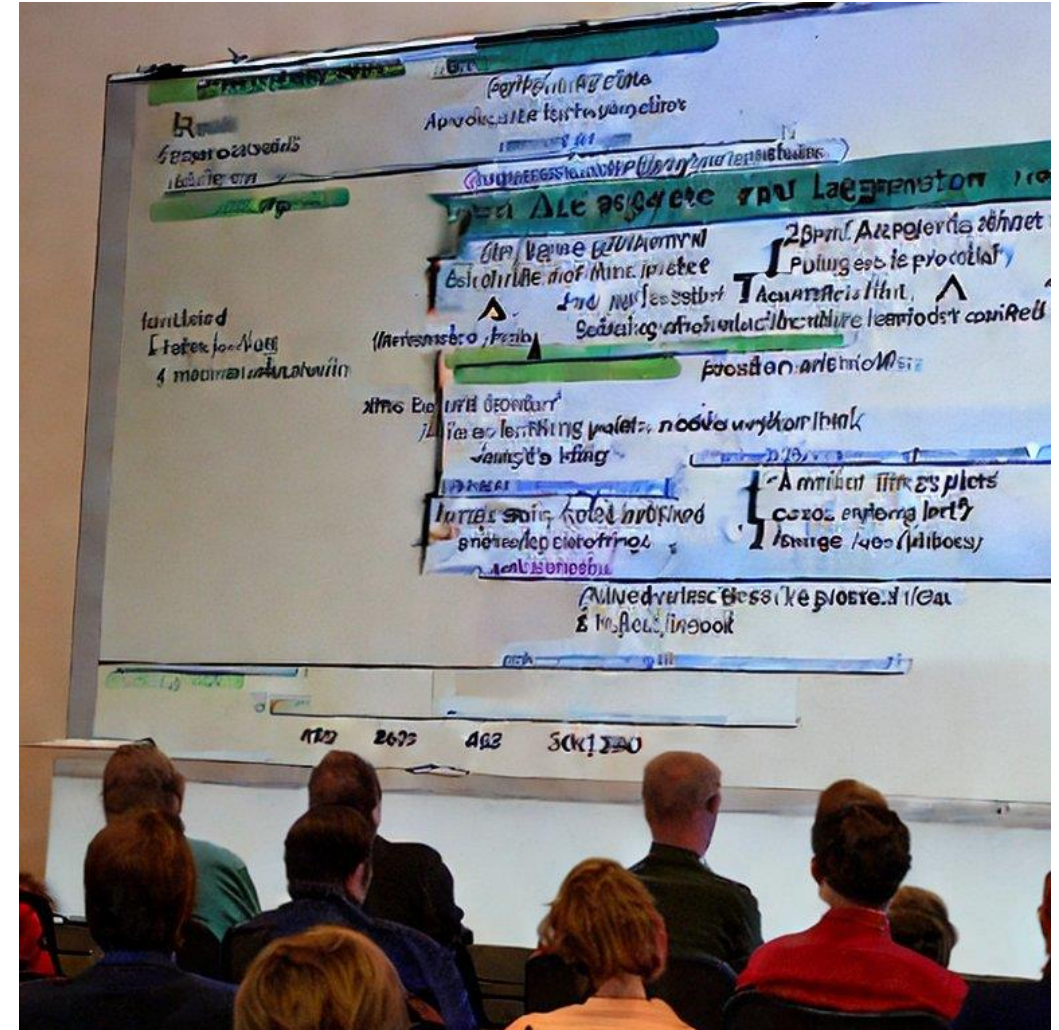


LLMs e IA Generativa

Clase 8

Optimización de LLMs
GenIA multimodal

Mg. Ing. Ezequiel Guinsburg
ezequiel.guinsburg@gmail.com



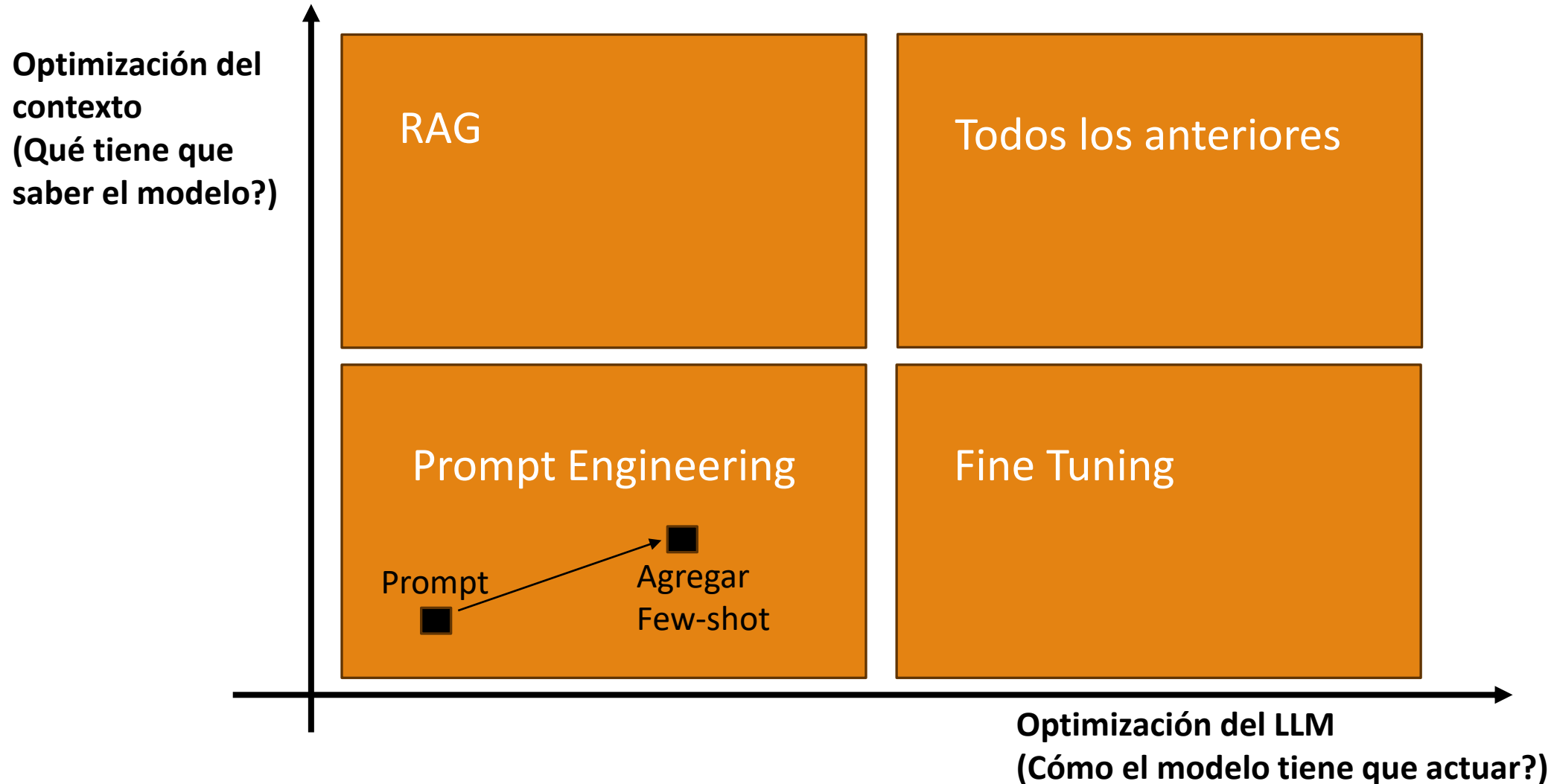
Referencias:

- Paper Visual Instruction Tuning - Haotian Liu
- Paper Hicherical Text-Conditional Image Generation with CLIP Latents - <https://arxiv.org/pdf/2204.06125>
- Denoising Diffusion Probabilistic Models - <https://arxiv.org/pdf/2006.11239>

Temas:

- El camino de la optimización en el uso de los LLMs.
- I.A. generativa multimodal.
 - Ejemplo teórico de modelo de texto-imagen.
 - Ejemplo práctico.
- Ejercicio 3.

Prompting Vs. RAG Vs. Fine tuning



Prompting Vs. RAG Vs. Fine tuning

Prompt Engineering

Bueno para:

- Testear y aprender rápidamente.
- Cuando se utiliza y se evalúa nos da una idea de cómo optimizar

Malo para:

- Introducir nueva información.
- Replicar consistentemente un estilo o método (i.e. aprender un nuevo lenguaje de programación).
- Minimizar el uso de tokens.

Prompting Vs. RAG Vs. Fine tuning

Prompt Engineering

Claves:

- Instrucciones claras.
- Dar tiempo para pensar (step-by-step).
- Dividir un problema complejo en instrucciones simples.
- Incluir ejemplos y evidencia -> Few-shot examples.

Prompting Vs. RAG Vs. Fine tuning

RAG

Bueno para:

- Introducir nueva información para actualizar la del modelo.
- Reducir alucinaciones controlando el contenido

Malo para:

- Entendimiento de temas muy abarcativos y complejos.
- Enseñar al modelo a aprender nuevos lenguajes, formatos o estilos.
- Reducir el uso de tokens.

Prompting Vs. RAG Vs. Fine tuning

Fine tuning

Objetivo: Continuar el proceso de entrenamiento en un dataset de dominio menor para optimizar el modelo en una tarea específica.

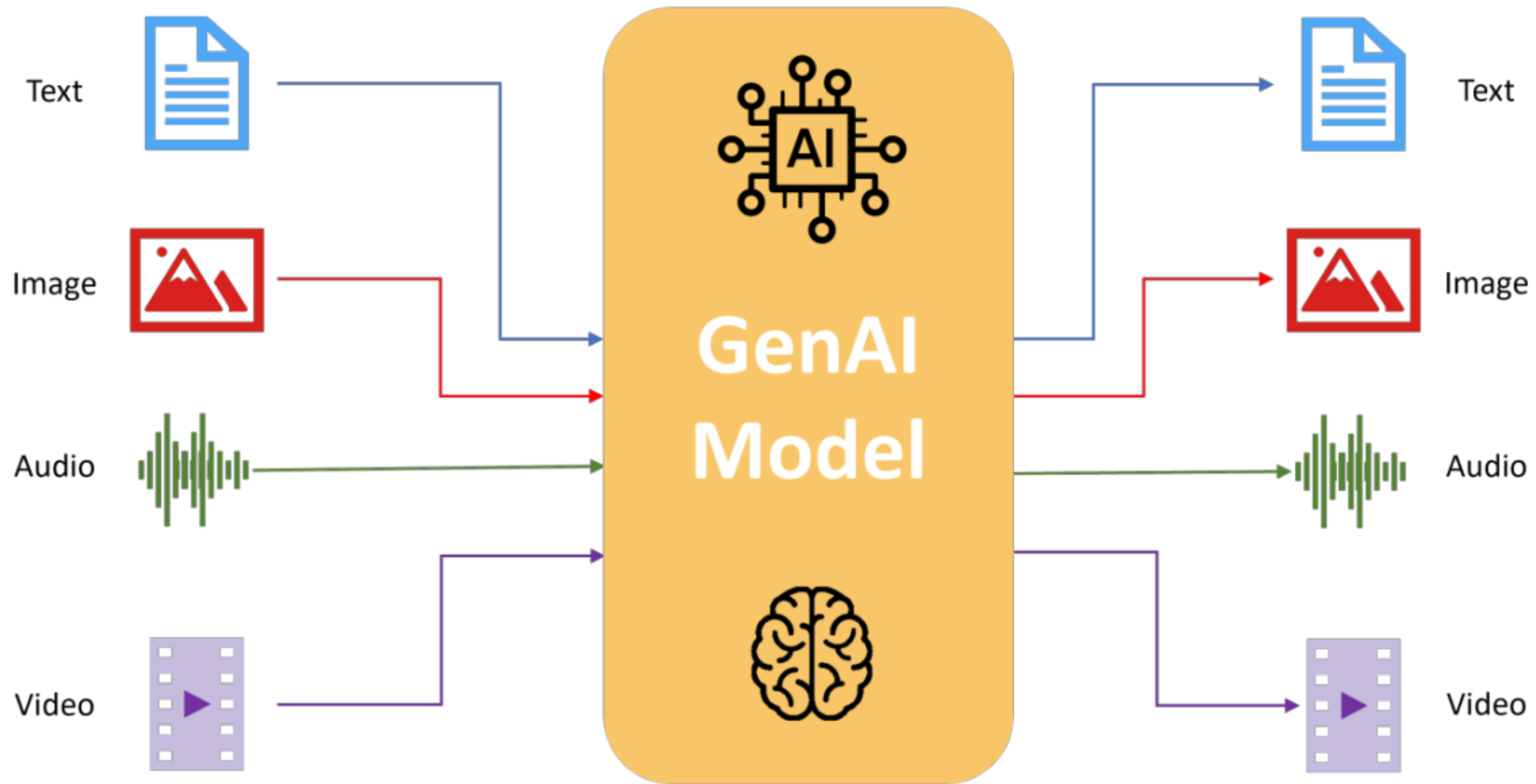
Bueno para:

- Mejorar la performance del modelo en un dominio específico.
- Enfatizar el conocimiento que ya existe en un modelo.
- Mejorar la eficiencia (reducción de la cantidad de tokens).

Malo para:

- Agregar nueva información al modelo.
- Iteración rápida en una optimización.

I.A. generativa multimodal



I.A. generativa multimodal

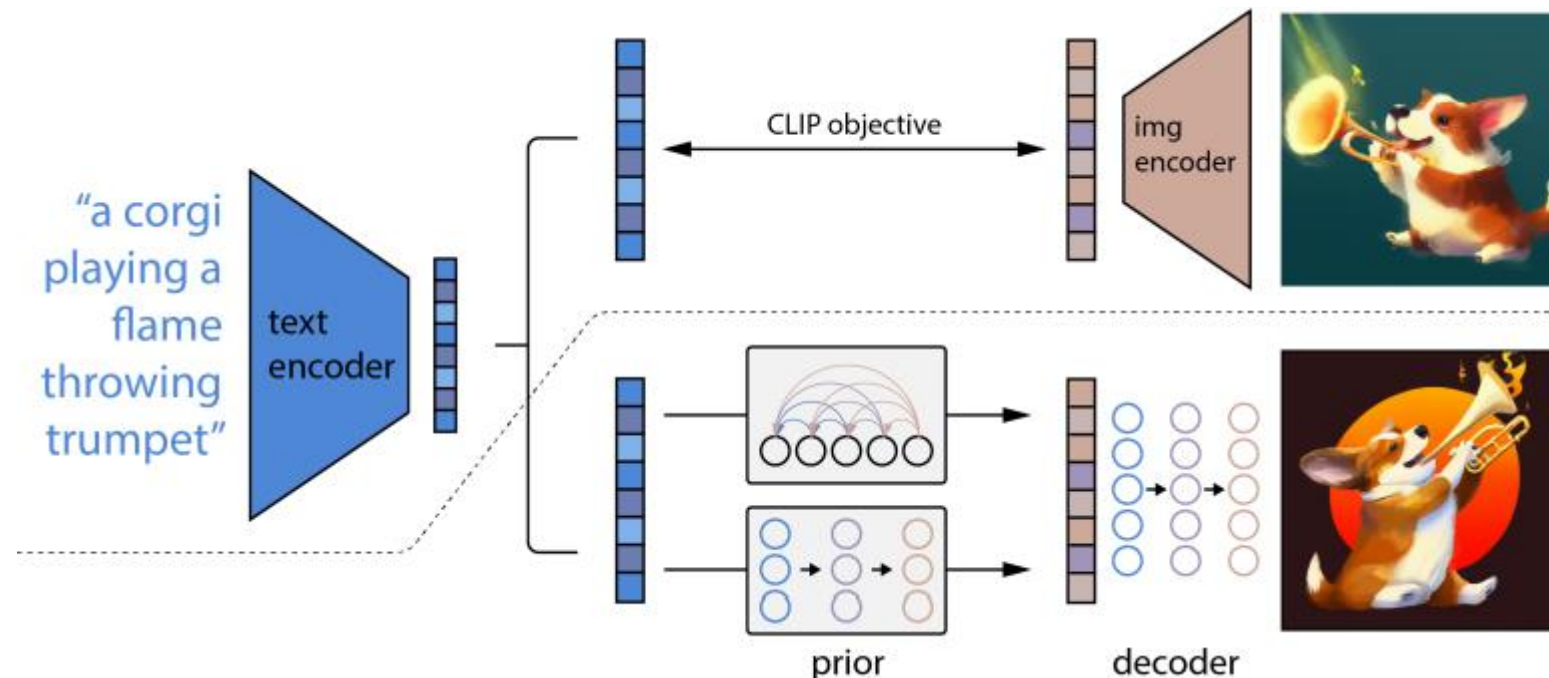


I.A. generativa multimodal – Ejemplo de modelo

Generación de imágenes condicionada por embeddings de texto.

Encoder (CLIP) – Decoder (Difussion model). **[unCLIP]**

Se utiliza el estado latente del modelo CLIP para manipular los embeddings y sacar distintos tipos de imagen



I.A. generativa multimodal – Ejemplo de modelo

Dataset de entrenamiento (x, y)

\mathbf{x} = imágenes | \mathbf{y} = descripciones

\mathbf{z}_i = embeddings de imagen | \mathbf{z}_t = embeddings de texto

Se utilizan dos componentes:

prior $\mathbf{P}(\mathbf{z}_i | \mathbf{y})$ = produce los embeddings de imagenes CLIP

decoder $\mathbf{P}(\mathbf{x} | \mathbf{z}_i, \mathbf{y})$ = produce las imágenes x condicionadas a z_i e y

Combinando los dos componentes tenemos el modelo generativo de imágenes \mathbf{x} dado las descripciones \mathbf{y} :

$$P(x|y) = P(x, z_i|y) = P(x|z_i, y)P(z_i|y)$$

I.A. generativa multimodal – Ejemplo de modelo

Reconstrucción de imágenes a partir de espacios latentes reducidos



Trabajo comparativo de modelos:

<https://distill.pub/2021/multimodal-neurons/>

Repositorio del código para utilizar:

<https://github.com/Stability-AI/stablediffusion>

I.A. generativa multimodal

Ejemplos de modelprácticos

Ejemplo 1: (text2img)

<https://huggingface.co/CompVis/stable-diffusion-v1-4>

<https://colab.research.google.com/drive/1SXnX2zlx5oE3sltz65stMYvz0nYyfEvl?usp=sharing>

Ejemplo 2 (Multimodal understanding)

<https://huggingface.co/deepseek-ai/Janus-1.3B>

Ejemplo 3 (Imagen a texto estructurado)

Ver “Codigo->Ejemplo_IMG2TXT.ipynb”

Ejercicio en clase:

- **Consigna:** Implementar una notebook para generar imágenes a partir de texto (text2img) utilizando un modelo open source.
- **Recomendaciones** (no excluyentes):
 - Utilizar modelo deHuggingFace.
 - Utilizar una instancia de Colab con GPU.
 - No es relevante la performance del modelo.
- **Entregables:** Link a la notebook de acceso público. Puede ser al final de la clase (una entrega grupal) o individualmente hasta la fecha límite de entrega (Martes 10 de Diciembre).
- **Objetivo:** Aprender a buscar e instanciar modelos multimodales de manera simple y rápida.