Universidade Federal da Paraíba

Centro de Informática

Departamento de Informática

Aprendizado Profundo

Modelos de Linguagem de Larga Escala (LLMs)

(Material Baseado em @CodeEmporium)

Tiago Maritan

(tiago@ci.ufpb.br)

Modelos de Linguagem de Larga Escala (LLMs)

- Com a possibilidade de paralelização das Redes Transformers, o **tamanho dos modelos neurais, especialmente os modelos de linguagem,** tem disparado nos últimos anos.

- Crescimento de até **5000x** no tamanho dos modelos nos últimos anos.

Modelos de Linguagem de Larga Escala (LLMs)

Model	Organization	Date	Size (# params)
ELMo	AI2	Feb 2018	94,000,000
GPT	OpenAI	Jun 2018	110,000,000
BERT	Google	Oct 2018	340,000,000
XLM	Facebook	Jan 2019	655,000,000
GPT-2	OpenAI	Mar 2019	1,500,000,000
RoBERTa	Facebook	Jul 2019	355,000,000
Megatron-LM	NVIDIA	Sep 2019	8,300,000,000
T5	Google	Oct 2019	11,000,000,000
Turing-NLG	Microsoft	Feb 2020	17,000,000,000
GPT-3	OpenAI	May 2020	175,000,000,000
Megatron-Turing NLG	Microsoft, NVIDIA	Oct 2021	530,000,000,000
Gopher	DeepMind	Dec 2021	280,000,000,000

Fonte: stanford-cs324.github.io/winter2022/lectures/introduction

Modelos de Linguagem de Larga Escala (LLMs)

Model	Organization	Date	Size (# params)
ELMo	AI2	Feb 2018	94,000,000
GPT	OpenAI	Jun 2018	110,000,000
BERT	Google	Oct 2018	340,000,000
XLM	Facebook	Jan 2019	655,000,000
GPT-2	OpenAI	Mar 2019	1,500,000,000
RoBERTa	Facebook	Jul 2019	355,000,000
Megatron-LM	NVIDIA	Sep 2019	8,300,000,000
T5	Google	Oct 2019	11,000,000,000
Turing-NLG	Microsoft	Feb 2020	17,000,000,000
GPT-3	OpenAI	May 2020	175,000,000,000
Megatron-Turing NLG	Microsoft, NVIDIA	Oct 2021	530,000,000,000
Gopher	DeepMind	Dec 2021	280,000,000,000

Fonte: stanford-cs324.github.io/winter2022/lectures/introduction

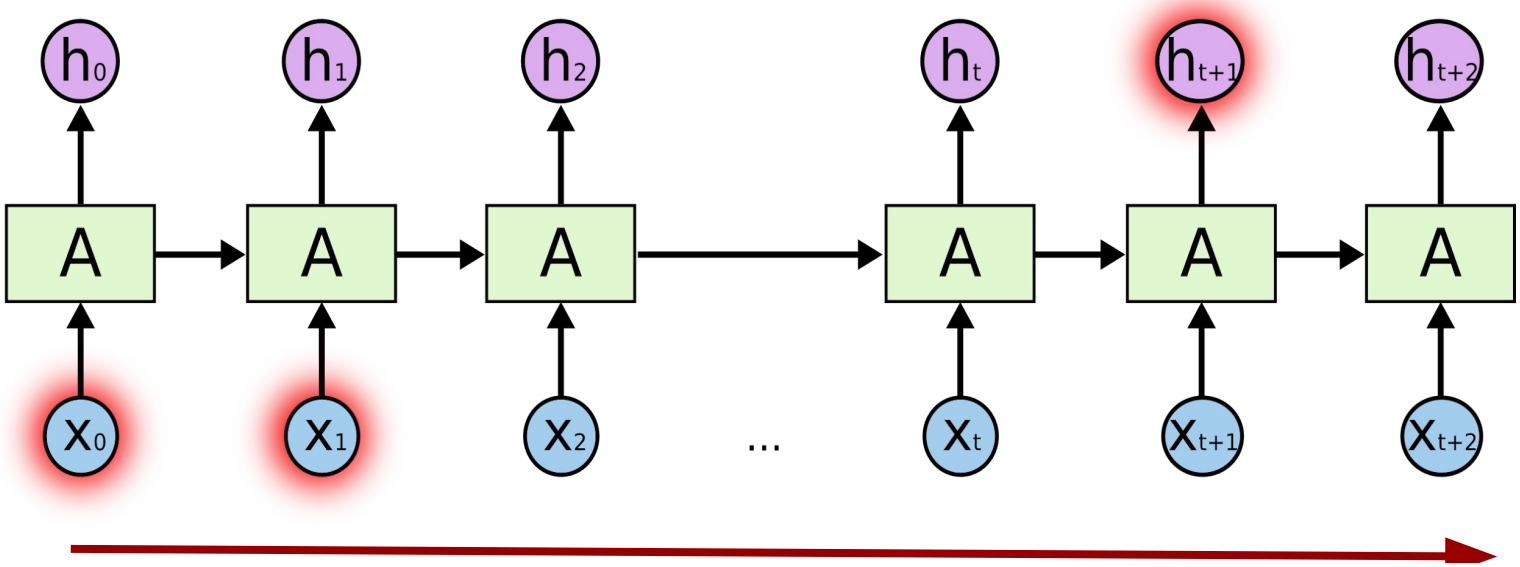
BERT- Bidirecional Encoders for Representation from Transformers

Artigo: https://arxiv.org/pdf/1810.04805.pdf

Motivação

- Modelos de linguagem são geralmente treinados de forma unidirecional

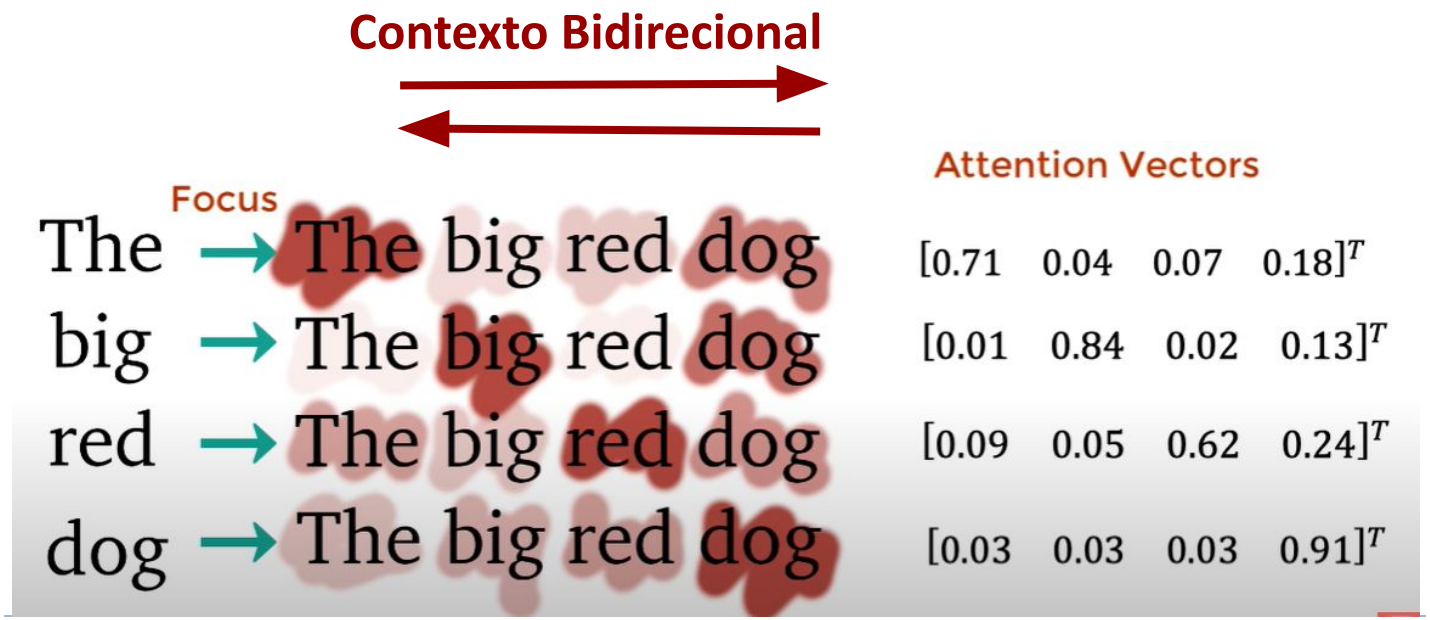
h_{t+1} é gerado a partir de $x_0 \ldots x_{t+1}; h_0 \ldots h_t$



Informações de contexto geralmente da esquerda para direita

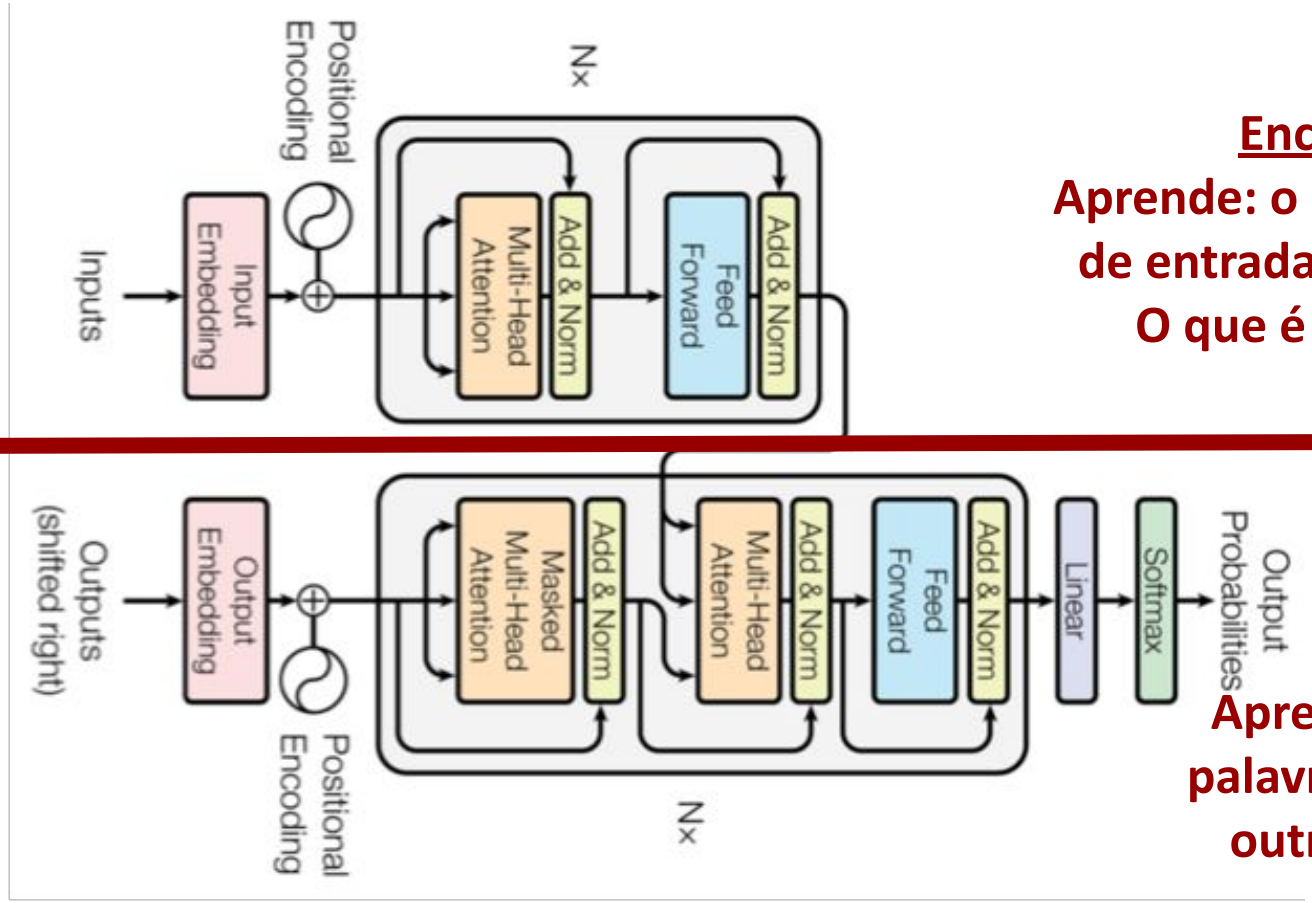
BERT

- BERT: Projetado para **pré-treinar** representações **bidirecionais** de textos não rotulados
  - Transformers são capazes de capturar de aprender contextos em **ambas as direções** simultaneamente
  - Mecanismo de **self-attention**



Transformer

- **Encoder** e **Decoder** possuem papéis distintos!

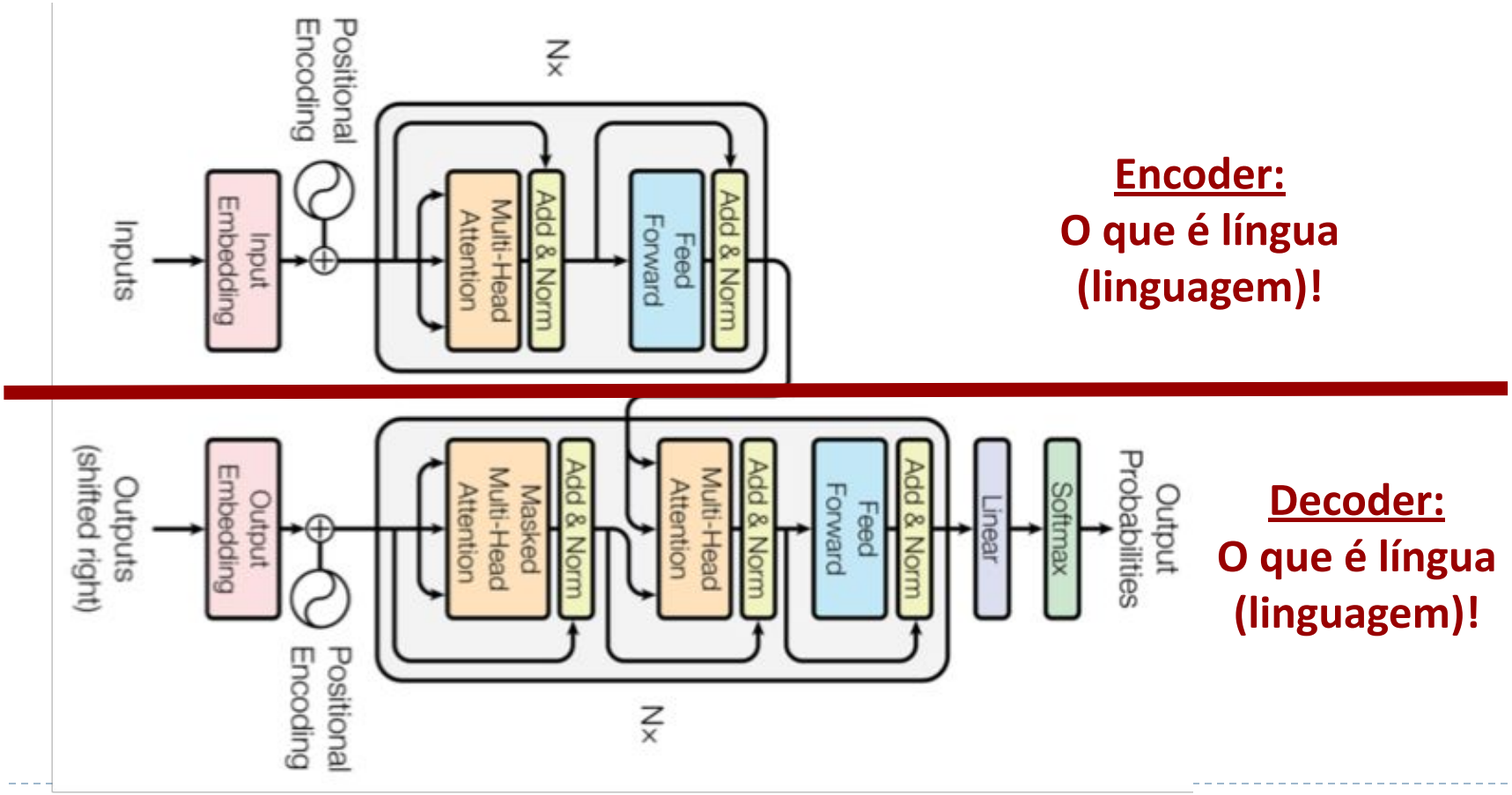


Encoder:
Aprende: o que é a língua de entrada (ex: Inglês)? O que é contexto?

Decoder:
Aprende: Como mapear palavras de uma língua na outra? (ex: Inglês para Francês)?

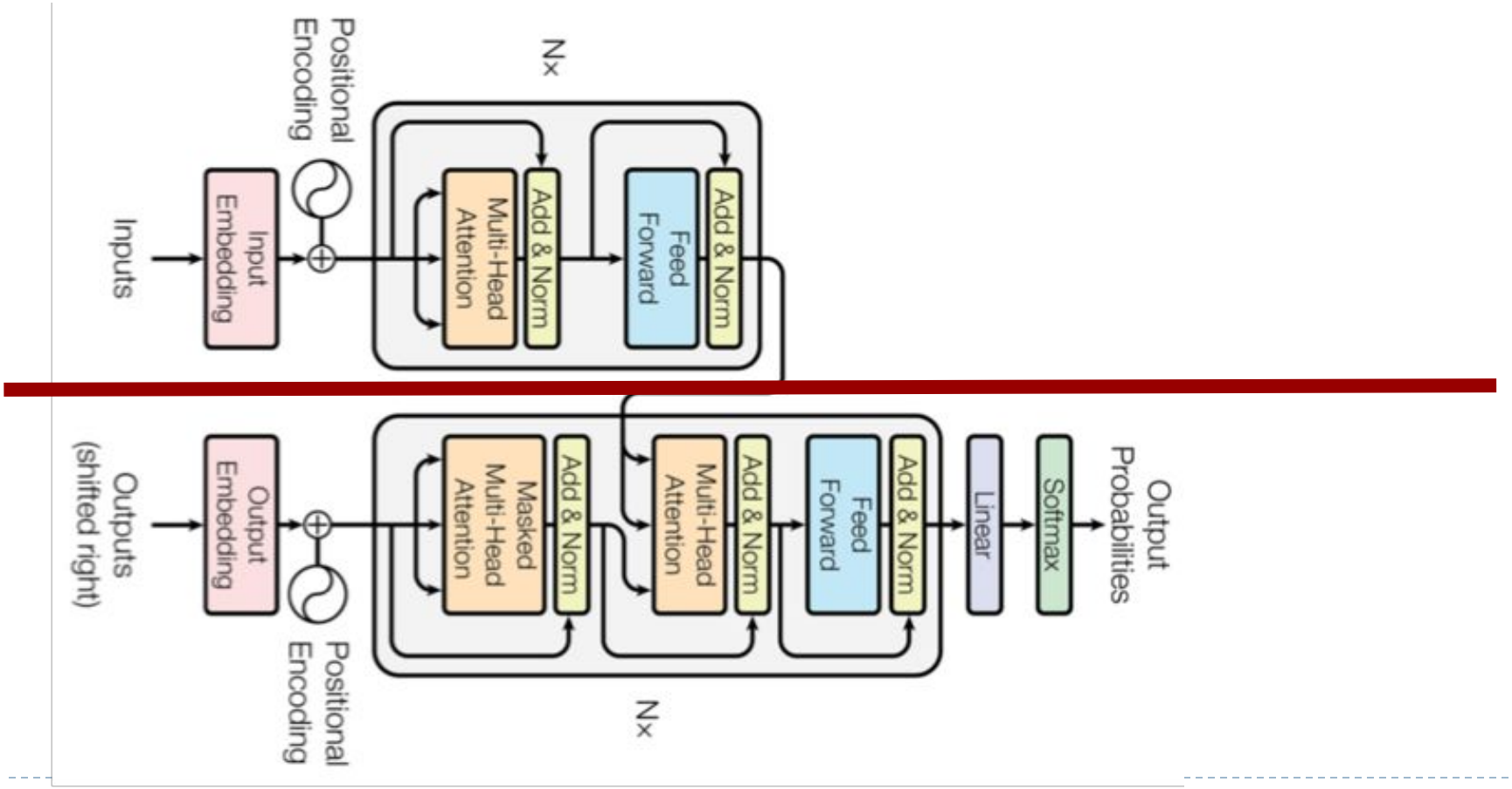
Transformer

- **Contudo, ambos aprendem separadamente a compreender língua (linguagem)!!!**



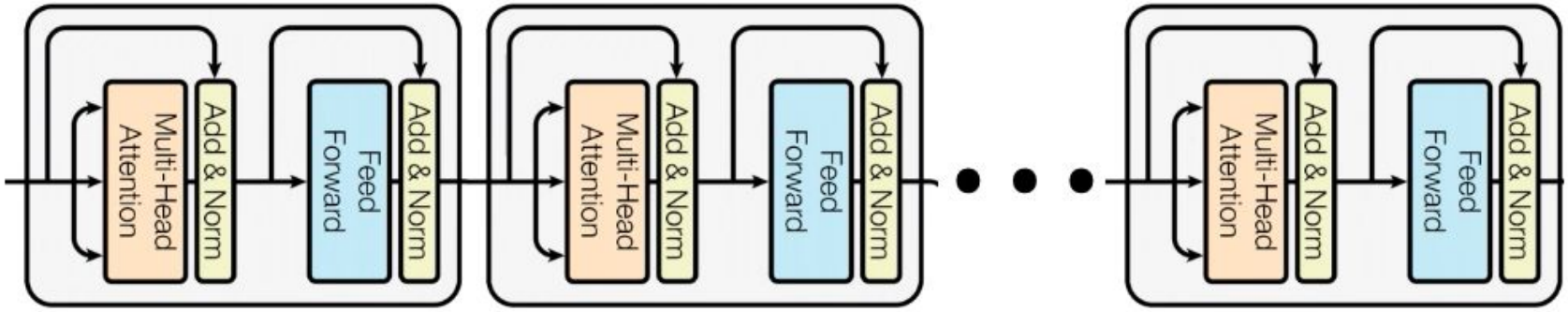
Transformer

- **É possível, portanto, separá-los e construir sistemas que compreendam o que é língua (linguagem)!**

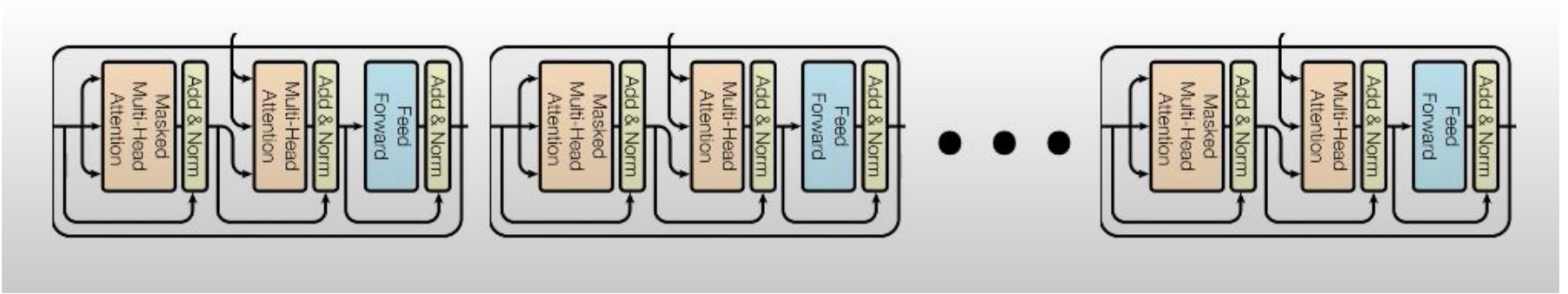


Transformers:

- **BERT:** Encadeia vários **"Encoders"** (Transformers)



- **GPT:** Encadeia vários **"Decoders"** (Transformers)

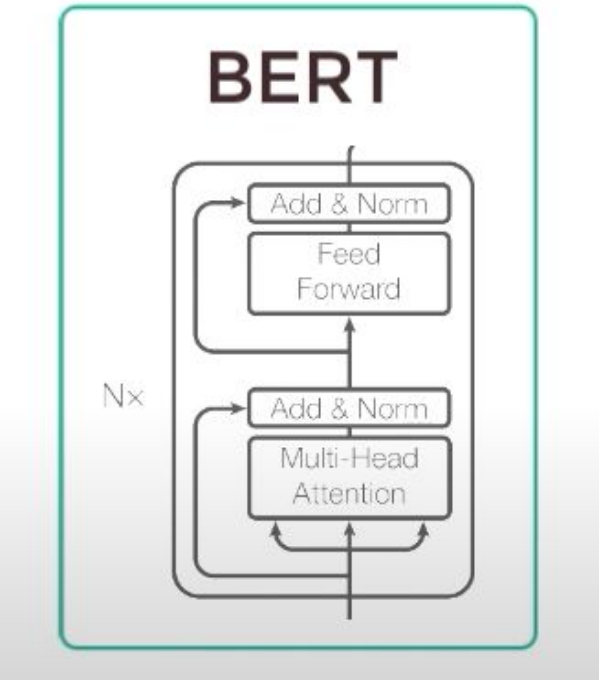


BERT - Ideia Geral

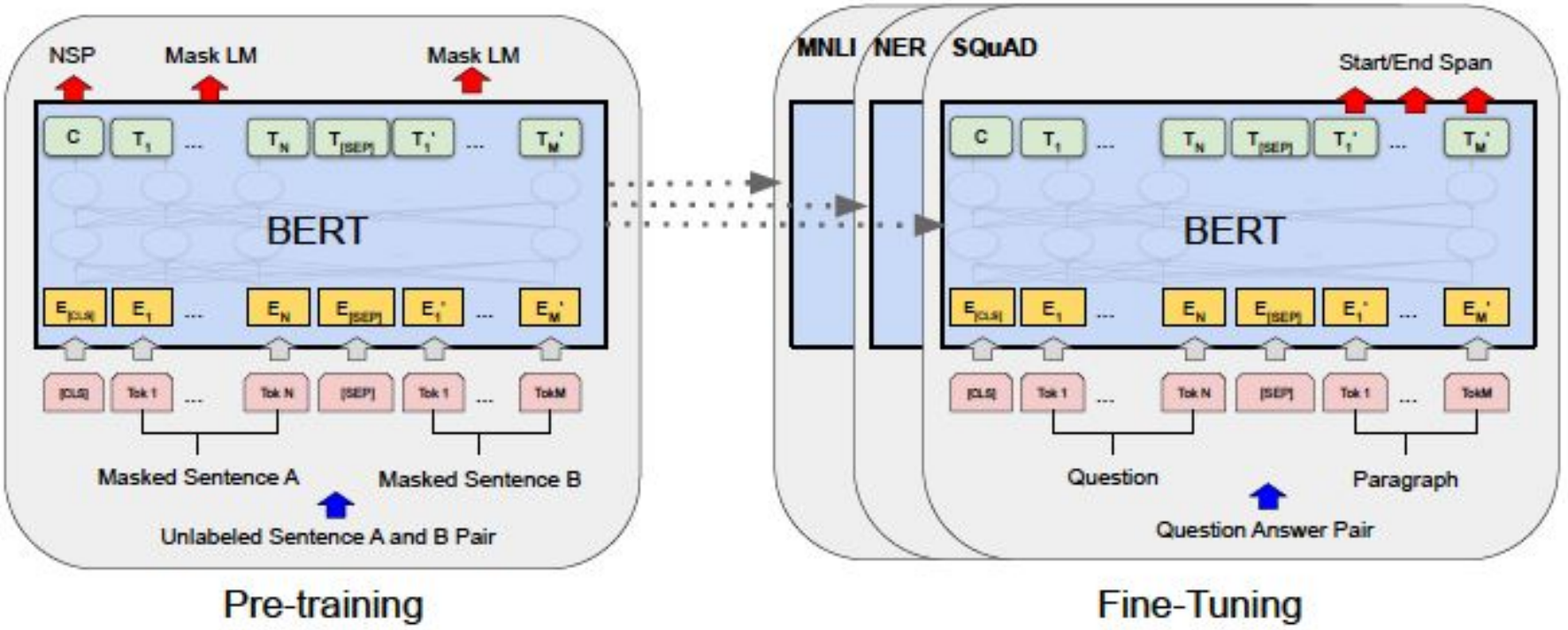
- Para resolver diferentes tarefas em PLN, é necessário **compreender língua**;

- **Ideia geral:**

1. **Pré-treina BERT** para compreender língua;
2. **Refina (fine-tuning)** o BERT para aprender tarefas específicas
   - Análise de sentimentos
   - Tradução Automática
   - Sumarização, etc.



BERT - Ideia Geral

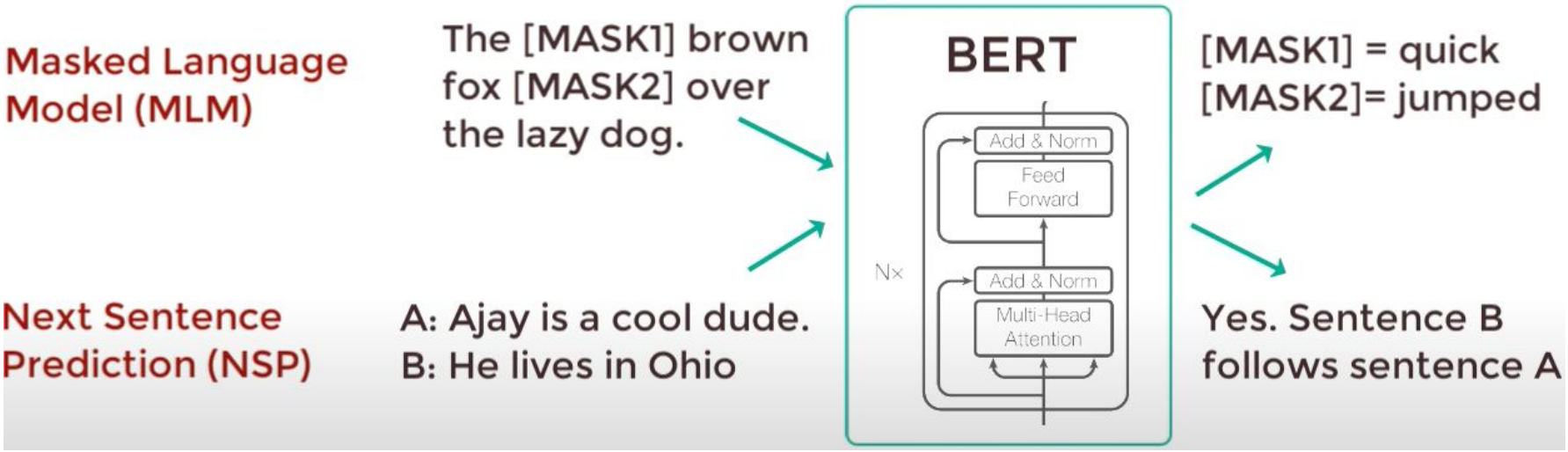


BERT - Pré-Treinamento

- **Pré-treinamento:** Aprende a **compreender língua (linguagem)**... com base em 2 tarefas:
  1. **Masked Language Model (MLM)**: sentenças com tokens aleatórios mascarados;
     - Objetivo é identificar os tokens mascarados;
     - Ajuda o BERT a **aprender contextos bidirecionais;**
  2. **Next Sentence Prediction (NSP)**: recebe duas sentenças e identifica se a 2a sentença segue a 1a.
     - Problema de classificação binária
     - Ajuda o BERT a **aprender contexto entre diferentes sentenças**

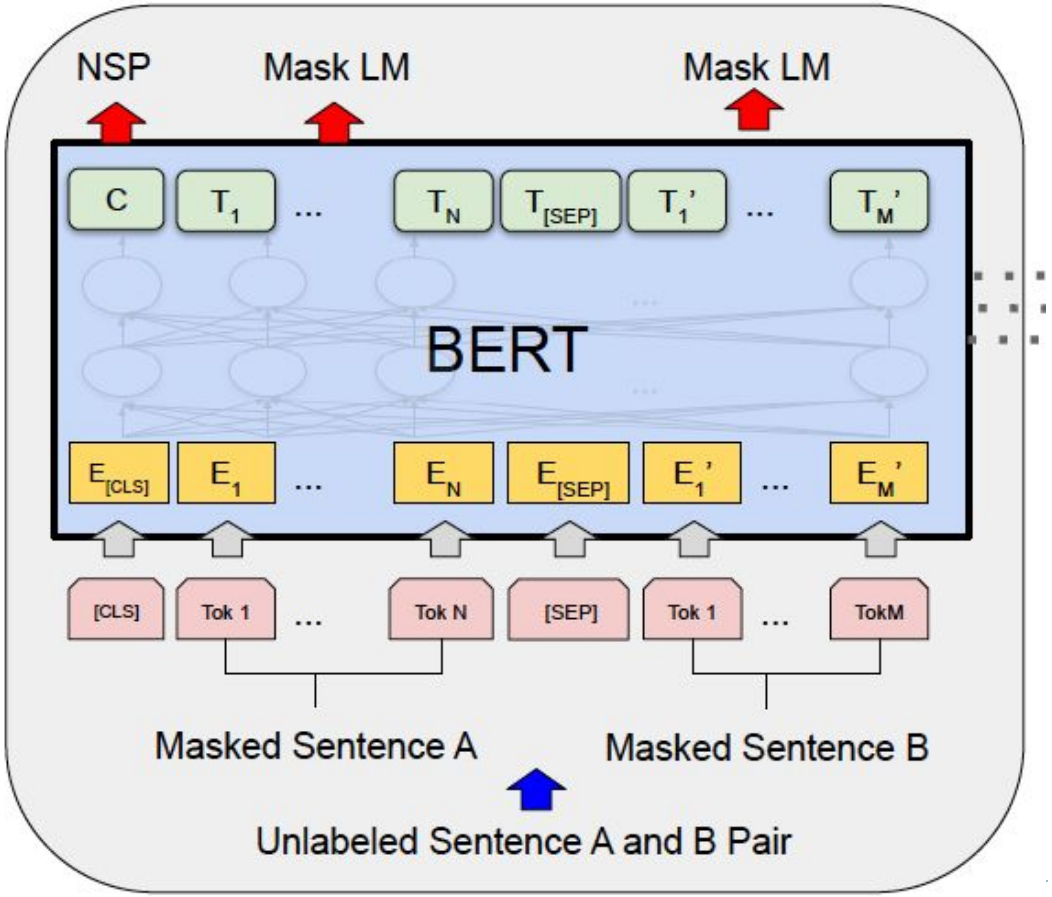
BERT - Pré-Treinamento

- **Pré-treinamento:** Aprende a compreender língua (linguagem)... com base em 2 tarefas:



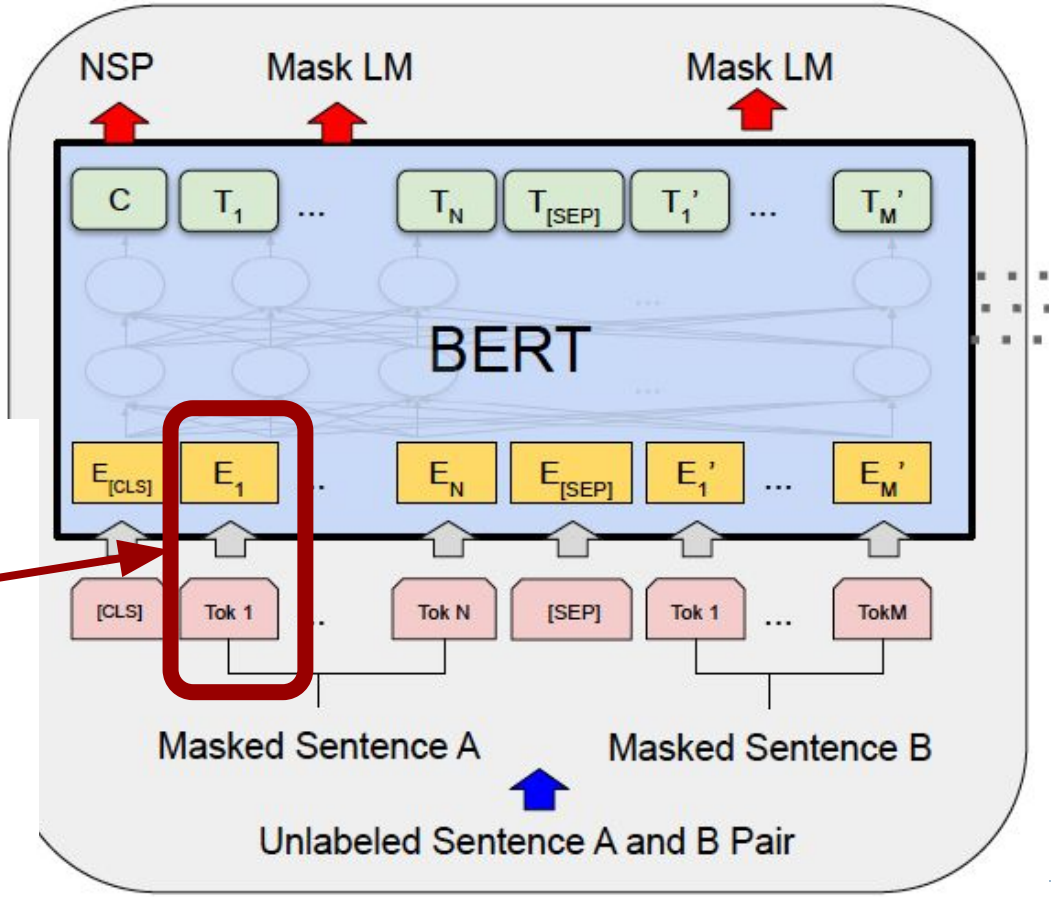
BERT - Pré-Treinamento

- **Pré-treinamento:** Treinamento simultâneo nos 2 problemas:
  - Entrada é um par de sentenças com tokens mascarados!



BERT - Pré-Treinamento

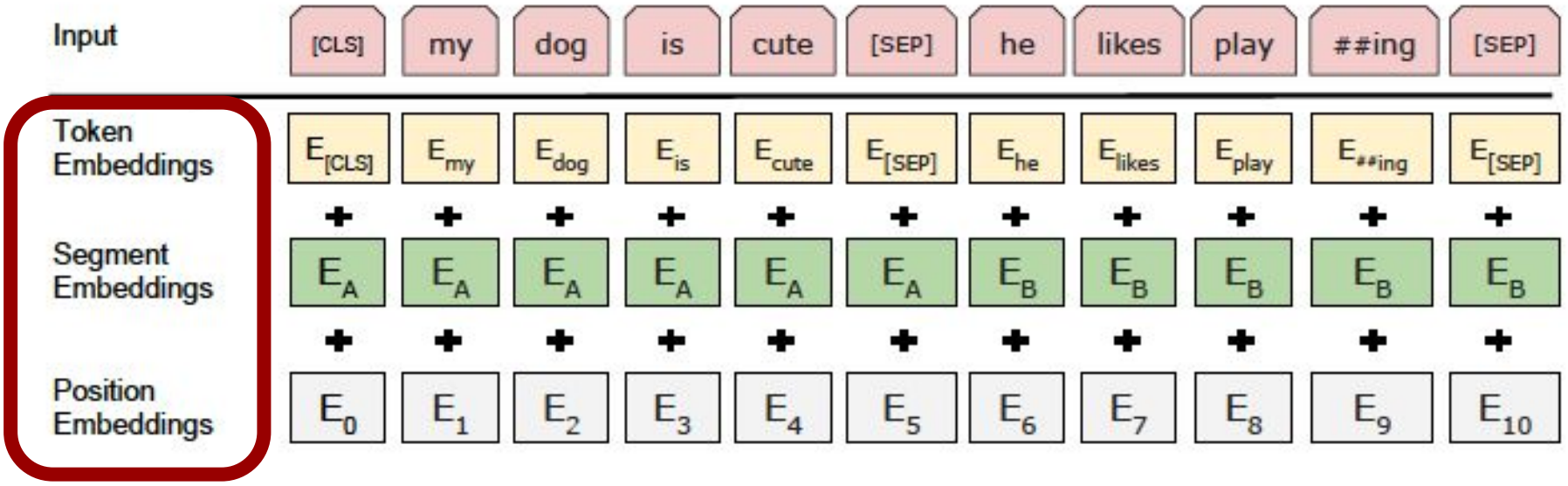
- **Pré-treinamento:** Treinamento simultâneo nos 2 problemas:
  - Entrada é um par de sentenças com tokens mascarados!



Tokens convertidos pra embeddings usando embeddings pré-treinados

BERT - Pré-Treinamento

- Representação dos embeddings de entrada
  - Construído a partir da junção de 3 vetores

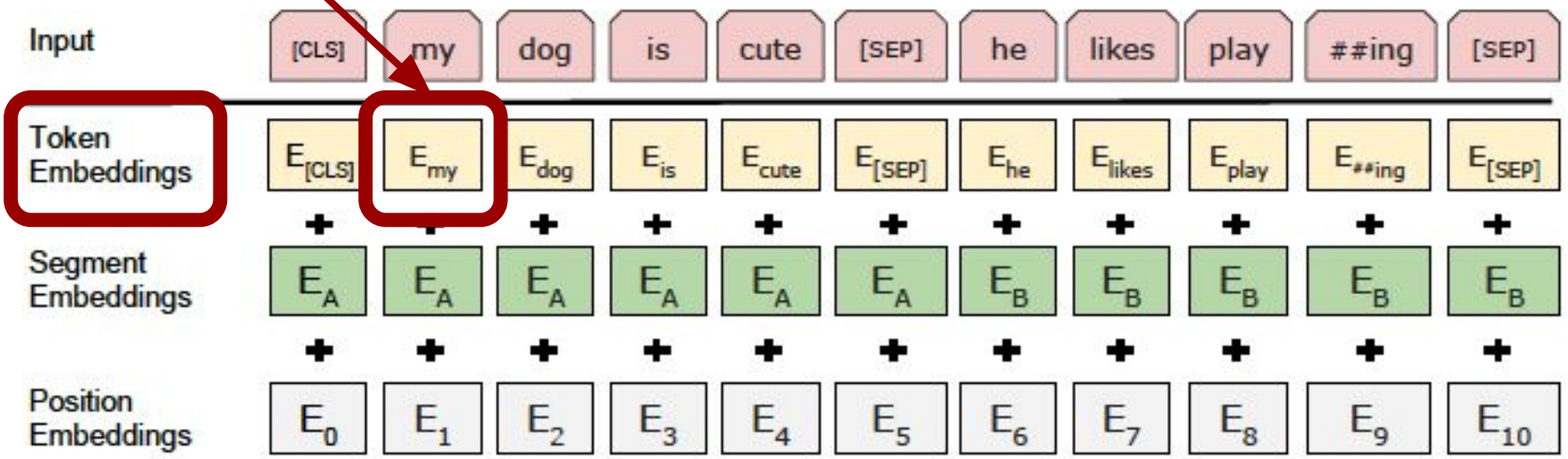


BERT - Pré-Treinamento

- Representação dos embeddings de entrada
  - Construído a partir da junção de 3 vetores

1. **Embedding do token (pré-treinado)**

Usa embeddings do "Wordpieces" com um vocabulário de 30k tokens

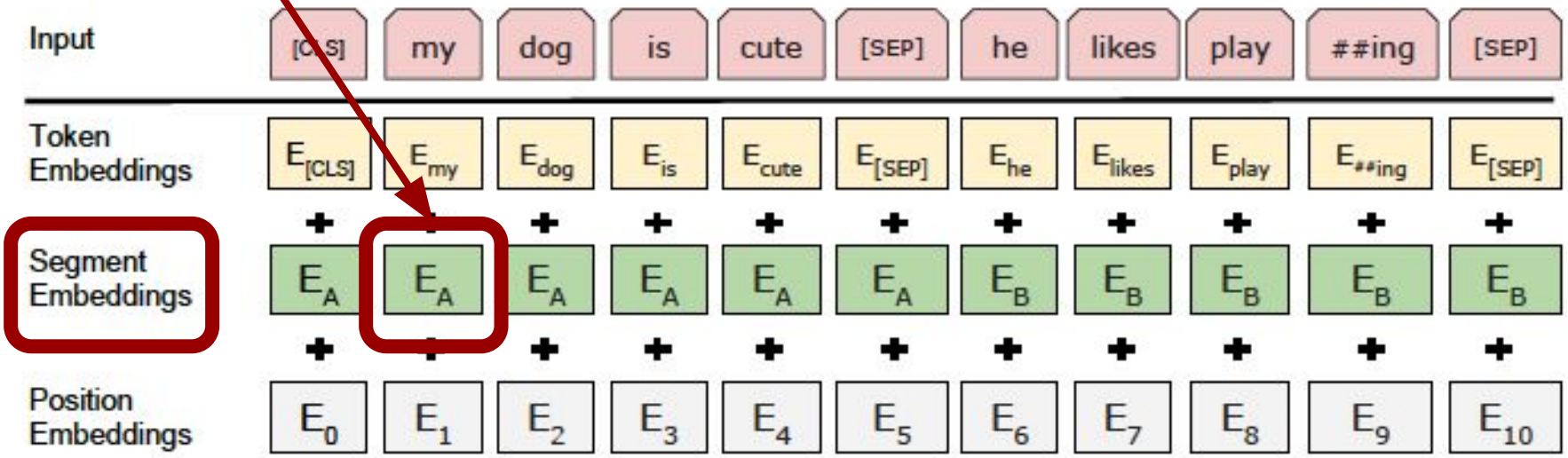


BERT - Pré-Treinamento

- Representação dos embeddings de entrada
  - Construído a partir da junção de 3 vetores

2. Embedding da sentença

Número da sentença codificado em um vetor

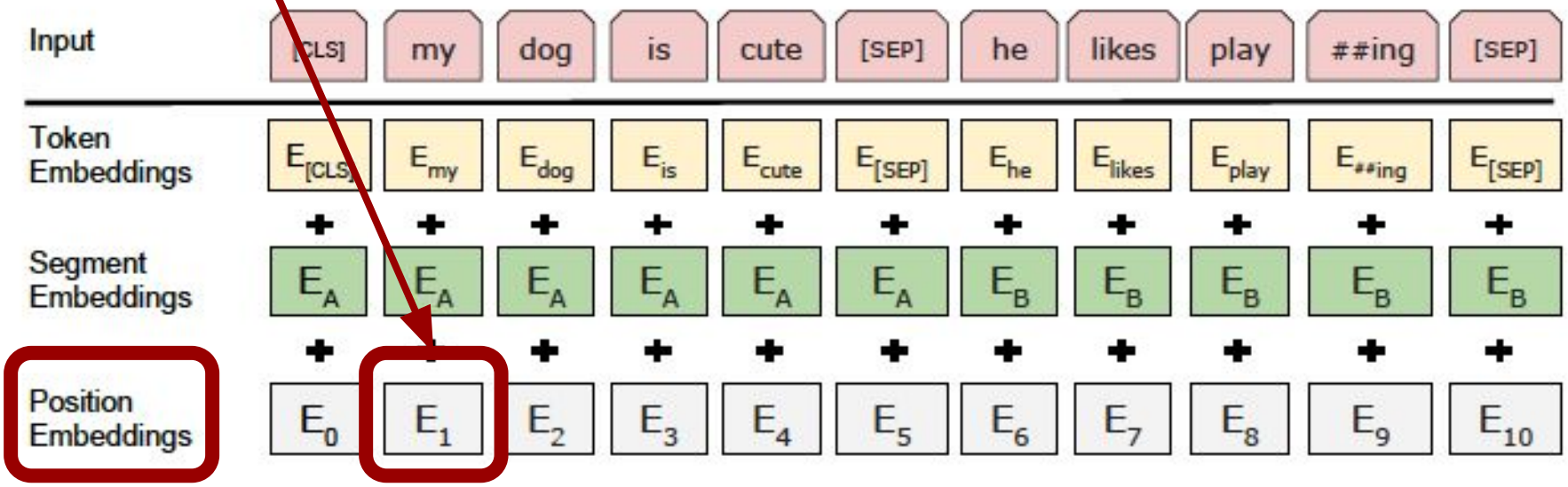


BERT - Pré-Treinamento

- Representação dos embeddings de entrada
  - Construído a partir da junção de 3 vetores

3. Embedding da posição

Posição da palavra dentro da sentença codificado em um vetor



BERT - Pré-Treinamento

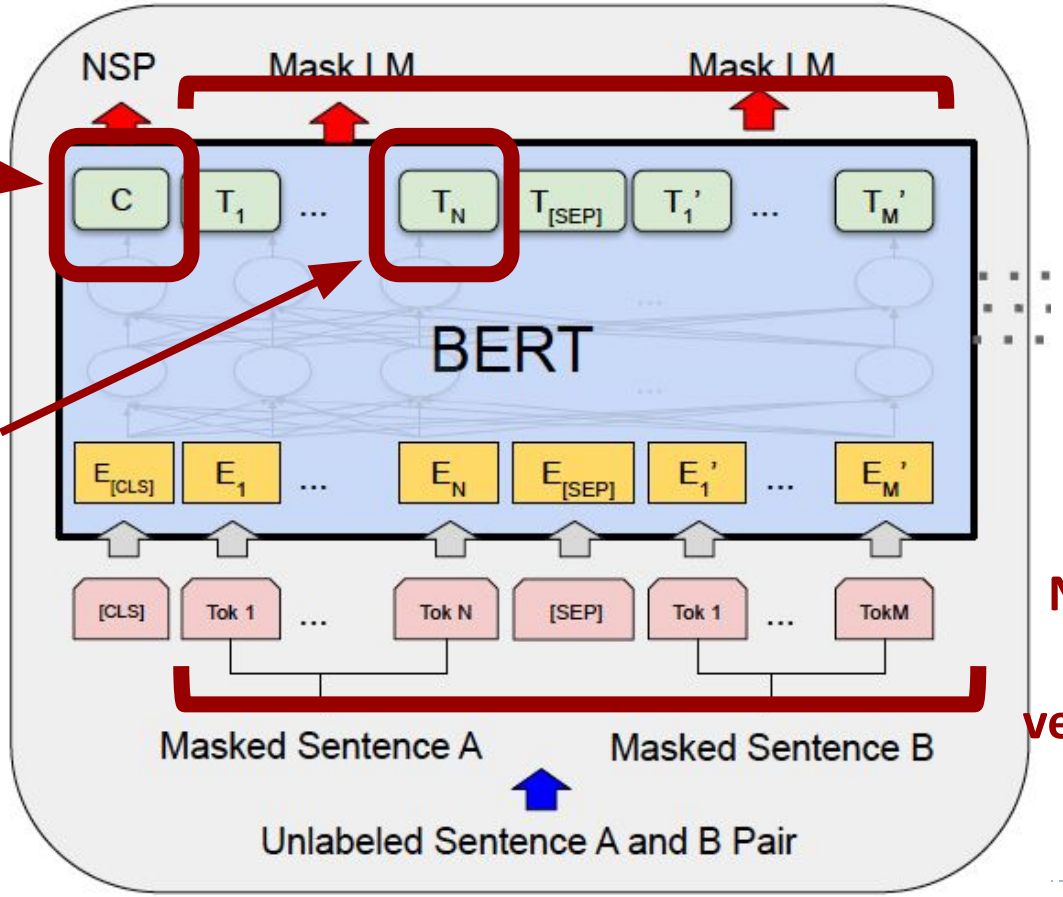
- Representação dos embeddings de entrada
  - **Embeddings de segmento** e de **posição** incluem ordenamento temporal nas palavras/tokens
  - Importante, pois os tokens/embeddings são incluídos simultaneamente no BERT
  - Modelos de linguagem necessitam que esse ordenamento seja preservado.

BERT - Pré-Treinamento

- **Pré-treinamento:** Treinamento simultâneo nos 2 problemas:
  - Saída é a classificação do NSP + sentença não mascarada

Saída binária do NSP (1 ou 0)

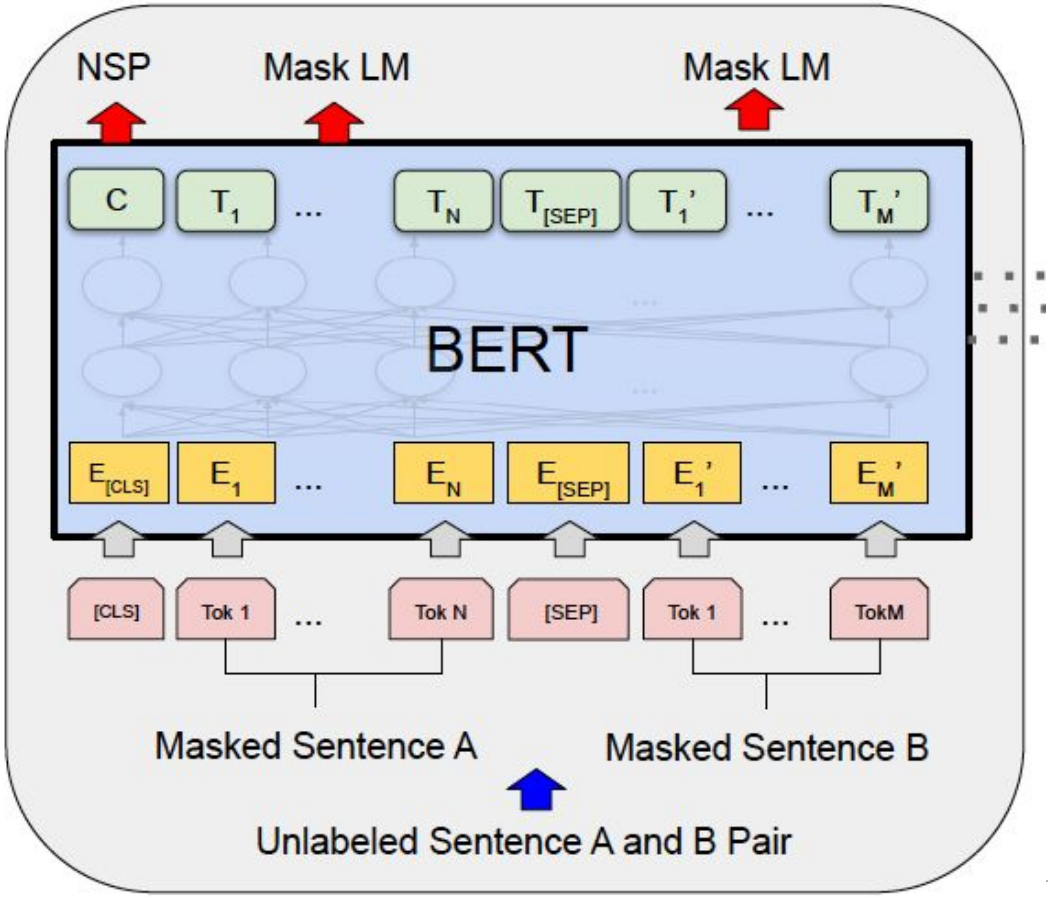
Cada T_i é um word vector do MLM



Nº de vetores de saída == Nº de vetores de entrada

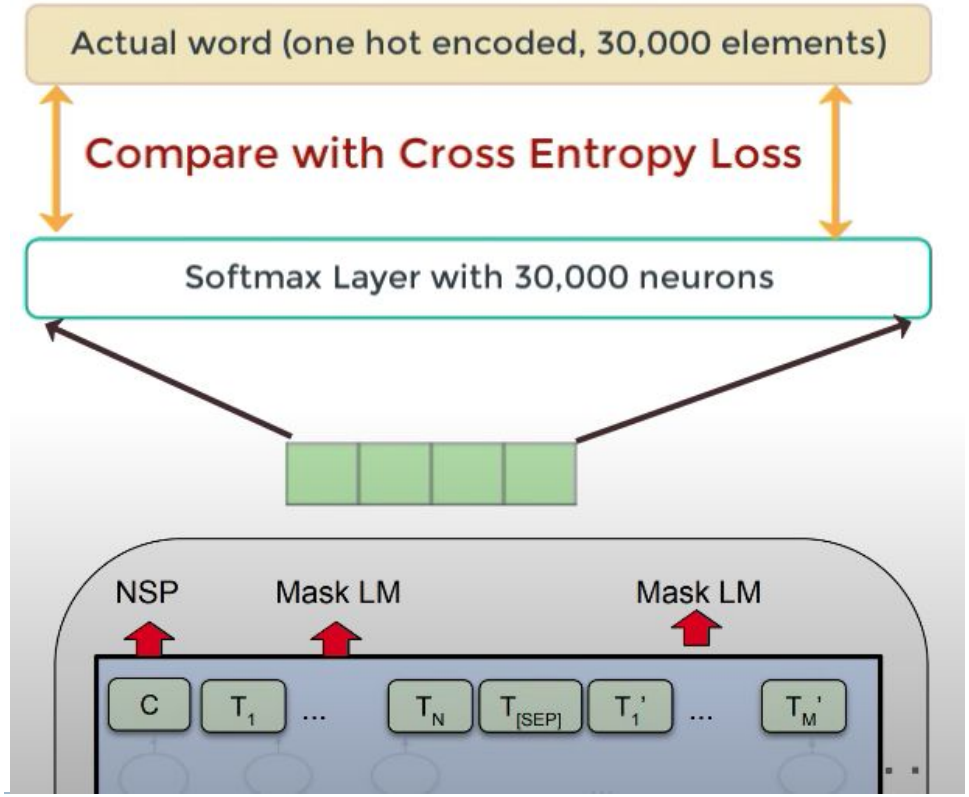
BERT - Pré-Treinamento

Vetores T_i possuem o mesmo tamanho e são gerados simultaneamente



BERT - Pré-Treinamento

- Cada vetor T_i passa por uma camada FC com 30k neurônios (tamanho do vocabulário) com função de ativação softmax
  - Treinada com a função de erro cross entropy

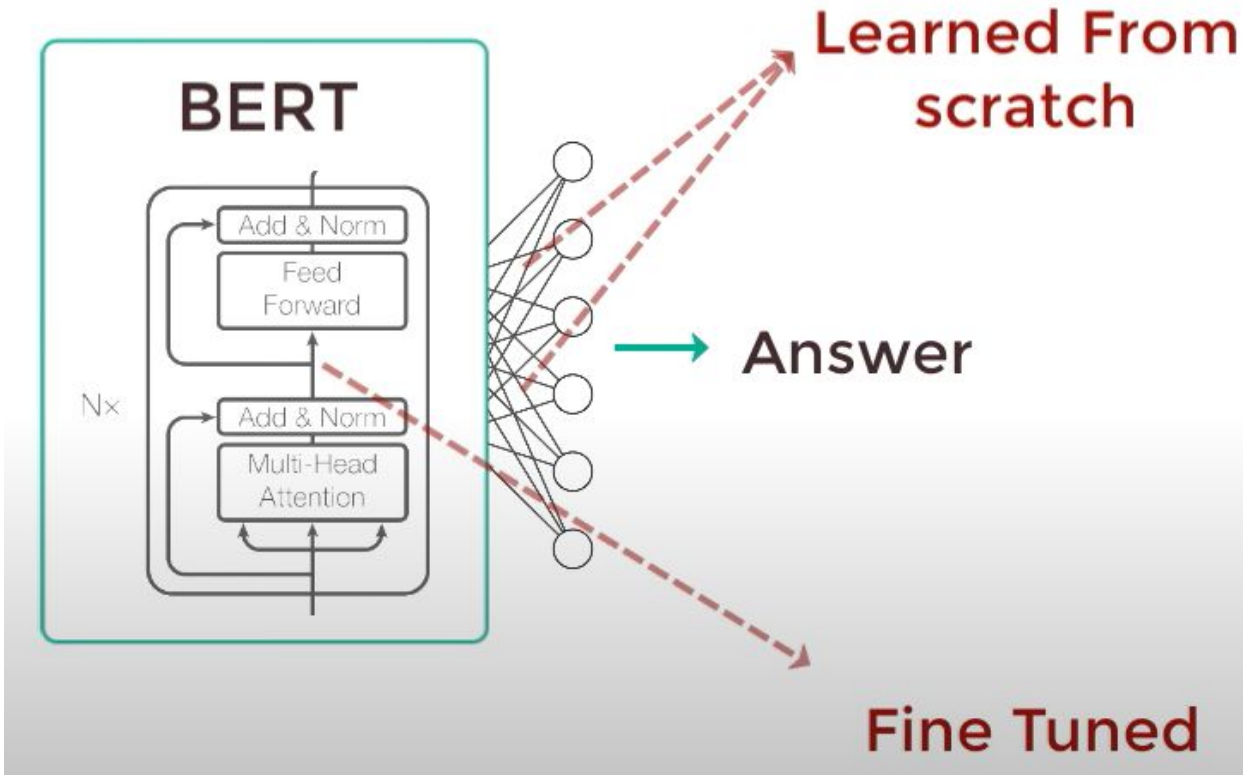


BERT - Fine-Tuning

- **Refinamento (Fine-tuning):** Usa o conhecimento em língua aprendido como base para resolver problemas específicos:
  - Treina (refina) o BERT para problemas específicos
    - Geralmente utilizando dados supervisionados.
  - Substitui as camadas de saída totalmente conectadas da rede e as treinam/adaptam para a tarefa específica
    - Pesos das novas camadas de saída são treinados do zero.
    - Demais parâmetros do modelo são "suavemente" refinados!
  - **Treinamento mais rápido e com necessidade de menos dados supervisionados!**

BERT - Fine-Tuning

- **Refinamento (Fine-tuning):** Usa o conhecimento em língua aprendido como base para resolver problemas específicos:



BERT - Fine-Tuning

- **Refinamento (Fine-tuning):**

  Ex: SQuAD v 1.1 - The Stanford Question Answering Dataset

Passage

S_1 : Pharmacists are healthcare professionals with specialized education and training who perform various roles to ensure optimal health outcomes for their patients through the quality use of medicines.

S_2 : Pharmacists may also be **small-business proprietors**, owning the pharmacy in which they practice.

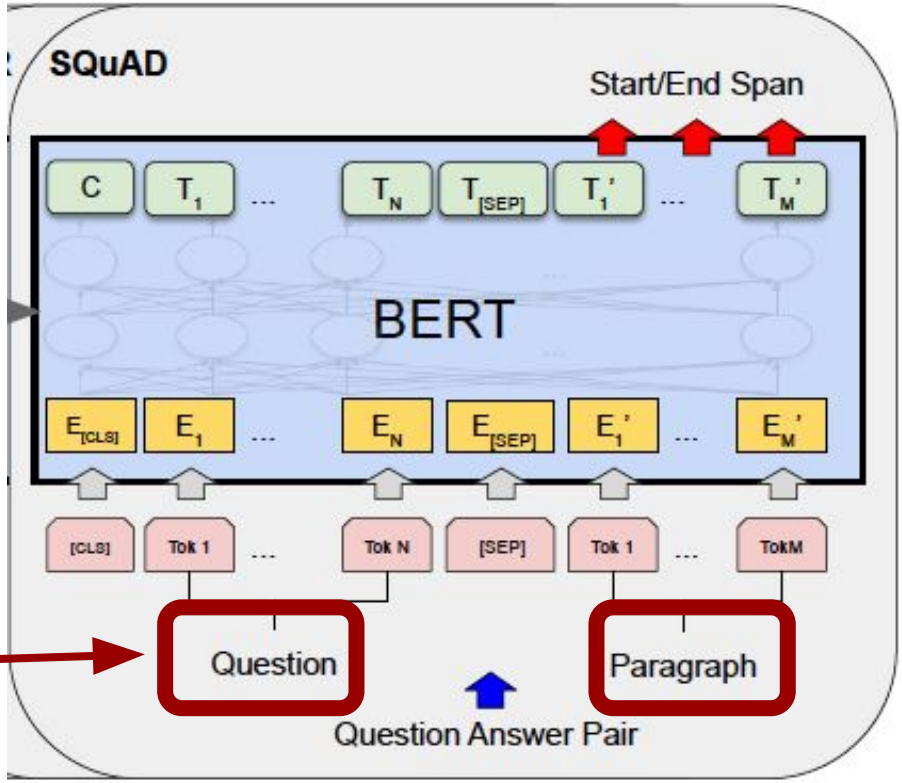
S_3 : Since pharmacists know about the mode of action of a particular drug, and its metabolism and physiological effects on the human body in great detail, they play an important role in optimization of a drug treatment for an individual.

Question: What other role do many pharmacists play?

Answer: **small-business proprietors**

BERT - Fine-Tuning

- **Refinamento (Fine-tuning):** Usa o conhecimento em língua aprendido como base para resolver problemas específicos:



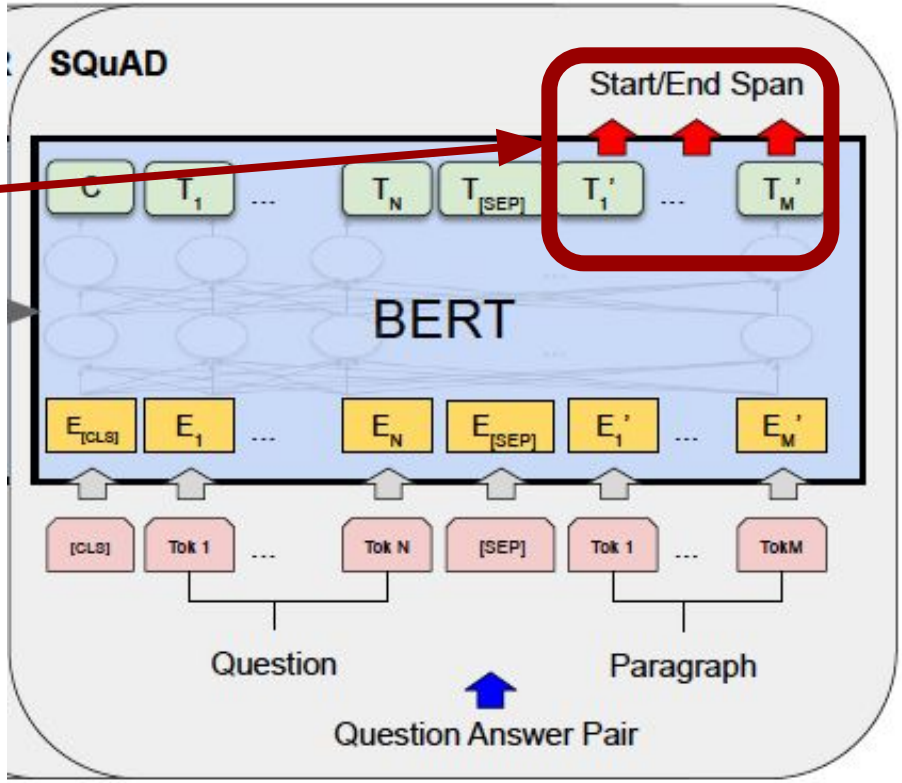
Modifica a entrada para ser a pergunta seguida da passagem (parágrafo) contendo a resposta

BERT - Fine-Tuning

- **Refinamento (Fine-tuning):** Usa o conhecimento em língua aprendido como base para resolver problemas específicos:

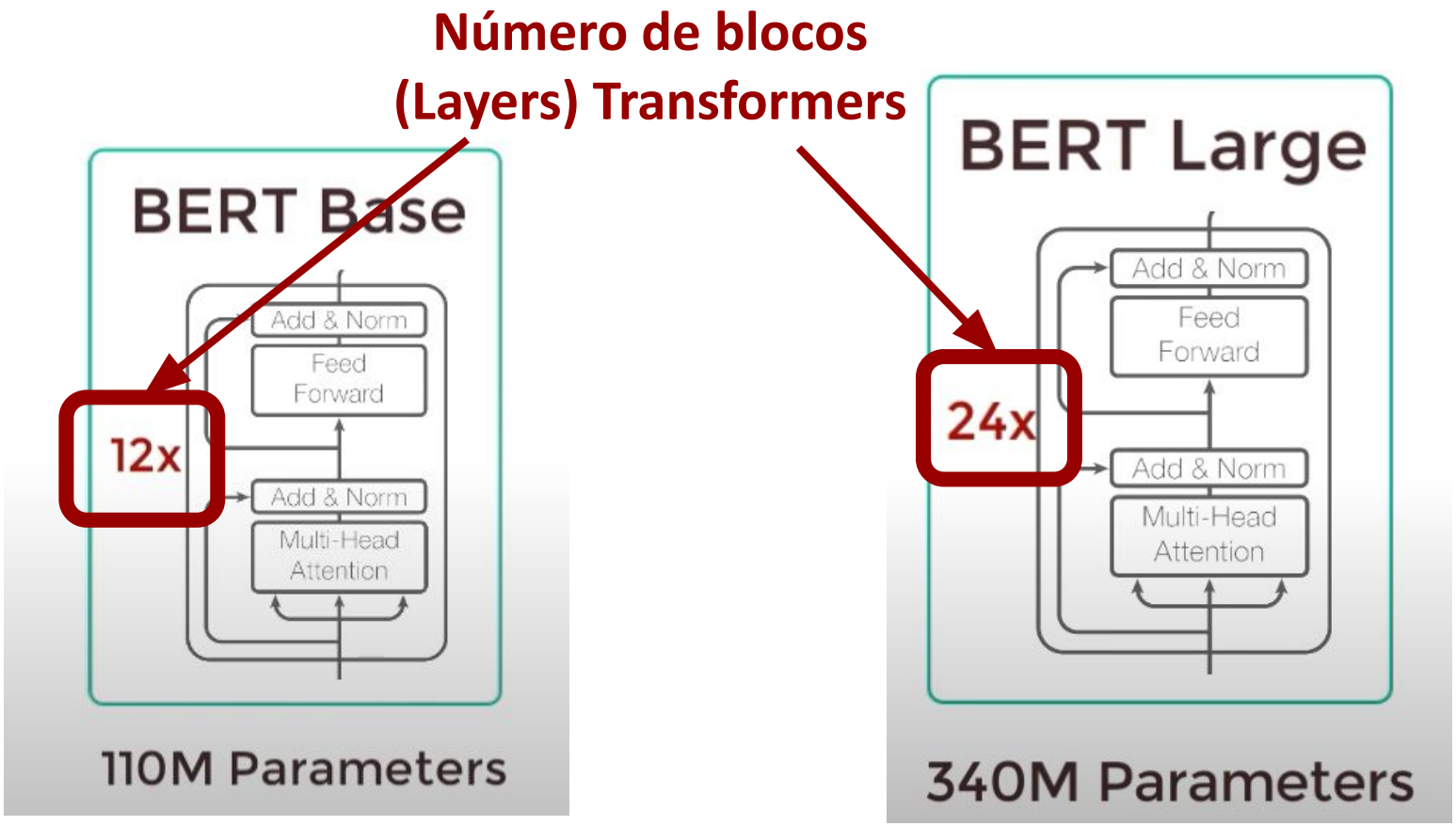
Saída inclui um vetor de start e end na parte da passagem que encapsula a resposta

Assumindo que a resposta está dentro da passagem.



BERT - Desempenho

- $BERT_{BASE}$ e $BERT_{LARGE}$



BERT - Desempenho

- Todos os resultados do paper para o fine-tuning podem ser replicados com 1 TPU com menos de 1h de treinamento

- BERT Large treinado para o SQuAD v1.1: 30 minutos de treinamento numa única TPU - F1-score de 91%

Generative Pretrained Transformers (GPT)

GPT-1:https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

GPT-2::https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

GPT-3: https://arxiv.org/abs/2005.14165

GPT - Ideia Geral

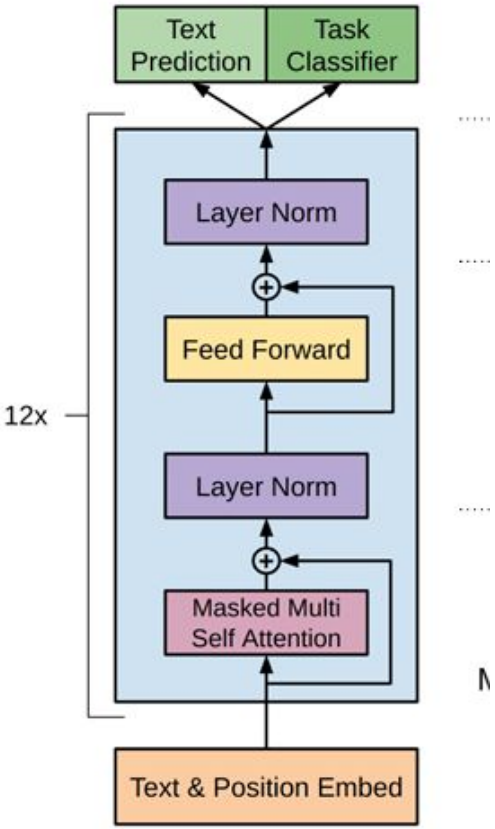
- Utiliza a mesma ideia geral do BERT

- **Ideia geral:**

1. **Pré-treinamento** para compreender língua;
2. **Fine-tuning** para aprender tarefas específicas

GPT-1 - Arquitetura

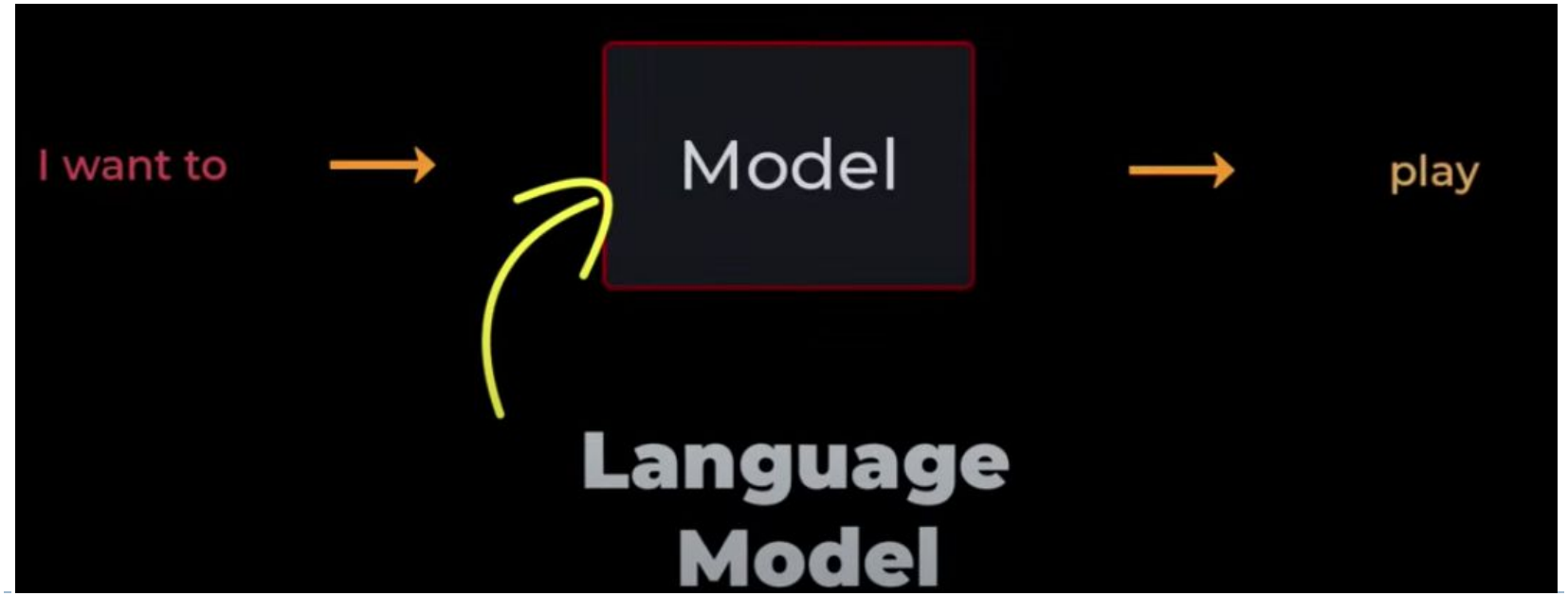
- Encadeamento de vários blocos Decoders (Transformers)



GPT-1:https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

GPT-1 - Pré-treinamento

- **Pré-treinamento:** aprende como um <u>modelo de linguagem</u>
  - Recebe uma sentença como entrada e tenta prever a próxima palavra
  - Possível aprender a partir de dados não-rotulados
    - Self-supervised learning



GPT-1 - Fine-Tuning

- **Fine-Tuning:** Usa o conhecimento em língua aprendido como base para resolver problemas específicos
  - Precisa de dados rotulados para isso!
  - Requer menos dados que um treinamento do zero;

- **Mas existem algumas questões com o fine-tuning!**

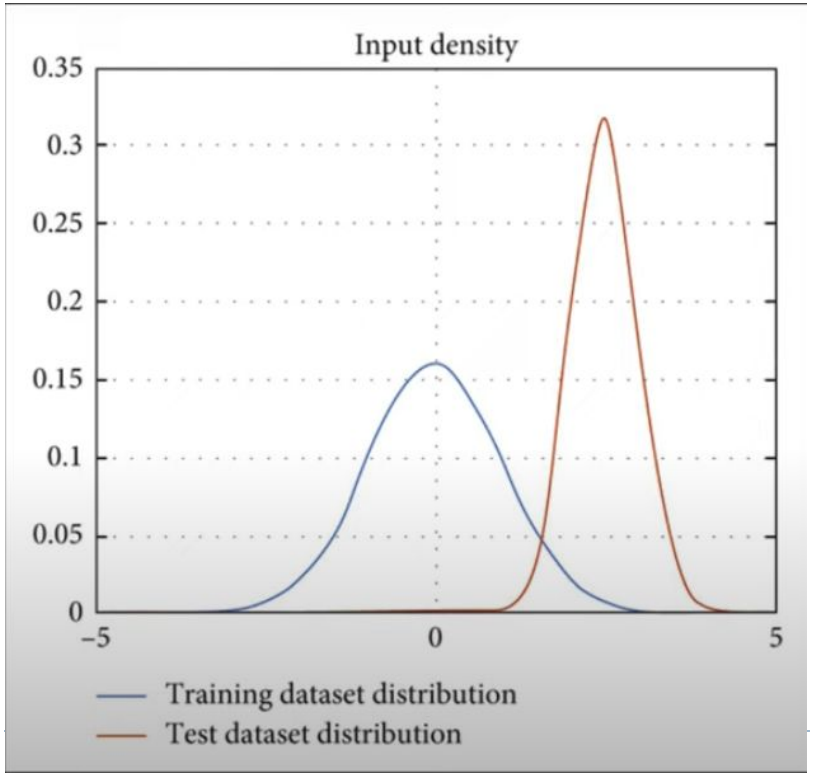
GPT-1 - Problemas com o Fine-Tuning

1. **Ainda requer muitos dados!**

Task	Data Requirements
Machine Translation	~100,000 samples
Question-Answer	~100,000 samples
Text Summarization	~100,000 samples
Sentence Similarity	~100,000 samples

GPT-1 - Problemas com o Fine-Tuning

2. **É fácil entrar em overfitting!**
   - Necessário avaliar se o dataset do fine-tuning é uma boa representação (amostra) da natureza!



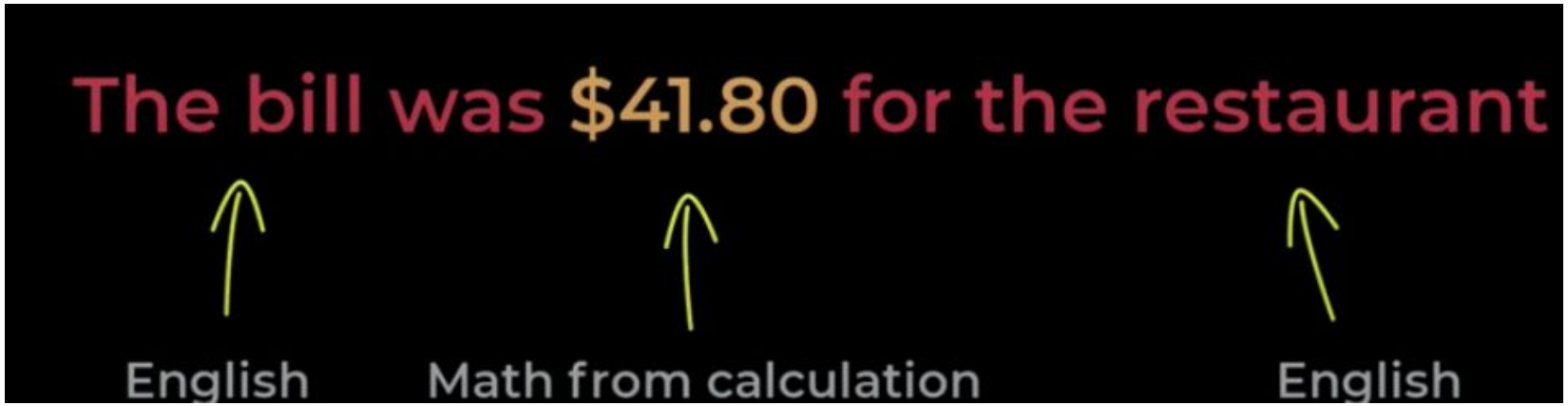
GPT-1 - Problemas com o Fine-Tuning

3. **Não é como os humanos aprendem!**
   - Humanos aprendem com poucos exemplos (não milhares)!

Task	Data Requirements	Human
Machine Translation	~100,000 samples	10 samples
Question-Answer	~100,000 samples	1 sample
Text Summarization	~100,000 samples	2 samples
Sentence Similarity	~100,000 samples	3 samples

GPT-1 - Problemas com o Fine-Tuning

4. **Não é natural para compreender questões mais abrangentes no contexto de PLN**



GPT-2 - Meta-Learning

- Uma alternativa para endereçar essas questões é o **Meta-Learning** - proposto no GPT2

- **Ideia geral:**

1. **Pré-treinamento** para compreender língua como no GPT1;
2. **Zero-shot learning** em vez de Fine-tuning;

GPT-2::https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

GPT-2 - Zero Shot Learning

- **Zero-Shot Learning:** Realiza uma tarefa específica quando é fornecido apenas uma **"instrução + entrada"**
- Não é necessário fazer atualização de parâmetros;
- Na inferência, é passado a entrada e também um prompt: que instrução deve ser feita com a entrada

GPT-2 - Zero Shot Learning

- **Zero-Shot Learning:** Realiza uma tarefa específica quando é fornecido apenas uma **"instrução + entrada"**

- Título do artigo: **"Language Models are Unsupervised Multitask Learners"**

GPT-2 - Zero Shot Learning

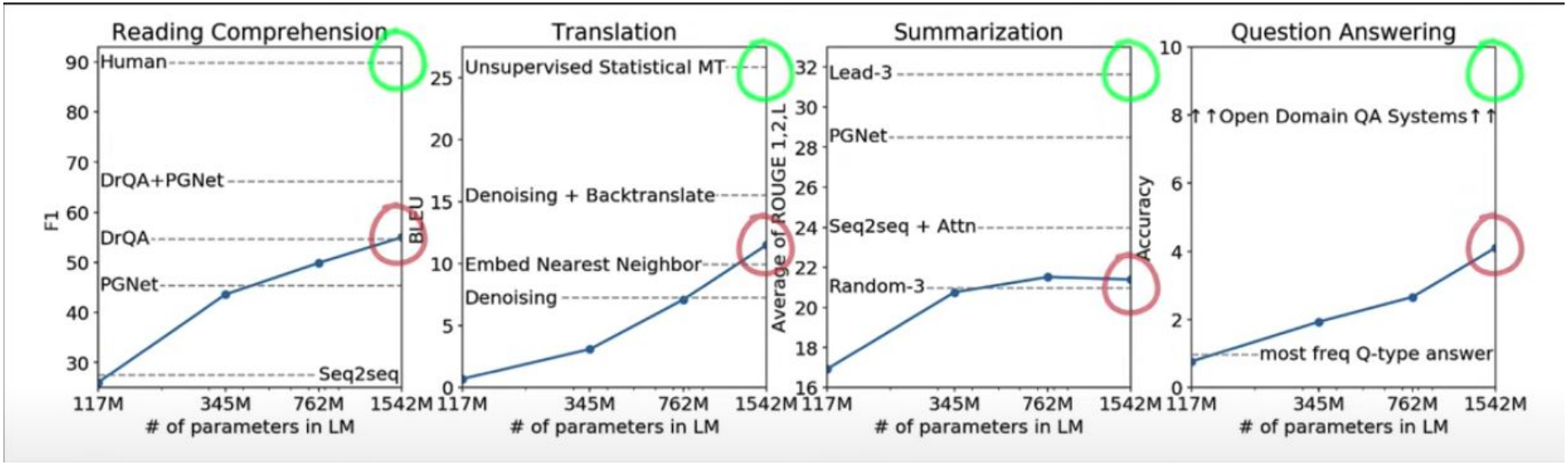
- Zero-short learning é muito difícil para o modelo!
  - É difícil até para humanos!

- Necessário escalar a arquitetura para capturar os mais variados padrões da língua para as mais variadas tarefas

- GPT-2 foi treinado com **1,5 bilhões de parâmetros!**

GPT-2 - Zero Shot Learning

- GPT-2 não se saiu tão bem em alguns benchmarks quando comparado com o estado da arte na área.



- **Mas escalar a arquitetura deu alguns sinais de ajuda na performance!**

GPT-3

- Explorou a ideia de **escalar ainda mais arquitetura** com o uso de **meta-learning** (da GPT-2)

- GPT-3 possui **175 bilhões de parâmetros!!!**

- Explora melhor o conceito de **Meta-Learning**, admitindo outras possibilidades:
  - Zero-shot Learning;
  - One-Shot Learning;
  - Few-shot Learning;

 GPT-3: https://arxiv.org/abs/2005.14165

GPT-3

- Explorou a ideia de **escalar ainda mais arquitetura** com o uso de **meta-learning** (da GPT-2)

- GPT-3 possui **175 bilhões de parâmetros!!!**

- Explora melhor o conceito de **Meta-Learning**, admitindo outras possibilidades:
  - Zero-shot Learning;
  - One-Shot Learning;
  - Few-shot Learning;

GPT-3

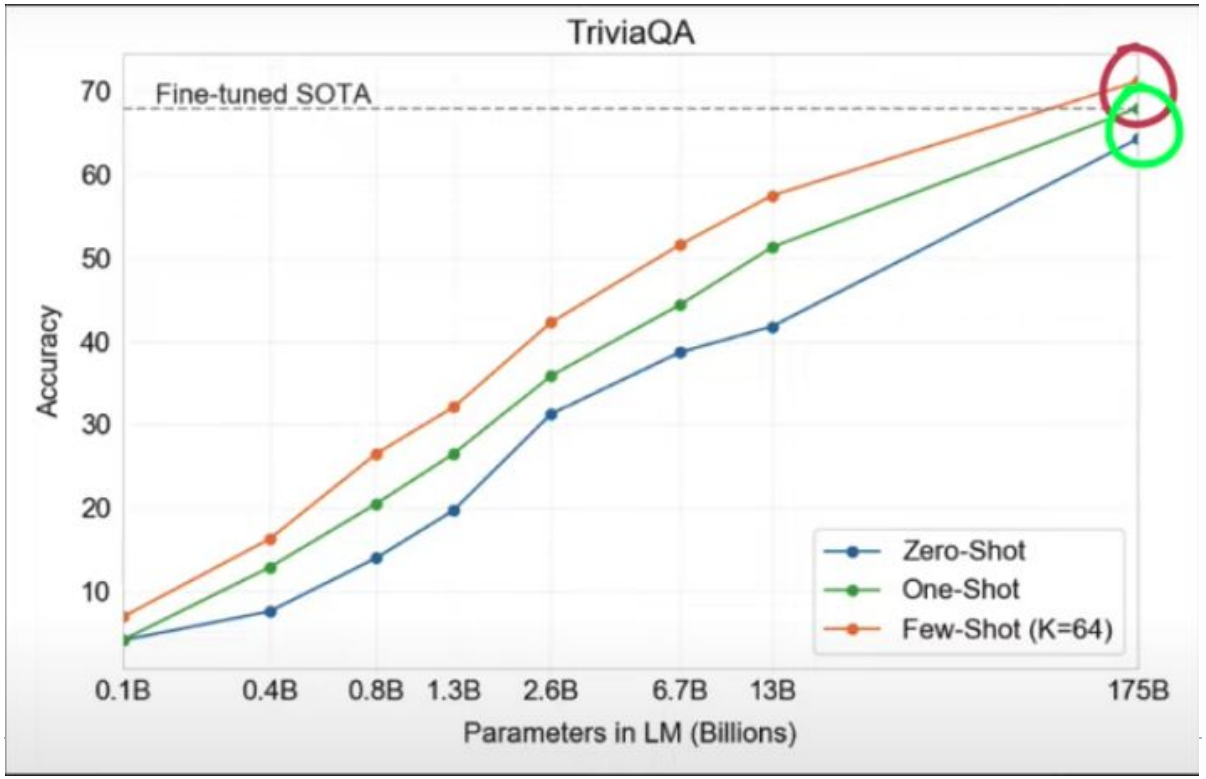
- One-Shot Learning: Um exemplo é passado para o modelo como contexto.
- **IMPORTANTE**: Não há atualização de parâmetros no GPT-3!

GPT-3

- Few-Shot Learning: Alguns exemplos são passados para o modelo como contexto.
  - Usualmente de **10 a 100 exemplos**!

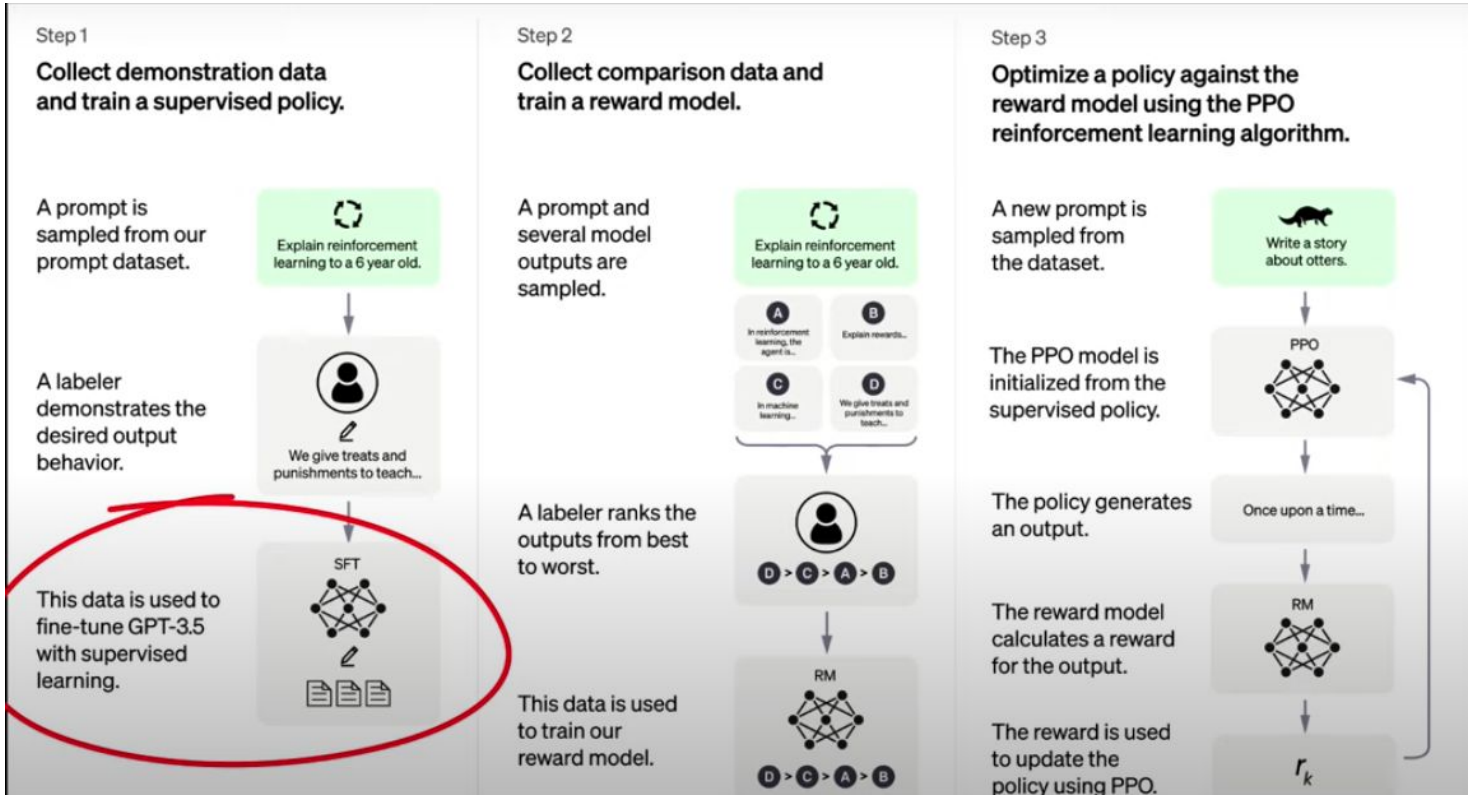
GPT-3

- Ótimo desempenho em algumas tarefas de PLN!
  - **Vermelho - GPT-3!**
  - **Verde - SOTA!**



ChatGPT

- Contudo, **ChatGPT** 1a versão (Dez/2022) já utilizava uma versão **fine-tuned do GPT!**



Universidade Federal da Paraíba

Centro de Informática

Departamento de Informática

Aprendizado Profundo

Modelos de Linguagem de Larga Escala (LLMs)

(Material Baseado em @CodeEmporium)

Tiago Maritan

(tiago@ci.ufpb.br)