

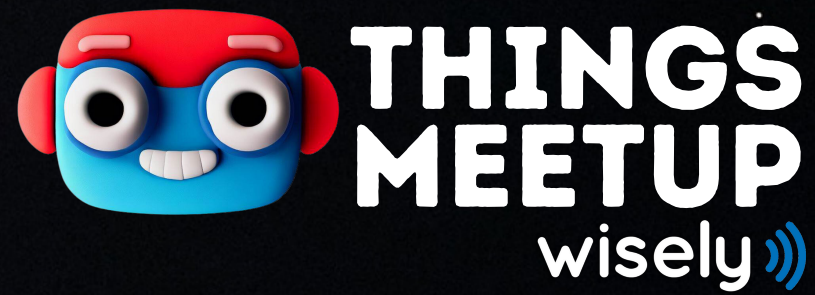
Bienvenid@s a nuestra comunidad



THINGS MEETUP

ORGANIZA Wisely)))

INTERNET DE LAS COSAS - INTELIGENCIA ARTIFICIAL
- REALIDAD VIRTUAL



Uno de los eventos público gratuito más importante y activo sobre IoT + VR + AI



wisely 

IoT | VR | AI

wisely.cl



BOND

Monitoreo y control
industrial con IoT



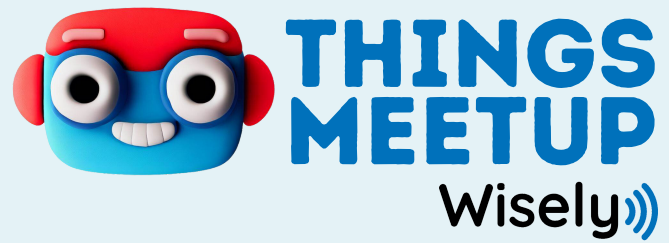
HOUND
GAMES

Realidad Virtual (VR) y
Realidad Aumentada (AR)




pymecopilot

Copiloto para inteligencia
empresarial con AI



PROMPTING AVANZADO Y PROTOTIPADO

Descubre cómo utilizar modelos de lenguaje para generar contenido de calidad y mejorar la interacción.

 26 de Septiembre  18 a 20 HS

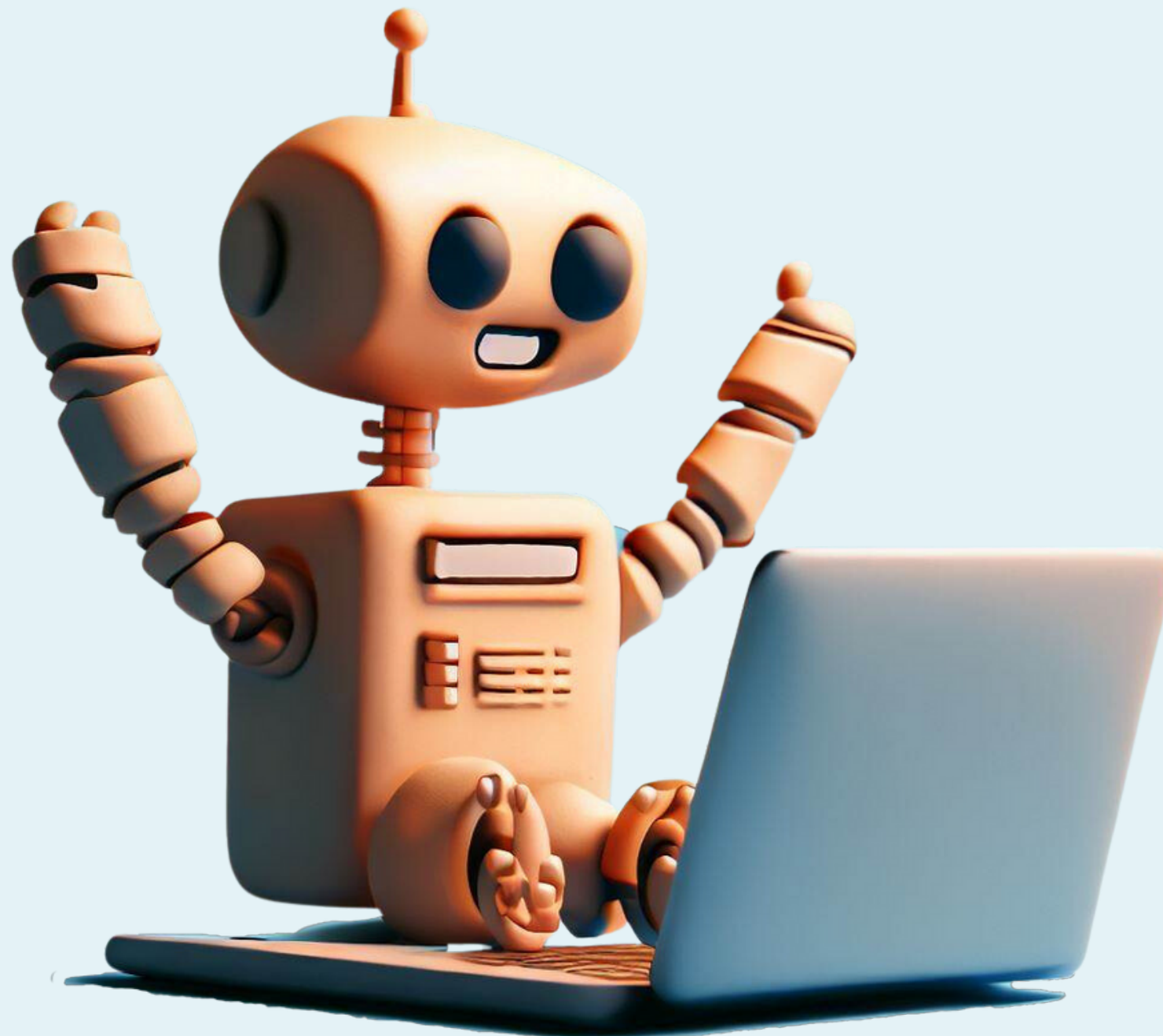
 Webinar **Microsoft Teams**

 Host **Av. Vitacura 6844 , Vitacura**

Colabora



Organiza





PROMPTING AVANZADO CON MODELOS GPT



HERNAN OLAVE

CINO Wisely

hernan.olave@wisely.cl

[linkedin.com/in/holave-bugini](https://www.linkedin.com/in/holave-bugini)

INDICE

- **Charla**

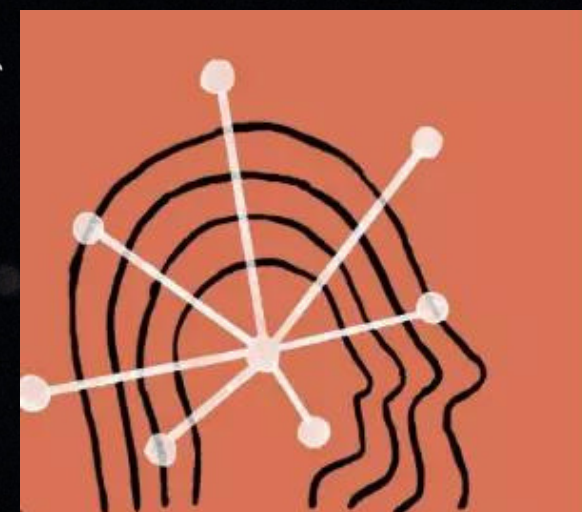
- Large Language Model (LLM)
- Que es una prompt
- Parametros (temperature, top_p, etc.)
- Tecnicas: Zero Shot, Few Shot, CoT y Self-Consistency

- **Workshop (python)**

- Prompt simples y Parametros
- Tecnicas: Zero Shot, Few Shot, CoT y Self-Consistency
- Optimizador de Prompts

LARGE LANGUAGE MODEL (LLM)

Un LLM es un tipo de modelo de AI que se centra en procesar y **generar lenguaje similar al humano**. Los LLMs se entrenan con grandes cantidades de datos de texto y utilizan técnicas de Deep Learning, para comprender y generar **respuestas relevantes en función del contexto**.



ANTHROPIC
CLAUDE 2



GPT

Generative Pre-trained Transformer, es una serie de LLMs desarrollados por OpenAI. Los modelos GPT se basan en la arquitectura Transformer(*), que utiliza **mecanismos de autoatención** para procesar y generar texto.



(*) Attention Is All You Need: <https://arxiv.org/pdf/1706.03762.pdf>

SELF ATTENTION

La auto-atención es un mecanismo que pondera y agrega diferentes partes de una secuencia de entrada para producir una nueva representación, enfatizando los tokens relevantes para el contexto de cada token.

As aliens entered our planet

Fuente: <https://www.youtube.com/watch?v=4Bdc55j80l8>

TOKENS

Tokens	Characters
163	381

Ejemplo: Cálculo del factorial de un número
Código en Python

```
def factorial(n):  
    if n == 0:  
        return 1  
    return n * factorial(n-1)
```

Explicación

La función factorial utiliza recursión para calcular el factorial de un número entero n . La base de la recursión es $n=0$, donde el factorial es 1. Para $n>0$, el factorial se calcula como $n \times \text{factorial}(n-1)$.

TEXT TOKEN IDS

TOKENS

Tokens

163

Characters

381


```
[36, 73, 18856, 78, 25, 327, 6557, 75, 3129, 78, 1619, 1109, 5132, 390, 555, 299, 21356, 647, 78, 198, 34, 10205, 12894, 78, 551, 11361, 198, 198, 4299, 1109, 5132, 7, 77, 2599, 198, 220, 220, 220, 611, 299, 6624, 657, 25, 198, 220, 220, 220, 220, 220, 220, 220, 1441, 352, 198, 220, 220, 220, 1441, 299, 1635, 1109, 5132, 7, 77, 12, 16, 8, 198, 198, 18438, 291, 32009, 18840, 198, 198, 14772, 1257, 979, 18840, 1109, 5132, 7736, 23638, 664, 1834, 72, 18840, 31215, 2386, 10440, 1288, 1109, 5132, 390, 555, 299, 21356, 647, 78, 920, 3529, 299, 77, 13, 4689, 2779, 390, 8591, 664, 1834, 72, 18840, 1658, 299, 28, 15, 77, 28, 15, 11, 288, 14378, 1288, 1109, 5132, 1658, 352, 13, 2547, 64, 299, 29, 15, 77, 29, 15, 11, 1288, 1109, 5132, 384, 5204, 64, 401, 78, 299, 12906, 22584, 5132, 7, 77, 14095, 16, 8, 77, 12906, 22584, 5132, 7, 77, 14095, 16, 737]
```

TEXT

TOKEN IDS

QUE ES UNA PROMPT

En el contexto de LLMs, una prompt es una **instrucción** o guía que se le entrega al modelo de inteligencia artificial para generar texto.

SYSTEM You will be provided with text, and your task is to translate it into emojis. Do not use any regular text. Do your best with emojis only.	USER Artificial intelligence is a technology with great promise.
	ASSISTANT 

OPENAI API

```
response = openai.ChatCompletion.create(  
    engine = "gpt-3.5-turbo",  
    messages = [  
        {"role": "system", "content": "You are a helpful assistant."},  
        {"role": "user", "content": "Hello, how are you?"},  
    ],  
    max_tokens = 600,  
    temperature = 0.7,  
    top_p = 1.0,  
    frequency_penalty = 0.0,  
    presence_penalty = 0.0,  
    stop = None  
)
```

PARAMETROS

Messages: Una lista de mensajes relacionadas a la conversación entre el asistente y el usuario. Debe contener los campos 'role' (system/user/assistant) y 'content'.

El 'system message' ayuda a establecer un determinado comportamiento al modelo (opcional).

Temperature (0.0 - 2.0) [Default: 1.0] Valores más altos como 0.8 harán la salida más aleatoria, mientras que valores más bajos como 0.2 la harán más enfocada y determinista.

PARAMETROS

Top P (0.0 - 1.0) [Default: 1] Una alternativa al parámetro 'temperature'. En donde el modelo considera los resultados de los tokens con probabilidad top_p. Por lo tanto, top_p = 0.1 significa que solo se consideran el top 10% de los resultados.

Presence Penalty (-2.0 - 2.0) [Default: 0] Los valores positivos penalizan los nuevos tokens en función de si aparecen en el texto, aumentando la probabilidad del modelo de abordar nuevos temas.

Frequency Penalty (-2.0 - 2.0) [Default: 0] Los valores positivos penalizan nuevos tokens en función de su frecuencia existente en el texto, disminuyendo la probabilidad de repetir la misma línea textualmente.

TECNICAS

Zero Shot: Capacidad del modelo para manejar tareas para las cuales no ha visto datos durante su entrenamiento.

USER Create a hello world python code but instead say "Wena Compadre" and use Chilean slang for naming methods and variables.

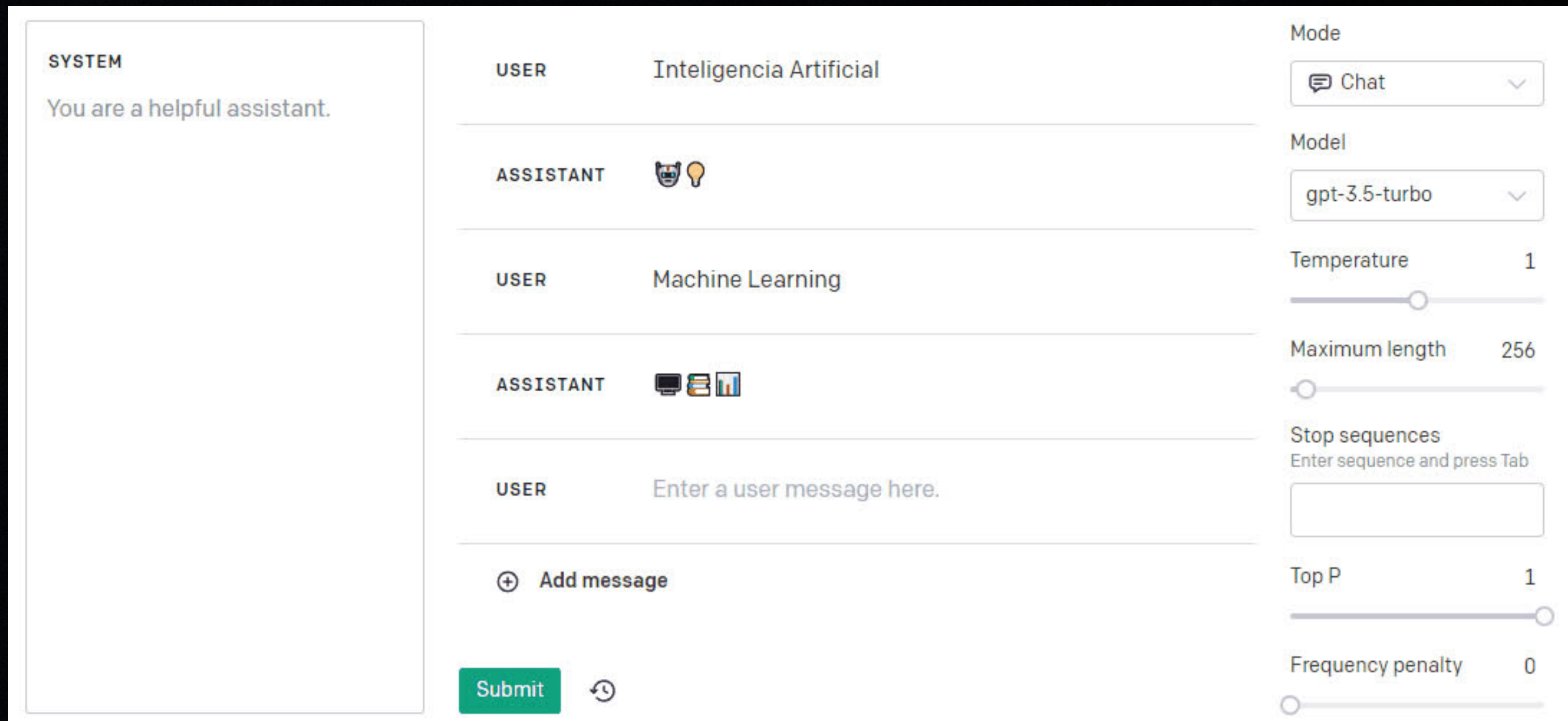
ASSISTANT Surely, here is how the Python code might look in a funny, "Chilean slang" way:

```
```python
def saludo_cumpita():
 mensaje_pulento = "Wena Compadre"
 return mensaje_pulento

print(saludo_cumpita())
```
```

TECNICAS

Few Shot: Técnica en la que se suministran ejemplos específicos de la tarea al modelo con el objetivo de orientar un rendimiento superior o un formato determinado.



The screenshot displays a chat interface with a system prompt and a few-shot example. The system prompt is "SYSTEM: You are a helpful assistant." The chat history shows a user message "USER: Inteligencia Artificial", an assistant response "ASSISTANT: 🐱💡", a user message "USER: Machine Learning", and an assistant response "ASSISTANT: 🖥️📄📊". The interface includes a "Submit" button and a settings panel on the right with parameters: Mode (Chat), Model (gpt-3.5-turbo), Temperature (1), Maximum length (256), Stop sequences (Enter sequence and press Tab), Top P (1), and Frequency penalty (0).

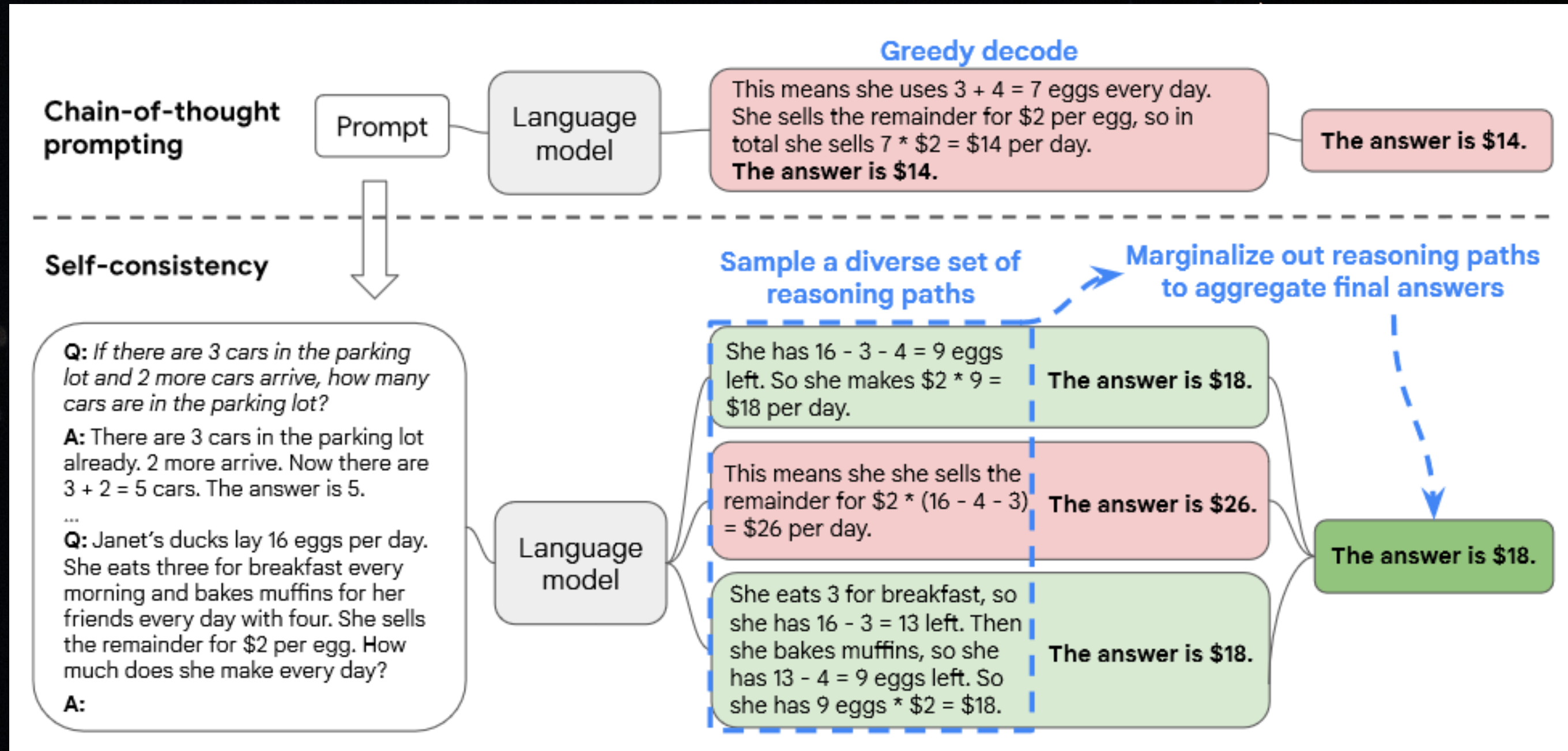
TECNICAS

Chain of Thought: Habilita capacidades de razonamiento complejo mediante etapas intermedias de razonamiento.

| Standard Prompting | Chain-of-Thought Prompting |
|---|---|
| <p>Model Input</p> <p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?</p> <p>A: The answer is 11.</p> <p>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</p> | <p>Model Input</p> <p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?</p> <p>A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.</p> <p>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</p> |
| <p>Model Output</p> <p>A: The answer is 27. ❌</p> | <p>Model Output</p> <p>A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅</p> |

TECNICAS

Self Consistency: Consiste en realizar multiples mustreos y elegir el mas consistente.



REPOSITORIO

<https://github.com/Wisely-ingenieria/ws-prompting-workshop>

EVENTOS PASADOS

<https://wisely.cl/meetup>



.ENV

OPENAI_API_TYPE = "openai"

OPENAI_API_KEY = "sk-

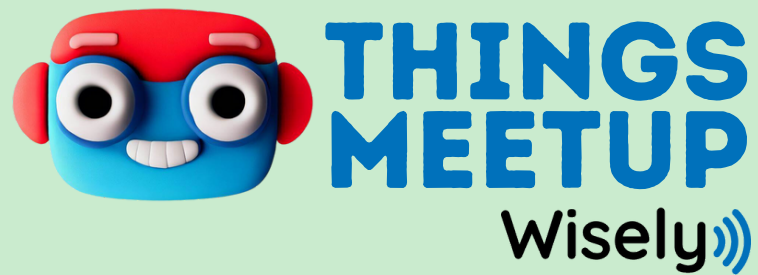
M1U7rrVarOaHlcj4MuzcT3BlbkFJUJ3wjfhIM6P5lBibQtm7"

OPENAI_EMBEDDING_MODEL = "text-embedding-ada-002"

OPENAI_GPT35_MODEL = "gpt-3.5-turbo"

OPENAI_GPT35_16K_MODEL = "gpt-3.5-turbo-16k"

OPENAI_GPT4_MODEL = "gpt-4"



CREACIÓN DE AGENTES CON PYTHON

Aprende a desarrollar agentes inteligentes utilizando Python nativo y bibliotecas populares de Inteligencia Artificial.



27 de Octubre



18 a 20 Horas



Webinar **Youtube Live**



Host **Por informar**


Organiza

Wisely

Colabora

 **Microsoft** 

Invita

 **El Club de la IA**

**GRACIAS POR
ACOMPañARNOS**

