

# Hiperparámetros de los modelos de lenguaje; prompting: Zero-Shot; Few-Shot

Fernando Carranza  
fernandocarranza86@gmail.com

Clase 3 Parte 1  
Martes 29/04/2025

- Clase 1: Nociones básicas de redes neuronales para el procesamiento del lenguaje natural
  - Parte 1: Tokenización, embeddings estáticos
  - Parte 2: el perceptrón; arquitectura básica de redes neuronales; funciones de activación; backpropagation.
- Clase 2: Grandes y pequeños modelos de lenguaje.
  - Parte 1: Mecanismo de atención; transformers; embeddings posicionales
  - Parte 2: Similitudes y diferencias entre modelos de lenguaje
- **Clase 3: *Prompt engineering***
  - **Parte 1: Hiperparámetros de los modelos de lenguaje; prompting: Zero-Shot; Few-Shot**
  - Parte 3: Prompting: RAG, Chain of Thought y otros

# Presentación

Estructura y temas de la clase de hoy:

- 1 Introducción
- 2 Hiperparámetros
- 3 Prompting
- 4 Recapitulación
- 5 Bibliografía

Para esta clase vamos a utilizar de base la siguiente bibliografía:

- Jurafsky, D. y Martin, J. H. (2025). *Model Alignment, Prompting, and In-Context Learning*. 3 edición. Online manuscript released January 12, 2025.
- <https://www.promptingguide.ai/es>

En la última clase dimos un pantallazo de los modelos existentes y qué características tienen.

Cabe mencionar que los modelos disponibles que se pueden descargar normalmente se alojan en Huggingface:

- <https://huggingface.co/>

En los modelos es posible reconocer dos grandes clases de valores:

- **Los parámetros:** Son los valores que se obtienen como resultado del entrenamiento.
- **Los hiperparámetros:** Son los valores que se asignan *a priori*.

Los hiperparámetros incluyen características tales como las siguientes:

- Tipo de embedding
- Cantidad de capas y tipo de red (LSTM, convolucional, Transformer, etc.).
- Valor del learning rate.
- Batch-size y número de épocas.

- Estos hiperparámetros conciernen a la forma y entrenamiento del modelo y, por ende, no pueden ser alterados por el usuario último.
- Ahora bien, existe un tipo de hiperparámetros que son lo suficientemente externos como para habilitar su manipulación por parte del usuario.



- Estos hiperparámetros incluyen la temperatura, el top\_k y el top\_p.
- El top\_p es un valor entre 0 y 1. La elección del token final se hará tomando alguno de los primeros más probables hasta sumar el valor de top\_p. Cuanto más bajo sea el valor de top\_p, más exactas van a ser las respuestas, ya que va a haber menor variedad de opciones menos probables. A valores más altos, las respuestas se vuelven más diversas.
- El top\_k es un valor de 0 a 100 y especifica la cantidad de tokens que se toman en cuenta para seleccionar la palabra siguiente.
- La temperatura afecta los valores de probabilidades de los tokens a elegir. Cuanto más baja sea la temperatura, más deterministas van a ser los resultados. A mayor temperatura, los resultados van a ser más creativos.
- <https://poloclub.github.io/transformer-explainer/>
- <https://huggingface.co/spaces/huggingface-projects/llama-2-7b-chat>

Es posible también interactuar con los modelos mediante chatbots. El chatbot Arena tiene la ventaja de que permite interactuar con una gran variedad de modelos de manera gratuita:

- <https://lmarena.ai/>

# Prompting

Las instrucciones lingüísticas que se le dan a un modelo para que realice una tarea se denominan *prompts*.

[https://www.instagram.com/p/DImua21sio/?igsh=MWQ0YzZmNXR5NmJzMA%3D%3D&img\\_index=1](https://www.instagram.com/p/DImua21sio/?igsh=MWQ0YzZmNXR5NmJzMA%3D%3D&img_index=1)

- Demonstrations son los ejemplos que se le dan a un modelo para que sea más clara la instrucción.
- Los *prompts* se pueden concebir también como señales de aprendizaje. Por eso, al *prompting* también se lo denomina *in-context-learning*.
- El proceso de encontrar los prompts más efectivos para una determinada tarea se conoce con el nombre de *prompt engineering*.

# Ejemplo de sentiment

El texto

*Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relax.*

se puede traducir como

*Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relax. In short, our stay was*

Las continuaciones que se obtienen se pueden mapear a las categorías deseadas.

Cuando se desea resolver tareas estándar, se suele recurrir a *templates*: textos orientados a tareas en particular acompañados de lugares para introducir un input en particular.

Summarization:	{input} ; tldr;
Translation	{input} ; translate to French:
Sentiment	{input}; Overall, it was
Fine-Grained-Sentiment	{input}; What aspects were important in this review?

Para lograr mejores resultados, el prompt tiene que ser lo menos ambiguo posible y restringir lo más posible las opciones

*Human: Do you think that "input" has negative or positive sentiment? Choices:*

*(P) Positive*

*(N) Negative*

*Assistant: I believe the best answer is: (*



## LLM Outputs for Basic Prompts

<b>Original Review (\$INPUT)</b>	Did not like the service that I was provided, when I entered the hotel. I also did not like the area, in which the hotel was located. Too much noise and events going on for me to feel relax and away from the city life.
<b>Sentiment</b>	<b>Prompt:</b> \$INPUT + In short, our stay was <b>Output:</b> not enjoyable
<b>Fine-grained Sentiment</b>	<b>Prompt:</b> \$INPUT + These aspects were important to the reviewer: <b>Output:</b> 1. Poor service 2. Unpleasant location 3. Noisy and busy area
<b>Summarization</b>	<b>Prompt:</b> \$INPUT + tl;dr <b>Output:</b> I had a bad experience with the hotel's service and the location was loud and busy.
<b>Translation</b>	<b>Prompt:</b> \$INPUT + Translate this to French <b>Output:</b> Je n'ai pas aimé le service qui m'a été offert lorsque je suis entré dans l'hôtel. Je n'ai également pas aimé la zone dans laquelle se trouvait l'hôtel. Trop de bruit et d'événements pour que je me sente détendu et loin de la vie citadine.

Figura: Tomado de Jurafsky y Martin 2025: 4

Una recomendación que se hace al diseñar prompts es decirle al modelo siempre qué hacer, no qué no hacer.

El siguiente es un agente que recomienda películas a un cliente. NO PIDA INTERESES. NO PIDA INFORMACIÓN PERSONAL. Cliente: Por favor, recomiéndeme una película basada en mis intereses. Agente:

Claro, puedo recomendar una película según tus intereses. ¿Qué tipo de película te gustaría ver? ¿Prefieres acción, comedia, romance o algo más?

<https://www.promptingguide.ai/es/introduction/tips>

Una recomendación que se hace al diseñar *prompts* es siempre decirle al modelo qué hacer, no qué no hacer.

El siguiente es un agente que recomienda películas a un cliente. El agente es responsable de recomendar una película de las más populares a nivel mundial. Debe abstenerse de preguntar a los usuarios por sus preferencias y evitar pedir información personal. Si el agente no tiene una película para recomendar, debe responder "Lo siento, no pude encontrar una película para recomendar hoy". Cliente: Por favor, recomiéndeme una película basada en mis intereses. Agente:

Lo siento, no tengo información sobre tus intereses. Sin embargo, aquí está una lista de las películas más populares a nivel mundial en este momento: [lista de películas]. Espero que encuentres algo que te guste.

<https://www.promptingguide.ai/es/introduction/tips>

En resumen, el *prompt* se realiza transformando cada tarea en una forma adecuada para la generación contextual por parte de un LLM, de la siguiente manera:

- 1 Para una tarea dada, se desarrolla una plantilla específica de la tarea que tiene un parámetro libre para el texto de entrada.
- 2 Dado el texto de entrada y la plantilla específica de la tarea, este se utiliza para instanciar un prompt lleno, que luego se pasa a un modelo de lenguaje preentrenado.
- 3 Se utiliza *autoregressive decoding* para generar una secuencia de salidas de tokens.
- 4 La salida del modelo puede usarse directamente como la salida deseada (como en el caso de tareas naturalmente generativas como la traducción o la resumen), o bien puede extraerse una respuesta adecuada para la tarea de la salida generada (como en el caso de la clasificación).

Es común enriquecer los prompts con ejemplos (*demonstrations*) que sirvan para guiar al modelo para que dé la respuesta más cercana a lo deseado.

- Este tipo de prompting se denomina *few-shot prompting*.
- Como contraposición, la versión sin ejemplos se denomina *zero-shot prompting*.

**Definition:** This task is about writing a correct answer for the reading comprehension task. Based on the information provided in a given passage, you should identify the shortest continuous text span from the passage that serves as an answer to the given question. Avoid answers that are incorrect or provides incomplete justification for the question.

**Passage:** Beyoncé Giselle Knowles-Carter (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their hiatus saw the release of Beyoncé's debut album, *Dangerously in Love* (2003), which established her as a solo artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".

**Examples:**

Q: In what city and state did Beyoncé grow up?

A: Houston, Texas

Q: What areas did Beyoncé compete in when she was growing up?

A: singing and dancing

Q: When did Beyoncé release *Dangerously in Love*?

A: 2003

Q: When did Beyoncé start becoming popular?

A:

Figura: Tomado de Jurafsky y Martin 2025: 5

- El número de *demonstrations* no tiene que ser muy grande. El rendimiento mejora considerablemente con un ejemplo y no parece ganar mucho más en los ejemplos siguientes.
- El modelo parece usar los ejemplos solo como modelos de respuesta.
- De hecho, ejemplos con respuestas erróneas pueden mejorar al modelo tanto como ejemplos correctos (ver Webson y Pavlick 2022).

Las *demonstrations* se crean típicamente formateando ejemplos tomados de un set de entrenamiento etiquetado.

Una heurística común para seleccionar los ejemplos es hacer sucesivas pruebas y evaluando qué ejemplo funciona mejor.



Un prompt puede contener cualquiera de los siguientes componentes:

- **Instrucción:** una tarea o instrucción específica que deseas que el modelo realice.
- **Contexto:** puede involucrar información externa o contexto adicional que puede dirigir al modelo a respuestas mejores.
- **Datos de entrada:** es la entrada o pregunta para la que nos interesa encontrar una respuesta.
- **Indicador de salida:** indica el tipo o formato de la salida.

No todos los componentes son necesarios para un prompt y el formato depende de la tarea en cuestión.

Tomado de

<https://www.promptingguide.ai/es/introduction/elements>

Hacer que un modelo aprenda preentrenándolo consiste en actualizar sus parámetros a partir del descenso de gradiente en función de una determinada función de pérdida.

Mediante el *prompting* también podemos hacer que el modelo aprenda. Esto se denomina in-context-learning. Pero este aprendizaje no sucede mediante la actualización de parámetros, sino que sucede mediante...

Hacer que un modelo aprenda preentrenándolo consiste en actualizar sus parámetros a partir del descenso de gradiente en función de una determinada función de pérdida.

Mediante el *prompting* también podemos hacer que el modelo aprenda. Esto se denomina in-context-learning. Pero este aprendizaje no sucede mediante la actualización de parámetros, sino que sucede mediante... bueno, en realidad no sabemos

Pero Olsson *et al.* (2022) tienen algunas intuiciones. Se ha descubierto que los transformers tienen lo que se conoce como *induction heads*: una parte del componente atencional de los transformers que se especializa en la predicción de secuencias repetidas. Se ha hipotetizado que estas heads son las responsables del *in-context-learning*. Estudios mostraron que al extirpar las *inductions heads* de los modelos, el *in-context-learning* decrece.

Para que a partir de la interacción con el modelo, este aprenda de manera más definitiva, es necesario hacer model alignment, es decir, es la tarea en la que se reentrena a un modelo para ajustarlo a la preferencia de sus usuarios.

El system prompt es un tipo de prompt general que se le agrega al modelo por sobre la interacción concreta y que regula el comportamiento del modelo.

En esta clase introducimos los siguientes temas:

- Algunos de los hiperparámetros relevantes de los grandes y pequeños modelos de lenguaje
- Qué es el prompting.
- En qué consiste el prompting zero-shot y few-shot.
- Qué es el in-context-learning.

# Bibliografía I

Jurafsky, D. y Martin, J. H. (2025). *Model Alignment, Prompting, and In-Context Learning*. 3 edición. Online manuscript released January 12, 2025.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., y Olah, C. (2022). In-context learning and induction heads. *arXiv*.

Webson, A. y Pavlick, E. (2022). Do prompt-based models really understand the meaning of their prompts? *arXiv*.