CS 639: Foundation Models
Architectures II

Fred Sala

University of Wisconsin-Madison

Feb. 19, 2026

Announcements

- Midterm: **March 11, 5:40 pm - 7:20 pm**
  - Location: **Ingraham Hall, Room B10**
- Homework 1: due Tues!
  - HW 2: coming out on Tues
- Class roadmap:

Thursday Feb. 19	Architectures: Others
Tuesday Feb. 24	Attention Variants
Thursday Feb. 26	Multimodal Architectures I
Tuesday March 3	Multimodal Architectures II

LLMs, FMs and Arch

Outline

- **Finish up last time: Encoder-only models**
  - Example: BERT, architecture, multitask training, fine-tuning
- **Decoder-only Models**
  - Example: GPT, architecture, basic functionality, properties of new models

Outline

- **Finish up last time: Encoder-only models**
  - Example: BERT, architecture, multitask training, fine-tuning
- **Decoder-only Models**
  - Example: GPT, architecture, basic functionality, properties of new models

Why Encoder-Decoder?

Wanted two things for translation:

- 1) **Outputs** in natural language
- 2) Tight alignment with **input**

What happens if we relax these?

1. Encoder-only models
2. Decoder-only models

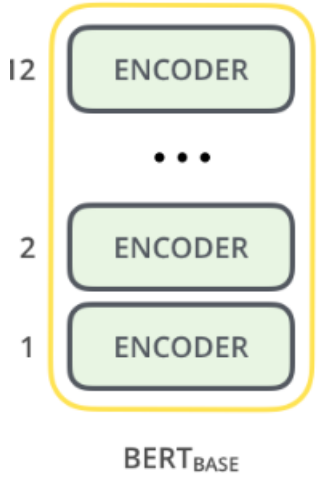
Encoder-Only Models: BERT

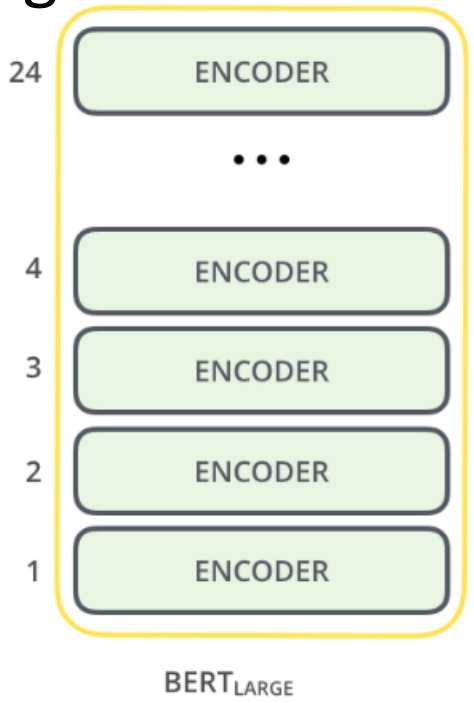
Let's get rid of the first part

- 1) **Outputs** in natural language
- 2) Tight alignment with **input**

- So **not** a generative model → get representations
  - Like we talked about in self-supervised learning

- Rip away decoders
  - Just stack encoders



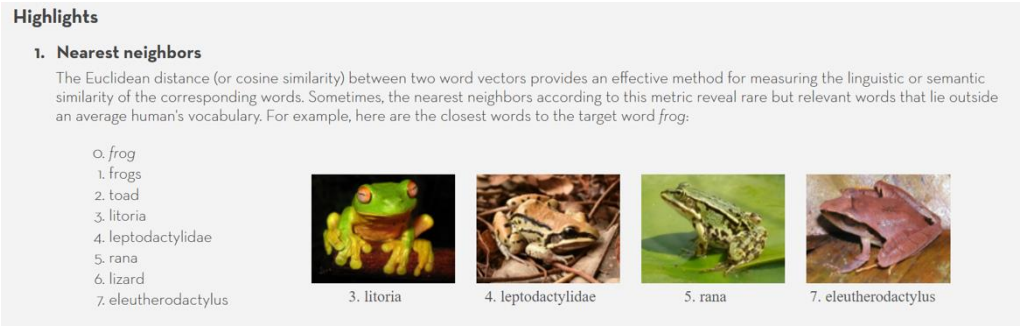


Interlude: Contextual Embeddings

Q: Why is it called "BERT"?

- **A**: In a sense, follows up ELMo

- Story:
  - **2013**: "Dense" word embeddings (**Word2Vec**, **Glove**)
  - Downside: fixed representations per word
    - "Bank": building or riverside?
  - Need: contextual representations
    - Using language model-like techniques
    - 2018: ELMo, BERT
    - ELMo: uses LSTMs, BERT uses transformers



https://nlp.stanford.edu/projects/glove/

Interlude: Contextual Embeddings

Q: Why is it called “BERT”?

- **A**: In a sense, follows up ELMo

BERT acronym:

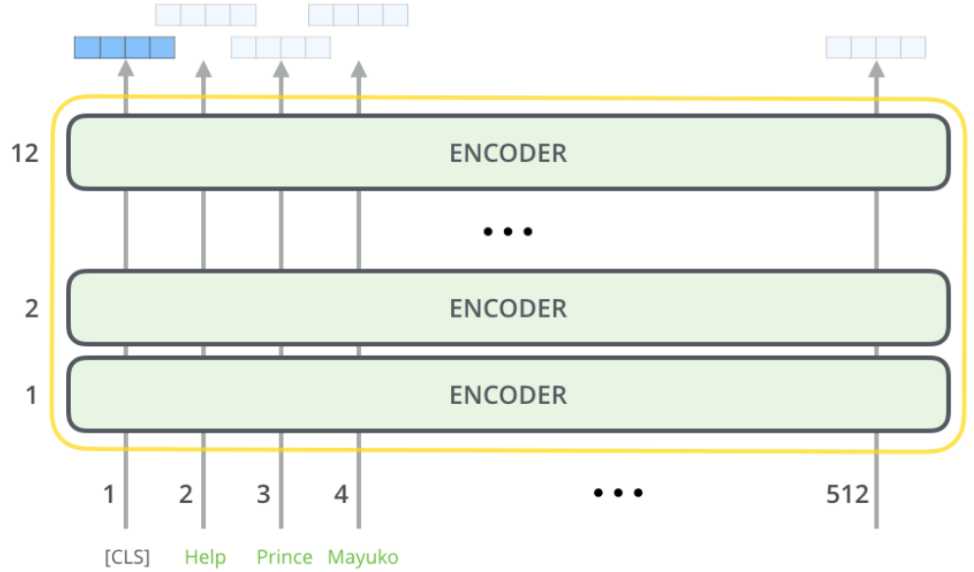
- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers.
- ERT should make sense,

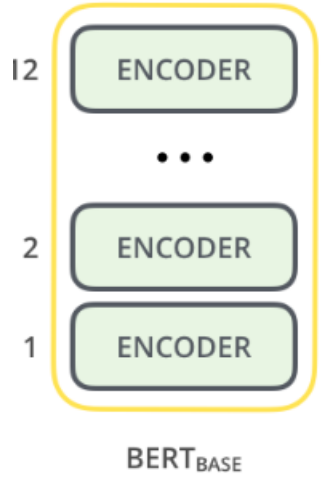
- Bidirectional: no causal masks, look at both sides of a word!
- Captured in self-attention block

BERT: Forward Pass

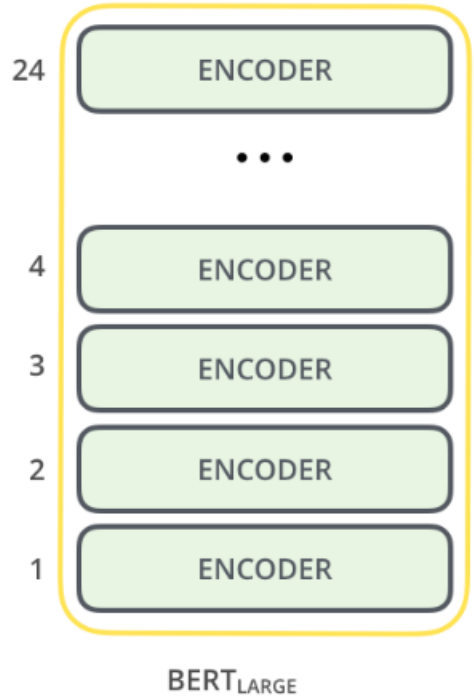
BERT architecture

- Rip away decoders
  - Just stack encoders





BERT$_{BASE}$



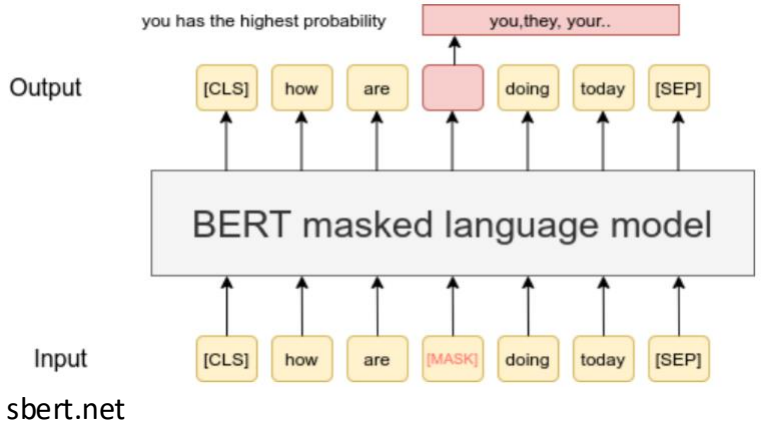
BERT$_{LARGE}$

BERT: Training

Training is more interesting!

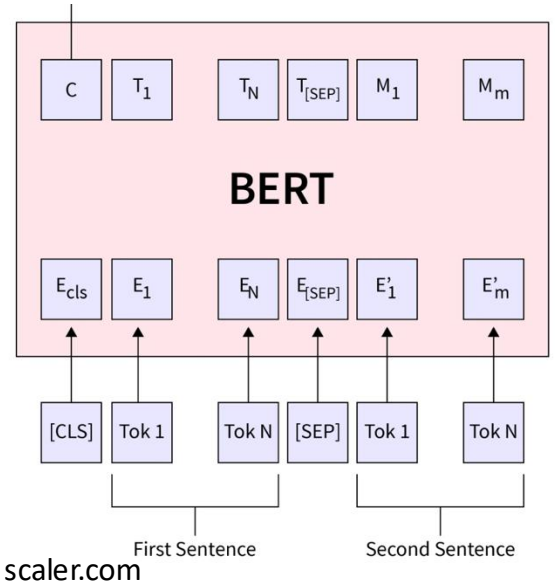
- Pretraining. Then fine-tuning on task of interest

- Back to **self-supervised learning!**
- Two tasks for **pretraining**.



sbert.net

1. Masked Language Modeling



scaler.com

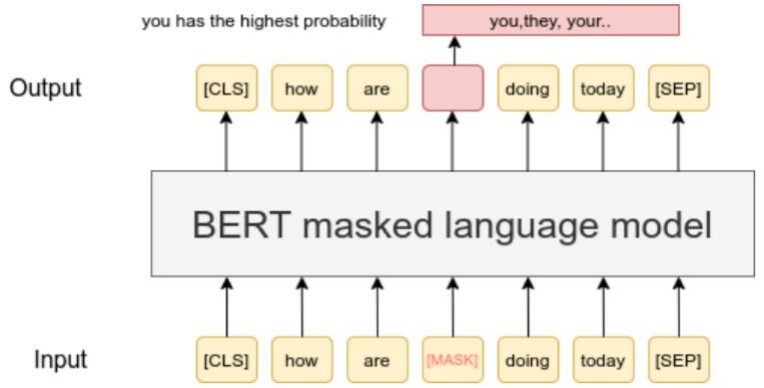
2. Next Sentence Prediction

BERT: Training Task 1

Masked Language Modeling Task

- Use [MASK] token for word to be predicted

- Which words to mask?
  - Original paper: 15% of words at random
  - But… of these
    - 10% of the time, no [MASK], flip word randomly
    - 10% of the time leave word unchanged



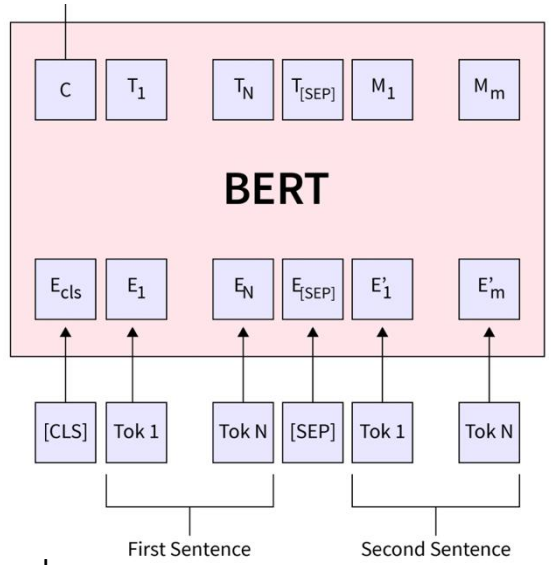
sbert.net

BERT: Training Task 2

Next sentence prediction

- Several ways of doing this, but basically binary classification
- Another self-supervision task
  - Later turned out that this is not strictly necessary!

- Place a sentence, then a candidate 2nd sentence
  - Use [SEP] token to distinguish 2 sentences
  - Predict if it's indeed the next sentence
    - Setup: 50% of sentences are indeed next,
    - Other 50% are random sentences we picked



scaler.com

BERT: Training Details

Where do we get our training data?

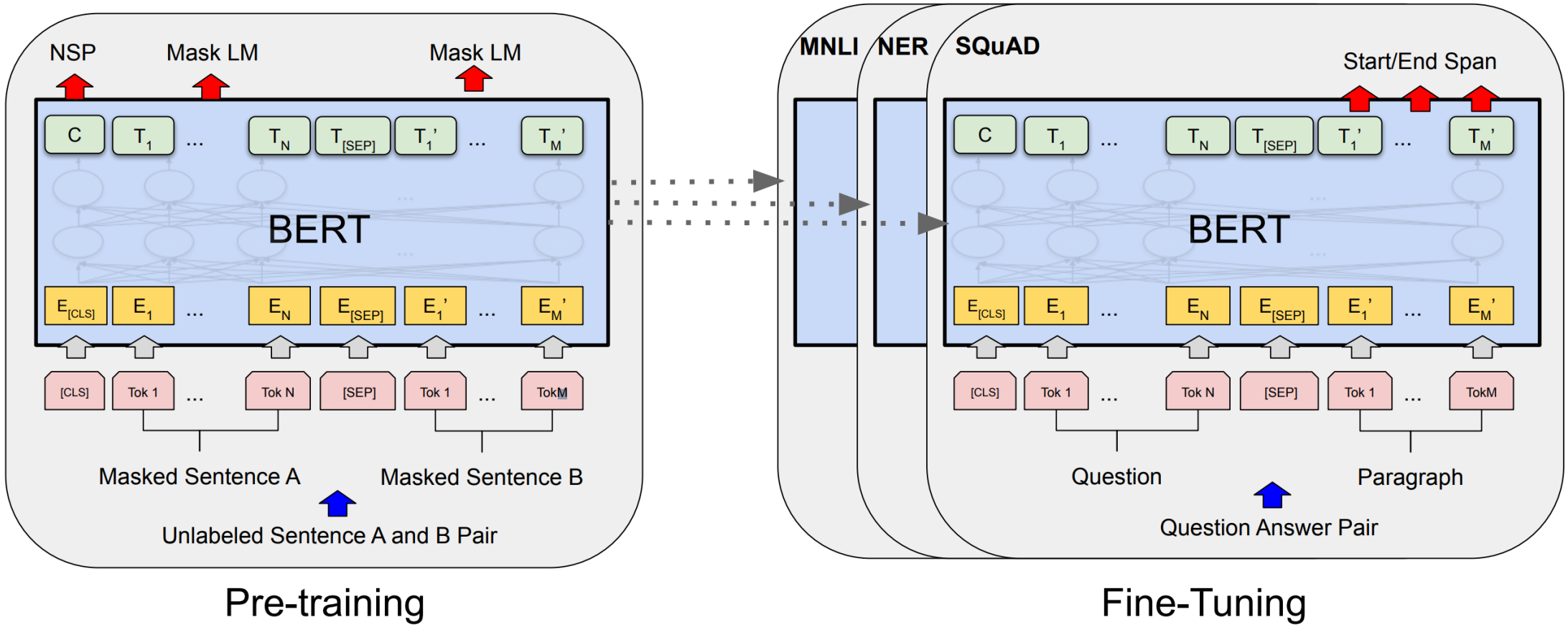
- We need a bunch of raw text, since our two objectives are **masked token prediction** + **next sentence prediction**

- BooksCorpus (800M words)
- English Wikipedia (2,500M words)
  - We'll get a lot larger than these as we go…
  - Some filtering: "Wikipedia we extract only the text passages and ignore lists, tables, and headers"… (Devlin et al '18).

BERT: Fine-Tuning

Once we're done pretraining, we can fine-tune

- This is now supervised learning again!



Devlin et al '18

Break & Questions

Outline

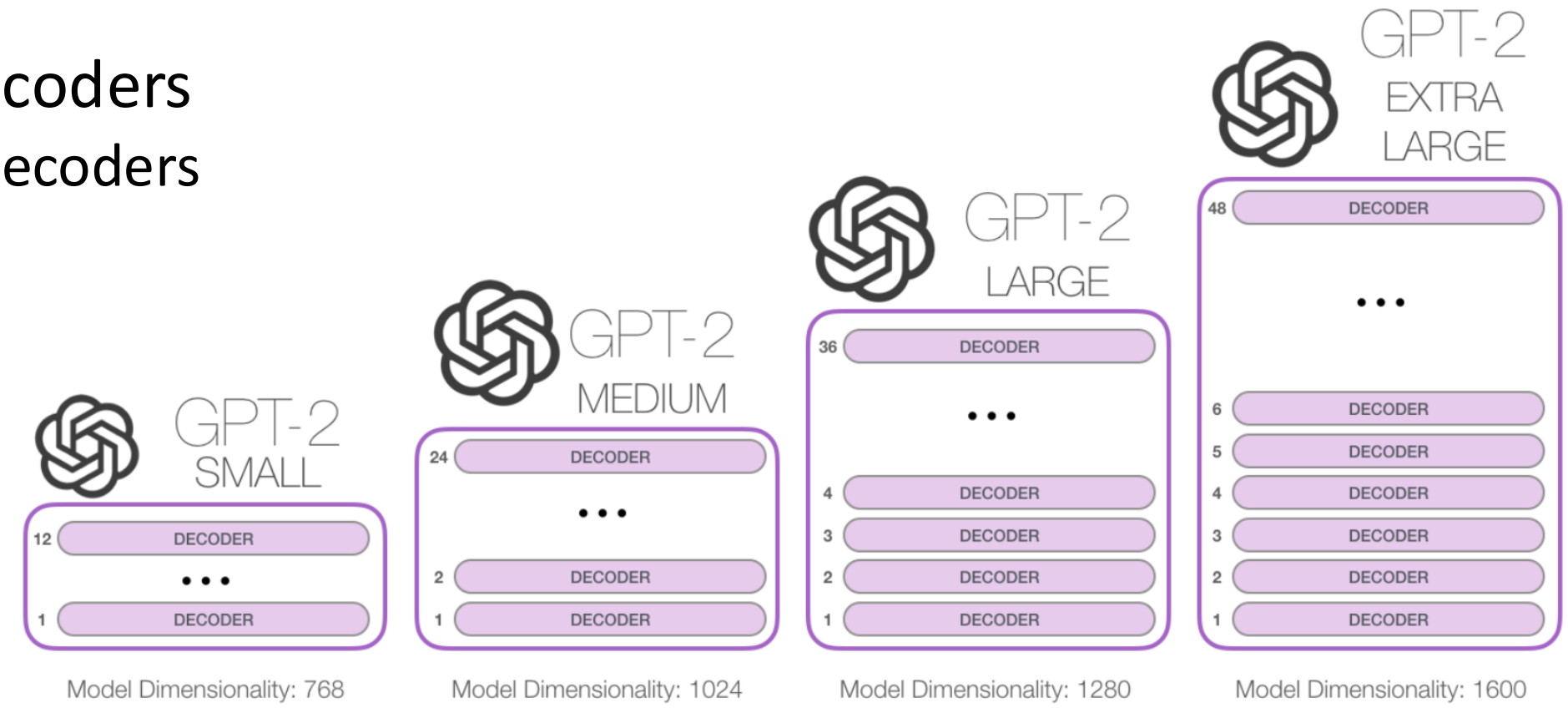
- **Finish up last time: Encoder-only models**
  - Example: BERT, architecture, multitask training, fine-tuning
- **Decoder-only Models**
  - Example: GPT, architecture, basic functionality, properties of new models

Decoder-Only Models: GPT

Let's get rid of the first part

- 1) **Outputs** in natural language
- 2) Tight alignment with **input**

- Rip away encoders
  - Just stack decoders



Decoder-Only Models: GPT

Let's get rid of the first part

- 1) **Outputs** in natural language
- 2) Tight alignment with **input**

- Let's handle this acronym as well: **GPT**
  - **G**enerative (i.e., a language model that generates rich content, as opposed to representations or predictions)
    - Unlike BERT!
  - **P**retrained
    - Like BERT!
  - **T**ransformers (also like BERT)

Quick Interlude: Language Models

What's a "language model"?

- Basic idea: use probabilistic models to assign a probability to a sentence W

$$P(W) = P(w_1, w_2, \ldots, w_n) \text{ or } P(w_{\text{next}} | w_1, w_2 \ldots)$$

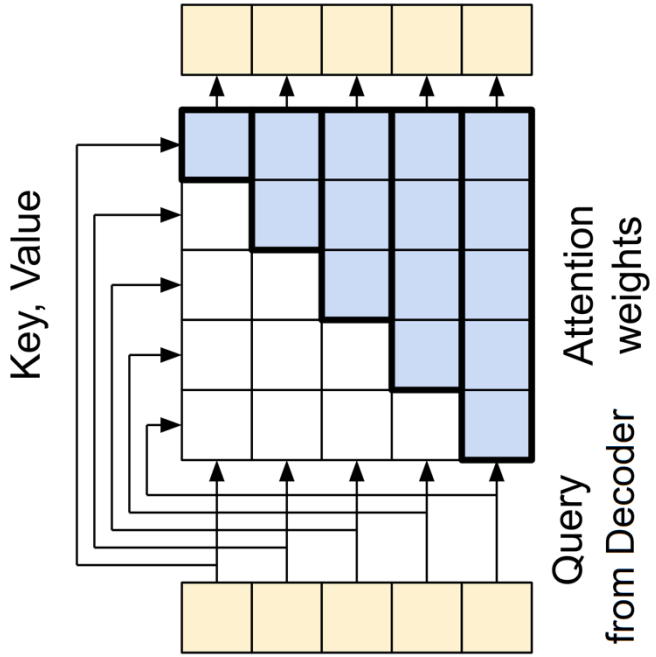
- We mostly care about the latter: gives us a **distribution to sample next word** (or next token)
- GPT models will also do this!
  - But idea's much older: Shannon's example

Zero-order approximation	XFOML RXKHRJFFJUJ ALPWXFWJXYJ FFJEYVJCQSGHYD QPAAMKBZAACIBZLKJQD
First-order approximation	OCRO HLO RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL
Second-order approximation	ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE
Third-order approximation	IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE
First-order word approximation	REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE

Decoder-Only Models: GPT

Rip away encoders

- Just stack decoders
- Use causal masking! NB: not a *mask token* like in BERT

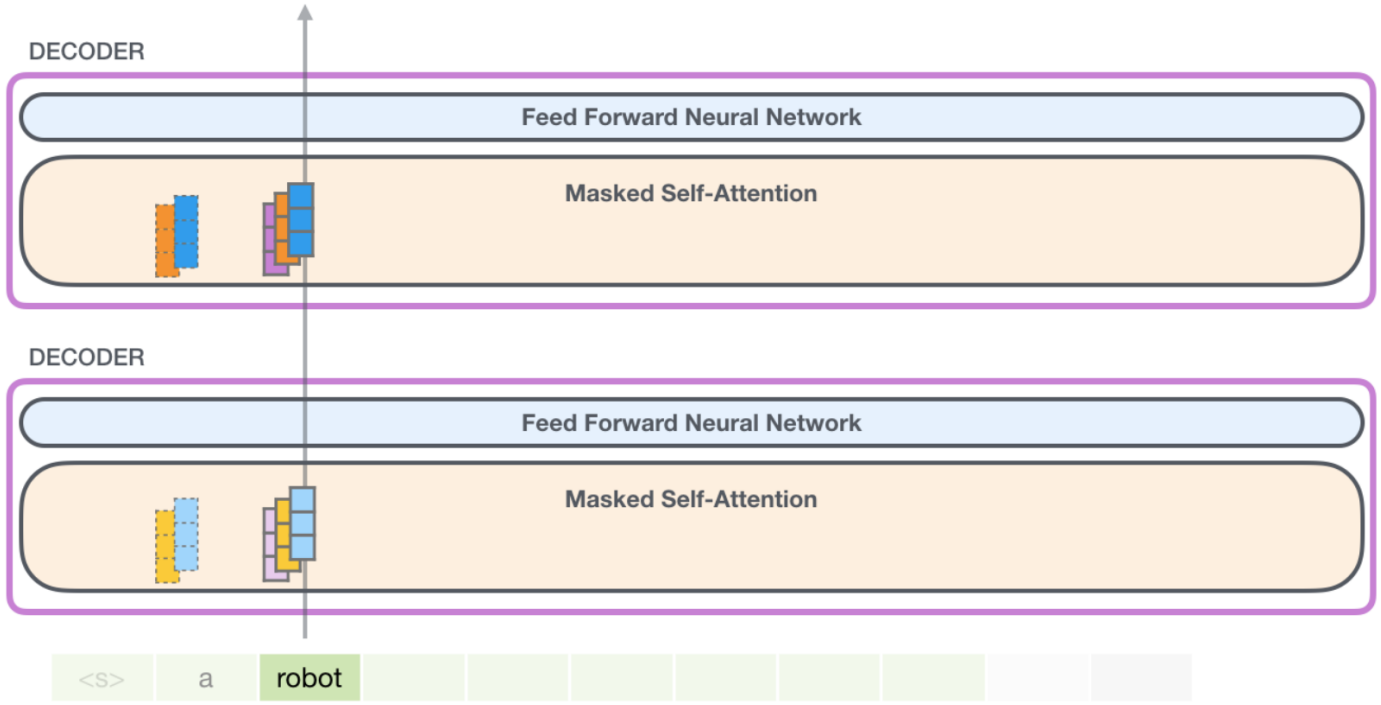


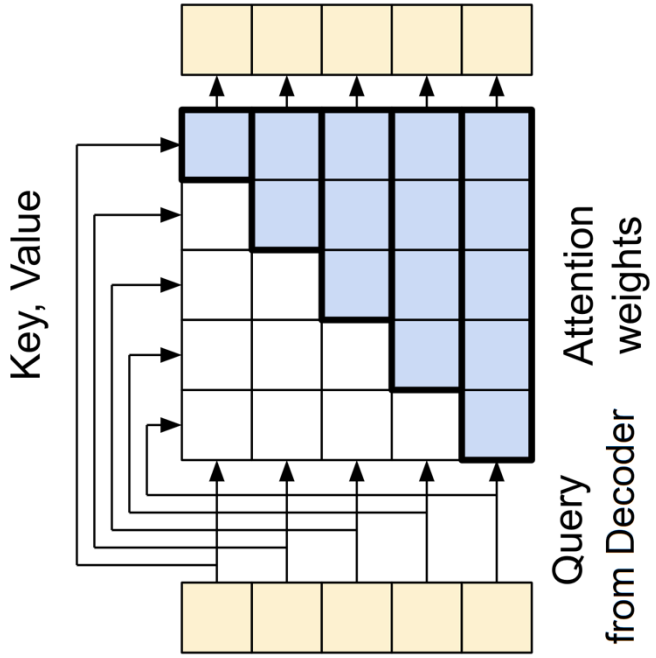
PyLessons

Decoder-Only Models: GPT

Rip away encoders

- Just stack decoders
- Decoders: get rid of **cross-attention** aspects (masked self-attention only)



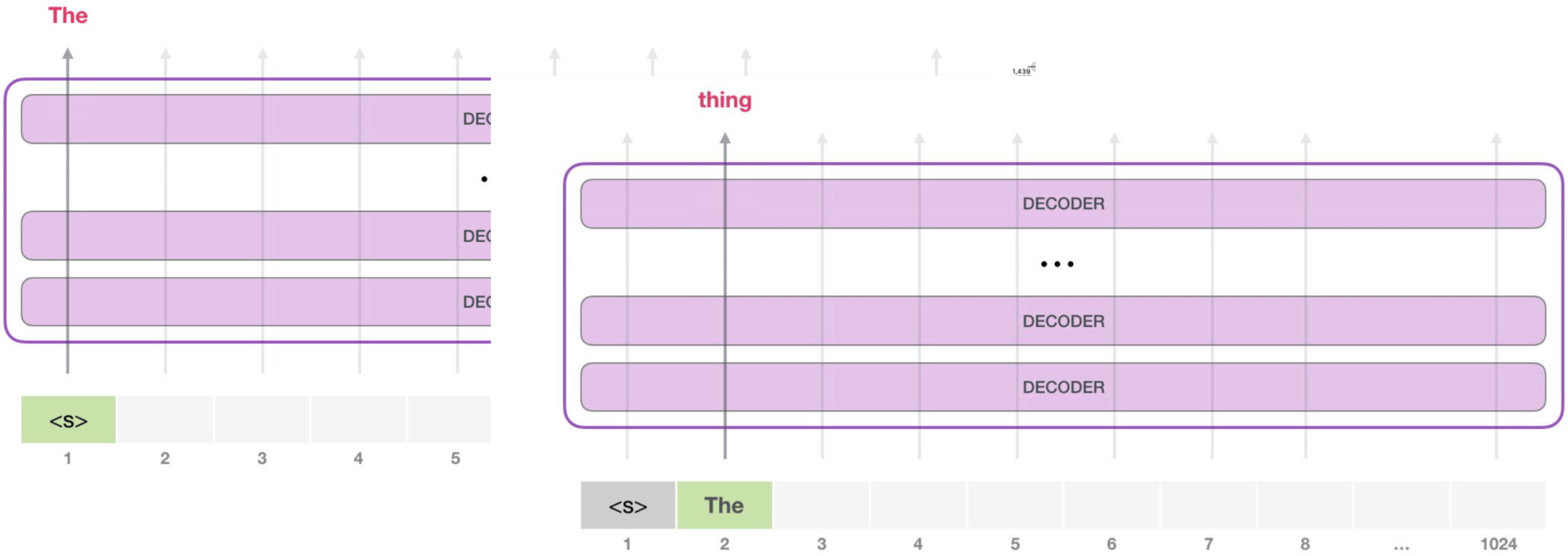


PyLessons

Decoder-Only Models: GPT

Autoregressive next token prediction mechanism:

- Plug in your current token, get next token
- Once you decode next token, plug it back in



From GPT2 to GPT3

Mainly make things larger!

- Why? **Scaling** produces emergent behaviors… more soon!
- 96 decoder blocks (getting very tall)
- Context size: **2048**
- 175 billion parameters in total (800GB!)

Training dat

GPT-3 training data[1]:9

Dataset	# tokens	Proportion within training
Common Crawl	410 billion	60%
WebText2	19 billion	22%
Books1	12 billion	8%
Books2	55 billion	8%
Wikipedia	3 billion	3%

https://en.wikipedia.org/wiki/GPT-3

Brown et al ‘20

Open Source: Llama 3.1

Mainly make things larger! Note: multiple model sizes:

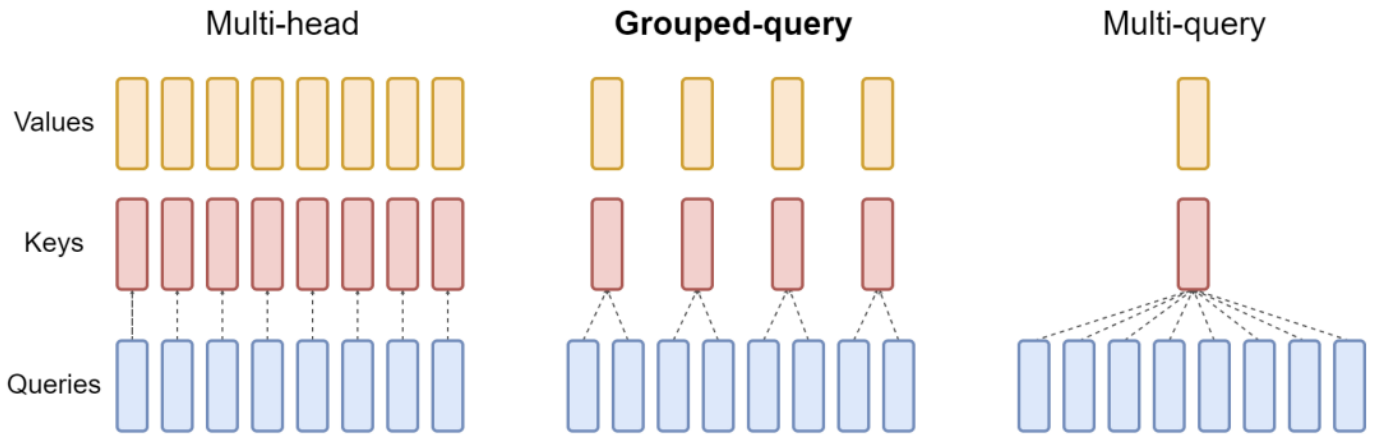
	8B	70B	405B
Layers	32	80	126
Model Dimension	4,096	8192	16,384
FFN Dimension	14,336	28,672	53,248
Attention Heads	32	64	128
Key/Value Heads	8	8	8
Peak Learning Rate	3×10^{-4}	1.5×10^{-4}	8×10^{-5}
Activation Function	SwiGLU		
Vocabulary Size	128,000		
Positional Embeddings	RoPE ($\theta = 500,000$)		

Dubey et al ‘24

Open Source: Llama 3.1

Some improvements for Llama 3.1:

- "We use an attention mask that **prevents self-attention between different documents** within the same sequence"
- "**grouped query attention** (GQA; Ainslie et al. (2023)) with 8 key-value heads to improve inference speed…"



Ainslie et al '23

ɔ et al '21

Open Source: Llama 3.1

Some improvements for Llama 3.1:

- "We use an attention mask that **prevents self-attention between different documents** within the same sequence"
- "**grouped query attention** (GQA; Ainslie et al. (2023)) with 8 key-value heads to improve inference speed…"
- "We use a **vocabulary with 128K tokens.** Our token vocabulary combines 100K tokens from the tiktoken3 tokenizer with 28K additional tokens to better support non-English languages"

Zhao et al '21

Thank You!