



Pre-training and Fine-tuning

Kun Yuan

Center for Machine Learning Research @ Peking University

Contents

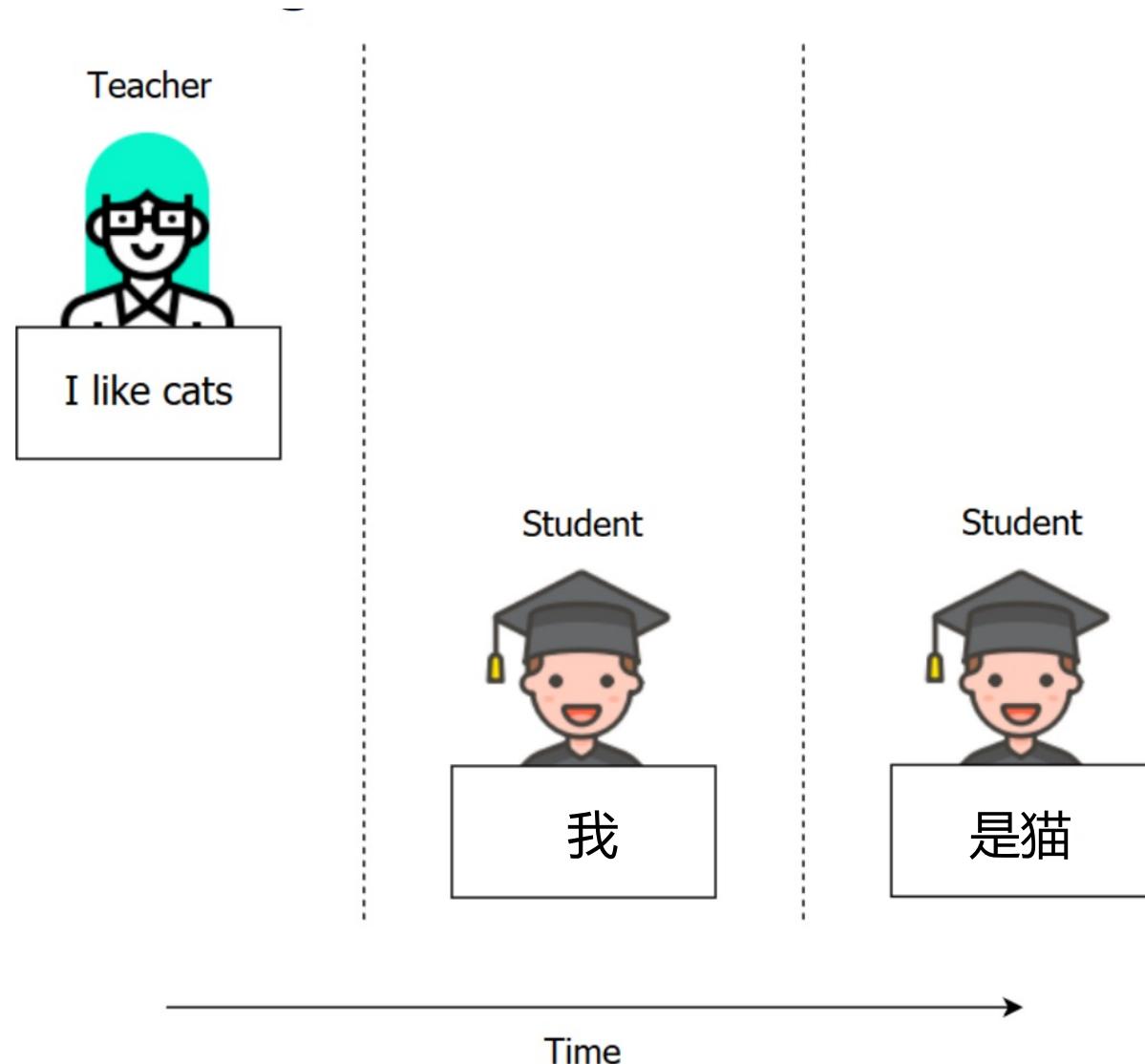


- Transformer review: teacher forcing
- BERT
- GPT

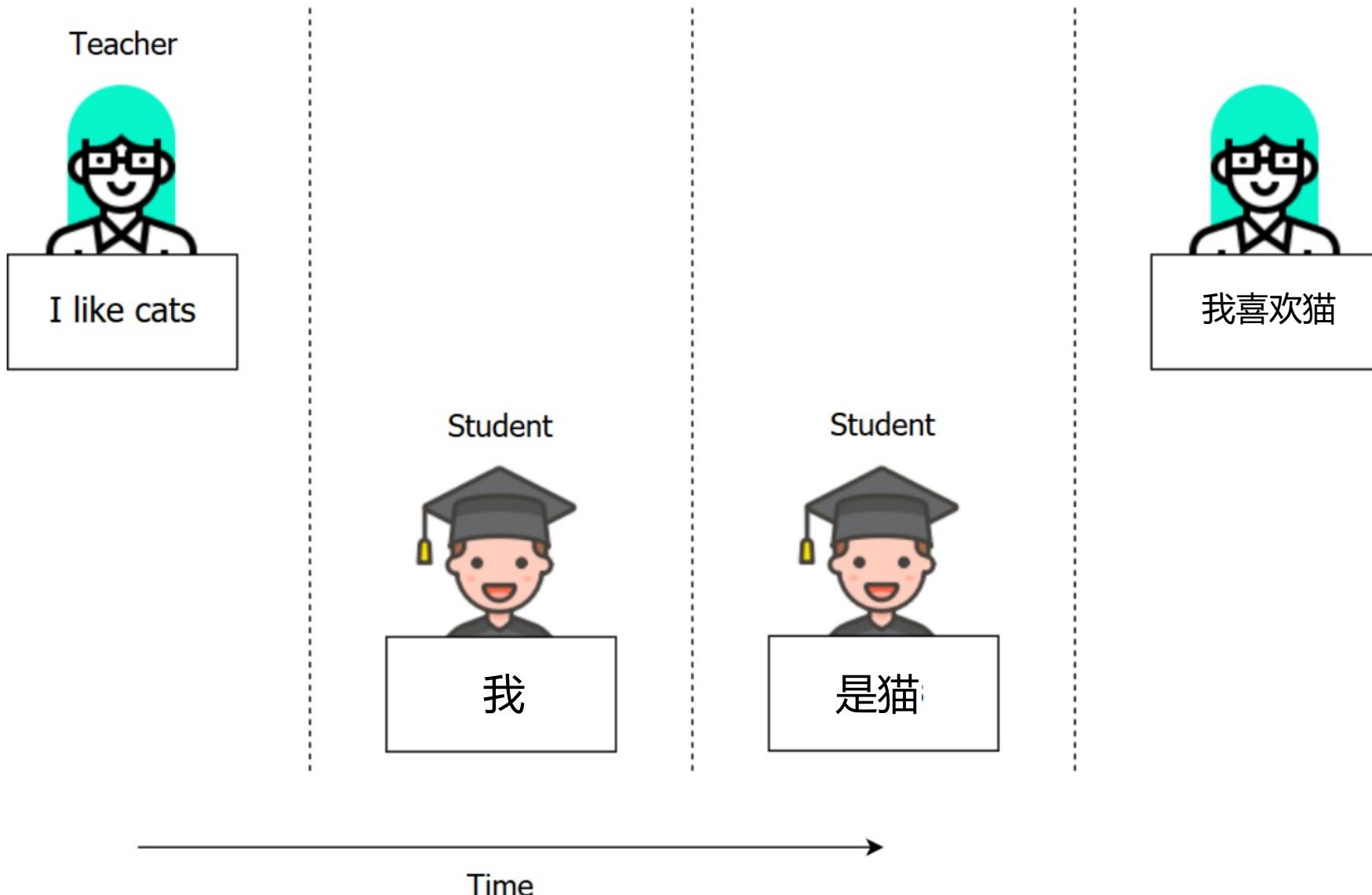
Some materials are from a great course [1]

[1] Stanford CS224n: Natural Language Processing with Deep Learning

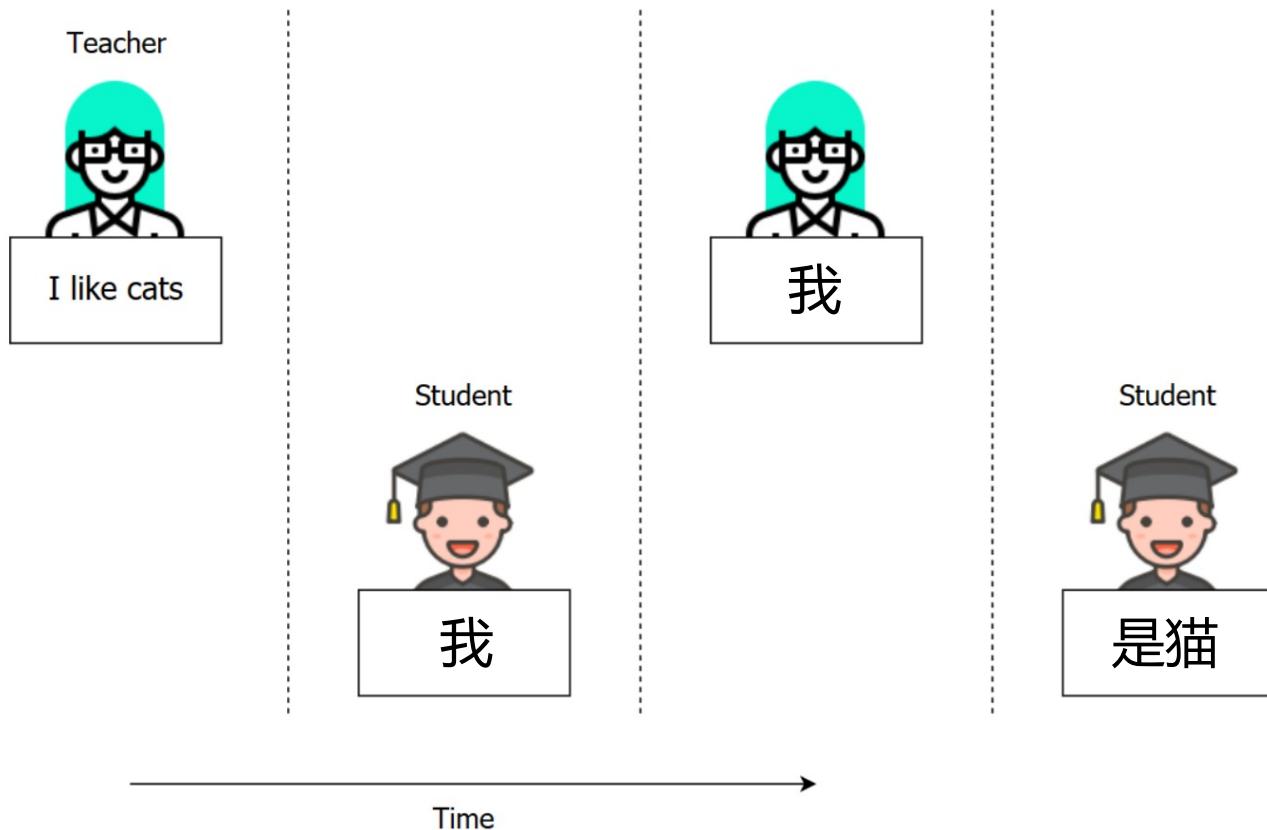
Training without teacher forcing



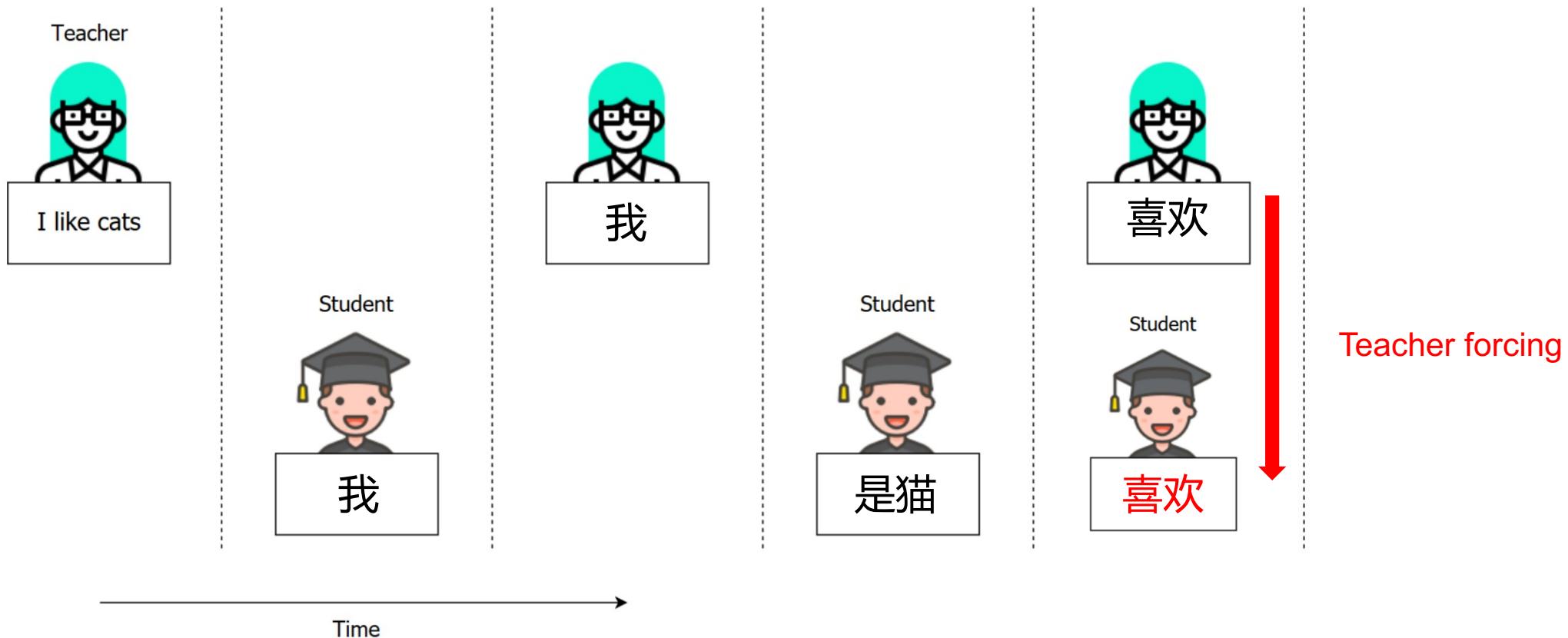
Training without teacher forcing



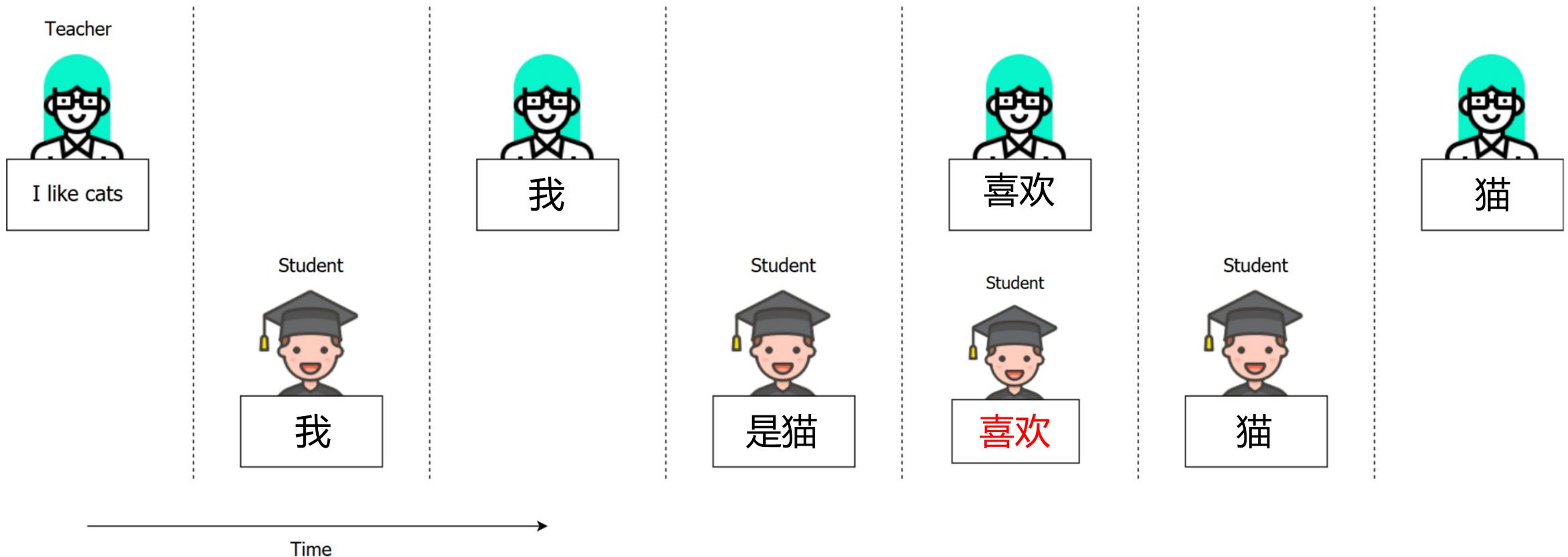
Training with teacher forcing



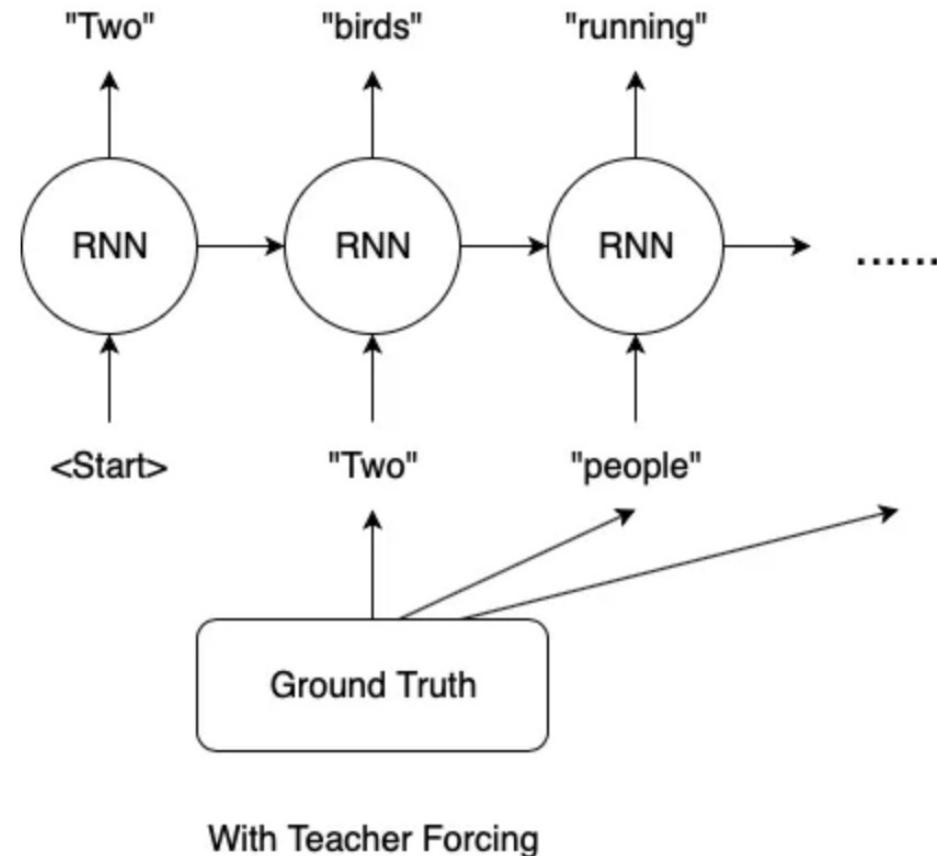
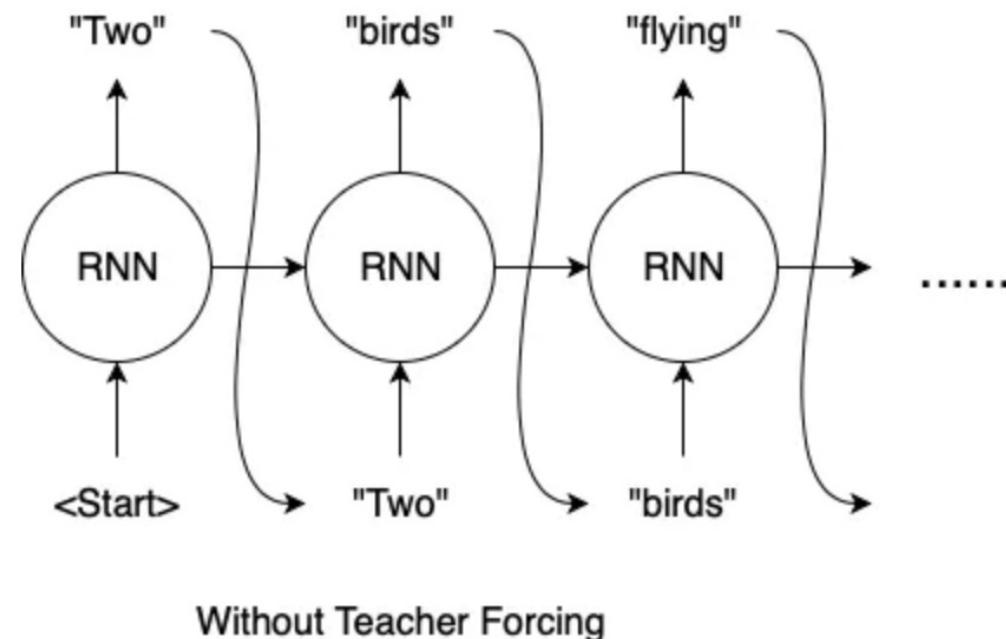
Training with teacher forcing



Training with teacher forcing



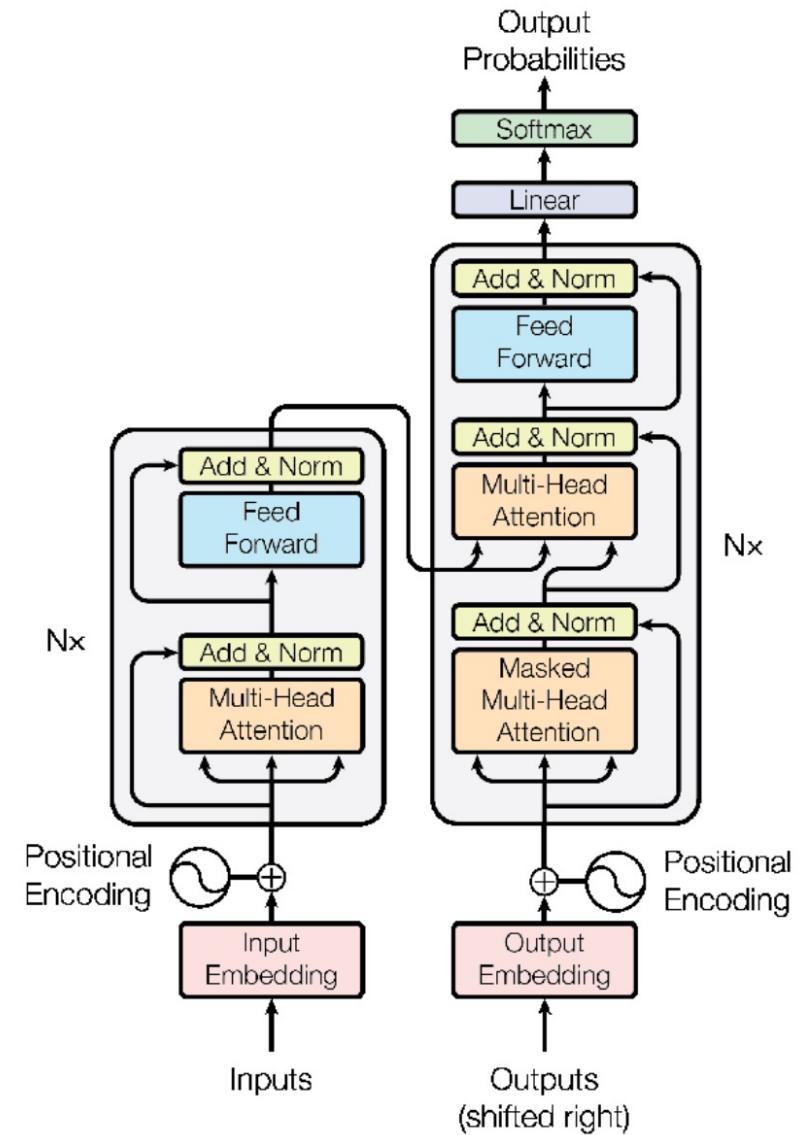
With and without teacher forcing



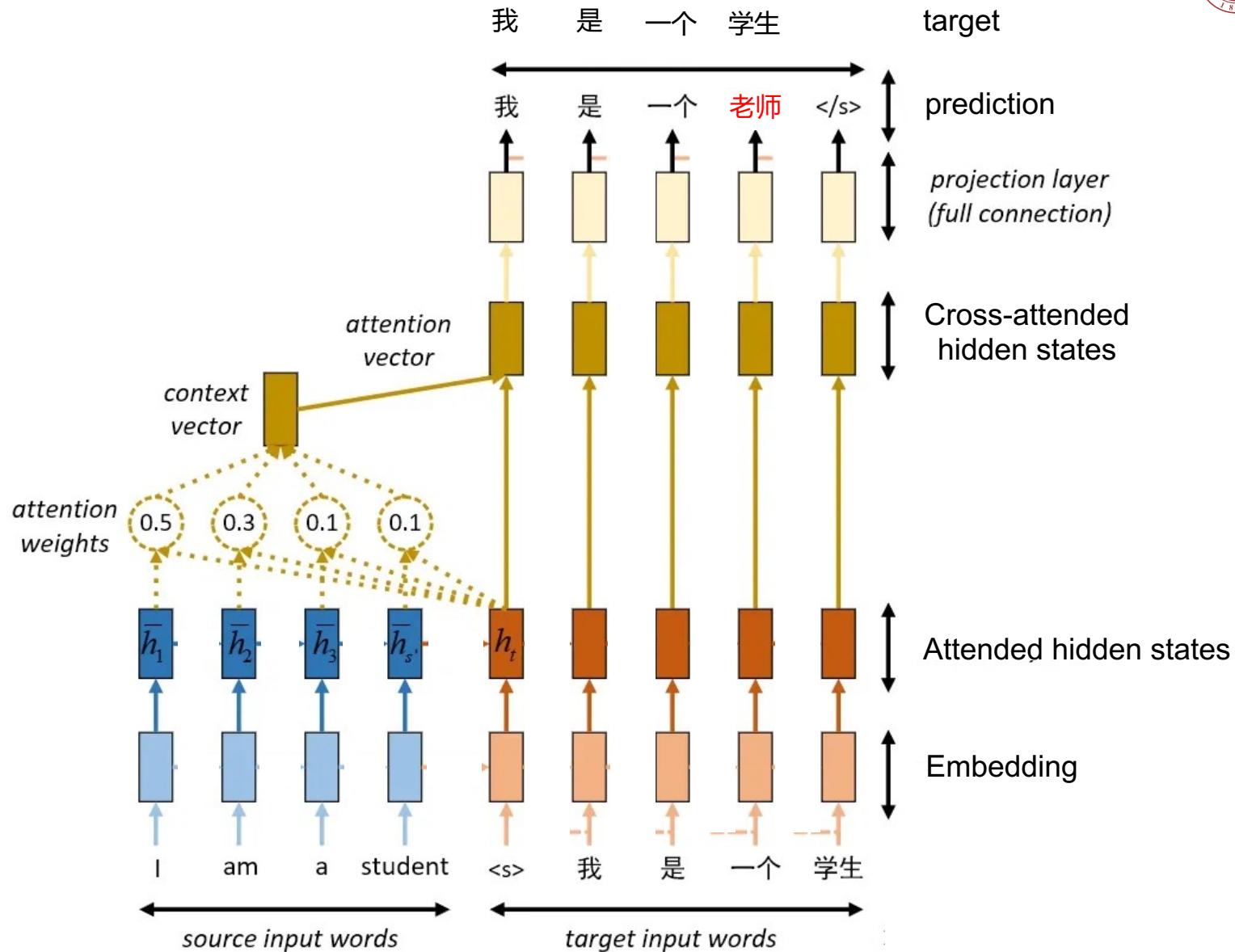
Transformer

Teacher forcing is used when training transformer

- To avoid error accumulation
- To accelerate training due to parallel computing endowed by masked attention

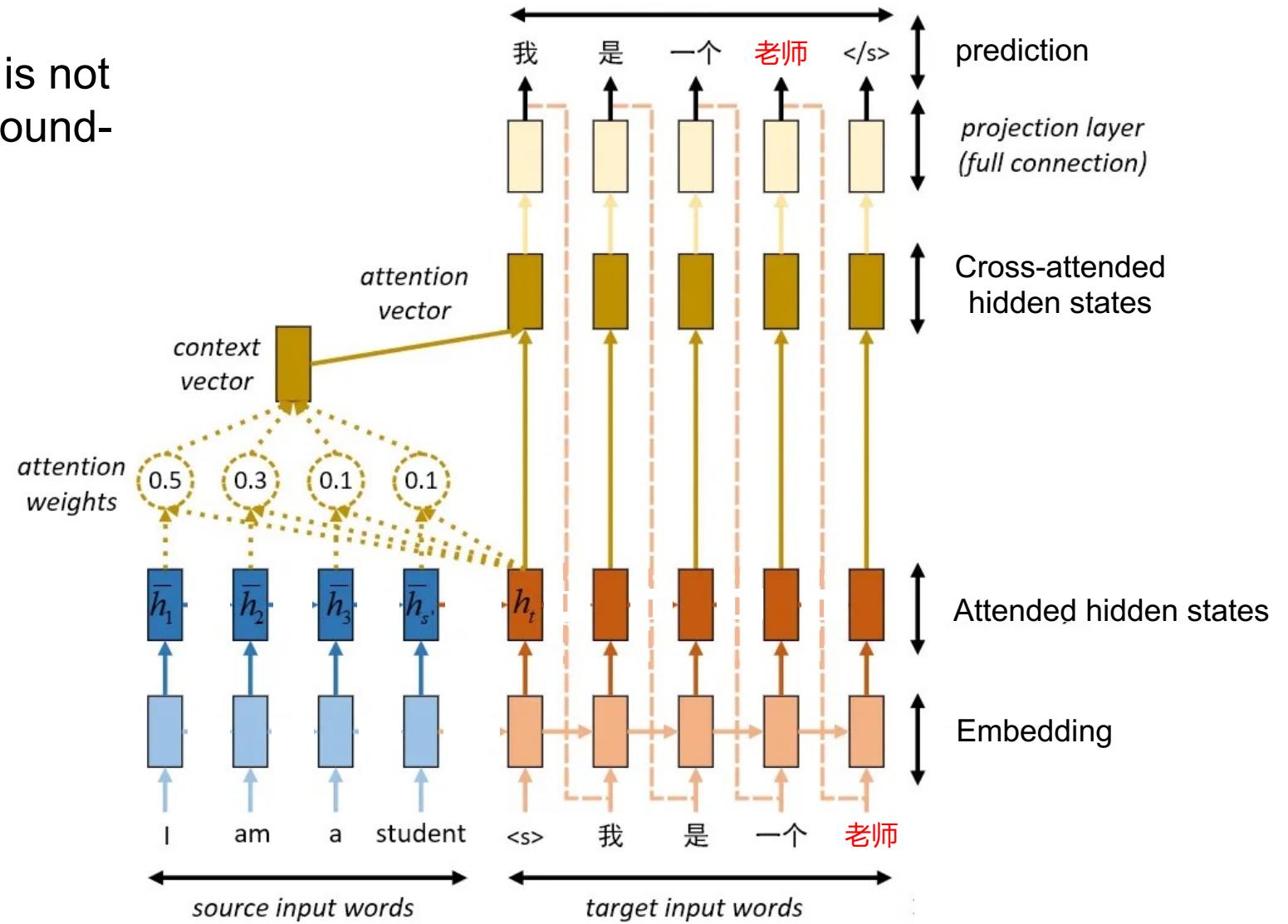


Training transformer with teacher forcing



During inference, teacher forcing is not used since we do not have the ground-truth label

- Cannot be parallel!

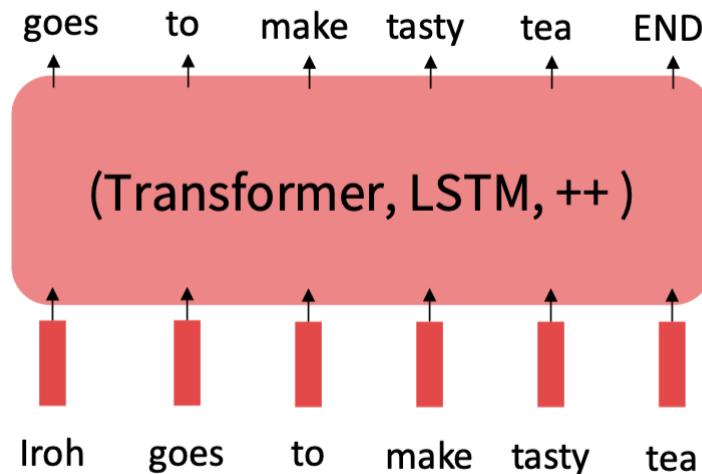


The Pretraining / Finetuning Paradigm

Pretraining can improve NLP applications by serving as parameter initialization.

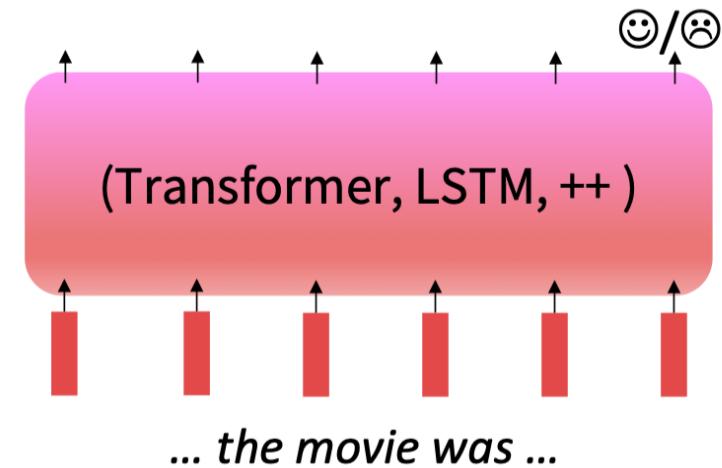
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



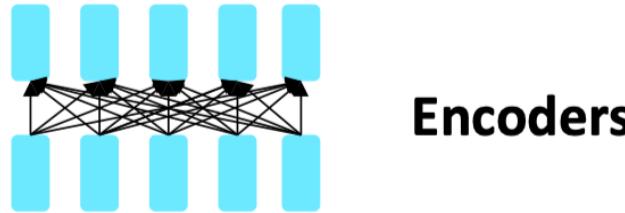
Step 2: Finetune (on your task)

Not many labels; adapt to the task!



[Figure is from Stanford CS224n]

BERT: Bidirectional Encoder Representations from Transformers

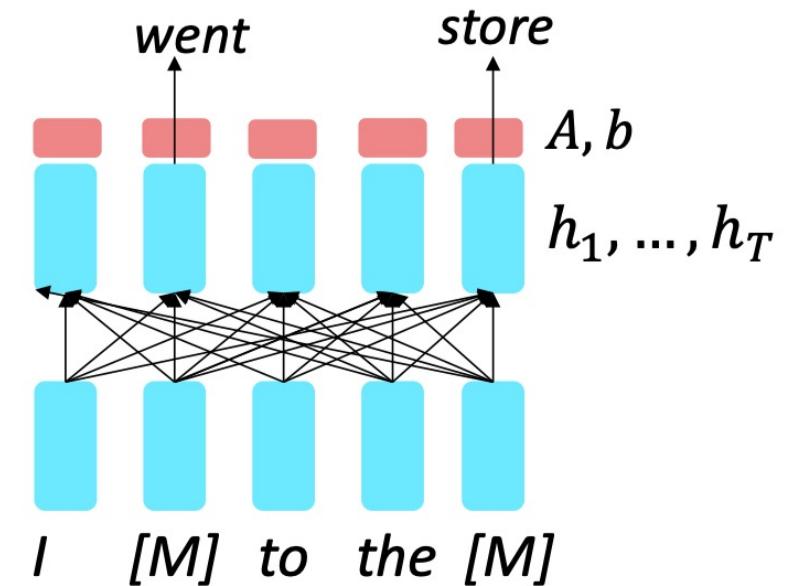


- Gets bidirectional context – can condition on future!
- How do we train them to build strong representations?

Devlin et al., 2018 proposed the “Masked LM” objective and **released the weights of a pretrained Transformer**, a model they labeled BERT.

BERT, roughly speaking, pre-trained the encoder of the transformer.

It replace some fraction of words in the input with a special [MASK] token; predict these words.



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova

Google AI Language

{jacobdevlin, mingweichang, kentonl, kristout}@google.com

BERT details

Details about BERT

- Two models were released:
 - BERT-base: 12 layers, 768-dim hidden states, 12 attention heads, 110 million params.
 - BERT-large: 24 layers, 1024-dim hidden states, 16 attention heads, 340 million params.
- Trained on:
 - BooksCorpus (800 million words)
 - English Wikipedia (2,500 million words)
- Pretraining is expensive and impractical on a single GPU.
 - BERT was pretrained with 64 TPU chips for a total of 4 days.
 - (TPUs are special tensor operation acceleration hardware)

BERT finetune

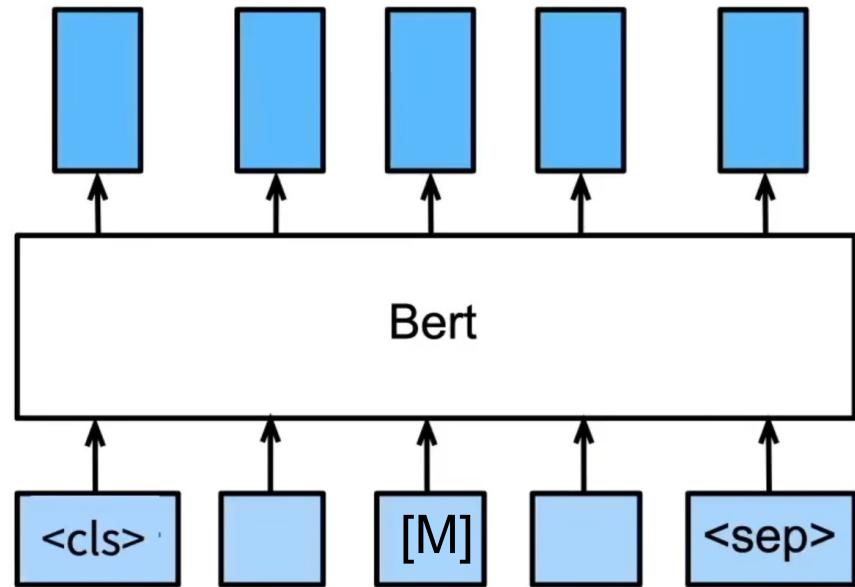
After pre-training, BERT can learn great representations for each word. We can use it as a starting point to finetune downstream tasks. In this way, BERT can perform well in that specific downstream task

To finetune BERT, we will add or change the output layer of the BERT model, and train all parameters with downstream dataset

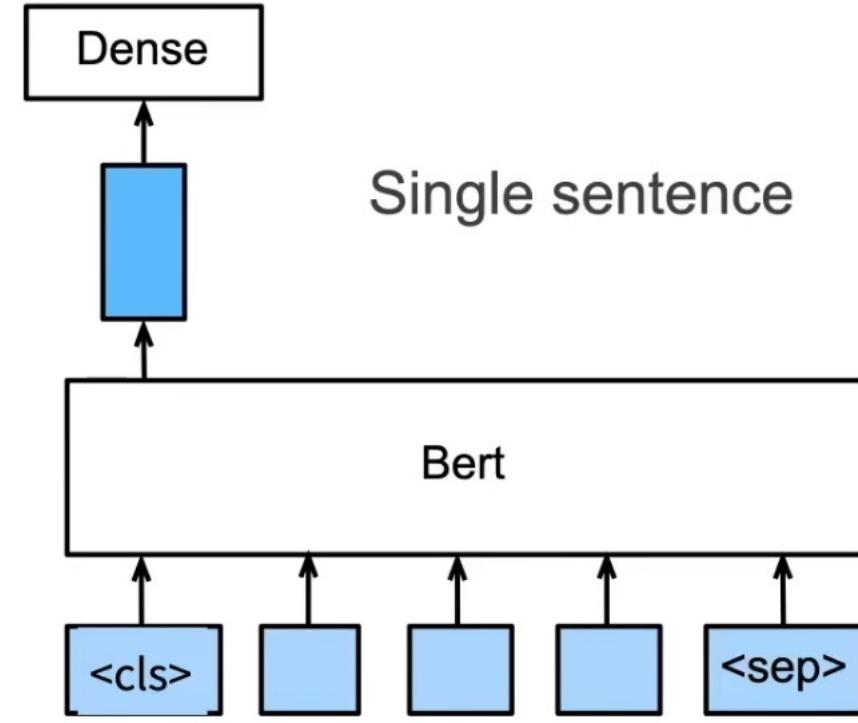
During the finetune stage, only a few downstream dataset is enough. This significantly widens the applications of natural language processing

Pretraining and finetuning become the standard practices in most AI fields (NLP, CV, etc.)

Finetune BERT to sentence classification

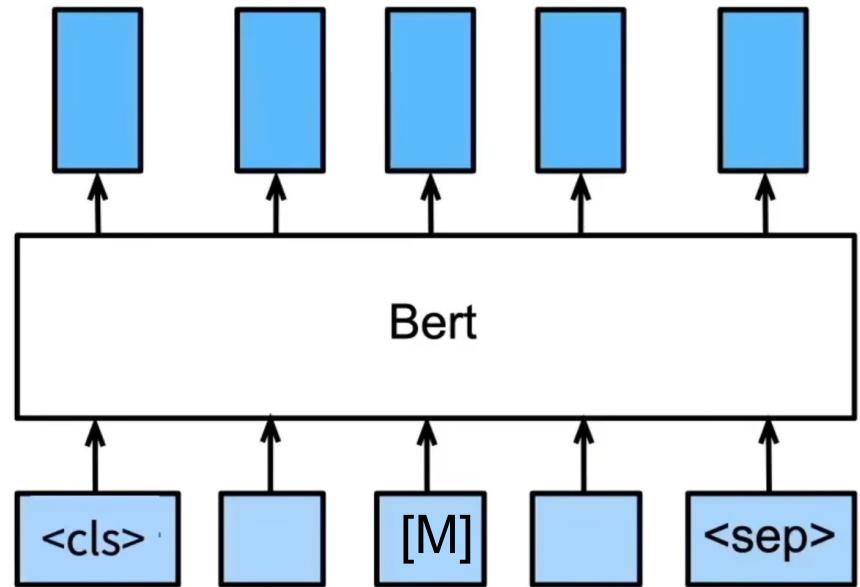


Pre-train

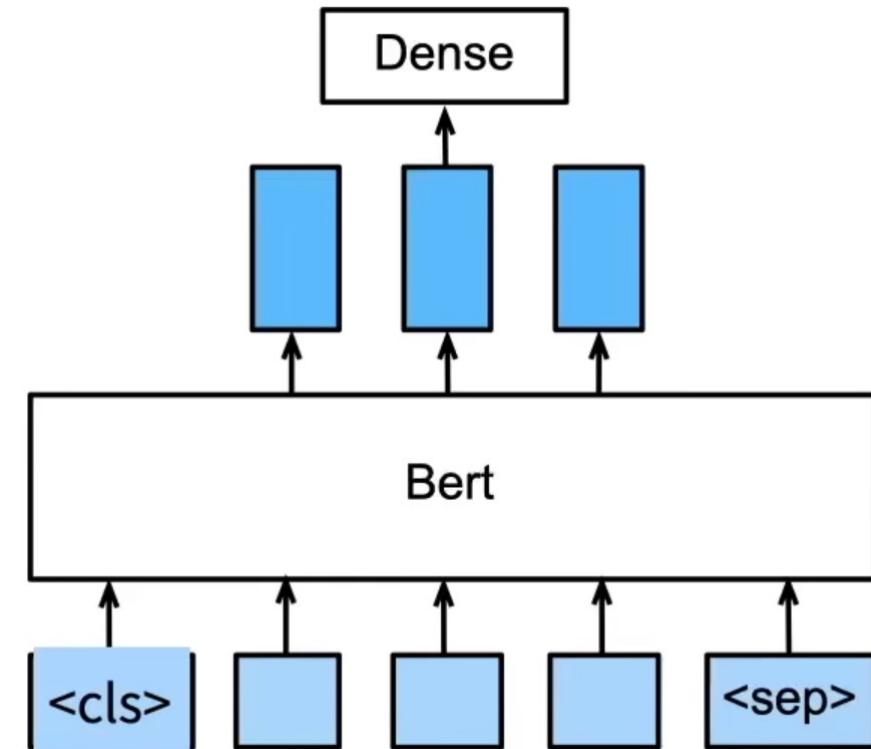


Finetune

Finetune BERT to entity recognition



Pre-train



Finetune

BERT finetune performance

BERT was massively popular and hugely versatile; finetuning BERT led to new state-of-the-art results on a broad range of tasks.

- **QQP:** Quora Question Pairs (detect paraphrase questions)
- **QNLI:** natural language inference over question answering data
- **SST-2:** sentiment analysis
- **CoLA:** corpus of linguistic acceptability (detect whether sentences are grammatical.)
- **STS-B:** semantic textual similarity
- **MRPC:** microsoft paraphrase corpus
- **RTE:** a small natural language inference corpus

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
BERT _{BASE}	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

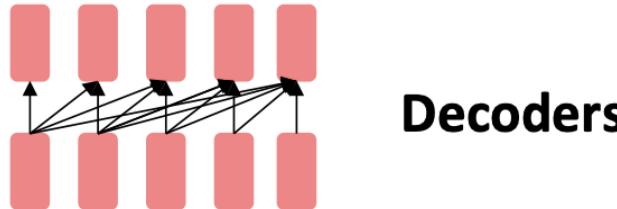
BERT limitations

Pretrain encoders such as BERT is great. It particularly suits in information extraction.

However, it is not good at text generation which cannot look into the future

If your task involves generating sequences, consider using a pretrained decoder; BERT and other pretrained encoders don't naturally lead to nice autoregressive (1-word-at-a-time) generation methods.

Pretrain decoder

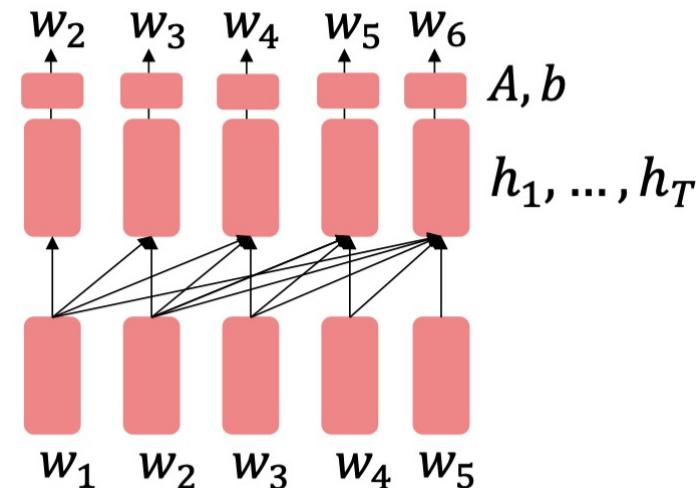


- Language models! What we've seen so far.
- Nice to generate from; can't condition on future words
- All the biggest pretrained models are Decoders.

Generative Pretrained Transformer (GPT) [Radford et al., 2018] was a big success in pretraining a decoder!

GPT, roughly speaking, pre-trained the decoder of the transformer.

GPT typically stands for “Generative Pretrained Transformer”



Improving Language Understanding by Generative Pre-Training

Alec Radford

OpenAI

alec@openai.com

Karthik Narasimhan

OpenAI

karthikn@openai.com

Tim Salimans

OpenAI

tim@openai.com

Ilya Sutskever

OpenAI

ilyasu@openai.com

GPT architecture

Let $U = (u_{-k}, \dots, u_{-1})$ be the context vector of tokens. The structure of GPT is as follows

$$h_0 = UW_e + W_p$$

$$h_l = \text{transformer_block}(h_{l-1}) \forall i \in [1, n]$$

$$P(u) = \text{softmax}(h_n W_e^T)$$

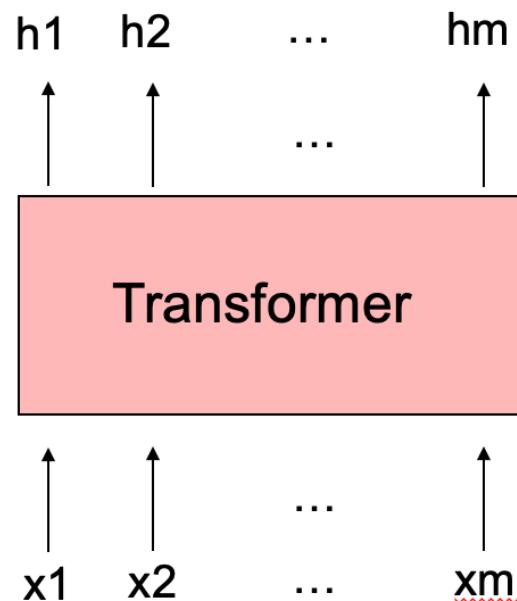
which is exactly the same as the transformer decoder

Language modelling, the target of GPT, is much harder than filling in the placeholder, the target of BERT. This explains that why BERT performs better than GPT.

But language modelling has much wider applications than filling in the placeholder

- Transformer decoder with 12 layers, 117M parameters.
- 768-dimensional hidden states, 3072-dimensional feed-forward hidden layers.
- Byte-pair encoding with 40,000 merges
- Trained on BooksCorpus: over 7000 unique books.
 - Contains long spans of contiguous text, for learning long-distance dependencies.

Given a sequence of downstream tokens $\{x_1, x_2, \dots, x_m\}$, and its label y



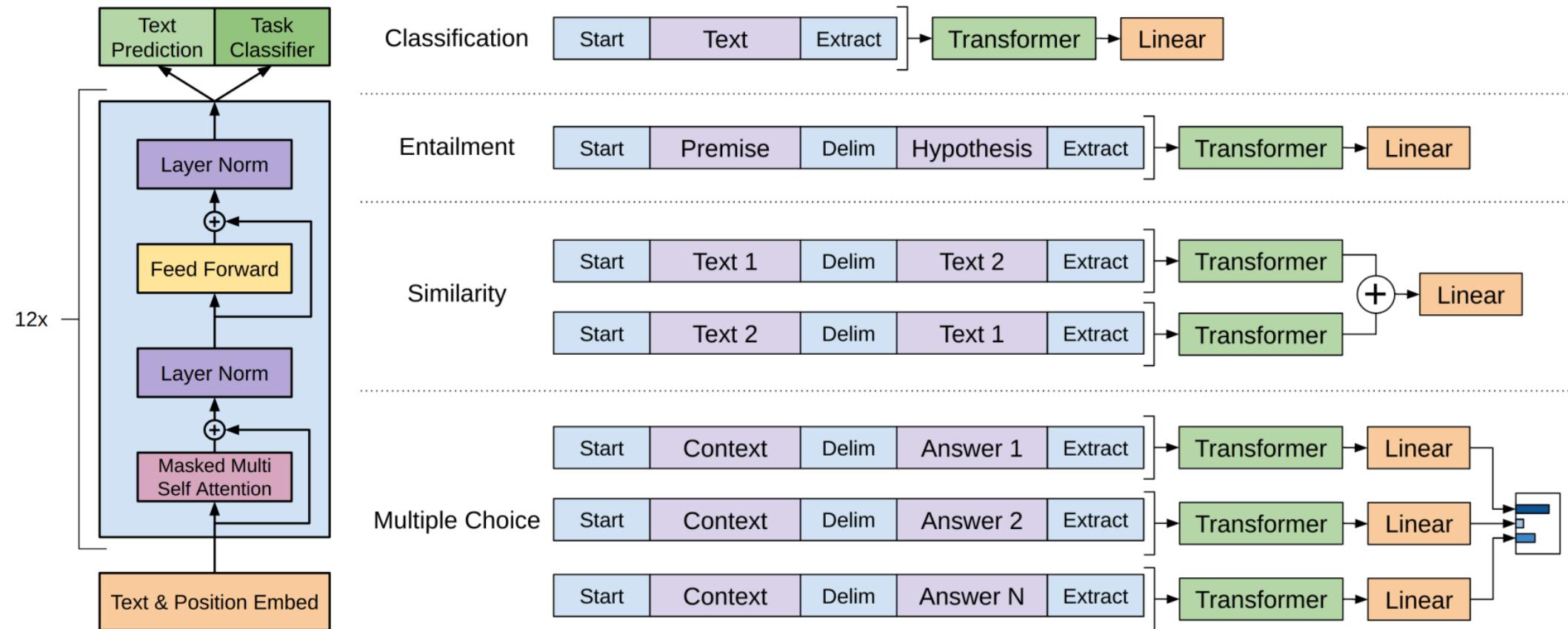
$$+ \quad \text{softmax}(h_\ell^m W_y) = L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$

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

$$P(y|x^1, \dots, x^m) = \text{softmax}(h_l^m W_y).$$

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \dots, x^m).$$

It is very natural to perform fine-tuning with GPT



GPT Performance

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	<u>89.3</u>	-	-	-
CAFE [58] (5x)	80.2	79.0	<u>89.3</u>	-	-	-
Stochastic Answer Network [35] (3x)	<u>80.6</u>	<u>80.1</u>	-	-	-	-
CAFE [58]	78.7	77.9	88.5	<u>83.3</u>		
GenSen [64]	71.4	71.3	-	-	<u>82.3</u>	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	<u>82.1</u>	61.7
Finetuned Transformer LM (ours)	82.1	81.4	89.9	88.3	88.1	56.0

GPT-1 is the first model that significantly improves performances over various tasks

However, BERT, with bidirectional information, outperforms GPT-1 by a large margin shortly

GPT is based on the decoder-only architecture, and BERT is based on the encoder-only architecture

BERT has shown much stronger performance than GPT

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
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
BERT _{BASE}	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

How to strike back? **Increasing the data and enlarging the models!**

Language Models are Unsupervised Multitask Learners

Alec Radford *¹ Jeffrey Wu *¹ Rewon Child¹ David Luan¹ Dario Amodei **¹ Ilya Sutskever **¹

GPT increases to 1.5B parameters

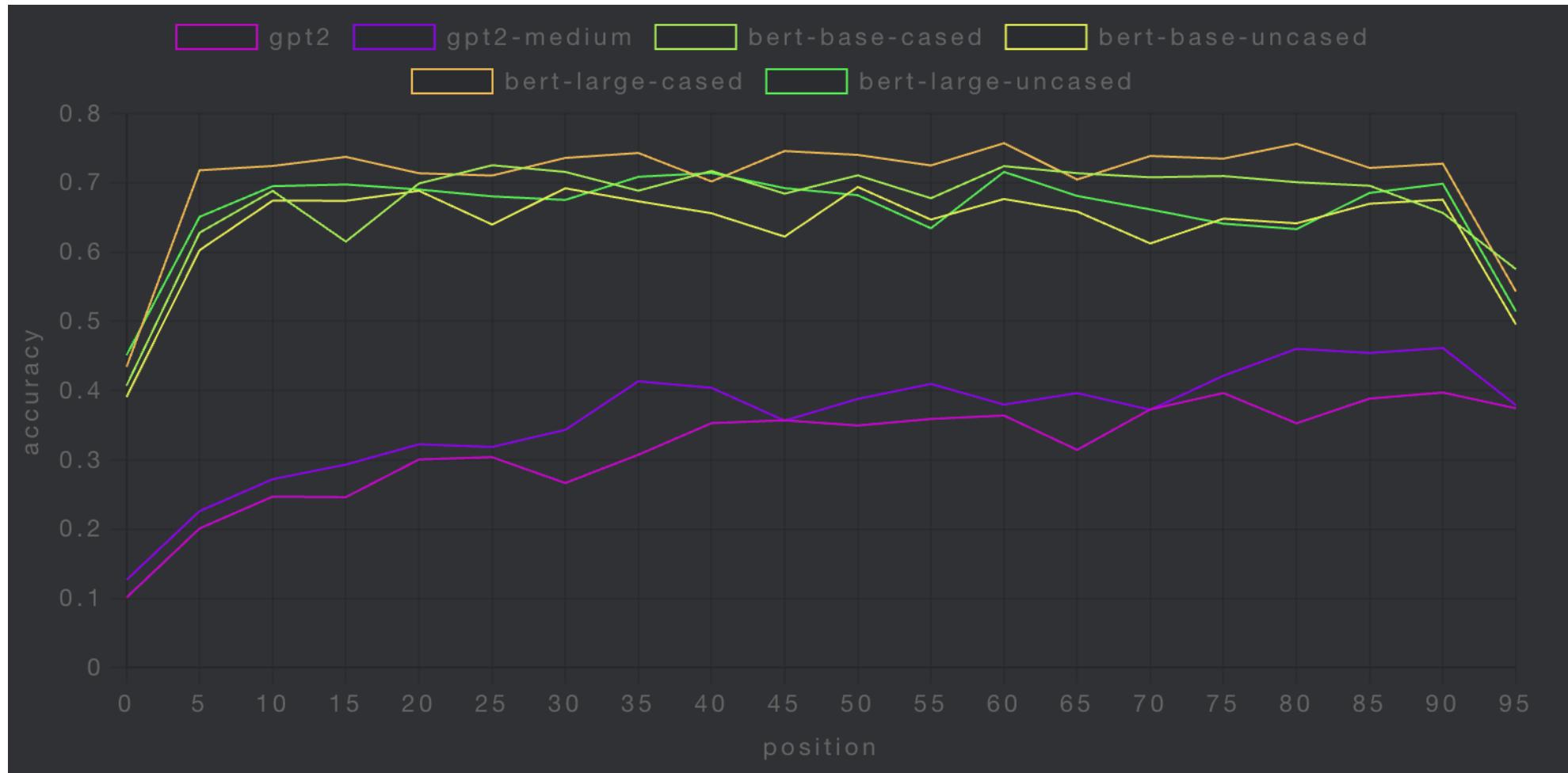
Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

BERT model size

Table 2. Architecture hyperparameters for the 4 model sizes.

However, GPT-2, even with much larger model, cannot compete with BERT and its variant

GPT-2 vs BERT



[Figure is from lukesalamone.github.io/posts/bert-vs-gpt2/]

However, GPT-2 is a zero-shot learner

No need of any downstream dataset. No need to fine-tune. **Prompt** is enough!



English reference

This re-release, titled The Next Day Extra, was presented in the form of three disks: the original album, unpublished studio sessions and remixes, plus a DVD containing the four clips that have already been unveiled.

(translate to French)



GPT-2 French translation

Les nouvelles re-releases, tout en premier disc, nécessaire de l'album, un studio session et remixes, plus une DVD de l'écran de quelques clips qui ont été déjà échappés.

In contrast, BERT and its variants do not show similar capabilities

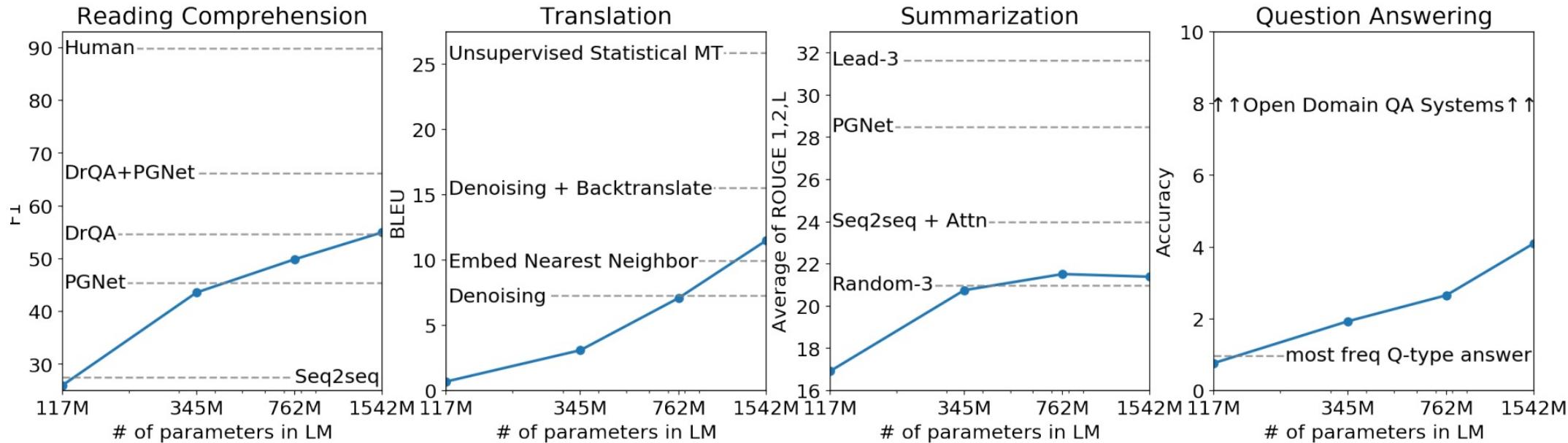
GPT-2 is a zero-shot learner

On some tasks, GPT-2 achieves SOTA results

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	21.8
117M	35.13	45.99	87.65	83.4	29.41	65.85	1.16	1.17	37.50	75.20
345M	15.60	55.48	92.35	87.1	22.76	47.33	1.01	1.06	26.37	55.72
762M	10.87	60.12	93.45	88.0	19.93	40.31	0.97	1.02	22.05	44.575
1542M	8.63	63.24	93.30	89.05	18.34	35.76	0.93	0.98	17.48	42.16

Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results.

GPT-2 is still far away from SOTA in some tasks



But it shows an important trend: **Larger Model, Better Performance!**

The zero-shot learning capability in GPT-2 model is quite novel!

In academia, the influence of one work can be determined by the formula

$$\text{Influence} = \text{Novelty} \times \text{Effectiveness}$$

GPT-2 is novel, but not very effective. GPT-3 comes up to show strong effectiveness

Language Models are Few-Shot Learners

Tom B. Brown*

Benjamin Mann*

Nick Ryder*

Melanie Subbiah*

Jared Kaplan[†]

Prafulla Dhariwal

Arvind Neelakantan

Pranav Shyam

Girish Sastry

Amanda Askell

Sandhini Agarwal

Ariel Herbert-Voss

Gretchen Krueger

Tom Henighan

Rewon Child

Aditya Ramesh

Daniel M. Ziegler

Jeffrey Wu

Clemens Winter

Christopher Hesse

Mark Chen

Eric Sigler

Mateusz Litwin

Scott Gray

Benjamin Chess

Jack Clark

Christopher Berner

Sam McCandlish

Alec Radford

Ilya Sutskever

Dario Amodei

OpenAI

GPT-3 gets extremely large. GPT-2 has 1.5B parameters, while GPT-3 has 175B parameters

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}

GPT-2 uses zero-shots, while GPT-3 uses few-shots

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

- 1 Translate English to French: ← *task description*
- 2 cheese => ← *prompt*

One-shot

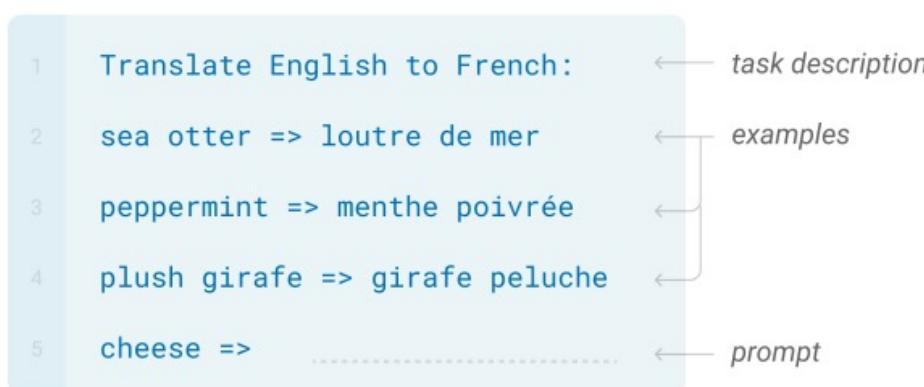
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

- 1 Translate English to French: ← *task description*
- 2 sea otter => loutre de mer ← *example*
- 3 cheese => ← *prompt*

GPT-2 uses zero-shots but GPT-3 uses few-shots

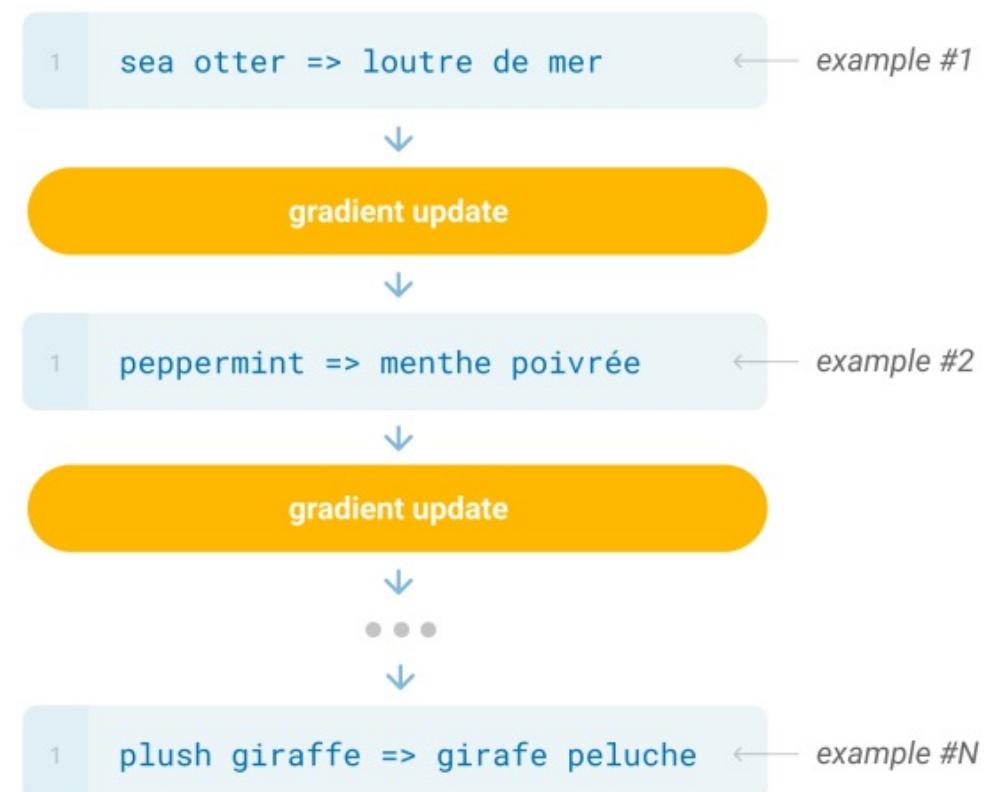
Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

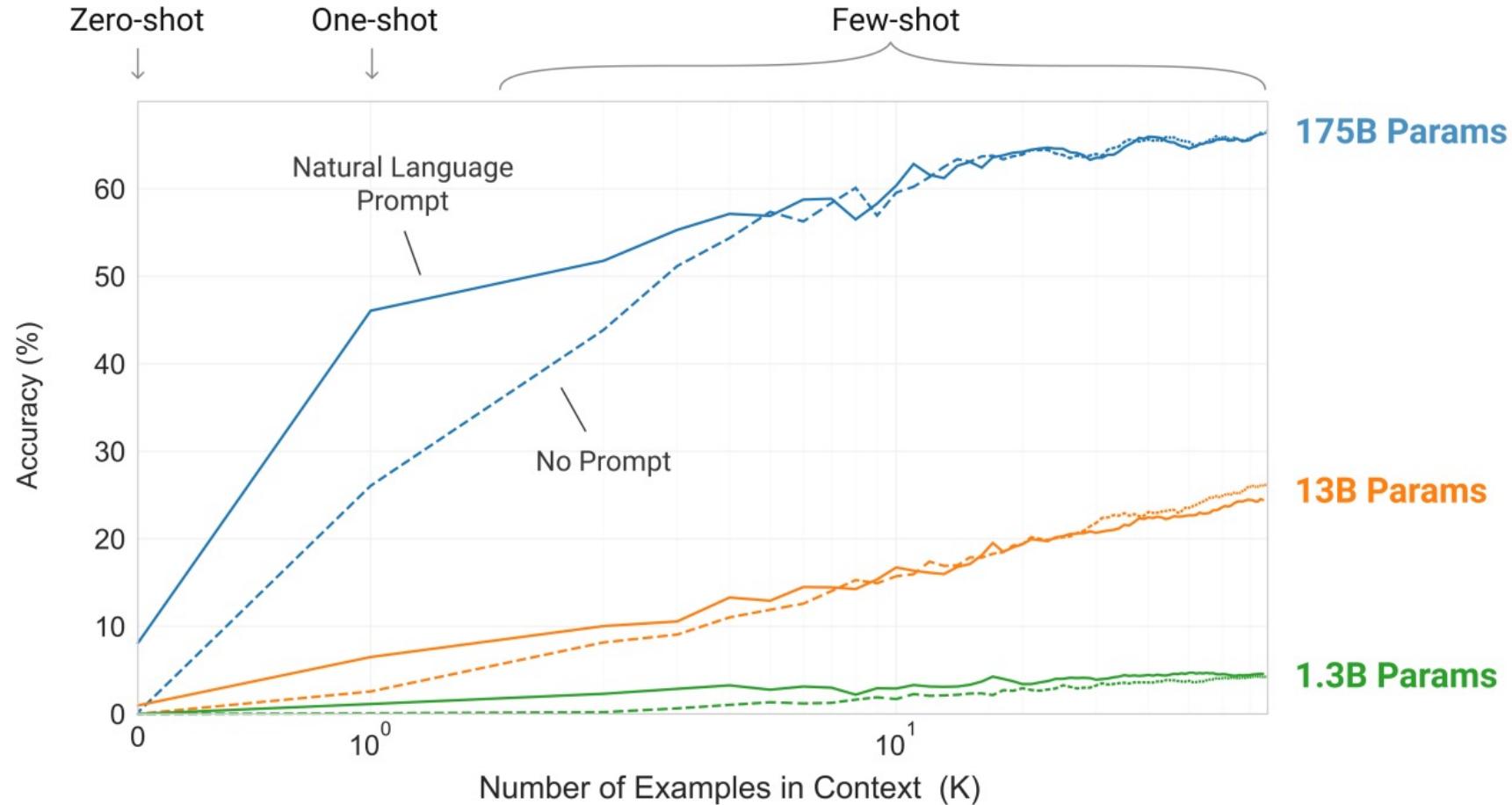


Fine-tuning

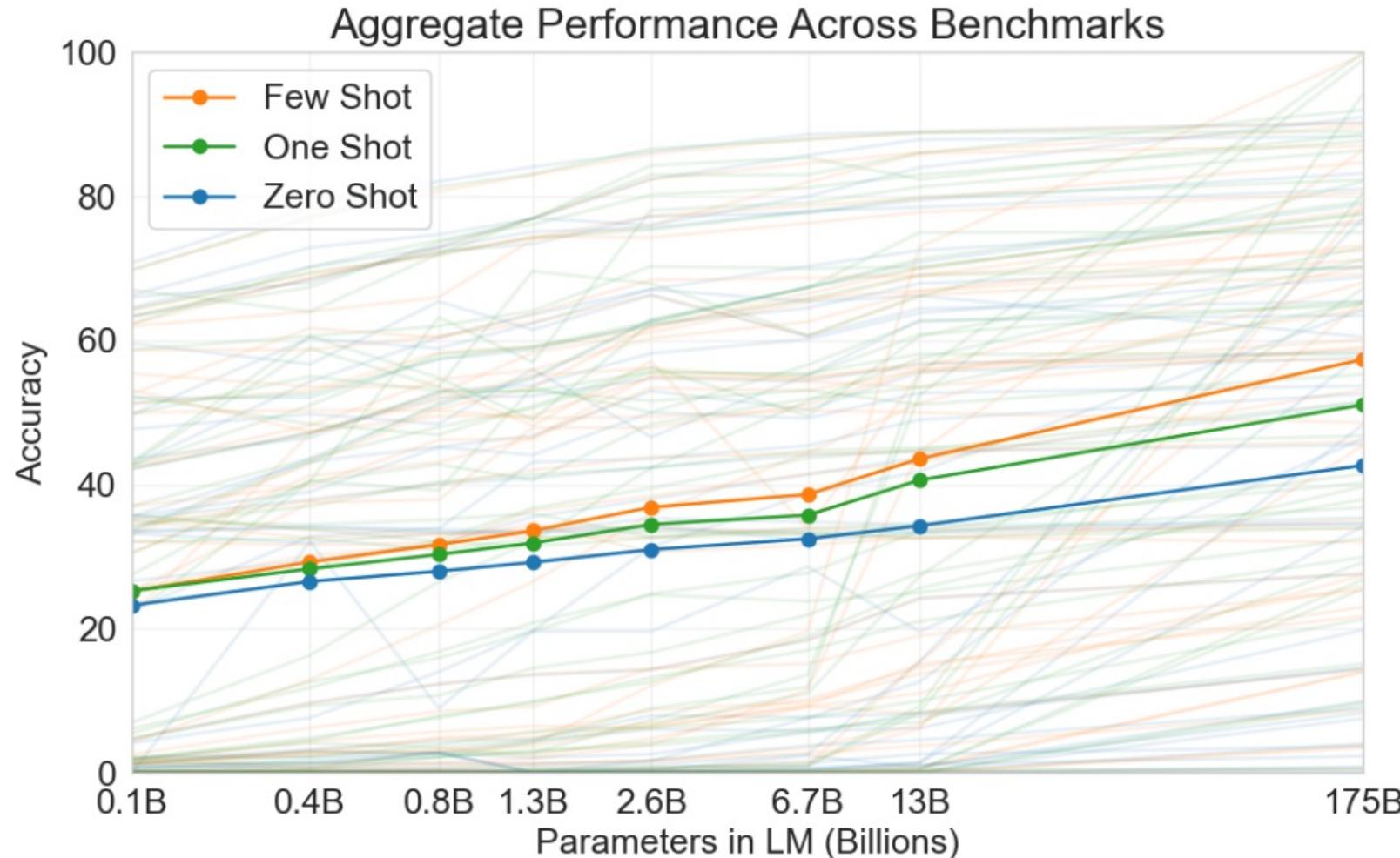
The model is trained via repeated gradient updates using a large corpus of example tasks.



Large model and few-shots are very useful

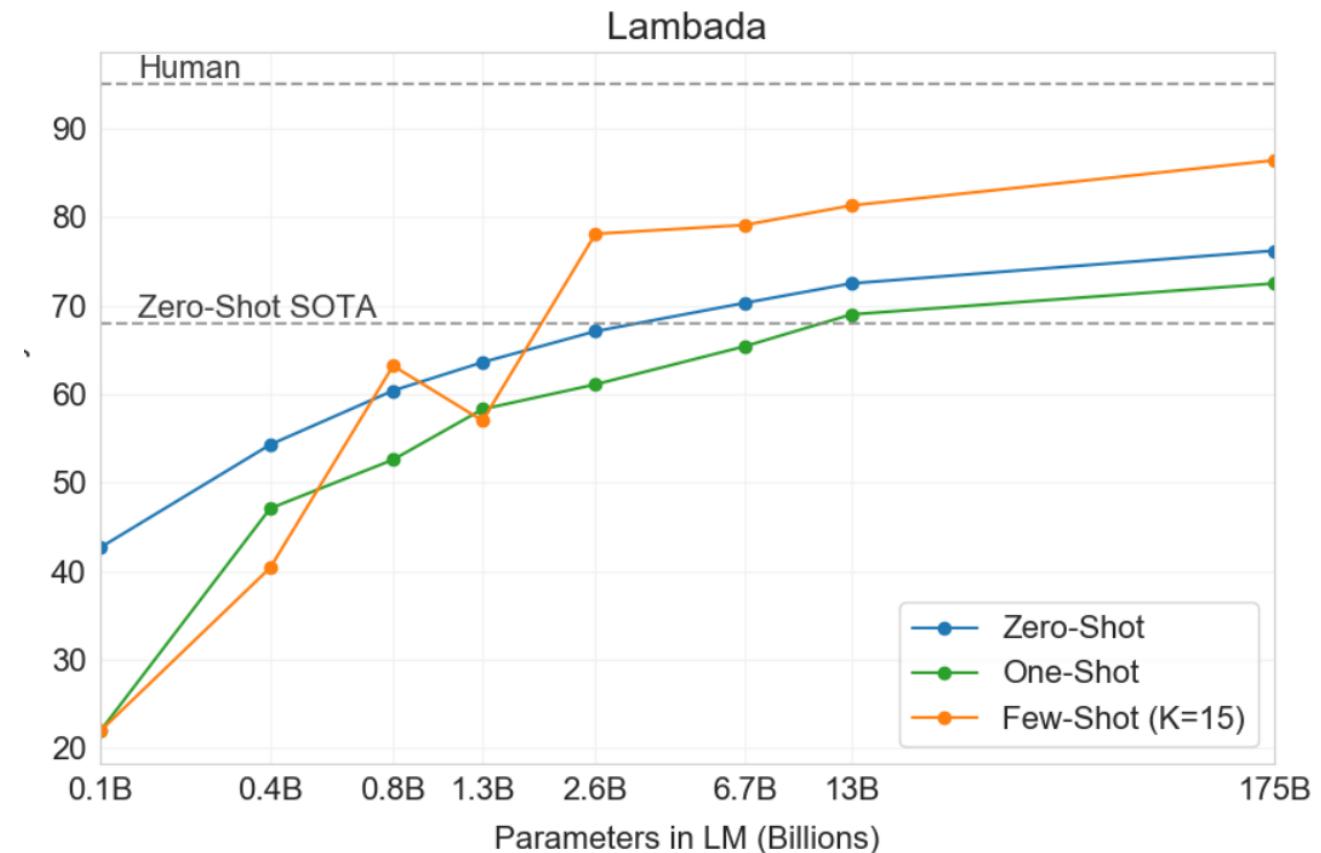


Large model and few-shots are very useful

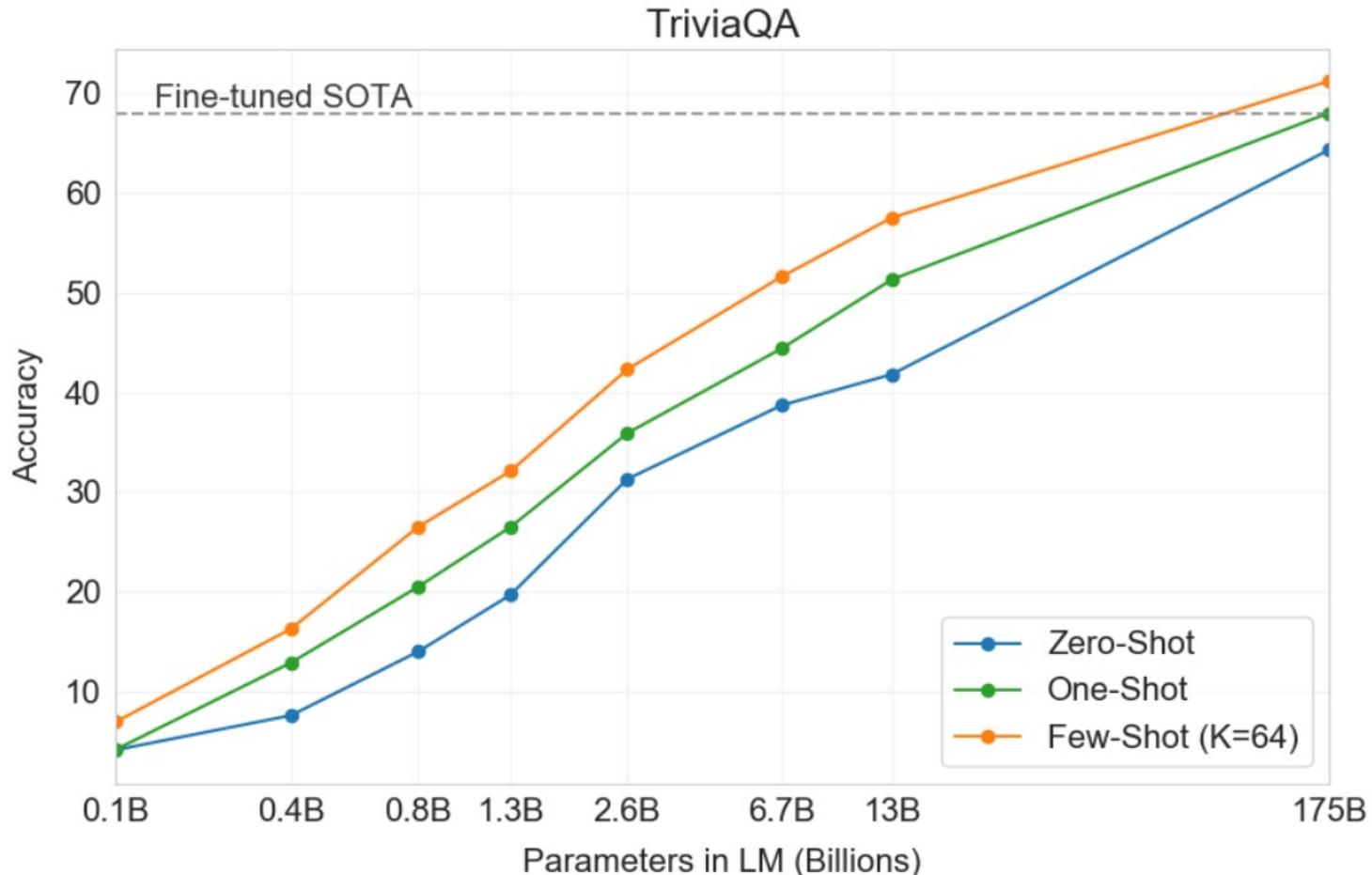


GPT-3 shows strong performance

Setting	LAMBADA (acc)	LAMBADA (ppl)
SOTA	68.0 ^a	8.63 ^b
GPT-3 Zero-Shot	76.2	3.00
GPT-3 One-Shot	72.5	3.35
GPT-3 Few-Shot	86.4	1.92



GPT-3 shows strong performance



Google v.s. OpenAI

By insisting on decoder-only architecture,
OpenAI achieves significant success in GPT

Who will be the final winner? We'll see

Any Chinese tech company? We'll see

