 % relative in control for some of the proposed variants in new conversations from the same language, while we observed performance decreases while switching between languages. Furthermore, we could not evidence improvements in the multiscale approach, although we believe it has the potential to do so.

Keywords: Turn-Taking, Spoken Dialogue, English, Spanish, Machine Learning, Neural Networks, BERT, Lexical Attributes, Syntactic Attributes.

AGRADECIMIENTOS

A mi director, Pablo, que me tuvo una paciencia infinita y estuvo siempre presente para darme una mano cuando la necesité, y porque básicamente fue quien hizo que este trabajo haya sido posible. No solo fuiste uno de los mejores ayudantes que tuve durante la carrera sino que, además, sos un increíble director y gran persona.

A mi familia, que me acompañó durante toda la carrera y fue partícipe tanto de los buenos como de los malos momentos.

A Tami, quién más de cerca vivió mi día a día con la tesis y que me bancó y motivó para terminarla.

A todos los increíbles docentes que he tenido a lo largo de la carrera, que me despertaron el interés en aprender y quienes hicieron posible que hoy esté en este lugar.

A mis amigos y todas las personas que de alguna u otra forma me acompañaron durante este camino e hicieron que haya sido más ameno.

Índice general

1.. Introducción 1
1.1. Trabajo previo 3
1.2. Estructura del informe 4

2.. Preliminares 5
2.1. Estudio de transiciones de turno 5
2.2. Redes neuronales *feed-forward* 6
2.3. Redes neuronales recurrentes 8
2.4. Redes neuronales recurrentes bidireccionales 9
2.5. Redes neuronales recurrentes GRU 10
2.6. BERT 12

3.. Fuentes de datos 15
3.1. Juegos de Objetos 15
3.1.1. Columbia Games Corpus 16
3.1.2. UBA Games Corpus 17

4.. Desarrollo de modelo predictivo de transiciones de turno 19
4.1. Unidad de análisis 19
4.2. Separación de datos 19
4.3. Modelo acústico - *bigru[acoustic]* 20
4.4. Modelo con texto - *bigru[acoustic+text]* 22
4.5. Modelo multiescala - *ms-bigru[acoustic+text]* 24

5.. Resultados 27
5.1. Replicación de resultados del modelo secuencial 29
5.2. Resultados en desarrollo 31
5.2.1. Modelo acústico - *bigru[acoustic]* 31
5.2.2. Modelo con texto - *bigru[acoustic+text]* 32
5.2.3. Modelo multiescala - *ms-bigru[acoustic+text]* 34
5.2.4. Mejores modelos 37
5.3. Resultados en control 37
5.3.1. Mejores modelos de desarrollo 38
5.3.2. Mejor modelo, reentrenado con atributos multilenguaje 39

6.. Conclusiones y trabajo futuro 43

1. INTRODUCCIÓN

En una conversación hablada entre personas existe una noción implícita de **manejo de turnos**, que implica que los participantes se alternan para hablar de manera ordenada y así tener un intercambio fluido de mensajes. A medida que los turnos ocurren, se han observado ciertos patrones que permiten al interlocutor anticipar la intención del hablante, y que en la literatura se conocen como PISTAS DE TRANSICIONES DE TURNO. Estas pistas ocurren hasta medio segundo antes de las transiciones de turno (Duncan, 1972) y pueden ser de tipo léxicas, prosódicas, gestuales, o incluso relacionadas con la mirada o la respiración de los hablantes. Por ejemplo, rápidas caídas o subidas en el tono de la voz suelen ser fuertes indicadores de cambios de turno (Gravano and Hirschberg, 2011). Estos estudios, así como varios otros, aportan evidencia de cómo las palabras utilizadas y sus características gramaticales contienen pistas relevantes para las transiciones de turno. Entender cómo sucede la dinámica de turnos y construir sistemas que puedan identificar automáticamente los distintos tipos de transiciones son problemas que han ganado mucha relevancia en la actualidad.

Una de las ramas del manejo de turnos más estudiadas consiste en el análisis *online* de conversaciones para la predicción en tiempo real de fin de turnos (DeVault et al., 2015; Maier et al., 2017; Masumura et al., 2017; Liu et al., 2017; Chowdhury, 2017; Skantze, 2017; Lala et al., 2018; Roddy et al., 2018; Masumura et al., 2018; Lala et al., 2019; Chowdhury et al., 2019; Razavi et al., 2019; Ekstedt and Skantze, 2020; Ishii et al., 2022; Chang et al., 2022; Jiang et al., 2023). La popularidad de este enfoque se debe a las múltiples aplicaciones prácticas en el mundo real en el contexto del desarrollo de sistemas de diálogos hablados —como Alexa o Siri, desarrollados por Amazon y Apple, respectivamente—. Poder predecir de manera certera el fin de un turno permite al sistema intervenir en la conversación en el momento correcto y de esta manera evitar interrupciones o demoras innecesarias en sus respuestas, lo cual deriva en que estos sistemas sean capaces de manejar las conversaciones con mayor fluidez.

Por otro lado, en el análisis *offline* de las conversaciones, no solo se considera la información disponible hasta el momento en que ocurren las transiciones entre los turnos, sino que se cuenta con un contexto más general que puede contener información proveniente de la grabación completa. Esto resulta particularmente útil para tareas como la de categorizar automáticamente el **tipo de una transición entre turnos**, lo cual consiste en la identificación y clasificación de cada una de estas transiciones para entender si estas se tratan de cambios de turno suaves, interrupciones, señales de denotación de atención sin intención de tomar el turno (*backchannels*), o incluso si el hablante hizo una breve pausa para luego continuar con su turno.

Contar con herramientas que evalúan conversaciones completas permite el análisis de grandes volúmenes de datos para comprender la dinámica en la que se desarrollan las conversaciones, y de esta manera responder preguntas tales como *¿Es verdad que el hablante A interrumpió con mayor frecuencia que el hablante B? ¿Existe alguna relación entre la cantidad de interrupciones y el género, edad o cierta característica coyuntural de una persona? ¿Cómo se relaciona la cantidad de interrupciones que produce un agente de ventas con la cantidad de productos vendidos?* Asimismo, las etiquetas de cambio de turno pueden ser utilizadas como atributos en otros sistemas. Por ejemplo, el trabajo de Inoue et al. (2024)

presenta un esquema de evaluación para evaluar objetivamente la efectividad de sistemas de diálogo hablados, entre los que se incluye como métrica la *cantidad de backchannels por minuto*, definiendo los *backchannels* como turnos en los que se dijeron ciertas palabras en específico, las cuales no necesariamente reflejan la naturaleza compleja de estas señales. En este sentido, un sistema de etiquetado automático del tipo de turnos podría ser una herramienta que posibilite la extracción más precisa de estos fenómenos sin la necesidad de anotaciones humanas.

En esta tesis, nos proponemos contribuir al etiquetado automático offline de transiciones de turno abordando distintas propuestas de modelado que incluyan tanto atributos acústico-prosódicos de la señal, como atributos léxicos y sintácticos. Para ello, consideramos el modelo secuencial propuesto en Brusco and Gravano (2023) como punto de partida —una red neuronal recurrente bidireccional que realiza el etiquetado en base a atributos acústico-prosódicos—, el cual extendemos a través de una serie de experimentos para incluir: (1) *embeddings* de las palabras, obtenidos a partir de modelos preentrenados de la familia de modelos BERT (Devlin et al., 2018b); (2) las categorías gramaticales de las palabras (*part-of-speech*); (3) información de contexto adicional, como por ejemplo si la palabra fue pronunciada de manera completa o parcial; (4) una adaptación de arquitecturas multi-escala presentadas en Roddy et al. (2018), diseñadas especialmente para la incorporación de palabras en modelos acústico-prosódicos. Estas alternativas propuestas son comparadas en conversaciones no utilizadas durante la fase de entrenamiento en dos idiomas distintos: español e inglés.

Cabe mencionar que, dado que el conjunto de datos que utilizaremos se basa en diálogos orientados a tareas —tal como se describe en la sección 3—, es posible que introducir información relacionada con las palabras impacte en la capacidad de generalización de los modelos, especialmente en aquellos diálogos no orientados a tareas. Por ejemplo, es posible que el modelo aprenda que luego de la palabra "mouse" ocurre un cambio de turno en el contexto de los diálogos utilizados, lo cual no necesariamente ocurre en una conversación libre y sin restricciones. A pesar de esto, creemos que tiene sentido investigar el impacto de incluir esta información, puesto que en el futuro podríamos contar con otros conjuntos de diálogos de distinta naturaleza.

A lo largo de este trabajo, intentaremos responder las siguientes preguntas de investigación relacionadas al uso de características lingüísticas en el contexto de modelado de manejo de turnos:

(P_1) ¿Es posible extender el modelo original de forma tal que reconozca, además de los patrones acústico-prosódicos, patrones en las transcripciones de las conversaciones? ¿Cómo afecta a su rendimiento?

(P_2) ¿Cómo impacta construir un modelo con una arquitectura diseñada específicamente para concatenación de series temporales multimodales?

(P_3) ¿Cómo se comportan estos modelos ante conversaciones nuevas, que no hayan sido vistas durante su entrenamiento?

(P_4) ¿Cómo se comportan estos modelos ante conversaciones en idiomas distintos a los que fueron utilizados para su entrenamiento?

(P_5) ¿Podremos modificar los modelos existentes de forma tal que el rendimiento en conversaciones en distintos idiomas mejore y que, a su vez, mantenga resultados similares para conversaciones en el mismo idioma?

1.1. Trabajo previo

Existen una variedad de trabajos fundacionales que exploraron tanto el rol de la información acústico-prosódica, como el de la información léxica y sintáctica de los hablantes para la dinámica de manejo de turnos. A continuación, enumeramos algunos de los trabajos sobresalientes en esta última dirección.

En Ward and Tsukahara (2000), los autores exploraron el uso de palabras de relleno (tales como "uh" o "um") en las conversaciones, y encontraron que no solo sirven el propósito de rellenar los espacios sino que también comunican la intención del hablante de continuar su turno. En Gravano and Hirschberg (2011), los autores identificaron una serie de pistas de transiciones de turno en diálogos provenientes de conversaciones en inglés norteamericano. Entre otros hallazgos, los autores muestran que (1) las palabras de relleno suelen indicar que el hablante mantendrá el turno (lo cual aporta evidencia sobre el trabajo de Ward and Tsukahara (2000), mencionado anteriormente); (2) oraciones que finalicen con ciertos bigramas de PoS suelen ser pistas de invitación a *backchannels*; (3) la compleción textual es un fuerte indicador de cambio de turno (condición necesaria pero no suficiente); y (4) en general, la acumulación de pistas de transiciones de turnos (tanto acústico-prosódicas como léxicas) incrementa la probabilidad de cambios de turno (hipótesis de Duncan (1972)).

A lo largo del tiempo, se han publicado una variedad de trabajos en los cuales se desarrollan modelos que consideran la información léxica y sintáctica para la predicción de transiciones de turno, usualmente con un enfoque *online* o de tiempo real. En su mayoría (DeVault et al., 2015; Maier et al., 2017; Masumura et al., 2017; Liu et al., 2017; Skantze, 2017; Lala et al., 2018; Roddy et al., 2018; Masumura et al., 2018; Lala et al., 2019; Chang et al., 2022), se basan en redes neuronales recurrentes (RNNs) con algún mecanismo de memoria, tales como LSTM o GRU. En Razavi et al. (2019), por el contrario, utilizan modelos basados en SVM, árboles de decisión y *Gaussian Naive Bayes*. Asimismo, otros trabajos como Chowdhury (2017); Chowdhury et al. (2019) utilizan SVM con redes neuronales *feed-forward*.

Por ejemplo, en Skantze (2017), el autor desarrolló modelos basados en RNNs de tipo LSTM, a partir de diálogos humano-humano orientado a tareas, en inglés. En este caso, únicamente las etiquetas gramaticales —*part-of-speech* (PoS) tags— se incluyeron como parte de los atributos del modelo. Al año siguiente, en Roddy et al. (2018), extendieron este trabajo planteando una alternativa multimodal (léxica, visual y acústica) *multiescala*, en la que se utilizan redes recurrentes LSTM para capturar los patrones de cada modalidad por separado. El modelo fue entrenado en base a un conjunto de diálogos diádicos abiertos (restringido a algunos temas en específico) y orientados a tareas. Una característica fundamental de este trabajo es que incluyeron en el diseño de la red la utilización de intervalos de tiempo de distinta duración por modalidad, considerando tanto intervalos regulares como irregulares. De esta manera, lograron incluir en el modelo series temporales que ocurren a distintas velocidades.

Junto a la aparición de la arquitectura GPT, surgieron otros trabajos relacionados, tales como el de Ekstedt and Skantze (2020), en el cual los autores desarrollaron un modelo basado en la arquitectura GPT-2 y utilizaron únicamente atributos léxicos de los hablantes. El modelo fue entrenado sobre un diverso conjunto de diálogos en inglés y otorgó mejores resultados frente a las alternativas allí propuestas: un modelo LSTM y otro modelo estadístico simplificado basado en bigramas PoS. A su vez, en Jiang et al. (2023), los autores extendieron este modelo condicionando la respuesta de los hablantes al realizar la siguiente

predicción, para así evitar potenciales cambios de turno prematuros.

Otro caso particular es el trabajo de Ishii et al. (2022), en donde estudiaron y encontraron beneficioso modelar, además, el factor de la predisposición de tomar el turno por parte de los hablantes. Para ello, desarrollaron modelos multimodales de carácter multitarea, entrenados para resolver las tareas de predicción de transiciones de turno y predisposición de los hablantes de manera conjunta, sobre un conjunto de conversaciones diádicas en japonés de diálogo libre pero restringido a ciertos temas en específico.

En otra línea, en el trabajo de Brusco and Gravano (2023) se analizan estrategias de modelado para el etiquetado *offline* de transiciones de turno, con el enfoque en procesar conversaciones de punta a punta en base a un conjunto de diálogos diádicos orientados a tareas en inglés, español y eslovaco. Con este fin, los autores proponen modelos basados en Random Forest y RNNs bidireccionales con células GRU, que incluyen atributos acústico-prosódicos (*pitch*, *shimmer*, *jitter*, *logHNR* e intensidad) y contextuales (tales como los géneros de los hablantes o el tiempo transcurrido desde el comienzo del turno, entre otros). En ambos casos se evaluó la capacidad de generalización para conversaciones nuevas en distintos idiomas a los que fueron utilizados durante su entrenamiento.

Al día de hoy, no conocemos trabajos publicados que propongan herramientas para el etiquetado *offline* de transiciones de turno que consideren la información léxica o sintáctica de las conversaciones.

1.2. Estructura del informe

Incluyendo este capítulo introductorio, esta tesis se compone por seis capítulos.

En el Capítulo 2 (“Preliminares”), presentamos los principales conceptos utilizados a lo largo de este trabajo. Este capítulo incluye una introducción al estudio de transiciones de turno; redes neuronales junto a algunas de sus variantes; y finalmente, una introducción al modelo de embeddings utilizado para codificar la información textual: BERT.

En el Capítulo 3 (“Fuentes de datos”) describimos el corpus que utilizaremos a lo largo de este trabajo para el desarrollo de los modelos.

En el Capítulo 4 (“Desarrollo de modelo predictivo de transiciones de turno”) exploramos distintas alternativas de modelado para construir una herramienta de etiquetado automático de transiciones de turno en conversaciones completas.

En el Capítulo 5 (“Resultados”) desarrollamos los modelos descritos en la sección anterior, considerando distintas combinaciones de hiperparámetros. Además, evaluamos los modelos en conversaciones nuevas y analizamos los resultados.

En el Capítulo 6 (“Conclusiones y trabajo futuro”) se resumen y comparan los resultados obtenidos en relación a los objetivos planteados. Además, se exponen las limitaciones de este trabajo y se plantean posibles mejoras a realizar en trabajos futuros.

2. PRELIMINARES

2.1. Estudio de transiciones de turno

Definimos como UNIDAD INTER-PAUSAL (*IPU*, según sus siglas en inglés) a un segmento continuo de habla contenido entre pausas de al menos 200 milisegundos. Estos segmentos de habla pueden variar en longitud, desde palabras sueltas hasta oraciones completas. En el estudio de transiciones de turno, las IPUs son fundamentales para estructurar las conversaciones y así poder analizar su dinámica posteriormente. En la Figura 2.1, cada segmento continuo representa una IPU.

Un TURNO es un intervalo de tiempo en el cual un hablante tiene participación activa en la conversación. Formalmente, un turno de un hablante se define como una secuencia maximal de IPUs en las que no hay presencia de IPUs de otro interlocutor durante los silencios del turno en curso. Un turno abarca desde el momento en que el hablante comienza a hablar hasta que cede la palabra, ya sea porque terminó de expresar su mensaje o porque fue interrumpido por otro interlocutor. En la Figura 2.1, las líneas grises representan los cuatro turnos esquematizados.

Una TRANSICIÓN DE CAMBIO DE TURNO es un evento en el cual el turno pasa de un interlocutor a otro en una conversación. Identificarlas es uno de los mayores desafíos que presentan los sistemas de diálogo hablado, con el fin de tener un manejo más eficiente de la conversación.

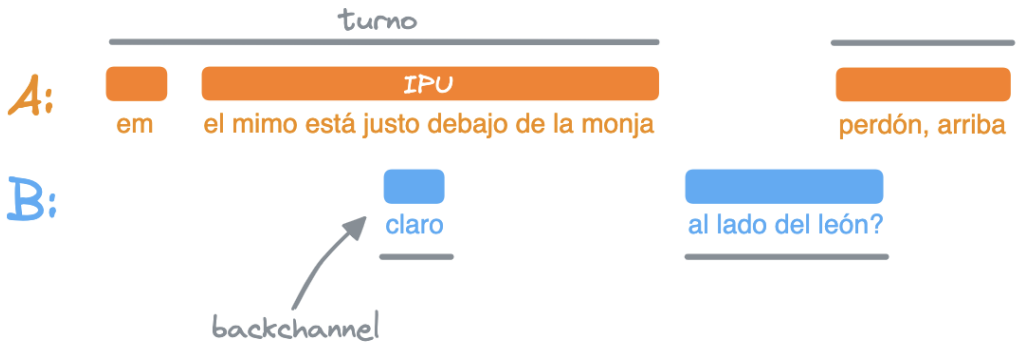


Fig. 2.1: *Ilustración de fragmento de una conversación estructurada con conceptos de manejo de turnos. Los turnos (denotados por las líneas grises) se componen por uno o más IPUs (denotadas por las barras coloreadas).*

En esta tesis, estamos interesados en una categorización particular de transiciones de turno definida en Beattie (1982), que especifica una taxonomía completa de los distintos tipos de transiciones. Esta categorización divide las transiciones en dos subcategorías disjuntas: con y sin solapamiento. Una transición *sin solapamiento* ocurre cuando el turno de un hablante comienza mientras el otro está en silencio. En cambio, una transición *con solapamiento* ocurre cuando el hablante comienza a hablar antes de que la IPU del hablante anterior finalice. Cabe aclarar que breves solapamientos pueden contribuir a conversaciones más fluidas y naturales, y no significan necesariamente interrupciones de turno.

- Categorías de transición de cambio de turno **sin solapamiento**:
  - SWITCH (S): cuando el interlocutor toma la palabra después de que el hablante complete su mensaje y deje un silencio.
  - INTERRUPCIÓN EN PAUSA (PI): cuando el interlocutor toma la palabra luego de un silencio por parte del hablante, pero sin que este haya terminado el mensaje a transmitir.
  - BACKCHANNEL (BC): una breve interlocución (tal como "ajá" o "claro") sin la intención de interrumpir el turno del hablante principal. Este recurso es utilizado por los hablantes para denotar atención.
- Categorías de transición de cambio de turno **con solapamiento**:
  - OVERLAP (O): equivalente a S pero con solapamiento; el interlocutor toma la palabra antes de que el hablante actual termine de transmitir su mensaje.
  - BACKCHANNEL CON SOLAPAMIENTO (BC_O): equivalente a BC pero con solapamiento; es decir, breves interlocuciones sin intención de tomar el turno, mientras otro hablante transmite su mensaje.
  - INTERRUPCIÓN CON SOLAPAMIENTO (I): equivalente a PI pero con solapamiento; el interlocutor interrumpe al hablante actual mientras este transmitía su mensaje.
  - BUTTING-IN (BI): similar a I pero con la diferencia de que el hablante no logra tomar la palabra y su interrupción queda inconclusa.

Además, se definen las siguientes categorías que completan los posibles casos a etiquetar dentro de una conversación:

- HOLD (H): cuando el hablante actual continúa hablando después de una breve pausa. En este caso, no es una transición entre turnos sino intra-turnos.
- X1: denota el comienzo del primer turno en la conversación.
- X2 y X2_O: suceden a BC y BC_O, respectivamente. Denotan el regreso al hablante original antes de que se produzca la interlocución.
- X3: cuando ocurren dos IPUs simultáneamente (a menos de 210 ms entre sí), la primera se categoriza siguiendo las reglas anteriormente mencionadas, mientras que la segunda se categoriza con esta etiqueta. Ocurre cuando hay un conflicto en quién comienza a hablar luego de un silencio.

Notar que las categorías H, X1, X2, X2_O y X3 son fáciles de determinar programáticamente en base a las IPUs y el resto de las etiquetas. Sin embargo, identificar el resto presenta un desafío ya que es necesario conocer las intenciones de los hablantes.

2.2. Redes neuronales *feed-forward*

Las REDES NEURONALES (Rosenblatt, 1958) son estructuras computacionales inspiradas en el funcionamiento del cerebro humano, que son capaces de aprender a **reconocer patrones** y realizar tareas complejas de clasificación, regresión y procesamiento de datos.

Dados los conjuntos $X \subseteq \mathbb{R}^d$ y $Y \subseteq \mathbb{R}^m$, una red neuronal es una función $\hat{f} : X \to Y$ que intenta aproximar lo mejor posible a una función subyacente $f : X \to Y$ a partir de un conjunto de instancias $(\mathbf{x}, \mathbf{y}) \in X \times Y$, conocido como CONJUNTO DE ENTRENAMIENTO.

La red neuronal de tipo *feed-forward* más simple, conocida como SINGLE LAYER PERCEPTRON, se define de la siguiente manera para una entrada $\mathbf{x} \in X$:

$$\mathbf{y_{pred}} = \alpha(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{2.1}$$

donde $\mathbf{W} \in \mathbb{R}^{m \times d}$ es la matriz de pesos, $\mathbf{b} \in \mathbb{R}^m$ es el vector de sesgos y α es una función —usualmente no lineal— conocida como FUNCIÓN DE ACTIVACIÓN[1], siendo *sigmoid*, *tanh* y *ReLU* algunas de las más frecuentemente utilizadas. $\mathbf{W}$ y $\mathbf{b}$ son los PARÁMETROS del modelo, cuyos valores se ajustan durante la fase de entrenamiento de la red.

Además, suelen agregarse capas ocultas intermedias; en este caso, la salida anterior $\mathbf{y_{pred}}$ se utiliza como entrada de la siguiente capa y así sucesivamente. Este tipo de red se conoce como MULTILAYER PERCEPTRON y, para una red de c capas, las operaciones pueden expresarse de la siguiente manera:

$$H_1(\mathbf{x}) = \alpha_1(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \tag{2.2}$$
$$H_2(\mathbf{x}) = \alpha_2(\mathbf{W}_2 H_1(\mathbf{x}) + \mathbf{b}_2) \tag{2.3}$$
$$\vdots$$
$$H_c(\mathbf{x}) = \alpha_c(\mathbf{W}_c H_{c-1}(\mathbf{x}) + \mathbf{b}_c) \tag{2.4}$$

donde $\mathbf{W}_i \in \mathbb{R}^{m_i \times m_{i-1}}$ es la matriz de pesos de la capa i, $\mathbf{b}_i \in \mathbb{R}^{m_i}$ es el vector de sesgos y α_i es su función de activación, que puede ser distinta para cada capa. Finalmente, la salida de la red se corresponde con la salida de la última capa oculta H_c:

$$\mathbf{y_{pred}} = H_c(\mathbf{x}) \tag{2.5}$$

Usualmente, la última capa de una red que realiza tareas de clasificación utiliza *softmax* como función de activación para convertir la salida obtenida en un vector de probabilidades asociadas a las clases objetivo. La Figura 2.2 ilustra gráficamente un ejemplo de este tipo de redes.

[1] Utilizamos el abuso de notación $\alpha(v)$ para denotar la aplicación de α elemento por elemento, para algún $v \in \mathbb{R}^n$

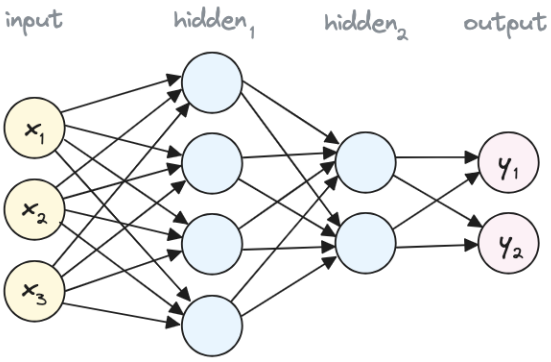


Fig. 2.2: *Ejemplo de red neuronal feed-forward con dos capas ocultas. Cada perceptrón está conectado con todos los de la capa siguiente y la información fluye únicamente hacia adelante.*

Aprendizaje de la red

En cada iteración de la fase de entrenamiento, utilizamos el conjunto de instancias de entrenamiento para realizar ajustes en los parámetros. En particular, para cada instancia $(\mathbf{x}, \mathbf{y})$, utilizamos una FUNCIÓN DE COSTO —siendo el error cuadrático medio la más frecuente para problemas de regresión y la entropía cruzada para problemas de clasificación— para medir el error entre el valor esperado $(\mathbf{y})$ y el valor predicho por el modelo $(\mathbf{y}_{\text{pred}} = \hat{f}(\mathbf{x}))$. Finalmente, ajustamos los parámetros del modelo de forma tal que se minimice el valor de la función de costo. El algoritmo más común para realizar el ajuste de los parámetros es el de *descenso por gradiente*, que los actualiza según la dirección que reduce el error más rápidamente.

El proceso de aprendizaje se repite por varias iteraciones hasta que la red aprende a realizar la tarea con un error aceptablemente bajo, permitiéndole generalizar lo suficientemente bien a nuevas instancias que no hayan sido anteriormente vistas durante el entrenamiento.

2.3. Redes neuronales recurrentes

Las REDES NEURONALES RECURRENTES (Rumelhart et al., 1986) son una clase de redes neuronales diseñadas para procesar datos **secuenciales**, tales como series temporales o texto. En particular, este tipo de redes mantiene un ESTADO OCULTO que almacena información del **contexto previo**, lo cual resulta útil para modelar problemas relacionados al análisis de mercado financiero, traductores y predicción de clima, entre otros.

A diferencia de las redes neuronales clásicas, las instancias de entrada y de salida son series temporales $\mathbf{x}(t)$ y $\mathbf{y}(t)$, respectivamente. El estado oculto en el instante t se define de la siguiente manera, para una secuencia $\mathbf{x}$ de longitud T:

$$h(t) = \alpha(\mathbf{W}\mathbf{x}(t) + \mathbf{U}h(t-1) + \mathbf{b}) \tag{2.6}$$

donde $\mathbf{W} \in \mathbb{R}^{n \times d}$ es la matriz de pesos aplicada sobre la entrada, $\mathbf{U} \in \mathbb{R}^{n \times n}$ es la matriz de pesos aplicada sobre el estado oculto anterior, $\mathbf{b} \in \mathbb{R}^n$ es el vector de sesgos y α es una función de activación. La dimensión del estado oculto en cada instante queda

determinada por n, que es un hiperparámetro de la red. El estado oculto $h(t)$ captura la información de la secuencia hasta el tiempo t.

Además, la salida de la red en cada paso temporal t se obtiene de la siguiente manera:

$$\mathbf{y_{pred}}(t) = g(\mathbf{V}h(t) + \mathbf{c}) \tag{2.7}$$

donde $\mathbf{V} \in \mathbb{R}^{m \times n}$ es la matriz de pesos que conecta el estado oculto con la salida, $\mathbf{c} \in \mathbb{R}^m$ es el vector de sesgos de la salida y g es otra función de activación. Los PARÁMETROS de este tipo de redes son $\mathbf{V}$, $\mathbf{W}$, $\mathbf{U}$, $\mathbf{b}$ y $\mathbf{c}$. En la Figura 2.3 se puede observar un ejemplo gráfico de este procedimiento.

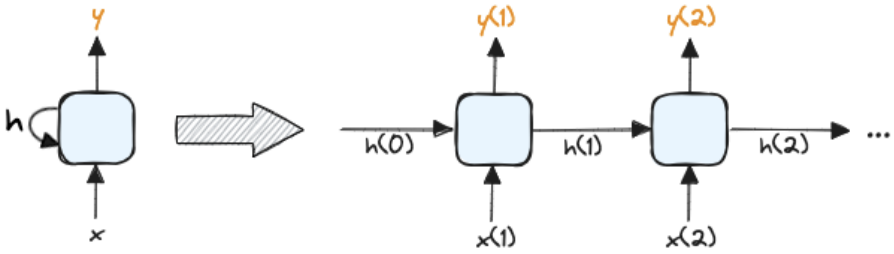


Fig. 2.3: *Representación de una red neuronal recurrente. Del lado izquierdo, la vista compacta del modelo; del derecho, la vista expandida por instante de tiempo. La salida en cada instante se obtiene a partir de la entrada en ese instante de tiempo y el estado oculto del instante anterior.*

El proceso de entrenamiento sigue un enfoque similar al de las redes neuronales *feed-forward*. La diferencia principal es que, debido a la naturaleza secuencial de las instancias, se considera la suma de los errores cometidos en la secuencia completa para realizar el ajuste de los parámetros del modelo, es decir $\sum_t e(y(t), y_{pred}(t))$. Además, el cálculo del gradiente debe considerar las dependencias temporales, lo cual se resuelve mediante el algoritmo conocido como *backpropagation through time*, que extiende el concepto de *backpropagation* para el manejo de series temporales.

2.4. Redes neuronales recurrentes bidireccionales

Las REDES NEURONALES RECURRENTES BIDIRECCIONALES (Schuster and Paliwal, 1997) son una extensión de las redes neuronales recurrentes que permiten la propagación de información tanto hacia adelante como hacia atrás en el tiempo, obteniendo así un **contexto completo** de la secuencia de entrada. Esto resulta especialmente útil cuando tener la información completa de la serie temporal es crucial, como en tareas de procesamiento de lenguaje natural.

Este tipo de redes utiliza un estado oculto h que, a su vez, se compone por dos sub-estados ocultos: uno con información del pasado y otro con información del futuro. Es decir, para una secuencia de entrada $\mathbf{x} \in X$, los estados ocultos hacia adelante y hacia atrás se definen de la siguiente manera:

$$h(t) = \alpha(\mathbf{W}\mathbf{x}(t) + \mathbf{U}h(t-1) + \mathbf{b}) \tag{2.8}$$

$$h'(t) = \alpha'(\mathbf{W}'\mathbf{x}(t) + \mathbf{U}'h'(t+1) + \mathbf{b}') \tag{2.9}$$

donde $\mathbf{W} \in \mathbb{R}^{n \times d}$ y $\mathbf{W}' \in \mathbb{R}^{n \times d}$ son las matrices de pesos aplicadas sobre la entrada, $\mathbf{U} \in \mathbb{R}^{n \times n}$ y $\mathbf{U}' \in \mathbb{R}^{n \times n}$ son las matrices de pesos aplicadas sobre los estados ocultos previos en las direcciones correspondientes, $\mathbf{b} \in \mathbb{R}^n$ y $\mathbf{b}' \in \mathbb{R}^n$ son los vectores de sesgos correspondientes y α, α' son funciones de activación. De esta manera, el estado oculto de la red h se define como la concatenación (denotada con $\oplus$) de los estados ocultos en ambas direcciones:

$$h(t) = h(t) \oplus h'(t) \tag{2.10}$$

La salida de la red en cada instante t se calcula de la siguiente manera:

$$\mathbf{y_{pred}}(t) = \beta(\mathbf{V}h(t) + \mathbf{c}) \tag{2.11}$$

donde $\mathbf{V} \in \mathbb{R}^{m \times 2n}$ es la matriz de pesos que conecta el estado oculto concatenado con la salida, $\mathbf{c} \in \mathbb{R}^m$ es el vector de sesgos de la salida y β es otra función de activación. La Figura 2.4 muestra una representación de la arquitectura de este tipo de redes.

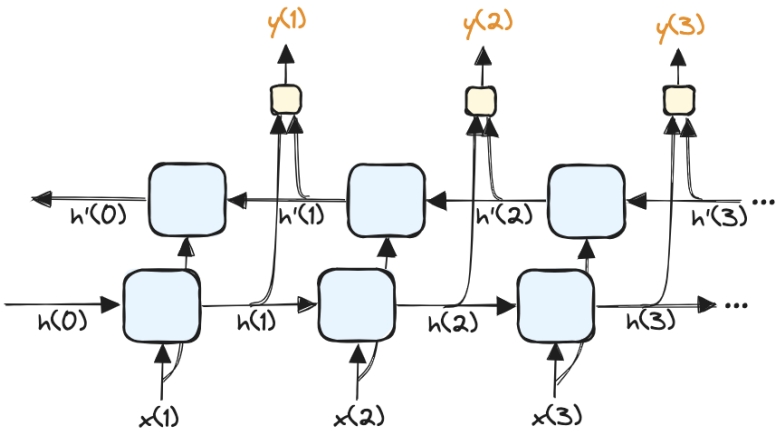


Fig. 2.4: *Representación de una red neuronal recurrente bidireccional. Cada una de las subredes almacena información sobre el contexto pasado o futuro, respectivamente. En cada instante, los estados ocultos de ambas se concatenan y a partir de ellos se obtiene la salida, obteniendo un contexto completo de la instancia.*

El entrenamiento de este tipo de redes se realiza de una forma similar a las redes neuronales recurrentes clásicas, ya que los cálculos de los estados ocultos en ambas direcciones son independientes entre sí.

2.5. Redes neuronales recurrentes GRU

Las GATED RECURRENT UNITS (Cho et al., 2014) son una variante de las redes neuronales recurrentes que, a través de COMPUERTAS, modelan una mejor dinámica de **memoria a largo plazo**, solucionando problemas como *vanishing* o *exploding gradients* de las redes recurrentes clásicas ocurridos durante la etapa de entrenamiento.

Cada unidad recurrente dentro de una celda GRU se compone por dos compuertas, que controlan el flujo de información dentro de la unidad: una de actualización (z) y otra de

reinicio (r). Para una secuencia de entrada $\mathbf{x} \in X$ y un instante de tiempo t, el cálculo de las compuertas y el estado oculto se realiza de la siguiente manera:

$$z(t) = \sigma(\mathbf{W}_z \mathbf{x}(t) + \mathbf{U}_z h(t-1) + \mathbf{b}_z) \tag{2.12}$$
$$r(t) = \sigma(\mathbf{W}_r \mathbf{x}(t) + \mathbf{U}_r h(t-1) + \mathbf{b}_r) \tag{2.13}$$
$$\tilde{h}(t) = \tanh(\mathbf{W}_h \mathbf{x}(t) + \mathbf{U}_h (r(t) \odot h(t-1)) + \mathbf{b}_h) \tag{2.14}$$
$$h(t) = (1 - z(t)) \odot h(t-1) + z(t) \odot \tilde{h}(t) \tag{2.15}$$

donde $\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}_h \in \mathbb{R}^{n \times d}$ son las matrices de pesos aplicadas sobre la entrada, $\mathbf{U}_z, \mathbf{U}_r, \mathbf{U}_h \in \mathbb{R}^{n \times n}$ son las matrices de pesos aplicadas sobre el estado oculto anterior, $\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_h \in \mathbb{R}^n$ son los vectores de sesgos, σ es la función sigmoidea, tanh es la función tangente hiperbólica, y $\odot$ denota la multiplicación elemento a elemento.

El estado oculto $h(t)$ en cada instante de tiempo se calcula combinando la nueva memoria candidata $\tilde{h}(t)$ y el estado oculto anterior $h(t-1)$, controlado por la compuerta de actualización $z(t)$, mientras que la compuerta de reinicio $r(t)$ controla la forma en la que el estado anterior influye en el cálculo de la nueva memoria candidata $\tilde{h}(t)$.

La salida de la red en cada paso temporal t se obtiene de la misma manera que en las RNNs clásicas:

$$\mathbf{y_{pred}}(t) = \alpha(\mathbf{V} h(t) + \mathbf{c}) \tag{2.16}$$

donde $\mathbf{V} \in \mathbb{R}^{m \times n}$ es la matriz de pesos que conecta el estado oculto con la salida, $\mathbf{c} \in \mathbb{R}^m$ es el vector de sesgos de la salida y α es otra función de activación. La Figura 2.5 muestra la arquitectura de una celda GRU.

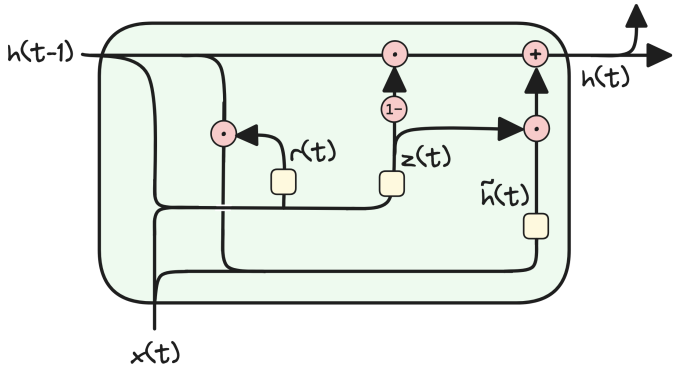


Fig. 2.5: *Representación de una celda GRU (adaptación de la Figura 1 de Jin et al. (2020)). El estado oculto en cada instante de tiempo se obtiene a partir de la entrada y el estado oculto anterior, en base a las compuertas que regulan la forma en la cual se adquiere nueva información.*

El entrenamiento de este tipo de redes también sigue un procedimiento similar al de las RNNs tradicionales, utilizando el mecanismo de *backpropagation through time* para ajustar los parámetros del modelo.

2.6. BERT

Los TRANSFORMERS (Vaswani et al., 2017) han marcado un antes y un después en las tareas de procesamiento de lenguaje natural, brindando resultados sin precedentes. A diferencia de las arquitecturas basadas en RNN, los *transformers* son capaces de procesar secuencias completas utilizando un mecanismo de **atención** que les permite capturar dependencias de largo plazo y de forma bidireccional en el tiempo, inherente a su naturaleza no secuencial ni recurrente. Además, la arquitectura de esta red permite la posibilidad de realizar **cómputos en paralelo**, lo cual resulta conveniente a la hora de entrenar estos modelos a mayor escala. Otra característica fundamental de este tipo de redes es que se pueden adaptar a una diversidad de tareas mediante *fine-tuning*; de hecho, en la actualidad se pueden encontrar ejemplos que van desde traductores hasta modelos generativos —usualmente basados en la familia de modelos GPT, tales como ChatGPT o DALL-E—.

En particular, el trabajo de Devlin et al. (2018b) introduce BERT (*Bidirectional Encoder Representations from Transformers*), un modelo basado en *transformers* que busca la comprensión de lenguajes naturales mediante representaciones contextuales. Al procesar una secuencia de palabras completas —y, por lo tanto, en ambas direcciones simultáneamente—, puede capturar el contexto completo de una palabra y modelar las relaciones entre ellas en el contexto de una oración, lo que ha derivado en mejoras significativas en una amplia gama de tareas de PLN, generando una comprensión y coherencia de texto excepcionales.

Durante la etapa de preentrenamiento, se entrena un modelo que resuelve dos tareas simultáneamente:

- **Modelo de Lenguaje Enmascarado (MLM)**: En esta tarea —ilustrada en la Figura 2.6—, se seleccionan algunas palabras al azar de la secuencia de entrada y se enmascaran. El modelo debe predecir estas palabras enmascaradas basándose en el contexto proporcionado por las palabras circundantes. En este proceso, el modelo aprende a capturar representaciones contextuales profundas de las palabras.

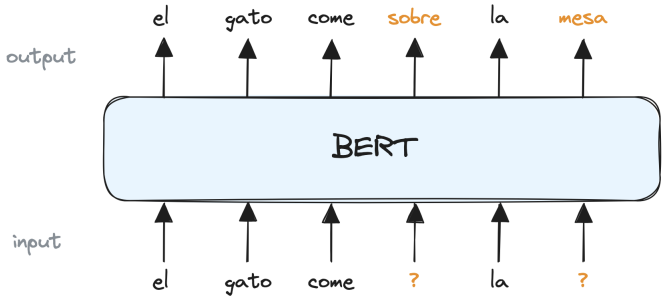


Fig. 2.6: *Entrenamiento de modelo MLM. Para cada secuencia de entrada, el modelo debe predecir cuáles son las palabras enmascaradas, denotadas con un signo de pregunta ('?').*

- **Predicción de la Siguiente Oración (NSP)**: En esta tarea —ilustrada en la Figura 2.7—, se utilizan dos oraciones como secuencias de entrada, algunas de las cuales son secuencias consecutivas en el texto original y otras que son secuencias aleatorias. El modelo debe aprender a predecir si ambas oraciones son consecutivas o

no, lo que ayuda a capturar relaciones más amplias entre las oraciones y a comprender la coherencia del texto.

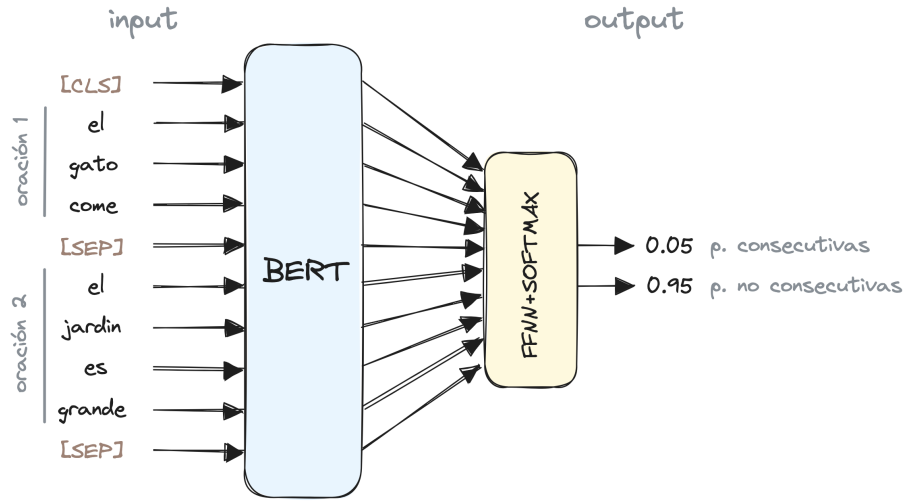


Fig. 2.7: *Entrenamiento de NSP. Se introducen dos oraciones como secuencias de entrada separadas por tokens especiales para delimitar el comienzo y fin de cada una. El modelo debe predecir la probabilidad de que estas sean consecutivas.*

El preentrenamiento de BERT resulta en la creación de modelos que comprenden el lenguaje en un nivel profundo, capturando tanto las dependencias locales entre palabras como las relaciones más amplias entre oraciones. Mediante *fine-tuning*, estos modelos pueden adaptarse para realizar tareas específicas, tales como clasificación de texto o análisis de sentimientos, entre otras. Esto implica agregar capas adicionales de neuronas en la parte superior del modelo preentrenado y entrenarlas utilizando un conjunto de datos etiquetados para la tarea en cuestión.

3. FUENTES DE DATOS

Para este trabajo, utilizaremos distintas versiones del Objects Games Corpus (Juego de Objetos), un experimento realizado en múltiples universidades donde se recolectaron datos de conversaciones diádicas espontáneas de hablantes nativos de inglés norteamericano y español argentino. Las conversaciones se basan en actividades colaborativas entre ambos participantes, en las cuales uno debe instruir al otro para ubicar correctamente un objeto faltante en su pantalla. En el contexto de este trabajo, utilizamos únicamente los siguientes corpus:

- Columbia Games Corpus (inglés) (2004)
- UBA Games Corpus (español) (2014)

Cada uno de ellos cuenta con más de 300 minutos de grabación junto a las etiquetas de transiciones de turno, así como las transcripciones completas de cada conversación.

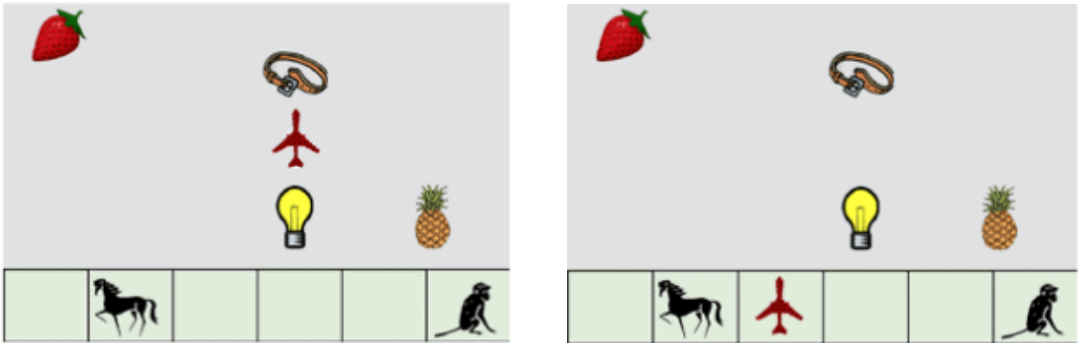
Previo al desarrollo de los modelos, realizamos exploraciones visuales de las palabras utilizadas. Observamos que, al generar *embeddings* con los mismos modelos preentrenados utilizados durante el desarrollo de los modelos, palabras similares quedaban ubicadas cerca en el espacio. De esta forma, la expectativa es que los modelos desarrollados en el Capítulo 5 puedan capturar información útil de estas para mejorar las predicciones.

En la siguiente sección se detalla la metodología, así como las características propias de cada uno de los corpus utilizados.

3.1. Juegos de Objetos

Un juego de objetos es una actividad en la que dos personas participan en juegos de computadora colaborativos, que requieren comunicación verbal y ofrecen una modesta compensación monetaria. Cada participante dispone de una computadora portátil que muestra un tablero con entre 5 y 7 objetos. Ambos ven el mismo conjunto de objetos en las mismas posiciones, excepto por un objeto que falta en uno de los tableros. En este contexto, uno de los participantes, denominado *Descriptor*, debe describir la posición del objeto faltante a su compañero, el *Seguidor*, quien debe colocar el objeto en la misma posición en su pantalla.

Cada sesión de este juego consta de 14 tareas, durante las cuales los participantes alternan los roles de *Descriptor* y *Seguidor*. Los puntos, que van de 1 a 100, se asignan en función de la precisión con la que el *Seguidor* coloca el objeto en comparación con la descripción del Descriptor. La Figura 3.1 muestra, a modo ilustrativo, las pantallas de cada uno de los jugadores para una de las tareas de la sesión.



(a) Descriptor *(b) Seguidor*

Fig. 3.1: *Pantallas de cada uno de los jugadores. En cada tarea, el jugador Descriptor debe instruir al otro participante para ubicar correctamente el objeto que titila en su pantalla —en este caso, el avión—.*

Los experimentos se llevaron a cabo en cabinas insonorizadas, cada participante utilizando una computadora diferente y separados por una cortina opaca para garantizar que toda la comunicación fuera exclusivamente verbal. A los participantes se les informó que debían acumular la mayor cantidad de puntos posible durante la sesión, ya que recibirían una compensación adicional por cada punto ganado. En todos los casos, la comunicación entre los participantes no estuvo restringida de ninguna manera.

Todos los participantes acordaron unirse al estudio firmando un formulario de consentimiento, por lo que fueron conscientes de la naturaleza del mismo.

3.1.1. Columbia Games Corpus

El Corpus Columbia fue recopilado y anotado conjuntamente por el *Grupo de Procesamiento del Habla* de la Universidad de Columbia y el Departamento de Lingüística de la Universidad Northwestern. Un total de 13 sujetos (6 mujeres, 7 hombres), con edades entre 20 y 50 (M = 30.0, SD = 10.9), participaron en el estudio en la ciudad de Nueva York en octubre de 2004. Once de los sujetos participaron en dos sesiones en diferentes días, cada vez con un compañero diferente. Todos los sujetos eran hablantes nativos y vivían en el área de la ciudad de Nueva York en el momento del estudio. Se registraron un total de 4.5 horas de diálogo en este corpus.

La Figura 3.2 muestra las palabras más frecuentes utilizadas en los diálogos de este corpus. Podemos notar que, en su gran mayoría, son palabras muy específicas de las tareas de este conjunto y no necesariamente se trasladan a otros diálogos de distinta naturaleza.

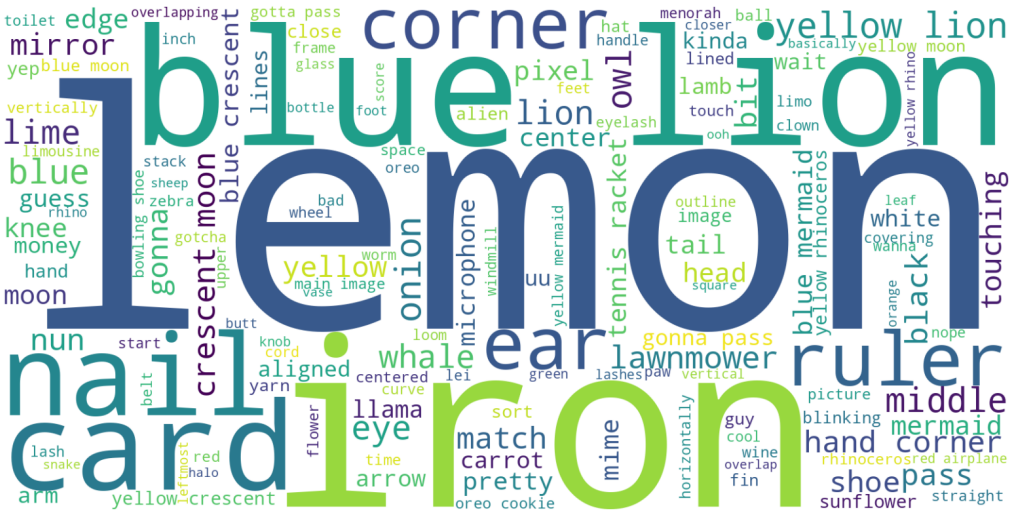


Fig. 3.2: *Palabras más frecuentes observadas durante las conversaciones de Columbia Games Corpus. Las palabras con baja entropía*[1] *no fueron consideradas para este diagrama.*

3.1.2. UBA Games Corpus

Las grabaciones de UBA Games Corpus se realizaron en dos lotes —en 2012 y 2014, respectivamente—. Todos los sujetos eran hablantes nativos de español argentino y vivían en el área de Buenos Aires en el momento del estudio. En este trabajo utilizamos únicamente los datos del primer lote, que tuvo lugar entre noviembre y diciembre de 2012, en el *Laboratorio de Investigaciones Sensoriales* (Hospital de Clínicas, UBA)[2]. Participaron un total de 14 sujetos (7 mujeres, 7 hombres), con edades comprendidas entre 19 y 59 años ($M = 28.6$, $SD = 12.7$). Se registraron un total de 6.4 horas en este lote.

Al igual que en el corpus anterior, podemos notar en la Figura 3.3 que las palabras más frecuentes utilizadas en estos diálogos son específicas de este tipo de tareas y no necesariamente son frecuentes en otro tipo de diálogos.

[1] Basándonos en una lista de las palabras más frecuentes en inglés.
[2] http://www.lis.secyt.gov.ar

Fig. 3.3: *Palabras más frecuentes observadas durante las conversaciones de UBA Games Corpus. Las palabras con baja entropía no fueron consideradas para este diagrama.*

4. DESARROLLO DE MODELO PREDICTIVO DE TRANSICIONES DE TURNO

En este capítulo analizamos distintas arquitecturas de modelado para el etiquetado *offline* de transiciones de turno. Como buscamos construir modelos que procesen conversaciones completas, a diferencia de los modelos en tiempo real, consideramos alternativas que analicen información tanto del pasado como del futuro.

El resto del capítulo se organiza de la siguiente manera: en la Sección 4.1 definimos nuestra unidad de análisis. La Sección 4.2 describe la separación de datos utilizada. En la Sección 4.3 presentamos el *modelo secuencial* propuesto en el trabajo de Brusco and Gravano (2023), al cual llamamos ***bigru[acoustic]*** en este trabajo. En segundo lugar, en la Sección 4.4, planteamos una extensión del modelo anterior, denominado ***bigru[acoustic+text]***, donde incorporamos atributos de texto en la entrada del modelo con la expectativa de que aprenda patrones en las transcripciones de las conversaciones y mejore el rendimiento obtenido. Finalmente, en la Sección 4.5 presentamos ***ms-bigru[acoustic+text]***, una adaptación del modelo *bigru[acoustic+text]* basada en la arquitectura multiescala presentada en el trabajo de Roddy et al. (2018), que permite a la red aprender los patrones de cada modalidad por separado y, por ende, podría contribuir a mejorar el rendimiento.

4.1. Unidad de análisis

Nuestra unidad de análisis, es decir, las instancias de nuestro problema de aprendizaje, son **tareas completas** del juego de objetos. Tal como fue descrito en el Capítulo 3, se trata de una conversación completa en la que un participante, con rol *descriptor*, debe instruir al otro, con rol de *seguidor*, para localizar ciertos objetos en su pantalla.

El objetivo de los modelos será predecir cada una de las transiciones de turno que ocurren dentro de cada tarea, es decir, etiquetar las clases: *switch* (S), *interrupción en pausa* (PI) y *backchannel* (BC); *overlap* (O), *backchannel con solapamiento* (BC_O), *interrupción con solapamiento* (I) y *butting-in* (BI); además de las etiquetas especiales *hold* (H), X1, X2, X2_O y X3 (presentadas en detalle en la Sección 2.1).

4.2. Separación de datos

Con fines metodológicos, además de utilizar la técnica de validación cruzada, particionamos cada uno de los corpus utilizados en dos subconjuntos: DESARROLLO y CONTROL (también llamado *held-out*, en inglés). El primero lo utilizamos para el desarrollo de los modelos mientras que, el segundo, únicamente para obtener resultados en conversaciones que no hayan sido utilizadas durante este proceso, y que serán utilizadas una vez que sean seleccionados los modelos finales.

Para construir los subconjuntos de control, seleccionamos dos tareas de cada sesión, así como también tres sesiones completas (cada una de 14 tareas). Esta selección se realizó de manera tal de lograr cierta diversidad en los tipos de transiciones, de forma que se pueda evaluar al modelo en conversaciones nuevas por pares de interlocutores conocidos (tareas aisladas) y desconocidos (sesiones aisladas). La Figura 4.1, describe la distribución de las clases de transición de turno en cada uno de los subconjuntos resultantes.

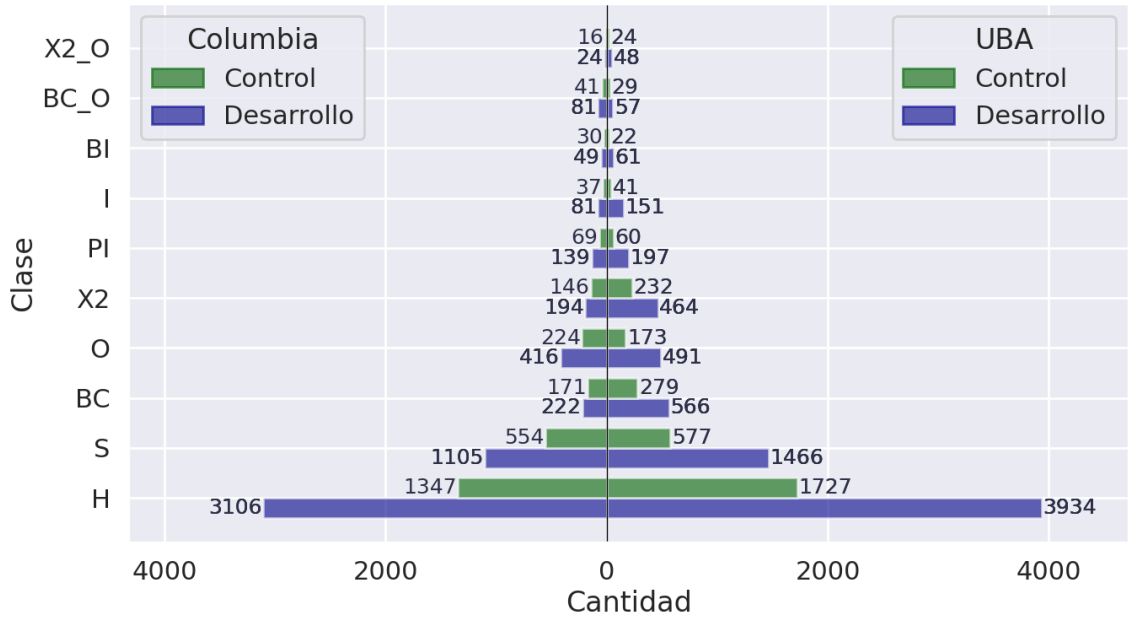


Fig. 4.1: *Apariciones por clase para cada subconjunto obtenido. Del lado derecho, los resultados para la partición realizada en el corpus en español (UBA); del lado izquierdo, los del corpus en inglés (Columbia).*

4.3. Modelo acústico - *bigru[acoustic]*

En el trabajo de Brusco and Gravano (2023), los autores presentan una arquitectura basada en redes neuronales recurrentes bidireccionales con celdas de tipo GRU, a la cual se refieren como *modelo secuencial*. A continuación, se detallan las principales características de esta propuesta, siendo que muchas de estas ideas se reutilizan en nuestros modelos.

Dentro de cada tarea y para cada interlocutor, se obtienen series temporales de atributos acústicos estandarizados mediante *z-scores*. Los atributos se extraen en ventanas de tiempo de 10 milisegundos utilizando la herramienta OpenSMILE (Eyben et al., 2010), e incluyen: *intensidad*, *tono*, *shimmer*, *jitter* y *logHNR*. En los instantes de tiempo donde alguna de estas señales no está definida (por ejemplo, en silencios o habla sin tono), se asigna un valor arbitrario por debajo y suficientemente alejado del mínimo de la señal correspondiente, de forma que los modelos puedan aprender a diferenciarlos del resto de los valores del dominio.

Un punto importante de estos modelos es que presuponen que se cuenta con:

(a) IPUs delimitadas para cada uno de los hablantes. Si no se cuenta con anotaciones realizadas por humanos, pueden obtenerse automáticamente utilizando Voice Activity Detectors (VADs), entrenados para estimar estos segmentos a partir de la señal acústica.

(b) Los momentos en que nos interesa predecir qué tipo de transición ocurrió, que llamamos *relevant places* (siguiendo la terminología propuesta en Sacks et al. (1978) para indicar lugares en la conversación en donde es probable que ocurra una transición de cambio de turno). Estas posiciones dentro de la señal se definen arbitrariamente (pero sin pérdida de generalidad) como aquellos instantes de tiempo en los que inicia

la IPU que sucede a la transición de turno. Los modelos deberán predecir la clase de una transición únicamente en estas posiciones relevantes.

La Figura 4.2 ilustra la partición de una tarea en las ventanas sobre las cuales se tiene toda la información descrita anteriormente.

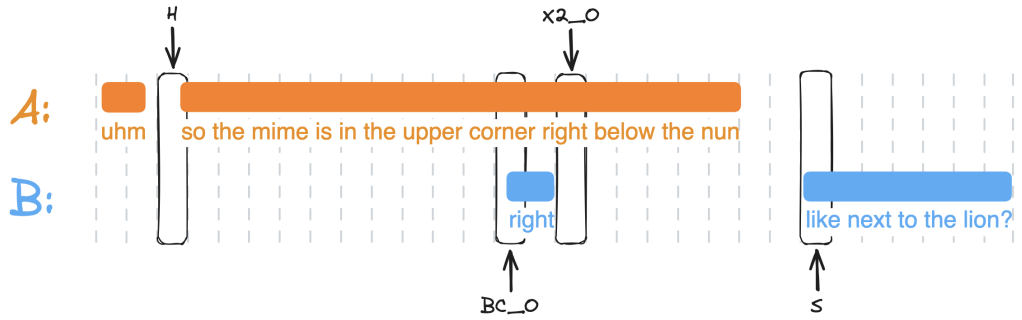


Fig. 4.2: *Partición de una tarea en ventanas, denotadas entre las líneas grises punteadas. Los segmentos en naranja y azul corresponden a la actividad vocal de cada uno de los interlocutores (IPUs). Para cada ventana, se tiene información tanto de la actividad vocal y sus atributos acústicos, como también si ocurrió o no una transición de turno en ese intervalo de tiempo. Los fragmentos resaltados corresponden a las posiciones relevantes donde se considera que ocurren las transiciones de turno.*

De esta manera, podemos definir formalmente una INSTANCIA de nuestra tarea de aprendizaje como una serie temporal de atributos correspondiente a una tarea completa del conjunto de datos. Más precisamente, para la k-ésima instancia, contamos con un vector de ATRIBUTOS INSTANTÁNEOS para un instante de tiempo t, definido como:

$$\mathbf{x^{(k)}}(t) = \mathbf{x^{(k)}_{TEMP}}(t) \oplus \mathbf{x^{(k)}_{VAD}}(t) \oplus \mathbf{x^{(k)}_{TURN}}(t)$$

Es decir, $\mathbf{x^{(k)}}(t)$ se forma a partir de la concatenación de $\mathbf{x^{(k)}_{TEMP}}(t)$ y $\mathbf{x^{(k)}_{VAD}}(t)$, que hacen referencia al vector instantáneo de atributos acústico-prosódicos y actividad vocal de ambos interlocutores (respectivamente), y $\mathbf{x^{(k)}_{TURN}}(t)$ es un vector con un único atributo que indica si esa posición contiene un *relevant place*, con información sobre la dirección en que ocurrió la transición de turno en dicho instante de tiempo (o cero si no hubo). Pese a no ser estrictamente necesario, este atributo adicional simplificará el aprendizaje del modelo y puede generarse automáticamente a partir del preprocesamiento de las tareas.

Las CLASES OBJETIVO del modelo son series temporales que indican, momento a momento, qué transición de turno ocurrió para cada instante de tiempo t. Además, utilizamos la clase *vacío* (denotada con un guión "$-$") como etiqueta especial para representar la ausencia de transiciones de turno.

La FUNCIÓN DE PÉRDIDA de este modelo cuenta con una particularidad: para evitar que el modelo realice ajustes en los parámetros producto del error cometido en momentos donde no ocurrieron transiciones de turno, se utiliza una alternativa ponderada que asigna peso cero a la clase *vacío*. Si bien esta modificación limita al modelo a no poder aprender a reconocer esta clase, no representa ningún inconveniente, puesto que el caso de uso considera valiosas únicamente las predicciones en los momentos donde ya se sabe que ocurrieron transiciones de turno.

La Figura 4.3 muestra un diagrama ilustrativo de la arquitectura de este modelo, que se compone por una capa GRU bidireccional conectada a una capa *feed-forward*, cuya salida contiene las predicciones por clase.

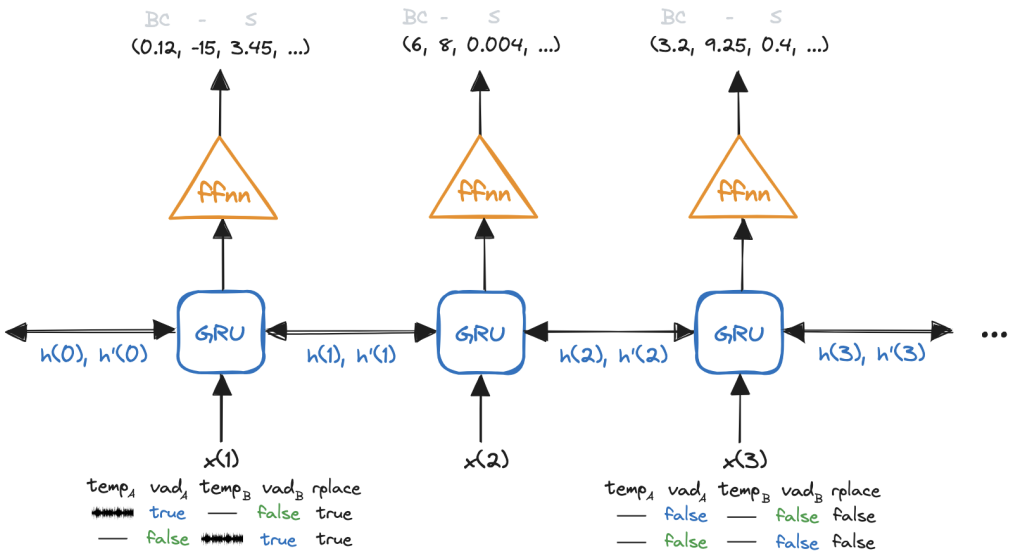


Fig. 4.3: *Representación de la arquitectura desplegada para una instancia x. Cada celda GRU acumula información del pasado y futuro de la conversación. Las etiquetas demarcadas con guión hacen referencia a la clase vacío. Los atributos instantáneos son procesados por la capa GRU y luego reducidos por la capa lineal, obteniendo las predicciones según las clases objetivo.*

4.4. Modelo con texto - *bigru[acoustic+text]*

En la sección anterior, analizamos la arquitectura del modelo *bigru[acoustic]*, que utiliza atributos acústicos de los interlocutores para el etiquetado automático de transiciones de turno. Tal como anticipamos en la introducción, el objetivo de esta sección es extender dicho modelo con atributos léxicos y sintácticos con la expectativa de aprender patrones útiles en las transcripciones de las conversaciones. Nos referiremos de ahora en más a este modelo como *bigru[acoustic+text]*.

Para cada tarea del corpus, contamos con la transcripción de la conversación entera de ambos participantes, así como también con sus oraciones[1]. Luego, para todas las palabras, calculamos sus *embeddings* y etiquetas *part-of-speech* asociadas utilizando modelos preentrenados que llamaremos σ y δ, respectivamente. En ambos casos, las palabras se consideran en el contexto de la oración a la cual pertenecen.

En las series temporales definidas en el modelo *bigru[acoustic]*, los instantes de tiempo t se corresponden uno a uno con las ventanas de la señal. Estos fragmentos se definieron en intervalos regulares fijos de 10 milisegundos, mientras que las duraciones de las palabras son, en promedio, significativamente más largas. Por lo tanto, para incorporar los nuevos atributos de texto a los atributos acústicos momento a momento, decidimos sobremuestrear la palabra en todos los fragmentos entre los cuales esta palabra ocurre. Además, si más de

[1] Las oraciones se calculan como las secuencias de palabras contenidas entre silencios

una palabra ocurre en un mismo fragmento para un mismo interlocutor, se considera solo una de ellas. La Figura muestra un ejemplo de este procedimiento.

Fig. 4.4: *Procedimiento de sobremuestreo realizado para ajustar las palabras a las ventanas regulares. Los espacios denotados entre las líneas en gris punteadas representan a los fragmentos originales. Del lado izquierdo, la transcripción de la conversación. Del lado derecho, el resultado, en el cual las palabras fueron sobremuestreadas y alineadas por fragmento.*

De esta manera, podemos definir las siguientes series temporales:

- $\mathbf{x}^{(\mathbf{k})}_{\mathbf{EMB}}$: *embeddings* de las palabras pronunciadas por cada interlocutor en el instante t, o vector nulo en caso de silencios.
- $\mathbf{x}^{(\mathbf{k})}_{\mathbf{POS}}$: etiquetas *part-of-speech* de las palabras pronunciadas por cada interlocutor en el instante t, o vector nulo en caso de silencios. Las etiquetas se codificaron utilizando *one-hot encodings*.

Además, definimos una serie temporal adicional $\mathbf{x}^{(\mathbf{k})}_{\mathbf{META}}$ con información adicional sobre las palabras ocurridas en cada instante de tiempo, para cada interlocutor:

- *Parcial*: indica si la palabra fue pronunciada parcialmente; es decir, si completa una palabra válida.
- *Completada*: indica si la palabra fue parcialmente enunciada, pero en las cuales el anotador dedujo de qué palabra se trataba y la completó.

La serie temporal que contiene todos los atributos relacionados a texto se define entonces como:

$$\mathbf{x}^{(\mathbf{k})}_{\mathbf{TEXT}}(t) = \mathbf{x}^{(\mathbf{k})}_{\mathbf{EMB}}(t) \oplus \mathbf{x}^{(\mathbf{k})}_{\mathbf{POS}}(t) \oplus \mathbf{x}^{(\mathbf{k})}_{\mathbf{META}}(t)$$

donde $\oplus$ denota la concatenación de vectores.

Finalmente, podemos redefinir una INSTANCIA de este modelo como la concatenación de los atributos acústicos-prosódicos y léxico-sintácticos de la siguiente manera:

$$\mathbf{x}^{(\mathbf{k})}(t) = \mathbf{x}^{(\mathbf{k})}_{\mathbf{ACOUS}}(t) \oplus \mathbf{x}^{(\mathbf{k})}_{\mathbf{TEXT}}(t)$$

donde $\mathbf{x}^{(\mathbf{k})}_{\mathbf{ACOUS}}(t)$ corresponde a la instancia definida para el modelo *bigru[acoustic]*.

Más allá de lo anterior, el modelo *bigru[acoustic+text]* mantiene la misma arquitectura y estrategias de modelado del modelo *bigru[acoustic]*, tal como fue explicado en la sección anterior.

4.5. Modelo multiescala - *ms-bigru[acoustic+text]*

En la sección anterior presentamos la arquitectura del *bigru[acoustic+text]*. Tal como fue descrito, para incluir los atributos de texto se realizó un sobremuestreo de las palabras para poder alinearlas a los fragmentos de la conversación. Esta estrategia no resulta conveniente ya que podemos dificultar la tarea de aprendizaje a los modelos al estar repitiendo información cuando el evento (la enunciación de la palabra) ocurrió sólo una vez.

En esta sección, analizamos una alternativa de modelado cuya arquitectura está basada en la propuesta de modelo multiescala del trabajo de Roddy et al. (2018), la cual utilizamos para modelar los atributos acústicos y textuales en distintos intervalos de tiempo, y que esperamos que supere la limitación anteriormente mencionada. Este modelo lo llamaremos *ms-bigru[acoustic+text]*.

La arquitectura de *ms-bigru[acoustic+text]* emplea varias subredes GRU bidireccionales[2] para procesar cada una de las modalidades (acústicas y textuales, en este caso) a distintas velocidades. Para la k-ésima instancia, cada subred (h_m) aprende los patrones de una modalidad específica m, permitiendo que las características acústico-prosódicas ($\mathbf{x}^{(\mathbf{k})}_{\mathbf{ACOUS}}$), que requieren una alta resolución temporal, se procesen en intervalos de 10 milisegundos, mientras que las características léxico-sintácticas ($\mathbf{x}^{(\mathbf{k})}_{\mathbf{TEXT}}$), se procesen en intervalos más lentos e irregulares, cada vez que termina de pronunciarse una palabra.

Para la subred h_{acoustic}, utilizamos las instancias ($\mathbf{x}^{(\mathbf{k})}_{\mathbf{ACOUS}}$) según fueron definidas en la Sección 4.3 para el modelo *bigru[acoustic]*. Por otro lado, definimos a las INSTANCIAS de la subred h_{text} de la siguiente manera, para cada *timestep* t:

$$\mathbf{x}^{(\mathbf{k})}_{\mathbf{TEXT}}(t) = \mathbf{x}^{(\mathbf{k})}_{\mathbf{EMB}}(t) \oplus \mathbf{x}^{(\mathbf{k})}_{\mathbf{POS}}(t) \oplus \mathbf{x}^{(\mathbf{k})}_{\mathbf{META}}(t) \oplus \mathbf{x}^{(\mathbf{k})}_{\mathbf{TURN}}(t)$$

$\mathbf{x}^{(\mathbf{k})}_{\mathbf{TEXT}}$ contiene los mismos atributos de texto del modelo anterior (según fue detallado en la Sección 4.4), con la diferencia que también se agrega la información adicional de turnos $\mathbf{x}^{(\mathbf{k})}_{\mathbf{TURN}}$, definida en la Sección 4.3. De esta manera, cada una de las instancias contiene atributos relacionados a su modalidad específicamente.

Para cada modalidad m, el estado oculto de la subred se actualiza de la siguiente manera:

$$h_m(t+1) = \text{GRU}(\mathbf{x}^{(\mathbf{k})}_{\mathbf{m}}(t), h_m(t))$$

La red *master*(h_{master}) muestrea el estado oculto de cada subred en intervalos irregulares, en los instantes de tiempo donde ocurren las transiciones de turno. Cada uno de estos estados ocultos contiene la información del futuro y pasado hasta el momento para cada una de las modalidades. La concatenación de los estados ocultos se utiliza como entrada en este modelo:

$$h_{\text{master}}(t+1) = \text{GRU}(h_{\text{acoustic}}(a) \oplus h_{\text{text}}(l), h_{\text{master}}(t))$$

donde a y l son los *timesteps* de cada subred —respectivamente— más cercanos a $t+1$. El proceso de muestreo permite ajustar la resolución temporal de las subredes, manteniendo la información relevante en diferentes escalas temporales.

[2] El trabajo de Skantze (2021) utilizó, a diferencia de este trabajo, subredes de tipo LSTM unidireccionales.

La salida del modelo *master* se corresponde con las predicciones de transiciones de turno momento a momento. Al igual que en los modelos anteriores, utilizamos una capa *feed-forward* para reducir la dimensión del resultado según la cantidad de clases objetivo:

$$\mathbf{y}^{(\mathbf{k})}(t+1) = \text{FFNN}(h_{\text{master}}(t+1))$$

La Figura 4.5 muestra la arquitectura del modelo, teniendo en cuenta los detalles de implementación mencionados.

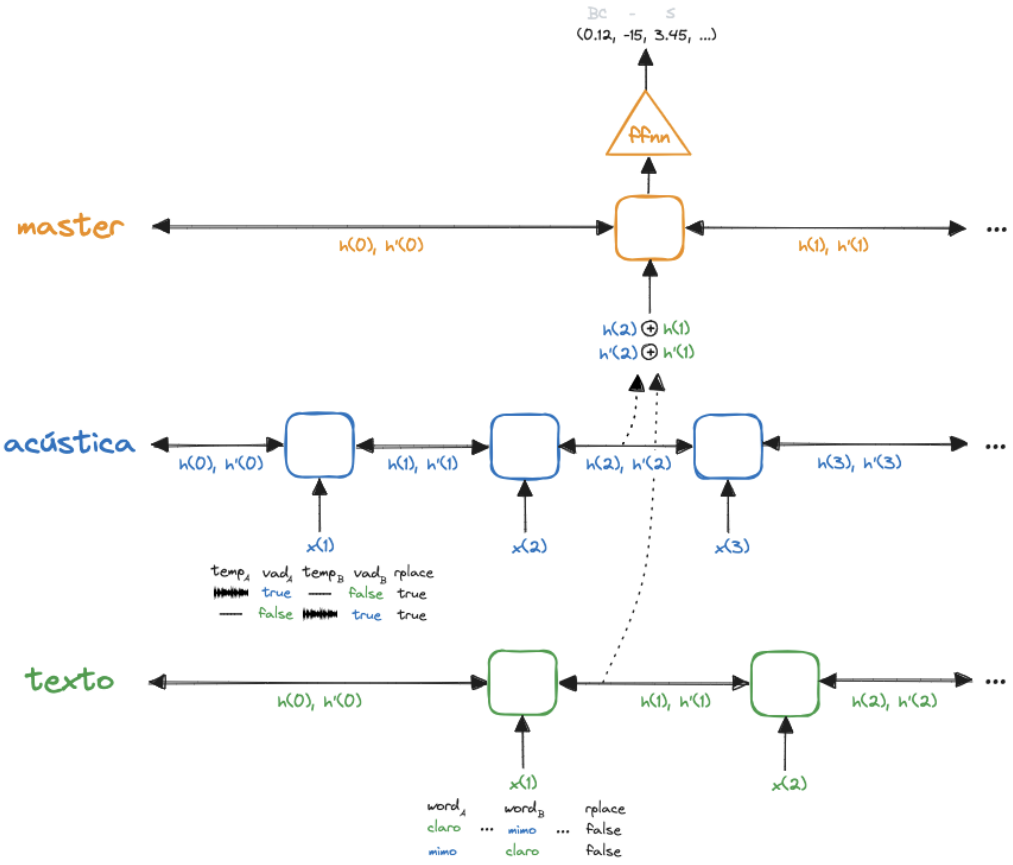


Fig. 4.5: *Arquitectura del modelo ms-bigru[acoustic+text]. La subred acústica analiza las características acústicas-prosódicas en intervalos regulares (de 10 milisegundos), mientras que la de texto analiza las características léxicas y sintácticas de los hablantes en intervalos irregulares. La red master muestrea periódicamente los estados ocultos de ambas subredes para generar las predicciones, considerando información del pasado y del futuro.*

5. RESULTADOS

En este capítulo, realizamos una serie de experimentos de aprendizaje automático con el fin de construir y analizar los distintos modelos descritos en la sección anterior para la categorización *offline* de transiciones de turno. La Figura 5.1 muestra, a modo ilustrativo, el etiquetado automático aplicado sobre una tarea en particular, realizado por uno de los modelos desarrollados en este capítulo.

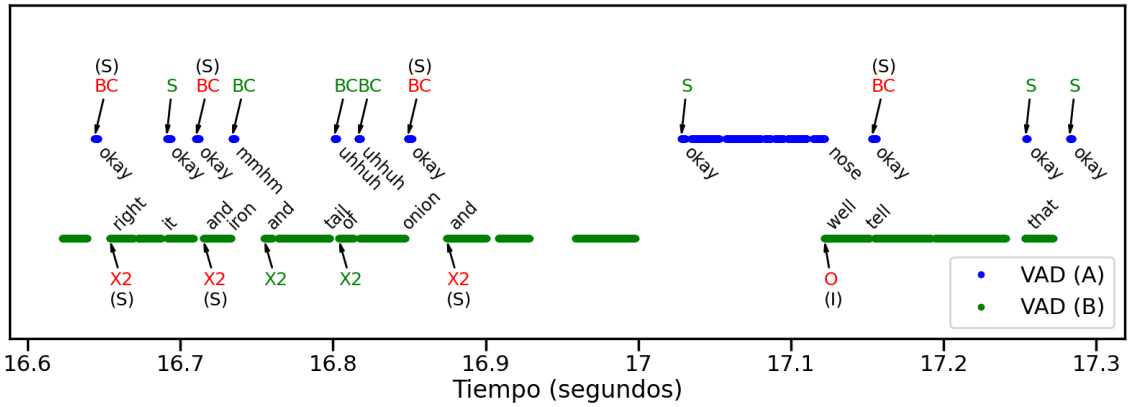


Fig. 5.1: *Etiquetado automático de transiciones de turno realizado sobre una tarea del conjunto de datos. Las etiquetas en verde corresponden a los aciertos del modelo, mientras que en rojo se muestran los errores, junto a la etiqueta correcta especificada en color negro. Por simplicidad, se omitieron los holds (H) y se incluyeron únicamente las palabras que ocurren cerca de las transiciones.*

En primer lugar, en la Sección 5.1, buscamos replicar los resultados obtenidos en el trabajo Brusco and Gravano (2023) para el que allí se refieren como modelo secuencial y que, en este trabajo, llamamos *bigru[acoustic]*. Para ello, entrenamos un modelo utilizando los mejores hiperparámetros allí especificados sobre el corpus en español. Este paso resulta fundamental siendo que el resto de los modelos presentados en este trabajo se construyen a partir de éste, por lo que debemos garantizar que partimos de un modelo confiable.

En segundo lugar, en la Sección 5.2.1, reentrenamos el modelo *bigru[acoustic]* pero, a diferencia del experimento anterior, utilizando el corpus en inglés y una variedad de combinaciones de hiperparámetros. El resto de los modelos presentados en este trabajo serán entrenados sobre este corpus y el trabajo citado no expone resultados para el modelo entrenado exclusivamente con éste, por lo que resulta necesario regenerarlo sobre este conjunto de datos para ser capaces de comparar resultados posteriormente.

En tercer lugar, en la Sección 5.2.2, intentamos responder la pregunta P_1: *¿Es posible extender el modelo original de forma tal que reconozca, además de los patrones acústico-prosódicos, patrones en las transcripciones de las conversaciones? ¿Cómo afecta a su rendimiento?* Para ello, entrenamos distintas alternativas del modelo *bigru[acoustic+text]*, considerando distintas combinaciones de hiperparámetros y estrategias de modelado y utilizando exclusivamente el subconjunto de desarrollo del corpus en inglés. Creemos que la inclusión de los atributos léxicos y sintácticos al modelo podría otorgar mejoras en el

rendimiento.

En cuarto lugar, en la Sección 5.2.3 exploramos, de forma similar al experimento anterior, distintas combinaciones de hiperparámetros y estrategias de modelado para el modelo *ms-bigru[acoustic+text]*, con el fin de responder P_2: *¿Cómo impacta construir un modelo con una arquitectura diseñada específicamente para concatenación de series temporales multimodales?*. Para el entrenamiento de los modelos, utilizamos exclusivamente el subconjunto de desarrollo del corpus en inglés. Creemos que la arquitectura de este modelo puede otorgar mejoras de rendimiento —con respecto a las alternativas anteriores— al capturar la información de cada modalidad de manera aislada, aunque también entendemos que este modelo resulta más complejo y puede requerir mayor experimentación para desarrollar una implementación que cumpla con las expectativas.

En quinto lugar, en la Sección 5.3.1, intentamos responder P_3: *¿Cómo se comportan estos modelos ante conversaciones nuevas, que no hayan sido vistas durante su entrenamiento?*. Con este fin, seleccionamos el mejor modelo obtenido para cada variante propuesta y analizamos su rendimiento al ser evaluados en conversaciones nuevas pertenecientes al subconjunto de control del corpus en inglés (idioma que fue utilizado durante el entrenamiento). Esperamos ver mejoras para los modelos que incluyan información textual.

En sexto lugar, en la Sección 5.3.1, repetimos un experimento similar al anterior, con el objetivo de responder P_4: *¿Cómo se comportan estos modelos ante conversaciones en idiomas distintos a los que fueron utilizados para su entrenamiento?*. En este caso, evaluamos los mismos modelos pero utilizando conversaciones en otro idioma sobre el cual fueron entrenados. Para ello, utilizamos el subconjunto de control del corpus en español. Esperamos ver caídas importantes en los rendimientos de los modelos que incluyen información textual (*bigru[acoustic+text]*, *ms-bigru[acoustic+text]*) al no haber utilizado modelos preentrenados que hayan sido entrenados sobre este lenguaje.

Finalmente, en la Sección 5.3.2, intentamos responder P_5: *¿Podremos modificar los modelos existentes de forma tal que el rendimiento en conversaciones en distintos idiomas mejore y que, a su vez, mantenga resultados similares para conversaciones en el mismo idioma?*. Con este fin, reentrenamos el mejor modelo obtenido en los experimentos anteriores pero, esta vez, con algunos ajustes en los atributos de texto con el objetivo de mejorar el desempeño en otros idiomas. Luego, analizamos el impacto de estos cambios en los rendimientos obtenidos, tanto para conversaciones en el mismo idioma como en otros. Para esto, volvemos a utilizar los subconjuntos de control del corpus en inglés y en español. Esperamos que en este caso el rendimiento obtenido al evaluar en otros idiomas mejore en relación a los anteriores, manteniendo resultados similares en conversaciones en el mismo idioma.

El entrenamiento de los modelos se realizó teniendo en cuenta las siguientes decisiones de diseño:

- Para la búsqueda de mejores hiperparámetros, utilizamos la técnica de *validación cruzada* de tipo *dejar-un-grupo-afuera*. Debido a los altos costos computacionales inherentes al entrenamiento de cada configuración, validamos en todos los casos sobre un único *fold*, que fue manualmente seleccionado de forma tal que contenga suficiente presencia de todas las clases objetivo.

- Utilizamos el método de optimización *Stochastic Gradient Descent*, que ajusta los parámetros del modelo iterativamente cada vez que se evalúa una secuencia completa

perteneciente al subconjunto de entrenamiento.

- En cada iteración, mezclamos el orden de las instancias de entrenamiento para evitar potenciales sesgos que pudiera introducirse en el modelo al observar las instancias en el mismo orden.
- Duplicamos las instancias de entrenamiento invirtiendo los atributos de ambos jugadores, con el objetivo de evitar que el modelo adquiera potenciales sesgos en relación a los participantes.
- El modelo final para una configuración dada se seleccionó utilizando la iteración con menor error en validación (técnica de *Early Stopping*).

Para medir el rendimiento de los modelos sobre el *fold* de validación, tenemos en cuenta el *F1-Score* para cada una de las clases objetivo, así como también la métrica $F1_{Macro}$, que se define como el *F1-Score* promedio de todas las clases objetivo.

La Tabla 5.1 muestra, a modo de resumen, los principales resultados obtenidos a lo largo del capítulo. Las primeras tres filas muestran los resultados asociados a los modelos desarrollados en la Sección 5.2, mientras que los de la última fila al modelo multilenguaje construido en la Sección 5.3.2. Todos los modelos fueron entrenados sobre el corpus en inglés (Columbia).

	Inglés		Español	
	Validación	Control	Validación	Control
bigru[acoustic]	.60	.58	-	**.61**
bigru[acoustic+text]	**.69**	**.63**	-	.49
ms-bigru[acoustic+text]	.60	.50	-	.33
bigru[acoustic+text]-mlang	-	.61	-	.60

Tab. 5.1: *$F1_{Macro}$ al evaluar los mejores modelos obtenidos para cada arquitectura propuesta en los distintos conjuntos de datos. En negrita, los mejores resultados por columna.*

5.1. Replicación de resultados del modelo secuencial

En esta sección, intentamos replicar los mejores resultados obtenidos para el modelo secuencial presentado en el trabajo de Brusco and Gravano (2023) —que en este trabajo llamamos *bigru[acoustic]*, detallado en la Sección 4.3—, con el objetivo de confirmar que estamos trabajando sobre un modelo que obtiene un rendimiento similar a los presentados. Esto es fundamental siendo que gran parte de este trabajo se basa en este modelo.

Resultados

Entrenamos el modelo por 150 iteraciones sobre el subconjunto de desarrollo del corpus en español, utilizando los hiperparámetros que otorgaron los mejores resultados según el trabajo citado: $\#layers = 2$, $\#hidden = 100$, *learning-rate* $= 0.0001$, enmascarado = general, balanceo = pesado.

Luego, seleccionamos la mejor iteración y evaluamos el modelo resultante en conversaciones no vistas anteriormente pertenecientes al subconjunto de control del mismo idioma.

Los resultados del rendimiento del modelo obtenido pueden observarse en la Figura 5.2, en donde comparamos el rendimiento de evaluar el modelo generado en el subconjunto de control con respecto a los resultados esperados.

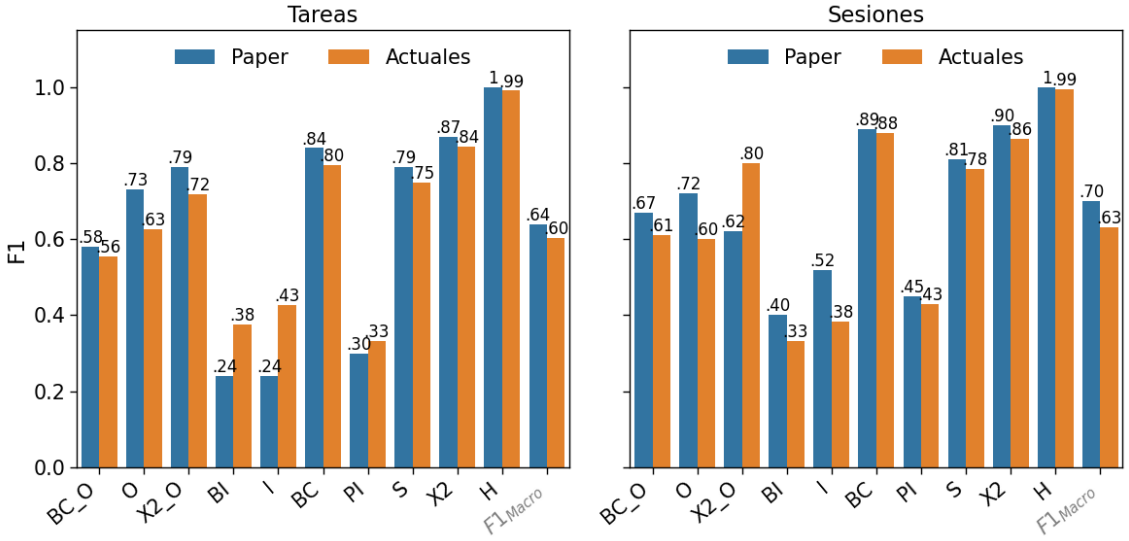


Fig. 5.2: *Resultados obtenidos al evaluar el modelo generado en el subconjunto de control (leyenda "Actuales") junto a los resultados esperados (leyenda "Paper"). La evaluación en control se realizó por separado: por un lado, para las tareas separadas de cada sesión (subfigura izquierda) y, por otro, para las sesiones completas de este conjunto (subfigura derecha).*

En la Figura 5.2 podemos observar que para la mayoría de las clases se obtienen resultados similares, a excepción de algunas como I, BI, O y X2_O, para las cuales notamos diferencias importantes. Además, podemos ver que ciertas relaciones no se mantienen: por ejemplo, X2_O otorga mejores resultados que O al evaluarse en sesiones en un modelo pero peor en el otro.

Debate

Si bien es cierto que existen algunas diferencias para las clases mencionadas, creemos que esto puede deberse principalmente a la forma en la que construimos el modelo, que difiere del método utilizado en el trabajo citado: en este trabajo, seleccionamos el mejor modelo usando el método de *Early Stopping* para un único *fold*, mientras que los autores utilizaron tres *folds* distintos, de los cuales se obtuvo un modelo final a partir de la combinación de sus parámetros. Por otro lado, la mayoría de estas clases son poco frecuentes en el conjunto de datos —tanto en control como en desarrollo, como se puede ver en la Figura 4.1—, por lo cual, es esperable que el modelo no aprenda a reconocerlas de forma óptima y, más aún, que cada error obtenido al evaluarse en control amplifique el error en la métrica de rendimiento para estas clases considerablemente.

En conclusión, teniendo lo anterior en cuenta, los resultados obtenidos parecen reflejar valores acordes a lo expuesto en Brusco and Gravano (2023).

5.2. Resultados en desarrollo

En esta sección, exploramos distintas configuraciones para los modelos planteados en la Sección 4. En todos los casos, utilizamos el subconjunto de desarrollo del corpus en inglés para el entrenamiento de los modelos, junto al resto de las consideraciones planteadas en la introducción de este capítulo. Luego, seleccionamos los mejores modelos de cada tipo —que denotamos con una estrella ('★')— para evaluar posteriormente, en la sección 5.3, en conversaciones nuevas utilizando el subconjunto de control.

5.2.1. Modelo acústico - *bigru[acoustic]*

En esta subsección, exploramos alternativas de modelado para el *bigru[acoustic]* —que, recordemos, no es más ni menos que el modelo secuencial presentado en Brusco and Gravano (2023)—, considerando distintas combinaciones de hiperparámetros. El objetivo de desarrollar este modelo es tener un *baseline* contra el cual comparar resultados posteriormente con el resto de los modelos a desarrollar en las siguientes subsecciones. Recordemos que el modelo desarrollado en la Sección 5.1 fue entrenado sobre el corpus en español, por lo que no resulta de utilidad para realizar comparaciones justas contra el resto de los modelos, siendo que estos fueron entrenados —como detallamos más adelante— sobre el corpus en inglés.

Resultados

Entrenamos una serie de modelos variando la cantidad de capas y la dimensión del estado oculto. En todos los casos se realizaron 150 iteraciones de entrenamiento, utilizando un *learning-rate* de 0.0001. Los resultados se muestran en la tabla de la Figura 5.2.

#Layers	#Hidden	BC	PI	S	X2	BC_O	O	X2_O	BI	I	H	$F1_{\text{Macro}}$
2	128	**.75**	**.36**	**.71**	**.69**	.40	**.72**	.82	.24	**.36**	**.99**	**.60**
1	128	.64	.30	.68	.68	**.60**	.68	.88	.15	.17	.97	.58
2	64	.73	.33	.64	.66	**.60**	.60	.71	**.25**	.19	.99	.57
1	64	.67	.28	**.71**	.60	**.60**	.61	**.86**	.00	.13	.97	.54
Español		.84	.38	.77	.85	.58	.61	.76	.35	.40	.99	.62

Tab. 5.2: *F1-Scores por clase para las configuraciones consideradas. Los modelos fueron evaluados en el fold de validación del subconjunto de desarrollo. Además, incluimos en la última fila, el promedio de los resultados obtenidos para tareas y sesiones en español por el modelo generado en la sección anterior. En negrita, los mejores resultados por clase (sin considerar los resultados en español)*

Los resultados de la Figura 5.2 muestran que el modelo más complejo resulta superior en el $F1_{Macro}$ obtenido, como así también en casi todas las clases, a excepción de BC_O, X2_O y BI, siendo BC_O la única clase para la cual esta diferencia, de 0.2, resulta significativa. Además, podemos ver que, si bien los resultados obtenidos para el corpus en español difieren considerablemente para la mayoría de las clases, las relaciones entre los rendimientos se mantienen.

Debate

Recordemos que el corpus utilizado cuenta con una cantidad limitada de apariciones para las clases mencionadas por lo que, no solo los modelos pueden presentar dificultades para reconocerlas sino que, además, cada error puede amplificar considerablemente los *F1-Scores* asociados. Creemos que la diferencia relacionada a la clase BC_O podría estar relacionada con esto.

Teniendo esto en cuenta, seleccionamos el modelo con la combinación de hiperparámetros $\#layers = 2$, $\#hidden = 128$ como el mejor modelo obtenido de esta variante, al cual nos referimos en futuras secciones como *bigru[acoustic]*$^{\bigstar}$.

5.2.2. Modelo con texto - *bigru[acoustic+text]*

En esta sección exploramos distintas configuraciones y estrategias de modelado para el modelo *bigru[acoustic+text]* —detallado en la sección 4.4—, con el objetivo de responder P_1: *¿Es posible extender el modelo original de forma tal que reconozca, además de los patrones acústico-prosódicos, patrones en las transcripciones de las conversaciones? ¿Cómo afecta a su rendimiento?*

Resultados

En el primer experimento, cuyo resultado se puede observar en la Figura 5.3a, entrenamos por 150 iteraciones el modelo variando la cantidad de capas y el tamaño del estado oculto, de forma similar al experimento realizado en la sección anterior. Fijamos el *learning-rate* en 0.0001 y, con respecto a los atributos de texto, se tuvieron las siguientes consideraciones:

- Para generar los *embeddings*, se utilizó un modelo preentrenado de BERT (sin distinción de mayúsculas y minúsculas), publicado en Devlin et al. (2018b). Al día de la fecha, el modelo se encuentra disponible en HuggingFace[1].
- Para generar las etiquetas de *part-of-speech*, utilizamos el modelo preentrenado de Flair para inglés, publicado en Akbik et al. (2018), que luego codificamos mediante *one-hot encodings*. El modelo se encuentra, al día de la fecha, disponible en HuggingFace[2].

.

En segundo lugar, repetimos el mismo experimento pero, esta vez, aplicamos un paso previo de reducción de dimensionalidad en los atributos de texto. El objetivo de este experimento es analizar si la reducción de la cantidad de parámetros del modelo facilita su aprendizaje. Para realizarlo, utilizamos PCA (*Principal Components Analysis*, en inglés) como método de reducción y lo aplicamos sobre la totalidad de los atributos de texto, llevándolos a dimensión 4. Este paso de reducción se realizó considerando la covarianza de los atributos de la totalidad de las palabras del corpus, lo cual entendemos que puede no ser la mejor decisión desde un punto de vista metodológico al incluir en este paso también palabras pertenecientes al subconjunto de control. Los resultados de este experimento se pueden observar en la Figura 5.3b.

[1] https://huggingface.co/google-bert/bert-base-uncased
[2] https://huggingface.co/flair/pos-english

#Layers	#Hidden	BC	PI	S	X2	BC_O	O	X2_O	BI	I	H	$F1_{\text{Macro}}$
2	128	**.77**	**.46**	**.71**	.58	**.92**	.76	.82	.00	.27	.95	**.62**
1	128	.71	.40	.50	.55	.86	.73	**.88**	.00	.26	.87	.57
2	64	.62	.34	.48	.60	.80	.70	.78	.00	.35	.97	.56
1	64	.70	.39	.54	.58	.86	**.80**	.82	.14	.32	.86	.60
bigru[acoustic]★		.75	.36	**.71**	**.69**	.40	.72	.82	**.24**	**.36**	**.99**	.60

(a) *Sin PCA*

#Layers	#Hidden	BC	PI	S	X2	BC_O	O	X2_O	BI	I	H	$F1_{\text{Macro}}$
2	128	.74	.39	.67	.68	.62	**.77**	**.93**	**.47**	**.58**	**1.00**	**.69**
1	128	.69	**.48**	.74	.63	.40	.72	.78	.26	.50	.96	.62
2	64	.67	.36	.68	.65	.55	.69	.74	.27	.33	.99	.59
1	64	**.77**	.43	**.81**	**.70**	**.80**	**.77**	.75	.42	.42	.99	**.69**
bigru[acoustic]★		.75	.36	.71	.69	.40	.72	.82	.24	.36	.99	.60

(b) *Con PCA*

Fig. 5.3: *F1-Scores por clase para las configuraciones consideradas. Los modelos fueron evaluados en el fold de validación del subconjunto de desarrollo. En negrita, los mejores resultados por clase. Además, incluimos el mejor modelo acústico como baseline.*

En la Figura 5.3a, para los modelos sin el paso de reducción dimensional, podemos observar que el modelo más complejo (con parámetros $\#layers = 2$, $\#hidden = 128$) resultó superior considerando el $F1_{Macro}$, superando incluso —aunque sutilmente— al modelo *bigru[acoustic]*★. Más precisamente, las clases BC, PI, BC_O y O de este modelo obtienen mejores resultados que *bigru[acoustic]*★, pero peores para las clases X2, BI, I, H. Además, podemos notar, en líneas generales, un rendimiento particularmente pobre para la clase BI en todos los casos.

En la Figura 5.3b, que muestra los resultados para los modelos con el paso de reducción de dimensionalidad, podemos notar que el modelo más complejo ($\#layers = 2$, $\#hidden = 128$) y el más simple ($\#layers = 1$, $\#hidden = 64$) obtienen los mejores resultados, superando a *bigru[acoustic]*★ con una diferencia más pronunciada para el $F1_{Macro}$ y para el *F1-Score* de la mayoría de las clases. El modelo más simple muestra mejores resultados para la mayoría de transiciones sin solapamiento (a excepción de PI), mientras que el modelo más complejo muestra los mejores resultados para las clases con solapamiento.

En ambas subfiguras de la Figura 5.3, podemos observar, en casi todos los casos, un incremento de performance bastante pronunciado en relación a BC_O.

Debate

Considerando los resultados obtenidos, podemos concluir que introducir los atributos textuales puede ser beneficioso para mejorar el rendimiento en el etiquetado de transiciones de cambio de turno, especialmente aplicando un paso previo de reducción de dimensionalidad. Esto aporta evidencia para responder P_1: *¿Es posible extender el modelo original de forma tal que reconozca, además de los patrones acústico-prosódicos, patrones en las transcripciones de las conversaciones? ¿Cómo afecta a su rendimiento?.*

Además, pudimos observar que en la mayoría de los modelos obtuvimos mejoras para las clases asociadas a *backchannels* (BC, BC_O) —principalmente en esta última—, lo cual se condice con los resultados presentados en Gravano and Hirschberg (2011).

Los mejores resultados obtenidos son arrojados por el modelo más simple y más complejo, aplicando el paso de reducción de dimensionalidad en los atributos de texto, obteniendo ambos un $F1_{Macro}$ de 0.69. Elegir uno u el otro depende en última instancia de en qué clases se desea obtener mayor precisión. En este trabajo, seleccionamos el modelo más complejo ($\#layers = 2$, $\#hidden = 128$) como el modelo seleccionado de esta sección que será utilizado para futuros experimentos y para el cual nos referiremos en futuras secciones como *bigru[acoustic+text]*$^{\bigstar}$.

Como trabajo futuro, queda pendiente experimentar con una mayor variedad de alternativas con respecto a la dimensión de los vectores destino luego de aplicar el paso de reducción dimensional, así como también cambiar la estrategia de qué atributos reducir o utilizar distintos métodos de reducción (como TSNE, por ejemplo). Además, queda pendiente experimentar con una mayor variedad de alternativas para generar los *embeddings* y etiquetas *part-of-speech* en los atributos de texto.

5.2.3. Modelo multiescala - *ms-bigru[acoustic+text]*

En esta sección, analizamos el modelo *ms-bigru[acoustic+text]* —detallado en la sección 4.5—, con el objetivo de responder P_2: *¿Cómo impacta construir un modelo con una arquitectura diseñada específicamente para concatenación de series temporales multimodales?* Con este fin, construimos una variedad de modelos de este tipo variando sus hiperparámetros y estrategias de modelado.

Resultados

En el primer experimento, cuyo resultado podemos observar en la figura 5.4a, buscamos construir distintos modelos de este tipo, variando la cantidad de capas y el tamaño de los estados ocultos. Para ello, entrenamos cada uno de los modelos por 150 iteraciones fijando el *learning-rate* en 0.0001 y teniendo en cuenta las siguientes consideraciones de diseño:

- Todas las subredes del modelo (tanto la red *master* como cada subred asociada a las distintas modalidades) cuentan con la misma cantidad de capas y dimensión del estado oculto.
- La subred asociada a los atributos acústicos se evalúa en intervalos regulares, cada 10 milisegundos; la subred asociada a los atributos de texto se evalúa en intervalos irregulares, cada vez que termina una palabra; la subred *master* se evalúa en intervalos irregulares, cada vez que ocurre una transición de turno.
- Para los atributos de texto, utilizamos los mismos modelos preentrenados utilizados para el modelo *bigru[acoustic+text]*, descritos en la Sección 5.2.2.

En segundo lugar, al igual que para los modelos anteriores, repetimos el experimento de aplicar un paso de reducción dimensional sobre los atributos de texto, teniendo en cuenta las mismas consideraciones anteriormente mencionadas. Los resultados de este experimento se pueden observar en la Figura 5.4b

#Layers	#Hidden	BC	PI	S	X2	BC_O	O	X2_O	BI	I	H	$F1_{\text{Macro}}$
2	64	.59	.33	.61	.67	.62	.66	**.88**	.23	**.53**	.84	**.60**
1	128	.57	.24	.57	.59	.71	**.72**	.78	.22	.29	.82	.55
2	128	.56	.29	.60	.52	**.80**	.51	.78	.18	.35	.82	.54
1	64	.61	.16	.59	.61	.67	.64	.78	.11	.19	.80	.52
bigru[acoustic]★		**.75**	**.36**	**.71**	**.69**	.40	**.72**	.82	**.24**	.36	**.99**	**.60**

(a) *Sin PCA*

#Layers	#Hidden	BC	PI	S	X2	BC_O	O	X2_O	BI	I	H	$F1_{\text{Macro}}$
2	128	.61	.17	.54	**.69**	.50	.66	**.93**	.15	.22	.83	.53
1	128	.54	.26	.46	.55	.53	.66	.78	.08	.32	.79	.50
2	64	.61	.22	.46	.56	.55	.56	.67	.08	.33	.83	.49
1	64	.55	.28	.49	.57	**.62**	.55	.75	.00	.18	.82	.48
bigru[acoustic]★		**.75**	**.36**	**.71**	**.69**	.40	**.72**	.82	**.24**	**.36**	**.99**	**.60**

(b) *Con PCA*

Fig. 5.4: *F1-Scores por clase para las configuraciones consideradas. Los modelos fueron evaluados en el fold de validación del subconjunto de desarrollo. En negrita, los mejores resultados por clase. Además, incluimos el mejor modelo bigru[acoustic] como baseline.*

En la Figura 5.4a vemos que el mejor modelo se obtiene con la combinación de hiperparámetros $\#layers = 2$ y $\#hidden = 64$, obteniendo un rendimiento de $F1_{Macro} = 0.60$, que coincide con el *bigru[acoustic]*★. Más precisamente, podemos ver que este modelo incrementa la performance en las categorías BC_O, X2_O y I, mientras que la disminuye en el resto.

En la Figura 5.4b podemos ver que el mejor modelo resulta el de la combinación $\#layers = 2$ y $\#hidden = 64$, que incrementa la performance de X2_O y BC_O con respecto al modelo base pero la disminuye para el resto de las clases, así como también lo hace con el $F1_{Macro}$.

En resumen, la Figura 5.4 muestra que en ningún caso se logra superar los resultados del *bigru[acoustic]*★. Además, podemos observar en ambas figuras que, al igual que en el modelo anterior, se obtienen incrementos de performance consistentemente para la clase BC_O.

Resultados - Con atributos de duraciones

Por otro lado, nos preguntamos si agregar atributos adicionales con respecto a la duración de las palabras y los silencios puede contribuir a la performance de este modelo. El modelo *bigru[acoustic+text]* puede aprender a interpretar estas propiedades al recibir fragmentos en intervalos de tiempo fijos en donde, en cada uno de ellos, especificamos la palabra en curso de cada hablante. De esta forma, el modelo podría inferir la duración de cada palabra en base a la secuencia de fragmentos en donde esta ocurrió, así como también el silencio transcurrido desde la palabra anterior o hasta la palabra siguiente, de una forma similar. Luego, siendo que el *ms-bigru[acoustic+text]* no recibe estos fragmentos en intervalos regulares —perdiendo así la capacidad de inferir estas propiedades—, creemos que podría tener sentido incluirlas explícitamente junto a los atributos de cada jugador, de la siguiente manera:

- *Duración de la palabra*: duración (medida en segundos) de la palabra actual o 0 en caso de silencio.
- *Tiempo transcurrido desde la última palabra*: intervalo de tiempo (medido en segundos) desde que el jugador en cuestión terminó de pronunciar la última palabra.
- *Tiempo transcurrido hasta la palabra siguiente*: intervalo de tiempo (medido en segundos) hasta que el jugador en cuestión comience a pronunciar la siguiente palabra.

De esta manera, la tabla 5.5 muestra los resultados de repetir los mismos experimentos anteriores pero considerando estas modificaciones en el modelo.

#Layers	#Hidden	BC	PI	S	X2	BC_O	O	X2_O	BI	I	H	$F1_{\text{Macro}}$
2	64	.56	.23	.64	.58	**.77**	.70	.82	.10	.35	.82	.56
2	128	.59	.23	.64	.61	.56	.68	.82	.17	.25	.85	.54
1	64	.56	.19	.60	.58	.67	**.73**	**.88**	.11	.21	.83	.54
1	128	.55	.32	.62	.59	.56	.67	.78	.21	.20	.84	.53
bigru[acoustic]*		**.75**	**.36**	**.71**	**.69**	.40	.72	.82	**.24**	**.36**	**.99**	**.60**

(a) *Sin PCA*

#Layers	#Hidden	BC	PI	S	X2	BC_O	O	X2_O	BI	I	H	$F1_{\text{Macro}}$
2	64	.64	.25	.59	.68	**.67**	.56	.75	.08	**.38**	.89	.55
1	64	.57	.25	.56	.60	.73	.61	.71	.20	.29	.88	.54
2	128	.62	.23	.60	.65	.67	.60	.71	.08	.32	.86	.53
1	128	.54	.19	.49	.60	.62	.56	.71	.19	.32	.84	.51
bigru[acoustic]*		**.75**	**.36**	**.71**	**.69**	.40	**.72**	**.82**	**.24**	.36	**.99**	**.60**

(b) *Con PCA*

Fig. 5.5: *F1-Scores por clase para las configuraciones consideradas. Los modelos fueron evaluados en el fold de validación del subconjunto de desarrollo. En negrita, los mejores resultados por clase. Además, incluimos el mejor bigru[acoustic] como baseline.*

En la Figura 5.5 podemos observar que tampoco se pudo superar los resultados obtenidos por el *bigru[acoustic]** para ninguna combinación de hiperparámetros, al igual que en los experimentos anteriores. En el único caso que se puede observar un consistente incremento de performance es para la clase BC_O; en el resto de los casos, la performance resulta similar o peor.

Debate

En base a los resultados obtenidos, seleccionamos como mejor modelo de este tipo a la variante con hiperparámetros $\#layers = 2$, $\#hidden = 64$ obtenido en el experimento 5.4a, sin reducción dimensional ni atributos adicionales de duraciones, al cual a partir de ahora nos referimos como *ms-bigru[acoustic+text]**.

Por otro lado, parecería que no tenemos evidencia suficiente para responder P_2: *¿Cómo impacta construir un modelo con una arquitectura diseñada específicamente para concatenación de series temporales multimodales?*. Si bien observamos incrementos de performance para algunas clases puntuales como BC_O, en ningún caso el rendimiento general superó

a *bigru[acoustic]*$^{\bigstar}$. Más aún, agregar atributos adicionales de duraciones no parece haber mejorado el rendimiento general.

Por último, si bien no pudimos obtener los resultados esperados con respecto a esta arquitectura, creemos que tiene el potencial para hacerlo y, debido a su mayor complejidad y la limitada capacidad de cómputo disponible durante este trabajo, queda como trabajo futuro explorar otras alternativas que podrían otorgar mejores resultados. Por ejemplo, se podría considerar utilizar valores distintos para la cantidad de capas y estado oculto de cada subred, así como también muestrear cada una de estas en intervalos fijos, variables e incluso evaluando distintas combinaciones de las mismas.

5.2.4. Mejores modelos

En las secciones anteriores analizamos distintas variantes para cada una de las arquitecturas desarrolladas en la sección 4 y seleccionamos como mejores modelos —que identificamos con estrellas '$\bigstar$'— a aquellos que otorgaron mejores resultados, según el $F1_{Macro}$. La Figura 5.3 resume los resultados obtenidos para estos modelos, que evaluamos en la sección siguiente en los subconjuntos de control.

	BC	PI	S	X2	BC_O	O	X2_O	BI	I	H	$F1_{\text{Macro}}$
bigru[acoustic+text]$^{\bigstar}$	.74	**.39**	.67	.68	.62	**.77**	**.93**	**.47**	**.58**	**1.00**	**.69**
ms-bigru[acoustic+text]$^{\bigstar}$	.59	.33	.61	.67	.62	.66	.88	.23	.53	.84	.60
bigru[acoustic]$^{\bigstar}$	**.75**	.36	**.71**	**.69**	.40	.72	.82	.24	.36	.99	.60

Tab. 5.3: *F1-Scores por clase para los modelos seleccionados de cada arquitectura, evaluados en el fold de validación, del subconjunto de desarrollo. En negrita, los mejores resultados por clase.*

En la próxima sección, utilizamos estos mismos modelos para evaluar en el subconjunto de control (en lugar de reentrenarlos utilizando, además, las conversaciones pertenecientes al subconjunto de validación) puesto que, si lo hiciéramos, (1) no tendríamos métricas en control que sean comparables con los resultados obtenidos en validación; y (2) no podríamos utilizar el método de *Stochastic Gradient Descent* para ajustar los parámetros en cada iteración del entrenamiento, dado que este requiere de un conjunto de validación para lograrlo.

5.3. Resultados en control

En la Sección 5.3.1, evaluamos los resultados de los modelos seleccionados en desarrollo (ver Figura 5.3) en conversaciones nuevas pertenecientes al subconjunto de control, es decir, conversaciones que no fueron utilizadas durante el entrenamiento de los modelos y, por lo tanto, nunca fueron vistas por ellos.

Luego, seleccionamos el mejor modelo entre ellos y, en la Sección 5.3.2 lo reentrenamos con algunas modificaciones para mejorar el rendimiento en conversaciones provenientes de idiomas distintos a los utilizados durante el entrenamiento. Finalmente, evaluamos el nuevo modelo en los subconjuntos de control y comparamos los resultados obtenidos en relación a su alternativa de un solo idioma.

5.3.1. Mejores modelos de desarrollo

Generalización en mismo idioma

En primer lugar, evaluamos el rendimiento de los modelos seleccionados en la sección 5.3 en el subconjunto de control del corpus en inglés, mismo idioma en el cual estos modelos fueron entrenados. Los resultados se muestran en la Figura 5.4.

	BC	PI	S	X2	BC_O	O	X2_O	BI	I	H	$F1_{\text{Macro}}$
bigru[acoustic+text]★	**.79**	.33	**.73**	**.79**	**.64**	**.68**	**.67**	**.34**	.36	**.99**	**.63**
ms-bigru[acoustic+text]★	.54	.08	.62	.60	.39	**.68**	**.67**	.23	.32	.84	.50
bigru[acoustic]★	.76	**.34**	.67	.77	.60	.58	.58	.19	**.37**	**.99**	.58

Tab. 5.4: *F1-Scores por clase para los modelos seleccionados de cada arquitectura, evaluados en el subconjunto de control en inglés. Además, incluimos los resultados del mejor modelo bigru[acoustic] como baseline. En negrita, los mejores resultados por clase.*

En la Figura 5.4 observamos que el modelo *bigru[acoustic+text]*★ resulta superador obteniendo el mejor $F1_{\text{Macro}}$ de 0.63 y superando en casi todas las clases al resto de los modelos, a excepción de I y PI, en donde *bigru[acoustic]*★ lo supera (aunque por una diferencia mínima). Además, vemos que el rendimiento de *ms-bigru[acoustic+text]*★ resulta bastante pobre, disminuyendo los resultados con respecto al modelo *bigru[acoustic]*★ en casi todos los casos, excepto para O y BI.

Generalización en otros idiomas

En esta sección, realizamos un experimento similar al de la sección anterior pero, en este caso, evaluamos a los modelos seleccionados ante conversaciones en un idioma distinto al que fue utilizado para el entrenamiento de estos modelos. En este caso, utilizamos el corpus UBA, en español, siendo un idioma distinto al corpus utilizado para entrenar a los modelos. Los resultados de este experimento se pueden observar en la Figura 5.5.

	BC	PI	S	X2	BC_O	O	X2_O	BI	I	H	$F1_{\text{Macro}}$
bigru[acoustic+text]★	.73	.25	.70	.42	**.51**	.54	.07	.36	.32	**.99**	.49
ms-bigru[acoustic+text]★	.24	.00	.33	.31	.30	**.66**	.44	.12	.13	.76	.33
bigru[acoustic]★	**.81**	**.32**	**.74**	**.74**	.45	.57	**.72**	**.38**	**.40**	**.99**	**.61**

Tab. 5.5: *F1-Scores por clase para los modelos seleccionados de cada arquitectura, evaluados en el subconjunto de control en español. En negrita, los mejores resultados por clase. Además, incluimos los resultados del mejor modelo bigru[acoustic] como baseline.*

En la Figura 5.5 observamos que, curiosamente, el modelo *bigru[acoustic]*★ otorgó mejores resultados con respecto a la evaluación en el subconjunto de control en el mismo idioma (ver Tabla 5.5), realizado en la sección anterior. Por otro lado, vemos un deterioro de *performance* para el resto de los modelos, con excepción de la clase BC_O, para la cual *bigru[acoustic+text]*★ otorga mejores resultados, que también notamos en desarrollo.

Debate

En esta sección pudimos observar el rendimiento de los modelos seleccionados en nuevas conversaciones en el mismo idioma y en otros, aportando evidencia para responder (P_3) *¿Cómo se comportan estos modelos ante conversaciones nuevas, que no hayan sido vistas durante su entrenamiento?* y (P_4) *¿Cómo se comportan estos modelos ante conversaciones en idiomas distintos a los que fueron utilizados para su entrenamiento?*.

En base a los resultados obtenidos, podemos concluir que *bigru[acoustic+text]*$^\bigstar$ parece ser la mejor opción al evaluar en conversaciones en el mismo idioma en el cual los modelos fueron entrenados, mejorando el rendimiento para la mayoría de las clases. Por el contrario, para conversaciones en distintos idiomas, el modelo *bigru[acoustic]*$^\bigstar$ parecería ser la mejor opción, a menos que se requiera precisión específicamente para la clase BC_O, en cuyo caso el modelo *bigru[acoustic+text]*$^\bigstar$ obtuvo mejores resultados que este.

El deterioro de rendimiento de los modelos con atributos de texto (*bigru[acoustic+text]*$^\bigstar$ y *ms-bigru[acoustic+text]*$^\bigstar$) no resulta sorprendente ya que los modelos preentrenados utilizados para para generar los atributos de texto se entrenaron sobre conjuntos de datos exclusivamente en inglés. Luego, resulta lógico pensar que estos no cumplan su función de manera eficaz en otros lenguajes y, en consecuencia, las redes no sean capaces de aprender patrones útiles sobre ellos.

En la próxima sección, reentrenamos el *bigru[acoustic+text]*$^\bigstar$ —siendo el que otorgó los mejores resultados para las alternativas planteadas en este trabajo— con ciertas modificaciones en los atributos de texto con las cuales esperamos contrarrestar, en alguna medida, la baja *performance* obtenida en conversaciones nuevas en distintos idiomas.

5.3.2. Mejor modelo, reentrenado con atributos multilenguaje

En la sección anterior, observamos como *bigru[acoustic+text]*$^\bigstar$ y *ms-bigru[acoustic+text]*$^\bigstar$, que incluyen atributos de texto, obtuvieron un notable deterioro de resultados al evaluarse en conversaciones en idiomas distintos sobre los que fueron entrenados. Como anticipamos anteriormente, creemos que esto se debe a los modelos preentrenados con los cuales generamos los atributos de texto. En esta sección, intentamos responder P_6: *¿Podremos modificar los modelos existentes de forma tal que el rendimiento en conversaciones en distintos idiomas mejore y que, a su vez, mantenga resultados similares para conversaciones en el mismo idioma?*. Para ello, nos proponemos reentrenar al modelo *bigru[acoustic+text]*$^\bigstar$ —mejor modelo obtenido basándonos en los resultados del experimento anterior— pero, esta vez, con las siguientes modificaciones:

- Para generar los *embeddings*, utilizamos otro modelo preentrenado de BERT (sin distinción de mayúsculas y minúsculas) que fue entrenado con datos en más de 100 lenguajes, incluidos inglés y español, publicado en Devlin et al. (2018a). El modelo se encuentra, al día de la fecha, disponible en HuggingFace[3].

- Para generar las etiquetas de *part-of-speech*, utilizamos otro modelo preentrenado, publicado en Akbik et al. (2018), que fue entrenado con múltiples corpus en distintos idiomas, incluyendo español e inglés. Al día de la fecha, el modelo se encuentra disponible en HuggingFace[4].

[3] https://huggingface.co/google-bert/bert-base-multilingual-uncased
[4] https://huggingface.co/flair/upos-multi

Para el entrenamiento del modelo, se tuvieron en cuenta exactamente las mismas consideraciones que para el entrenamiento del resto de los modelos de desarrollo, según lo detallado en la Sección 5.2. Además, se utilizaron los mismos hiperparámetros y arquitectura del modelo *bigru[acoustic+text]*★.

En las secciones siguientes, evaluamos nuevamente el modelo reentrenado —al que a partir de ahora nos referiremos como *bigru[acoustic+text]-mlang*★— en el subconjunto de control con conversaciones en el mismo y distintos idiomas al utilizado para el entrenamiento, tal como realizamos en la sección anterior.

Generalización en mismo idioma

Tal como anticipamos en la introducción, evaluamos nuevamente el modelo reentrenado, *bigru[acoustic+text]-mlang*★, utilizando conversaciones pertenecientes al subconjunto de control del mismo idioma en el cual fue entrenado. Los resultados se pueden observar en la Figura 5.6.

	BC	PI	S	X2	BC_O	O	X2_O	BI	I	H	$F1_{\text{Macro}}$
bigru[acoustic+text]-mlang★	**.82**	.33	**.73**	**.81**	.61	**.75**	.61	.20	.22	**.99**	.61
bigru[acoustic]★	.76	**.34**	.67	.77	.60	.58	.58	.19	**.37**	**.99**	.58
bigru[acoustic+text]★	.79	.33	**.73**	.79	**.64**	.68	**.67**	**.34**	.36	**.99**	**.63**

Tab. 5.6: *F1-Scores por clase para el modelo bigru[acoustic+text]-mlang, evaluado en el subconjunto de control en inglés. Además, agregamos los resultados de los mejores modelos bigru[acoustic] y bigru[acoustic+text] como baselines. En negrita, los mejores resultados por clase.*

En la Figura 5.6 vemos que el modelo *bigru[acoustic+text]-mlang*★ disminuyó sutilmente el rendimiento con respecto a *bigru[acoustic+text]*★, aunque este continúa siendo superior a *bigru[acoustic]*★. Más precisamente, vemos que estas modificaciones brindaron mejores resultados para las clases BC, X2 y O, pero iguales o peores para el resto.

Generalización en otros idiomas

En el próximo experimento, cuyos resultados podemos visualizar en la tabla de la Figura 5.7, evaluamos a *bigru[acoustic+text]-mlang*★ en conversaciones pertenecientes al subconjunto de control en español, cuyos datos son de un idioma distinto a los utilizados para su entrenamiento.

	BC	PI	S	X2	BC_O	O	X2_O	BI	I	H	$F1_{\text{Macro}}$
bigru[acoustic+text]-mlang★	.72	**.33**	**.74**	.72	.41	**.70**	.68	**.41**	.28	**.99**	.60
bigru[acoustic]★	**.81**	.32	**.74**	**.74**	.45	.57	**.72**	.38	**.40**	**.99**	**.61**
bigru[acoustic+text]★	.73	.25	.70	.42	**.51**	.54	.07	.36	.32	**.99**	.49

Tab. 5.7: *F1-Scores por clase para el mejor modelo bigru[acoustic+text]-mlang, evaluado en el subconjunto de control en español. Además, agregamos los resultados de los mejores modelos bigru[acoustic] y bigru[acoustic+text] como baseline. En negrita, los mejores resultados por clase.*

Los resultados de la Figura 5.7 muestran un incremento significativo del rendimiento con respecto a *bigru[acoustic+text]*$^{\bigstar}$, superándolo en la mayoría de las clases a excepción de BC, BC_O y I. Por otro lado, vemos que, si bien no se logró superar el $F1_{\text{Macro}}$ del *bigru[acoustic]*$^{\bigstar}$, se obtuvo uno muy similar. Más aún, observamos que incrementó el rendimiento para las clases PI, O y BI, aunque se mantuvo o disminuyó en el resto de los casos.

Debate

En base a los resultados obtenidos en esta sección, podemos concluir que las modificaciones propuestas contribuyeron positivamente en los resultados al ser evaluado en conversaciones para idiomas distintos a los utilizados durante la etapa de entrenamiento. Sin embargo, parecería ser que los resultados para conversaciones en el mismo idioma otorgan ligeramente peores resultados en su alternativa de un lenguaje.

En resumen, creemos que utilizar estas modificaciones puede ser viable para obtener mejor rendimiento en otros idiomas, aunque sacrificando, potencialmente, precisión en el mismo idioma, lo cual responde P_6: *¿Podremos modificar los modelos existentes de forma tal que el rendimiento en conversaciones en distintos idiomas mejore y que, a su vez, mantenga resultados similares para conversaciones en el mismo idioma?.*

6. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo nos propusimos analizar la inclusión de atributos léxicos y sintácticos para el etiquetado *offline* de transiciones de turno. Es decir, exploramos la construcción de modelos que procesan conversaciones de manera completa y etiquetan el tipo de transición de turno que ocurrió en cada instante de tiempo. Las transiciones incluyen cambios de turno suaves, interrupciones y señales de denotación de atención sin intención de tomar el turno, entre otras.

Con este fin, realizamos distintos experimentos aplicando una variedad de estrategias de aprendizaje automático para el desarrollo de modelos que resuelvan esta tarea. Todos ellos fueron entrenados utilizando exclusivamente conversaciones en inglés y evaluados en su capacidad de generalización ante conversaciones nuevas (no utilizadas durante el entrenamiento), tanto en el mismo idioma como en español.

Nuestro punto de partida fue el modelo secuencial presentado en Brusco and Gravano (2023), que en esta tesis llamamos *bigru[acoustic]* y que utiliza únicamente atributos acústico-prosódicos para resolver esta tarea. Al replicar este modelo, obtuvimos un rendimiento F_1 de **58 %** y **61 %** para inglés y español, respectivamente.

En segundo lugar, analizamos la variante que llamamos *bigru[acoustic+text]*, que extiende el modelo anterior con atributos léxico-sintácticos, a partir de la inclusión de etiquetas gramaticales (*PoS tags*) y *embeddings* de las palabras, lo cual implicó realizar un sobremuestreo de los atributos de texto para equiparar la frecuencia de muestreo de las series temporales correspondientes a los atributos acústicos (que ocurren con mayor frecuencia). Para la codificación de las palabras utilizamos *embeddings* BERT, generados por un modelo preentrenado únicamente en inglés (*bert-base-uncased*[1]). Mediante diversos experimentos, observamos mejores resultados que el modelo *bigru[acoustic]* para conversaciones en inglés, alcanzando un puntaje F_1 de **63 %** en conversaciones nuevas. Estos resultados sugieren que el modelo logró encontrar patrones útiles en los atributos relacionados con las palabras enunciadas en la conversación.

En tercer lugar, analizamos una variante de modelado basada en la propuesta multiescala de Roddy et al. (2018), a la que llamamos *ms-bigru[acoustic+text]*. Esta arquitectura es más compleja que una red recurrente convencional, ya que está específicamente diseñada para el procesamiento multimodal a través de subredes que procesan series temporales (según las distintas modalidades) a distintas velocidades. En este caso, no obtuvimos el desempeño esperado, alcanzando un puntaje F_1 de solo **50 %** para conversaciones nuevas. En el caso particular de esta arquitectura, creemos que podría explorarse el uso de distintos intervalos de tiempo e hiperparámetros para cada una de las subredes.

En cuarto lugar, nos propusimos reentrenar el modelo *bigru[acoustic+text]*, pero esta vez utilizando *embeddings* multilenguaje (generados con *bert-base-multilingual-uncased*[2], un modelo preentrenado sobre múltiples idiomas que incluyen al español y al inglés). Los resultados obtenidos muestran (1) mejoras con respecto al modelo acústico en inglés (F_1 **de 61 %** vs el **58 %** alcanzado por *bigru[acoustic]*); (2) deterioro de rendimiento con respecto al del modelo acústico en español (F_1 de **60 %** vs el **61 %** alcanzado por *bigru[acoustic]*); y (3) leves deterioros con respecto al rendimiento alcanzado por el modelo léxico entrenado

[1] https://huggingface.co/google-bert/bert-base-uncased
[2] https://huggingface.co/google-bert/bert-base-multilingual-uncased

sobre inglés (F_1 **de 61 %** vs el **63 %** alcanzado por *bigru[acoustic+text]*). Concluimos que la elección de implementar esta modificación en el modelo *bigru[acoustic+text]* resulta útil ante casos de uso donde sea necesario mejorar el rendimiento multilenguaje, pagando el costo de una potencial disminución en el rendimiento sobre el mismo idioma.

Los resultados de este trabajo parecen respaldar la hipótesis de que agregar los atributos léxicos y sintácticos puede resultar beneficioso para el etiquetado automático *offline* de transiciones de cambio de turno. Creemos que, si bien en algunos casos los resultados no lograron superar los del modelo *bigru[acoustic]* (como en el caso de *ms-bigru[acoustic+text]*), hay mucho espacio para mejorarlos. Por ejemplo, se pueden explorar alternativas para generar los *embeddings* o etiquetas *part-of-speech*; agregar más información de contexto en los atributos; e incluso aplicar de otra forma el paso de reducción dimensional, reduciendo solo una parte de los atributos o cambiando los métodos de reducción. Por último, sería interesante también extender el conjunto de datos con otro tipo de diálogos más diversos, para asegurarnos de que los modelos no sobreajustan a las características propias de los diálogos orientados a tareas del corpus utilizado.

Bibliografía

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649.

Beattie, G. W. (1982). Turn-taking and interruption in political interviews: Margaret thatcher and jim callaghan compared and contrasted.

Brusco, P. and Gravano, A. (2023). Automatic offline annotation of turn-taking transitions in task-oriented dialogue. *Computer Speech & Language*, 78:101462.

Chang, S.-y., Li, B., Sainath, T. N., Zhang, C., Strohman, T., Liang, Q., and He, Y. (2022). Turn-taking prediction for natural conversational speech. *arXiv preprint arXiv:2208.13321*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chowdhury, S. A. (2017). *Computational modeling of turn-taking dynamics in spoken conversations*. PhD thesis, University of Trento.

Chowdhury, S. A., Stepanov, E. A., Danieli, M., and Riccardi, G. (2019). Automatic classification of speech overlaps: feature representation and algorithms. *Computer Speech & Language*, 55:145–167.

DeVault, D., Mell, J., and Gratch, J. (2015). Toward natural turn-taking in a virtual human negotiation agent. In *2015 AAAI Spring Symposium Series*.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018a). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018b). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Duncan, S. (1972). Some signals and rules for taking speaking turns in conversations. *Journal of personality and social psychology*, 23(2):283.

Ekstedt, E. and Skantze, G. (2020). Turngpt: a transformer-based language model for predicting turn-taking in spoken dialog. *arXiv preprint arXiv:2010.10874*.

Eyben, F., Wöllmer, M., and Schuller, B. (2010). Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462.

Gravano, A. and Hirschberg, J. (2011). Turn-taking cues in task-oriented dialogue. *Computer Speech & Language*, 25(3):601–634.

Inoue, K., Lala, D., Ochi, K., Kawahara, T., and Skantze, G. (2024). An analysis of user behaviours for objectively evaluating spoken dialogue systems. *arXiv preprint arXiv:2401.04867*.

Ishii, R., Ren, X., Muszynski, M., and Morency, L.-P. (2022). Trimodal prediction of speaking and listening willingness to help improve turn-changing modeling. *Frontiers in Psychology*, 13:774547.

Jiang, B., Ekstedt, E., and Skantze, G. (2023). Response-conditioned turn-taking prediction. *arXiv preprint arXiv:2305.02036*.

Jin, G., Zhu, T., Akram, M. W., Jin, Y., and Zhu, C. (2020). An adaptive anti-noise neural network for bearing fault diagnosis under noise and varying load conditions. *IEEE access*, 8:74793–74807.

Lala, D., Inoue, K., and Kawahara, T. (2018). Evaluation of real-time deep learning turn-taking models for multiple dialogue scenarios. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 78–86.

Lala, D., Inoue, K., and Kawahara, T. (2019). Smooth turn-taking by a robot using an online continuous model to generate turn-taking cues. In *2019 International Conference on Multimodal Interaction*, pages 226–234.

Liu, C., Ishi, C. T., and Ishiguro, H. (2017). Turn-taking estimation model based on joint embedding of lexical and prosodic contents. In *Interspeech*, pages 1686–1690.

Maier, A., Hough, J., Schlangen, D., et al. (2017). Towards deep end-of-turn prediction for situated spoken dialogue systems.

Masumura, R., Asami, T., Masataki, H., Ishii, R., and Higashinaka, R. (2017). Online end-of-turn detection from speech based on stacked time-asynchronous sequential networks. In *Interspeech*, volume 2017, pages 1661–1665.

Masumura, R., Tanaka, T., Ando, A., Ishii, R., Higashinaka, R., and Aono, Y. (2018). Neural dialogue context online end-of-turn detection. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 224–228.

Razavi, S. Z., Kane, B., and Schubert, L. K. (2019). Investigating linguistic and semantic features for turn-taking prediction in open-domain human-computer conversation. In *INTERSPEECH*, pages 4140–4144.

Roddy, M., Skantze, G., and Harte, N. (2018). Multimodal continuous turn-taking prediction using multiscale rnns. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 186–190.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Sacks, H., Schegloff, E. A., and Jefferson, G. (1978). A simplest systematics for the organization of turn taking for conversation. In *Studies in the organization of conversational interaction*, pages 7–55. Elsevier.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Skantze, G. (2017). Towards a general, continuous model of turn-taking in spoken dialogue using lstm recurrent neural networks. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 220–230.

Skantze, G. (2021). Turn-taking in conversational systems and human-robot interaction: a review. *Computer Speech & Language*, 67:101178.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Ward, N. and Tsukahara, W. (2000). Prosodic features which cue back-channel responses in english and japanese. *Journal of pragmatics*, 32(8):1177–1207.