# Language Models are Few-Shot Learners

GPT-3

# 1. Questions

1. **Explicação de conceitos importantes do artigo**
2. **A contribuição do artigo**
3. **Resultados interessantes/inesperados**
4. ~~Uma dúvida "básica" que você ou os colegas possam ter~~
5. ~~Um tópico "avançado" para discutirmos~~

# 2. Explicação de conceitos importantes do artigo

1. **Language models**: A type of machine learning model that is trained on large amounts of text data and can generate coherent and fluent text.
2. **Zero/One/Few-shot learning**: A type of machine learning in which a model is trained to learn from zero, one and a few number of examples (typically less than 100).
    1. **zero-shot**: task description + 0 example + prompt
    2. **one-shot**: task description + 1 example + prompt
    3. **few-shot**: task description + 1+ example + prompt
3. **Transfer learning**: A machine learning technique in which a model is trained on a large dataset to learn general patterns, and then fine-tuned on a smaller dataset to learn specific patterns.

# 3. Contribuição

1. by the time was the largest LLM ever created 175B params (size does matter)
2. introduced the GPT-3 (Generative Pre-trained Transformer 3) language model, which demonstrated impressive few-shot learning capabilities. GPT-3 was trained on a massive dataset of text and could perform a variety of natural language processing tasks with only a few examples of each task.

# 3. Resultados interessantes/inesperados

1. The authors found that GPT-3 could perform well on a wide range of language tasks without any task-specific training, including question answering, language translation, and even programming.
2. GPT and GPT-2 required more task-specific training data to achieve good performance in few-shot learning (size does matter again)

# 4. Uma dúvida "básica" que você ou os colegas possam ter

1. What is a language model and how does it work?
2. How Transformer architecture works?

# 4.1 What is a language model and how does it work?

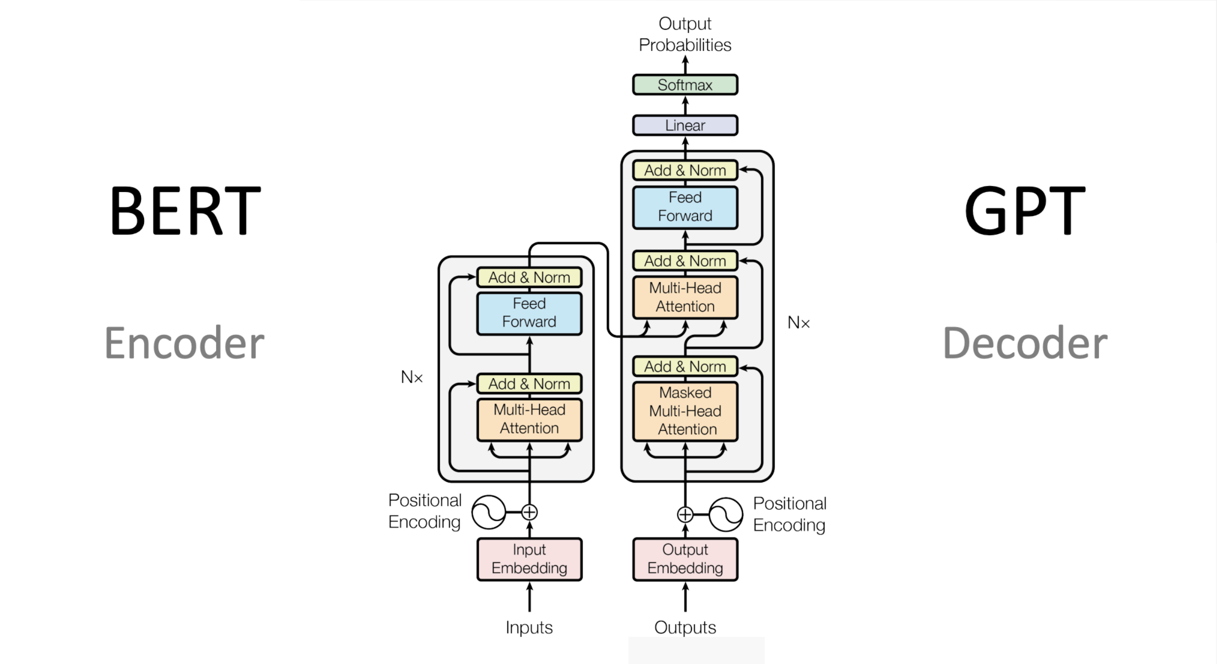
> “ A language model is a probability distribution over sequences of words in a language. It is a statistical model that assigns a probability to every possible sequence of words in a given language.
>
> The goal of a language model is to predict the probability of a sequence of words given the preceding words in the sequence.
>
> This is done by learning the conditional probability distribution of each word in the sequence given the previous words. ”

We can build a LM with RNN, CNN, LSTM, BERT...

# 4.2 How Transformer architecture works?

The Transformer is a deep learning architecture that was introduced in a 2017 paper called "Attention Is All You Need." It aims to solve sequence-to-sequence tasks while handling long-range dependencies with ease.

The Transformer architecture consists of two main components: **Encoder** and **Decoder**



# 4.2.1 Encoder (BERT)

> “ The encoder takes the input sequence of words and generates a sequence of hidden representations that capture the meaning of the input. It consists of several identical layers, each of which performs two operations:
>
> - **Self-attention**: This operation allows the model to weigh the importance of each word in the input sequence when generating the hidden representation for each word. Each word is assigned a weight based on how similar it is to the other words in the sequence. This allows the model to focus on the most relevant words for each task.
> - **Feedforward**: This operation applies a non-linear transformation to each hidden representation to further capture the meaning of the input sequence. ”

## 4.2.2 Decoder (GPT)

> The decoder takes the hidden representations generated by the encoder and generates the output sequence of words. Like the encoder, it consists of several identical layers, each of which performs two operations:
>
> - **Masked self-attention**: This operation is similar to self-attention in the encoder, but it's applied in a masked way to prevent the model from looking ahead and cheating by using future words to generate the output.
> - **Cross-attention**: This operation allows the model to weigh the importance of each hidden representation generated by the encoder when generating the output sequence. It helps the model to align the input and output sequences and generate accurate translations or summaries.