Exploring Prompt Engineering with Python

Connect.it The Future of Python

Author: Eng. Carlos Andrés Sierra, M.Sc.
`carlos.andres.sierra.v@gmail.com`

Professor
Computer Engineer
Master on Computer Engineering

March 12th 2024

Outline

1. Welcome to Endava's Connect.IT
2. What are Large Language Models?
3. Prompt Engineering
   - Basic Techniques
4. Let's Code with Star Coder Model

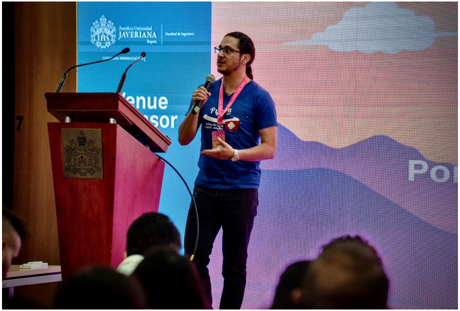
Outline

1. Welcome to Endava's Connect.IT
2. What are Large Language Models?
3. Prompt Engineering
   - Basic Techniques
4. Let's Code with Star Coder Model

Welcome!

Endava's Connect.IT is a series of events that bring together the best minds in tech to discuss the latest trends and innovations in the industry.

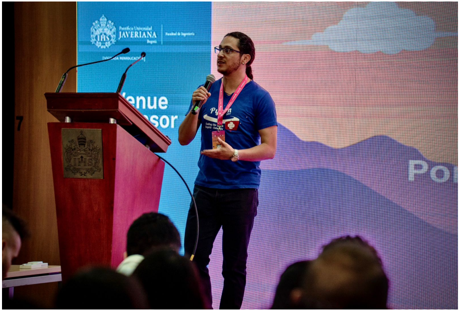
Meet the Speaker II



2016 ~

- *PyCon Colombia* and *Python Bogotá* **Co-organizer**.
- Former **Software Engineer** and **Technical Leader** of Machine Learning and Data Science.
- **Professor** at *UD Francisco José de Caldas* and **MLOps Engineer** at *KLYM*.
- **Speaker** at IEEE conferences, colleges, meetups, . . .

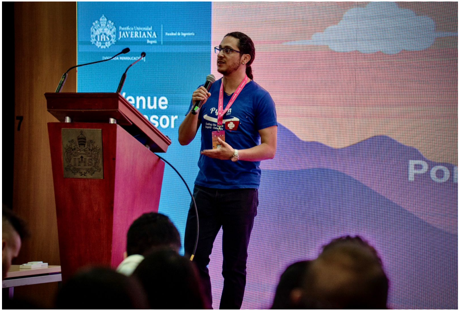
Meet the Speaker II



- *PyCon Colombia* and *Python Bogotá* **Co-organizer**.
- Former **Software Engineer** and **Technical Leader** of Machine Learning and Data Science.
- **Professor** at *UD Francisco José de Caldas* and **MLOps Engineer** at *KLYM*.
- **Speaker** at IEEE conferences, colleges, meetups, . . .

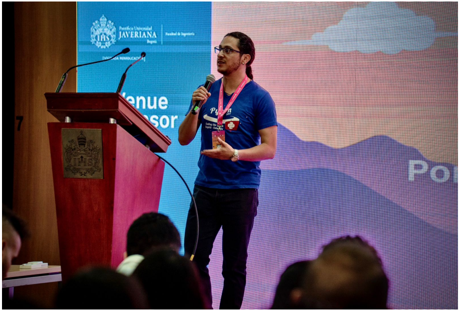
endava

Meet the Speaker II



- *PyCon Colombia* and *Python Bogotá* **Co-organizer**.
- Former **Software Engineer** and **Technical Leader** of Machine Learning and Data Science.
- **Professor** at *UD Francisco José de Caldas* and **MLOps Engineer** at *KLYM*.
- **Speaker** at IEEE conferences, colleges, meetups, . . .

Meet the Speaker II



- *PyCon Colombia* and *Python Bogotá* **Co-organizer**.
- Former **Software Engineer** and **Technical Leader** of Machine Learning and Data Science.
- **Professor** at *UD Francisco José de Caldas* and **MLOps Engineer** at *KLYM*.
- **Speaker** at IEEE conferences, colleges, meetups, . . .

Outline

1. Welcome to Endava's Connect.IT
2. What are Large Language Models?
3. Prompt Engineering
   - Basic Techniques
4. Let's Code with Star Coder Model

Neural Networks and Natural Language Processing

80's → > 2000

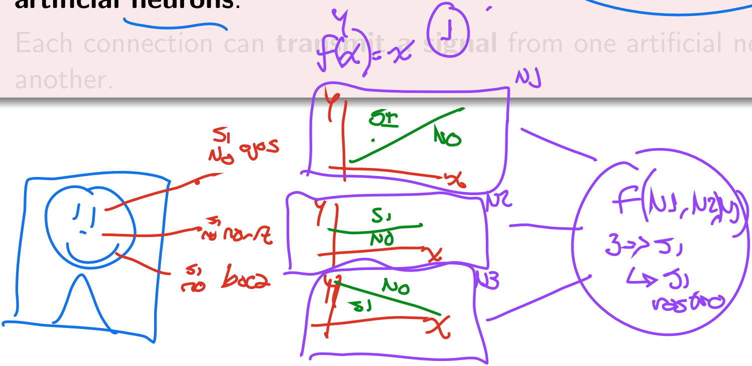
Neural Networks

- **Artificial Neural Networks** are a computational model inspired by the human brain.
- They are based on a collection of connected units or nodes called **artificial neurons**.
- Each connection can **transmit a signal** from one artificial neuron to another.

Neural Networks and Natural Language Processing

Neural Networks

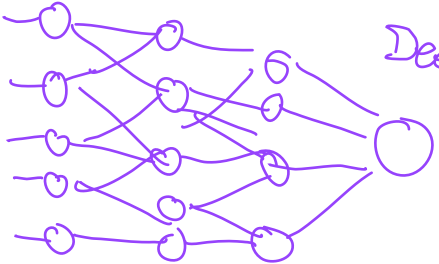
- **Artificial Neural Networks** are a computational model inspired by the human brain.
- They are based on a collection of connected units or nodes called **artificial neurons**.
- Each connection can **transmit a signal** from one artificial neuron to another.



Neural Networks and Natural Language Processing

Neural Networks

- **Artificial Neural Networks** are a computational model inspired by the human brain.
- They are based on a collection of connected units or nodes called **artificial neurons**.
- Each connection can **transmit a signal** from one artificial neuron to another.



Deep Learning

Neural Networks and Natural Language Processing

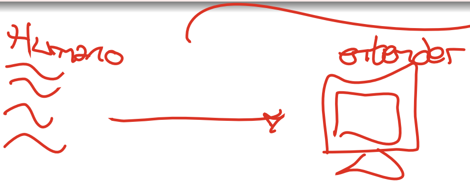
60's → >2008

Natural Language Processing

- **Natural Language Processing** (NLP) is a subfield of linguistics, computer science, and artificial intelligence.
- It is concerned with the interactions between computers and human language.

Sujeto + Verbo + Complemento

- In particular, the programming of computers to **process and analyze** large amounts of natural language data.



Neural Networks and Natural Language Processing

Natural Language Processing

- **Natural Language Processing** (NLP) is a subfield of linguistics, computer science, and artificial intelligence.
- It is concerned with the interactions between computers and human language.
- In particular, the programming of computers to **process and analyze** large amounts of natural language data.

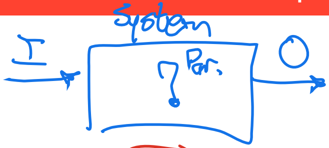
Andrej Kaparty < ex-Tesla
OpenAI (ChatGPT)

Neural Networks and Natural Language Processing

Natural Language Processing

- **Natural Language Processing** (NLP) is a subfield of linguistics, computer science, and artificial intelligence.
- It is concerned with the interactions between computers and human language.
- In particular, the programming of computers to **process and analyze** large amounts of natural language data.

Inputs, Parameters, Outputs



token token2

Las naranjas som amarillas

In the context of LLMs, we can say that: 3 trillions

- **Inputs** are sequences of tokens.
- **Parameters** are weights and biases. 150 billions
- **Outputs** are sequences of tokens.

weight bias

$$f(x) = \hat{a}x + b$$

Las frutas co....

→ GPT-4

3 weeks ≈ 25000 gpu

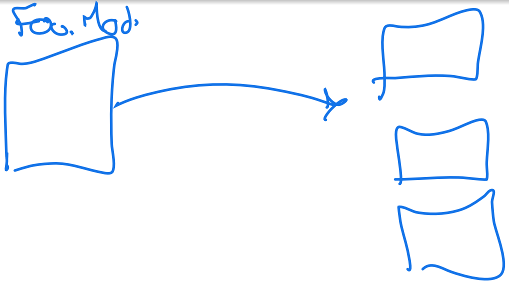
$ 190 millions USD

Foundational Models

What are foundational models?

- **Foundational models** are the first and largest language models.
- They are pre-trained on large corpora of text.
- They could be fine-tuned on specific tasks.

Few days



Foundational Models

What are foundational models?

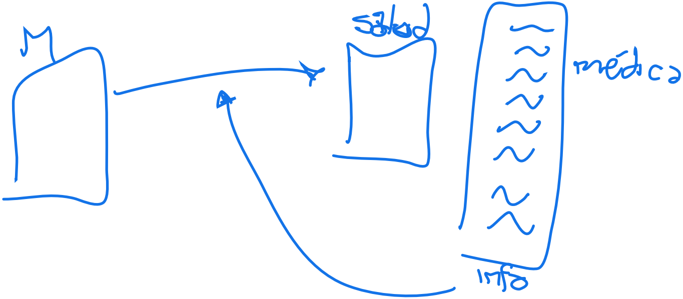
- **Foundational models** are the first and largest language models.
- They are pre-trained on large corpora of text.
- They could be fine-tuned on specific tasks.

endava

Foundational Models

What are foundational models?

- **Foundational models** are the first and largest language models.
- They are pre-trained on large corpora of text.
- They could be fine-tuned on specific tasks.



endava

Outline

1 Welcome to Endava's Connect.IT

2 What are Large Language Models?

3 Prompt Engineering
- Basic Techniques

4 Let's Code with Star Coder Model

Basic Concepts

What is Prompt Engineering?

- **Prompt engineering** is the process of designing and constructing prompts for large language models.
- Make the model more effective and efficient, also avoiding bias and unwanted outputs. . . or hallucinations.

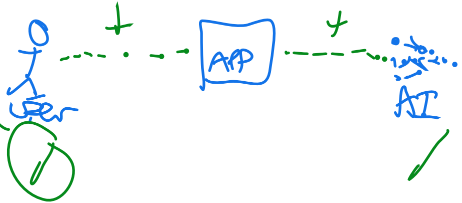
Input
- question
- request

Basic Concepts

What is Prompt Engineering?

- **Prompt engineering** is the process of designing and constructing prompts for large language models.
- Make the model more effective and efficient, also avoiding bias and unwanted outputs. . . or hallucinations.

wrong response



Outline

1. Welcome to Endava's Connect.IT

2. What are Large Language Models?

3. Prompt Engineering
   - Basic Techniques

4. Let's Code with Star Coder Model

Chain of Thought Prompting

Keypoints

- **Chain of Thought Prompting** is a technique that involves prompting a model with a series of questions.
- The answers to the questions are used as inputs to subsequent questions.
- This technique is useful for eliciting detailed and coherent responses.

Q. código Python sumar?

A.
```
def sum(n1, n2):
    return n1+n2
```

Q. código python restar?

A.
```
def rest(n1, n2):
    return n1-n2
```

código Python multiplicar?

ENTER

↳
```
def mult(n1, n2):
    return n1*n2
```

Zero-Shot Prompting

- **Zero-shot prompting** is a technique that involves prompting a model with a single question.
- The model is expected to generate a coherent and relevant response.

¿ _ _ _ ? → ____

Few-Shot Prompting

- **Few-shot prompting** is a technique that involves prompting a model with a small number of examples.
- This technique is useful for eliciting specific and targeted responses.

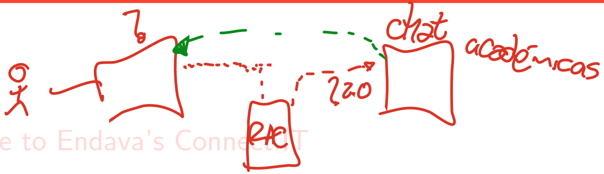
Código Python sumar?

```
def sum(n1: int, n2: int) -> int:
    """
    """
    return n1+n2
```

Código Python restar?

```
def rest(n1: int, n2: int)->int:
    """
    """
    return n1-n2
```

Outline



1. Welcome to Endava's ConnectIT

2. What are Large Language Models?

3. Prompt Engineering
   - Basic Techniques

4. Let's Code with Star Coder Model

Star Coder from Hugging Faces

- **Star Coder** is a large language model developed by Hugging Face.
- It is pre-trained on a large corpus of code. Also, it is fine-tuned on a large dataset of code.
- Star Coder 2 support more than 80 programming languages.
- The largest version of this *family* has 15B parameters. There are another versions of 3B and 7B parameters.
- Those versions had been trained using between three and four trillion of tokens. The Stack is used as main source of data.

Setup Python & Any IDE

- Here you need to have Python installed and VS Code and Jupyter Notebook extension.
- It works at any python version, but it is recommended to use python 3.10 or newer.
- As IDE, it is recommended to use VS Code. It is FOSS, and have a lot of useful extensions.
- The required packages are:

```
# no one
```

Open Assistant's Dataset. . . for your own chat assistant

Open Assistant's dataset is a large dataset of code that useful to make fine-tunning on Star Coder. It is available on Hugging Face's model hub.

Open Assistant's has more than 40000 conversations, switching roles between human and assistant.

Open Assistant's has a permissive licence and had been totally produced by humans.

Link: https://huggingface.co/datasets/OpenAssistant/oasst1

Steps to Make the Tests

1. Create an account at Hugging Face.
2. Create a bearer token, and install StarCoderEX extension in VS Code.
3. Open in the web browser the Code Llama Playground.
4. It is time to interact with the code generators.

Steps to Make the Tests

1. Create an account at Hugging Face.
2. Create a bearer token, and install StarCoderEX extension in VS Code.
3. Open in the web browser the Code Llama Playground.
4. It is time to interact with the code generators.

Steps to Make the Tests

1. Create an account at Hugging Face.
2. Create a bearer token, and install StarCoderEX extension in VS Code.
3. Open in the web browser the Code Llama Playground.
4. It is time to interact with the code generators.

endava

Steps to Make the Tests

1. Create an account at Hugging Face.
2. Create a bearer token, and install StarCoderEX extension in VS Code.
3. Open in the web browser the Code Llama Playground.
4. It is time to interact with the code generators.

Let's Give a Shoot. . . or a few

Please, try next prompts:

1. *"Create in Python a function to sum two numbers."*
2. *"Create in Python a function called my_sum_2 to calculate the sum of two integer parameters called num_1 and num_2, adding doctring to the function."*

Let’s Give a Shoot. . . or a few

Create a multiplication function, but with another examples.

Q. Create in Python a function to calculate the sum of two numbers.

A.

```
def my_own_sum(num_1: int, num_2: int) -> int:
    """This method sums two numbers.
    Parameters:
    - num_1 (int): The first number to be added.
    - num_2 (int): The second number to be added.
    Returns:
    - int: The sum of the two numbers.
    """
    return num_1 + num_2
```

Let's Give a Shoot. . . or a few

Create a multiplication function, but with another examples.

Q. Create in Python a function to calculate the subtraction of two numbers.

```
A.  def my_own_subtract(num_1: int, num_2: int) -> int:
        """This method subtracts two numbers.
        Parameters:
        - num_1(int): The first number to be subtracted.
        - num_2(int): The second number to be subtracted.
        Returns:
        - int: The subtract of the two numbers.
        """
        return num_1 - num_2
```

Outline

1. Welcome to Endava's Connect.IT

2. What are Large Language Models?

3. Prompt Engineering
   - Basic Techniques

4. Let's Code with Star Coder Model

Thanks!!

Linkedin: *casierrav*