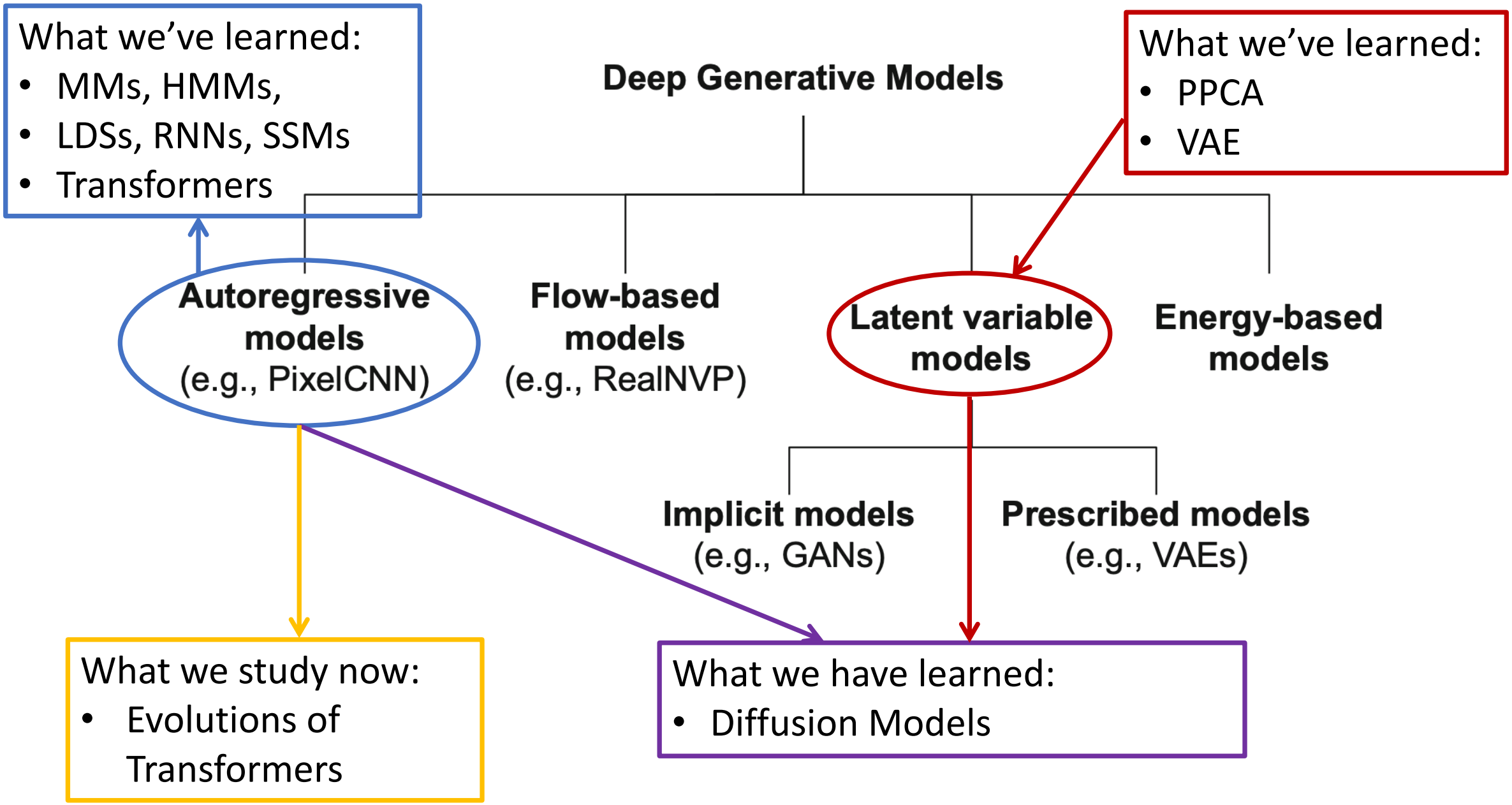
Deep Generative Models: Transformers for Text

Fall Semester 2025

René Vidal
Director of the Center for Innovation in Data Engineering and Science (IDEAS)
Rachleff University Professor, University of Pennsylvania
Amazon Scholar & Chief Scientist at NORCE

Penn
University of Pennsylvania

Taxonomy of Generative Models



Autoregressive Models

- Many kinds of models
  - Markov Chains
  - Hidden Markov Models
  - Markov Random Fields
  - Linear Dynamical Systems
  - Recurrent Neural Networks
  - State Space Models
  - **Transformers**

- Last lecture
  - **Word Embedding**
  - **Positional Encoding**
  - **Attention Mechanism**
  - **Multi-head Attention**
  - **Attention Visualization**

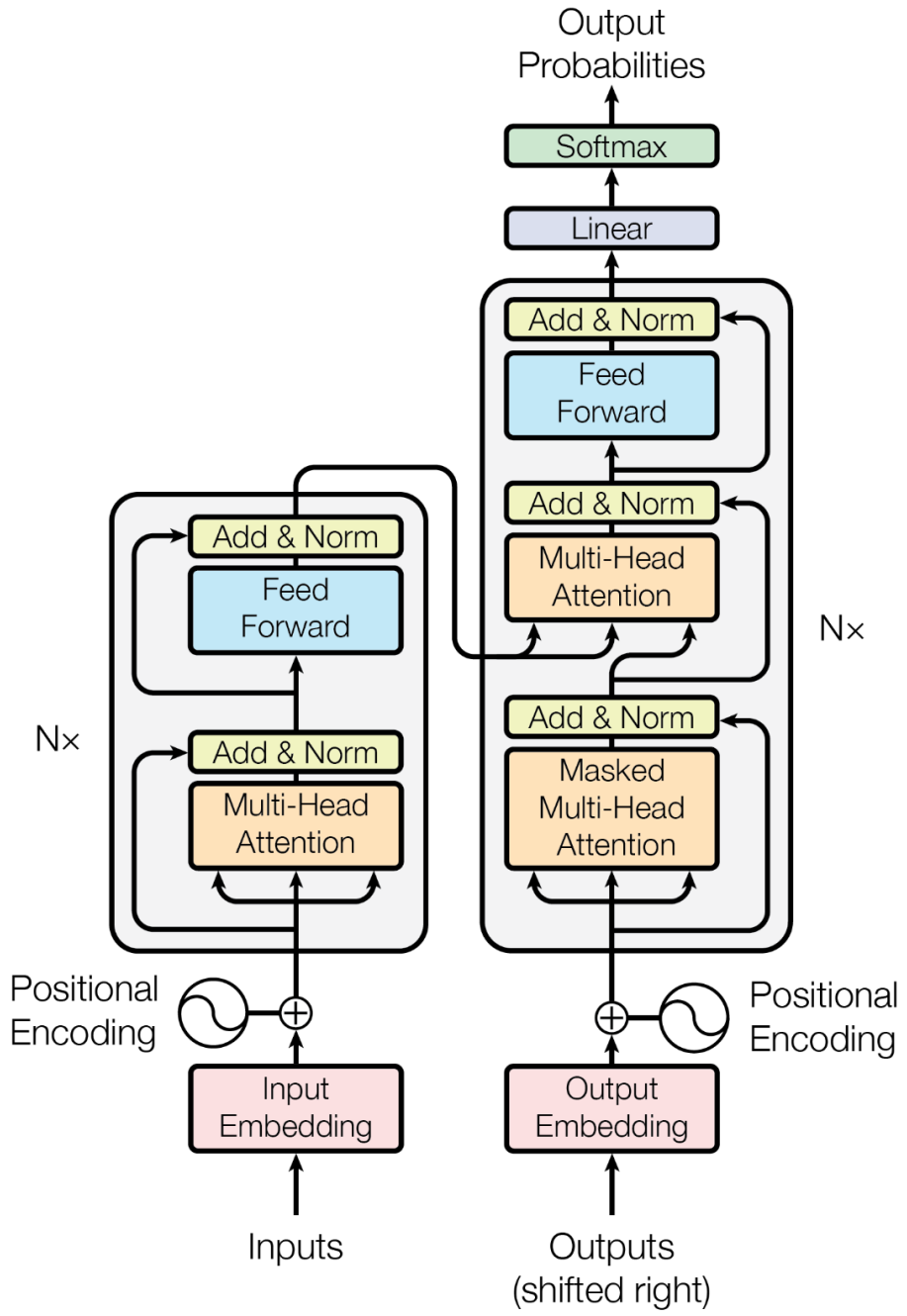


Figure 1: The Transformer - model architecture.

From Last Lecture

- [NeurIPS 2017] Attention is all you need: the Transformer that contains **Encoder Block** and **Decoder Block**.
- The design allows engineers to stack multiple blocks all together in large-scale training, which enables the emergence of foundation models.

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

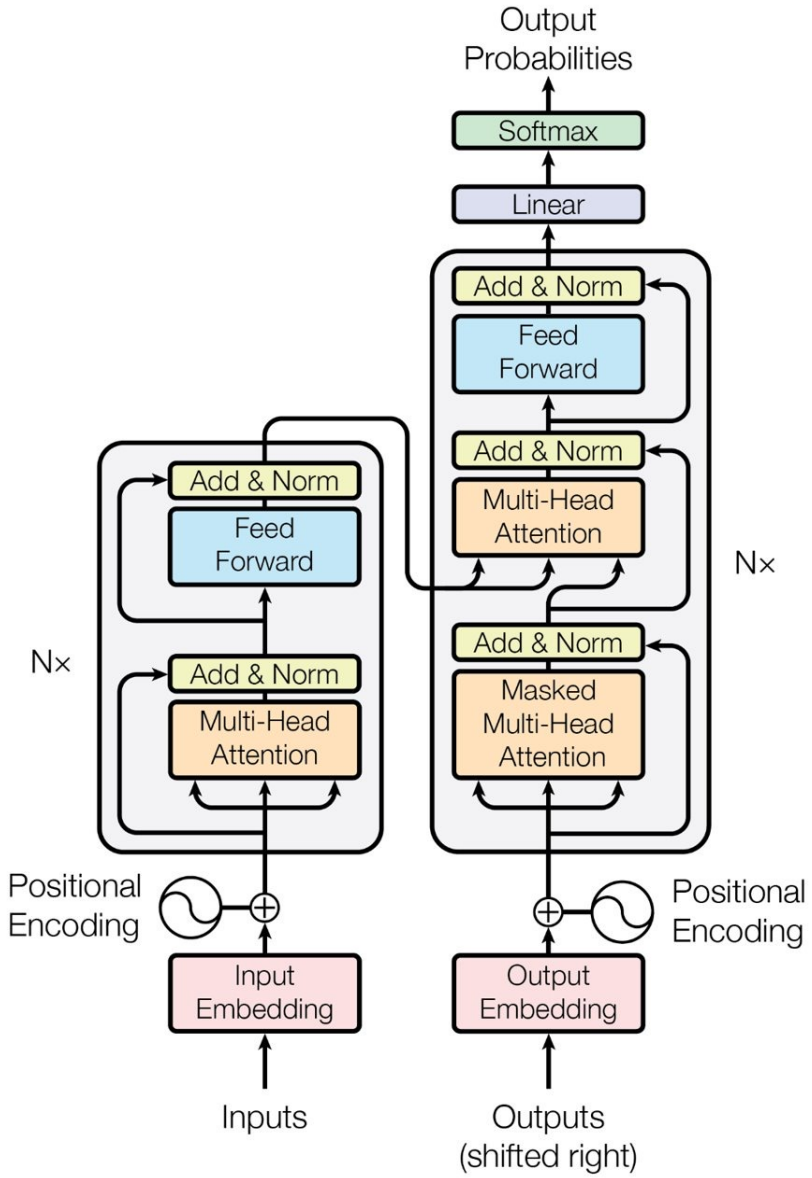
Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com



Evolutions of Transformers

Natural Language Processing:

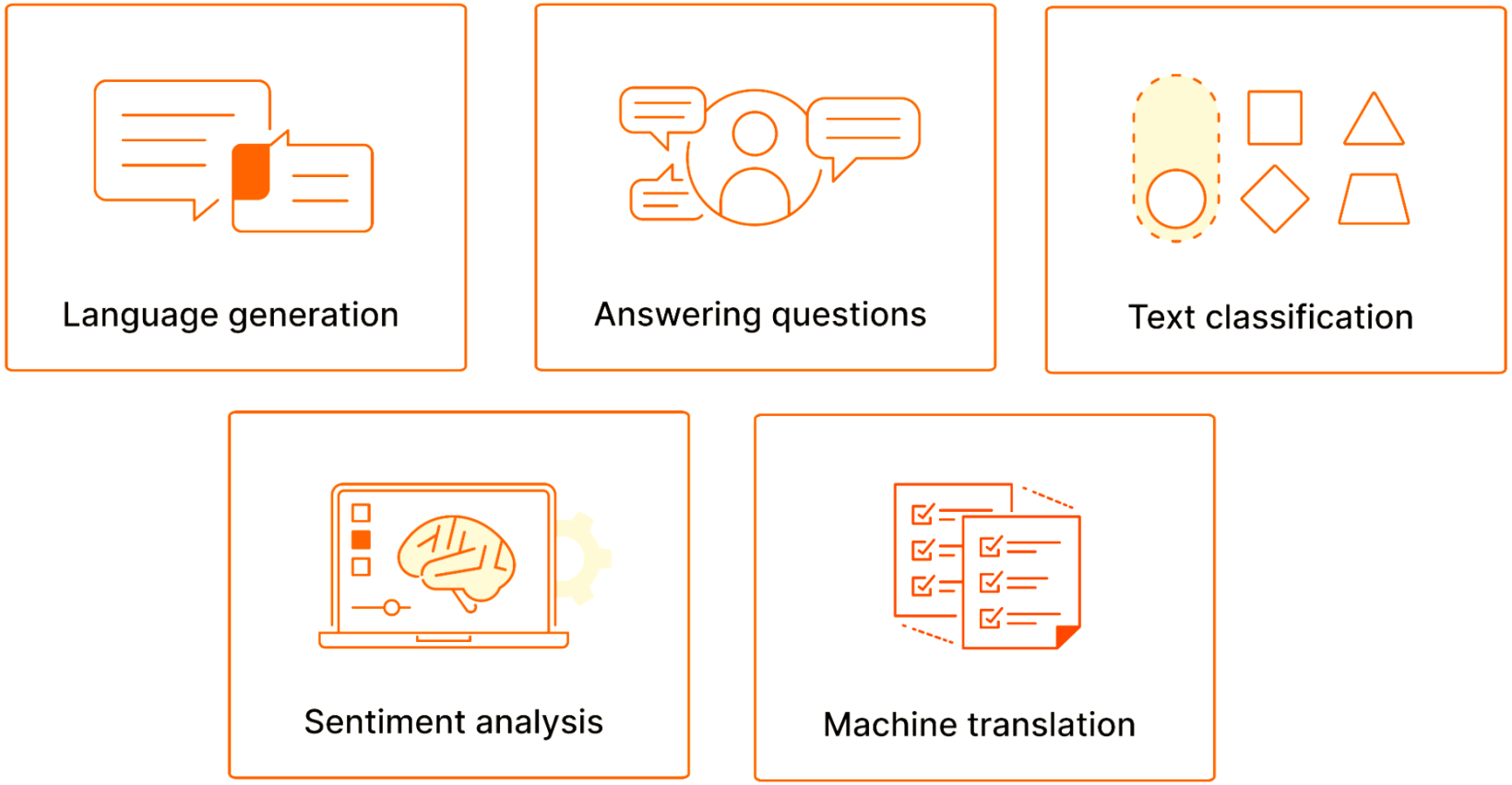
- **BERT (Bidirectional Encoder Representations from Transformers)**
- GPT (Generative Pre-trained Transformer)
- RoBERTa (Robustly Optimized Bert Pre-training)
- T5 (Text-to-Text Transfer Transformer)

When it comes to Vision:

- **Vision Transformer**
- Swin Transformer, Pyramid Vision Transformer

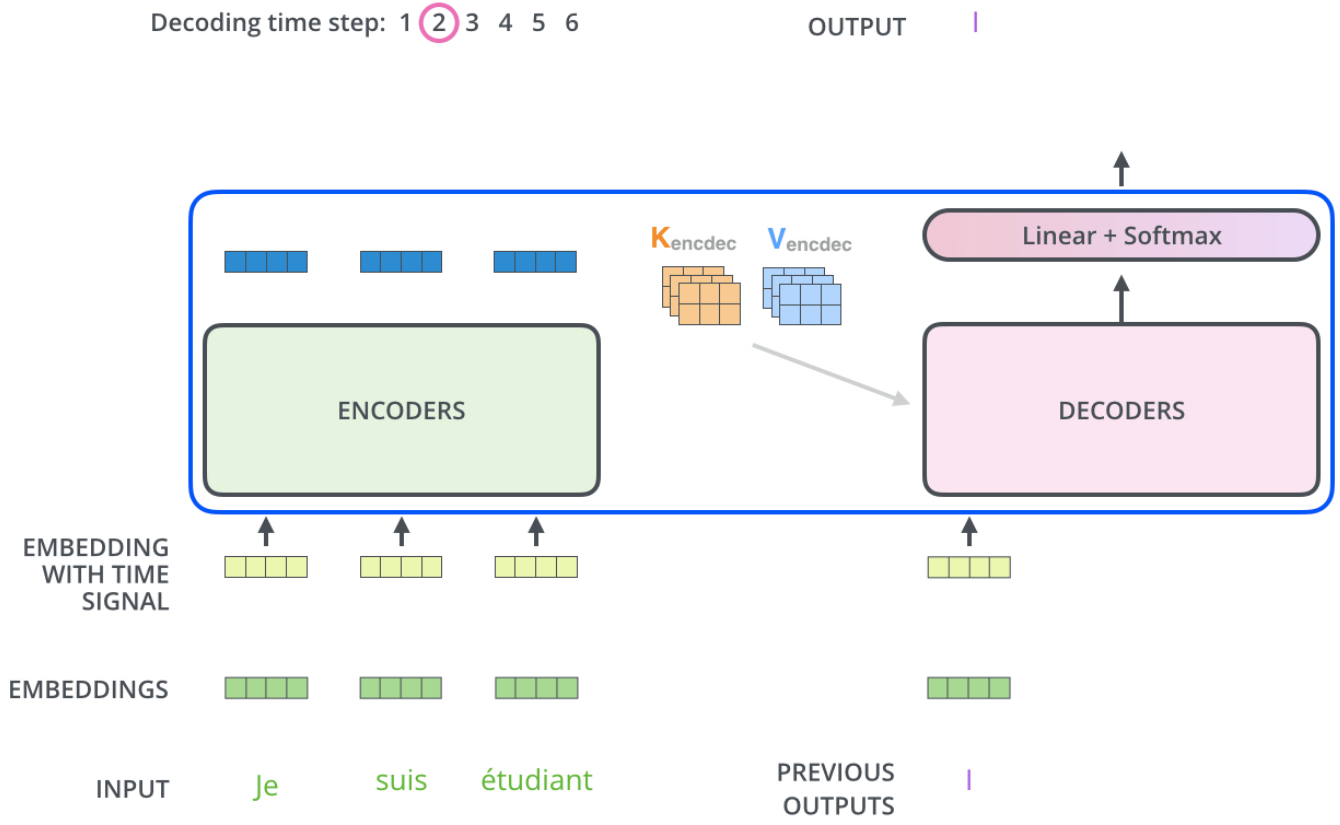
Natural Language Processing Tasks

- Natural Language Processing is the process through which AI is taught to understand the rules and syntax of language, programmed to develop complex algorithms to represent those rules and then made to use those algorithms to carry out specific tasks like these.



Transformer for NLP Tasks

- The Transformer was originally designed as an encoder-decoder model for sequence-to-sequence tasks, like translation.
  - During inference, it generates one word/token at a time and **feeds it back into the model as input for the next word/token.**
  - The process continues until a special end-of-sequence token (<EOS>) is produced or a maximum length is reached.
- Such a paradigm might not be optimal for sentence-level tasks (e.g., sentiment analysis, QA, etc.)
- Can we create a model that incorporates a more comprehensive view of context and serves as a versatile foundation for various NLP tasks?



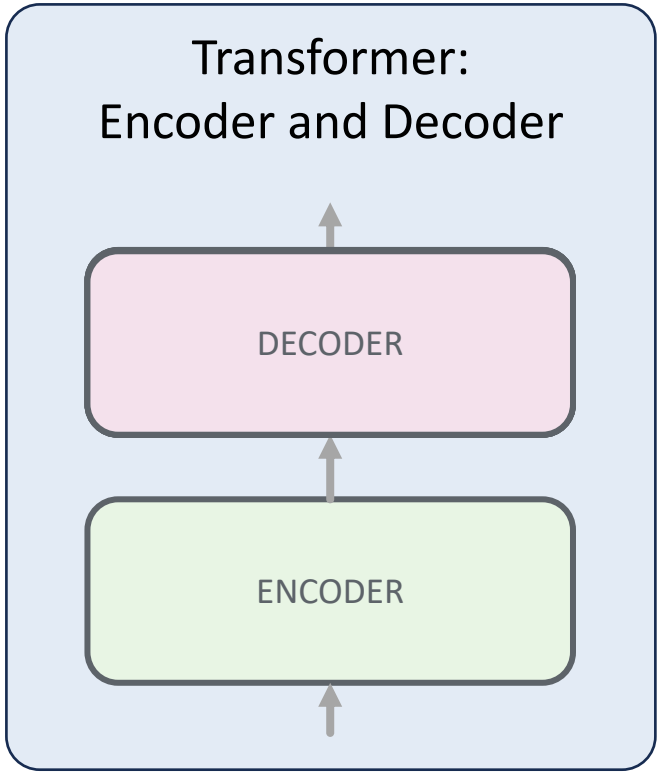
The Need for a General Language Model

- **Goal:** Develop a language model with broad understanding capabilities.
- **Why:** This model can be adapted to various NLP tasks easily, we don't have to retrain a model from scratch every time.
- **How?** This requires language understanding.
  - Pretrain a model that learns universal language patterns.
  - Finetune the language model to learn specific tasks.
- **Pretrained Transformer Models** focuses on the idea of pre-training on vast amounts of generic text data to capture universal language patterns.
  - This allows the models to learn rich contextual representations that can be fine-tuned on specific tasks with minimal data.

"For several years, people have been getting very good results pre-training [Deep Neural Networks] as a language model and then fine-tuning on some downstream NLP tasks (question answering, natural language inference sentiment analysis)"
– BERT author

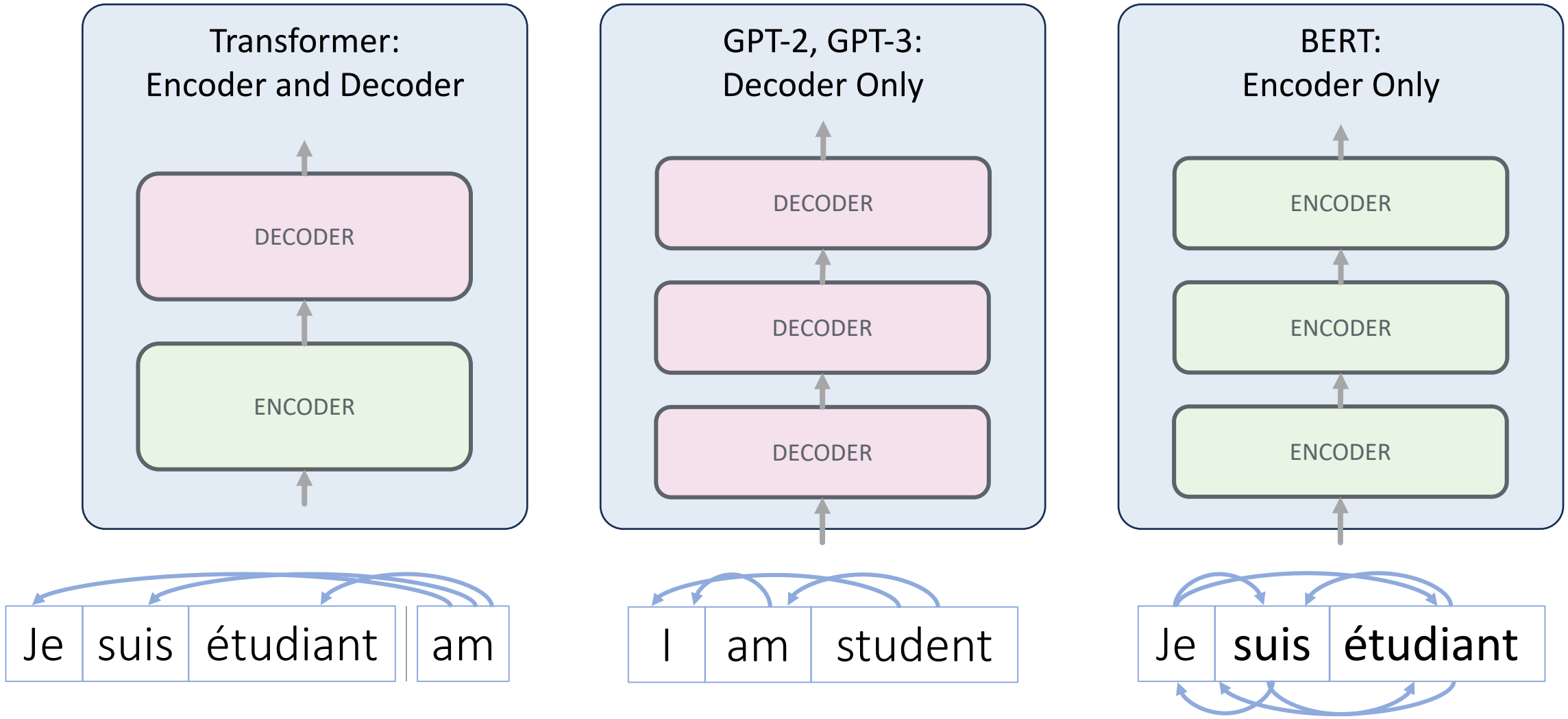
Pretrained Transformer Models

- The transformer architecture inspired the creation of pretrained transformers like BERT and GPT. Both models build directly on the original Transformer architecture but apply it differently.



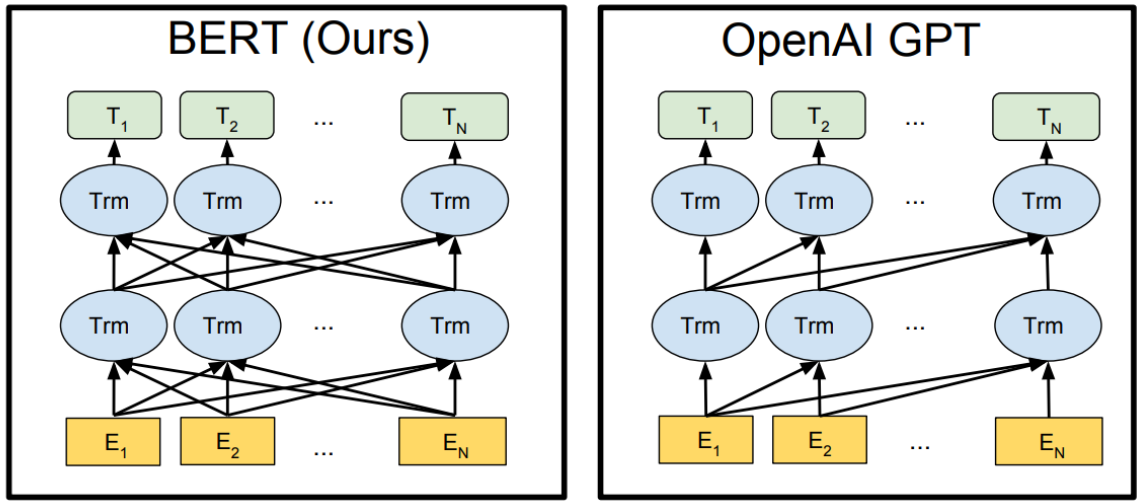
Pretrained Transformer Models

- The transformer architecture inspired the creation of pretrained transformers like BERT and GPT. Both models build directly on the original Transformer architecture but apply it differently.



Bidirectional Encoder Representations from Transformers (BERT)

- BERT is a transformer-based model whose language model is conditioned on both left and right context (bi-directionally).
  - Models like GPT process text left-to-right (uni-directionally).



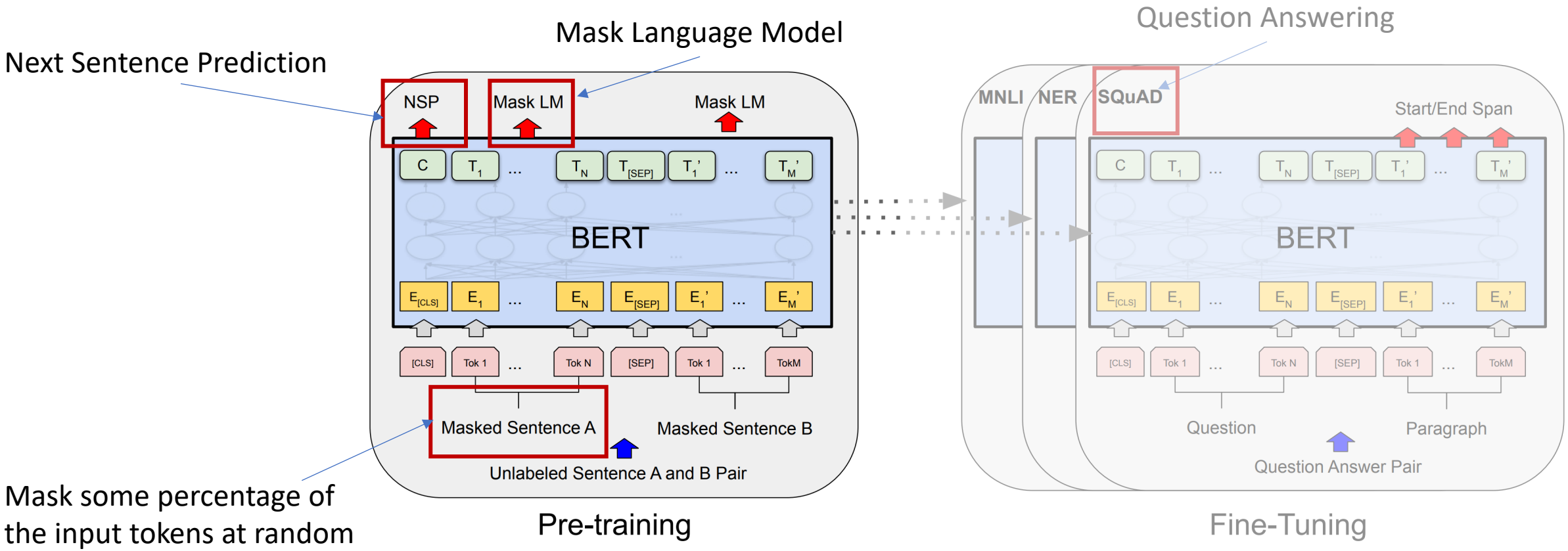
- BERT works on both sentence-level and token-level tasks.

- BERT Training:
  - Pretraining: Understand the language
    - Trained on entire Wikipedia (2.5B words) and BookCorpus (800M words).
  - Finetuning: Learn specific NLP tasks
    - Can be finetuned easily for downstream tasks.
    - Targeted at multi-task objective.

	$BERT_{BASE}$	$BERT_{LARGE}$	Transformer
Layers	12	24	6
Feedforward networks (hidden units)	768	1024	512
Attention heads	12	16	8

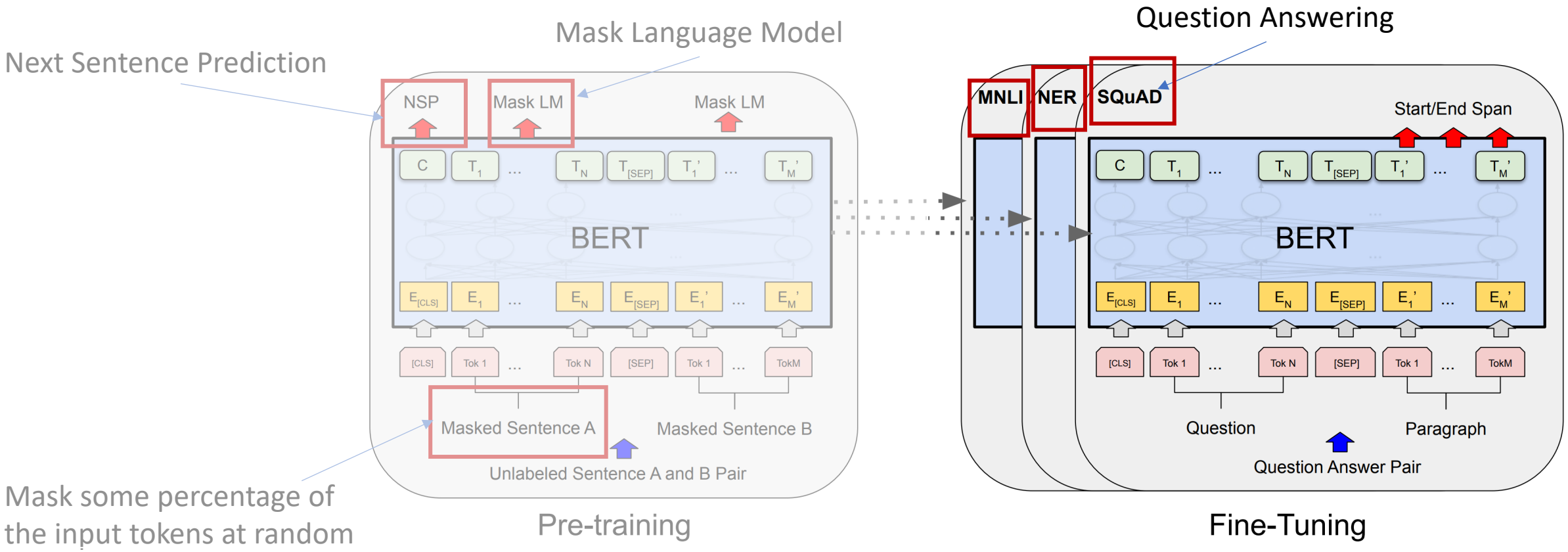
Bidirectional Encoder Representations from Transformers (BERT)

- **Pre-training:** BERT is pre-trained using two main objectives:
  - **Masked Language Modeling (MLM):** Randomly masks words in a sentence and trains the model to predict them, encouraging it to learn context from both directions.
  - **Next Sentence Prediction (NSP):** Trains the model to understand the relationship between two sentences, useful for tasks like question answering and sentence coherence.



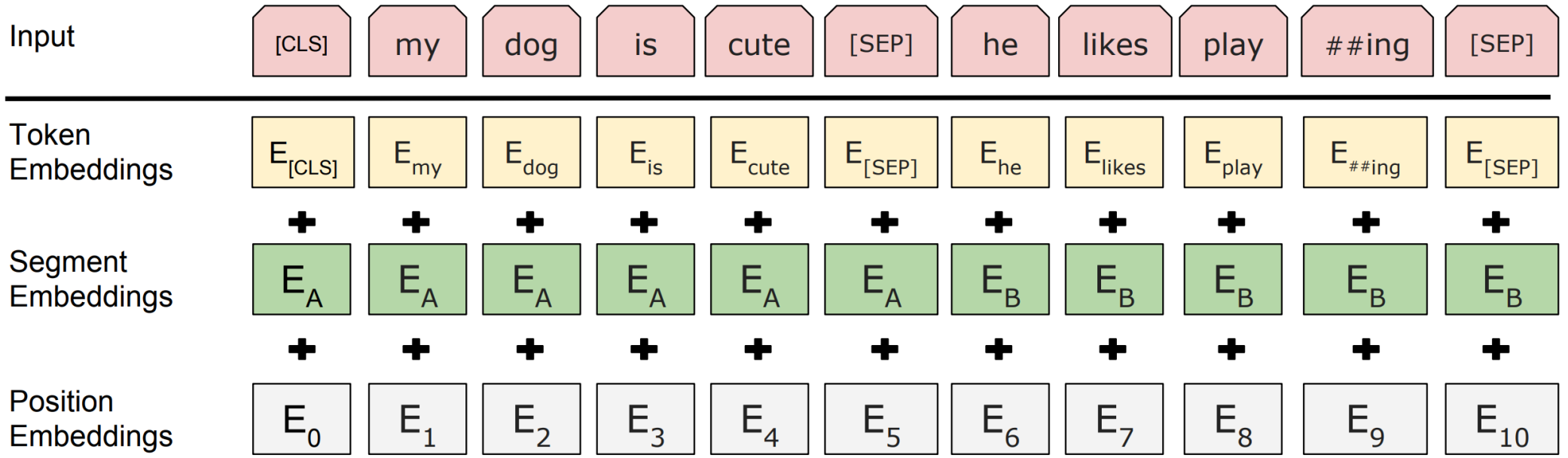
Bidirectional Encoder Representations from Transformers (BERT)

- **Fine-tuning:** Once pre-trained, BERT is fine-tuned on specific NLP tasks with minimal adjustments, making it highly versatile for different applications.
  - Apart from output layers, the same architectures are used in both pre-training and fine-tuning.



Bidirectional Encoder Representations from Transformers (BERT)

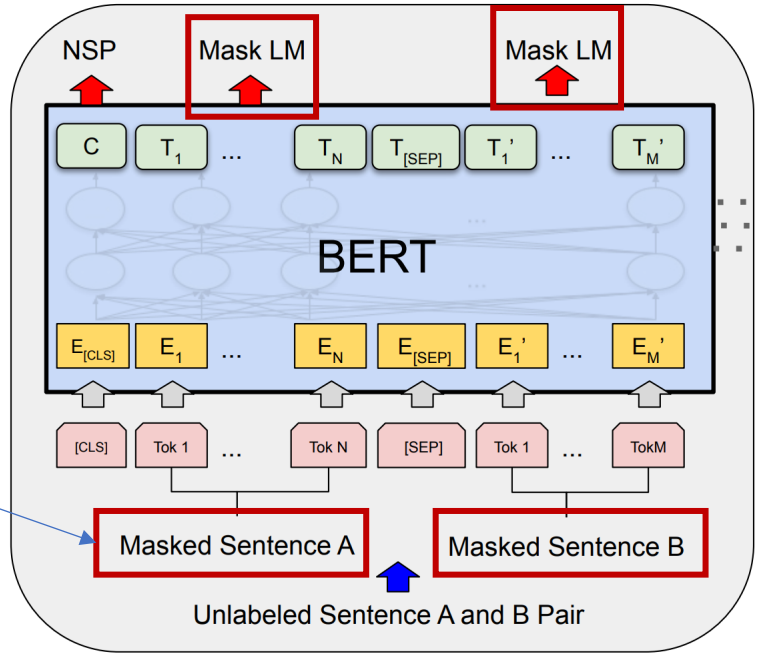
- BERT's input representation combines three types of embeddings: token embeddings (words), segment embeddings (sentence distinction), and position embeddings (word order within the sentence).
- Each input sequence begins with a special [CLS] token, whose final hidden state is used for classification tasks.
- The [SEP] token marks the end of each sentence or segment.



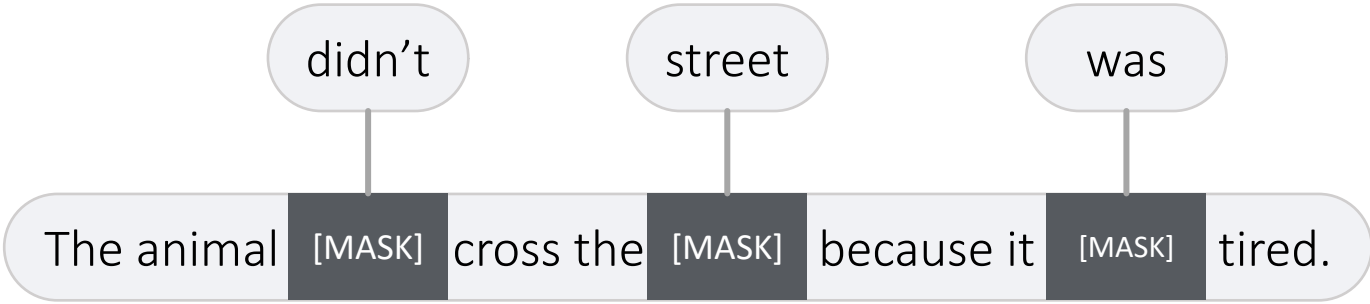
BERT Pre-training Stage: Masked Language Modeling (MLM)

- BERT masks a portion of the input words and trains to predict these masked words using context from the sequence.
  - Typically, 15% of the tokens are selected for masking. Among the candidates: 80% chance to be masked, 10% chance to be altered, 10% chance remain the same.
- The task is reconstructing the token sequence given the masked one.

Mask some percentage of the input tokens at random

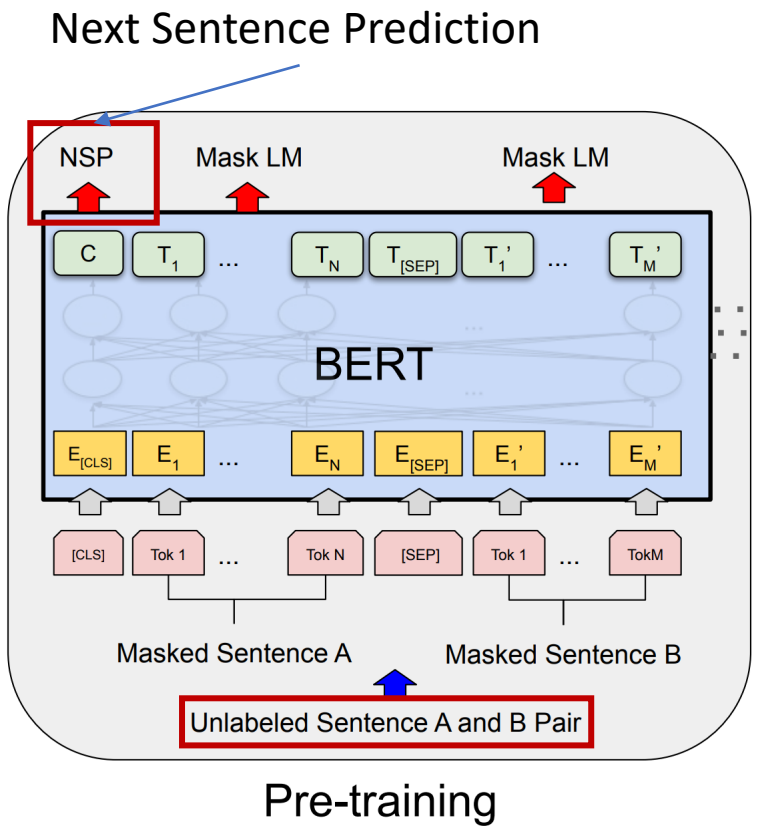


Pre-training



BERT Pre-training Stage: Next Sentence Prediction (NSP)

- NSP helps BERT understand relationships between sentences, which is essential for tasks like question answering and natural language inference.
- Given pair of sentences, it predicts whether the second sentence naturally follows the first. It learns to classify each pair as “Is Next” or “Not Next”.



Sentence A	She studied hard for the final exam.
Sentence B	Thus, she passed with flying colors.

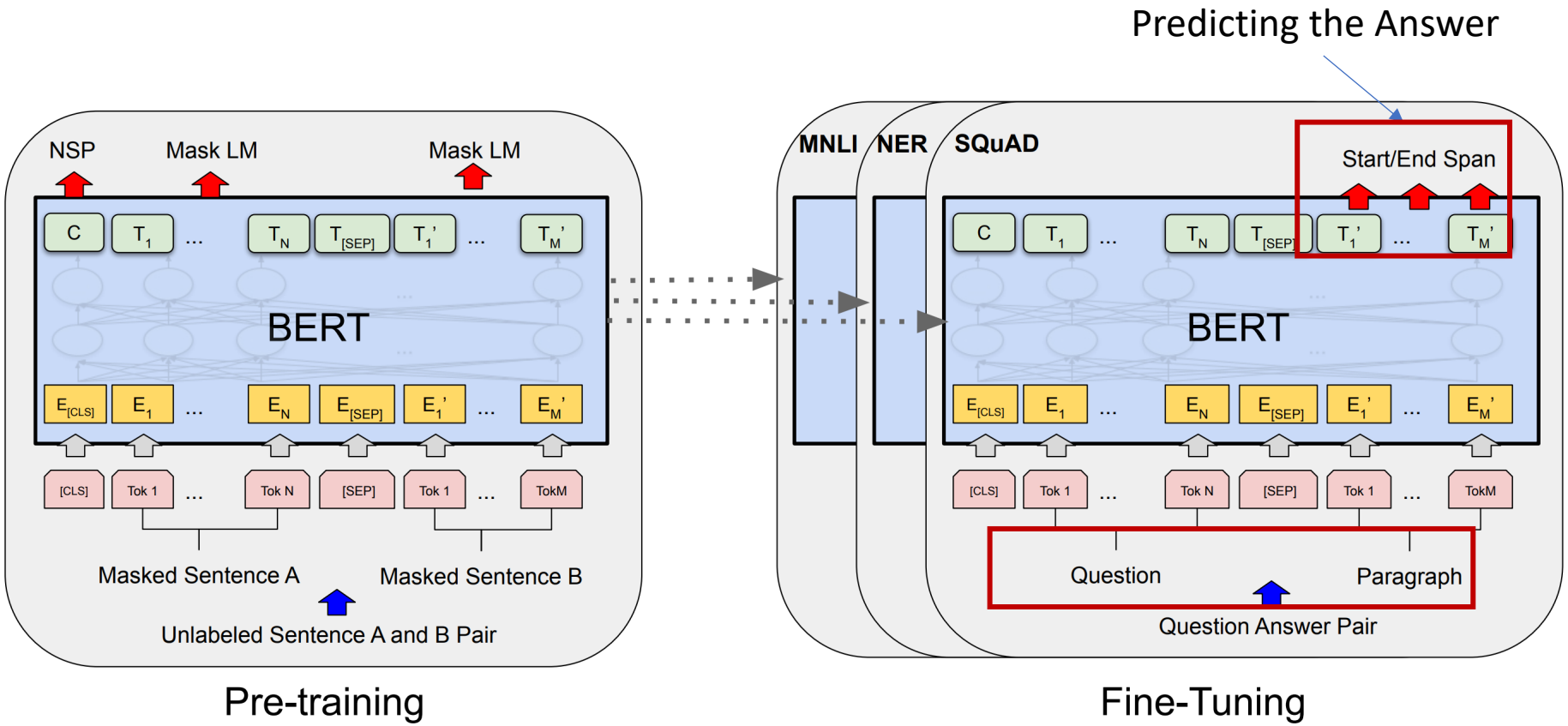
IsNext

Sentence A	The children were playing in the park.
Sentence B	He then started cooking the soup.

NotNext

BERT Fine-tuning Stage

- Fine-tuning adapts BERT's general language understanding (learned during pre-training) to specific NLP tasks like **sentiment analysis**, **question answering**, and **named entity recognition (NER)**.
  - For instance, given a QA dataset that consists of training samples (Question, Paragraph, Answer), we naturally utilize the pre-trained weights to fine-tune BERT.



Finetuning Benchmarks:
MNLI: Multi-Genre Natural Language Inference

NER: Named Entity Recognition

SQuAD: Stanford Question Answering Dataset

BERT: Results

MNLI – Multi-Genre Natural Language Inference

Task: Decide whether the *hypothesis* is **entailment**, **neutral**, or **contradiction** based on the *premise*.

Premise: "The dog is playing in the park." Hypothesis: " The dog is outdoors."	Label: **Entailment**

QQP – Quora Question Pairs

Task: Given two questions, decide whether they have the same meaning.

"How can I become a better public speaker?" "What are some ways to improve my public speaking skills?"	Label: **Paraphrase**

SST – Sentiment Analysis

Task: sentence-level sentiment classification

"The movie drags endlessly and fails to build any tension."	Label: **Negative**

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Can BERT generate sentences?

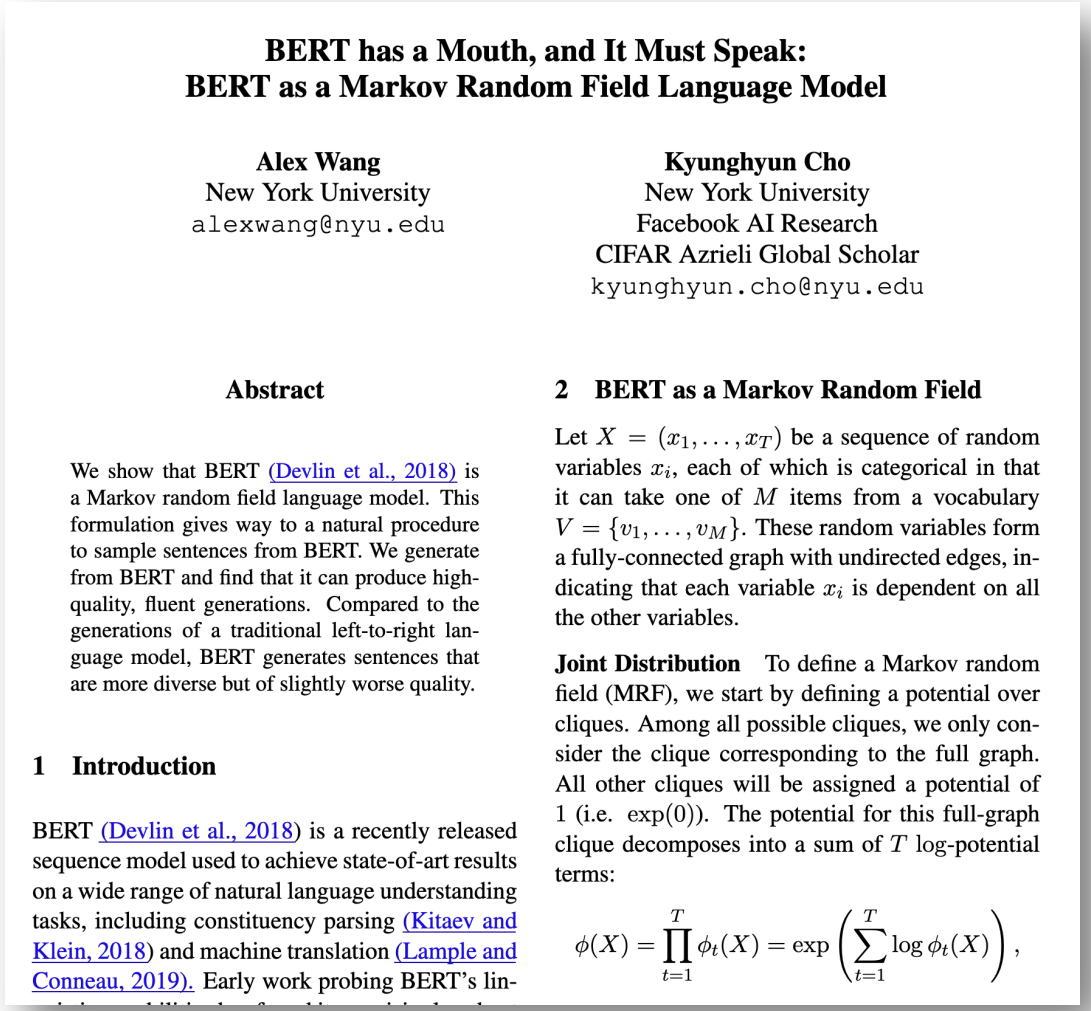
- Sentence generation requires sampling from a language model, which gives the probability distribution of the next word given the previous contexts.
- But BERT has a bidirectional nature and predicts masked tokens given all the other:

$$P(x_{masked}|x_{context}).$$

- However, BERT can be viewed as modeling the joint distribution over all tokens:

$$P(x_1, x_2, \ldots, x_T)$$

  - And once you have a joint distribution, you can sample from it.
- You can start with a sentence of all **[MASK]** tokens and generate words one by one in an arbitrary order.



BERT has a Mouth, and It Must Speak:
BERT as a Markov Random Field Language Model

Alex Wang
New York University
alexwang@nyu.edu

Kyunghyun Cho
New York University
Facebook AI Research
CIFAR Azrieli Global Scholar
kyunghyun.cho@nyu.edu

Abstract

We show that BERT (Devlin et al., 2018) is a Markov random field language model. This formulation gives way to a natural procedure to sample sentences from BERT. We generate from BERT and find that it can produce high-quality, fluent generations. Compared to the generations of a traditional left-to-right language model, BERT generates sentences that are more diverse but of slightly worse quality.

1 Introduction

BERT (Devlin et al., 2018) is a recently released sequence model used to achieve state-of-art results on a wide range of natural language understanding tasks, including constituency parsing (Kitaev and Klein, 2018) and machine translation (Lample and Conneau, 2019). Early work probing BERT's lin-

2 BERT as a Markov Random Field

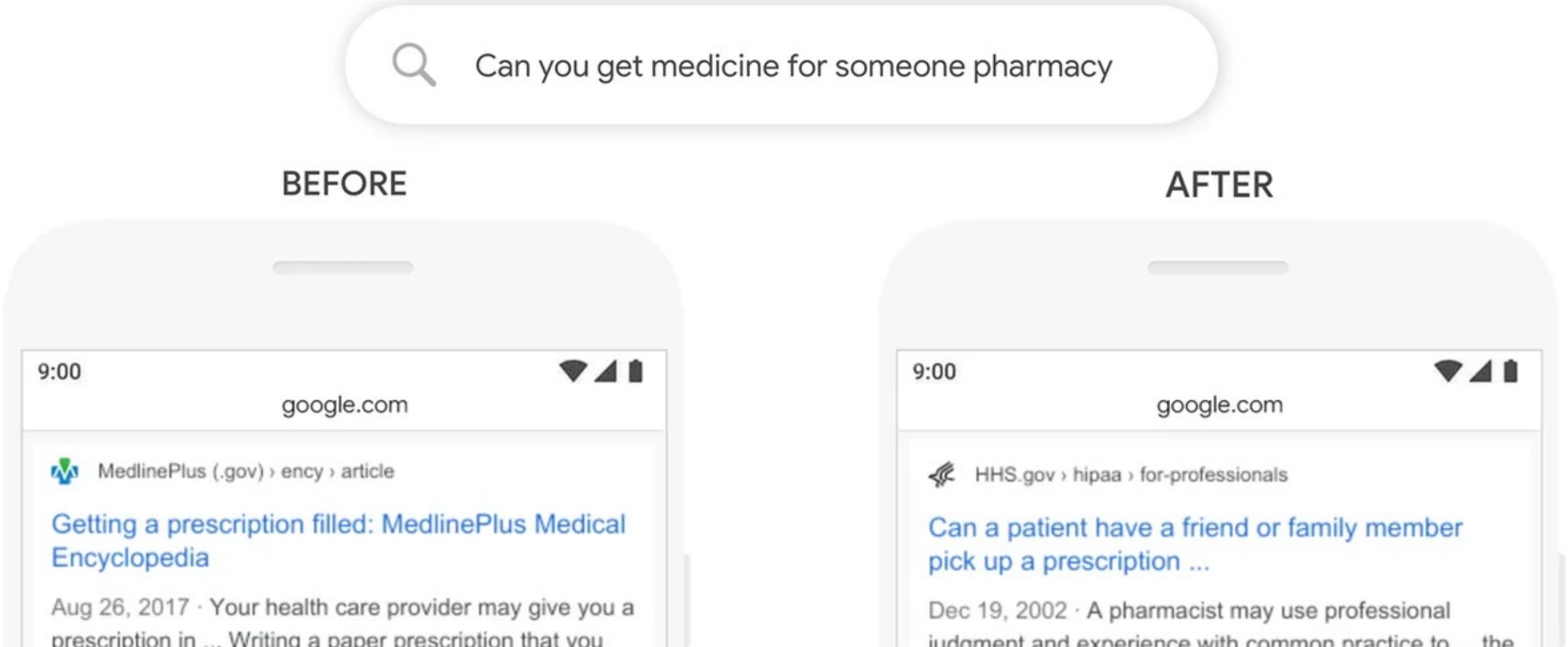
Let $X = (x_1, \ldots, x_T)$ be a sequence of random variables x_i, each of which is categorical in that it can take one of M items from a vocabulary $V = \{v_1, \ldots, v_M\}$. These random variables form a fully-connected graph with undirected edges, indicating that each variable x_i is dependent on all the other variables.

Joint Distribution To define a Markov random field (MRF), we start by defining a potential over cliques. Among all possible cliques, we only consider the clique corresponding to the full graph. All other cliques will be assigned a potential of 1 (i.e. $\exp(0)$). The potential for this full-graph clique decomposes into a sum of T log-potential terms:

$$\phi(X) = \prod_{t=1}^{T} \phi_t(X) = \exp\left(\sum_{t=1}^{T} \log \phi_t(X)\right),$$

BERT: Conclusion

- BERT inspired many derivative architectures and paved the way for larger models.
- BERT improved search and Q&A; Google reported gains in query understanding.



Next Lecture: Transformers for Vision

Natural Language Processing

- **BERT (Bidirectional Encoder Representations from Transformers)**
- GPT (Generative Pre-trained Transformer)
- RoBERTa (Robustly Optimized Bert Pre-training)
- T5 (Text-to-Text Transfer Transformer)

Computer Vision

- **ImageGPT**
- **Vision Transformer**
- Swin Transformer, Pyramid Vision Transformer