

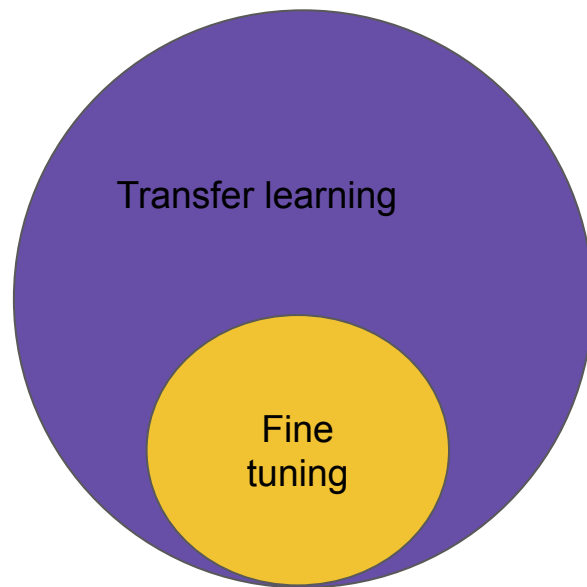
# M5. Minería de Texto

**Clase 5b. ¿Cómo interactuamos con un LLM?  
Una introducción a la ingeniería de prompts**



# Transfer learning y fine tuning

- Transfer learning: aplicar un modelo general pre-entrenado a una tarea nueva pero relacionada.
- Fine tuning: usar un modelo pre-entrenado pero re-entrenándolo (es decir, modificando algunos, varios o todos sus parámetros) para una tarea nueva.

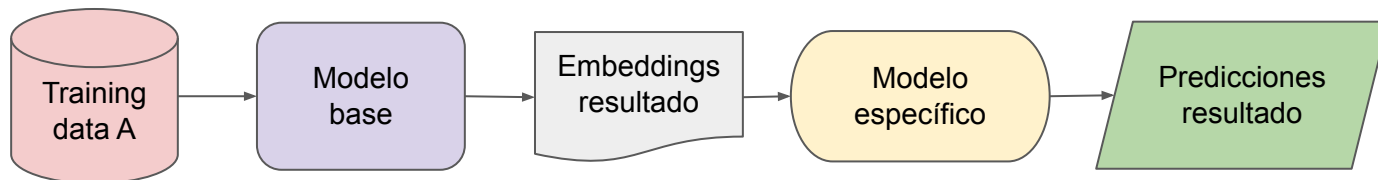


# Transfer learning y fine tuning



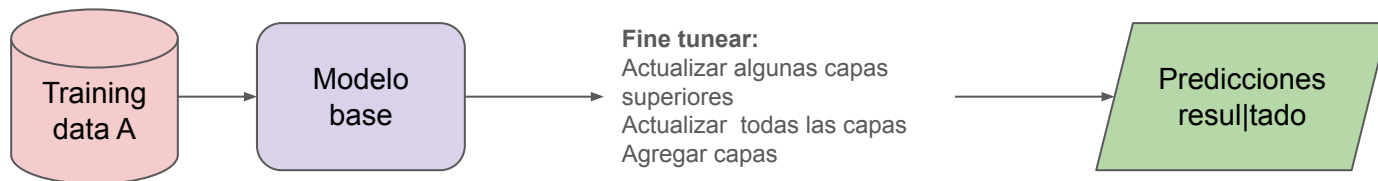
Ejemplo:

GPT4



Ejemplo:

Usar BERT para generar embeddings para usarlos como input de un random forest



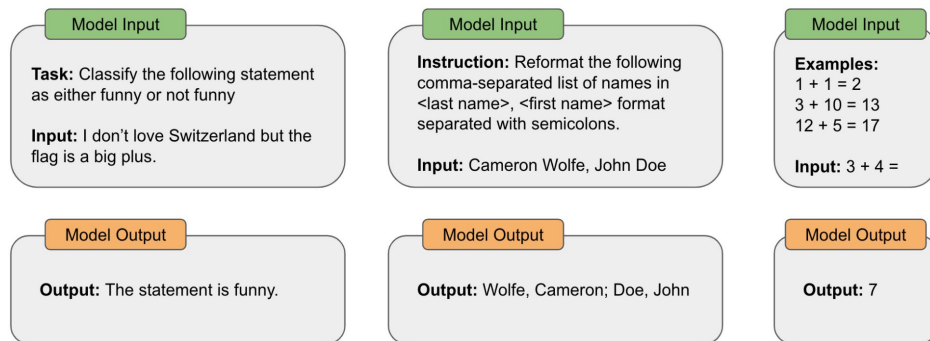
Ejemplo:

Hacer fine tuning sobre Llama para extraer información estructurada de texto no estructurado

# Prompt engineering

- La ingeniería de prompts se refiere a la formulación de instrucciones para Modelos de Lenguaje de gran tamaño (LLMs) con el objetivo de realizar tareas específicas. Estas instrucciones guían el comportamiento del modelo y determinan la calidad del resultado.

## Practical Prompt Engineering



# Componentes básicos de un prompt

- Instrucción:
  - ¿Qué debe hacer el modelo?
- Contexto:
  - ¿Qué información es útil?
- Formato:
  - ¿Qué tipo de salida queremos?
- Ejemplos (opcional)

# Prompt engineering

## Importancia

- **Definir la tarea:** Establecer claramente el objetivo del análisis de texto. ¿Qué tipo de información específica o “insight” se quiere extraer del texto?
- **Determinar el resultado deseado:** Identificar el tipo de respuesta esperada (información factual, opiniones subjetivas, etc.).
- **Considerar longitud y especificidad:** Equilibrar entre un prompt conciso y uno exploratorio según los objetivos del análisis.



# Prompt engineering

- **Incluir instrucciones o contexto:** Proveer instrucciones o información contextual que pueda ser relevante. Esto puede incluir “pedirle” al LLM que considere ciertos aspectos del texto
- **Basarse en investigaciones previas:** Utilizar instrucciones de codificación humana como referencia.
- **Hacer los resultados analizables:** Asegurar que la salida del modelo sea consistente y sistemática. Por ejemplo, si la salida esperada es una escala de 1 a 4, podría incluirse “[Contestar en formato “0, 1, 2, 3”. No expliques tu respuesta.]’
- **Iterar y probar:** Experimentar con diferentes formulaciones y ajustar según los resultados obtenidos.



# Prompt engineering

## Importancia

- Guía al modelo en el análisis del texto.
- Influye en la salida del modelo.
- Es una habilidad crucial para dirigir el análisis de textos

## Concepto

- Las instrucciones para el modelo representan la forma en que un concepto social científico se codifica.
- Podemos pensar a la ingeniería de prompts como un método cualitativo que busca capturar algún aspecto de la realidad social.



# Componentes avanzados de un prompt

- **Persona:**
  - Describir el rol que el LLM debe asumir. Por ejemplo, “Sos un experto en el análisis de la distribución del ingreso”
- **Instrucción:**
  - ¿Qué debe hacer el modelo?
- **Contexto:**
  - ¿Qué información es útil?
- **Formato:**
  - ¿Qué tipo de salida queremos?
- **Audiencia:**
  - El público/target del texto generado.
- **Tono:**
  - El tono de la “voz” que el LLM debe usar (formal, informal, académico, etc.)
- **Ejemplos (opcional)**



# Roles en los prompts

## Rol “system”

- Función: establecer el marco general de la conversación.
- Es un mensaje invisible para el usuario final, pero fundamental para dar instrucciones al modelo.
- Ejemplo de uso:
  - Indicar tono: “Respondé de manera académica y formal.”
  - Definir identidad: “Sos un asistente experto en sociología rural.”
  - Marcar límites: “Nunca des información personal ni inventes citas.”
- Podríamos pensarlo como el **contexto de interacción (o las “expectativas de rol” parsonianas)**: fija las reglas antes de que empiece el diálogo.



# Roles en los prompts

## Rol “user”

- Función: corresponde al mensaje del interlocutor humano (quien consulta).
- Es la entrada principal de texto o prompt.
- Ejemplo de uso:
  - “Explicá qué es una API con un ejemplo sencillo.”
  - “Resumí este texto en 3 puntos.”
- Sería algo así como el **acto de habla** del usuario: lo que inicia o guía la interacción.

## Otros roles: assistant o function/tool



# Ejercicio 1 - Elementos de un prompt

- Abran chatGPT o Gemini y loguéense a su cuenta.
- Diseñar un prompt que haga un resumen del [siguiente paper](#) usando solamente los elementos de “instrucción” y “datos”
- Diseñar un segundo prompt que haga un resumen del mismo texto usando solamente los elementos de “persona”, “instrucción” y “datos”
- Diseñar un tercer prompt que haga un resumen del mismo texto usando solamente los elementos de “persona”, “instrucción”, “contexto”, “formato”, “audiencia”, “tono” y “datos”



## Ejercicio 2 - Cambio de estilo

- Tomen el siguiente abstract y diseñen un prompt que modifique su estilo a un tono periodístico.

*El discurso de odio encontró en las redes sociales un espacio propicio. En el contexto argentino, nuevos partidos de derecha han irrumpido en el panorama político, ganando las elecciones de 2023. Muchas de estas nuevas figuras de derecha se hicieron populares gracias a su uso de las redes sociales, en un contexto de creciente violencia política. En este artículo, utilizamos herramientas cuantitativas y cualitativas para investigar la prevalencia del discurso de odio dirigido a mujeres políticas y analizar el papel de las diferentes afiliaciones políticas en la promoción de dicho discurso. Además, proponemos un modelo que predice las alineaciones políticas de los usuarios a partir de las descripciones de sus perfiles, lo que nos permite explorar la distribución del discurso de odio entre diferentes orientaciones políticas. Nuestros resultados ofrecen una descripción de la relación entre el discurso de odio de políticos y otros usuarios, y muestran que las figuras políticas y sus simpatizantes de derecha son fuertes emisores de discurso de odio, mientras que las mujeres, especialmente las de izquierda, son más propensas a recibir contenido violento. Fuente: [de este paper](#)*

- Ahora que modifique su estilo y formato para transformarlo en un hilo de

## Ejercicio 3 - NLP: clasificación

- Diseñar un prompt que permita clasificar el siguiente texto en función de su tono (“positivo”, “neutro”, “negativo”)

*Gran café de Villa Crespo que cuenta con amplios espacios y excelentes precios para la calidad que ofrecen. Óptima relación precio calidad. La atención brindada por los chicos es excelente, son muy carismáticos y buena onda. Los baños están bien limpios y son amplios. Sin embargo, las porciones son un poco pequeñas. Sin lugar a dudas un lugar para visitar y repetir.*

- Usar todos los elementos que vimos.



## Ejercicio 4 - NLP: temas

- Diseñar un prompt que permita identificar el texto anterior según los temas que toca.
- Usar todos los elementos que vimos.

# Ejercicio 5. NLP: extracción de entidades

- Diseñar un prompt que extraiga del siguiente texto las menciones a personas y lugares.
- Ahora hazlo usando los elementos de “persona”, “instrucción”, “contexto” y “datos”





## Ejercicio 6 - Roles / personas

- Diseñar un prompt que asuma el rol de Carlitos Marx para que explique qué es una clase social. Limitar la salida a 200 palabras.
  - Repetir con el rol Max Weber
  - Repetir con el rol Erik Olin Wright
  - Repetir con el rol David Grusky
- 
- ¿Qué elementos de un prompt están utilizados en estos ejemplos?



# X-shot learning

- Simplemente, supone proveer ejemplos (x-ejemplos) de la nueva tarea

```
pipeline(  
    ""For each tweet, describe its sentiment:  
    [Tweet]: "I hate it when my phone battery dies."  
    [Sentiment]: Negative  
    ###  
    [Tweet]: "My day has been 👍"  
    [Sentiment]: Positive  
    ###  
    [Tweet]: "This is the link to the article"  
    [Sentiment]: Neutral  
    ###  
    [Tweet]: "This new music video was incredible"  
    [Sentiment]: """)
```

Instruction

Few-shot examples

Prompt



# X-shot learning

## Ventajas

- No requiere grandes datasets etiquetados de forma ad-hoc
- No hace falta crear copias del modelo original
- Hay cierto carácter “intuitivo” en el diseño de un prompt

## Desventajas

- El prompt engineering es manual
- Los prompts son específicos de cada modelo
- El largo del contexto es una limitación:
  - Si agregamos más ejemplos, hay menos espacio para las instrucciones
  - Contextos más largos => mayor tiempo de respuesta
  - A veces los LLMs “olvidan” las partes intermedias ([Liu et al -2023-](#))



# Ejercicio 5

- Diseñar un prompt que permita clasificar el siguiente texto en función de su tono (“positivo”, “neutro”, “negativo”) usando la técnica de few-shot learning

*Gran café de Villa Crespo que cuenta con amplios espacios y excelentes precios para la calidad que ofrecen. Óptima relación precio calidad. La atención brindada por los chicos es excelente, son muy carismáticos y buena onda. Los baños están bien limpios y son amplios. Sin embargo, las porciones son un poco pequeñas. Sin lugar a dudas un lugar para visitar y repetir.*

- Usar todos los elementos que vimos.



# Chain of Thought (CoT) prompting

- Desglosar un problema o una pregunta compleja en una serie de pasos más pequeños y manejables. La idea es que al dividir la tarea en subproblemas y resolver cada uno de ellos secuencialmente, el modelo puede producir respuestas más precisas y coherentes.
- ¿Cómo funciona?
  - Descomposición de la tarea: Se divide la tarea principal en varios pasos intermedios que son más fáciles de resolver.
  - Razones explícitas: Se pide al modelo que genere una "cadena de pensamiento" en lugar de simplemente dar una respuesta directa. Esto significa que el modelo explica su proceso de pensamiento paso a paso.
  - Generación de la respuesta: Después de desglosar el problema y razonar a través de cada paso, el modelo genera la respuesta final.



# Chain of Thought (CoT) prompting

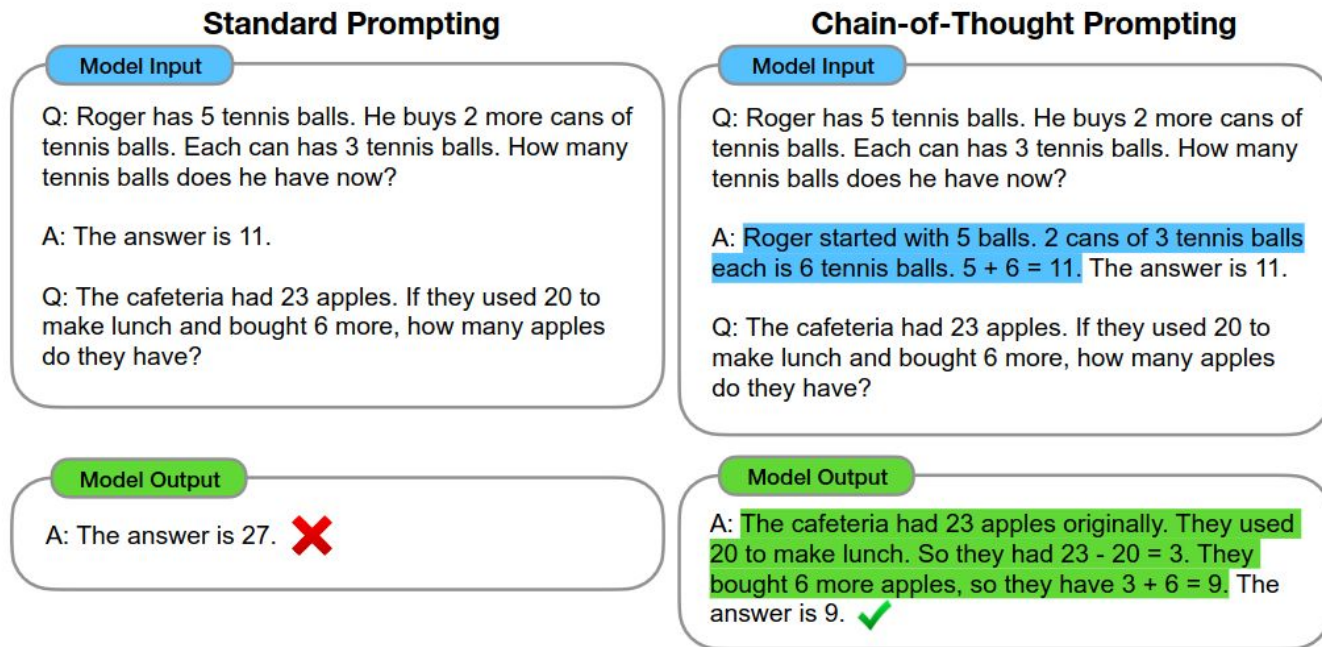


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

# Chain of Thought (CoT) prompting

## Ventajas

- Mejor precisión: Al abordar problemas complejos en pasos más pequeños, se mejora la precisión de las respuestas.
- Transparencia: Proporciona una explicación paso a paso, lo que hace que el proceso de pensamiento del modelo sea más transparente.
- Mejor manejo de problemas complejos: Ayuda al modelo a manejar tareas y preguntas complejas que de otro modo serían difíciles de resolver de una sola vez.



# Chain of Thought (CoT) prompting

Fuente: [Kojima et al \(2023\)](#)

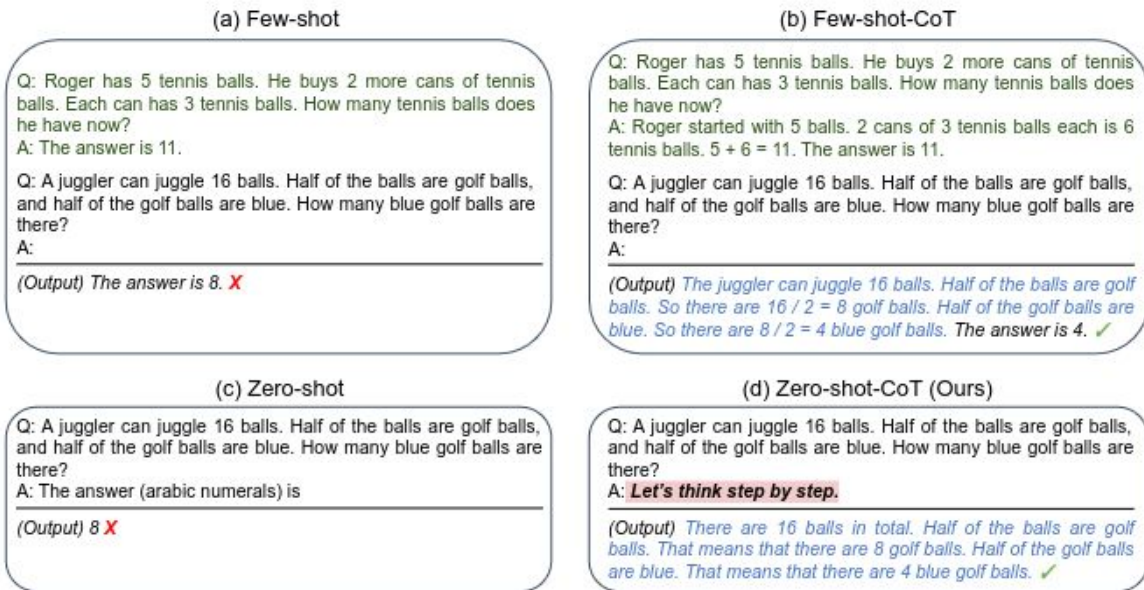


Figure 1: Example inputs and outputs of GPT-3 with (a) standard Few-shot ([Brown et al., 2020]), (b) Few-shot-CoT ([Wei et al., 2022]), (c) standard Zero-shot, and (d) ours (Zero-shot-CoT). Similar to Few-shot-CoT, Zero-shot-CoT facilitates multi-step reasoning (blue text) and reach correct answer where standard prompting fails. Unlike Few-shot-CoT using step-by-step reasoning examples **per task**, ours does not need any examples and just uses the same prompt “Let’s think step by step” *across all tasks* (arithmetic, symbolic, commonsense, and other logical reasoning tasks).





# Chain of Thought (CoT) prompting

## Desventajas

- Requiere más recursos computacionales: implica producir y procesar mayor cantidad de texto.
- Potencial para errores acumulativos: cada paso de la cadena de pensamiento puede introducir errores, y estos errores pueden acumularse, llevando a una respuesta final incorrecta.
- Limitaciones en la capacidad del modelo: algunos modelos pueden no ser lo suficientemente avanzados para manejar efectivamente CoT, lo que puede limitar su utilidad en ciertos contextos.

# Resumen...

- Los LLMs tiene la potencialidad de avanzar en la “automatización” de algunas tareas más o menos rutinarias (y no tanto) del proceso de investigación.
  - Análisis de datos
  - Clasificación de textos
  - Resúmenes
  - Brainstorming
  - Generación de hipótesis
- No obstante, hay límites:
  - Alucinaciones
  - Sesgos



# Resumen...

- Una forma (no la única, ni la más efectiva en cualquier situación) para mitigar estos problemas es el diseño correcto de los prompts (“Prompt engineering”)
- Algunas guías útiles:
  - [ChatGPT](#)
  - [Prompt Guide AI](#)

