CS60075
Natural Language Processing
Autumn 2020

Module 9:

Part 2

6 November 2020

Where to get BERT?

- The Transformers library: https://github.com/huggingface/transformers

- Provides state-of-the-art implementation of many models, including BERT and RoBERTa

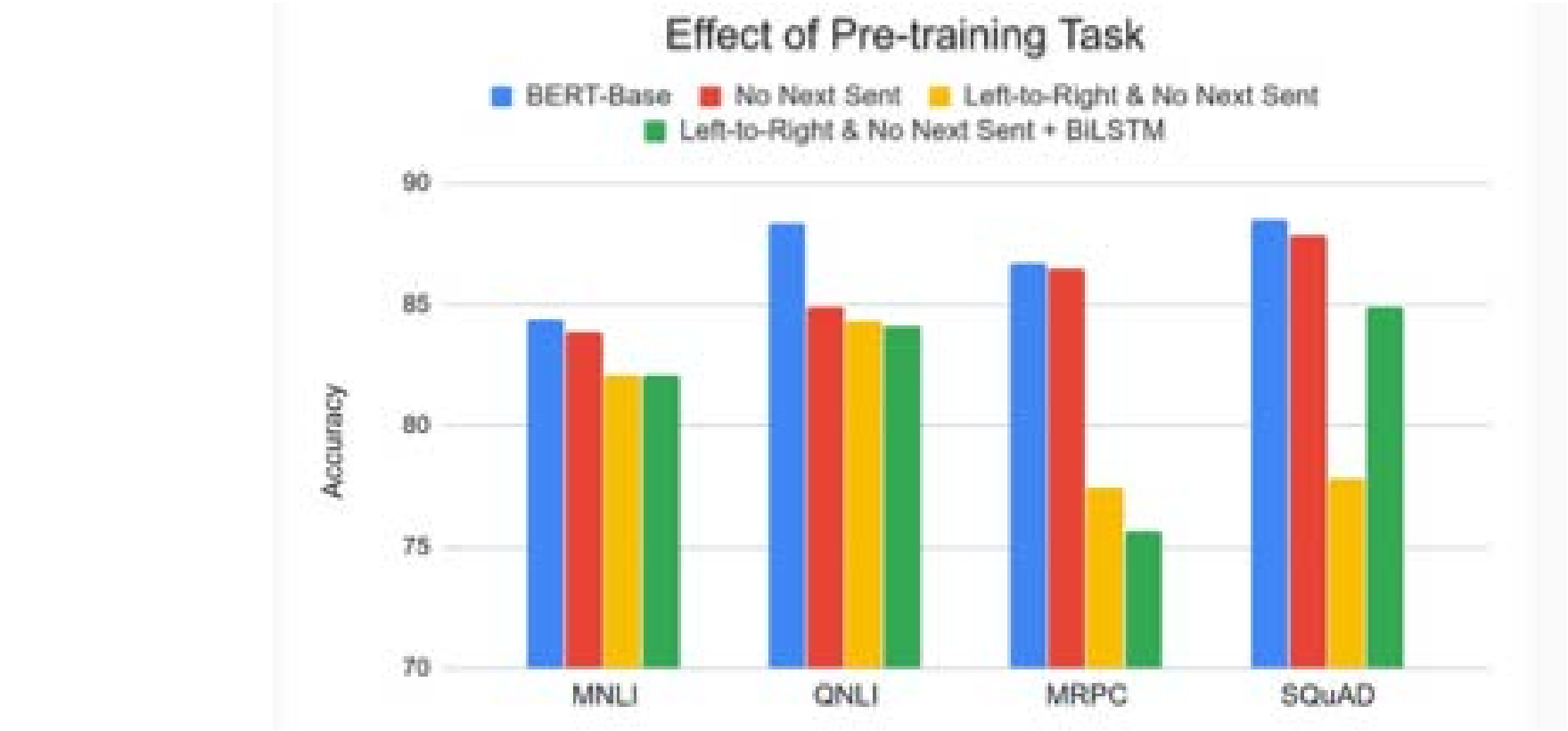
- Including pre-trained models

Hard to do with BERT

- BERT cannot generate text (at least not in an obvious way)

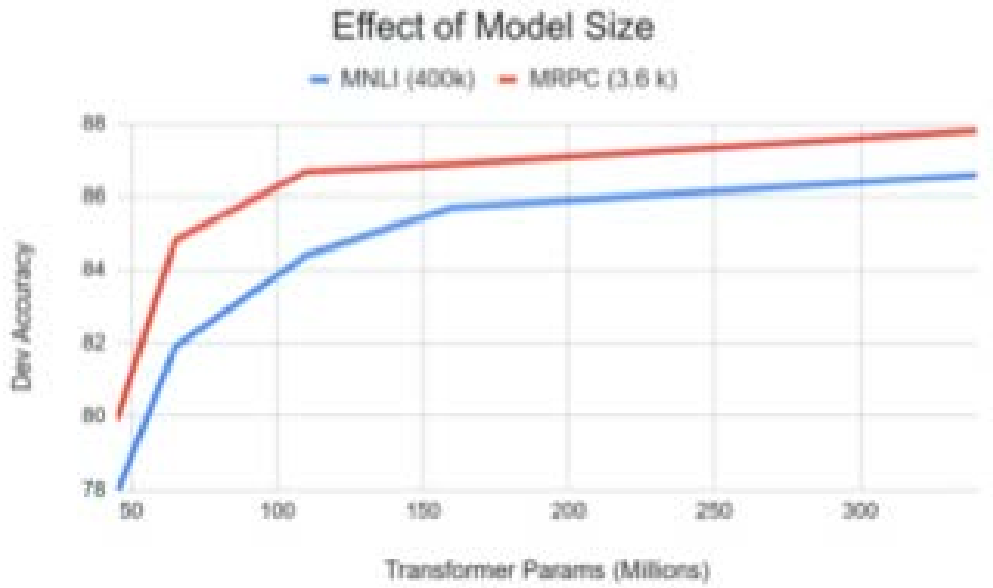
- Masked language models are intended to be used primarily for "analysis" tasks

Effect of pre-training tasks



- Masked LM (compared to left-to-right LM) is very important on some tasks, Next Sentence Prediction is important on other tasks.
- Left-to-right model does very poorly on word-level task (SQuAD), although this is mitigated by BiLSTM

Effects of Model Size



- Big models help *a lot*
- Going from 110M -> 340M params helps even on datasets with 3,600 labelled examples
- Improvements have *not* asymptoted

Slide from Jacob Delvin

References

- The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
- BERT

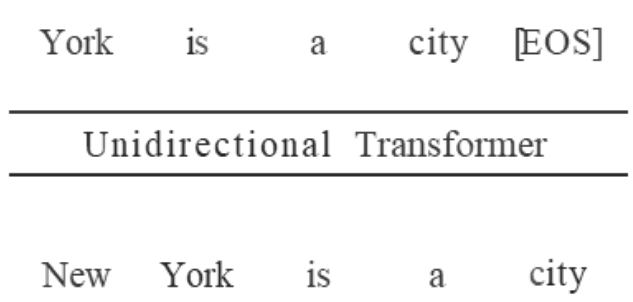
Improved variants of BERT

RoBERTa

- Bigger batch size is better!
- Next sentence prediction doesn't help
- Pretrain on more data for as long as possible!

Two Notable Objectives for Language Pretraining

Auto-regressive Language Modeling

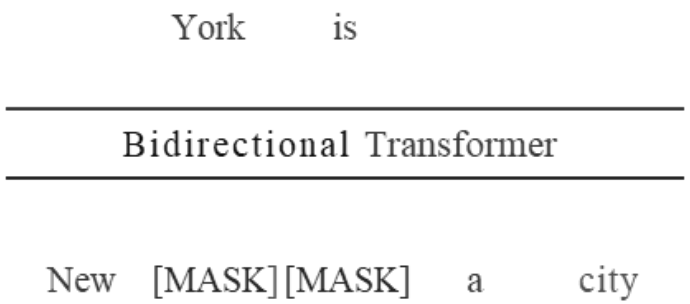


Full Auto-regressive Dependence

Free from artificial Noise

@ No Bidirectional Context

Denoising Auto-encoding (BERT)



@ Independent Predictions

@ Artificial Noise: [MASK]

Natural Bidirectional Context

5

XLNet Key Ideas

- Autoregressive: use context to predict the next word
- Bidirectional context from permutation language modeling
- Self-attention mechanisms, uses Transformer-XL backbone

Bidirectional context from permutation language modeling

Peter's cat likes yarn

- Peter's cat likes yarn
- Peter's cat yarn likes
- Peter's likes cat yarn
- Peter's likes yarn cat
- Peter's yarn cat likes
- Peter's yarn likes cat
- yarn Peter's cat likes
- yarn Peter's likes cat
- yarn cat Peter's likes
- yarn cat likes Peter
- yarn likes Peter's cat
- yarn likes cat Peter's

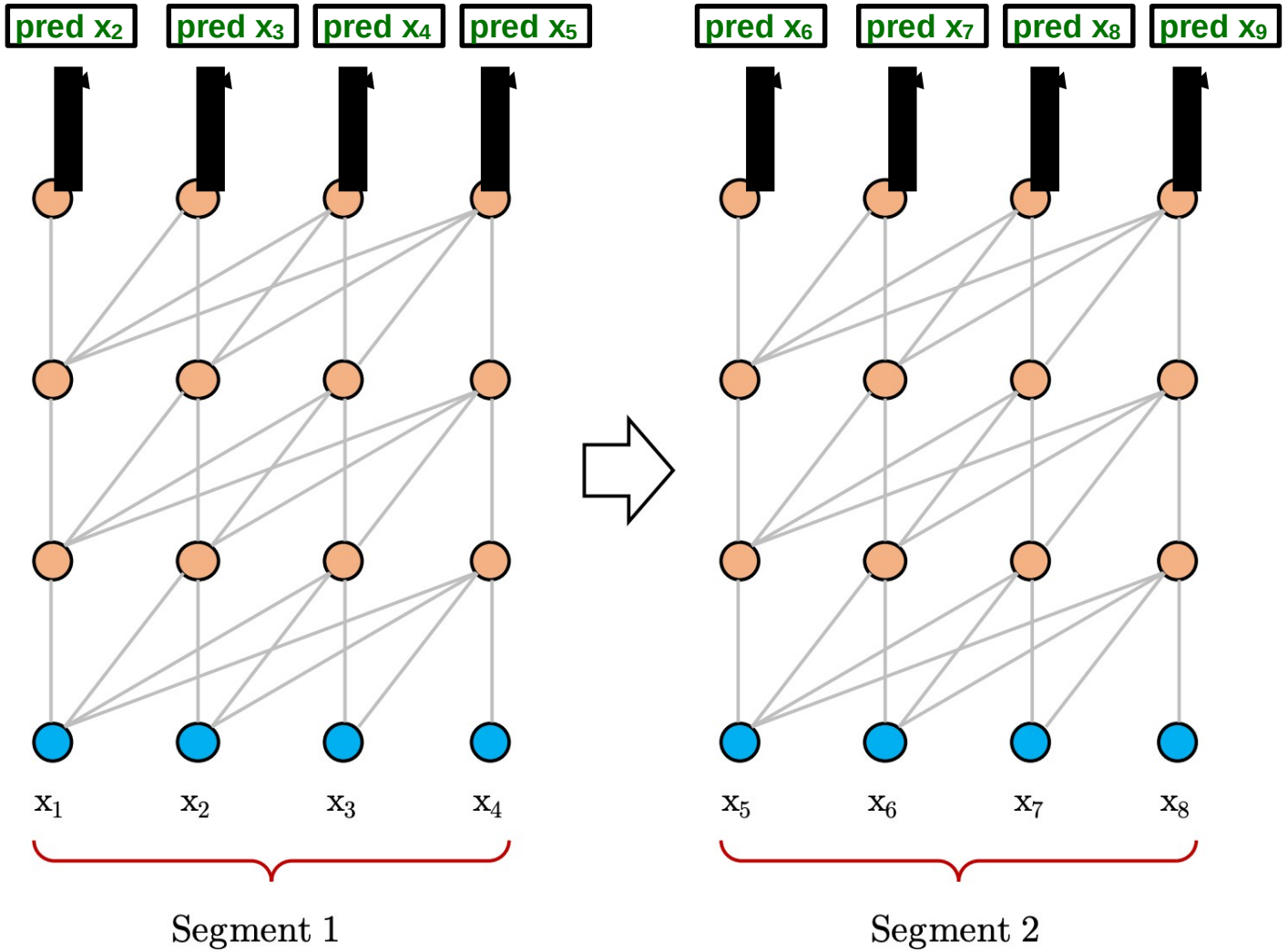
Transformer-XL

- Increases context through segment-level recurrence and a novel positional encoding scheme
  - Cache and reuse hidden state from the previous segment
  - Allows variable-length context, great for capturing long-term dependencies
  - Resolves the problem of context fragmentation
- Increases context through segment-level recurrence and a novel positional encoding scheme

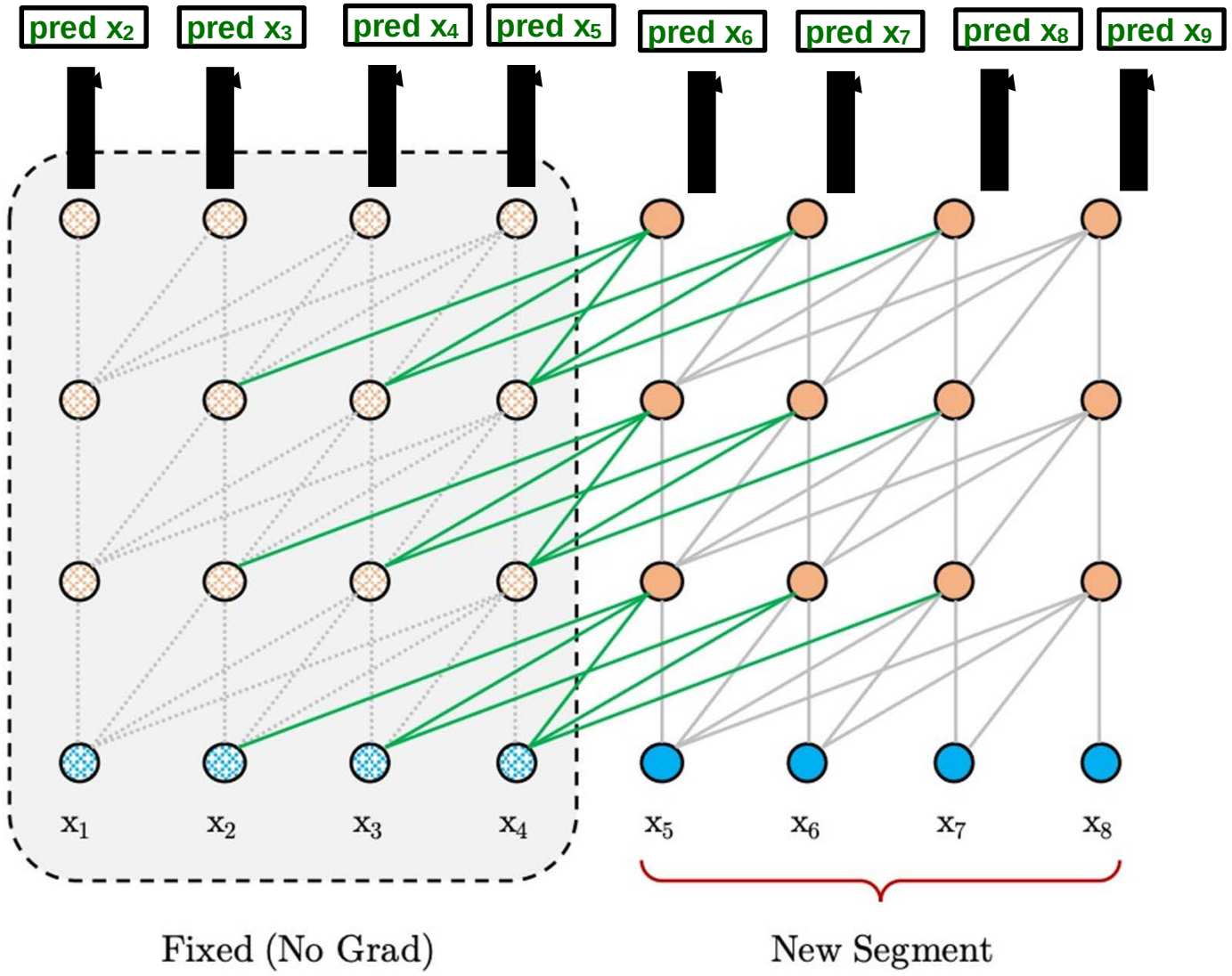
Need a way to keep positional information coherent when we reuse the states

  - Need to encode relative position

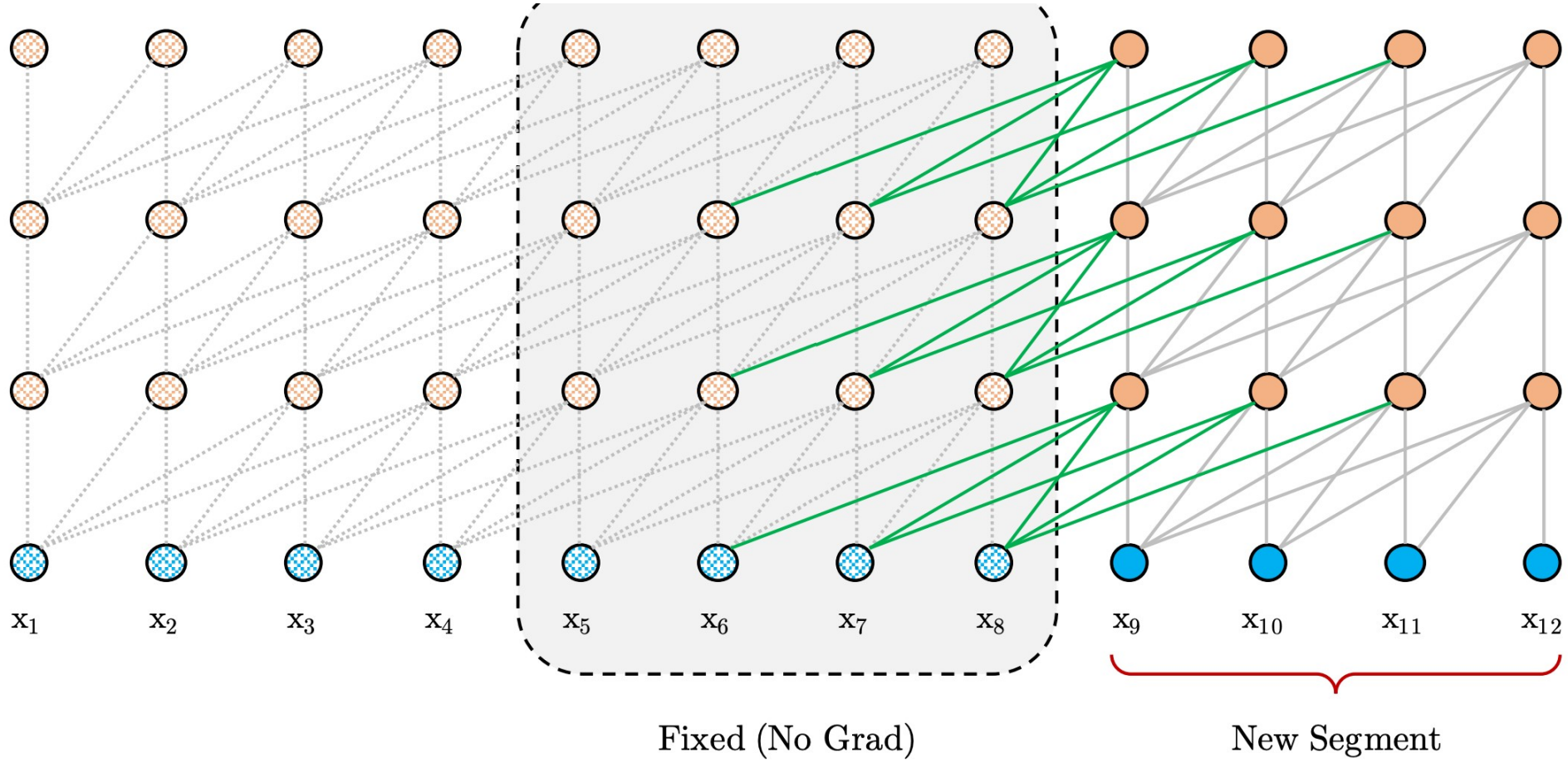
Standard Transformer LM



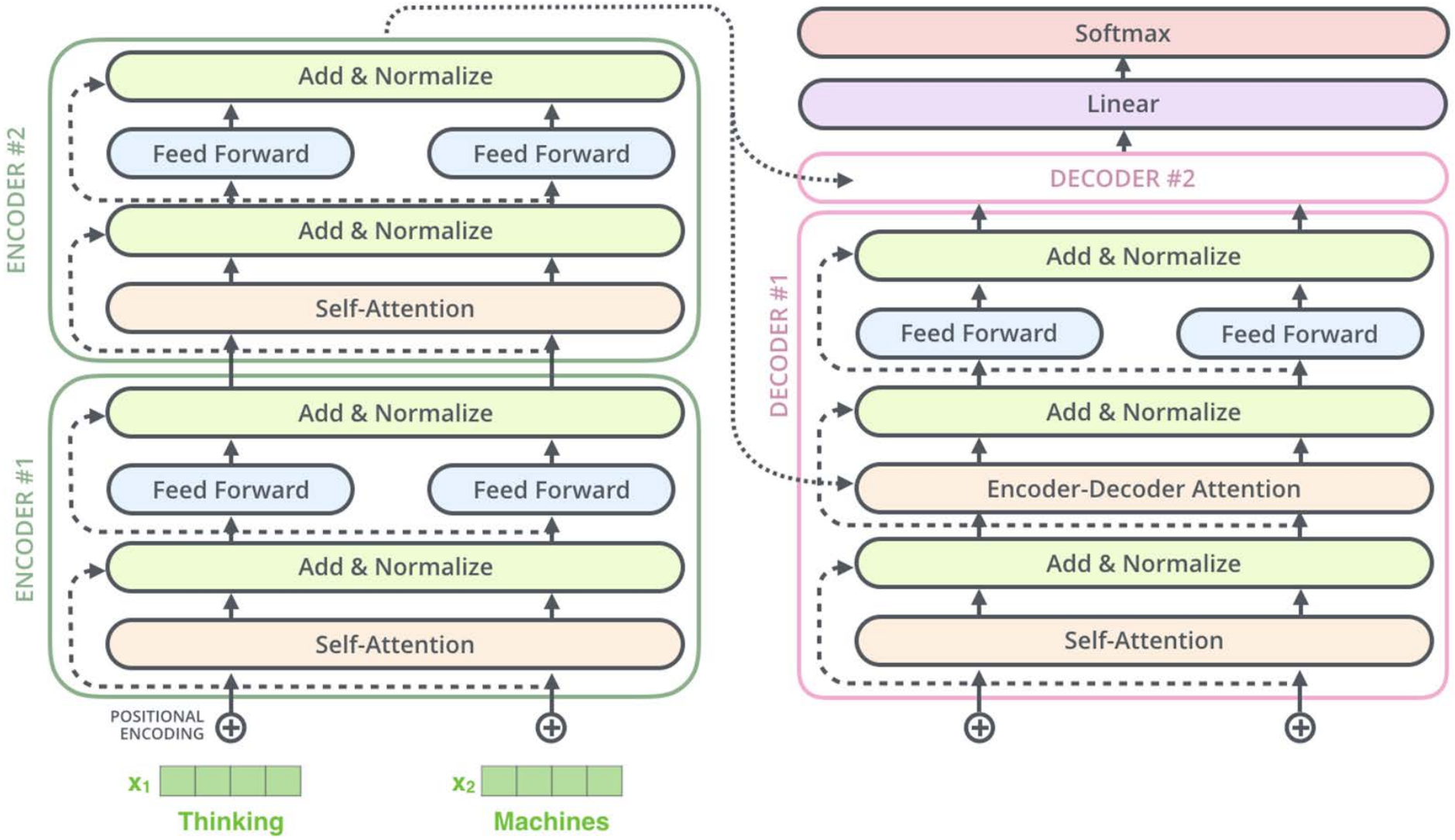
TransformerXL



TransformerXL



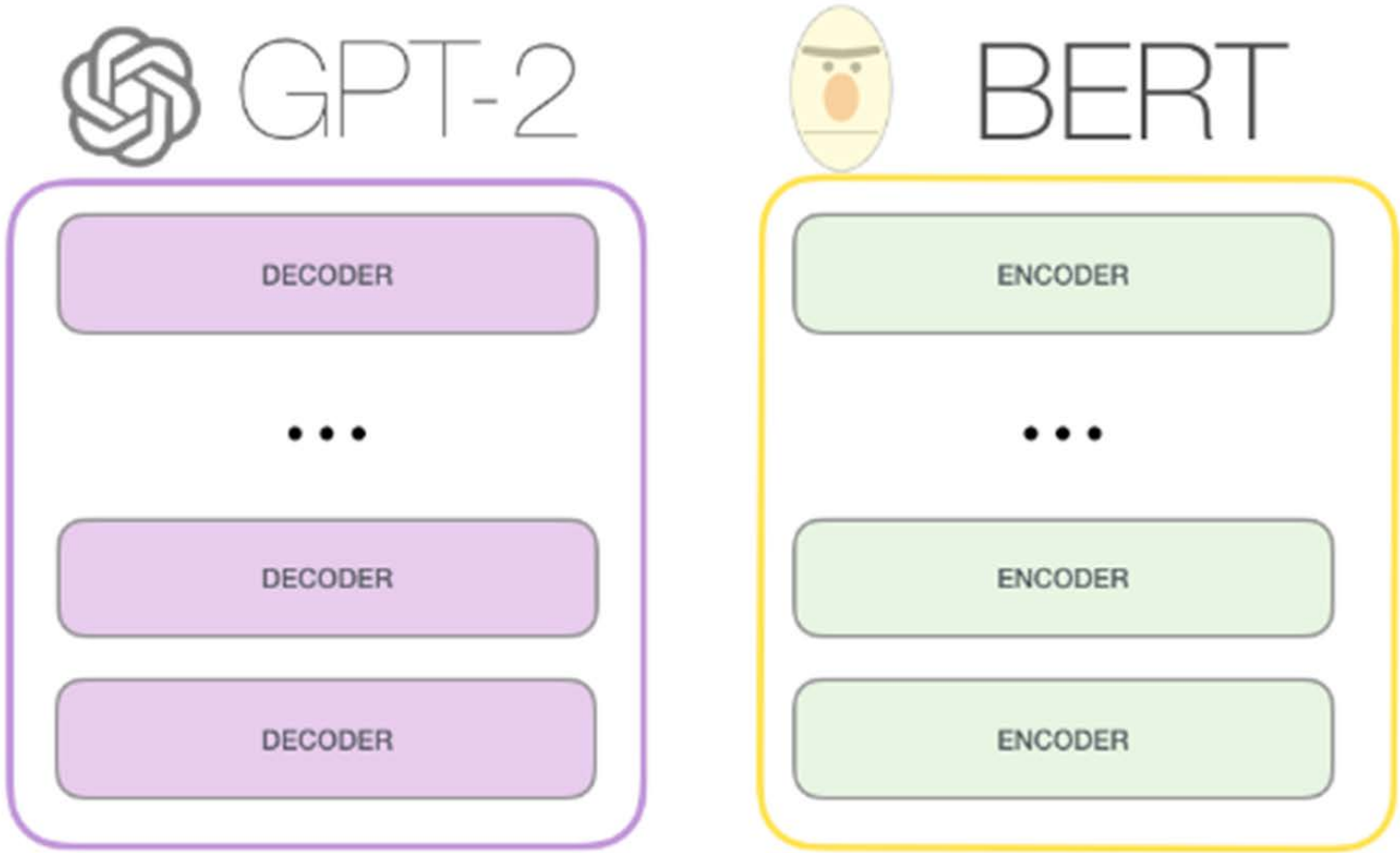
Transformer



GPT-2, BERT

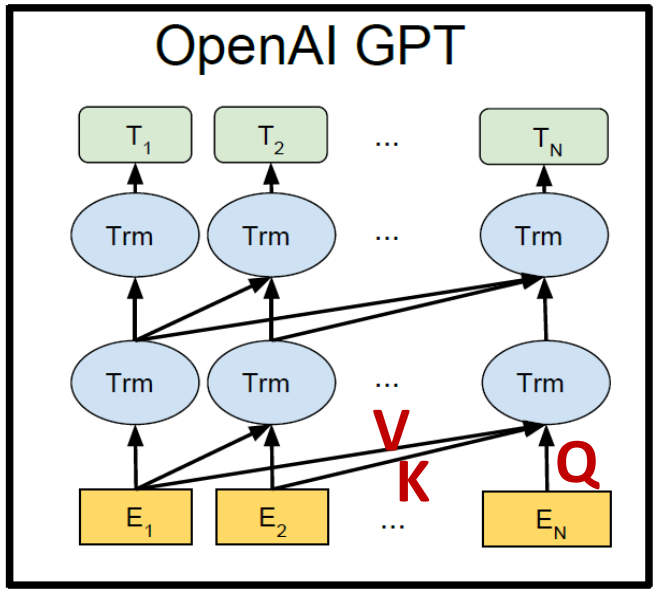
- Transformers, GPT-2, and BERT
  1. A transformer uses Encoder stack to model input, and uses Decoder stack to model output (using input information from encoder side).
  2. But if we do not have input, we just want to model the “next word”, we can get rid of the Encoder side of a transformer and output “next word” one by one. This gives us GPT.
  3. If we are only interested in training a language model for the input for some other tasks, then we do not need the Decoder of the transformer, that gives us BERT.

GPT-2, BERT

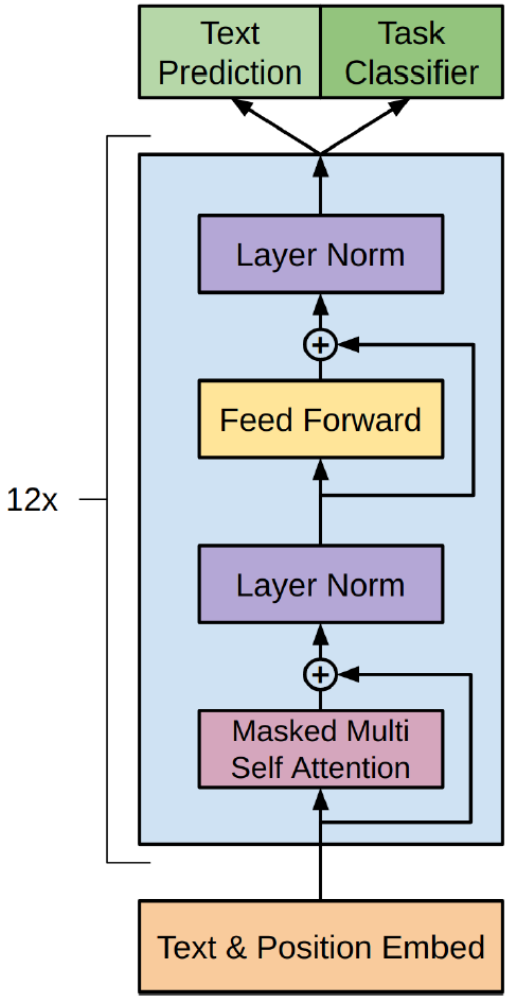

GPT-2
DECODER
DECODER
DECODER
BERT
ENCODER
ENCODER
ENCODER

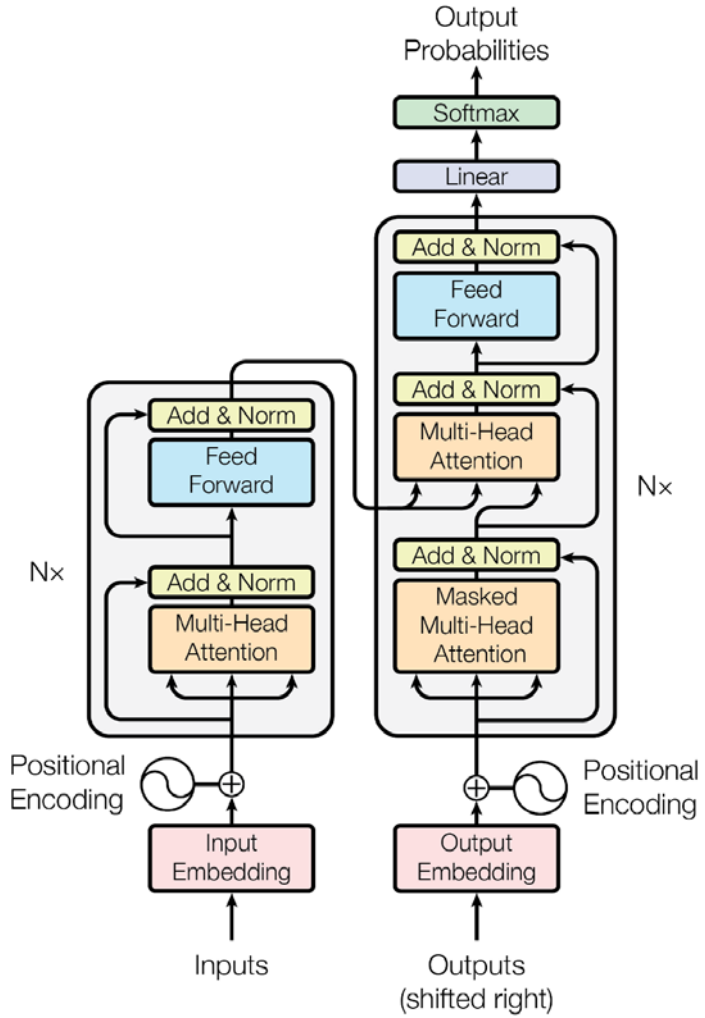
GPT

Generative Pre-Training (OpenAI) is a transformer-based generator using only a simplified transformer decoder stage:



GPT





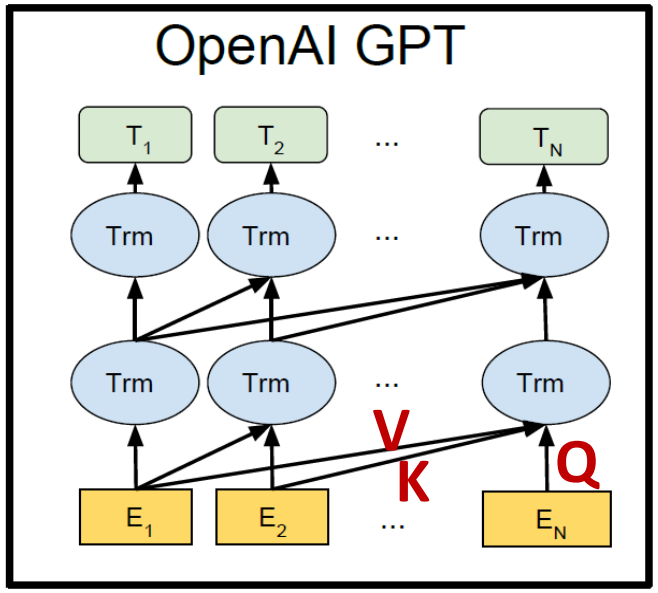
Full transformer

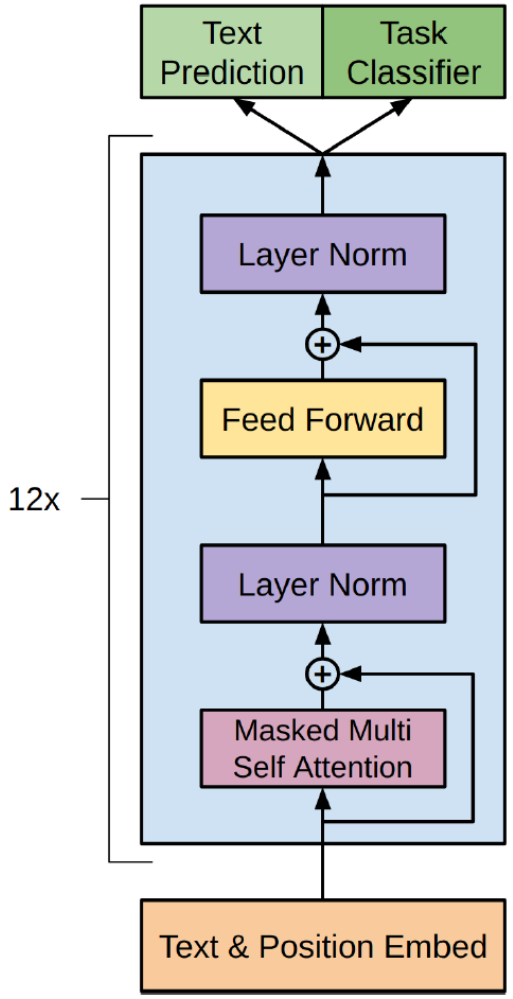
OpenAI GPT-2 implementation:

https://github.com/openai/gpt-2

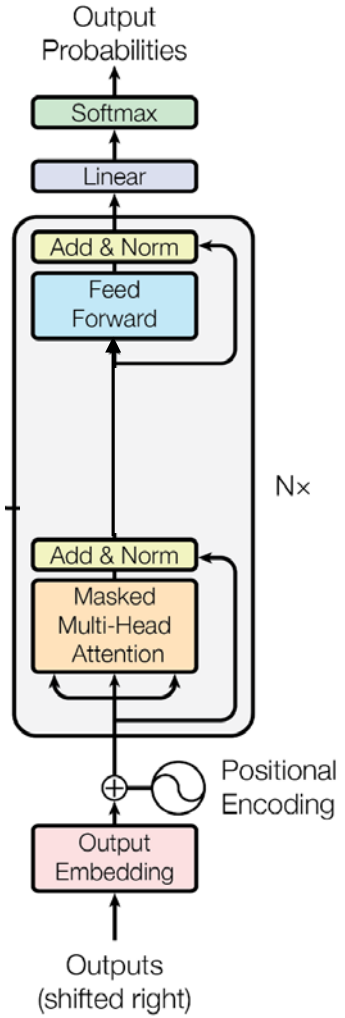
GPT

Generative Pre-Training (OpenAI) is a transformer-based generator using only a simplified transformer decoder stage:





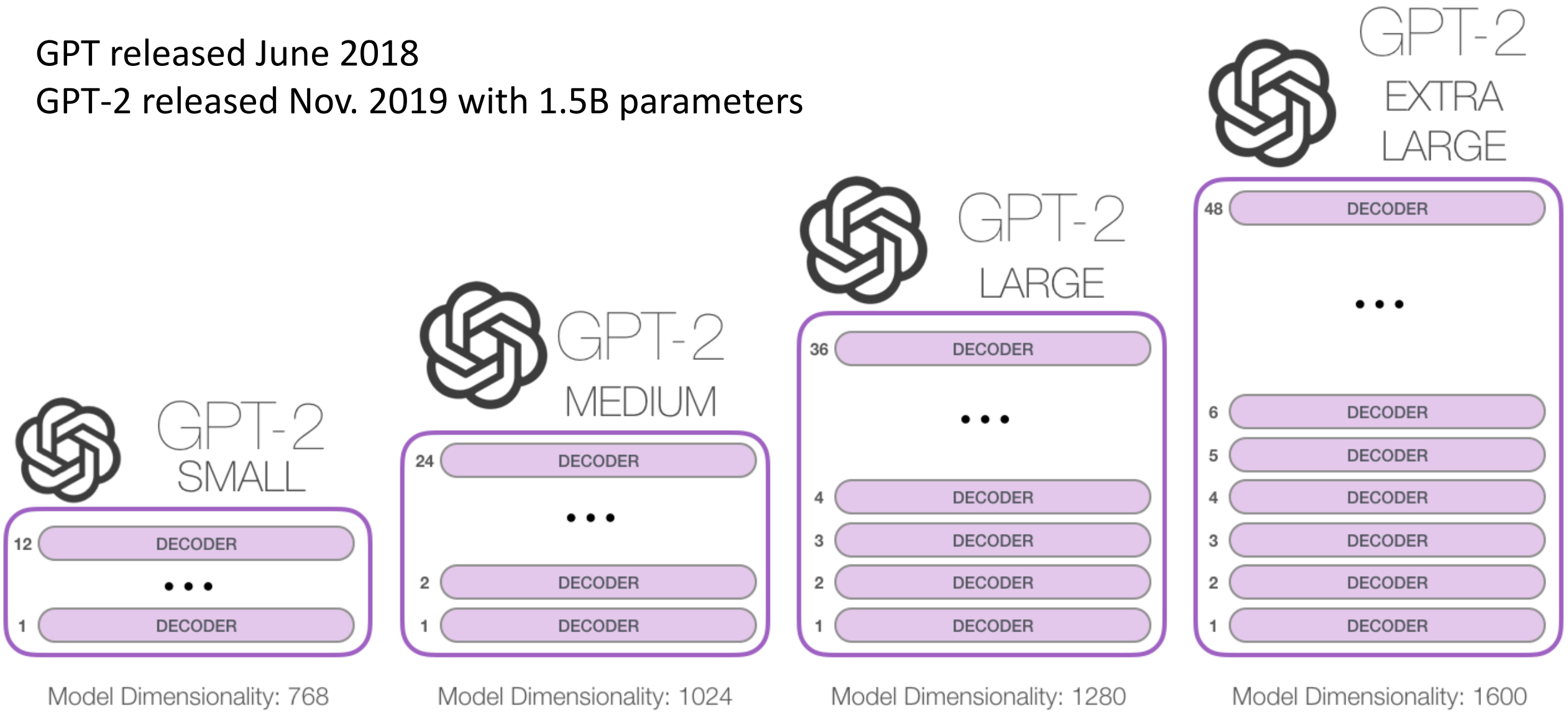
GPT



Full transformer

GPT released June 2018

GPT-2 released Nov. 2019 with 1.5B parameters



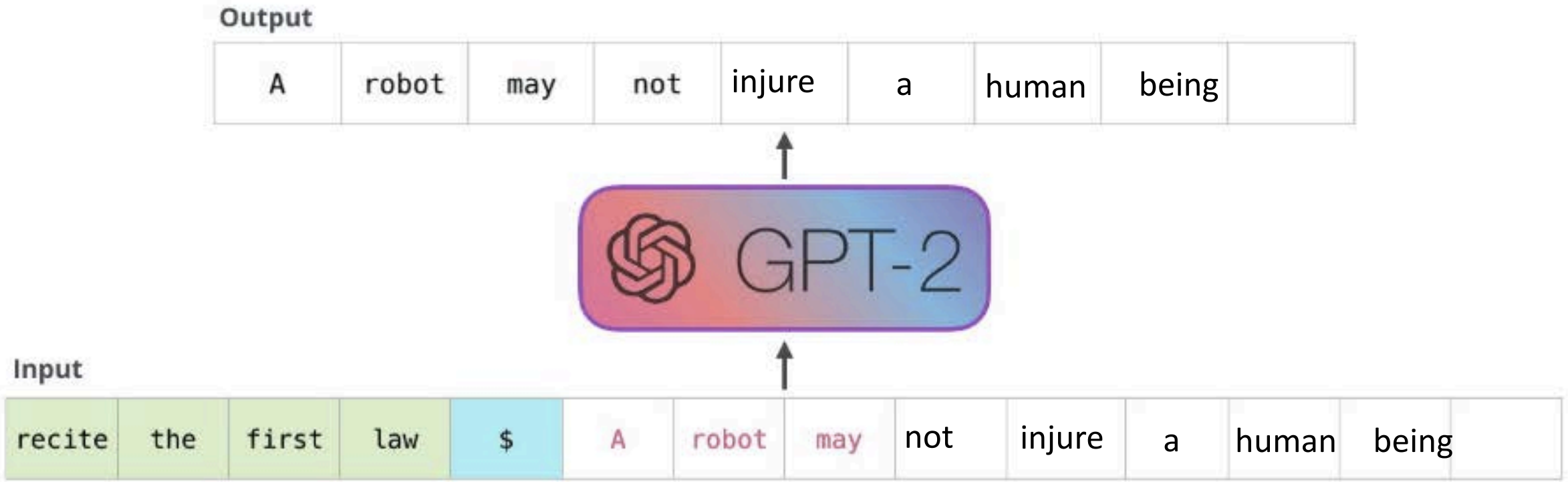
117M parameters

345M

762M

1542M

GPT-2 in action



GPT-2, BERT

Byte Pair Encoding (BPE)

- Word embedding sometimes is too high level, pure character embedding too low level. For example, if we have learned

  old older oldest

- We might also wish the computer to infer

  smart smarter smartest

- But at the whole word level, this might not be so direct. Thus the idea is to break the words up into pieces like er, est, and embed frequent fragments of words.

- GPT adapts this BPE scheme.

GPT-2, BERT

Byte Pair Encoding (BPE)

GPT uses BPE scheme. The subwords are calculated by:

1. Split word to sequence of characters (add </w> char)
2. Joining the highest frequency pattern.
3. Keep doing step 2, until it hits the pre-defined maximum number of sub-words or iterations.

Example:

{‘l o w </w>’: 5, ‘l o w e r </w>’: 2, ‘n e w e s t </w>’: 6, ‘w I d e s t </w>’: 3 }
{‘l o w </w>’: 5, ‘l o w e r </w>’: 2, ‘n e w es t </w>’: 6, ‘w I d es t </w>’: 3 }
{‘l o w </w>’: 5, ‘l o w e r </w>’: 2, ‘n e w est </w>’: 6, ‘w I d est </w>’: 3 }
{‘l o w </w>’: 5, ‘l o w e r </w>’: 2, ‘n e w est</w>’: 6, ‘w I d est</w>’: 3 }
…..

Note that </w> is also an important character.

GPT

GPT is trained initially on a large text corpus to minimize a language modeling loss

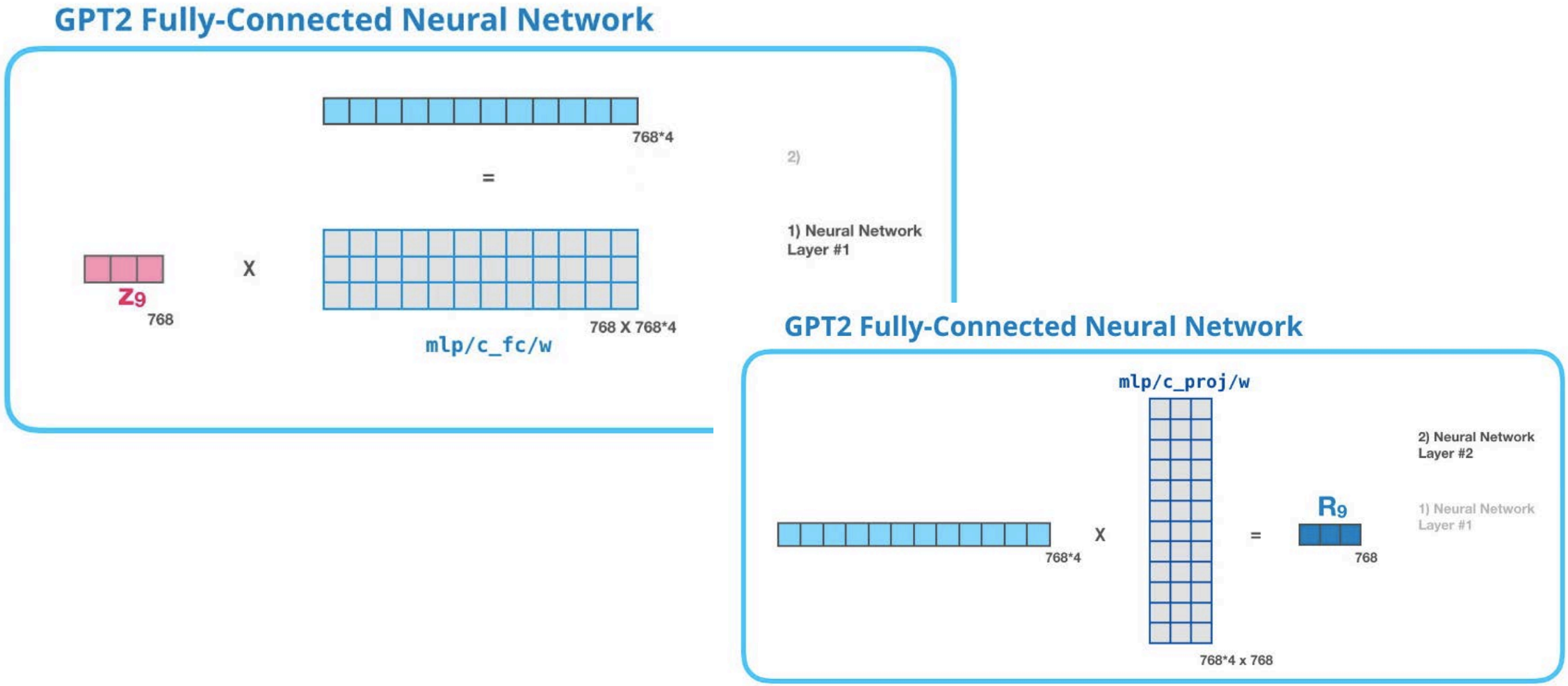
$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

Then for each task, a custom linear layer is added and the entire network retrained (with slower adjustment of the transformer weights).

The language model loss is retained during task-specific retraining of the model.

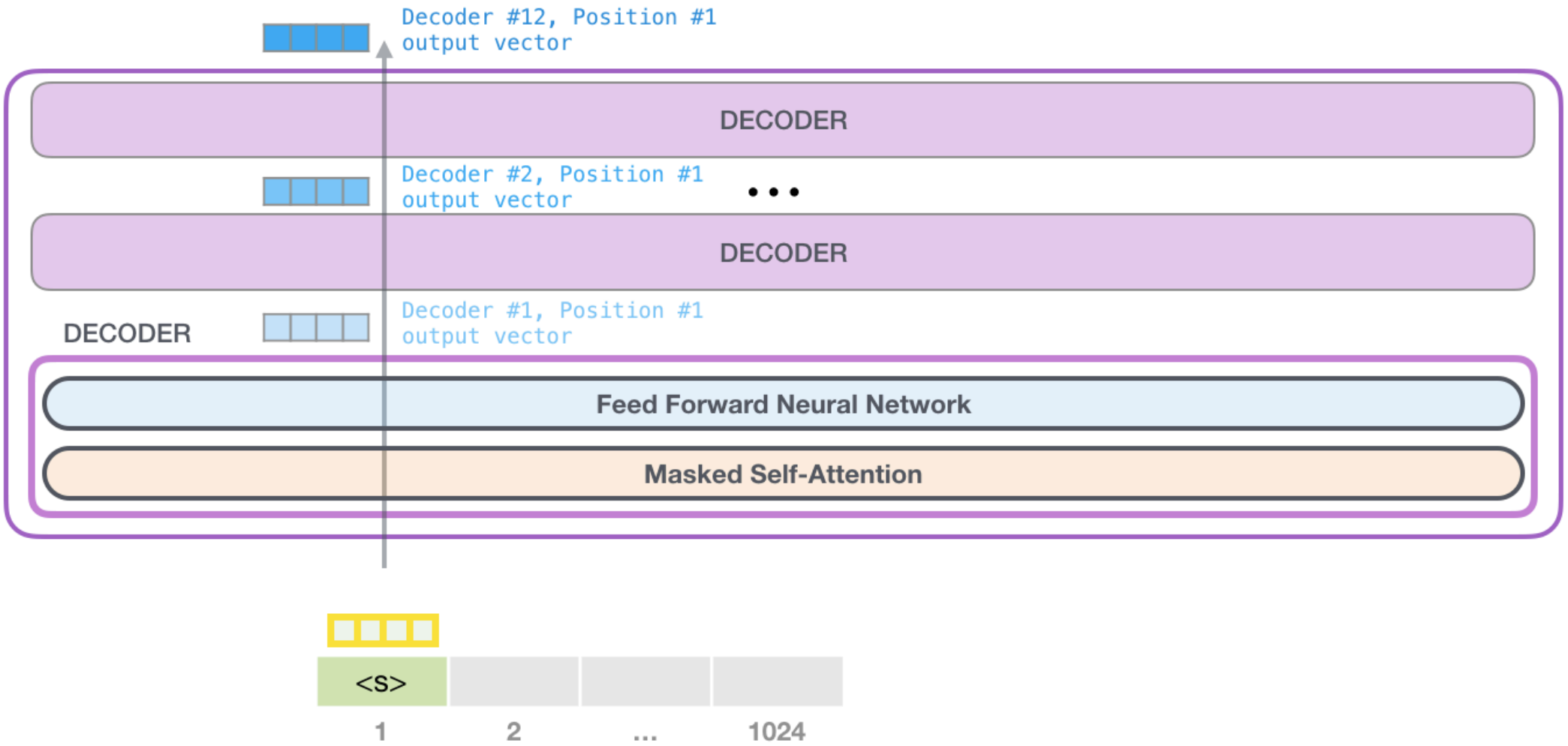
GPT-2 fully connected network has two layers

(Example for GPT-2 small)



GPT-2, BERT

GPT-2 has a parameter top-k, so that we sample works from top k (highest probability from softmax) words for each each output



GPT Training

- GPT-2 uses unsupervised learning approach to training the language model.
- There is no custom training for GPT-2, no separation of pre-training and fine-tuning like BERT.

A story generated by GPT-2

“The scientist named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

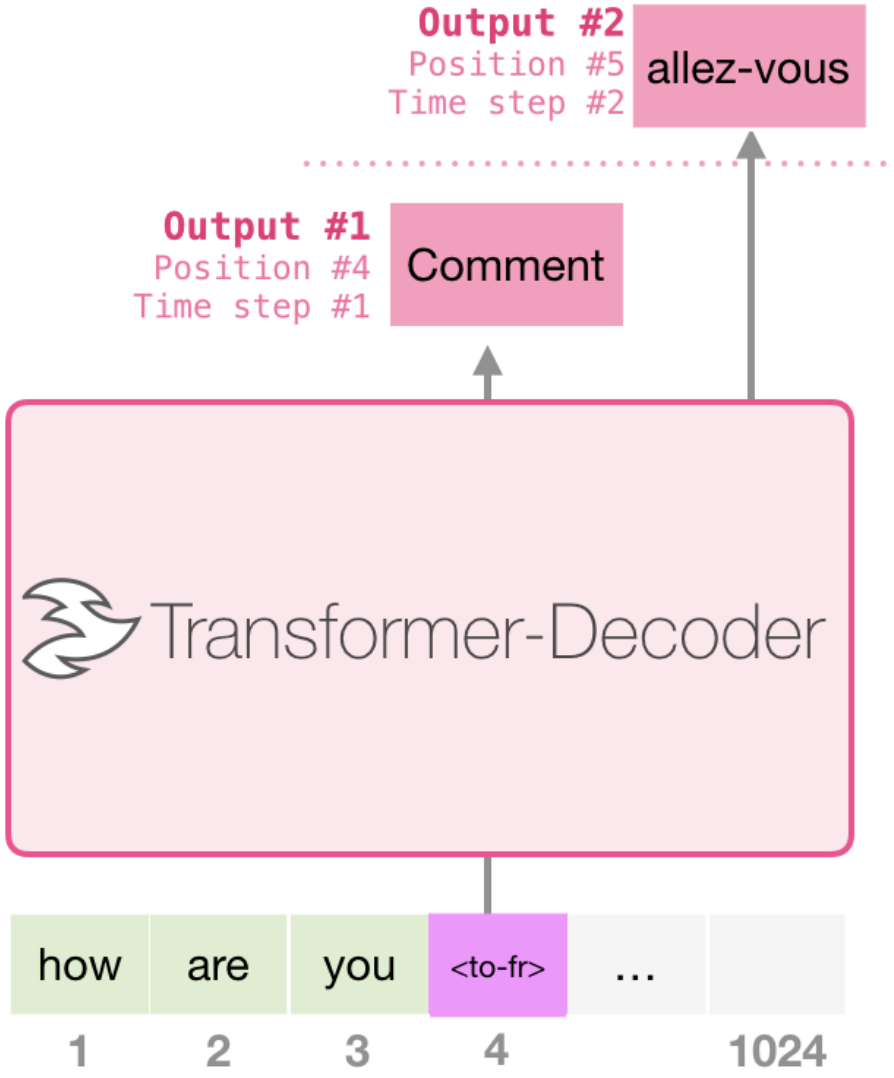
Pérez and the others then ventured further into the valley. `By the time we reached the top of one peak, the water looked blue, with some crystals on top,’ said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns."

GPT-2 Application: Translation

Training Dataset

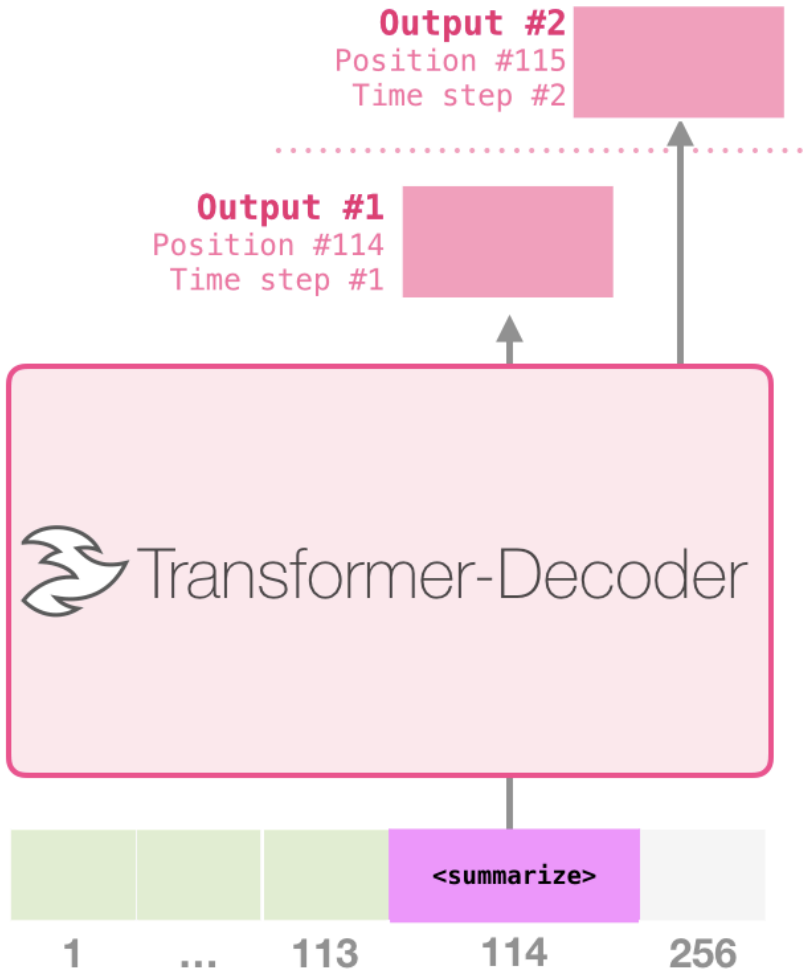
I	am	a	student	<to-fr>	je	suis	étudiant
let	them	eat	cake	<to-fr>	Qu'ils	mangent	de
good	morning	<to-fr>	Bonjour				



GPT-2 Application: Summarization

Training Dataset

Article #1 tokens		<summarize>	Article #1 Summary
Article #2 tokens	<summarize>	Article #2 Summary	padding
Article #3 tokens		<summarize>	Article #3 Summary



GPT-3

- GPT-2 is a giant transformer based on a language model with 1.5 billion parameters, and was trained for predicting the next word in 40GB of Internet text.
- GPT3: A state-of-the-art language model made up of 175 billion parameters.

- http://jalammar.github.io/illustrated-gpt2/