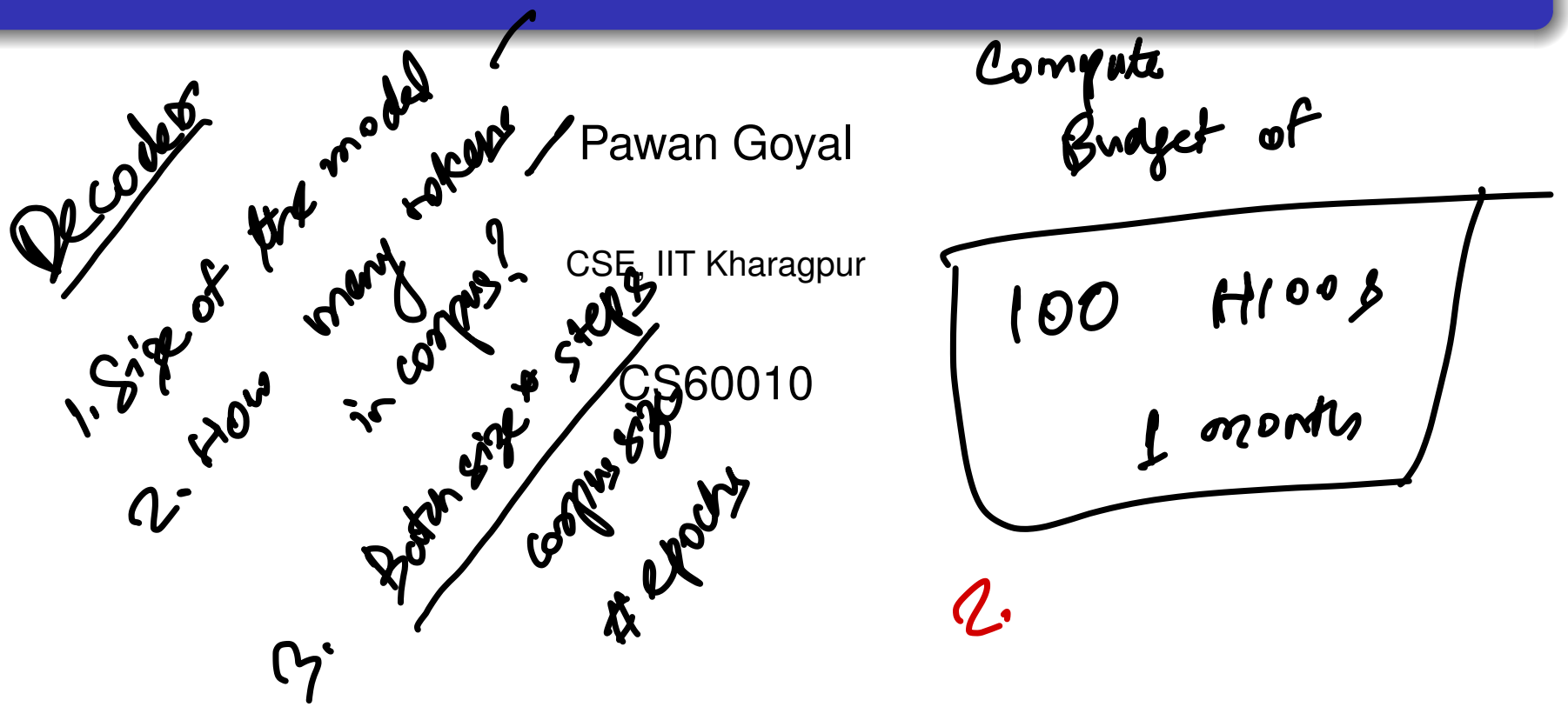
Scaling Laws; Large Language Models

Pawan Goyal

CSE, IIT Kharagpur

CS60010

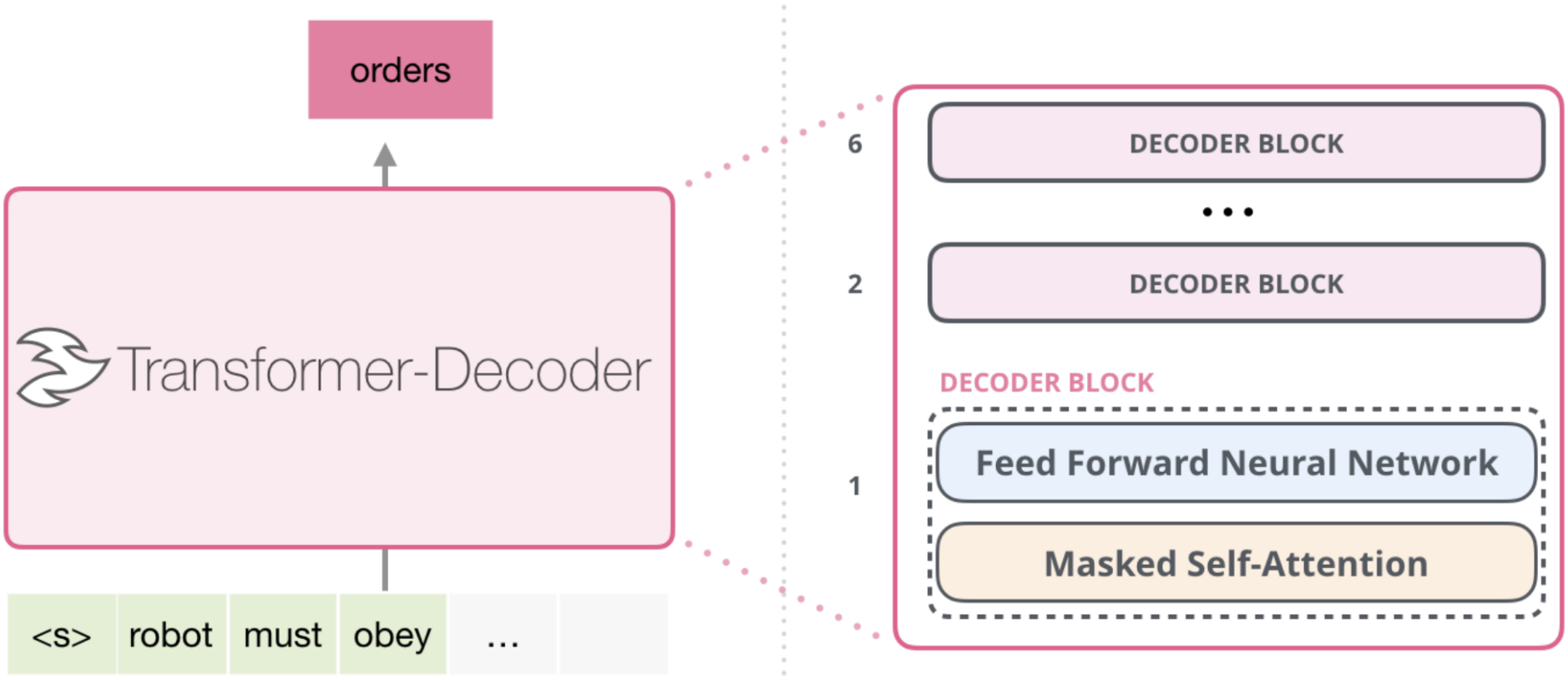


Practice Problem

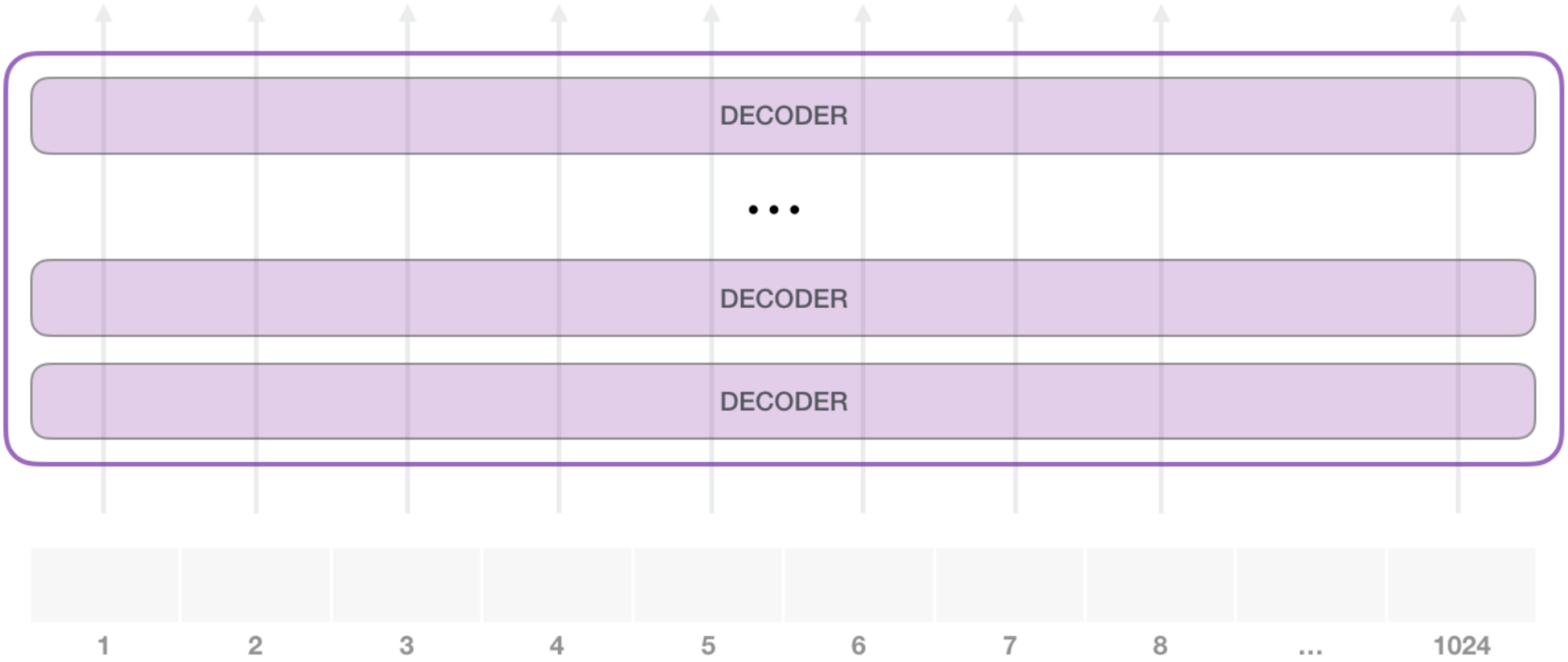
Suppose you are using BERT for reading comprehension based question answering. Suppose your input paragraph is, "You can ignore the **bias terms**", and assume that each individual word is part of the vocabulary and the bold span is the ground truth answer. For the sake of simplicity, assume that you are working with 2-dimensional hidden states. Let your start and end vectors be [1,-1] and [-1,1], respectively, and the final embedding for the words in the input sentence be [-1,-1], [1,1], [1,2], [2,1], [1,-2] and [2,-1], respectively.

- If this was a sentence in the training, what would be the corresponding loss for this sentence?
- If this was a sentence during inference, how many spans would you have to consider?
- Assume that it is given that the span can have at most 2 words. What will be the output of the model at inference time?

GPT-2: Transformer-Decoder

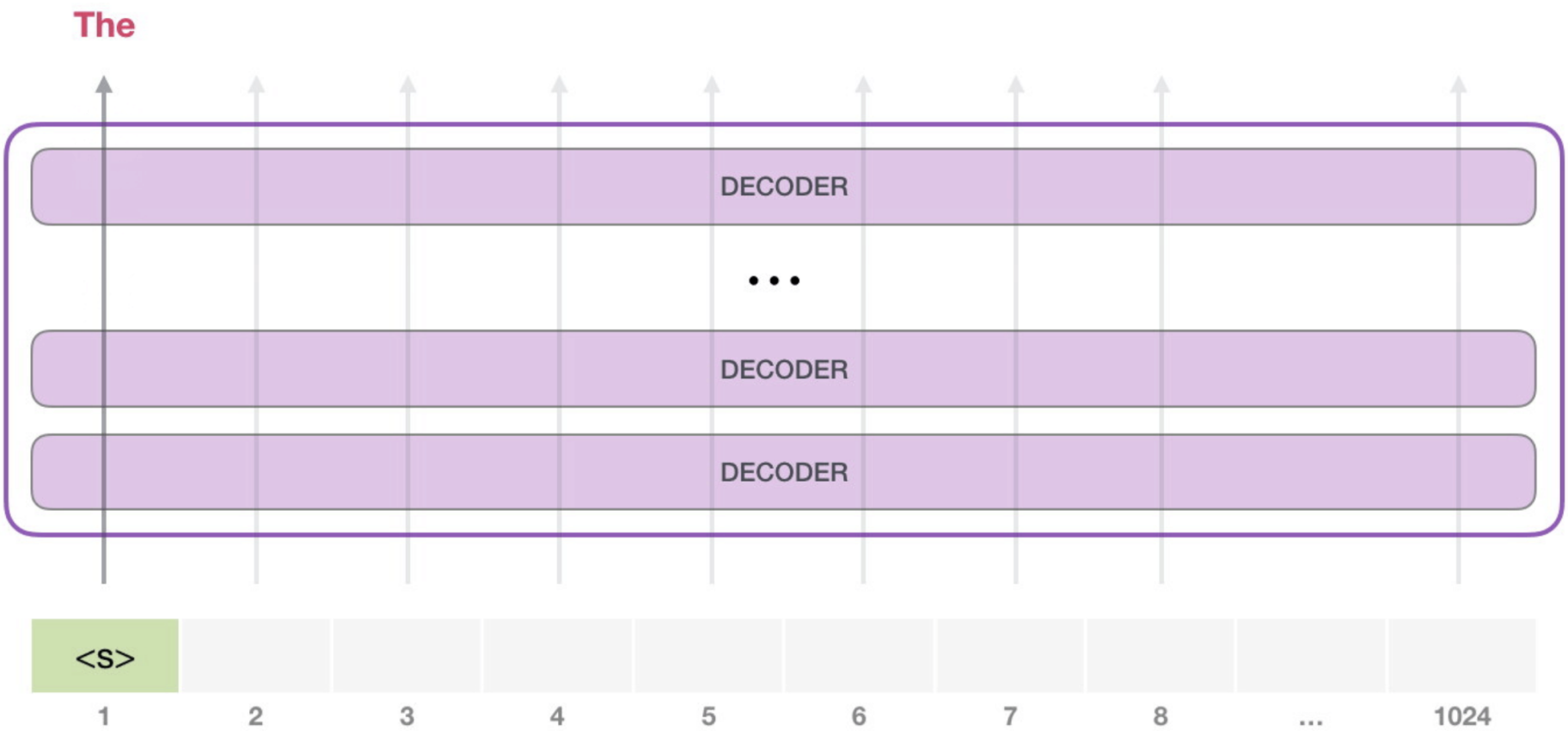


GPT-2: Quick Summary

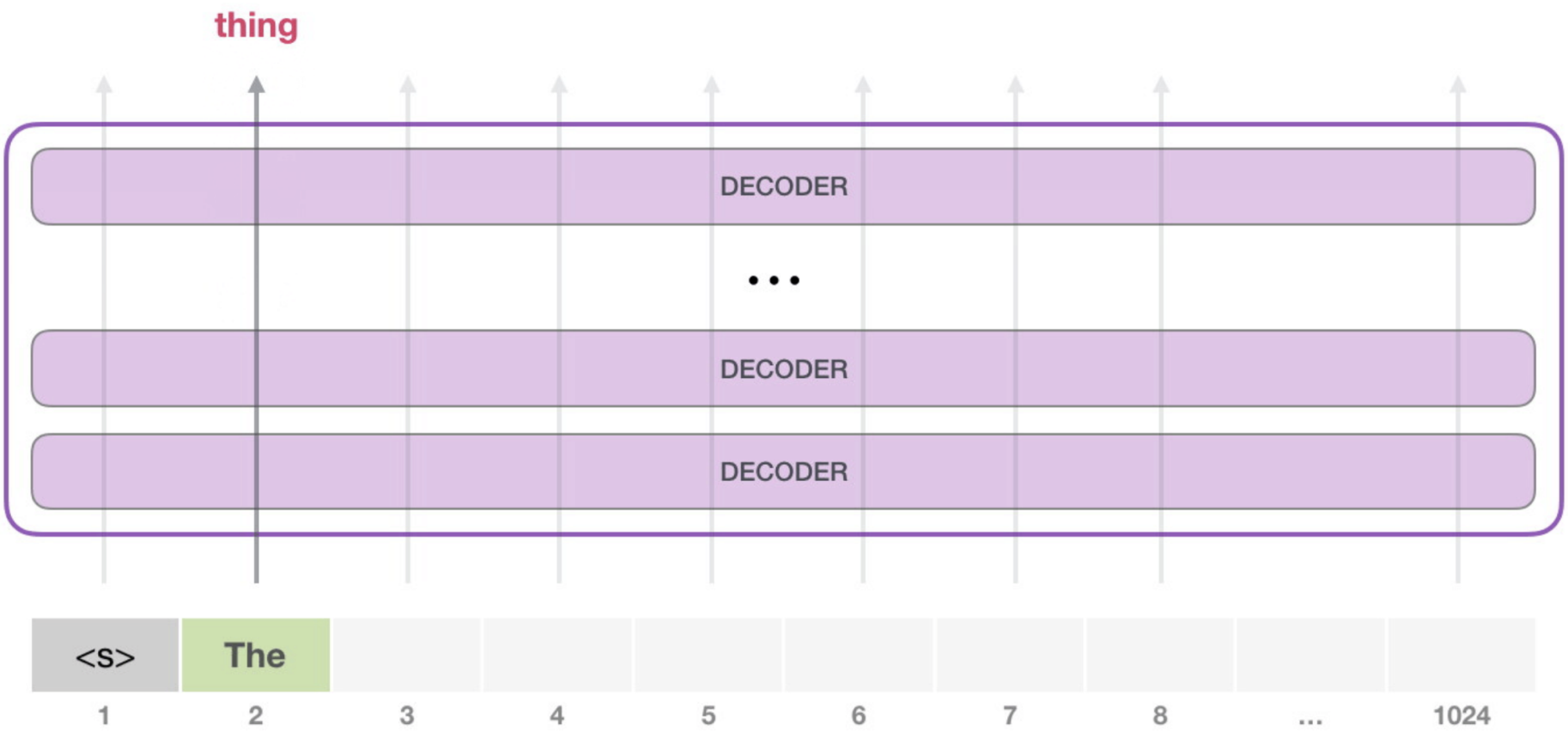


The GPT-2 can process 1024 tokens. Each token flows through all the decoder blocks along its own path.

GPT-2: Quick Summary



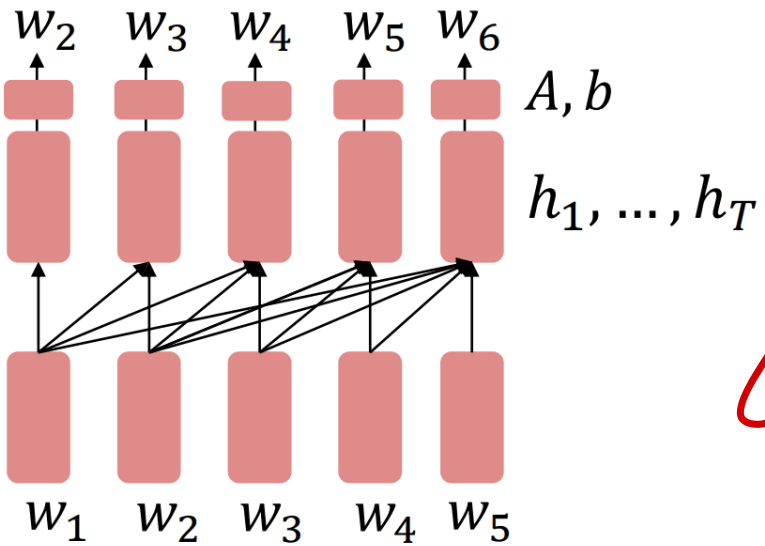
GPT-2: Quick Summary



How to use Pretrained Decoders?

This is helpful in tasks where the output is a sequence

- Dialogue (context=dialogue history)
- Summarization (context=document)



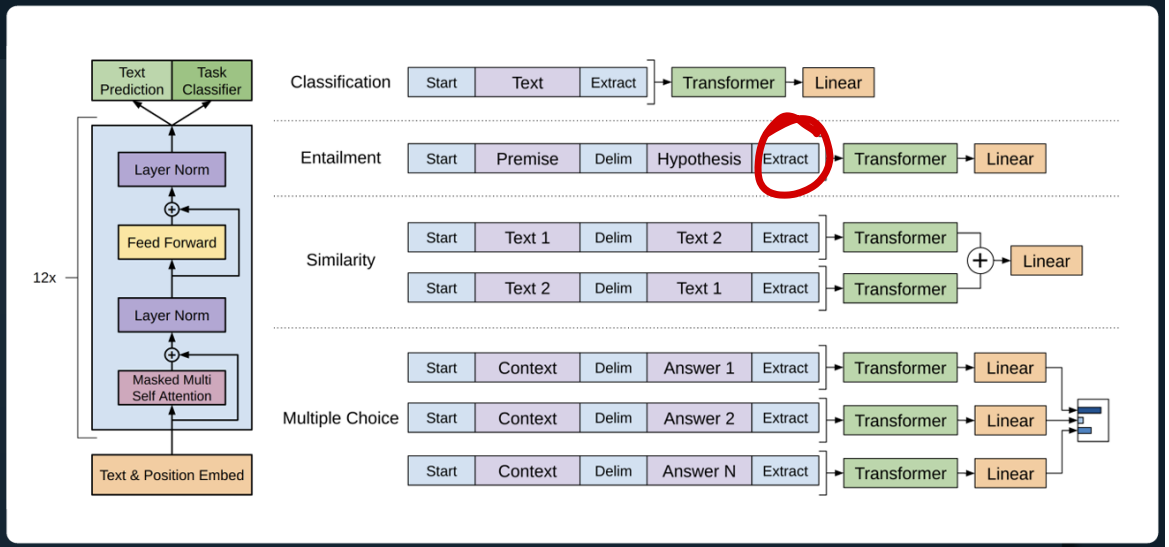
GPT for Language Understanding

Step 1:

Model "pretraining" on large unsupervised dataset

Step 2:

model "finetuning" on small supervised dataset

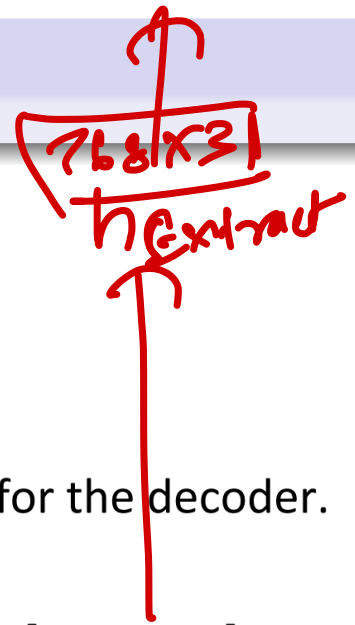


Improving Language Understanding by Generative Pre-Training, Radford et al. 2018 **(GPT-1)**

GPT: How to format inputs?

Natural Language Inference

Premise: *The man is in the doorway*
Hypothesis: *The person is near the door*
} **entailment**



Radford et al., 2018 evaluate on natural language inference.
Here's roughly how the input was formatted, as a sequence of tokens for the decoder.

[START] *The man is in the doorway* [DELIM] *The person is near the door* [EXTRACT]

The linear classifier is applied to the representation of the [EXTRACT] token.

GPT-2: Beyond Language Modeling

Basic Idea

The decoder can work with a prompt!

Any NLP task can be expressed in a probabilistic framework as estimating a conditional distribution $p(output|input)$.

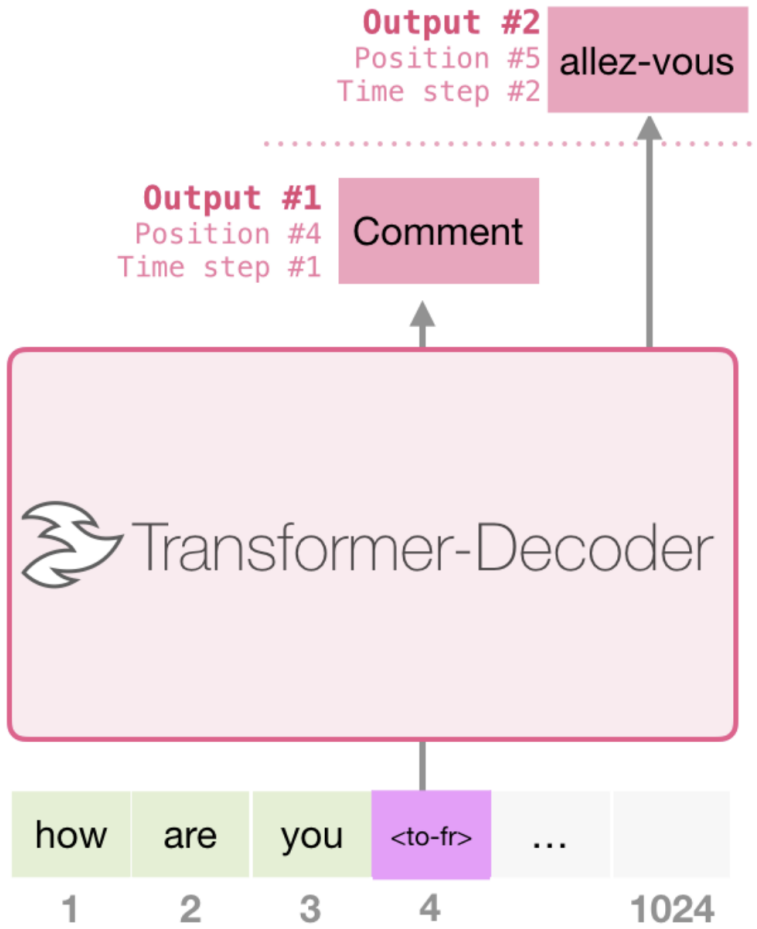
For instance, reading comprehension training example

(answer the question, document, question, answer)

GPT-2: Machine Translation

Training Dataset

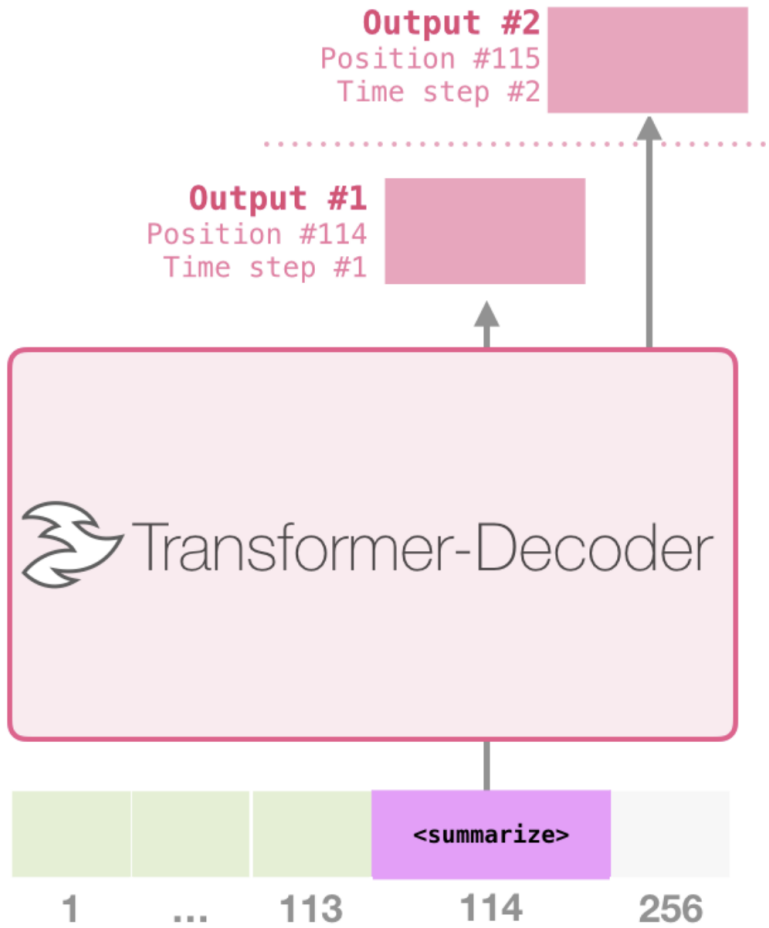
I	am	a	student	<to-fr>	je	suis	étudiant
let	them	eat	cake	<to-fr>	Qu'ils	mangent	de
good	morning	<to-fr>	Bonjour				



GPT-2: 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	



Base models can be prompted into completing tasks

Context (passage and previous question/answer pairs)

Tom goes everywhere with Catherine Green, a 54-year-old secretary. He moves around her office at work and goes shopping with her. "Most people don't seem to mind Tom," says Catherine, who thinks he is wonderful. "He's my fourth child," she says. She may think of him and treat him that way as her son. He moves around buying his food, paying his health bills and his taxes, but in fact Tom is a dog.

Catherine and Tom live in Sweden, a country where everyone is expected to lead an orderly life according to rules laid down by the government, which also provides a high level of care for its people. This level of care costs money.

People in Sweden pay taxes on everything, so aren't surprised to find that owning a dog means more taxes. Some people are paying as much as 500 Swedish kronor in taxes a year for the right to keep their dog, which is spent by the government on dog hospitals and sometimes medical treatment for a dog that falls ill. However, most such treatment is expensive, so owners often decide to offer health and even life _ for their dog.

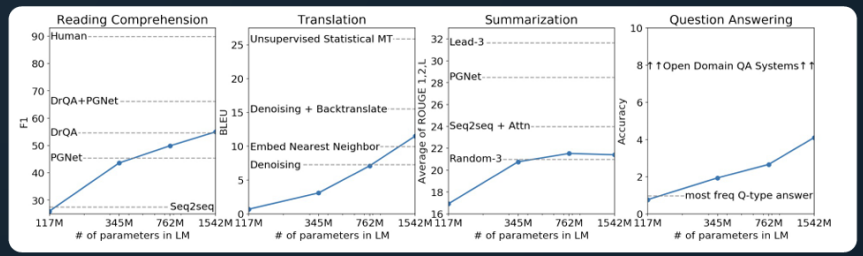
In Sweden dog owners must pay for any damage their dog does. A Swedish Kennel Club official explains what this means: if your dog runs out on the road and gets hit by a passing car, you, as the owner, have to pay for any damage done to the car, even if your dog has been killed in the accident.

Q: How old is Catherine?
A: 54

Q: where does she live?
A:

GPT-2 is "tricked" into performing a task by completing the document

GPT-2 kicked off the era of prompting over finetuning



Language Models are Unsupervised Multitask Learners, Radford et al. 2019 **(GPT-2)**

GPT-3: In-context learning

Typical interaction with pretrained models so far

Fine-tune them on a task we care about, and take their predictions

In-context learning: No-gradient steps!

- Very large language models seem to perform some kind of learning without gradient steps simply from examples you provide within their contexts.
- GPT-3 is the canonical example of this. The largest T5 model had 11 billion parameters. GPT-3 has 175 billion parameters.

GPT-3: In-context learning

The in-context examples seem to specify the task to be performed, and the conditional distribution mocks performing the task to a certain extent.

Input (prefix within a single Transformer decoder context):

“ thanks -> merci

hello -> bonjour

mint -> menthe

otter -> ”

Output (conditional generations):

loutre...”

Emergent Few-shot learning

- Specify a task by simply prepending examples of the task before your example
- Also called in-context learning, to stress that no gradient updates are performed when learning a new task

'Prompting' vs fine-tuning

Zero/few-shot prompting

```
1  Translate English to French:
2  sea otter => loutre de mer
3  peppermint => menthe poivrée
4  plush girafe => girafe peluche
5  cheese =>
```

```
1  sea otter => loutre de mer
```

gradient update

```
1  peppermint => menthe poivrée
```

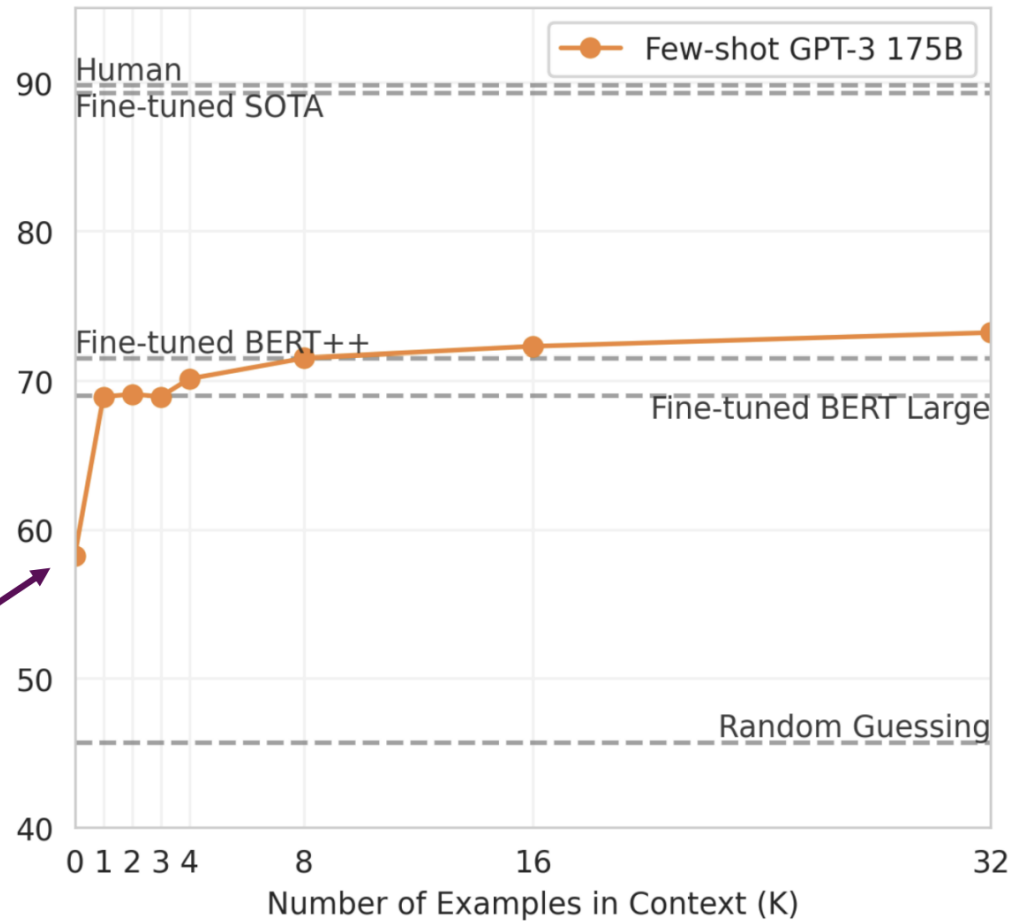
gradient update

```
1  cheese =>
```

Emergent Few-shot learning

Zero-shot

```
1  Translate English to French:
2  cheese =>
```



Why Scale? Scaling Laws

Performance of LLMs

Mainly determined by three factors

- model size (number of parameters N excluding embeddings)
- dataset size (amount of pretraining data D)
- amount of compute C (PF-days)

A petaflop/s-day (PF-day) consists of performing 10^{15} neural net operations per second for one day (roughly 10^{20} operations per day)

Scaling Laws

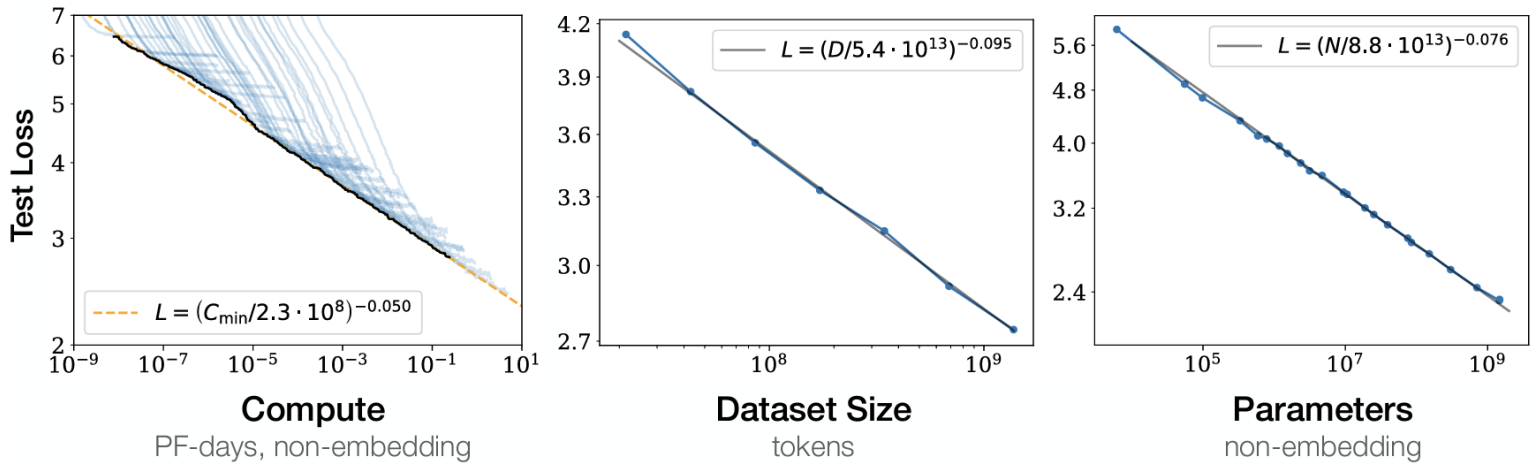


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

$$L(N) = (\frac{N_c}{N})^{\alpha_N} \quad \alpha_N \approx 0.076, N_c \approx 8.8 \times 10^{13}$$

$$L(D) = (\frac{D_c}{D})^{\alpha_D} \quad \alpha_D \approx 0.095, D_c \approx 5.4 \times 10^{13} \text{ tokens}$$

$$L(C) = (\frac{C_c}{C})^{\alpha_C} \quad \alpha_C \approx 0.050, D_c \approx 3.1 \times 10^8 \text{ PF-days}$$

Nice findings from the Scaling laws

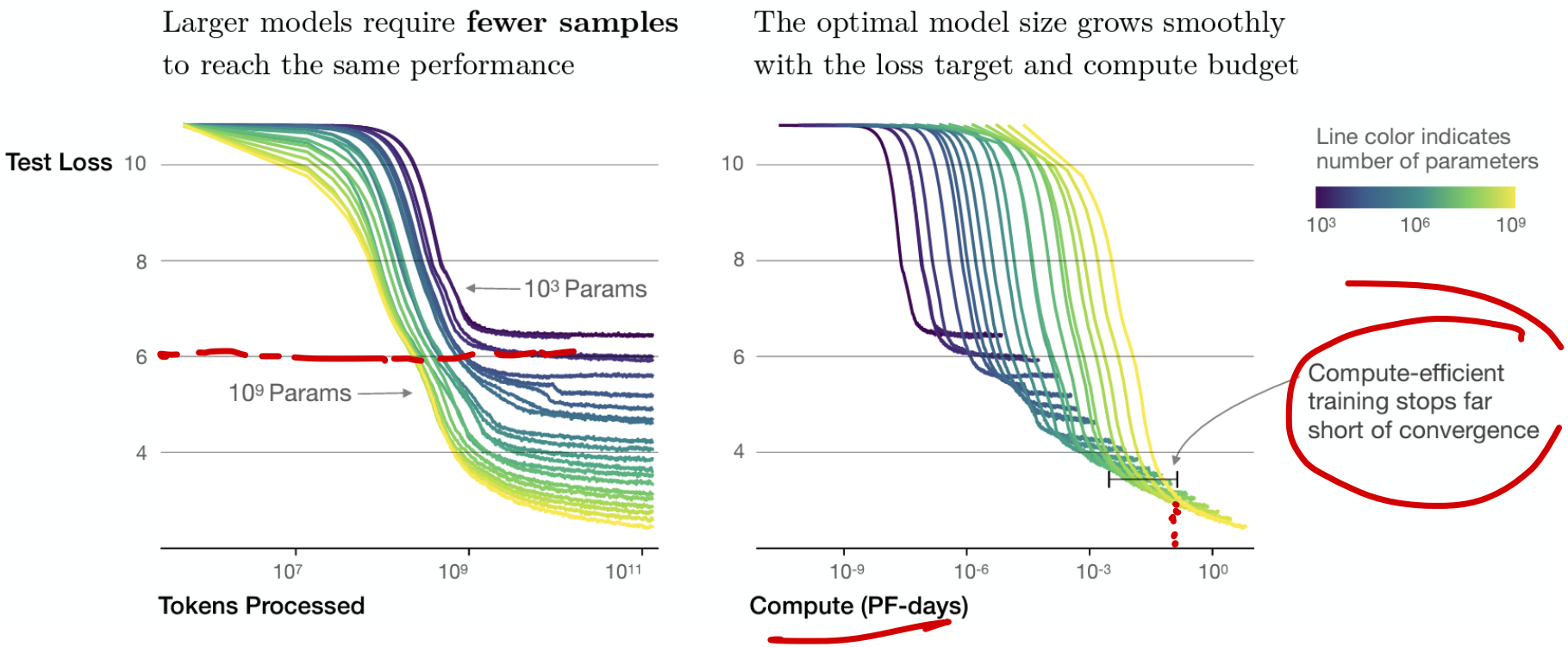
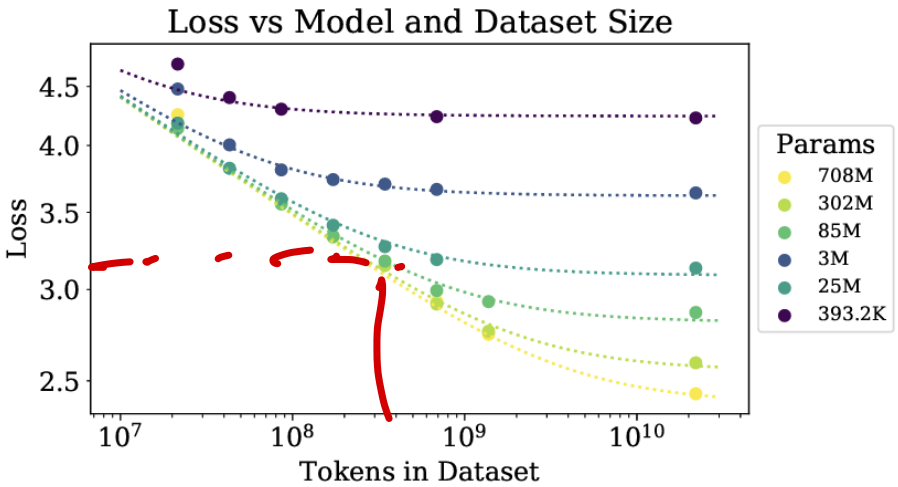
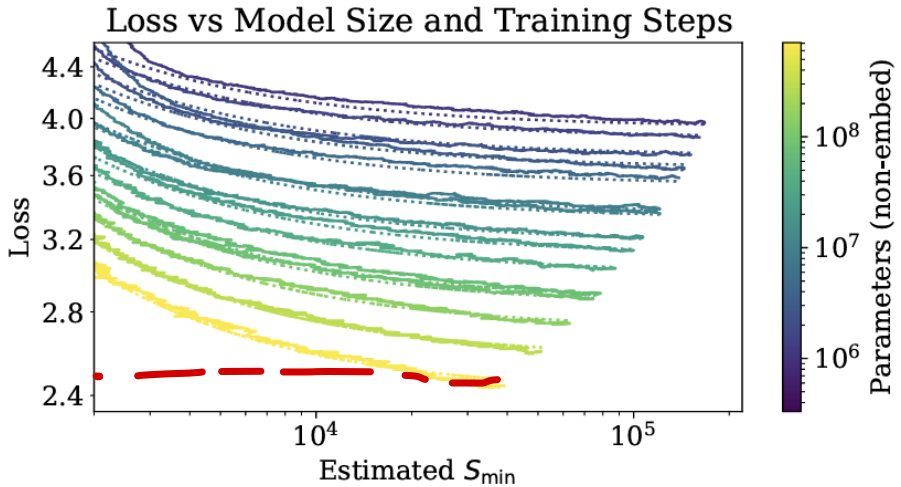


Figure 2 We show a series of language model training runs, with models ranging in size from 10^3 to 10^9 parameters (excluding embeddings).

Nice findings from the Scaling laws





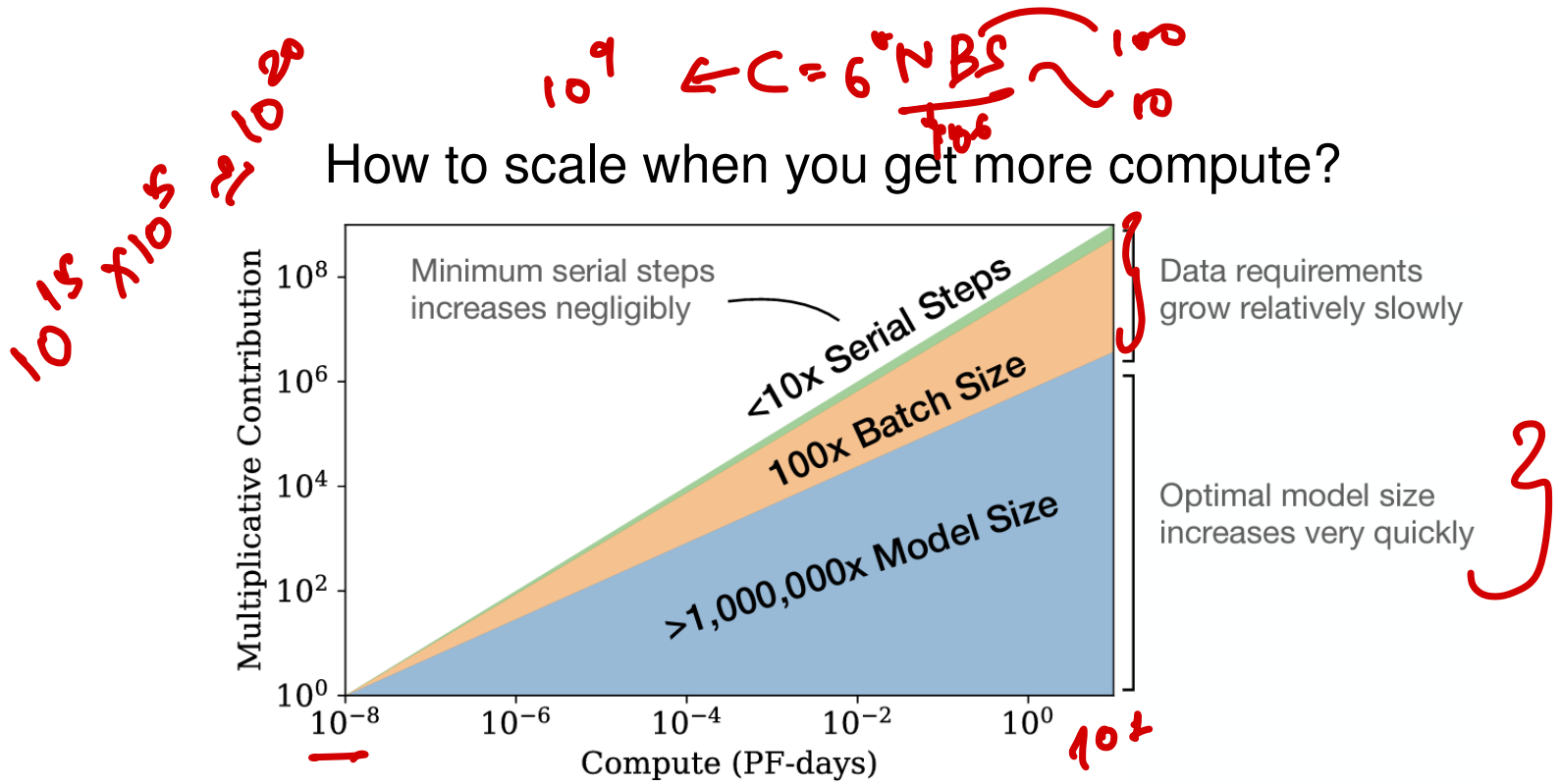


Figure 3 As more compute becomes available, we can choose how much to allocate towards training larger models, using larger batches, and training for more steps. We illustrate this for a billion-fold increase in compute. For optimally compute-efficient training, most of the increase should go towards increased model size. A relatively small increase in data is needed to avoid reuse. Of the increase in data, most can be used to increase parallelism through larger batch sizes, with only a very small increase in serial training time required.

Operation	Parameters	FLOPs per Token
Embed	$(n_{\text{vocab}} + n_{\text{ctx}})\, d_{\text{model}}$	$4d_{\text{model}}$
Attention: QKV	$n_{\text{layer}} d_{\text{model}} 3 d_{\text{attn}}$	$2 n_{\text{layer}} d_{\text{model}} 3 d_{\text{attn}}$
Attention: Mask	—	$2 n_{\text{layer}} n_{\text{ctx}} d_{\text{attn}}$
Attention: Project	$n_{\text{layer}} d_{\text{attn}} d_{\text{model}}$	$2 n_{\text{layer}} d_{\text{attn}} d_{\text{embd}}$
Feedforward	$n_{\text{layer}} 2 d_{\text{model}} d_{\text{ff}}$	$2 n_{\text{layer}} 2 d_{\text{model}} d_{\text{ff}}$
De-embed	—	$2 d_{\text{model}} n_{\text{vocab}}$
Total (Non-Embedding)	$N = 2 d_{\text{model}} n_{\text{layer}} (2 d_{\text{attn}} + d_{\text{ff}})$	$C_{\text{forward}} = 2N + 2 n_{\text{layer}} n_{\text{ctx}} d_{\text{attn}}$

Table 1 Parameter counts and compute (forward pass) estimates for a Transformer model. Sub-leading terms such as nonlinearities, biases, and layer normalization are omitted.

Pretraining Data Sources

Download a large amount of publicly available data

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Table 1: **Pre-training data.** Data mixtures used for pre-training, for each subset we list the sampling proportion, number of epochs performed on the subset when training on 1.4T tokens, and disk size. The pre-training runs on 1T tokens have the same sampling proportion.

[Training data mixture used in Meta's LLaMA model]

Example Models

GPT-3 (2020)

50,257 vocabulary size
2048 context length
175B parameters
Trained on 300B tokens

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

Training: (rough order of magnitude to have in mind)

- O(1,000 - 10,000) V100 GPUs
- O(1) month of training
- O(1-10) $M

LLaMA (2023)

32,000 vocabulary size
2048 context length
65B parameters
Trained on 1-1.4T tokens

params	dimension	n heads	n layers	learning rate	batch size	n tokens
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

Table 2: **Model sizes, architectures, and optimization hyper-parameters.**

Training for 65B model:

- 2,048 A100 GPUs
- 21 days of training
- $5M

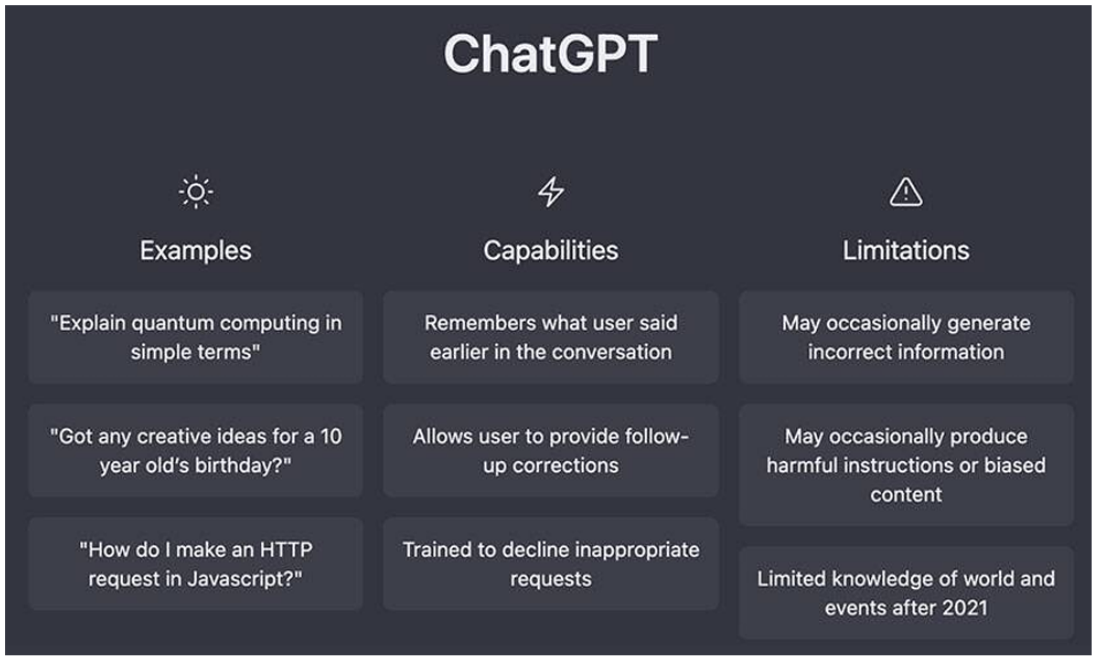
[Language Models are Few-Shot Learners, OpenAI 2020]
[LLaMA: Open and Efficient Foundation Language Models, Meta AI 2023]

Language models as multi-task assistants

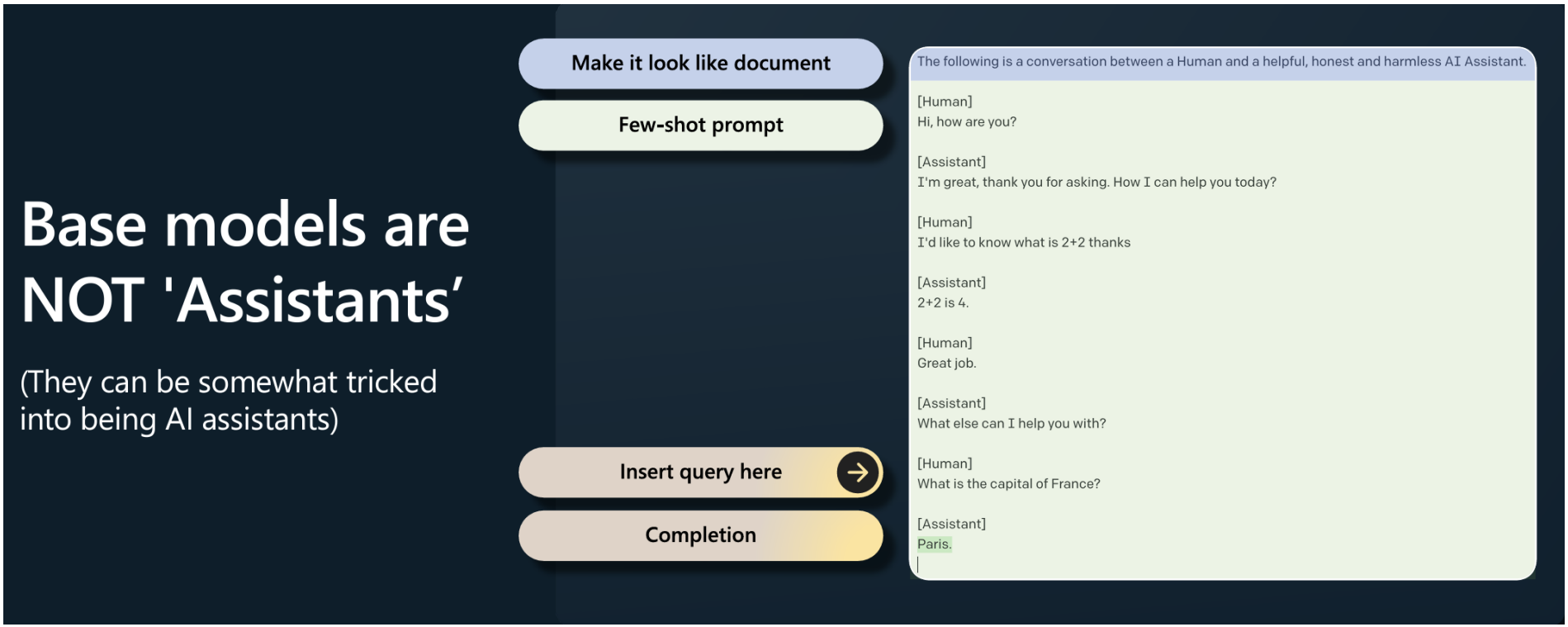
How do we get from *this*

Stanford University is located in _____

to *this*?



But we saw prompting can help?



Base models are not assistants

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

Language models are not *aligned* with user intent [Ouyang et al., 2022].

Base models are not assistants

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION **Human**
A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.

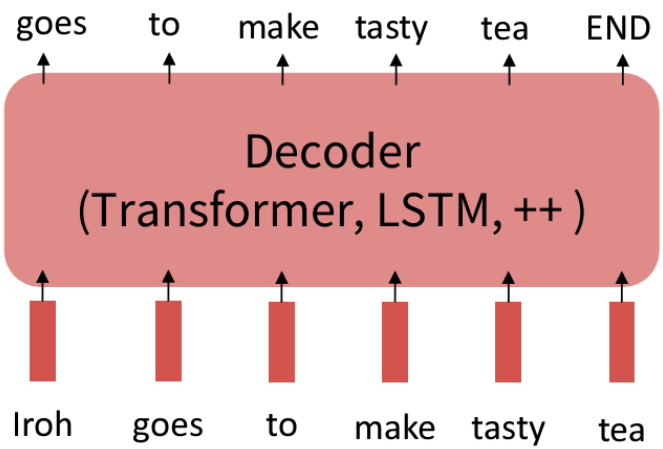
Language models are not *aligned* with user intent [Ouyang et al., 2022].
Finetuning to the rescue!

Scaling up Fine-tuning!

Pretraining can improve NLP applications by serving as parameter initialization.

Step 1: Pretrain (on language modeling)

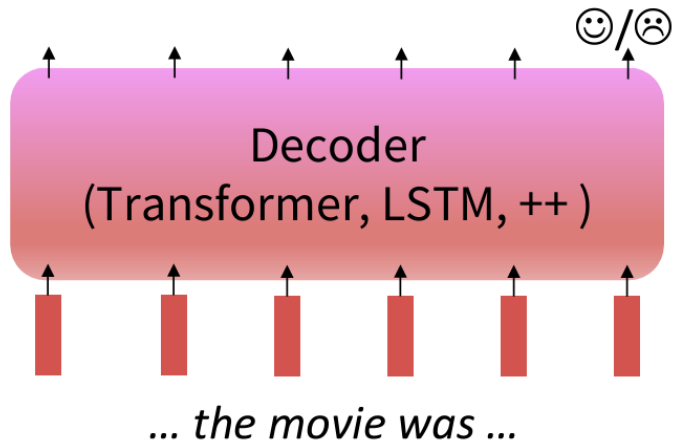
Lots of text; learn general things!



Base model

Step 2: Finetune (on many tasks)

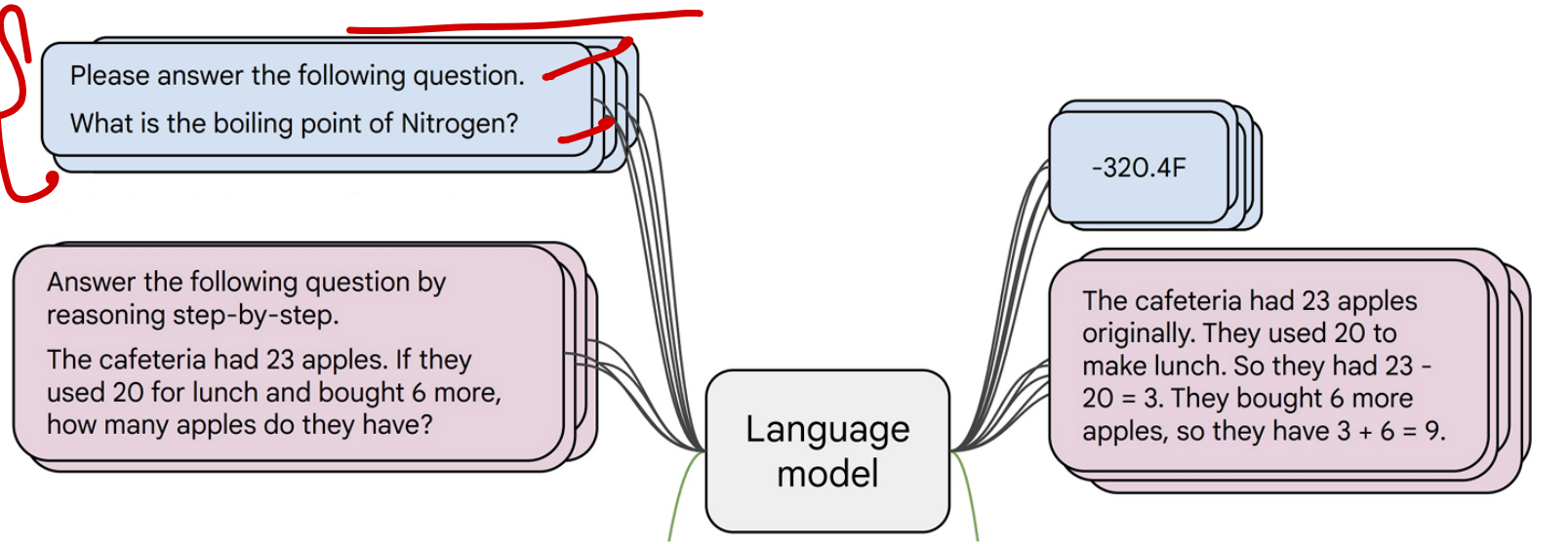
~~Not~~ many labels; adapt to the tasks!



Instruction fine-tuning!

FLAN-T5

- **Collect examples** of (instruction, output) pairs across many tasks and finetune an LM



- Evaluate on **unseen tasks**

How many tasks?

Finetuning tasks

T0-SF

Commonsense reasoning
Question generation
Closed-book QA
Adversarial QA
Extractive QA
Title/context generation
Topic classification
Struct-to-text
...

55 Datasets, 14 Categories, 193 Tasks

Muffin

Natural language inference
Code instruction gen.
Program synthesis
Dialog context generation
Closed-book QA
Conversational QA
Code repair
...

69 Datasets, 27 Categories, 80 Tasks

CoT (Reasoning)

Arithmetic reasoning
Commonsense Reasoning
Implicit reasoning
Explanation generation
Sentence composition
...

9 Datasets, 1 Category, 9 Tasks

Natural Instructions v2

Cause effect classification
Commonsense reasoning
Named entity recognition
Toxic language detection
Question answering
Question generation
Program execution
Text categorization
...

372 Datasets, 108 Categories, 1554 Tasks

- A **Dataset** is an original data source (e.g. SQuAD).
- A **Task Category** is unique task setup (e.g. the SQuAD dataset is configurable for multiple task categories such as extractive question answering, query generation, and context generation).
- A **Task** is a unique <dataset, task category> pair, with any number of templates which preserve the task category (e.g. query generation on the SQuAD dataset.)

How much compute?

Params	Model	Architecture	Pre-training Objective	Pre-train FLOPs	Finetune FLOPs	% Finetune Compute
80M	Flan-T5-Small	encoder-decoder	span corruption	1.8E+20	2.9E+18	1.6%
250M	Flan-T5-Base	encoder-decoder	span corruption	6.6E+20	9.1E+18	1.4%
780M	Flan-T5-Large	encoder-decoder	span corruption	2.3E+21	2.4E+19	1.1%
3B	Flan-T5-XL	encoder-decoder	span corruption	9.0E+21	5.6E+19	0.6%
11B	Flan-T5-XXL	encoder-decoder	span corruption	3.3E+22	7.6E+19	0.2%

Limitations of Instruction Fine-tuning

- One limitation of instruction finetuning is obvious: it's **expensive** to collect ground-truth data for tasks.
- But there are other, subtler limitations too. Can you think of any?
- **Problem 1:** tasks like open-ended creative generation have no right answer.
  - *Write me a story about a dog and her pet grasshopper.*
- **Problem 2:** language modeling penalizes all token-level mistakes equally, but some errors are worse than others.
- Even with instruction finetuning, there a mismatch between the LM objective and the objective of "satisfy human preferences"!
- Can we **explicitly attempt to satisfy human preferences?**