Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales

Clasificación de transiciones de turno en conversaciones humano-humano utilizando LLMs

Tesis de Licenciatura en Ciencias de Datos

Tomás Ravel

Director: Pablo Brusco

Buenos Aires, 2025

CLASIFICACIÓN DE TRANSICIONES DE TURNO EN CONVERSACIONES HUMANO-HUMANO UTILIZANDO LLMS

El análisis automático de eventos en conversaciones habladas entre humanos, o entre humanos y sistemas, es una tarea fundamental para el desarrollo de sistemas de diálogo más naturales y eficientes. En particular, **la clasificación de transiciones de turno** en conversaciones permite a los asistentes virtuales entender cuándo es un buen momento para interrumpir en una conversación, cuándo no, entender la intención del usuario, entre otros aspectos. Asimismo, la comunidad lingüística puede beneficiarse de sistemas que en pocos segundos crean reportes sobre estas interacciones que, tiempo atrás, habrían requerido horas de escucha y anotación manual.

En las últimas dos décadas, este problema se ha abordado mediante modelos de aprendizaje supervisado que utilizan una combinación de atributos acústico-prosódicos y léxicos. Esta tesis explora un paradigma alternativo: el uso de **Modelos de Lenguaje de Gran Escala (LLMs)** para la clasificación offline de transiciones de turno, sin necesidad de entrenamiento específico para esta tarea. El objetivo principal es evaluar la viabilidad de este enfoque con muy poca supervisión – cada vez más popular como paradigma de resolución de problemas – y comparar su rendimiento contra los métodos tradicionales, que sí requieren de una cantidad significativa de datos de entrenamiento.

Para ello, realizamos una serie de experimentos sobre el UBA Games Corpus, una colección de diálogos en español orientados a tareas. Se evaluaron los modelos LLaMA 3.3-70B y Gemini 2.5 Pro – que utilizan únicamente las transcripciones del diálogo como entrada, es decir, no utilizan información de la señal acústica – mediante técnicas de prompting, explorando sistemáticamente distintas estrategias de representación del problema, como la inclusión de ejemplos (few-shot), el formato de los datos y las distintas representaciones del conjunto de etiquetas. El rendimiento se midió utilizando la métrica Macro F1 y se comparó con un modelo de referencia basado en Redes Neuronales Recurrentes (RNNs) entrenado con atributos acústicos.

Los resultados demuestran que, si bien el enfoque basado en LLMs no logra superar el rendimiento global del baseline acústico (Macro F1 de 0.55 frente a 0.67), sí muestra una capacidad notable para comprender la tarea a pesar de no haber sido entrenado específicamente para ello. Es destacable que el mejor modelo (Gemini 2.5 Pro) superó al baseline en la clasificación de categorías semánticamente complejas, como las interrupciones. Se concluye que, aunque la información prosódica sigue siendo crucial, los LLMs son capaces de capturar eficazmente pistas léxicas y estructurales del texto. Este trabajo sienta las bases para futuras investigaciones, no solo en sistemas híbridos que combinen la potencia semántica de los LLMs con la sensibilidad de los sistemas clásicos, sino también en la exploración de arquitecturas multimodales capaces de procesar directamente la señal de audio.

Palabras clave: manejo de turnos, clasificación de transiciones, modelos de lenguaje grandes (LLMs), procesamiento del lenguaje natural, análisis de conversaciones, prompting.

AGRADECIMIENTOS

A mi director, Pablo.

Tuve la enorme suerte de contar con un director que no solo es un muy buen docente, sino que además trabaja desde hace años en los temas que aborda esta tesis. Más allá de eso, lo que realmente hizo la diferencia fue su voluntad. Desde el primer momento me abrió la puerta a aprender de su forma de encarar problemas, su rigor y su mirada estratégica. Me acompañó con gran dedicación, siempre dispuesto a juntarnos en su oficina para pensar juntos cualquier obstáculo y para discutir en detalle cada idea. No puedo imaginar un mejor contexto para hacer una tesis que este: con alguien que confía, que desafía, y que se involucra con entusiasmo genuino. Fue un verdadero placer y un honor.

También quiero agradecer a mi familia y a mis amigos, por el acompañamiento constante, la paciencia infinita y el apoyo en cada momento importante de la carrera. Estuvieron siempre, bancando las largas jornadas de trabajo, los tiempos de estrés, y también celebrando cada pequeño avance. Esta tesis también es un pedacito de todo eso.

Índice general

1. INTRODUCCIÓN

Cuando dos personas mantienen una conversación hablada, existe una dinámica implícita de manejo de turnos que permite que el intercambio de mensajes se produzca de forma fluida y ordenada. Esta dinámica se basa en múltiples señales que los hablantes producen —léxicas, prosódicas, acústicas e incluso gestuales— y que, combinadas, permiten inferir quién continuará hablando o cuándo ocurrirá un cambio de interlocutor.

A lo largo de las últimas décadas, el estudio de esta mecánica ha cobrado especial relevancia, no solo desde un punto de vista lingüístico o cognitivo, sino también por sus aplicaciones prácticas en sistemas de diálogo automático, análisis de conversaciones y evaluación del comportamiento en contextos humanos o híbridos.

Uno de los problemas más estudiados en esta área es la **clasificación automática de transiciones de turno**, es decir, la tarea de construir sistemas que permitan, a partir de una conversación entre dos hablantes, identificar y etiquetar eventos en los que un hablante inicia o intenta iniciar un turno, y cuál es su relación respecto del turno actual o anterior del interlocutor. Las categorías más frecuentes incluyen transiciones suaves con y sin solapamiento, interrupciones exitosas y fallidas, backchannels – pequeñas alocuciones que hacen entender al interlocutor que estamos prestando atención, tales como *ahá*, *mm-hh*, *sí* – entre otras.

El estudio de las transiciones de turno es clave para lograr interacciones más naturales entre humanos y sistemas automáticos. Hoy en día, asistentes como Siri, Alexa o Google Assistant dependen de una dinámica rígida de turnos bien separados: el usuario habla, el sistema espera, procesa y responde. Esta estructura limita la fluidez de la conversación y contrasta con la realidad del habla humana, donde son frecuentes las interrupciones, los solapamientos y señales breves como "ajá", "sí" o "entiendo".

Poder detectar y clasificar automáticamente estos fenómenos es fundamental para que los sistemas de diálogo avancen hacia una experiencia verdaderamente conversacional, donde logremos que los modelos no solo sepan manejar nuestras interrupciones, sino que también sean ellos los que nos interrumpan o solapen cuando corresponda.

A su vez, este problema se divide en dos vertientes similares, pero con características distintivas: (a) la tarea de **clasificación de conversaciones finalizadas (modo *offline*)** – en donde el sistema tiene acceso a la conversación completa y puede, por ejemplo, utilizar información futura a la transición que se esté analizando para tomar una decisión; y (b) la tarea de **clasificación en tiempo real (modo *online*)** en donde el sistema debe predecir qué transición de turno sucederá a continuación en una conversación que continúa sucediendo. En esta tesis, trabajamos sobre el problema offline. Sin embargo, solucionar el problema sobre conversaciones completas puede ser una herramienta útil para el caso online, en particular como paso previo para entrenar modelos online más precisos, al proporcionar etiquetas de alta calidad sin la necesidad de intervención humana.

Tradicionalmente, los enfoques de construcción predominantes se han basado en arquitecturas pensadas para el procesamiento de series temporales, tales como redes neuronales

recurrentes (RNNs), que utilizan como entrada atributos acústico-prosódicos (tono, intensidad, velocidad del habla, etc) y léxico-sintácticos (etiquetado gramatical, frecuencia de palabras, embeddings, etc) extraídos a partir de la señal acústica de las conversaciones junto a sus transcripciones. Si bien es cada vez mayor la evidencia en modelado de turnos que destaca la relevancia de las señales acústico-prosódicas —especialmente en contextos de predicción en tiempo real— [1], [2], también se ha mostrado que existen pistas valiosas en el contenido verbal (es decir, en lo que se dijo) y en propiedades estructurales del lenguaje, como la completitud sintáctica o el cierre pragmático de una intervención [3], [4]. En el caso de la clasificación offline, se cuenta con la ventaja de disponer del futuro inmediato posterior a una transición, lo que permite resolver ambigüedades que, incluso para un humano, serían difíciles de discernir. Esta disponibilidad de contexto futuro no reemplaza la riqueza de las señales prosódicas, pero puede mitigar parcialmente su ausencia. En este escenario, resulta posible explorar modelos que operan exclusivamente sobre texto, con el objetivo de evaluar en qué medida las pistas léxicas y estructurales permiten identificar con precisión eventos conversacionales.

En este trabajo exploramos un enfoque novedoso en el ámbito del manejo de turnos, en línea con el paradigma actual en procesamiento de lenguaje natural: el uso de Modelos de Lenguaje de gran escala (LLMs) como herramientas generales para resolver tareas específicas, sin necesidad de entrenamiento adicional. En lugar de entrenar modelos especializados desde cero —como se ha hecho tradicionalmente en esta tarea—, adoptamos LLMs preentrenados como clasificadores de transiciones de turno en un setting offline, a través de técnicas de prompting. Dado que se trata de un primer acercamiento exploratorio, y en línea con lo discutido anteriormente, decidimos excluir del análisis los atributos derivados de la señal de audio y trabajar exclusivamente con las transcripciones de las conversaciones. Así, nos enfocamos en estudiar si la información sintáctico-semántica contenida en el texto es suficiente para predecir estas transiciones con mayor calidad que las técnicas clásicas. A través de una serie de experimentos sobre el UBA Games Corpus [5], compuesto por conversaciones humanas en español orientadas a tareas específicas, exploramos distintas estrategias de representación del problema y de prompting, que incluyen variantes con y sin ejemplos, preprocesamiento parcial de etiquetas, simplificaciones de las clases y diferentes formatos de presentación de los datos. Evaluamos el rendimiento de los modelos en conjuntos de datos preestablecidos para desarrollo y para evaluación final (conjunto de evaluación), y comparamos los resultados obtenidos con los de modelos baselines basados en RNNs bidireccionales.

El objetivo general de este trabajo es responder a una pregunta central: ¿es viable utilizar modelos de lenguaje grandes como clasificadores de transiciones de turno offline sin información acústica? Y, en caso de serlo, ¿en qué condiciones funcionan mejor o peor estos modelos al compararlos contra otros enfoques tradicionales?

1.1. Trabajo previo

El estudio del manejo de turnos en conversaciones humanas ha sido abordado extensamente desde diversas disciplinas, incluyendo la lingüística, la fonética y el procesamiento del habla. Una de las investigaciones fundacionales en esta área es la de Duncan [1], quien observó que ciertas señales (como el contacto visual o los cambios de entonación) prece-

den sistemáticamente a las transiciones de turno. Además, propuso que estas señales no operan de forma aislada, sino que su efecto es acumulativo: mientras más señales coincidan temporalmente, mayor es la probabilidad de que se produzca un cambio de hablante. Desde entonces, se han explorado múltiples pistas que pueden indicar cuándo un hablante está por ceder la palabra, incluyendo aspectos acústico-prosódicos, léxicos, sintácticos y gestuales [6].

En el campo del procesamiento automático del habla, la mayoría de los sistemas de modelado del manejo de turnos han adoptado un enfoque online. Estos modelos intentan predecir en tiempo real cuándo un hablante va a terminar su turno, con aplicaciones directas en asistentes conversacionales como el sistema *Alexa* de Amazon o el sistema *Siri* de Apple. Estos sistemas se benefician enormemente de modelos capaces de anticipar el fin de un turno con suficiente antelación como para generar una respuesta fluida. Trabajos como los de Skantze [7] y Masumura [8] exploran redes neuronales recurrentes (LSTM, GRU) y atributos acústicos para esta tarea.

El procesamiento en modo offline ha sido menos explorado en la literatura. Un trabajo en esta línea es el realizado por Brusco [9], quien en su tesis doctoral estudió el manejo de turnos desde una perspectiva translingüística, utilizando datos de conversaciones en inglés, español y eslovaco. En ella, el autor exhibe conjuntos de atributos acústico-prosódicos consistentes entre idiomas, relevantes para la construcción de modelos de predicción de transiciones de turno. Además, propone un sistema de etiquetado offline de transiciones de turno que, a diferencia de los modelos online, aprovecha la conversación completa para categorizar los cambios de turno mediante una taxonomía completa que cubre las posibles interacciones entre turnos [10]. Pese a mostrar la factibilidad de construir estos sistemas, los autores muestran que ciertas categorías (por ejemplo algunos tipos de interrupciones poco frecuentes) sufren de baja performance debido, principalmente, a la poca cantidad de datos de entrenamiento para los modelos.

En relación a este trabajo, se han desarrollado modelos que incorporan además información léxica o sintáctica como parte de los atributos utilizados o exploran otras técnicas de modelado pertinentes para la tarea [11], [12]. Un ejemplo relevante es la tesis de Licenciatura de Scherman [13], quien investiga la inclusión de atributos léxicos y gramaticales en modelos basados en redes neuronales recurrentes mostrando que estos elementos pueden aportar mejoras significativas en la predicción de transiciones de turno cuando se combinan con atributos acústico-prosódicos. No obstante, su enfoque continúa dependiendo de arquitecturas clásicas del mundo del aprendizaje automático supervisado, las cuales requieren una gran cantidad de datos de entrenamiento para obtener buenos resultados. Además, estos datos son complejos de recolectar debido a la dificultad de la tarea y el elevado costo de obtención.

Más recientemente, la introducción de modelos de lenguaje grandes (LLMs), como GPT, ha abierto la posibilidad de abordar el problema con menos necesidad de datos de entrenamiento. El trabajo TurnGPT [4] presenta uno de los primeros intentos de utilizar un modelo basado en la arquitectura *Transformer* [14] para predecir transiciones de turno. En este caso, basándose exclusivamente en información lingüística. Este modelo, una versión modificada de GPT-2 (modelo preentrenado por OpenAI), fue entrenado en corpora de diálogo anotados con tokens especiales para marcar los cambios de hablante. A través de estudios de ablación y análisis de gradientes, los autores muestran que TurnGPT es

capaz de identificar la completitud sintáctica y pragmática de las intervenciones, y que el contexto previo es fundamental para realizar predicciones precisas. Una extensión reciente de este modelo incorpora también la respuesta potencial del siguiente hablante como señal para la predicción, lo que permite capturar mejor el carácter intencional de los cambios de turno [15].

Sin embargo, incluso TurnGPT está orientado a la predicción online de transiciones, donde el modelo únicamente accede al pasado del diálogo. En contraste, nuestro trabajo se centra en el etiquetado offline de transiciones, una tarea que permite utilizar la conversación completa para emitir una decisión informada sobre la naturaleza de cada cambio de turno. Esta diferencia es clave: en lugar de buscar intervenir en tiempo real, buscamos construir herramientas que analicen conversaciones a posteriori, posibilitando aplicaciones como el análisis masivo de interacciones, la creación de datasets anotados, o la supervisión de sistemas conversacionales. Además, la arquitectura de TurnGPT, a diferencia de las que vamos a utilizar nosotros, no permite dar instrucciones o ejemplos.

En otras áreas, es cada vez más común la utilización del paradigma *few-shot* learning, en donde se busca resolver tareas a partir de modelos preentrenados, mediante la técnica de prompting y la utilización de cero o pocos ejemplos para que los modelos puedan resolver las tareas. Por ejemplo, en Pinto [16] exploran el uso de LLMs modernos como GPT-4 y Gemini para anticipar puntos de finalización de turno en interacciones humano-robot, utilizando un enfoque basado en prompting que analiza únicamente información textual. Su estudio, centrado en conversaciones con adultos mayores, demuestra que los LLMs pueden superar enfoques tradicionales basados en umbrales de silencio, especialmente cuando se combinan con técnicas de Voice Activity Detection (VAD), ofreciendo predicciones más naturales y sensibles al contexto conversacional. También, en Allen [17] utilizan prompting con LLMs para detectar alucinaciones sin entrenamiento supervisado, y en Hassani [18] muestran que LLMs fine-tuneados o con pocos ejemplos superan ampliamente a baselines clásicos en la clasificación de disposiciones legales sobre seguridad alimentaria. Todos estos trabajos se inscriben en un nuevo paradigma de resolución de tareas con LLMs, en donde ya no se requiere entrenamiento supervisado tradicional, sino que se aprovecha la capacidad generalista de estos modelos a través de técnicas de prompting, construcción de ejemplos, y en algunos casos, ajuste fino de los modelos. Sin embargo, hasta donde sabemos, no existen aún trabajos previos que exploren el uso exclusivo de LLMs para la clasificación offline de transiciones de turno en conversaciones humano-humano.

Esta tesis busca evaluar el potencial de modelos puramente semánticos para identificar patrones conversacionales complejos, sin apoyarse en atributos acústicos ni entrenamientos supervisados clásicos. Además, se plantea la posibilidad de utilizar los resultados de estos modelos como insumo para entrenar futuros modelos online, actuando como un primer paso en un pipeline de anotación automática.

1.2. Estructura de la tesis

Este trabajo está organizada en cinco capítulos principales, incluyendo este capítulo introductorio y cuatro más que se describen a continuación.

En el Capítulo 2 se presentan los fundamentos teóricos y técnicos necesarios para comprender el trabajo. Se explica en detalle qué son las transiciones de turno, cómo se han

clasificado tradicionalmente, y se ofrece una introducción general a los Modelos de Lenguaje Grandes (LLMs), destacando sus características, formas de uso y ventajas frente a otros enfoques. También se discute cómo se representó el problema en nuestra tarea específica.

El Capítulo 3 describe la metodología adoptada. Se detallan las características del corpus utilizado, la división en conjuntos de entrenamiento, desarrollo y evaluación. También se explica el diseño de la grilla de hiperparámetros para evaluar distintas variantes de prompts y configuraciones, así como la implementación técnica de los experimentos y las métricas seleccionadas para la evaluación.

En el Capítulo 4 se presentan los resultados obtenidos. Primero se discuten los rendimientos de los modelos en el conjunto de desarrollo y luego se analiza el desempeño en el conjunto de evaluación, que incluye tareas no vistas durante el entrenamiento. Se comparan distintas configuraciones y se analiza el impacto de cada elemento del prompt sobre la calidad de las predicciones.

Finalmente, el Capítulo 5 recoge las conclusiones del trabajo y propone direcciones para trabajos futuros. Se reflexiona sobre los aportes de utilizar LLMs en esta tarea, se reconocen las limitaciones del enfoque y se plantean posibles mejoras, incluyendo la integración con información acústica.

2. FUNDAMENTOS TEÓRICOS Y TÉCNICOS

2.1. Clasificación de transiciones de turno

En el análisis de conversaciones, es crucial segmentar el habla en unidades significativas que permitan estudiar su estructura y dinámica. Una forma común de segmentación es mediante las llamadas *unidades interpausales*. Definimos como *unidad interpausal* (IPU, por sus siglas en inglés) a un segmento continuo de habla contenido entre pausas de al menos 200 milisegundos. Estos segmentos 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 analizar su dinámica. En la Figura 2.1, se visualiza un ejemplo sintético, inspirado en el juego de objetos que explicamos más adelante. En esta figura, cada rectángulo representa una IPU junto a su transcripción.

Un *turno* es un intervalo de tiempo en el cual un hablante tiene participación activa en la conversación. Formalmente, se define como una secuencia maximal de IPUs en las que no hay ninguna actividad del interlocutor durante los silencios entre IPUs. En la Figura 2.1 vemos 4 turnos (ilustrados con las barras continuas), un primer turno con dos IPUs, y otros tres turnos de una sola IPU.

Una etiqueta de *transición de cambio de turno* describe la relación entre un turno y el turno inmediatamente anterior del interlocutor. Identificar correctamente estas transiciones es clave para que sistemas de diálogo hablado sean más naturales y eficientes y para el estudio de conversaciones desde la perspectiva lingüística. La Figura 2.1 muestra las etiquetas X1, BI, S y O como indicadores de transición. Notar que para el caso del primer BI, el turno inmediato anterior no finalizó, en cambio para el caso del primer S, el turno inmediato anterior sí finalizó.

Fig. 2.1: Ejemplo inventado para ilustrar unidades interpausales (IPUs), agrupación en turnos y posibles etiquetas de transición como `X1`, `BI`, `O` y `S`. Cada rectángulo con texto (verdes y violetas) representa una IPU y las líneas negras indican los turnos. La etiqueta BI se da cuando un hablante hace un intento no exitoso de tomar el turno interrumpiendo al otro hablante. En este ejemplo claramente se ve la intención de interrumpir en el "pero el", pero esta no logra concretarse, ya que el hablante original continúa su discurso.

En esta tesis seguimos una taxonomía propuesta por Beattie [10], que clasifica las transi-

ciones en dos grandes grupos: **con solapamiento** y **sin solapamiento**. Cabe aclarar que el solapamiento refiere al momento en que comienza el turno del interlocutor, no a lo que suceda después. Es decir, puede haber turnos que en su totalidad se encuentran solapados (como por ejemplo el primer turno de B en la figura) o turnos que en algún punto se encuentran solapados, como el primer o segundo turno de A.

Transiciones sin solapamiento

- **Switch (S):** el interlocutor toma la palabra luego de que el hablante complete su mensaje y haya un silencio.
- **Interrupción en pausa (PI):** el interlocutor habla tras una pausa, pero el hablante anterior no había terminado su mensaje.
- **Backchannel (BC):** breve señal del oyente tras una pausa (por ejemplo, "ajá", "claro") para mostrar atención sin intención de tomar el turno.

Transiciones con solapamiento

- **Overlap (O):** equivalente a S, pero el nuevo hablante comienza antes de que el turno anterior finalice.
- **Backchannel con solapamiento (BC_O):** versión solapada del BC; breve intervención sin intención de tomar el turno que comienza antes de que el interlocutor haya terminado su turno.
- **Interrupción con solapamiento (I):** el nuevo hablante interrumpe al actual, sin esperar a que termine su mensaje mientras el interlocutor seguía produciendo sonido.
- **Butting-in (BI):** intento de interrupción en la que el hablante no logra tomar control de la conversación y el interlocutor continúa con su turno actual.

Categorías especiales

- **X1:** inicio del primer turno de la conversación.
- **X2 / X2_O:** regreso (con o sin solapamiento, respectivamente) al hablante original tras un backchannel.
- **X3:** se asigna a la segunda IPU cuando dos comienzan casi simultáneamente (menos de 210 ms de diferencia). Esta etiqueta se utiliza porque los humanos típicamente no reaccionan ante tiempos muy cortos, y por lo tanto, la segunda IPU generalmente se interrumpe rápidamente al detectar que la otra persona quería comenzar a hablar o continuar con su turno.

Categorías intra-turno

- **Hold (H):** el mismo hablante continúa hablando tras una pausa breve. Esta categoría se define porque muchos de los sistemas, en particular los que hacen predicción en tiempo real, tienen que determinar si el turno se completó o continúa). En el etiquetado offline, esta es una tarea sencilla, programática, que se puede realizar al momento de la creación de los turnos.

2.2. Modelos de Lenguaje Grandes (LLMs)

Los *Modelos de Lenguaje Grandes* (LLMs, por sus siglas en inglés) son sistemas generalmente compuestos por redes neuronales entrenadas sobre enormes cantidades de texto con el objetivo de modelar la probabilidad de aparición de secuencias lingüísticas. Estos modelos, usualmente basados en arquitecturas tipo *transformer*, han demostrado capacidades sobresalientes en tareas como generación de texto, traducción automática, respuesta a preguntas, clasificación de texto, y razonamiento lógico [17], [18].

Los LLMs se entrenan en grandes corpus de texto sin anotaciones explícitas, mediante tareas de modelado del lenguaje. Por ejemplo, se les presenta un fragmento de texto y deben predecir la siguiente palabra (*auto-regresivos*, como GPT[19]) o completar palabras enmascaradas dentro del texto (*auto-codificadores*, como BERT[20]). Este entrenamiento masivo les permite adquirir un conocimiento implícito sobre el lenguaje, las entidades del mundo real y patrones conversacionales.

Una vez entrenados, los LLMs suelen pasar por etapas de postprocesamiento que los preparan para su uso práctico. Estas etapas incluyen técnicas como el aprendizaje con instrucciones (*instruction tuning*), que les enseña a seguir indicaciones expresadas en lenguaje natural, así como el ajuste para mantener coherencia en diálogos largos mediante ventanas de contexto extendidas. También pueden aplicarse filtros o ajustes para mejorar la seguridad, reducir sesgos o evitar respuestas inapropiadas. Luego de este postprocesamiento, los modelos pueden utilizarse para resolver tareas sin necesidad de un ajuste fino supervisado. A través de técnicas como el *prompting*, es posible inducir en el modelo un comportamiento deseado simplemente proporcionando ejemplos o instrucciones en lenguaje natural. Esto ha permitido usar los LLMs para tareas de clasificación, extracción de información, razonamiento paso a paso o análisis semántico, sin requerir arquitectura adicional.

El prompting se volvió una disciplina en sí misma, donde una de las principales distinciones es entre el enfoque zero-shot y el few-shot. En el *enfoque zero-shot*, se presenta al modelo únicamente una consigna en lenguaje natural (por ejemplo, “Clasificá esta frase como positiva o negativa”), confiando en su conocimiento general para resolver la tarea. En el *few-shot prompting*, se le muestran algunos ejemplos anotados dentro del prompt antes de la consigna, lo que permite guiar al modelo hacia un comportamiento más preciso. Estos enfoques explotan la habilidad del modelo para generalizar a nuevas tareas sin entrenamiento específico.

Las principales ventajas de los LLMs incluyen su flexibilidad, su capacidad de adaptación rápida a nuevas tareas mediante prompts, y su conocimiento enciclopédico del lenguaje y del mundo. Además, permiten reducir significativamente el costo de anotación de datos y acelerar la experimentación.

Sin embargo, presentan también limitaciones. Su desempeño puede ser sensible a la redacción del prompt, y su razonamiento puede ser inconsistente o superficial en tareas complejas. Además, requieren grandes recursos computacionales para su entrenamiento y ejecución, y su comportamiento puede ser poco interpretable. En contextos sensibles, también es importante considerar sus sesgos inherentes y el riesgo de generar contenido inexacto o inapropiado si no se usan con cautela.

En este trabajo utilizaremos dos modelos relativamente actuales: LLaMA 3.3-70B [21] y Gemini 2.5 Pro [22]. LLaMA [23], abreviación de Large Language Model Meta AI, es una familia de modelos desarrollada por Meta, entrenados con un enfoque auto-regresivo sobre grandes corpus de texto en múltiples idiomas. Su entrenamiento se basa en predecir la siguiente palabra en secuencias de texto, utilizando únicamente texto plano extraído de fuentes como libros, artículos científicos, y páginas web filtradas. Los modelos LLaMA fueron optimizados para ser eficientes en recursos y facilitar su uso en investigaciones académicas y desarrollos personalizados, y se destacan por su buena relación entre tamaño y rendimiento. Existen 3 tamaños diferentes para LLaMA 3.3: 8B, 70B y 405 B. Nosotros, como su nombre indica, estamos usando la versión de 70B.

Gemini, por su parte, es una línea de modelos desarrollada por Google DeepMind, que combina entrenamiento en lenguaje con razonamiento multimodal. En su versión 2.5, el modelo fue entrenado no solo con texto, sino también con imágenes, audio y código, utilizando técnicas avanzadas como el aprendizaje por refuerzo con retroalimentación humana (RLHF) [22] para alinear sus respuestas con los objetivos del usuario. Esto le permite responder de forma más contextual y razonada, manteniendo un foco en utilidad y precisión. Por ser un modelo completamente cerrado, no se conoce el dato oficial sobre la cantidad de parámetros, aunque por la efectividad que está demostrando y la magnitud de su empresa desarrolladora, es razonable esperar a que se encuentre en el rango de los modelos más extensos de la actualidad. Al comparar ambos modelos, podremos evaluar diferencias entre enfoques abiertos y cerrados, auto-regresivos puros versus entrenamiento multimodal, y su impacto en tareas de análisis conversacional.

2.3. Representación del problema

Uno de los primeros desafíos al trabajar con LLMs para analizar conversaciones es definir un formato de entrada adecuado. Las conversaciones humanas por naturaleza son intercaladas, espontáneas y muchas veces desordenadas: un hablante puede comenzar a hablar después que otro pero finalizar antes, generando solapamientos difíciles de capturar linealmente. Cualquier representación que utilicemos para construir el prompt implica imponer un orden explícito, simplemente por el hecho de que el texto será leído por el modelo de forma secuencial, de izquierda a derecha y de arriba hacia abajo. Como se ve en la Figura 2.2, se probaron distintas formas de representar el diálogo para que el modelo pueda procesarlo como texto:

- **Representación separada:** las IPUs de cada hablante se agrupan por separado. Esto permite visualizar con claridad la alternancia entre interlocutores, aunque pierde parte del flujo temporal real.
- **Representación intercalada:** las IPUs y turnos se presentan en el orden en que

ocurrieron, indicando explícitamente qué hablante emitió cada una. Esta variante busca preservar la dinámica real de la conversación.

- **Frases vs. turnos:** en algunas representaciones se muestran IPUs individuales; en otras, se agrupan en bloques correspondientes a turnos completos. Esto permite evaluar si el modelo se beneficia de una mayor cantidad de contexto por hablante.

- **Uso de tablas:** se exploraron formatos tabulares simples, donde cada fila contiene una IPU junto con su hablante, momento relativo y (en modo few-shot) su etiqueta correspondiente. Esta estructura busca clarificar el objetivo de clasificación.

a) Formato separado con frases y turnos

Frases

Sujeto A
0.00 1.37 `bueno`
1.45 3.60 `el barquito esta arriba de la mesa`
6.30 7.10 `claro`

Sujeto B
2.10 2.60 `pero el`
4.50 6.00 `abajo de la lampara?`

Turnos

Sujeto A
0.00 3.60 `X1`
6.30 7.10 `S`

Sujeto B
1.45 3.60 `BI`
4.50 6.00 `S`

b) Formato intercalado con frases y turnos

Frases
A 0.00 1.37 `bueno`
A 1.45 3.60 `el barquito esta arriba de la mesa`
B 2.10 2.60 `pero el`
B 4.50 6.00 `abajo de la lampara?`
A 6.30 7.10 `claro`

Turnos
A 0.00 3.60 `X1`
B 1.45 3.60 `BI`
B 4.50 6.00 `S`
A 6.30 7.10 `S`

c) Formato intercalado solo con turnos

Turnos
A 0.00 3.60 `bueno el barquito esta arriba de la mesa X1`
B 1.45 3.60 `pero el BI`
B 4.50 6.00 `abajo de la lampara? S`
A 6.30 7.10 `claro S`

Fig. 2.2: Ejemplos comparativos de representación del diálogo. a) Formato separado con frases y turnos. b) Formato intercalado con frases y turnos. c) Formato intercalado solo con turnos. Para todos los formatos intercalados se tomó la decisión de ordenar por tiempo de inicio

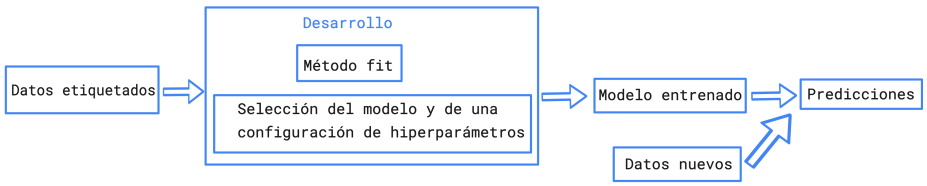
Si hablamos sobre el objetivo de clasificación, la tarea se plantea como una predicción supervisada: El punto de partida son las ya mencionadas unidades interpausales (IPUs), que representan segmentos continuos de habla delimitados por pausas de al menos 200 milisegundos. Estas IPUs se agrupan en turnos conversacionales, definidos como secuencias máximas de IPUs de un mismo hablante sin intervención del otro. La tarea de clasificación consiste en predecir el tipo de transición que ocurre entre dos turnos consecutivos, usando como contexto parte del diálogo circundante. Este objetivo se integró dentro del prompt mediante formatos estructurados, en lenguaje natural o con marcas especiales que indican la posición de la transición a predecir.

En esta sección no se detallan aún las variantes exactas del prompt ni las combinaciones evaluadas, ya que esto será abordado en el capítulo tres al describir el diseño experimental.

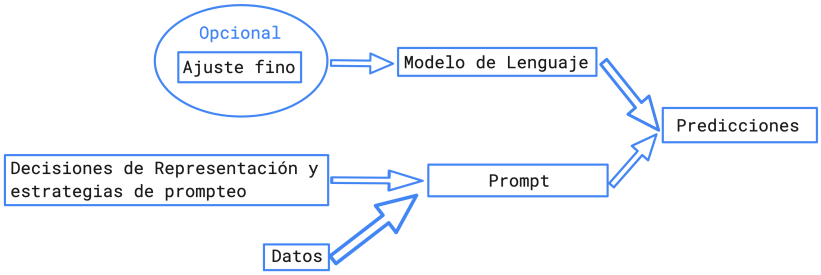
2.4. Comparación entre enfoques clásicos y basados en LLMs

En este trabajo adoptamos un enfoque diferente al del aprendizaje automático clásico. Mientras que el paradigma tradicional suele involucrar etapas bien definidas como la extracción de atributos, la selección de un modelo base, el ajuste de hiperparámetros y el entrenamiento supervisado, aquí partimos de un modelo de lenguaje grande (LLM) ya preentrenado, y nos enfocamos en cómo estructurar el problema de forma adecuada para que dicho modelo lo resuelva directamente a través de una respuesta generada.

La Figura 2.3 muestra una comparación conceptual entre ambos esquemas. Arriba se representa el pipeline clásico. En este caso, el modelo, una vez ajustado, puede predecir etiquetas para nuevas instancias. Abajo, en cambio, se ilustra el enfoque basado en LLMs, donde el eje central no es el entrenamiento sino el diseño del prompt: cómo se representa el problema, con qué formato se le presentan los datos al modelo, si se incluyen ejemplos (few-shot) o no (zero-shot), y qué tipo de respuesta se espera obtener.



a) Pipeline clásico de aprendizaje supervisado



b) Pipeline basado en LLMs para clasificación

Fig. 2.3: Comparación conceptual entre el pipeline clásico y el enfoque con LLMs.

Una de las principales diferencias es que en el enfoque tradicional el conocimiento sobre la tarea se aprende durante el entrenamiento, mientras que en los LLMs gran parte de ese conocimiento está latente desde el preentrenamiento y se activa al presentar el problema en

un formato comprensible. Por otro lado, el pipeline clásico requiere necesariamente datos etiquetados para entrenar el modelo, mientras que los LLMs no nos obligan a realizar ningún tipo de entrenamiento, lo que nos permite comenzar a experimentar directamente, con una inversión mucho menor en anotación.

También cambia radicalmente la representación del problema: en el enfoque clásico se parte de variables estructuradas o vectores de atributos, mientras que en los LLMs todo debe expresarse como texto o pseudotexto. Esto genera nuevos desafíos de diseño pero también habilita una gran flexibilidad. Finalmente, cabe destacar que si bien los modelos clásicos permiten un control más granular y explícito del proceso de entrenamiento, los LLMs ofrecen una solución más directa y general, capaz de adaptarse a una amplia variedad de tareas sin necesidad de cambios estructurales en el modelo.

3. METODOLOGÍA

En esta sección detallamos el diseño experimental y las decisiones metodológicas tomadas a lo largo del proyecto.

3.1. Dataset

Para el presente trabajo utilizamos exclusivamente el **UBA Games Corpus** [5], una colección de diálogos y monólogos espontáneos producidos por hablantes nativos de Español Argentino. Tiene 706 minutos de diálogos en los cuales se resuelven tareas colaborativas (juegos de posicionamiento de objetos), y además tiene 119 minutos de monólogos que consisten en instrucciones para moverse en la ciudad, con versiones espontáneas y leídas. El corpus incluye audios, transcripciones ortográficas, anotaciones de transiciones de turnos, entre otras anotaciones. La Figura 3.1 muestra un ejemplo de cómo la señal acústica de cada hablante fue anotada por transcriptores profesionales.

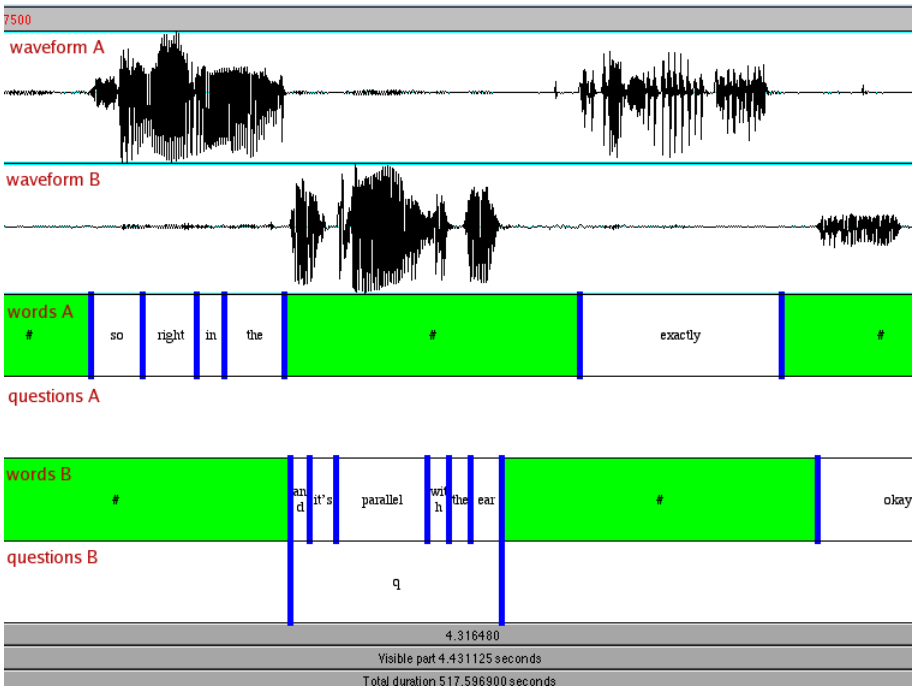
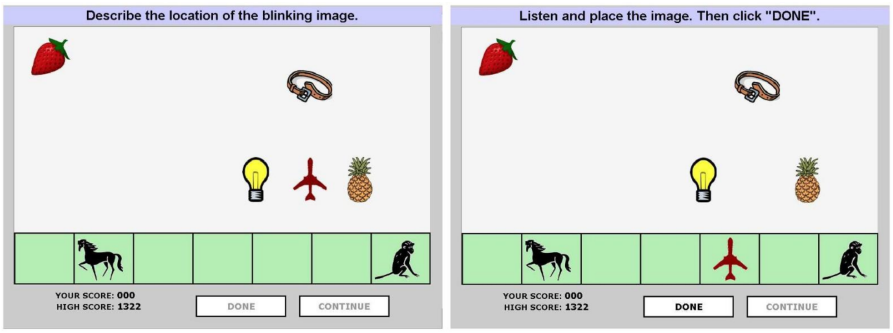


Fig. 3.1: Tipo de anotaciones presentes en el Games Corpus (imagen tomada del *Columbia Games Corpus*, la versión estadounidense del corpus). En la imagen se puede observar las señales acústicas de los dos hablantes, las palabras alineadas a nivel temporal y otras anotaciones presentes en el dataset.

En esta tesis trabajamos únicamente con el primer lote de grabaciones del corpus (Batch 1), que consta de 14 sesiones. Estas sesiones se segmentan en 196 tareas individuales (14 tareas por sesión), de las cuales 132 se utilizaron para entrenamiento y desarrollo, y 64 se

reservaron como conjunto de evaluación. La separación exacta por tareas fue definida manualmente, asegurando que las tareas de evaluación no hubieran sido utilizadas durante la etapa de exploración y ajuste de modelos. Cabe destacar que la separación es exactamente la misma utilizada en Brusco [9], lo que nos permite una comparación directa de resultados con el "Benchmark" de los trabajos previos.

Cada sesión del corpus incluye 14 tareas del *Juego de Objetos* en las cuales los participantes alternan entre los roles de descriptor y seguidor. En cada tarea, uno de ellos debe indicar verbalmente la posición de un objeto faltante en su pantalla, para que el otro lo ubique correctamente en su propia vista. Las sesiones se grabaron en cabinas insonorizadas, con los participantes separados por una cortina opaca, garantizando que toda la comunicación fuese exclusivamente oral. Las grabaciones fueron transcritas y alineadas temporalmente a nivel de palabra e IPU (unidad interpausal), y se etiquetaron manualmente las transiciones de turno. La Figura 3.2 muestra un ejemplo de qué veían los participantes al momento de participar del experimento.



(Describer) (Follower)

Fig. 3.2: Juego de objetos. En la imagen izquierda se observa lo que ve el descriptor de la tarea (en este caso, ubicar un avión rojo en medio de la lámpara y el ananá. En la imagen derecha, lo que observa el que sigue las instrucciones, quien deberá posicionar el avión en la posición correcta.

Para interactuar con el corpus, utilizamos una herramienta desarrollada por los creadores del corpus [5], una biblioteca en `Python` llamada `games_corpus`. Esta biblioteca permite cargar el corpus completo, acceder a sesiones, tareas, turnos, unidades interpausales y etiquetas de transiciones de forma estructurada, y realizar análisis tanto textuales como acústicos. Entre sus funcionalidades se encuentran la visualización de espectrogramas, la extracción de atributos como los coeficientes cepstrales en las frecuencias de Mel (MFCCs), y el cálculo de estadísticas básicas.

Durante el preprocesamiento se descartaron algunas tareas con errores de grabación o metadatos incompletos. También se filtraron IPUs vacías. Cada tarea se representó como una secuencia de frases anotadas, agrupadas por hablante, junto con la secuencia completa de etiquetas de transición entre turnos. Estas estructuras sirvieron como entrada para los modelos evaluados.

La combinación de estructura conversacional rica, transcripciones alineadas, y anotaciones manuales convierte al UBA Games Corpus en una fuente ideal para el estudio del manejo de turnos y el entrenamiento de modelos semánticos de clasificación, más allá de la discusión

abierta sobre qué tan representativo es el contexto de este juego con respecto a cualquier otro tipo de conversación cotidiana. Esto último es relevante a la hora de poner en práctica los modelos para otros casos de uso.

3.2. Definición de grilla de experimentos

La etapa inicial de diseño de los prompts fue completamente exploratoria. Realizamos múltiples pruebas manuales directamente en la interfaz web de distintos modelos (como ChatGPT[1] y DeepSeek[2]), ajustando el formato de entrada, el número de ejemplos provistos, el orden de los elementos, entre otros aspectos. Si bien este proceso permitió adquirir intuiciones valiosas, resultaba evidente que se estaban tomando muchas decisiones de manera subjetiva y sin control sistemático sobre los factores en juego.

A partir de ese diagnóstico, decidimos formalizar el espacio de decisiones mediante una grilla de hiperparámetros que nos permitiera evaluar variantes de manera estructurada. La Tabla 3.1 muestra la grilla construida, compuesta por seis dimensiones que describimos a continuación.

Hiperparámetro	Valores posibles
#Ejemplos	`plain`, `onetask`, `perlabel`
Prefill	`prefill`, `noprefill`
Overlap	`full`, `boolean`, `simplified`
Estructura A-B	`separated`, `interleaved`
Frases	`before`, `inline`
Transition table	`transition`, `notransition`

Tab. 3.1: Grilla de hiperparámetros explorada. Cada combinación de estos valores define una configuración distinta.

El producto cartesiano de estas dimensiones genera un total de $3 \times 2 \times 3 \times 2 \times 2 \times 2 = 144$ combinaciones posibles. Cada configuración implica un system message (el mensaje inicial que define el rol o comportamiento del modelo) y un prompt message (el mensaje que contiene la instrucción concreta y el contenido de entrada) a medida. Debido a limitaciones de tiempo y recursos, decidimos evaluar únicamente ocho configuraciones representativas que consideramos prometedoras, seleccionadas manualmente para cubrir la mayor variedad posible de opciones sin superponer variantes triviales.

Esta grilla no solo permitió formalizar las decisiones que antes eran ad-hoc, sino que también facilitó el análisis comparativo posterior entre prompts. A largo plazo, el diseño modular de esta estructura permite incorporar nuevos factores (por ejemplo, longitud máxima, temperatura, etc.) sin modificar la lógica de evaluación.

[1] `chatgpt.com`, versión Marzo 2025. Los modelos probados fueron GPT-4o y GPT-o1
[2] `deepseek.com` versión Marzo 2025. Los modelos probados fueron DeepSeek-V3 y DeepSeek-R1

Hiperparámetros

- Número de **ejemplos** en la instrucción: `plain`, `onetask` o `perlabel`. El primero no incluye ningún ejemplo; el segundo incluye un ejemplo de clasificación de una tarea completa; el tercero presenta un ejemplo por clase (o sea un ejemplo por cada tipo de transición), lo cual ayuda a cubrir el espectro de etiquetas pero puede inducir sesgos en la distribución predicha, ya que le estaríamos mostrando tanto un ejemplo de la categoría más frecuente como un ejemplo de la menos frecuente. Pueden observarse ejemplos en la Figura 3.3.

- **Prefill**: como se observa en la Figura 3.4, este hiperparámetro indica si las clases fácilmente determinables de forma programática (X1, X3 y A) fueron pre-completadas en el prompt. Un argumento a favor de hacer esto es que las etiquetas X1 y X3 dependen exclusivamente de las marcas de tiempo y se pueden agregar de manera determinística a la perfección. Por otro lado, un pensamiento que empuja a seguir probando sin esta opción puede ser que el prompt se alargaría al explicarle que algunas etiquetas ya van a aparecer asignadas y que no tiene que predecir ninguna de esas 3 clases en ningún caso. Los prompts ya son considerablemente largos, y alargandolos uno podría pensar que la atención del modelo se diluye entre todas nuestras indicaciones, algo que no querríamos que ocurra. Y además, siendo algo menos concreto, podríamos querer que el modelo se enfrente con el desafío de predecir la conversación completa, por su carácter secuencial y la fuerte relación entre una etiqueta y la siguiente.

- **Overlap**: tres variantes. En modo `full`, se presentan todas las clases tal como están en el corpus. En modo `boolean`, se agrega un atributo explícito indicando si hay solapamiento entre hablantes para ayudar al modelo. En `simplified`, las clases que difieren solo por solapamiento (como S vs. O, o BC vs. BC_O) se unifican en una única categoría, lo cual podría favorecer el enfoque semántico reduciendo la carga cognitiva del modelo. Luego la división entre overlap o no overlap se podría postprocesar programáticamente. Podemos visualizar la diferencia entre full y boolean en la Figura 4.2.

- **Estructura A-B**: `separated` o `interleaved`. En el primer caso, las frases y turnos de cada hablante se presentan por separado; en el segundo, se intercalan cronológicamente siguiendo el orden real de la conversación. Esta última opción busca replicar el flujo natural del diálogo y facilitar el seguimiento contextual por parte del modelo. En la Figura 3.6 podemos notar la diferencia entre ambas.

- **Frases**: `before` o `inline`. En el modo `before`, las IPUs se listan por separado, y los turnos aparecen luego como referencias a bloques de IPUs. En el modo `inline`, cada turno ya contiene el texto completo (concatenado) de las IPUs que lo componen, lo cual simplifica el formato pero podría hacer menos transparente la segmentación interna. La alternativa inline es más compacta, y esto lo podemos visualizar en el ejemplo de la Figura 3.7.

- **Transition table**: presencia o ausencia de una tabla explícita indicando los turnos que deben ser clasificados. Esta tabla (que podemos observar en la Figura 3.8) actúa como guía para ordenar la salida y evitar ambigüedades estructurales que surgen, por

ejemplo, cuando un mismo hablante tiene dos turnos seguidos o cuando hay múltiples posibles transiciones simultáneas.

Ejemplos de variantes del hiperparámetro **ejemplos**

Ejemplo:

Turnos

A1 0.41 2.18 bueno está el mimo

B1 2.43 2.85 sí

A2 3.33 10.35 está titilando arriba de la lechuza justo en ángulo de noventa con la lechuza y la mm oreja

B2 10.56 11.84 ah intersección total

A3 11.88 12.22 claro

B3 13.40 18.85 okay sea digamos que la línea de abajo de todo del mimo coincide con la parte de abajo de la oreja

A4 20.06 20.47 cómo

B4 20.21 22.85 están a la misma altura el final de la oreja y el final del mimo

salida esperada:

0 → *A*1 X1

*A*1 → *B*1 BC

*B*1 → *A*2 X2

*A*2 → *B*2 S

*B*2 → *A*3 S

*A*3 → *B*3 S

*B*3 → *A*4 S

*A*4 → *B*4 X3

*B*4 → *A*5 S

*A*5 → *B*5 S

`(onetask)`

Ejemplo 1 (no es un task completo):

turnos

A1 0.41 2.18 bueno está el mimo

B1 2.43 2.85 sí

A2 3.33 10.35 está titilando arriba de la lechuza justo en ángulo de noventa con la lechuza y la mm oreja

Salida esperada

0 → *A*1 X1

*A*1 → *B*1 BC

*B*1 → *A*2 X2

Ejemplo 2 (no es un task completo):

turnos

B1 1158.65 1162.05 si si si el cuadrado que esta en el medio

A1 1259.47 1160.090 el final

A2 1160.41 1161.01 arriba de

A3 1161.49 1163.20 exacto esa esta en diagonal

B2 1163.32 1163.63 sí

A4 1164.37 1165.40 al costado del circulo verde

B3 1164.42 1164.75 y ahí

B4 1165.44 1169.89 queda el hombro del pirata quedan justo eh

A5 1165.49 1165.94 perfecto

A6 1170.14 1171.92 de ese lugar salen las tiritas del

B5 1171.19 1171.56 sí

respuesta esperada

0 → *B*1 S

*B*1 → *A*1 A

*B*1 → *A*2 BI

*B*1 → *A*3 I

*A*3 → *B*2 S

*B*2 → *A*4 S

*A*4 → *B*3 X3

*A*4 → *B*4 S

*B*4 → *A*5 X3

*B*4 → *A*6 PI

*A*6 → *B*5 O

Ejemplo 3 (no es un task completo):

B1 282.37 289.64 deberia ir justo abajo de donde pusiste el avion

A1 289.62 290.01 si

B2 289.94 291.75 pero asegurate de que no caiga en el medio

salida esperada

0 → *B*1 S *B*1 → *A*1 BC_O *A*1 → *B*2 X2_O

`(perlabel)`

Fig. 3.3: Variantes del hiperparámetro **ejemplos**. No se incluye la variante plain, ya que esta es justamente sin darle ejemplos al modelo.

Ejemplos de variantes del hiperparámetro **prefill**

ahora hace la clasificacion para estos turnos:

A1 82.256313 85.165231 mm no me titila ahí está la oreja

B1 85.71 86.02 sí

A2 86.33 91.43 y está en eh arriba del mimo y a la izquierda del león ahí justito pegadita

B2 91.92 92.13 cómo

transiciones:

$0 \rightarrow A1$

$A1 \rightarrow B1$

$B1 \rightarrow A2$

$A2 \rightarrow B2$

(a) `noprefill`

ahora hace la clasificacion para estos turnos:

A1 82.26 85.17 mm no me titila ahí está la oreja

B1 85.71 86.02 sí

A2 86.33 91.43 y está en eh arriba del mimo y a la izquierda del león ahí justito pegadita

B2 91.92 92.13 cómo

transiciones:

$0 \rightarrow A1$ X1

$A1 \rightarrow B1$

$B1 \rightarrow A2$

$A2 \rightarrow B2$

(b) `prefill`

Fig. 3.4: Variantes del hiperparámetro **prefill**.

Ejemplos de variantes del hiperparámetro **overlap**

Ahora hace la clasificación para estos turnos:

A 22.98 24.61 sí exacto está al mismo nivel

B 25.04 25.74 genial

A 25.60 26.63 y encima de la lechuza

B 27.26 32.59 y ocupa esta más a la derecha o a la izquierda o sea coincide con los bordes de la rama de la lechuza

A 33.48 33.75 sí

B 33.47 41.65 o está okay yo para que quede bien centradito esto está bien esto está bien debería estar bien

Agrega tu prediccion en la siguiente tabla

A 22.98 24.61

B 25.03 25.74

A 25.60 26.63

B 27.26 32.59

A 33.48 33.75

B 33.47 41.65

(a) `full`

Ahora hace la clasificación para estos turnos:

A 22.98 24.61 sí exacto está al mismo nivel

B 25.04 25.74 genial

A 25.60 26.63 y encima de la lechuza

B 27.26 32.59 y ocupa esta más a la derecha o a la izquierda o sea coincide con los bordes de la rama de la lechuza

A 33.48 33.75 sí

B 33.47 41.65 o está okay yo para que quede bien centradito esto está bien esto está bien debería estar bien

Agrega tu prediccion en la siguiente tabla

A 22.98 24.61

B 25.04 25.74

A 25.60 26.63 overlap

B 27.26 32.59

A 33.48 33.75

B 33.47 41.65

(b) `boolean`

Fig. 3.5: Variantes del hiperparámetro **overlap**. en la imágen b) se puede notar cómo ayudamos al modelo a saber donde hubo overlap. Esta idea busca no depender de la capacidad del modelo para tratar con números. La variante simplified no se incluye, ya que consta únicamente de "eliminar" del problema a las etiquetas con overlap hasta finalizar la predicción.

Ejemplos de variantes del hiperparámetro **estructura A-B**

ahora hace la clasificacion para estos turnos:

Sujeto A

A1 82.26 85.17 mm no me titila ahí está la oreja

A2 86.33 91.43 y está en eh arriba del mimo y a la izquierda del león ahí justito pegadita

A3 92.73 93.52 arriba del mimo

Sujeto B

B1 85.71 86.02 sí

B2 91.92 92.13 cómo

B3 92.67 92.88 ah

B4 93.72 94.45 okay sí

(a) `separated`

ahora hace la clasificacion para estos turnos:

A1 82.26 85.17 mm no me titila ahí está la oreja

B1 85.71 86.02 sí

A2 86.33 91.43 y está en eh arriba del mimo y a la izquierda del león ahí justito pegadita

B2 91.92 92.13 cómo

B3 92.67 92.88 ah

A3 92.73 93.52 arriba del mimo

B4 93.72 94.45 okay sí

(b) `interleaved`

Fig. 3.6: Variantes del hiperparámetro **estructura A-B**.

Ejemplos de variantes del hiperparámetro **frases**

ahora hace la clasificacion para estos turnos:

frases:

A 82.26 83.41 mm no me titila

A 83.57 83.89 ahí está

A 84.64 85.17 la oreja

B 85.71 86.02 sí

A 86.33 86.69 y está

A 87.49 88.78 en eh arriba del mimo

A 89.19 90.08 y a la izquierda del león

A 90.35 91.43 ahí justito pegadita

—-

turnos:

A1 82.26 85.17

B1 85.71 86.02

A2 86.33 91.43

(a) `before`

ahora hace la clasificacion para estos turnos:

A1 82.26 85.17 mm no me titila ahí está la oreja

B1 85.71 86.015 sí

A2 86.33 91.43 y está en eh arriba del mimo y a la izquierda del león ahí justito pegadita

(b) `inline`

Fig. 3.7: Variantes del hiperparámetro **frases**. Se puede observar que inline es una notación más compacta que before, al unir todas las frases (IPUs) del hablante por turno.

Ejemplos de variantes del hiperparámetro **transition table**

turnos:

A1 0.32 7.40 tenes que fijarte que el avion quede justo abajo de la pelota

B1 2.40 3.60 no lo

B2 8.10 9.25 ah ahí lo ví

(a) `sin transition table`

A1 0.32 7.40 tenes que fijarte que el avión quede justo abajo de la pelota

B1 2.40 3.60 no lo

B2 8.10 9.25 ah ahí lo ví

transiciones:

$0 \rightarrow A1$

$A1 \rightarrow B1$

$A1 \rightarrow B2$

(b) `con transition table. Notar que por la decisión que tomamos de ordenar los turnos por tiempo de inicio, no es tan inmediato ver que tanto B1 como B2 transicionan ambos desde A1. La transition table ayuda en marcarle al modelo exactamente cuáles fueron las transiciones.`

Fig. 3.8: Variantes del hiperparámetro **transition table**.

3.3. Implementación y ejecución

Las pruebas correspondientes a cada configuración probada de la grilla de hiperparámetros fueron ejecutadas de manera programática, automatizando tanto la construcción de los user prompts como el envío a distintos modelos de lenguaje a través de sus respectivas APIs (interfaces de programación de aplicaciones, por sus siglas en inglés), que permiten interactuar con los modelos de manera automática mediante solicitudes estructuradas.

Dado que nos interesaba hacer suficientes pruebas, la búsqueda de modelos fue de menor a mayor, empezando primero con modelos muy pequeños como LLaMA 3.2 y DeepSeek reasoner 7B, corriéndolos en Ollama. El rendimiento de estos modelos para la tarea fue mucho menor a lo esperado, rara vez la etiqueta era correcta y, más importante aún, ni siquiera respetaban el formato de salida. Es muy posible que entre la dificultad intrínseca del problema y la longitud del prompt explicando el contexto, sea mucho para modelos de ese tamaño.

Luego de investigar un poco más, evaluamos principalmente dos familias de modelos: por un lado, **LLaMA-3.3-70b-versatile**, ejecutado mediante la plataforma *Groq Cloud*[3], y por otro lado, **Gemini Pro**, a través de *Google AI Studio*[4]. Ambos entornos ofrecen acceso gratuito a APIs con ciertas limitaciones de tokens y llamadas por minuto o día, lo que

[3] https://groq.com/
[4] https://aistudio.google.com/

condicionó la planificación de los experimentos.

Groq Cloud resultó especialmente útil por ofrecer acceso a una amplia variedad de modelos open-source (como LLaMA, Mistral, Gemma, Mixtral, entre otros) con tiempos de respuesta reducidos y facilidad de integración vía API. Por el contrario, descartamos el uso de *Ollama*, que habíamos explorado inicialmente, ya que al ser local sólo permitía correr modelos relativamente pequeños (por ejemplo, LLaMA 4b o DeepSeek 7b) que mostraron un desempeño muy deficiente para esta tarea.

3.4. Métricas de evaluación

Para evaluar el rendimiento del modelo en nuestra tarea de clasificación, utilizamos métricas estándar en problemas de clasificación multiclase, principalmente Macro F1-score, que calcula el F1-score de cada clase individual y luego promedia los resultados. Esta elección se justifica en base a la desbalanceada distribución de clases en el corpus: etiquetas como `S` o `X2` aparecen con mucha mayor frecuencia que otras como `PI` o `BI`, por lo que métricas agregadas como accuracy pueden ser engañosas. El Macro F1 permite valorar si el modelo también logra capturar correctamente los casos menos frecuentes.

Usar accuracy en este contexto es especialmente no recomendable, ya que al tener 13 clases, los Falsos Negativos son excesivamente altos y distorsionan mucho la métrica.

Es importante remarcar que, al trabajar con modelos de lenguaje generativo como LLaMA o Gemini, no contamos con scores, o probabilidades, para cada clase en cada predicción, como sí ofrecen muchos modelos de aprendizaje supervisado clásico. Esto nos impide utilizar herramientas muy valiosas de evaluación como las curvas ROC, AUC, métricas de calibración, o el análisis del top-k de predicciones más probables.

Esto también imposibilita afinar umbrales de decisión para cada clase o construir clasificadores ajustados a distintos trade-offs entre *recall* y *precision*, algo común en entornos donde hay clases especialmente sensibles (como interrupciones o backchannels). En nuestro caso, la evaluación se basa exclusivamente en la clase que el modelo predice en primer lugar, lo cual es una limitación significativa frente a modelos supervisados tradicionales.

4. RESULTADOS

Evaluación cuantitativa de los modelos en el conjunto de desarrollo y en el conjunto de evaluación.

4.1. Resultados en desarrollo

A lo largo de esta sección analizamos los resultados obtenidos por las distintas combinaciones de hiperparámetros en el conjunto de desarrollo. En línea con trabajos previos, utilizamos como métrica principal el Macro F1. La Tabla 4.1 contiene los nombres de las configuraciones evaluadas, que serán referenciadas a lo largo de esta sección, en función de los hiperparámetros definidos en la Sección 3.2.

ID	Ejemplos	Prefill	Overlap	Formato	Ubicación	Transiciones
C1	perlabel	prefill	full	interleaved	inline	transition
C2	perlabel	prefill	simplified	separated	before	notransition
C3	plain	noprefill	full	interleaved	inline	transition
C4	plain	prefill	full	interleaved	before	notransition
C5	onetask	noprefill	boolean	separated	before	transition
C6	onetask	noprefill	full	interleaved	inline	notransition
C7	onetask	prefill	full	interleaved	inline	transition
C8	onetask	prefill	full	separated	inline	transition

Tab. 4.1: Resumen de las combinaciones de hiperparámetros evaluadas durante el desarrollo del sistema.

Como valor de referencia principal, tomamos los resultados de Brusco [24] – un artículo que resume y extiende la tesis de doctorado Brusco [9] –, obtenidos con un modelo supervisado clásico (Redes Neuronales Recurrentes) que utiliza atributos acústico-prosódicos. En la siguiente tabla se muestran los valores de F1 por clase y el macro promedio:

	BC	PI	S	X2	BC_O	O	X2_O	BI	I	Macro F1
Baseline	0.83	0.38	0.73	0.80	0.69	0.60	0.88	0.35	0.51	**0.64**

Tab. 4.2: Benchmark de referencia obtenido con modelos acústicos tradicionales basados en RNNs [24].

La Tabla 4.3 muestra los valores de F1 score, por etiqueta y en promedio, obtenidos para las ocho configuraciones con el modelo LLaMA-3.3-70b-versatile utilizando la plataforma Groq. Como puede verse, principalmente mirando los Macro F1 scores, la performance de este modelo fue completamente insuficiente. En particular, ninguna de las ocho configuraciones superó al baseline.

Config	X2	X2_O	S	O	PI	I	BC	BC_O	BI	F1 Macro	X1	X3
C1	0.07	0.03	0.56	0.04	0.04	0.04	0.57	0.08	0.03	0.16	1.00	0.71
C2	0.00	0.00	**0.61**	**0.54**	0.16	**0.28**	**0.64**	**0.21**	0.00	**0.27**	1.00	0.99
C3	**0.35**	0.00	0.53	0.05	0.07	0.07	0.50	0.02	0.08	0.19	0.99	0.01
C4	0.01	0.00	0.56	0.07	0.04	0.08	0.41	0.00	0.09	0.14	0.91	0.81
C5	0.27	0.00	0.53	0.00	0.00	0.00	0.28	0.00	0.00	0.12	0.99	0.00
C6	0.34	0.00	0.60	0.10	0.00	0.05	0.58	0.08	0.03	0.19	0.99	0.04
C7	0.25	0.00	0.58	0.07	0.01	0.03	0.46	0.02	0.00	0.16	0.99	0.69
C8	0.27	0.00	0.55	0.00	0.00	0.00	0.36	0.00	0.00	0.13	0.99	0.63
Baseline	0.80	0.88	0.73	0.60	0.38	0.51	0.83	0.69	0.35	**0.64**	-	-

Tab. 4.3: Resultados de F1 en el conjunto de desarrollo con LLaMA-3.3-70b-versatile (Groq). En negrita, los mejores resultados por cada categoría. Se incluyen también las columnas X1 y X3 que no se incluyeron en el baseline. Estas categorías se pueden ubicar de manera determinística, por lo que una performance mala en estas denota un entendimiento bajo de la tarea. En rojo, los valores para estas dos columnas que son menores a 0.95

La Tabla 4.4 presenta los resultados obtenidos con el modelo Gemini en Google AI Studio. En general, los scores son superiores a los de LLaMA, aunque siguen por debajo del benchmark. Sin embargo, categorías como **S**, **X2**, **O**, **PI**, **I**, **BC** y **BI** obtienen valores más cercanos, donde C2, C4 y C6 superan el benchmark en S; C1 y C4 lo hacen en PI; C4 en la clase I y C2 en BI.

Config	X2	X2_O	S	O	PI	I	BC	BC_O	BI	F1 Macro	X1	X3
C1	0.76	0.26	0.69	0.45	**0.48**	0.47	0.77	0.31	0.21	0.49	0.99	0.74
C2	**0.80**	**0.63**	**0.76**	0.05	0.15	0.39	**0.79**	**0.37**	**0.40**	0.48	1.00	0.99
C3	0.72	0.21	0.65	0.45	0.41	0.42	0.74	0.29	0.27	0.46	0.99	0.27
C4	**0.77**	0.26	**0.75**	0.49	**0.47**	**0.53**	0.77	0.29	0.15	0.50	1.00	0.98
C5	0.700	0.13	0.59	0.32	0.29	0.33	0.72	0.25	0.23	0.40	0.99	0.34
C6	**0.77**	0.26	**0.74**	**0.53**	0.45	0.50	**0.79**	0.31	0.27	**0.51**	0.99	0.40
C7	0.74	0.21	0.68	0.50	0.38	0.47	0.75	0.32	0.29	0.48	0.99	0.72
C8	0.71	0.21	0.63	0.39	0.29	0.29	0.73	0.26	0.19	0.41	0.99	0.69
Baseline	0.80	0.88	0.73	0.60	0.38	0.51	0.83	0.69	0.35	**0.64**	-	-

Tab. 4.4: Resultados de F1 en el conjunto de desarrollo con gemini-2.5-pro-exp-03-25 (Google AI Studio). En negrita, los mejores resultados por cada categoría. Se incluyen también las columnas X1 y X3 que no se incluyeron en el baseline. Estas categorías se pueden ubicar de manera determinística, por lo que una performance mala en estas denota un entendimiento bajo de la tarea. En rojo, los valores para estas dos columnas que son menores a 0.95

En términos generales, el modelo `Gemini 2.5 Pro` logró resultados notablemente superiores a los obtenidos con `LLaMA 3.3-70B` en todas las configuraciones. Esto es consistente con lo esperable: Gemini es un modelo cerrado de última generación optimizado para tareas generales y con muchos más parámetros, mientras que LLaMA es un modelo open-source que, si bien potente, suele necesitar ajustes para trabajar en tareas específicas. Esto se debe en principio a que la cantidad de parámetros en LLaMA es ampliamente inferior a la de Gemini.

En cuanto al macro F1-score, Gemini alcanzó un valor máximo de 0.51, muy por encima del mejor resultado con LLaMA (0.27) y más cercano al benchmark clásico (0.64). Esto sugiere que los modelos de lenguaje modernos pueden capturar patrones útiles incluso sin acceso a información acústica, aunque en este trabajo no se haya alcanzado la performance

de los modelos que si utilizan información no semántica.

Cabe destacar que este es el conjunto de desarrollo, en el cual los autores realizaron una búsqueda de hiperparámetros rigurosa que les permitió elegir el modelo reportado. Lo que estamos viendo en el benchmark es la "mejor" tabla obtenida con distintos modelos que se probaron. En cambio, en nuestro modelo basado en LLMs, esos datos no fueron vistos de ninguna manera, ya que no hay un entrenamiento de parámetros, por lo que nuestro modelo podría contar con cierta desventaja a la hora de la comparación.

Además, de las 144 configuraciones de hiperparámetros que teníamos para el modelo semántico, probamos únicamente ocho. El rango de F1 Macro obtenidos es relativamente amplio, ya que va desde 0.40 para la peor configuración hasta 0.51 para la mejor, o de 0.44 a 0.59 si incluimos X1 y X3 en el promedio. Esto sugiere que vale la pena explorar las otras 136 combinaciones restantes, o algunas de ellas.

Estas observaciones refuerzan la hipótesis de que, aunque los LLMs no cuentan con acceso a información prosódica o temporal detallada, logran compensarlo parcialmente a través del entendimiento semántico y estructural del diálogo, sobre todo cuando el prompt está cuidadosamente diseñado.

Si bien se busca alcanzar un rendimiento convincente en todos los escenarios, hay casos de uso para los cuales alguna etiqueta puede ser más relevante que otra. Por ejemplo, en el contexto de un call center donde se quieran analizar relaciones entre la actitud del agente comercial y la tasa de éxito, las interrupciones podrían tomar mucha relevancia, a pesar de no ser una etiqueta muy frecuente. En ese sentido, podemos destacar el caso de la clase PI, donde 5 de las configuraciones probadas superaron el benchmark de 0.38. Sería bueno seguir explorando en esta dirección para mantener estos valores y al mismo tiempo lograr una performance aceptable en el resto de las etiquetas.

Más allá de comparar el rendimiento contra un benchmark clásico, resulta igual de valioso analizar cómo se comportan las distintas configuraciones entre sí. Esto permite no solo identificar buenas prácticas para promptear este tipo de tarea, sino también evitar combinaciones que sistemáticamente generaron malos resultados.

En esta línea, se presentan a continuación seis gráficos que ilustran el valor de F1 Macro (incluyendo X1 y X3) en función de los diferentes hiperparámetros, evaluadas de manera aislada. Este análisis puede servir como guía para priorizar qué combinaciones vale la pena explorar entre las 136 configuraciones restantes que aún no fueron evaluadas.

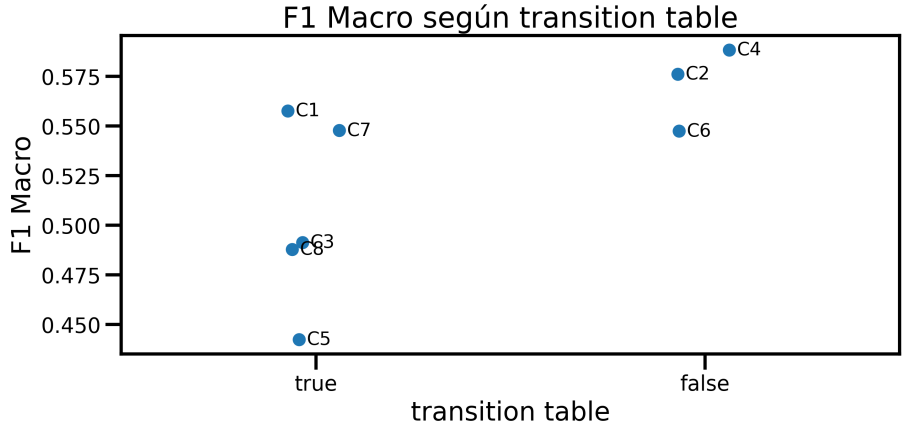


Fig. 4.1: F1 en base a si colocamos o no la transition table

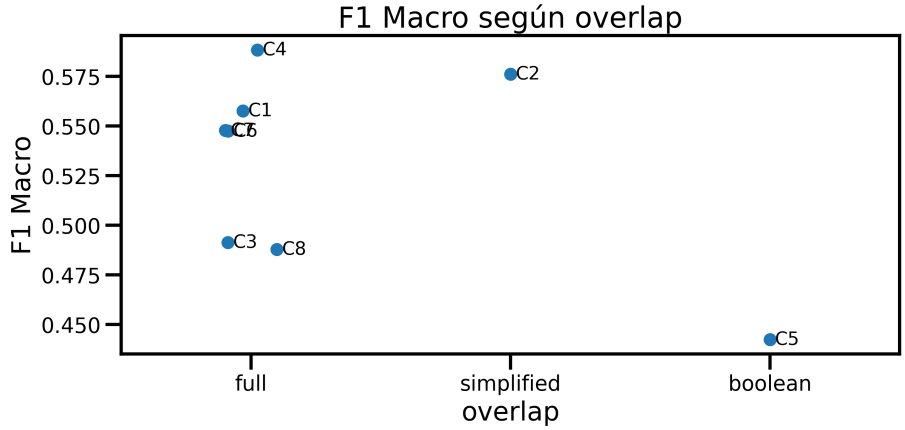


Fig. 4.2: F1 en base a las 3 alternativas que propusimos para tratar el overlap

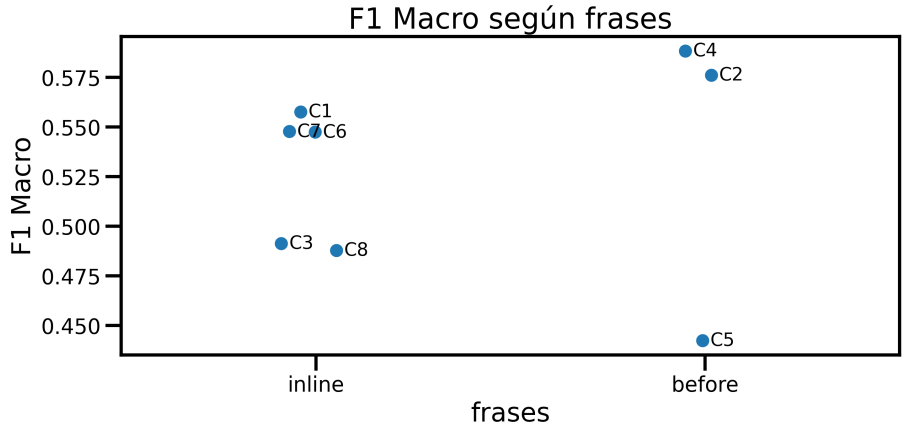


Fig. 4.3: F1 en función de si las frases van superpuestas con los turnos, o si tienen un apartado anterior

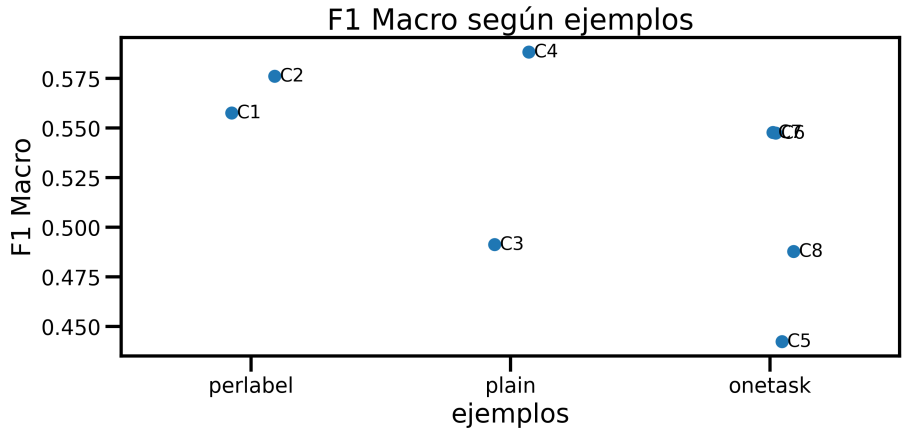


Fig. 4.4: F1 en base a la cantidad y tipos de ejemplos

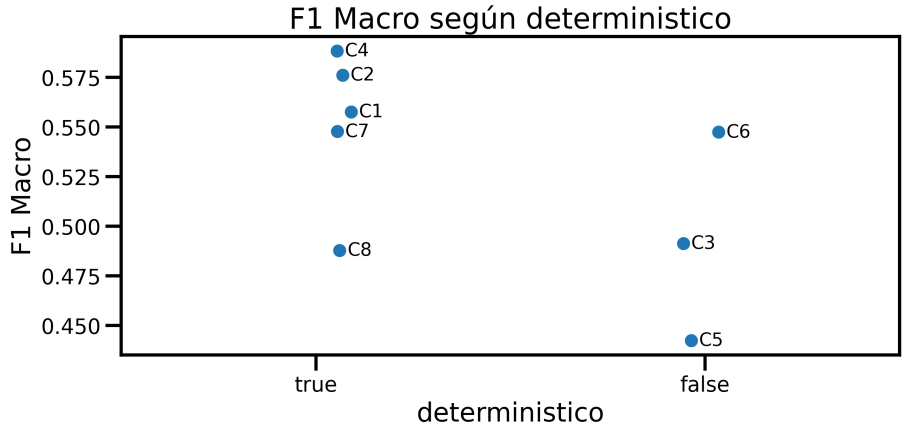


Fig. 4.5: F1 en base a la identificación determinística de las etiquetas X1, X3 y A

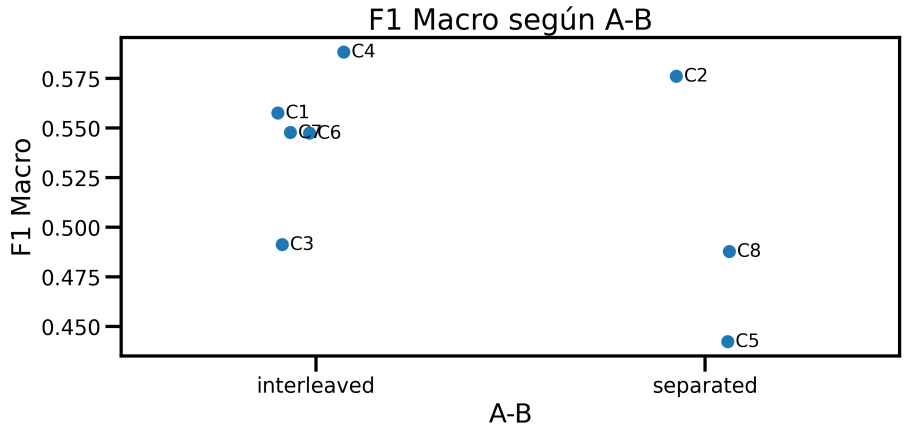


Fig. 4.6: F1 en base a la separación o intercalamiento de los turnos y frases de los hablantes A y B

A partir de estos gráficos podemos observar algunas cosas interesantes. Para empezar, y

yendo fuertemente contra nuestra intuición inicial, la tabla de transposiciones (que a modo de recordatorio indicaba de manera explícita todos los cambios de turno que hubo, indicando exactamente de que turno se transicionaba a cual otro) parecería afectar negativamente al modelo. En cuanto al hiperparámetro overlap, podemos ver que la única vez que tomo el valor boolean, la performance fue relativamente mala. Si bien con un solo valor de muestra no podemos descartarlo, nos lleva a pensar que posiblemente la configuración ideal no incluya overlap = boolean, y heurísticamente podríamos acortar el espacio de búsqueda restante. En cuanto a la cantidad de ejemplos, la opción de darle una tarea entera como ejemplo parecería ser la menos prometedora, mientras que la alternativa de dar un ejemplo por etiqueta dio buenos resultados, a pesar del desbalance de clases. En cuanto al determinismo, se ve una clara tendencia a mejores F1 Macro cuando se aprovecha el conocimiento del problema para clasificar determinísticamente a las X1, X3 y A. De manera similar, parece haber una tendencia a favor de usar el formato interleaved entre los hablantes, en vez de separar las frases y turnos de ambos en párrafos diferentes.

Si bien es sutil intentar sacar conclusiones universales a partir de estos 6 gráficos y 8 configuraciones, el hecho de que casi todas las nubes de puntos tengan alguna tendencia es un gran argumento para decir que la *ingeniería de prompting* influye en los resultados que obtengamos, o que los hiperparámetros que elegimos son relevantes. Si el prompt no afectara en nada a la capacidad del modelo, o si hubiéramos elegido dimensiones del prompt poco relevantes, esperaríamos ver nubes de puntos mas horizontales, sin valores favoritos para los hiperparámetros.

Si bien basamos la comparación con el benchmark en la métrica F1, es importante también considerar *precision* y *recall* a la hora de analizar estos modelos, para conocer mejor sus puntos fuertes y sus focos de mejora. La *precision* mide qué proporción de las predicciones positivas realizadas por el modelo son realmente correctas, mientras que la *recall* indica qué proporción de los casos positivos reales fueron correctamente identificados por el modelo. En la Tabla 4.5 y la Tabla 4.6 vemos respectivamente la *precision* y *recall* de los modelos de LLaMA. La Tabla 4.7 y la Tabla 4.8 muestran estos resultados para Gemini.

Config	X1	X2	X2_O	X3	S	O	PI	I	BC	BC_O	BI	Precision Macro
C1	0.99	0.50	0.06	0.99	0.47	0.3	0.12	0.10	0.50	0.08	0.09	**0.38**
C2	0.99	0.00	0.00	**1.00**	**0.62**	**0.56**	0.13	0.25	**0.56**	0.12	0.00	**0.38**
C3	0.98	0.51	0.00	0.33	0.45	0.24	0.13	0.11	0.42	0.02	0.00	0.29
C4	0.99	0.17	0.00	**1.00**	0.44	0.20	0.05	0.09	0.42	0.00	**0.50**	0.35
C5	0.98	0.48	0.00	0.00	0.40	0.00	0.00	0.00	0.39	0.00	0.00	0.20
C6	0.98	0.62	0.00	0.23	0.47	0.19	0.00	0.08	0.52	0.08	0.20	0.31
C7	0.98	**0.67**	0.00	0.98	0.46	0.31	0.25	0.07	0.49	0.03	0.00	**0.38**
C8	0.98	0.49	0.00	0.93	0.43	0.14	0.00	0.00	0.49	0.00	0.00	0.32

Tab. 4.5: Resultados de *precision* en el conjunto de desarrollo con LLaMA-3.3-70b-versatile (Groq).

Config	X1	X2	X2_O	X3	S	O	PI	I	BC	BC_O	BI	Recall Macro
C1	1.00	0.04	0.02	0.56	0.70	0.02	0.02	0.03	0.66	0.05	0.02	0.28
C2	1.00	0.00	0.00	**0.98**	0.61	**0.52**	0.23	0.28	**0.75**	**0.68**	0.00	**0.46**
C3	1.00	0.27	0.00	0.00	0.63	0.03	0.05	0.05	0.60	0.02	0.00	0.24
C4	0.84	0.00	0.00	0.68	0.80	0.04	0.03	0.07	0.40	0.00	0.05	0.26
C5	1.00	0.18	0.00	0.00	0.78	0.00	0.00	0.00	0.22	0.00	0.00	0.20
C6	1.00	0.24	0.00	0.02	**0.84**	0.07	0.00	0.03	0.64	0.09	0.02	0.27
C7	1.00	0.15	0.00	0.53	0.78	0.04	0.01	0.02	0.44	0.02	0.00	0.27
C8	1.00	0.18	0.00	0.48	0.77	0.00	0.00	0.00	0.28	0.00	0.00	0.25

Tab. 4.6: Resultados de *recall* en el conjunto de desarrollo con LLaMA-3.3-70b-versatile (Groq).

Config	X1	X2	X2_O	X3	S	O	PI	I	BC	BC_O	BI	Precision Macro
C1	0.98	0.71	0.16	0.99	0.89	**0.80**	0.70	0.47	0.69	0.19	0.53	**0.65**
C2	1.00	**0.73**	0.49	0.99	0.82	0.22	0.24	0.26	**0.74**	0.23	0.32	0.55
C3	0.98	0.64	0.12	0.60	0.88	0.73	0.69	0.41	0.65	0.18	0.43	0.57
C4	1.00	0.69	0.16	0.98	**0.91**	0.72	0.67	0.46	0.68	0.17	0.32	0.61
C5	0.98	0.62	0.07	0.67	0.77	0.50	0.59	0.41	0.66	0.15	0.47	0.54
C6	0.98	0.71	0.16	0.55	0.79	0.72	0.66	0.46	0.72	0.18	0.44	0.58
C7	0.98	0.68	0.13	0.98	0.88	0.74	**0.72**	**0.51**	0.68	0.20	**0.55**	**0.64**
C8	0.98	0.68	0.13	0.89	0.78	0.55	0.45	0.35	0.67	0.16	0.50	0.56

Tab. 4.7: Resultados de *precision* en el conjunto de desarrollo con Gemini.

Config	X1	X2	X2_O	X3	S	O	PI	I	BC	BC_O	BI	Recall Macro
C1	1.00	0.83	0.81	0.60	0.56	0.31	0.37	0.47	0.86	0.79	0.13	0.61
C2	1.00	**0.90**	**0.89**	**1.00**	**0.70**	0.03	0.11	**0.80**	0.86	0.93	**0.53**	**0.70**
C3	1.00	0.84	0.75	0.17	0.52	0.32	0.29	0.42	0.85	0.77	0.20	0.56
C4	0.99	0.87	0.68	0.99	0.64	0.37	0.37	0.63	**0.90**	**0.98**	0.10	0.68
C5	0.99	0.80	0.43	0.22	0.48	0.24	0.19	0.27	0.78	0.68	0.15	0.48
C6	1.00	0.86	0.70	0.32	0.69	0.42	0.34	0.55	0.88	0.93	0.20	0.63
C7	0.99	0.80	0.62	0.57	0.55	0.38	0.26	0.43	0.83	0.83	0.20	0.59
C8	0.99	0.74	0.57	0.57	0.52	0.30	0.21	0.25	0.79	0.65	0.12	0.52

Tab. 4.8: Resultados de *recall* en el conjunto de desarrollo con Gemini.

Nuevamente es notoria la supremacía de Gemini por sobre LLaMA. Viendo las tablas del modelo de Google, podemos observar que la etiqueta S tiene bastante buena *precision*, superando el 90 % en uno de los modelos.

En cuanto al *recall*, las categorías más recuperadas son BC y BC_O, con valores en torno a un 80 %.

Algo interesante es que estas observaciones se mantienen parcialmente en los modelos de LLaMA, donde el Switch (S) también fue de las etiquetas con mejor *precision* y el BC tuvo en general un mejor *recall* que las otras clases. Este alineamiento entre las clases que le cuestan o le resultan fácil a dos modelos distintos es algo sumamente deseable, ya que podría avalar el uso de modelos de menor tamaño y capacidad para la búsqueda de la configuración óptima.

Luego de estos análisis tenemos que hacer la *seleccion del modelo* para llevar al conjunto de evaluación. Dada su superioridad, vamos a seleccionar a Gemini como modelo, y dentro de las 8 configuraciones la C4, que corresponde a *plain_prefill_full_interleaved_before_notransition*. Consideramos que esta configuración fue la que mejor se desempeño, ya que estuvo únicamente 1 punto por debajo de C6 en cuanto a F1 macro, y mostro un entendimiento

significativamente superior de la tarea al tener 0.98 en X3, contra el valor de 0.4 por parte de C6. A continuación, el mensaje de sistema de C4:

Prompt del modelo seleccionado

Sos el mayor experto del mundo en clasificación de cambios de turno en conversaciones humano-humano. Te voy a pasar datos de una conversación, que tiene un hablante A y un hablante B. Esos datos son: momento de inicio de cada IPU, momento de fin de cada IPU, y palabras de la IPU.

A continuación, te comparto información sobre las categorías a predecir:

Estudio de transiciones de turno:

Definimos como unidad inter-pausal (IPU) a un segmento continuo de habla delimitado por pausas de al menos 200 milisegundos. Estos segmentos de habla pueden variar en longitud, desde palabras sueltas hasta oraciones completas.

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.

Una transición de cambio de turno es un evento en el cual el turno pasa de un interlocutor a otro en una conversación. Ya que no existe una única manera de delimitar una transición entre turnos, decidimos declarar la transición de turno, en el instante en que empieza el turno 2. 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. 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 SIN SOLAPAMIENTO:

- **Switch (S):** el interlocutor toma la palabra después de que el hablante complete su mensaje y deje un silencio.
- **Interrupción en pausa (PI):** el interlocutor toma la palabra luego de un silencio por parte del hablante, pero sin que este haya terminado su mensaje.

- **Backchannel (BC):** breve interlocución sin intención de tomar el turno (como "ajá", "claro", "sí").

CATEGORÍAS CON SOLAPAMIENTO:

- **Overlap (O):** igual que S, pero el nuevo hablante comienza antes de que el anterior termine.
- **Backchannel con solapamiento (BC_O):** igual que BC pero con solapamiento.
- **Interrupción con solapamiento (I):** igual que PI pero con solapamiento.
- **Butting-in (BI):** similar a I pero el hablante no logra tomar el turno (la interrupción queda inconclusa).

CATEGORÍAS AUXILIARES:

- **X1:** comienzo del primer turno en la conversación.
- **X2 y X2_O:** suceden a BC y BC_O cuando el hablante original retoma el turno.
- **X3:** dos IPUs inician casi simultáneamente (menos de 210 ms entre sí); la segunda lleva esta etiqueta.

Notar que las categorías X1, X2, X2_O y X3 son fáciles de determinar programáticamente en base a las IPUs y el resto de las etiquetas. Sin embargo, no basta con analizar los tiempos: es necesario comprender la semántica de lo que se está diciendo, ya que la intención del hablante juega un rol clave en la clasificación. Tu tarea es clasificar todos los instantes que PRECEDEN a cada cambio de turno, según las categorías mencionadas.

ACLARACIONES IMPORTANTES:

- Vas a recibir frases e información de los turnos ya segmentados, por lo tanto, tu tarea es **SOLO CLASIFICAR**, no segmentar.
- Siempre se te pasará un task completo.
- Las etiquetas `X1`, `X3` y `A` (para "ambiguo") te serán ya colocadas programáticamente.

INSTRUCCIONES IMPORTANTES SOBRE EL FORMATO DE SALIDA:

1) El formato de salida debe ser una tabla de cuatro columnas separadas por tabuladores: HABLANTE TAB INICIO TAB FIN TAB ETIQUETA

2) Clasifica EXCLUSIVAMENTE los turnos que aparecen en la sección "Turnos:" del user prompt.

- NO inventes ni añadas turnos con otros tiempos.
- NO modifiques los tiempos "INICIO" y "FIN" que ya están dados.

3) No incluyas texto adicional, ni explicación. Sólo las líneas de clasificaciones.

Ejemplo de salida:

A 0.719427 11.412306 X1

B 12.325461 17.961326 S

4.2. Resultados en el conjunto de evaluación

Esta instancia se usa generalmente para ver la capacidad de generalización del modelo, al testearlo en datos no vistos durante el entrenamiento. A pesar de que nuestro modelo no tuvo un proceso estándar de entrenamiento ni cuenta con parámetros (si con hiperparámetros), sigue siendo útil y necesaria esta evaluación. Primero por el simple hecho de que el pasaje de información de los datos al modelo no es solo a través del entrenamiento de los parámetros, sino que todo el proceso que hace quien desarrolla el modelo da lugar a decisiones de diseño en pos de una mejor performance, por lo que se necesita un conjunto de datos nunca visto ni por el modelo ni por el equipo de desarrollo para reportar una performance confiable de generalización.

Además, en una etapa de desarrollo se prueban varias configuraciones (en nuestro caso ocho), y es probable que la configuración ganadora además de ser realmente buena haya tenido una cuota de suerte.

A continuación vemos la performance del modelo benchmark de RNNs sobre el conjunto de evaluación, donde cabe aclarar que este conjunto si fue utilizado una única vez por este modelo (a diferencia del conjunto de datos de desarrollo), y por ende permite una comparación más justa.

Conjunto	S	BC	PI	X2	O	BC_O	I	BI	X2_O	Macro F1
#1 new tasks	.79	.84	.30	.87	.73	.58	.24	.24	.79	.64
#2 new sessions	.81	.89	.45	.90	.72	.67	.52	.40	.62	.70
Promedio	.80	.87	.38	.89	.73	.63	.38	.32	.71	**.67**

Tab. 4.9: Resultados del modelo benchmark RNN en el conjunto de evaluación. `new_tasks` y `new_sessions` son conceptos que tienen sentido en el contexto de este modelo, pero las `held_out tasks` son las últimas 2 tasks de cada sesión, mientras que las sesiones 7, 12 y 13 fueron usadas completas como `held_out sessions`.

y por otro lado, observemos la tabla obtenida por nuestro modelo. Se resaltan los resultados para **I** y **PI**, las dos categorías donde el baseline fue superado.

Conjunto	S	BC	PI	X2	O	BC_O	I	BI	X2_O	Macro F1
#1 new tasks	.73	.78	.49	.78	.55	.34	.39	.20	.44	.52
#2 new sessions	.75	.82	.55	.82	.56	.46	.41	.43	.39	.58
Promedio	.74	.80	**.42**	.80	.56	.40	**.40**	**.32**	.42	**.55**

Tab. 4.10: Resultados de nuestro mejor modelo en el conjunto de evaluación.

En esta tabla, para facilitar la comparación, excluimos las etiquetas X1 y X3 de nuestro modelo, como en la tabla del modelo benchmark en su paper original [24]. De esa manera podemos notar como el modelo obtiene una performance claramente inferior. En particular, parecería que las categoráas que más le están costando en comparación al modelo de RNNs, son las clases con overlap: X2_O, BC_O y O. Detectar estas cosas es fundamental, ya que además de probar las 136 configuraciones restantes, es muy importante identificar puntos de mejora estructurales en base a puntos débiles de los LLM que vayan saliendo a la luz.

Por otro lado, tenemos etiquetas con una performance aceptable, como bien pueden ser S, BC, X2 y BI, y luego estan PI e I, donde nuestro modelo suepró al benchmark.

Además del análisis cuantitativo, resulta muy informativo observar la `matriz de confusión` del modelo sobre el conjunto de evaluación

	X2	X2_O	S	O	PI	I	BC	BC_O	BI	Total
X2	206	18	1	1	0	0	3	1	1	231
X2_O	2	22	0	0	0	0	0	0	0	24
S	68	20	360	6	4	4	102	11	1	576
O	3	17	9	75	0	18	0	46	5	173
PI	1	0	21	0	22	4	7	4	0	59
I	0	1	0	14	0	18	0	8	0	41
BC	2	3	2	0	0	0	260	12	0	279
BC_O	0	0	0	0	0	0	0	29	0	29
BI	0	1	2	3	0	5	0	6	5	22

Tab. 4.11: Matriz de confusión del modelo sobre el conjunto de evaluación. Filas: etiqueta real. Columnas: predicción del modelo.

Esta tabla permite identificar patrones sistemáticos de error y evaluar si ciertas etiquetas tienden a ser confundidas entre sí, como podría ser el caso de la etiqueta S, que muchas veces fue confundida por un Backchannel por el modelo. Este efecto se da más marcado en

la etiqueta PI, donde prácticamente la mitad se clasificó como Switch.

Viendo las filas de la matriz podemos tener una muy buena idea del *recall*. Y si bien algunas etiquetas tienen un *recall* interesante, como por ejemplo X2, X2_O, BC o BC_O (dónde de esta última se recuperan 29 de 29 con un *recall* de 1), puede resultar menos interesante para buscar fortalezas del modelo que permitan armar un modelo parcial confiable en algunas subtareas.

Por otro lado, si miramos las columnas de la tabla podemos hablar sobre la *precision*, donde las más interesantes parecerían ser S o PI, y podrían darnos a pensar en un modelo que se corra inicialmente sobre el dataset a etiquetar y solo se les preste atención a las predicciones de estas 2 etiquetas. Un modelo así podría reducir la cantidad de clasificaciones que luego tendría que hacer el etiquetador manual u otro modelo.

5. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se exploró un enfoque novedoso para la clasificación offline de transiciones de turno en conversaciones humano-humano, utilizando Modelos de Lenguaje de Gran Escala (LLMs) a través de técnicas de prompting, sin necesidad de entrenamiento supervisado.

Se trabajó exclusivamente con las transcripciones del primer lote de grabaciones del Juego de Objetos del UBA Games Corpus, con el objetivo de evaluar si la información sintáctico-semántica contenida en el diálogo es suficiente para identificar con precisión patrones conversacionales complejos, y comparar su rendimiento con enfoques tradicionales basados en atributos acústico-prosódicos. Con este fin, se diseñó una grilla de hiperparámetros para evaluar sistemáticamente distintas estrategias de representación del problema, y se experimentó con los modelos LLaMA 3.3-70B y Gemini 2.5 Pro.

Los resultados obtenidos indican que, si bien el enfoque basado en LLMs es viable y demuestra una comprensión de la tarea, en su estado actual no logró superar el rendimiento global del modelo de referencia basado en Redes Neuronales Recurrentes que sí utiliza información acústica. El mejor modelo evaluado, Gemini 2.5 Pro, alcanzó un Macro F1 de 0.55 en el conjunto de evaluación, en comparación con el 0.67 del baseline. No obstante, se observó un desempeño competitivo e incluso superior en clases específicas de alta complejidad semántica, superando al baseline en la clasificación de interrupciones en pausa (PI) e interrupciones con solapamiento (I), con F1-scores de 0.42 y 0.40 respectivamente. El análisis también reveló una clara superioridad de Gemini 2.5 Pro sobre LLaMA 3.3-70B, y confirmó que el diseño del prompt (prompt engineering) es un factor crítico, ya que distintas configuraciones arrojaron variaciones significativas en el rendimiento.

A partir de estos hallazgos, se abren varias líneas de trabajo futuro. En primer lugar, es necesario continuar la exploración del espacio de hiperparámetros. En esta tesis se evaluaron solo 8 de las 144 configuraciones posibles; una búsqueda más exhaustiva, guiada por las tendencias observadas, podría descubrir combinaciones más efectivas. En segundo lugar, se propone un rediseño del prompt enfocado en las debilidades detectadas. El análisis de la matriz de confusión mostró dificultades para distinguir clases con solapamiento (como O, BC_O y X2_O). Futuros prompts podrían incluir marcos de razonamiento paso a paso (chain-of-thought) para guiar al modelo a diferenciar estas categorías de forma más robusta.

Otra dirección prometedora es el desarrollo de modelos híbridos y la adopción de nuevas tecnologías. Dado que los LLMs mostraron fortalezas en ciertas clases (como I y PI) y los modelos acústicos en otras, un sistema de ensamble que combine las predicciones de ambos podría alcanzar un rendimiento superior al de cualquiera de los dos por separado. Asimismo, el vertiginoso avance en el campo de los LLMs sugiere que la repetición de estos mismos experimentos con futuras generaciones de modelos, previsiblemente más potentes, podría por sí sola mejorar los resultados. La siguiente frontera tecnológica, sin embargo, radica en la emergencia de modelos genuinamente multimodales, capaces de recibir la señal de audio directamente como entrada. Esto permitiría unificar el análisis prosódico y

el semántico dentro de un único sistema al que se le podría hacer prompting sobre el audio directamente, superando las limitaciones del enfoque puramente textual explorado en esta tesis. Finalmente, se recomienda explorar el uso de modelos más pequeños y de código abierto (como LLaMA) para las etapas de desarrollo y prototipado rápido, reservando los modelos más potentes para la evaluación final. Si bien su rendimiento es inferior, su accesibilidad acelera la iteración sobre el diseño de prompts.

En definitiva, aunque el análisis puramente textual no reemplaza por completo la riqueza de las señales prosódicas, esta tesis demuestra que los LLMs son una herramienta con un potencial considerable para el análisis conversacional. Los resultados sientan las bases para futuros trabajos y abren el camino hacia nuevos enfoques multimodales en el etiquetado automático de transiciones de turno.

BIBLIOGRAFíA

[1] S. Duncan, "Some Signals and Rules for Taking Speaking," *Journal of Personality and Social Psychology*, vol. 23, n.º 2, págs. 283-292, 1972.

[2] A. Gravano, P. Brusco y S. Benus, "Who Do You Think Will Speak Next? Perception of Turn-Taking Cues in Slovak and Argentine Spanish.," en *INTERSPEECH*, 2016, págs. 1265-1269.

[3] A. Gravano y J. Hirschberg, "Turn-taking cues in task-oriented dialogue," *Computer Speech & Language*, vol. 25, n.º 3, págs. 601-634, 2011.

[4] E. Ekstedt y G. Skantze, "TurnGPT: a transformer-based language model for predicting turn-taking in spoken dialog," *arXiv preprint arXiv:2010.10874*, 2020.

[5] A. Gravano, J. E. Kamienkowski y P. Brusco, "Uba games corpus," Tech. Rep., Consejo Nacional de Investigaciones Científicas y Técnicas ..., inf. téc., 2023.

[6] H. Sacks, E. A. Schegloff y G. Jefferson, "A simplest systematics for the organization of turn-taking for conversation," *Language*, vol. 50, págs. 696-735, 1974.

[7] G. Skantze, "Towards a General, Continuous Model of Turn-taking in Spoken Dialogue using LSTM Recurrent Neural Networks," en *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 2018, págs. 220-230. DOI: 10.18653/v1/w17-5527.

[8] R. Masumura, T. Tanaka, A. Ando, R. Ishii, R. Higashinaka e Y. Aono, "Neural Dialogue Context Online End-of-Turn Detection," en *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, 2019, págs. 224-228. DOI: 10.18653/v1/w18-5024.

[9] P. Brusco, "Estudio translingüístico de pistas del manejo de turnos en diálogos hablados," Tesis doct., Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, 2021.

[10] G. W. Beattie, "Turn-taking and interruption in political interviews: Margaret Thatcher and Jim Callaghan compared and contrasted," *Semiotica*, vol. 39, n.º 1-2, págs. 93-114, 1982, ISSN: 16133692. DOI: 10.1515/semi.1982.39.1-2.93.

[11] C. Liu, C. T. Ishi y H. Ishiguro, "A Neural Turn-Taking Model without RNN.," en *INTERSPEECH*, 2019, págs. 4150-4154.

[12] S. Sicardi, "Aplicación de Redes Neuronales Convolucionales al Problema de Turn-Taking Usando Espectrogramas," Tesis de mtría., Universidad de Buenos Aires, 2020.

[13] J. Scherman, "Inclusión léxica y sintáctica en modelos de etiquetado offline de transiciones de turno," Tesis de mtría., Universidad de Buenos Aires, 2024.

[14] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[15] B. Jiang, E. Ekstedt y G. Skantze, "Response-conditioned turn-taking prediction," *arXiv preprint arXiv:2305.02036*, 2023.

[16] M. J. Pinto y T. Belpaeme, "Predictive turn-taking: Leveraging language models to anticipate turn transitions in human-robot dialogue," en *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, IEEE, 2024, págs. 1733-1738.

[17] B. P. Allen, F. Polat y P. Groth, "Shroom-indelab at semeval-2024 task 6: Zero- and few-shot llm-based classification for hallucination detection," *arXiv preprint arXiv:2404.03732*, 2024.

[18] S. Hassani, M. Sabetzadeh y D. Amyot, "An empirical study on LLM-based classification of requirements-related provisions in food-safety regulations," *Empirical Software Engineering*, vol. 30, n.º 3, pág. 72, 2025.

[19] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever et al., "Improving language understanding by generative pre-training," 2018.

[20] J. Devlin, M.-W. Chang, K. Lee y K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," en *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, págs. 4171-4186.

[21] A. Grattafiori, A. Dubey, A. Jauhri et al., "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[22] G. Team, R. Anil, S. Borgeaud et al., "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[23] H. Touvron, T. Lavril, G. Izacard et al., "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[24] P. Brusco y A. Gravano, "Automatic offline annotation of turn-taking transitions in task-oriented dialogue," *Computer Speech & Language*, vol. 78, pág. 101 462, 2023.