

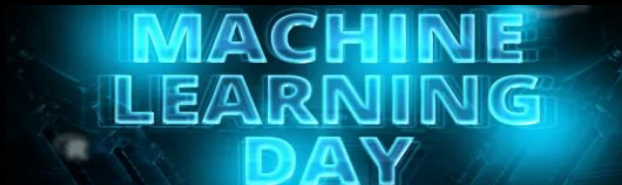
# Projeto final: PLN na prática



**Prof. Dr. Diego Renan Bruno**

Education Tech Lead na DIO

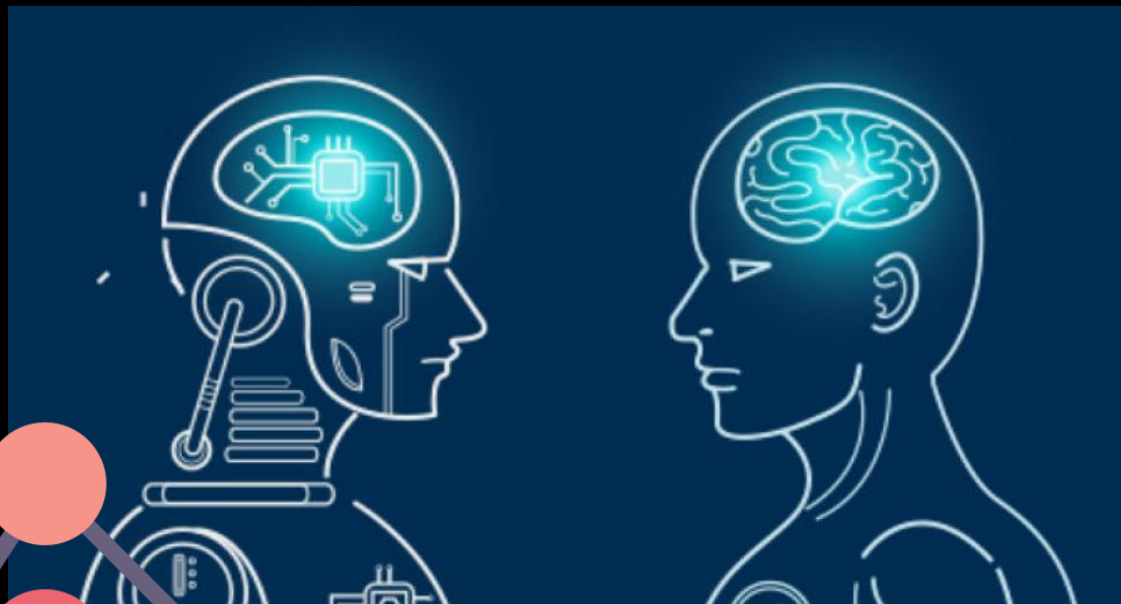
Doutor em Robótica e *Machine Learning* pelo ICMC-USP



OpenCV

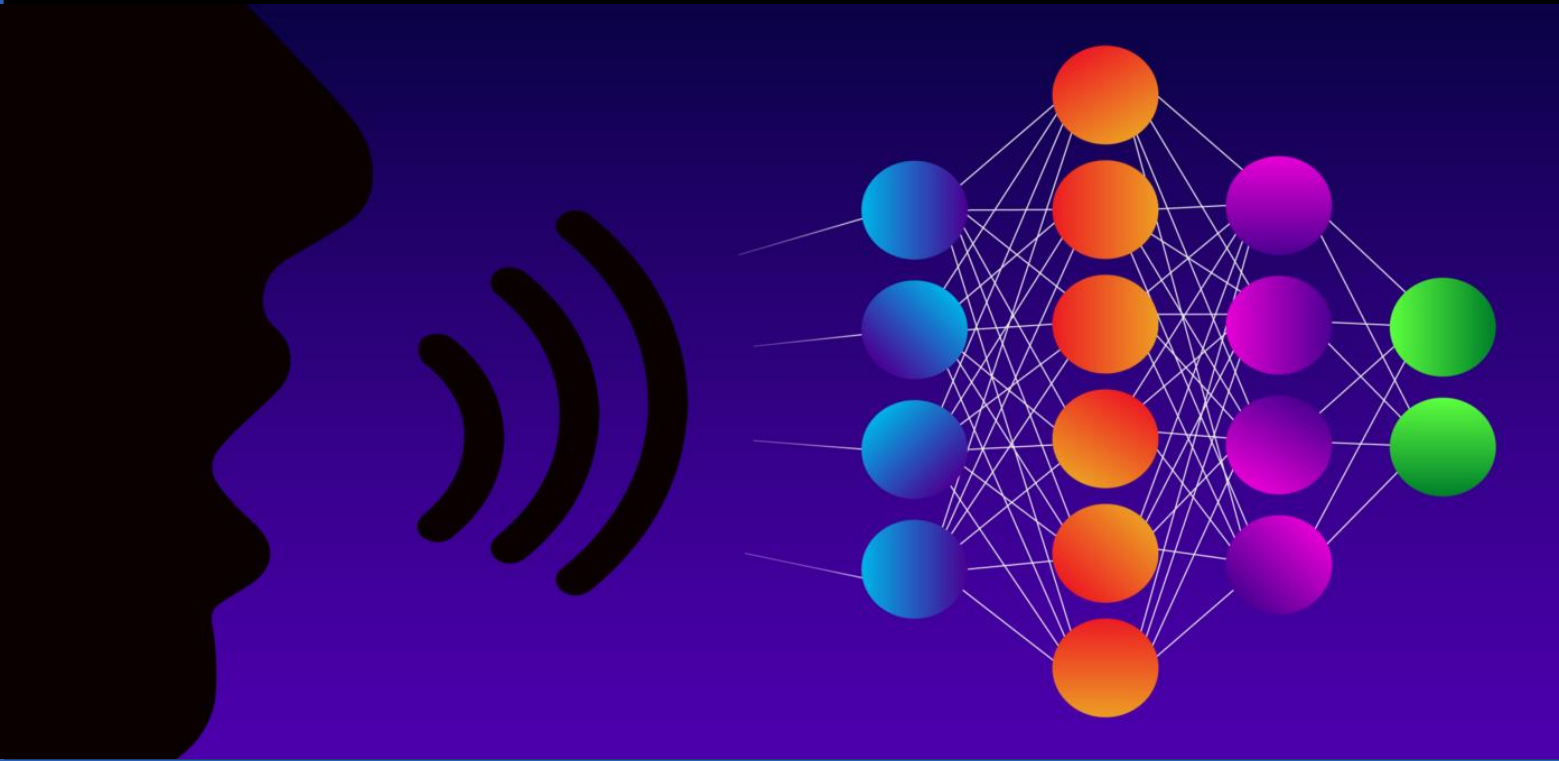
# Assistente virtual

*Machine Learning*



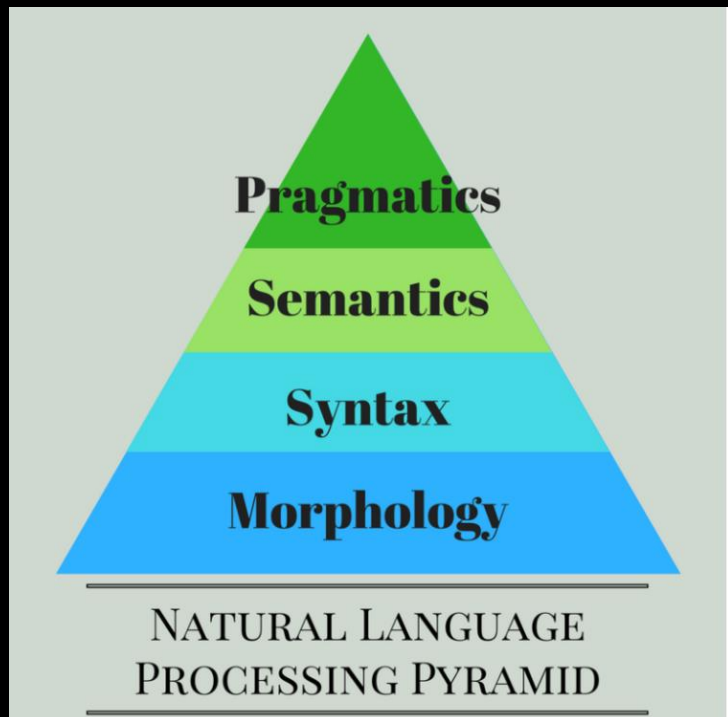
# Deep Learning para PLN

→ Sistema de interpretação de linguagem natural.



# Níveis do processamento

→ Sistemas: recomendação, comando por voz, chatbots...



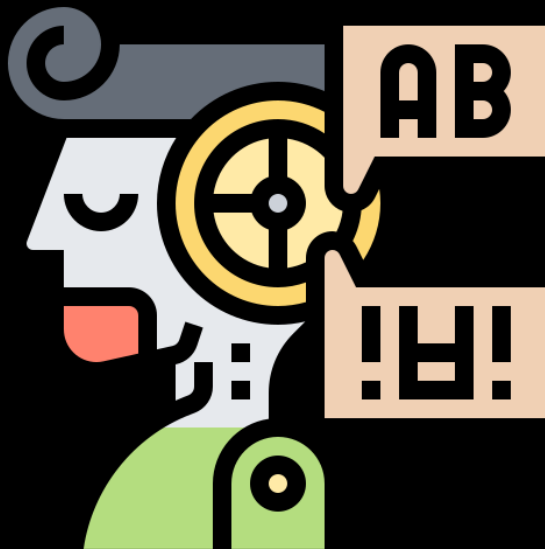
**Pragmática** é o ramo da linguística que estuda a linguagem no contexto de seu uso na comunicação.

Relacionado com à análise detalhada da composição, derivação, flexão das palavras e de seus processos de formação.

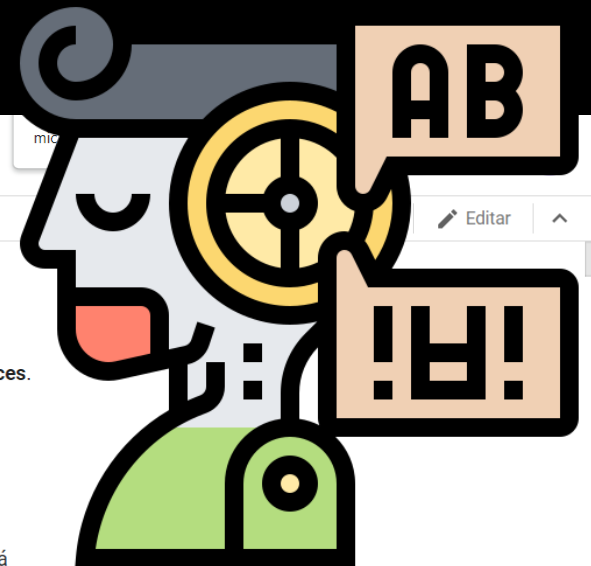
# Deep Learning para PLN

Os sistemas de NLP permitem que a **tecnologia usada não apenas entenda o significado** literal de cada palavra que está sendo dita, como também considere aspectos como:

- Contexto da conversa;
- Significados sintáticos e semânticos;
- Interprete os textos;
- Análise sentimentos e mais.



# Deep Learning para PLN



jupyter-bert-classificacao.ipynb ☆

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Não é possível salvar as alterações

+ Código + Texto Copiar para o Drive

## Classificação com BERT

Neste *notebook* você verá um exemplo de como usar o [BERT](#) para tarefa de classificação, usando a biblioteca **Transformers** do **Hugging Faces**.

Fontes:

- [BERT Fine-Tuning Tutorial with PyTorch](#).
- [Hugging Faces - Github](#) e [Hugging Faces - site](#)

Primeiro, vamos verificar se temos GPU disponível para nossa execução. Não se preocupe caso não possua GPU, apenas o treinamento será mais demorado.

Caso você esteja executando no Colab, acesse: Edit → Notebook Settings → Hardware accelerator → (GPU)

```
import torch

# Verificar se tem GPU disponível
if torch.cuda.is_available():

    # Informa PyTorch para usar GPU
    device = torch.device("cuda")
```

# Obrigado!

*Machine Learning*

Prof. Dr. Diego Bruno



OpenCV