CS388: Natural Language Processing
Lecture 9: Pre-trained Decoders, GPT

TEXAS

The University of Texas at Austin

Administration

- Homework 2: due 24th
- Quiz 2: 09/29

Recap: BERT Objective

- Input: [CLS] Text chunk 1 [SEP] Text chunk 2
- BERT objective: masked LM + next sentence prediction

[CLS] *John visited* **[MASK]** *yesterday and really* **[MASK]** *it* [SEP] *I* **[MASK]** *Madonna.*

Devlin et al. (2019)

Today

- Decoder language models (GPT): scaling LMs further
- Decoding strategies: beam search, nucleus sampling
- Prompting: a new way of using large language models without taking any gradient steps
- Seq2seq pre-trained models (BART, T5): how can we leverage the same kinds of ideas we saw in BERT for seq2seq models like machine translation?
- Do we need seq2seq?

GPT

OpenAI GPT/GPT2

- Very large language models using the Transformer architecture
- Straightforward **decoder** language model, trained on raw text
- GPT2: trained on 40GB of text

	Parameters	Layers	d_{model}
	117M	12	768
approximate size of BERT	345M	24	1024
	762M	36	1280
GPT-2	1542M	48	1600

Radford et al. (2019)

Encoders vs. Decoders

- BERT is a Transformer **encoder**: **bidirectional** attention, trained with masked language modeling

$$P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$$

- GPT-n and other Transformer language models are **decoders**: **unidirectional** attention, trained to predict the next word

$$P(x_i \mid x_1, \ldots, x_{i-1})$$

Encoders vs. Decoders

Encoder: $P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$

- **To use in practice:** Ignore this probability distribution. Fine-tune the model for some other task P(*y* | ***x***)

Decoder: $P(x_i \mid x_1, \ldots, x_{i-1})$

- You can treat this like a decoder: ignore this probability distribution and train a model for P(*y* | ***x***). But encoders are better for this due to bidirectional attention

- **To use in practice:** we use this model to actually **generate** text

- **Surprisingly:** this turns out to be better at scale

OpenAI GPT2

SYSTEM PROMPT (HUMAN-WRITTEN)	*Miley Cyrus was caught shoplifting from Abercrombie and Fitch on Hollywood Boulevard today.*

MODEL COMPLETION (MACHINE-WRITTEN, SECOND TRY)	`The 19-year-old singer was caught on camera being escorted out of the store by security guards.`
	
	`The singer was wearing a black hoodie with the label 'Blurred Lines' on the front and 'Fashion Police' on the back.`
	
	`Scroll down for video`
	
	`Shoplifting: Miley Cyrus was caught shoplifting from Abercrombie and Fitch on Hollywood Boulevard today (pictured)`

- We'll see in a few mins how this was generated!

slide credit: OpenAI

Pre-Training Cost (with Google/AWS)

- BERT: Base $500, Large $7000
- GPT-2 (as reported in other work): $25,000
- This is for a single pre-training run…developing new pre-training techniques may require many runs
- *Fine-tuning* these models can typically be done with a single GPU
- Since then: OLMo 32B: 2.75M, GPT4 (100M)

https://syncedreview.com/2019/06/27/the-staggering-cost-of-training-sota-ai-models/
https://arxiv.org/abs/2512.13961, https://arxiv.org/pdf/2405.21015

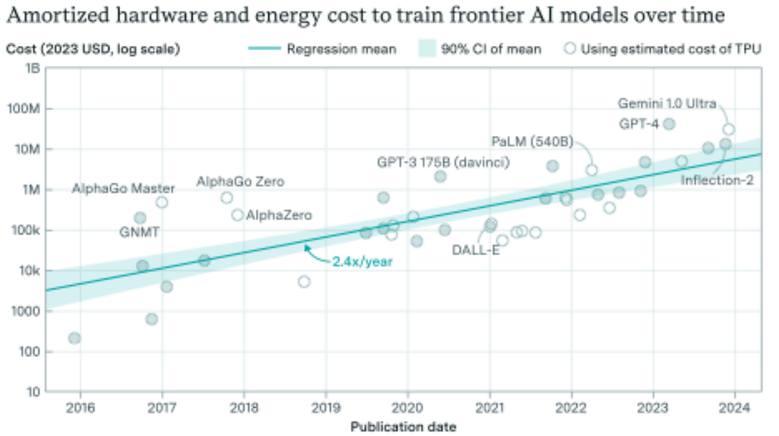
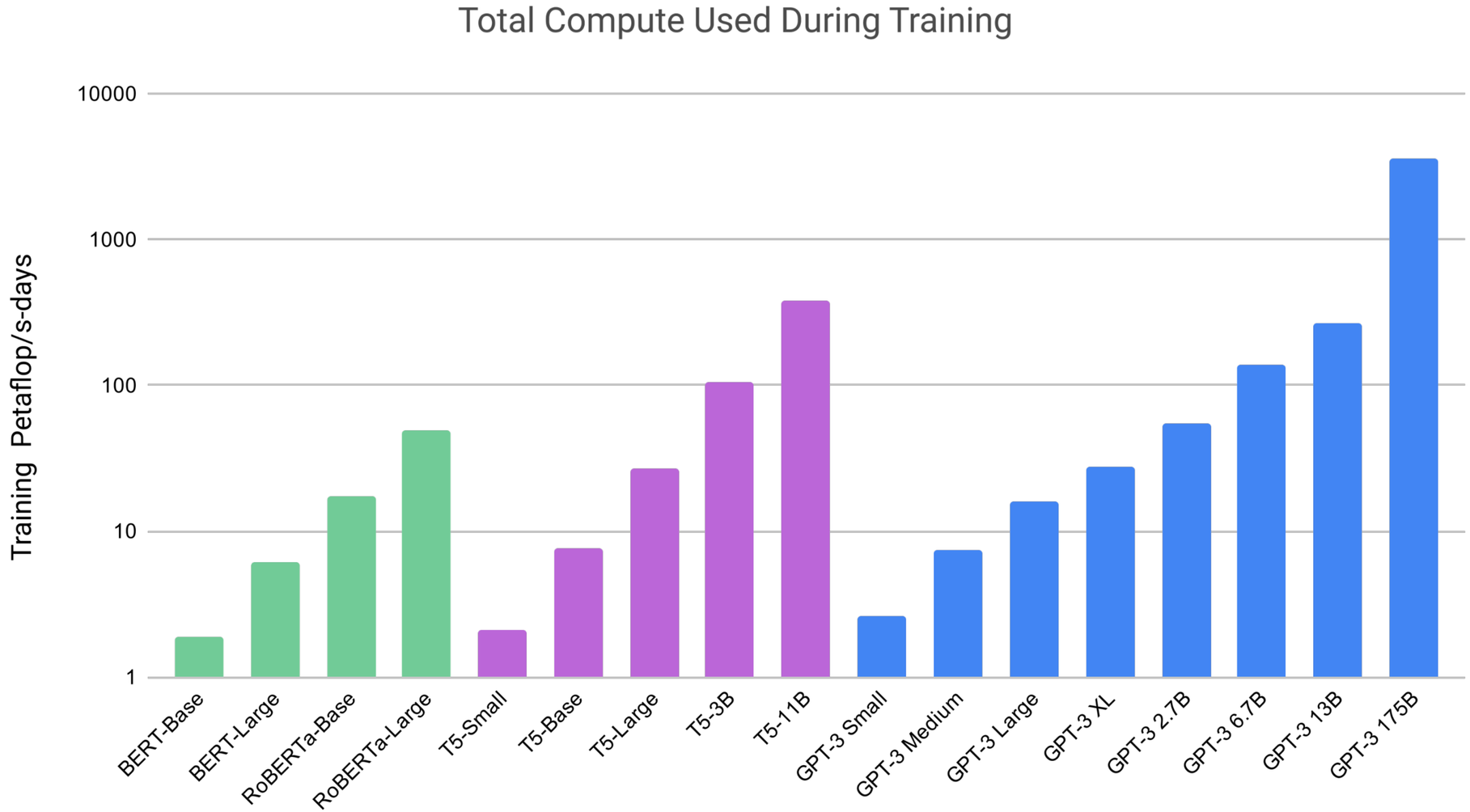


Figure 1: Amortized hardware cost plus energy cost for the final training run of frontier models. The selected models are among the top 10 most compute-intensive for their time. Amortized hardware costs are the product of training chip-hours and a depreciated hardware cost, with 23% overhead added for cluster-level networking. Open circles indicate costs which used an estimated production cost of Google TPU hardware. These costs are generally more uncertain than the others, which used actual price data rather than estimates.

Pushing the Limits: GPT-3

- 175B parameter model: 96 layers, 96 heads, 12k-dim vectors
- Trained on Microsoft Azure, estimated to cost roughly $10M



Brown et al. (2020)

Llama 1 + 2 + 3 + (4?)

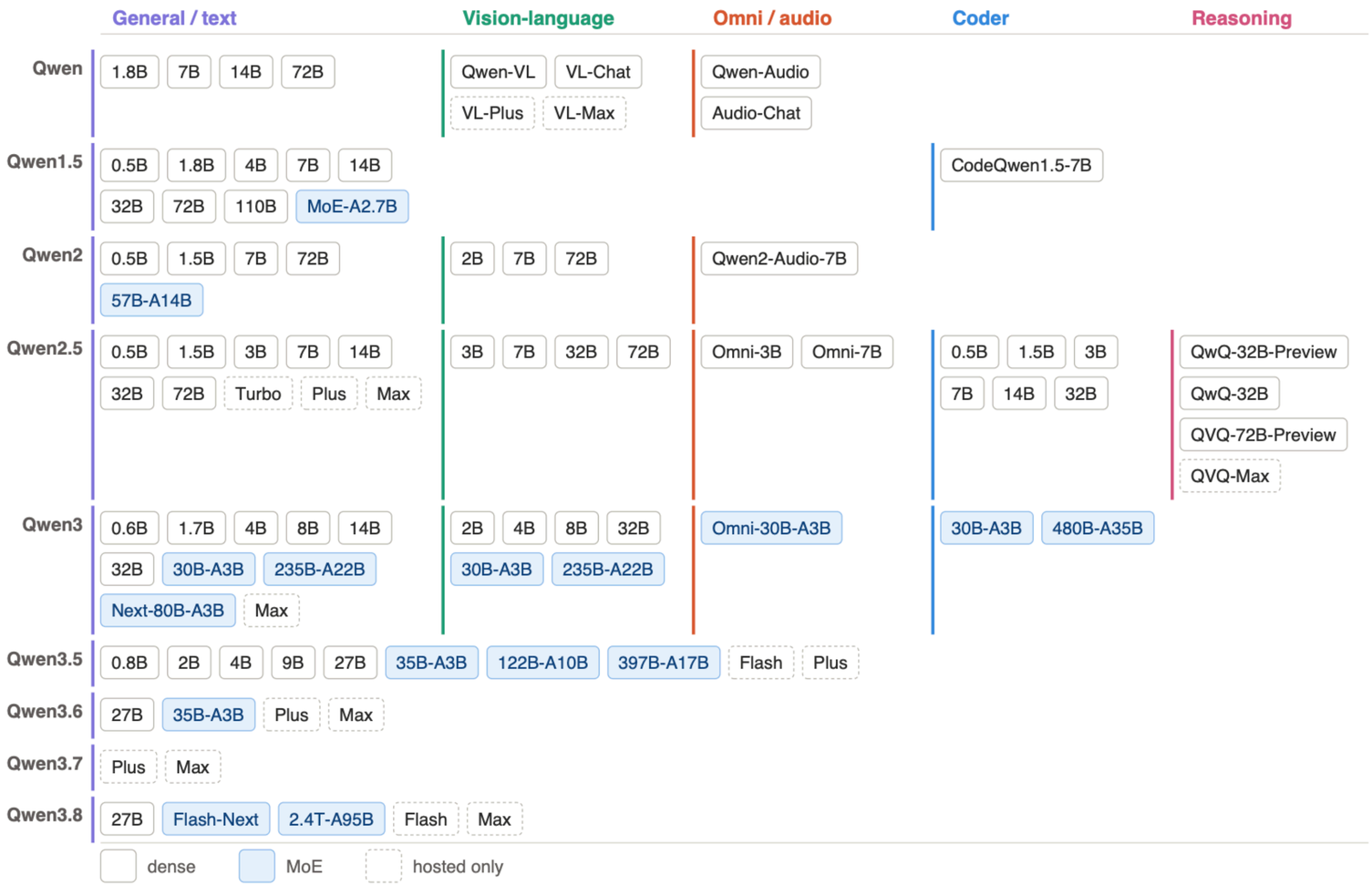
params	dimension	n heads	n layers	learning rate	batch size	n tokens
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

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

- Changes since GPT-3: approx. same architecture, relatively small tweaks
  - Tokenizer: byte pair encoding (what we said didn't work well…)
  - Rotary positional encodings, a few other small architecture changes
  - Optimized mix of pre-training data: Common Crawl, GitHub, Wikipedia, Books, etc.

Qwen (Alibaba)

- Very capable series of open-source models, has become defacto default



Decoding Methods

Decoding Strategies

- LMs place a distribution $P(x_i \mid x_1, \ldots, x_{i-1})$

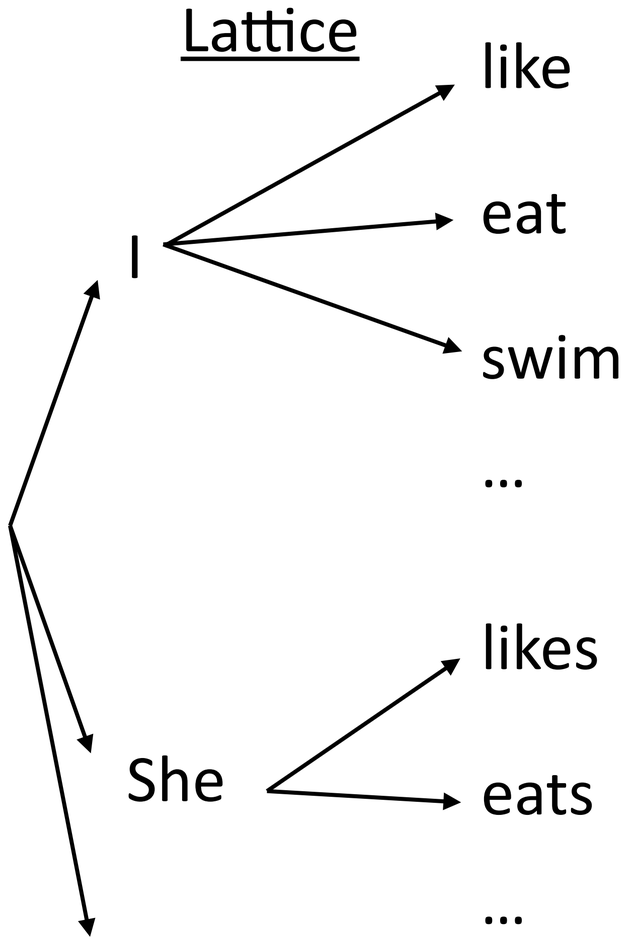
- How do we generate text from these?
  - Option 1: $\max x_i \; P(x_i \mid x_1, \ldots, x_{i-1})$ — take greedily best option
  - Option 2: sample from the model; draw x_i from that distribution
  - Option 3: use beam search to find the sequence with the highest prob.

- How do we find the highest probability option?

Holtzman et al. (2019)

Beam Search

- Time-synchronous search over the timesteps of generation, with a fixed number of options kept on the fringe (beam size=3 on this slide):



Step 1 beam:

I	0.01
She	0.003
He	0.002

- All other options pruned

Step 2 beam:

I like	0.003
She likes	0.002
I eat	0.001

- Have to consider k * |V| options for this beam

Decoding Strategies

- Story generation (this is with GPT-2):

Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

- Beam search degenerates and starts repeating. If you see a fragment repeated 2-3x, it has very high probability to keep repeating
- Sampling is too noisy — introduces many grammatical errors

Holtzman et al. (2019)

Degeneration

- Beam search fails because the model is *locally normalized*
- Let's look at all the individual decisions that get made here

Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

P(Nacional | … Universidad) is high

P(Autónoma | … Universidad Nacional) is high

P(de | … Universidad Nacional Autónoma) is high

P(México | Universidad Nacional Autónoma de) is high

P(/ | … México) and P(Universidad | … México /) — these probabilities may be low. But those are just 2/6 words of the repeating fragment

- **Each word is likely given the previous words but the sequence is bad**

Holtzman et al. (2019)

Drawbacks of Sampling

- Sampling is “too random”

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV

P(*y* | … they live in a remote desert uninterrupted by)

0.01 roads
0.01 towns
0.01 people
0.005 civilization

Good options, maybe accounting for 90% of the total probability mass. So a 90% chance of getting something good

…
0.0005 town

Long tail with 10% of the mass

Holtzman et al. (2019)

Nucleus Sampling

P(*y* | … they live in a remote desert uninterrupted by)

0.01 roads

0.01 towns

0.01 people

0.005 civilization

cut off after *p*% of mass

renormalize and sample

- Define a threshold *p.* Keep the most probable options account for *p*% of the probability mass (the *nucleus*), then sample among these.
- This idea existed in speech to some extent since 90's (Histogram Pruning, Adaptive Beam Search)
- To implement: sort options by probability, truncate the list once the total exceeds *p*, then renormalize and sample from it

Van Hamme & Van Aelten (1996) Steinbiss et al. (1994) Holtzman et al. (2019)

Prompting, In-Context Learning

Pre-GPT-3: Fine-tuning

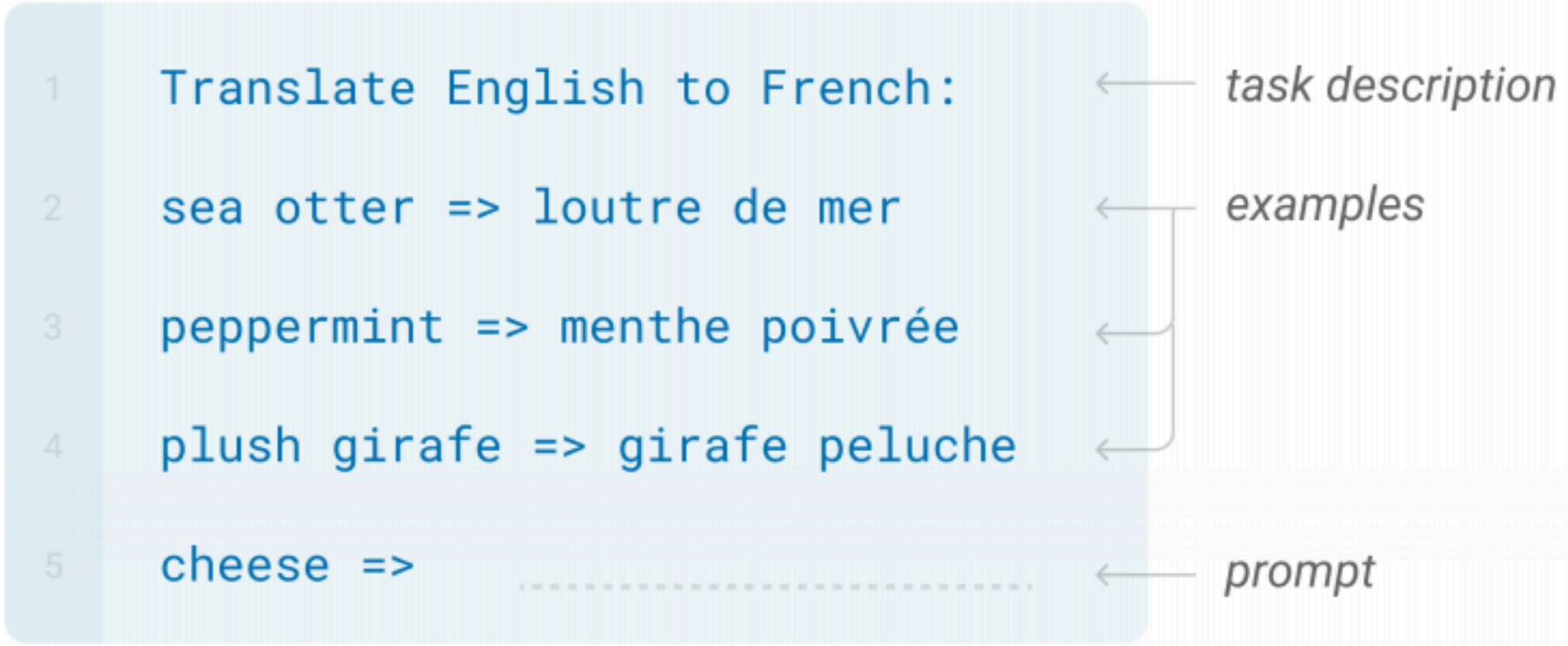
- Fine-tuning: this is the "normal way" of doing learning in models like GPT-2
- Requires computing the gradient and applying a parameter update on every example
- **This is super expensive with 175B parameters**

Brown et al. (2020)

GPT-3: Few-shot Learning

- GPT-3 proposes an alternative: **in-context learning.** Just uses the off-the-shelf model, no gradient updates

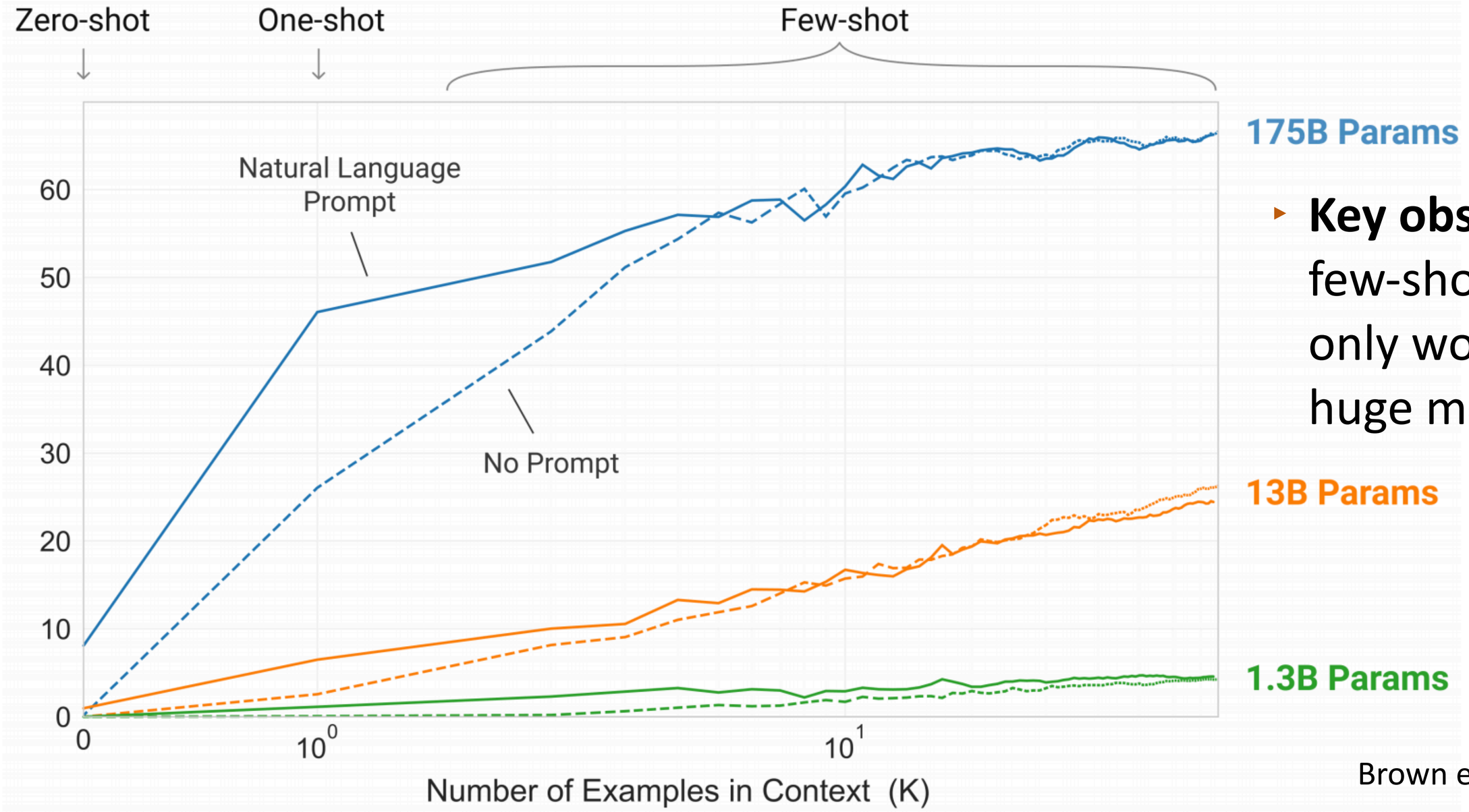
- This procedure depends heavily on the examples you pick as well as the prompt (“*Translate English to French*”)



Brown et al. (2020)

Why does ICL work?

GPT-3



- **Key observation:** few-shot learning only works with huge models!

Brown et al. (2020)

Why ICL+Transformers?

GPT-3

	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

- Sometimes very impressive, (MultiRC, ReCoRD), sometimes very bad
- Results on other datasets are equally mixed — but still strong for a few-shot model!

Brown et al. (2020)

Prompts

- Prompts can help induce the model to engage in certain behavior
- In the GPT-2 paper, “tl;dr:” (too long; didn't read) is mentioned as a prompt that frequently shows up in the wild **indicating a summary**
- tl;dr is an indicator that the model should “switch into summary mode” now — and if there are enough clean instances of tl;dr in the wild, maybe the model has been trained on a ton of diverse data?
- Good prompt + a few training examples in-context = strong task performance?

Brown et al. (2020)

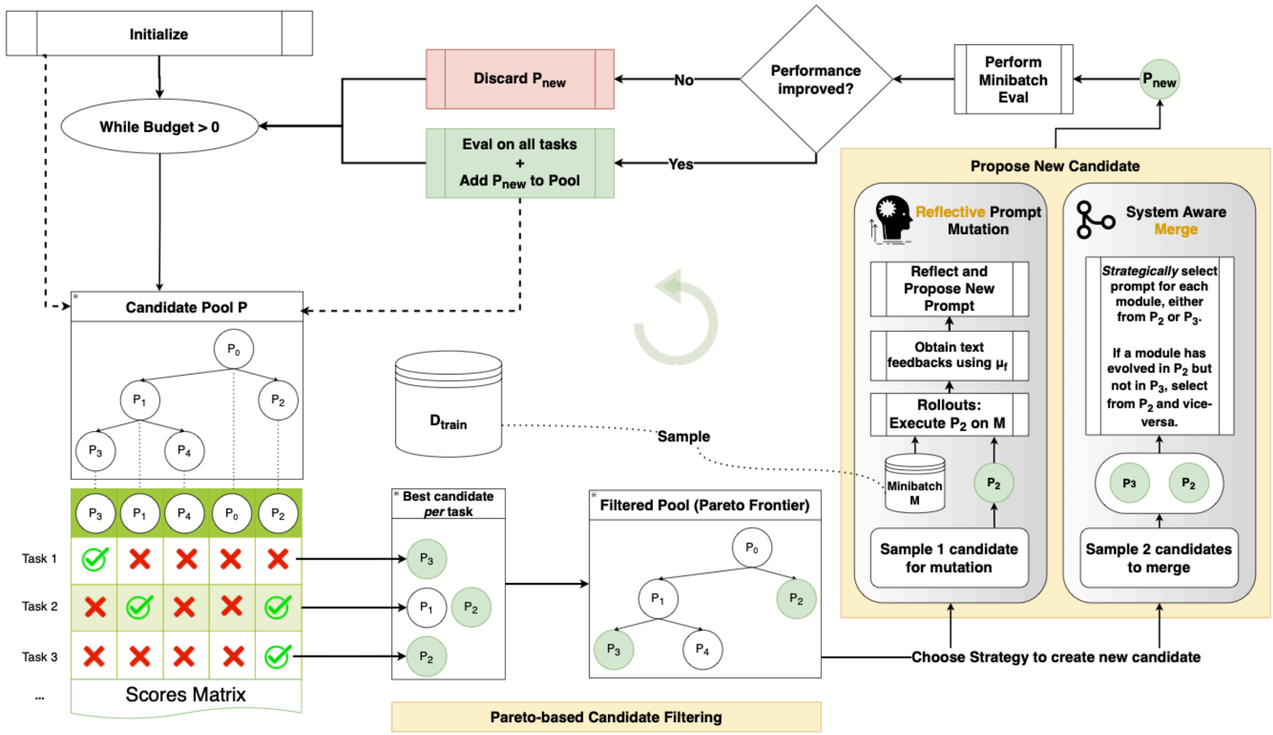
Prompts now

- Prompt design is still important
- Industry: outside of people training models, most of what people are doing is prompt design.
- Decide:
  - What task do you want the model to do?
  - What information should you condition on?
  - How should the output be formatted?

Prompt Optimization

- Can we automate this process?
- Start with a prompt and a loss —> ask LLM to rewrite —> keep better-performing prompts
- Easy way to improve but can overfit

Qwen3 8B	HotpotQA	IFBench	Hover	PUPA
Baseline	42.33	36.90	35.33	80.82
GRPO	43.33	35.88	38.67	86.66
MIPROv2	55.33	36.22	47.33	81.55
GEPA	62.33	**38.61**	**52.33**	**91.85**
GEPA+Merge	**64.33**	28.23	51.67	86.26



GEPA: Agrawal et al. (2026)

Seq2seq Pre-trained Models: BART, T5

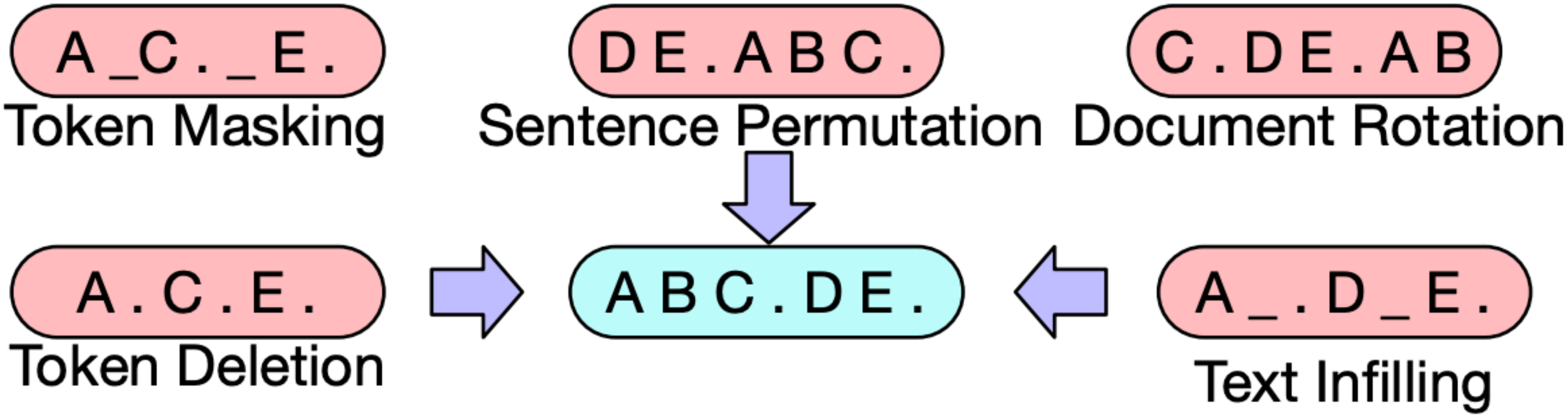
How do we pre-train seq2seq models?

- LMs P(**y**): trained unidirectionally
- Masked LMs: trained bidirectionally but with masking
- How can we pre-train a model for P(**y**|**x**)?
- Well, why was BERT effective?
  - Predicting a mask requires some kind of text “understanding”:
- What would it take to do the same for sequence prediction?

How do we pre-train seq2seq models?

- How can we pre-train a model for P(**y**|**x**)?
- Requirements: (1) should use unlabeled data; (2) should force a model to attend from **y** back to **x**

BART



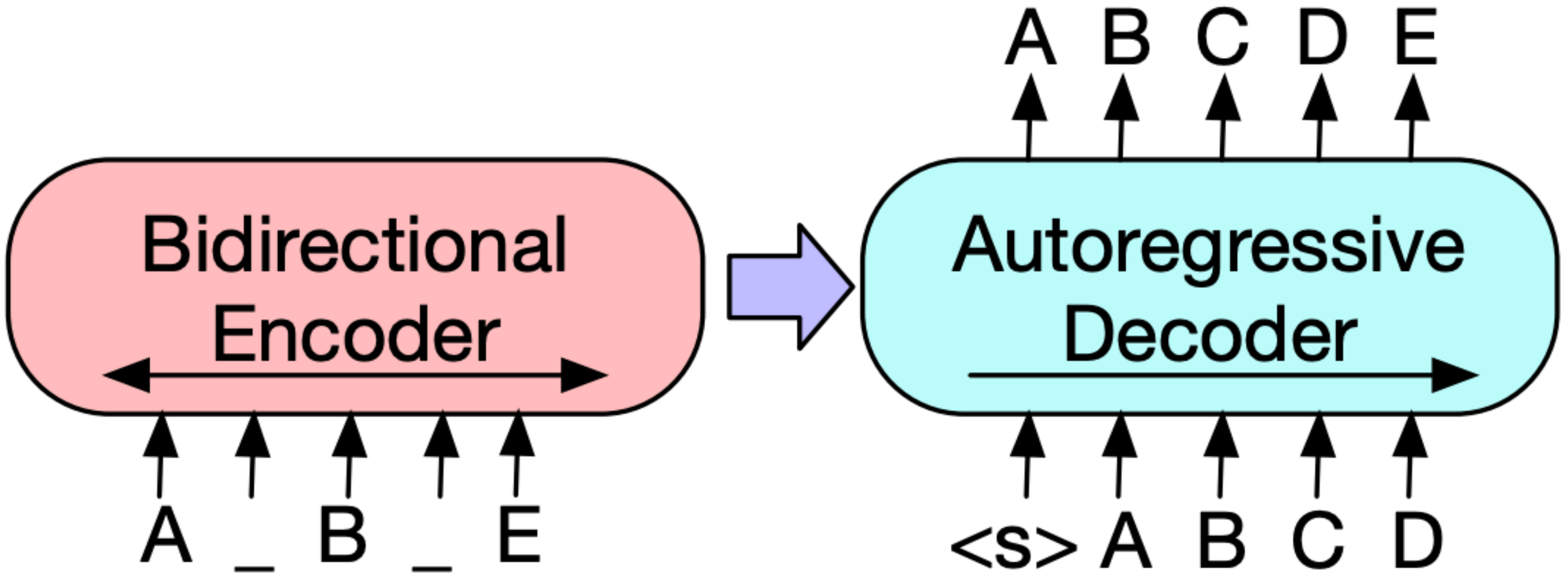
Infilling is longer spans than masking

- Several possible strategies for corrupting a sequence are explored in the BART paper

Lewis et al. (2019)

BART

- Sequence-to-sequence Transformer trained on this data: permute/ make/delete tokens, then predict full sequence autoregressively

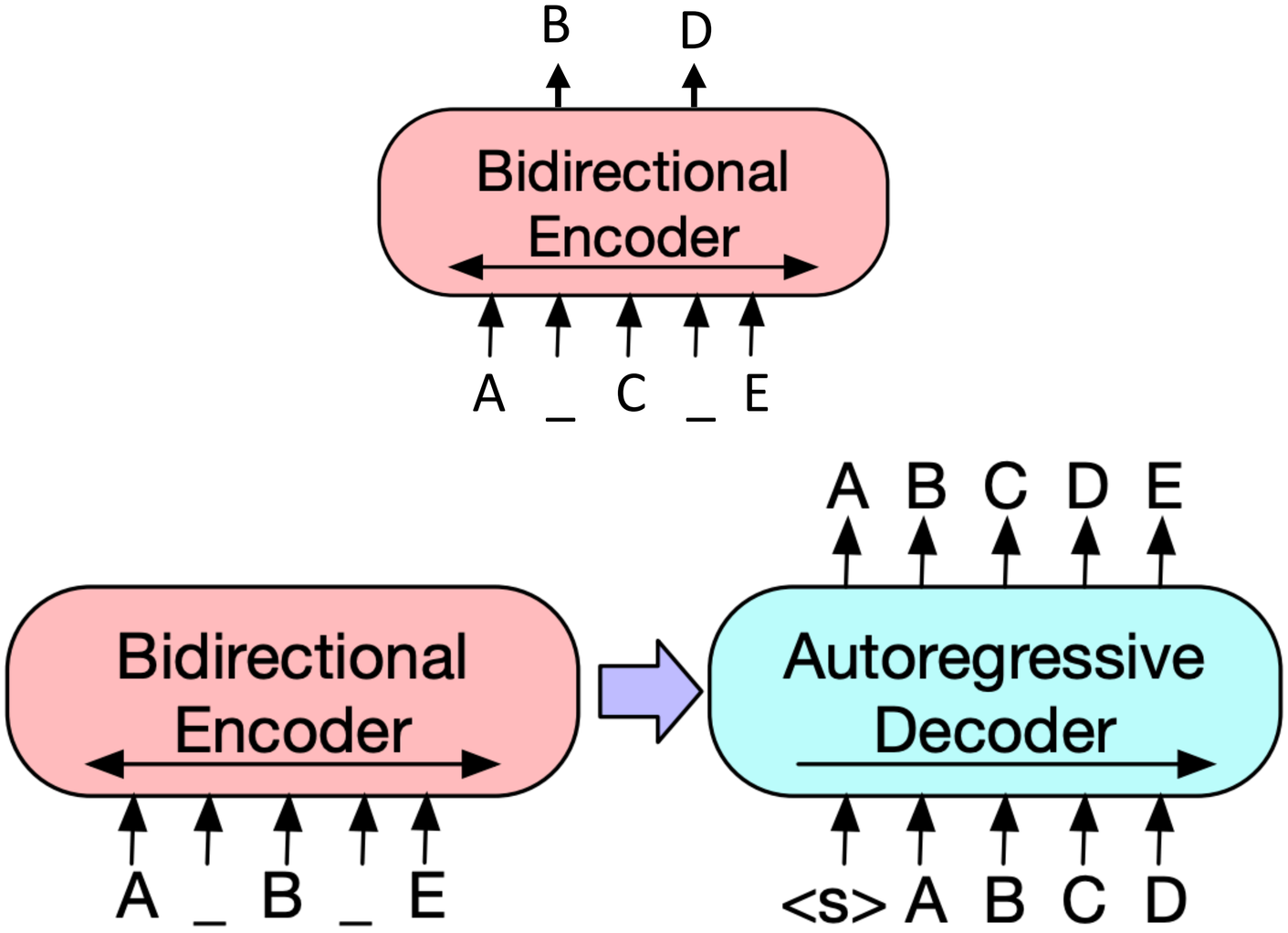


Lewis et al. (2019)

BERT vs. BART

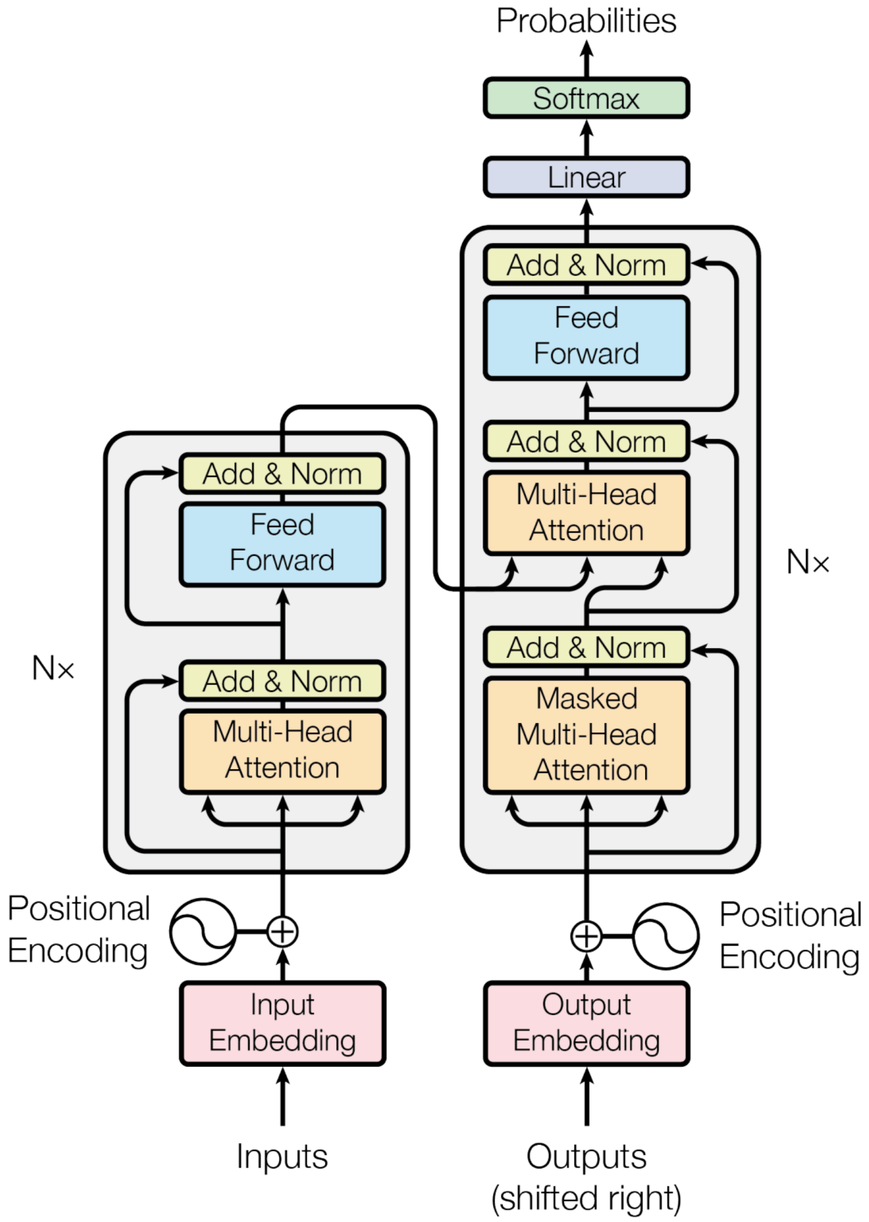
- BERT: only parameters are an encoder, trained with masked language modeling objective. Cannot generate text or do seq2seq tasks

- BART: both an encoder and a decoder. Can also use just the encoder wherever we would use BERT



Lewis et al. (2019)

Seq2seq Architecture

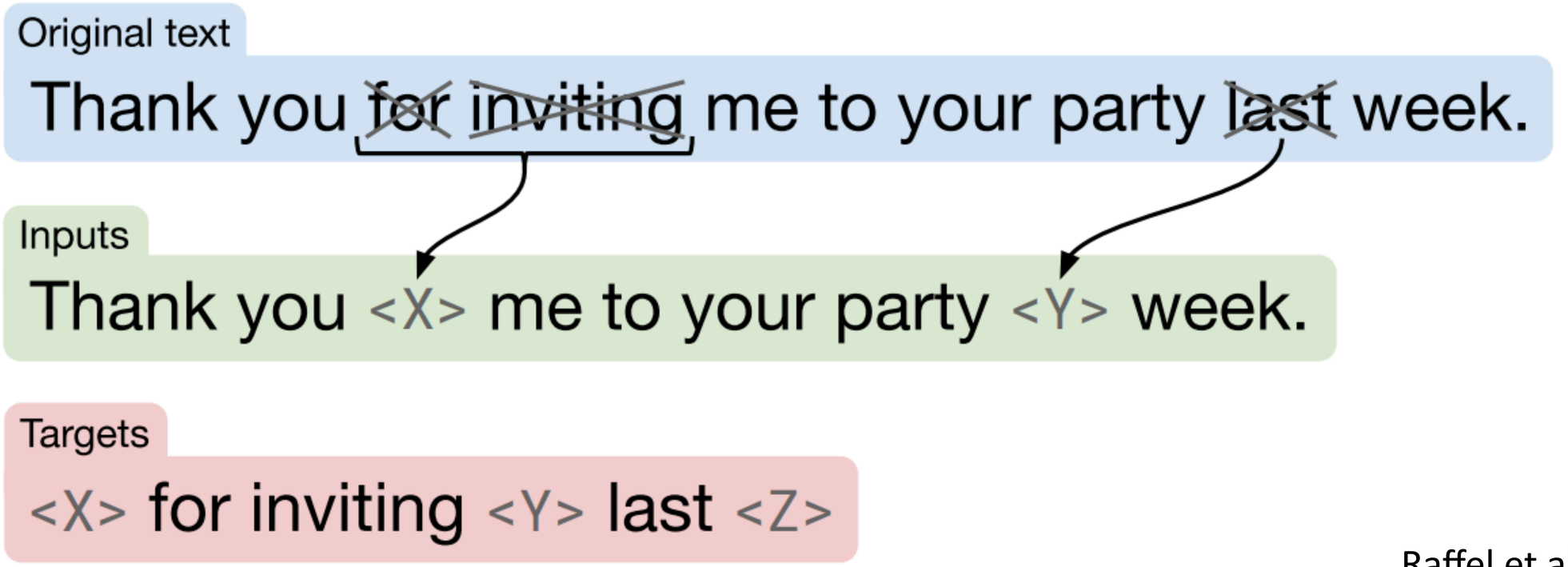


- Encoder-decoder model is structurally similar to your language model
- Modification: decoder now attends back to the input. But the input doesn't change, so this just needs to be encoded once

Lewis et al. (2019)

T5

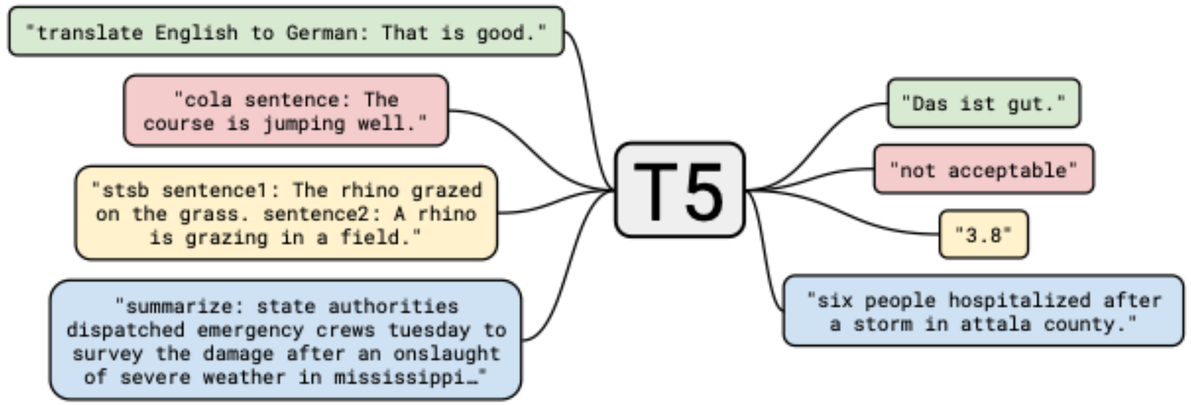
- Pre-training: similar denoising scheme to BART (they were released within a week of each other in fall 2019)
- Input: text with gaps. Output: a series of phrases to fill those gaps.



Raffel et al. (2019)

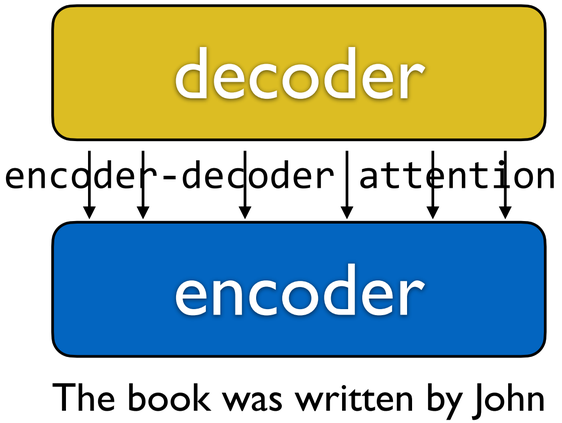
T5

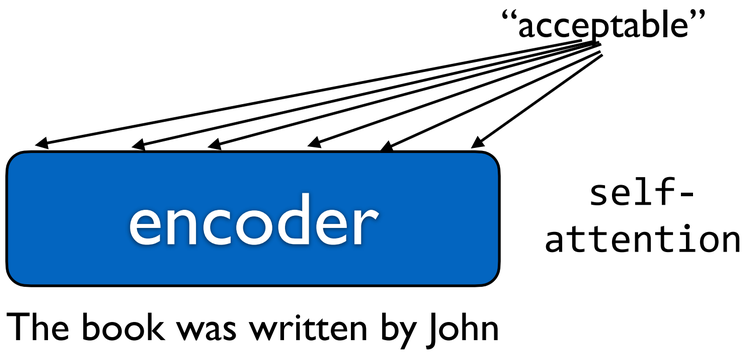
- Multi-task learning: finetune model for many NLP tasks at once (translation, acceptability, entailment, summarization)
- Key change: formulate everything as a seq2seq problem
- Example
  - Before: COLA Acceptability sentence —> [0, 1]
  - Now: sentence —> “acceptable”, “not acceptable”



Do we still need seq2seq?

- Short answer: not really
- Just use more self-attention!





Takeaways

- Pre-trained seq2seq models and generative language models can do well at lots of generation tasks
- Decoding strategy can matter a lot (beam search vs. sampling)
- Prompting is a way to harness their power and learn to do many tasks with a single model. Can be done without fine-tuning