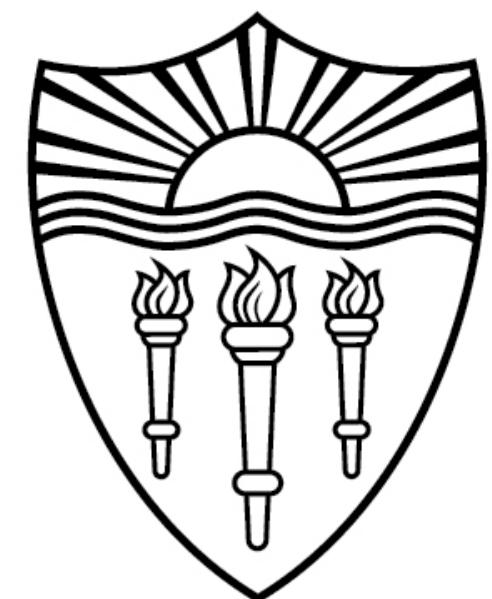


CSCI 544: Applied Natural Language Processing

Contextualized Embeddings & Large-scale Pre-training

Xuezhe Ma (Max)

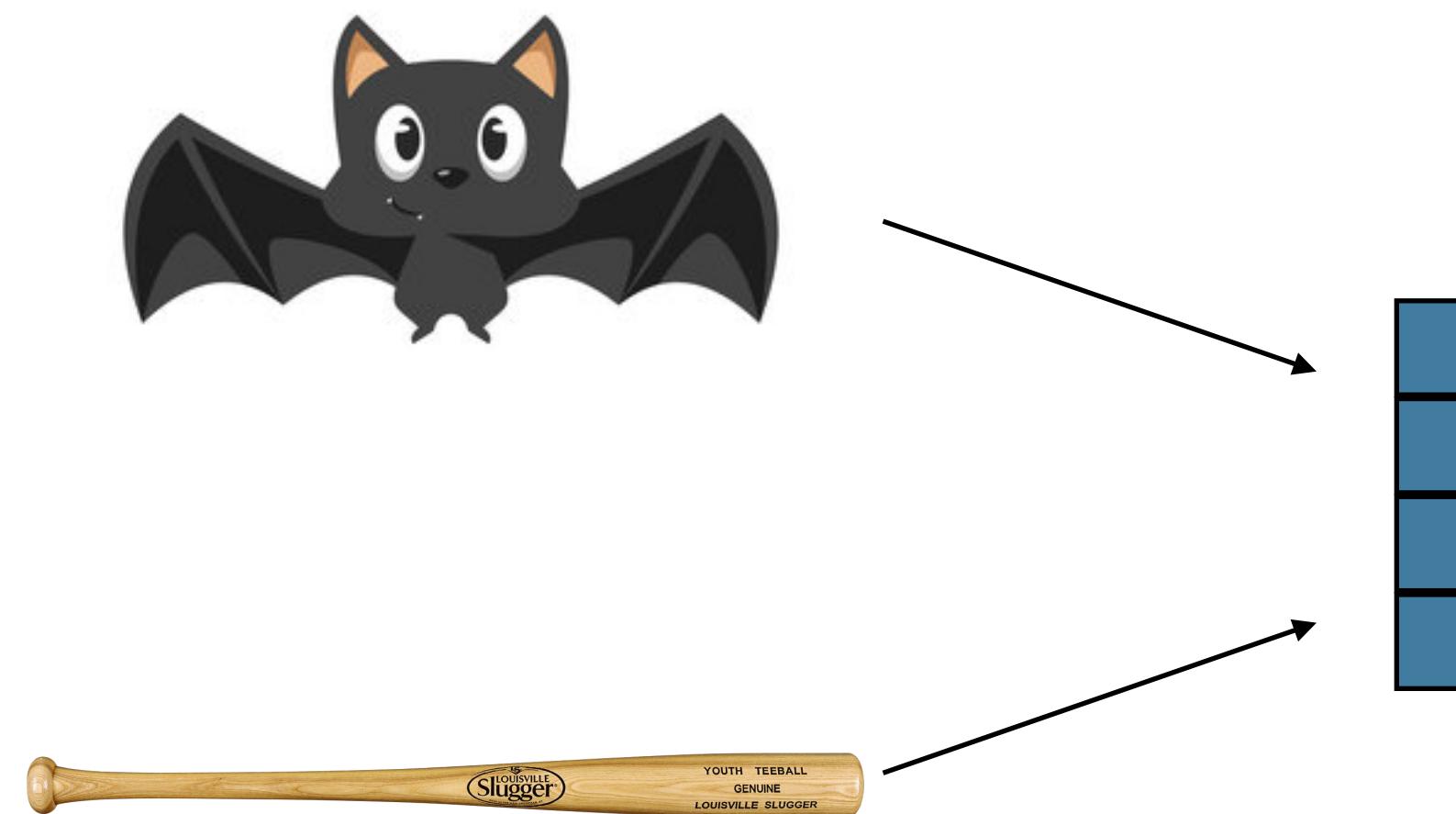


USC University of
Southern California

What's Wrong with Word Embeddings?

- One vector for each word type
- Complex characteristics of word use: syntax and semantics
- Polysemous words

Bat



What's Wrong with Word Embeddings?

- The semantic meaning of a word depends on this **context**

hit with bat

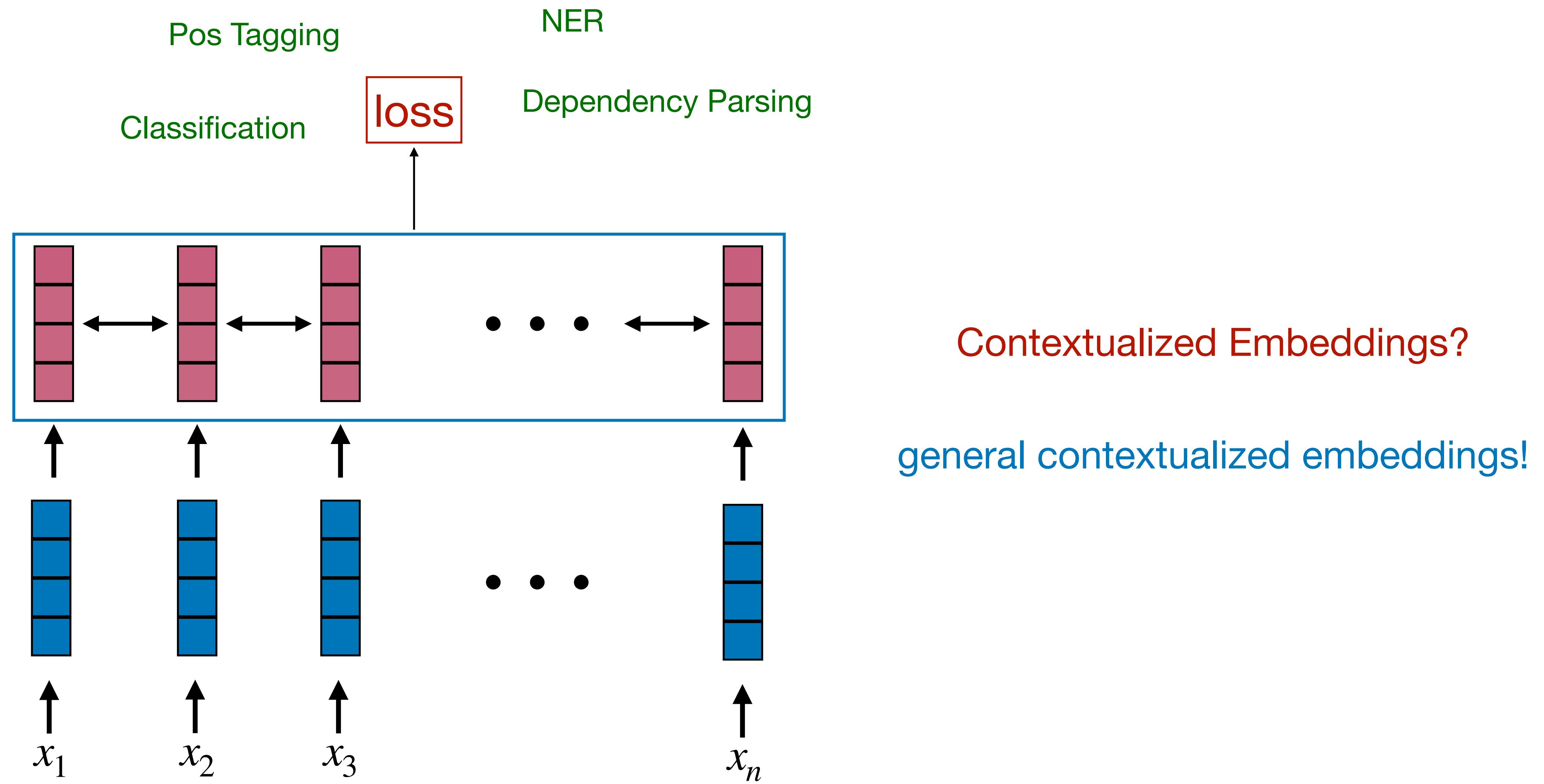


hit the bat



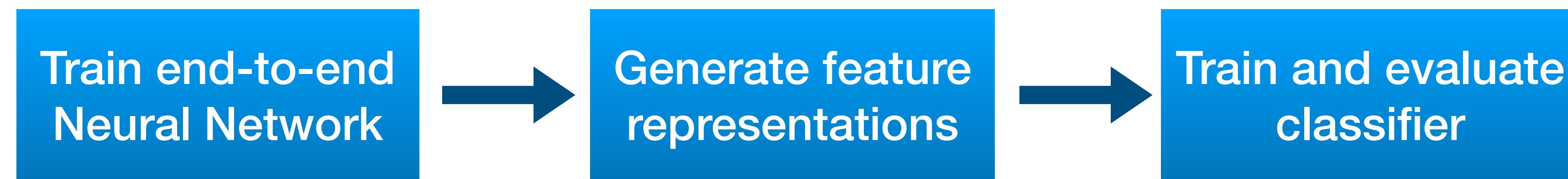
What's Wrong with Task-Specific Learning?

- We have contextualized models!

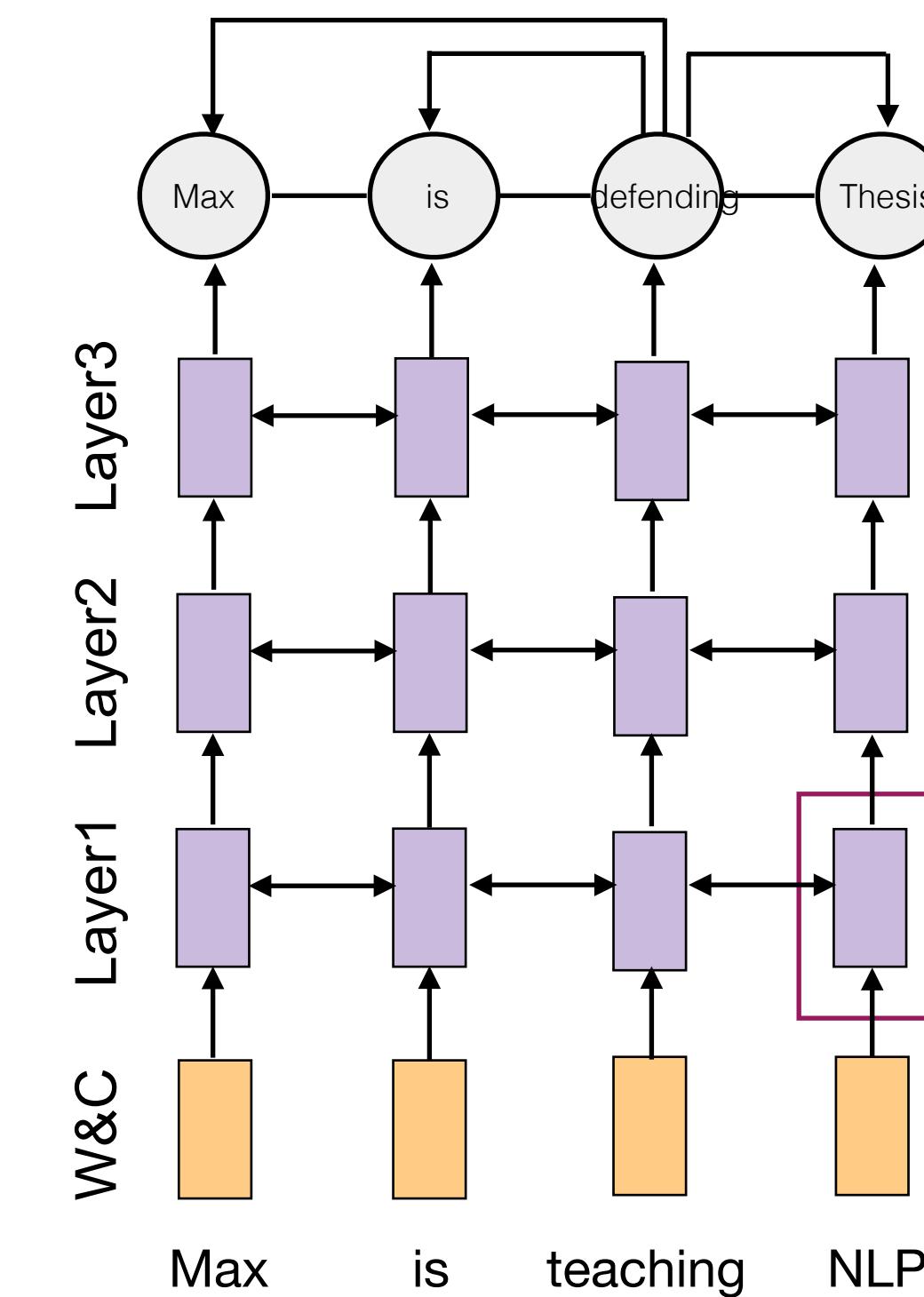
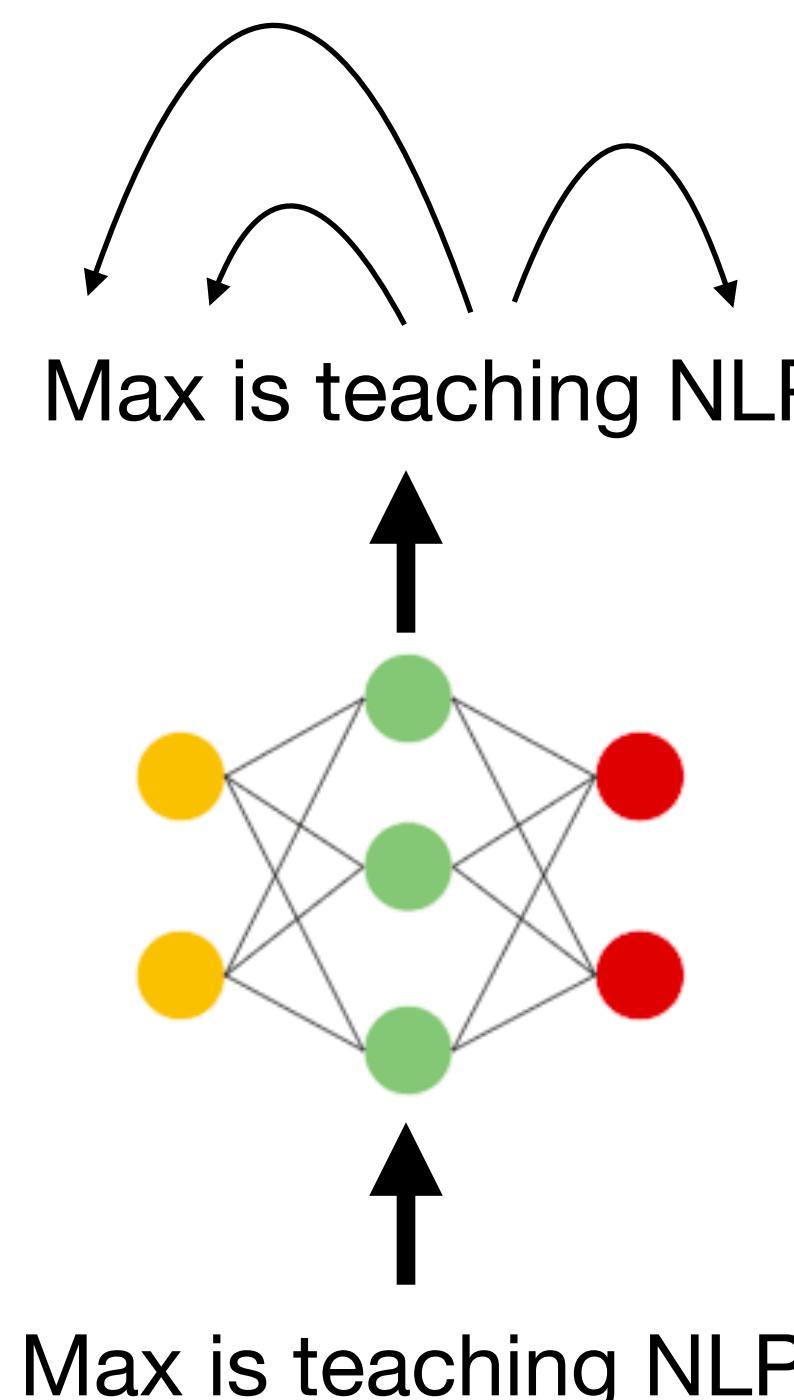


An Interesting Observation

- A Probing Experiment

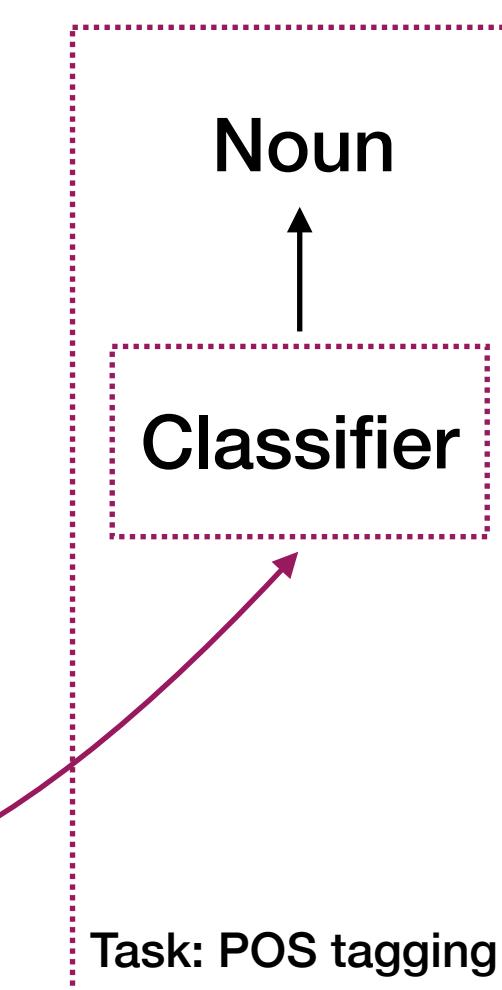


Main task: Dependency Parsing



Probing task:

- POS tagging



An Interesting Observation

POS Tagging	
BLSTM-CNN-CRF	97.6%
LSTM1 + SVM	97.7%
LSTM2 + SVM	97.8%

Neural Representations learned from a more challenging tasks can be applied to down-stream tasks!

Objective for Pre-training

- Easy to collect a large amount of data
 - Requiring no labeled data
- General and Semantic
 - Learning almost full knowledge about data, not only for specific tasks

Outline

- **Large-scale Pre-training**

- **Key Idea:** Training a large-scale model with a **general semantic** objective
- **Encoder:** Contextualized Embeddings
- **Encoder-Decoder:** Denoising Seq2seq Modeling
- **Decoder:** Neural Language Modeling

- **Using Pre-trained Models**

- Fully Fine-tuning
- Parameter-Efficient Fine-tuning
- Prompting/In-context Learning

Contextualized Embeddings: Pre-trained Encoders

Contextualized Embeddings: Pre-trained Encoders



- ELMo = Embeddings from Language Models
- BERT = Bidirectional Encoder Representations from Transformers

Deep contextualized word representations

[PDF] arxiv.org

[ME Peters, M Neumann, M Iyyer, M Gardner... - arXiv preprint arXiv ..., 2018 - arxiv.org](#)

We introduce a new type of deep contextualized word representation that models both (1) complex characteristics of word use (eg, syntax and semantics), and (2) how these uses vary across linguistic contexts (ie, to model polysemy). Our word vectors are learned functions of ...

☆ 99 Cited by 6367 Related articles All 20 versions ▾

Bert: Pre-training of deep bidirectional transformers for language understanding

[PDF] arxiv.org

[J Devlin, MW Chang, K Lee, K Toutanova - arXiv preprint arXiv ..., 2018 - arxiv.org](#)

We introduce a new **language** representation model called BERT, which stands for **Bidirectional Encoder Representations from Transformers**. Unlike recent **language** representation models, BERT is designed to pre-train **deep bidirectional** representations ...

☆ 99 Cited by 17552 Related articles All 26 versions ▾

Contextualized Word Embeddings

Source	Nearest Neighbors
GloVe play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM Chico Ruiz made a spectacular <u>play</u> on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent <u>play</u> .
biLM Olivia De Havilland signed to do a Broadway <u>play</u> for Garson {...}	{... } they were actors who had been handed fat roles in a successful <u>play</u> , and had talent enough to fill the roles competently , with nice understatement .

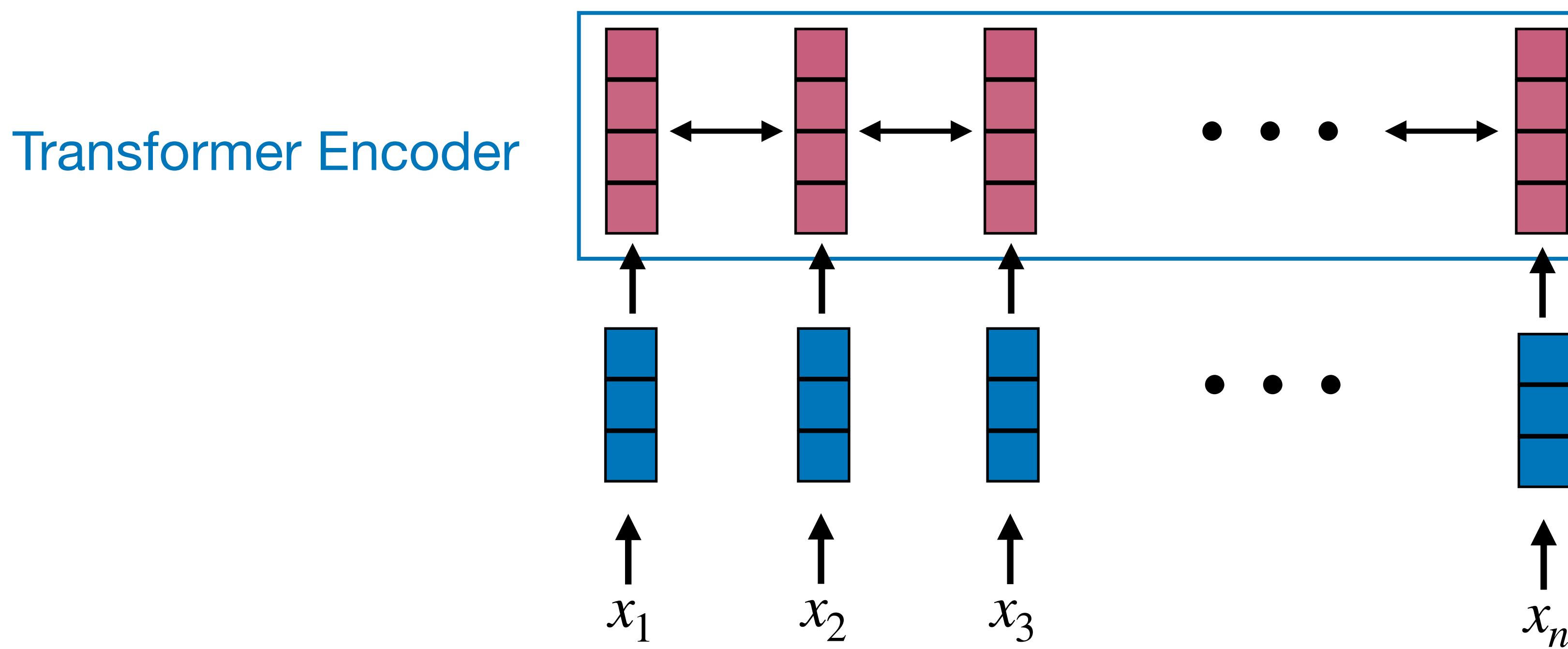
Deep contextualized word representations (Peters et al., 2018)

How can we get these contextualized embeddings?

- The key idea of BERT:
 - Objective: masked language modeling
 - Reconstructing masked words with surrounding tokens

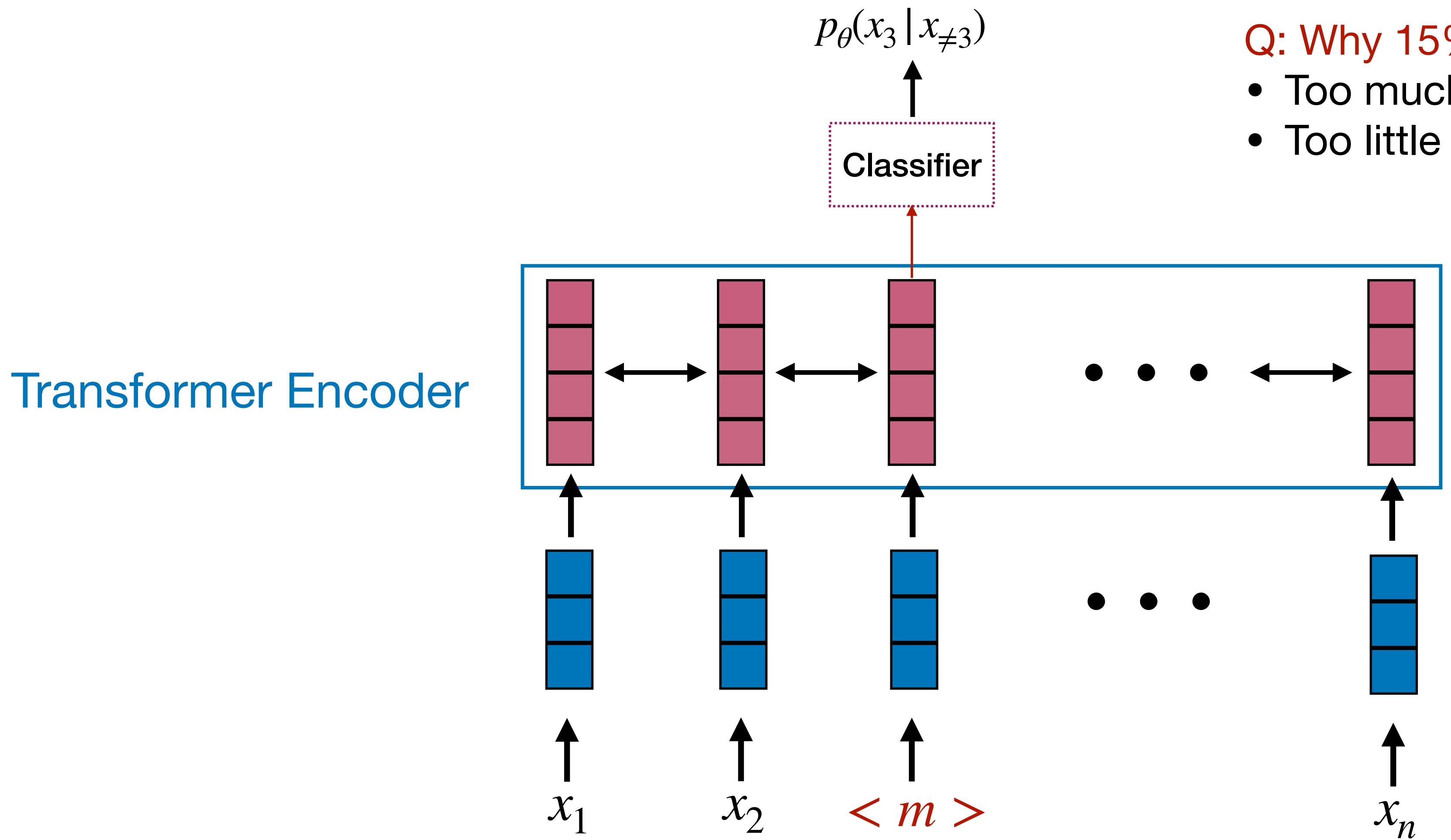
Masked Language Modeling

- Mask out 15% of the input words, and then predict the masked words



Masked Language Modeling

- Mask out 15% of the input words, and then predict the masked words



Q: Why 15%

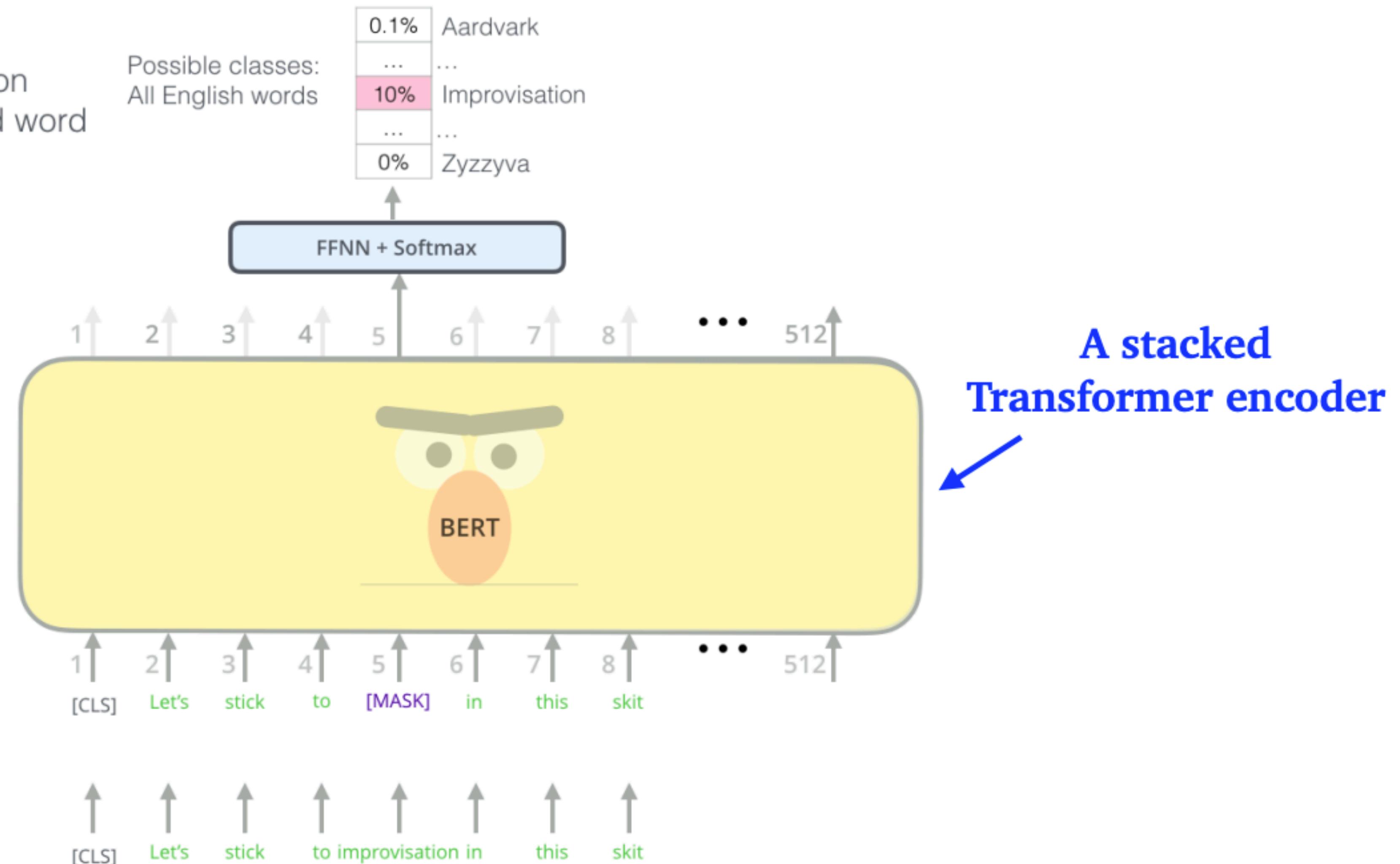
- Too much masking: not enough context
- Too little masking: too expensive to train

Masked Language Modeling (MLM)

Use the output of the masked word's position to predict the masked word

Randomly mask 15% of tokens

Input



Why Masked Language Modeling

- An **semantic-level task**
 - General contextualized embeddings
- Able to access both left and right context
 - Bidirectionality is **VERY** crucial in language classification tasks!

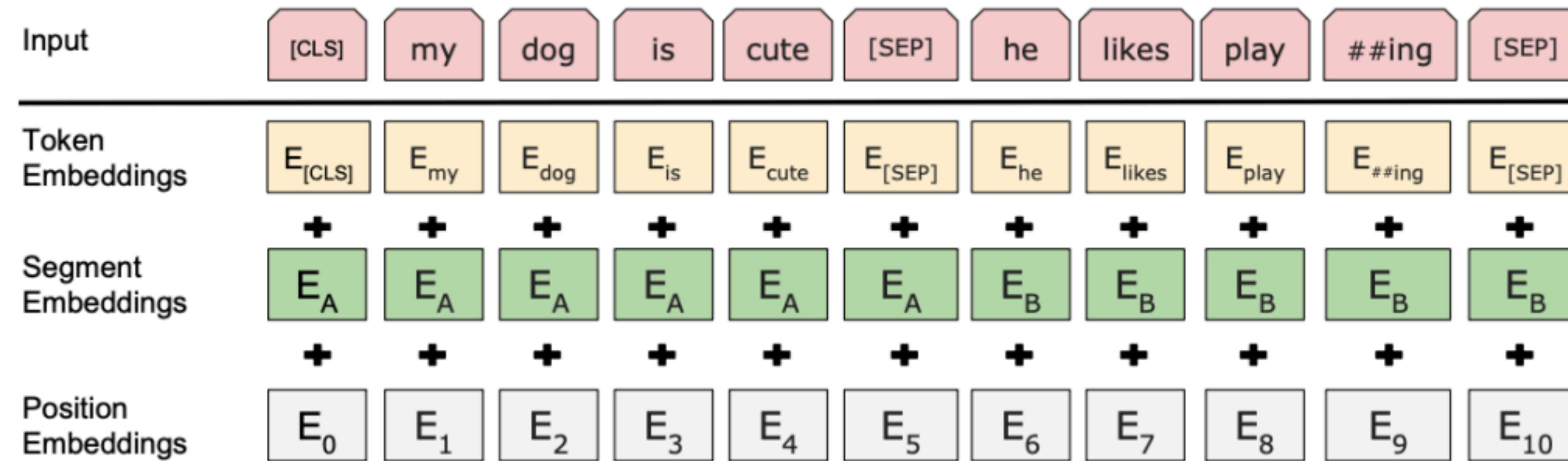
We will see some examples soon!

Training a BERT!

- **Training Data**
 - Wikipedia (2.5B words)
 - BooksCorpus (800M words)
- **Preprocessing**
 - BPE
 - Each segment: 512 BPE tokens
- **Transformer Encoder**
 - BERT-base: L=12, H=768, A=12, #parameters=110M
 - BERT-large: L=24, H=1024, A=16, #parameters=340M
- **Next sentence prediction (NSP)**
 - Later work shows that NSP hurts performance, so we omit it here

BERT: Pre-training

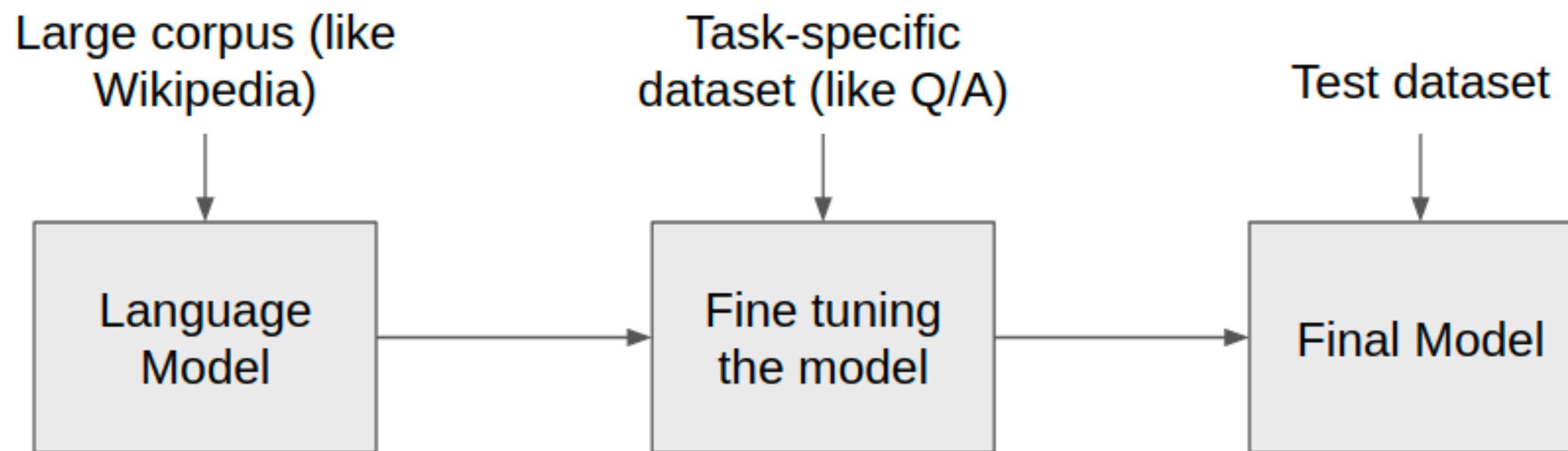
- Input representations



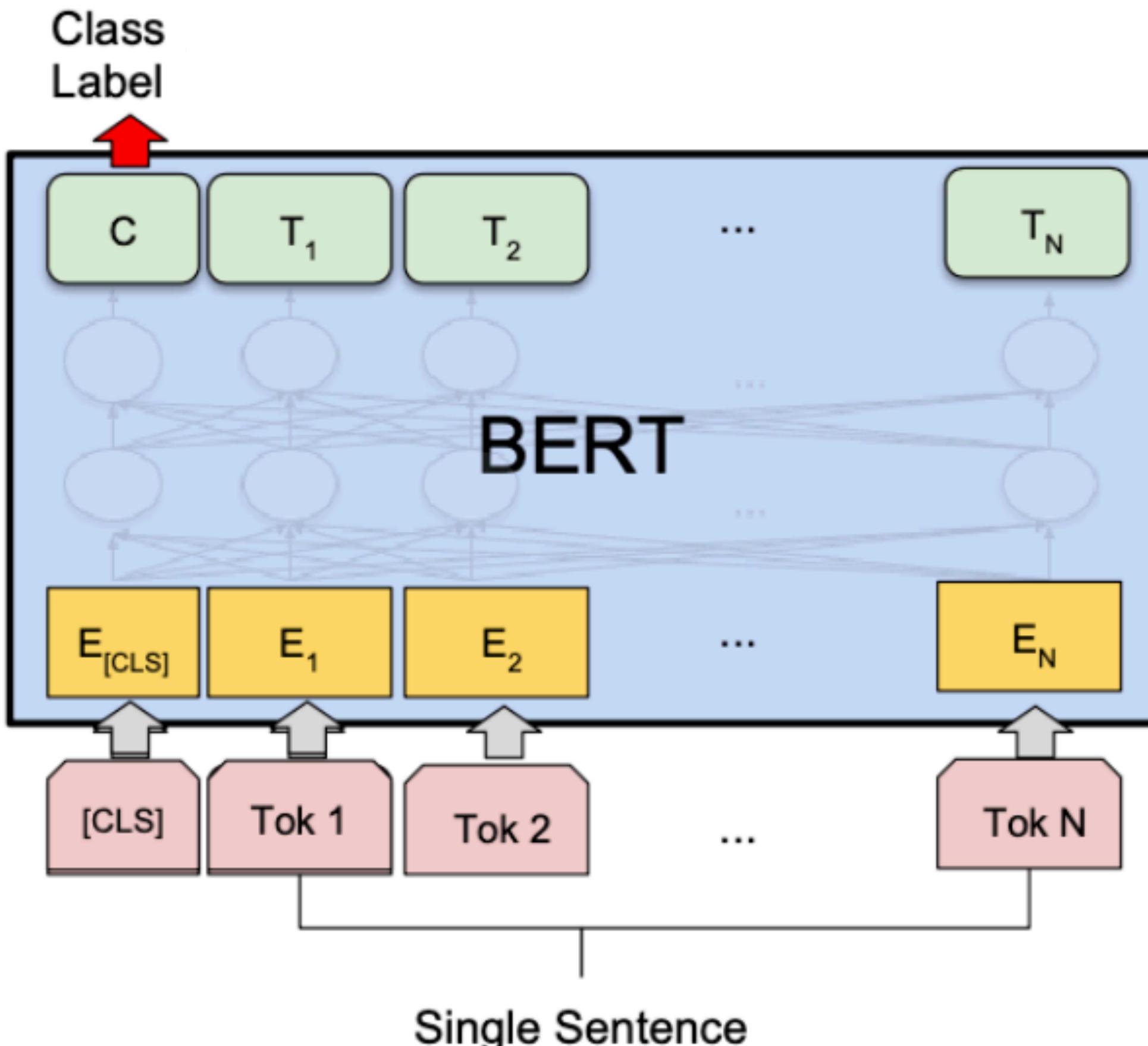
- Segment length: 512 BPE (=byte pair encoding) tokens
- Trained 40 epochs on Wikipedia (2.5B tokens) + BookCorpus (0.8B tokens)
- Released two model sizes: BERT_base, BERT_large

How to use BERT?

- Fine-tuning BERT for downstream tasks!



Fine-Tuning



Fine-tuning notes:

- Learning rate needs to be **small**
- Sufficient regularization to prevent overfitting

All the parameters will be learned together (original BERT parameters + new classifier parameters)

BERT: Results

BiLSTM: 63.9

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	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
BERT _{BASE}	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	91.1	94.9	60.5	86.5	89.3	70.1	81.9

BERT: Ablation Studies

Tasks	Dev Set					<small>Unidirectional LMs don't work!</small>
	MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)	
BERT _{BASE}	84.4	88.4	86.7	92.7	88.5	
No NSP	83.9	84.9	86.5	92.6	87.9	
LTR & No NSP	82.1	84.3	77.5	92.1	77.8	
+ BiLSTM	82.1	84.1	75.7	91.6	84.9	

Table 5: Ablation over the pre-training tasks using the BERT_{BASE} architecture. “No NSP” is trained without the next sentence prediction task. “LTR & No NSP” is trained as a left-to-right LM without the next sentence prediction, like OpenAI GPT. “+ BiLSTM” adds a randomly initialized BiLSTM on top of the “LTR + No NSP” model during fine-tuning.

#L	#H	#A	LM (ppl)	Dev Set Accuracy		
				MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

Table 6: Ablation over BERT model size. #L = the number of layers; #H = hidden size; #A = number of attention heads. “LM (ppl)” is the masked LM perplexity of held-out training data.

The bigger, the better..

BERT: Summary

- **Masked Language Modeling**

- Capturing both left and right contexts

- **Pre-training on large corpus**

- Segment-level inputs (multiple sentences)

- **Large models**

- BERT-base: 110M parameters
 - BERT-large: 340M parameters

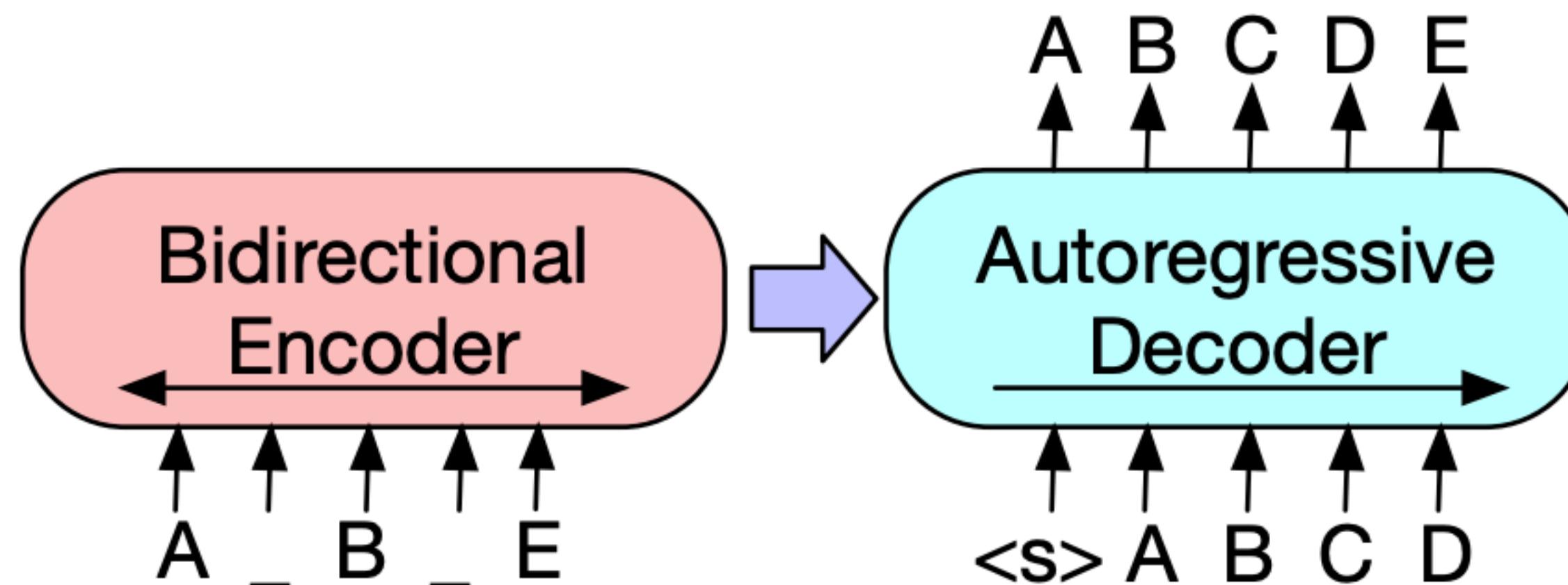
- **Transformer Encoder**

- **Unable to generate sentences!**

De-noising Seq2seq Modeling: Pre-trained Encoder-Decoders

BART: Denoising Seq2seq Pre-training

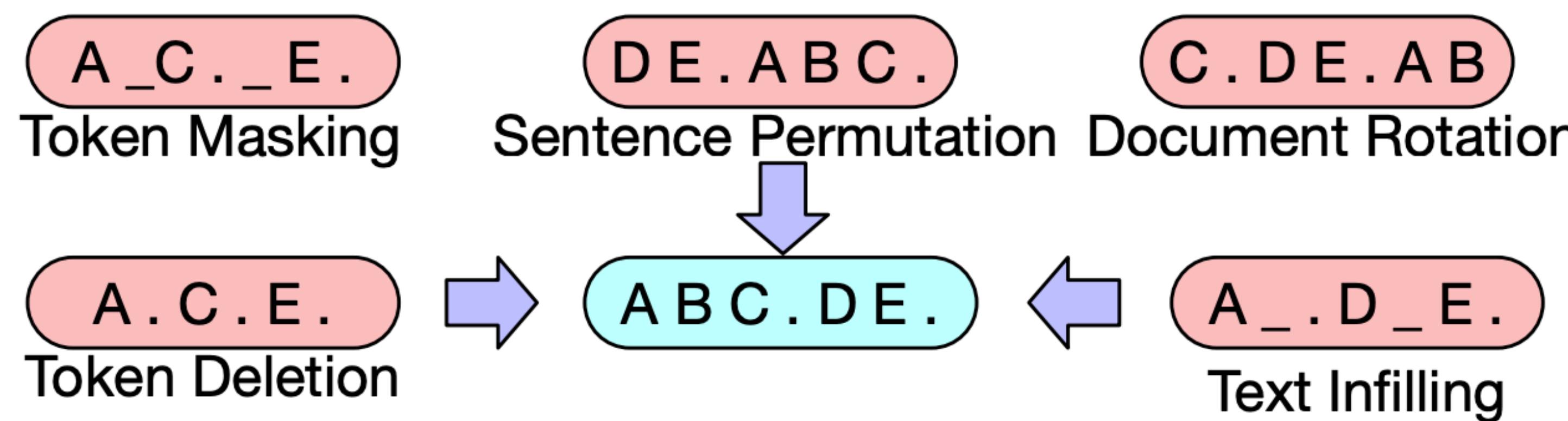
- Key idea: formulate pre-training as seq2seq generation
 - Encoder: input sentences with noisy transformations
 - Decoder: reconstruct the original input from the noisy one



Very useful for seq2seq tasks such as summarization!

BART: Denoising Seq2seq Pre-training

- **Token Masking:** similar to BERT
- **Token Deletion:** need to decide the positions of missing tokens
- **Text Infilling:** replacing a span of texts with a single <mask>
- **Sentence Permutation:** shuffling the order of sentences inside a document
- **Document Rotation:** Rotating a document/sentence such that the chosen token becomes the start token



BART on Summarization

	CNN/DailyMail			XSum		
	R1	R2	RL	R1	R2	RL
Lead-3	40.42	17.62	36.67	16.30	1.60	11.95
PTGEN (See et al., 2017)	36.44	15.66	33.42	29.70	9.21	23.24
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38	28.10	8.02	21.72
UniLM	43.33	20.21	40.51	-	-	-
BERTSUMABS (Liu & Lapata, 2019)	41.72	19.39	38.76	38.76	16.33	31.15
BERTSUMEXTABS (Liu & Lapata, 2019)	42.13	19.60	39.18	38.81	16.50	31.27
BART	44.16	21.28	40.90	45.14	22.27	37.25

Multilingual BART

- **Training BART on monolingual corpus over 25/100 languages:**
 - Shared BPE vocabulary
 - Shared model parameters

Multilingual BART

- mBART-25

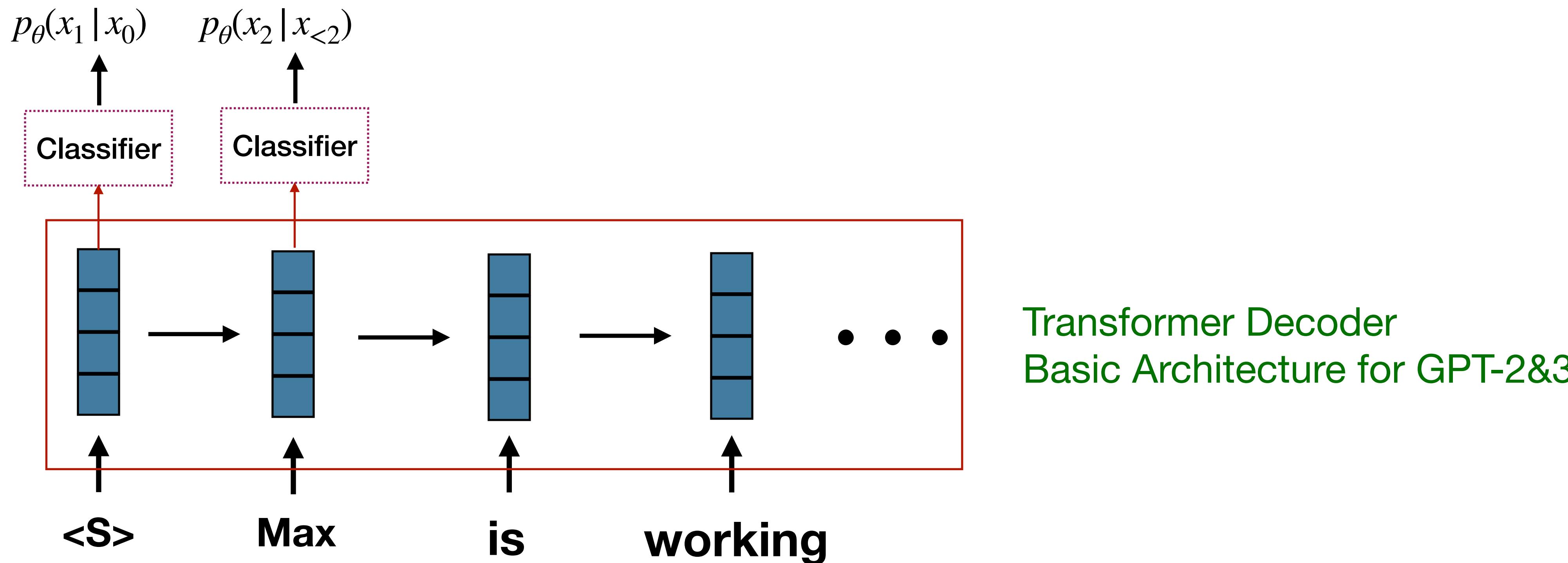
Languages	En-Gu		En-Kk		En-Vi		En-Tr		En-Ja		En-Ko	
Data Source	WMT19		WMT19		IWSLT15		WMT17		IWSLT17		IWSLT17	
Size	10K		91K		133K		207K		223K		230K	
Direction	←	→	←	→	←	→	←	→	←	→	←	→
Random	0.0	0.0	0.8	0.2	23.6	24.8	12.2	9.5	10.4	12.3	15.3	16.3
mBART25	0.3	0.1	7.4	2.5	36.1	35.4	22.5	17.8	19.1	19.4	24.6	22.6
Languages	En-Nl		En-Ar		En-It		En-My		En-Ne		En-Ro	
Data Source	IWSLT17		IWSLT17		IWSLT17		WAT19		FLoRes		WMT16	
Size	237K		250K		250K		259K		564K		608K	
Direction	←	→	←	→	←	→	←	→	←	→	←	→
Random	34.6	29.3	27.5	16.9	31.7	28.0	23.3	34.9	7.6	4.3	34.0	34.3
mBART25	43.3	34.8	37.6	21.6	39.8	34.0	28.3	36.9	14.5	7.4	37.8	37.7
Languages	En-Si		En-Hi		En-Et		En-Lt		En-Fi		En-Lv	
Data Source	FLoRes		ITTB		WMT18		WMT19		WMT17		WMT17	
Size	647K		1.56M		1.94M		2.11M		2.66M		4.50M	
Direction	←	→	←	→	←	→	←	→	←	→	←	→
Random	7.2	1.2	10.9	14.2	22.6	17.9	18.1	12.1	21.8	20.2	15.6	12.9
mBART25	13.7	3.3	23.5	20.8	27.8	21.4	22.4	15.3	28.5	22.4	19.3	15.9

Neural Language Modeling: Pre-trained Decoders

Auto-regressive Generative Models

- Auto-regressive Neural Language Models

$$p_{\theta}(X) = \prod_{t=1}^T p_{\theta}(x_t | x_{<t})$$



GPT-3

- **Training data**

- Common Crawl (410B tokens)
- WebText2 (19B tokens)
- Books1 & Books2 (12B + 55B tokens)
- Wikipedia (3B tokens)

- **Transformer Encoder**

- Medium: 350M
- Large: 760M
- X-Large: 1.3B

Q: How to use GPT-3?

- Fine-tuning is too expensive
- Prompting!

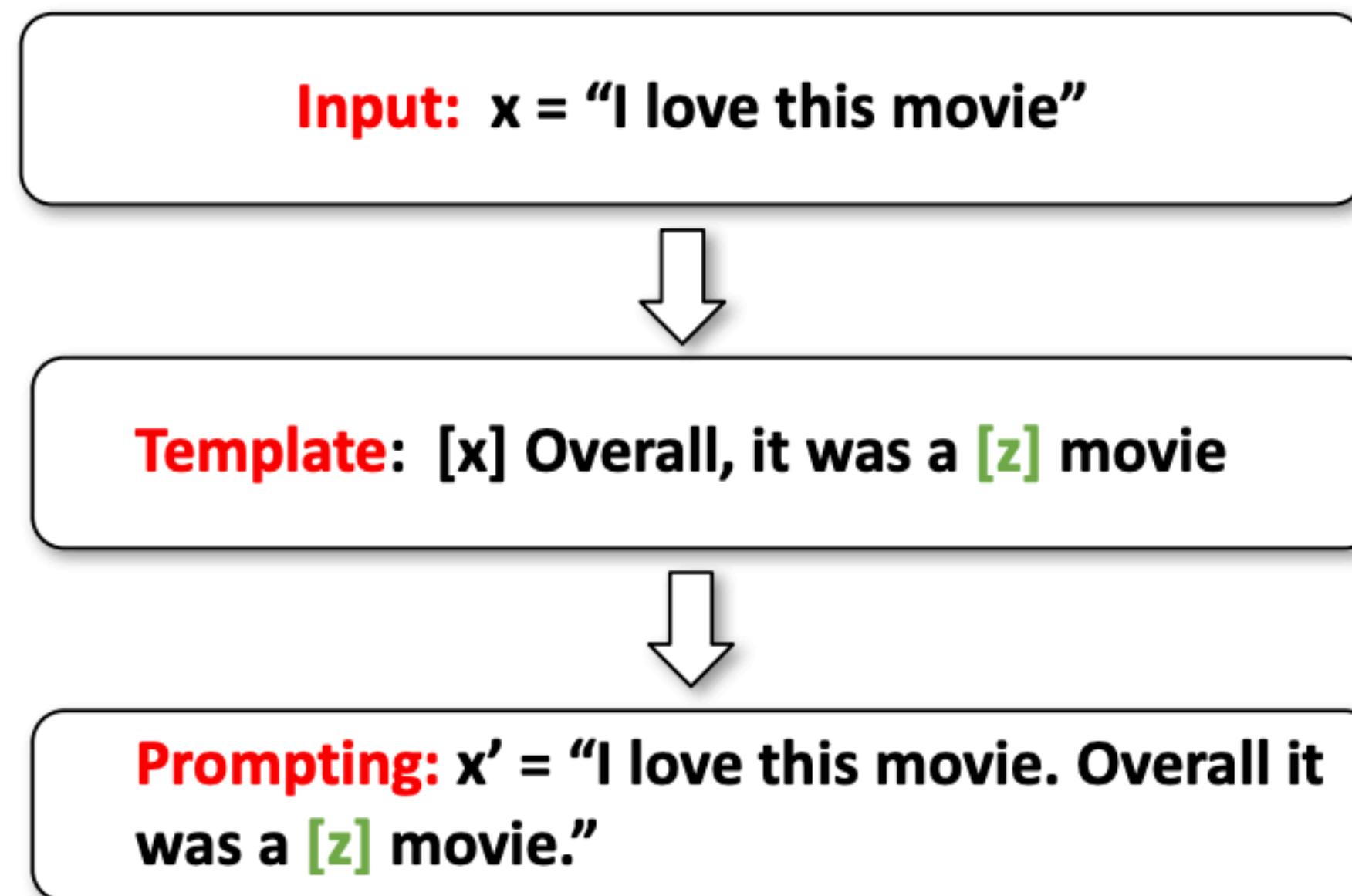
we will see some examples soon!

Prompting

- Prompt Addition
- Answer Prediction
- Answer-Label Mapping

Prompt Addition

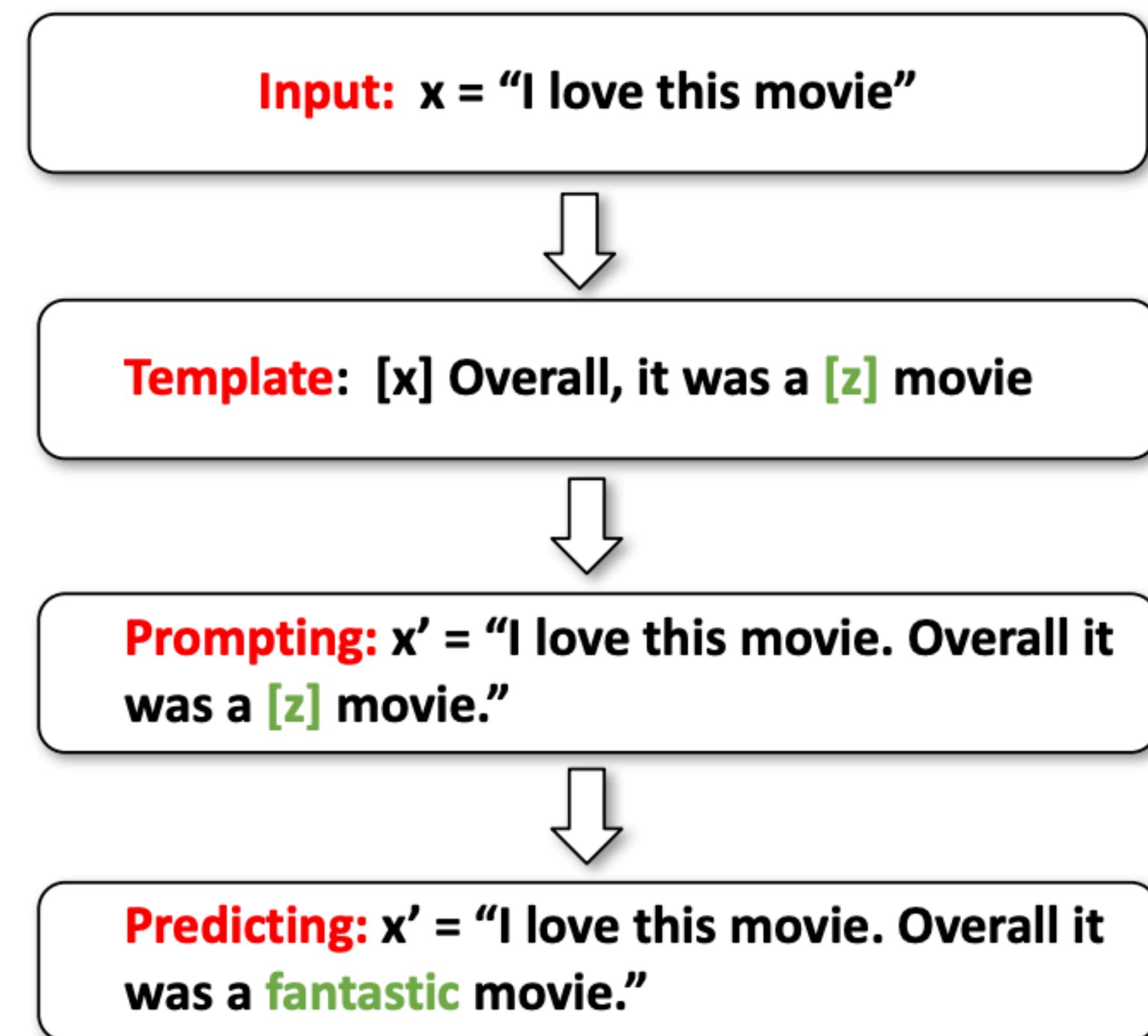
- Given input x , we create a prompt with two steps:
 - Define a template with two slots, one for input $[x]$, and one for the answer $[z]$
 - Fill in the input with slot $[x]$



Example: sentiment classification

Answer Prediction

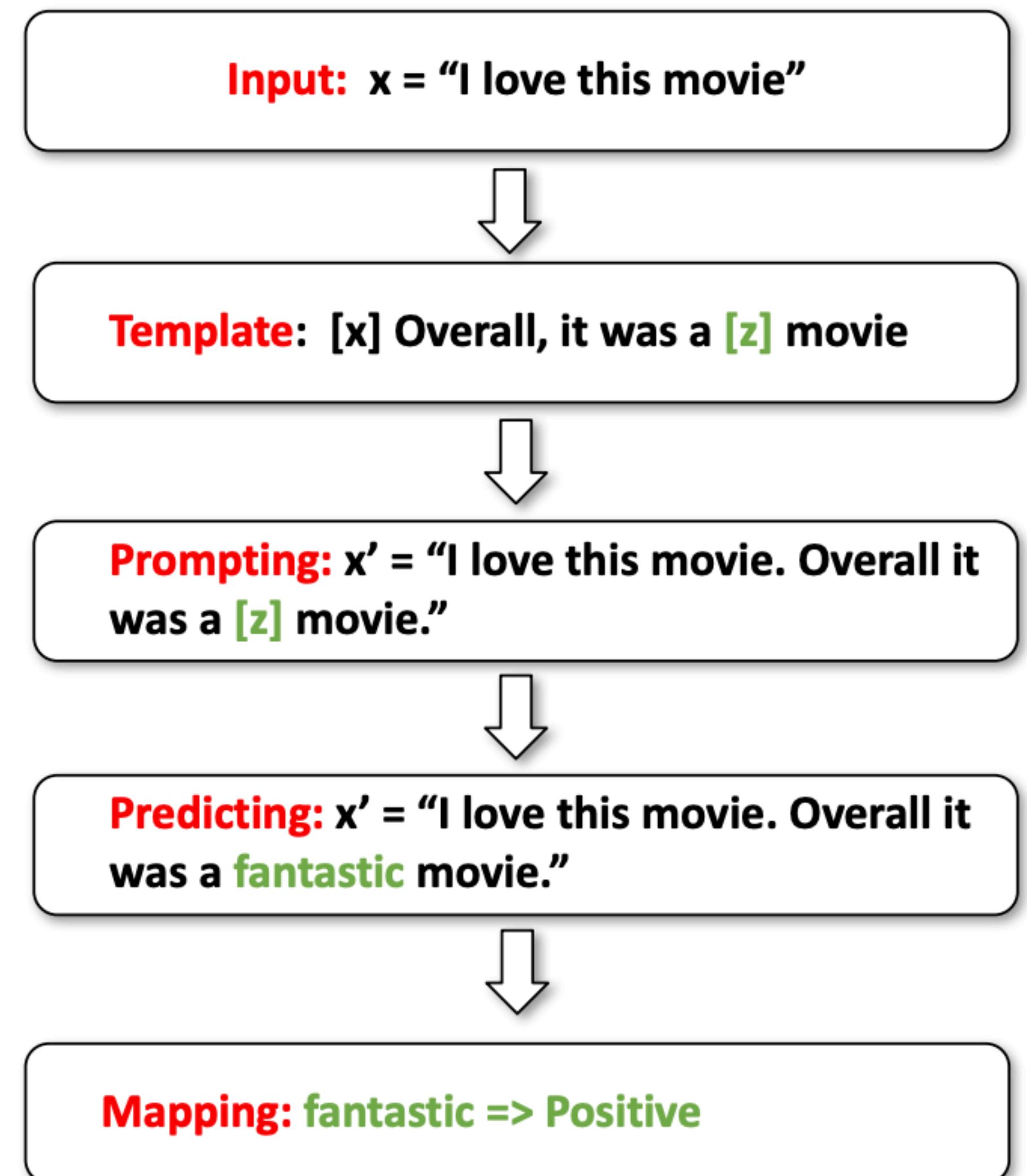
- Given a prompt, predict the answer [z]



Example: sentiment classification

Answer-Labeling Mapping

- Given an answer, map it into a class label



Example: sentiment classification

Types of Prompts

- **Cloze Prompt:**

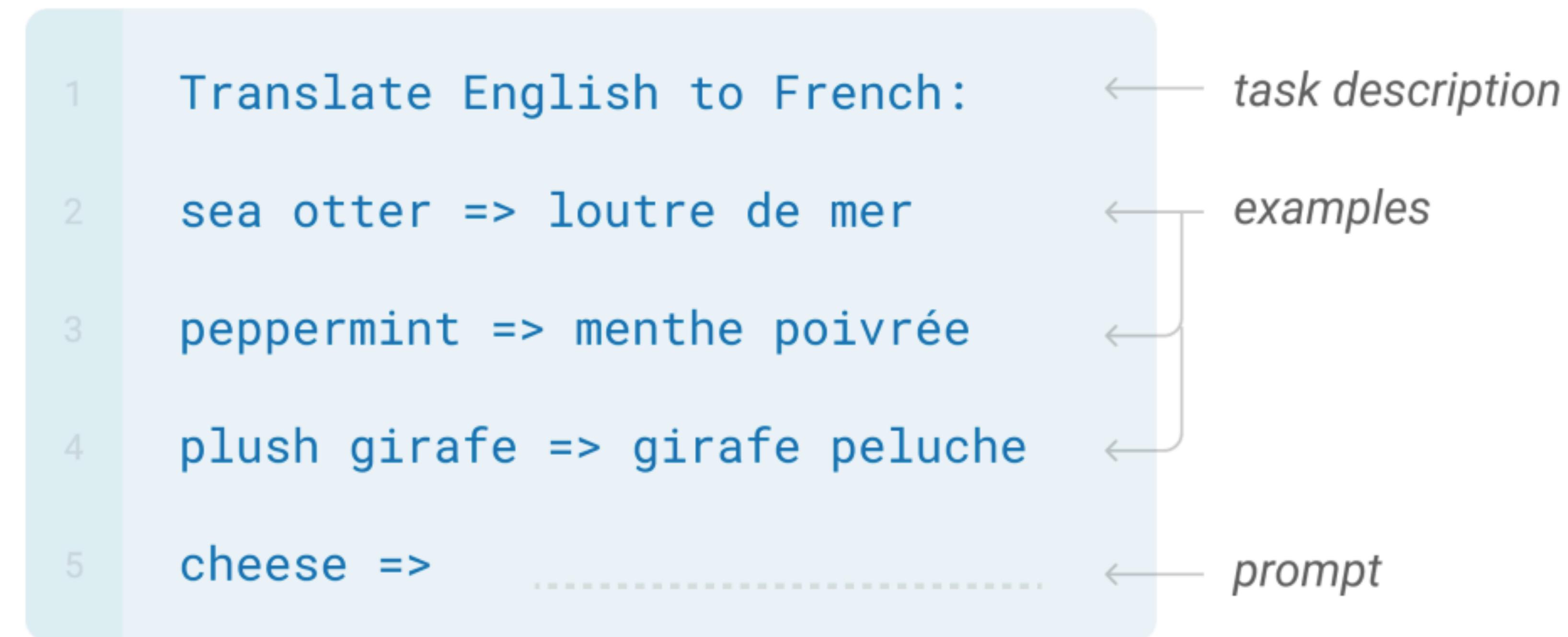
- I love this Movie. Overall, it was a [z] movie
- Masked Language Modeling (BERT)

- **Prefix Prompt**

- I love this Movie. Overall, this movie is [z]
- Auto-regressive Language Modeling (GPT-3)

Prompting in Few-shot Learning

- Suppose we have a few (less than 20) examples of the task



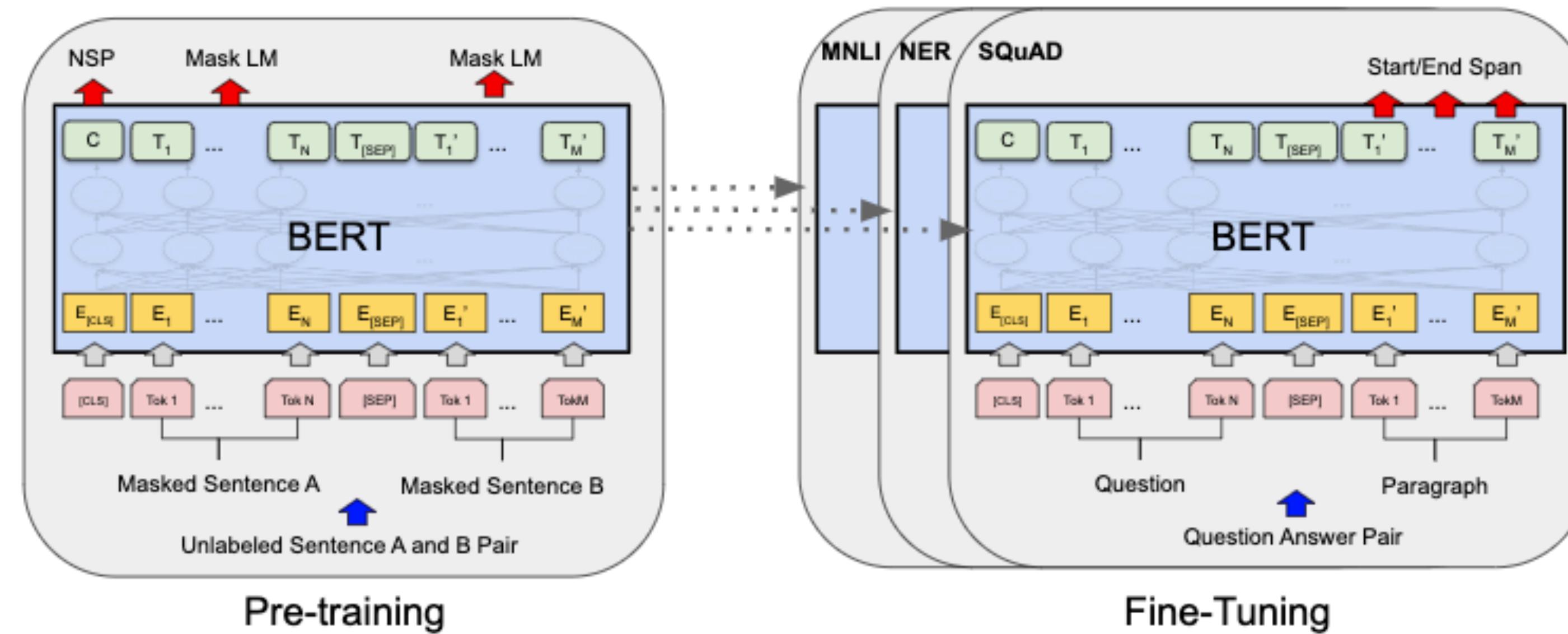
Prompting in GPT-3 for English to French Translation

Open Questions

- **Fine-tuning is expensive**
 - Parameter-efficient fine-tuning
- **Design of prompts is tricky**
 - Human Instructions
 - Reinforcement Learning with Human Feedback

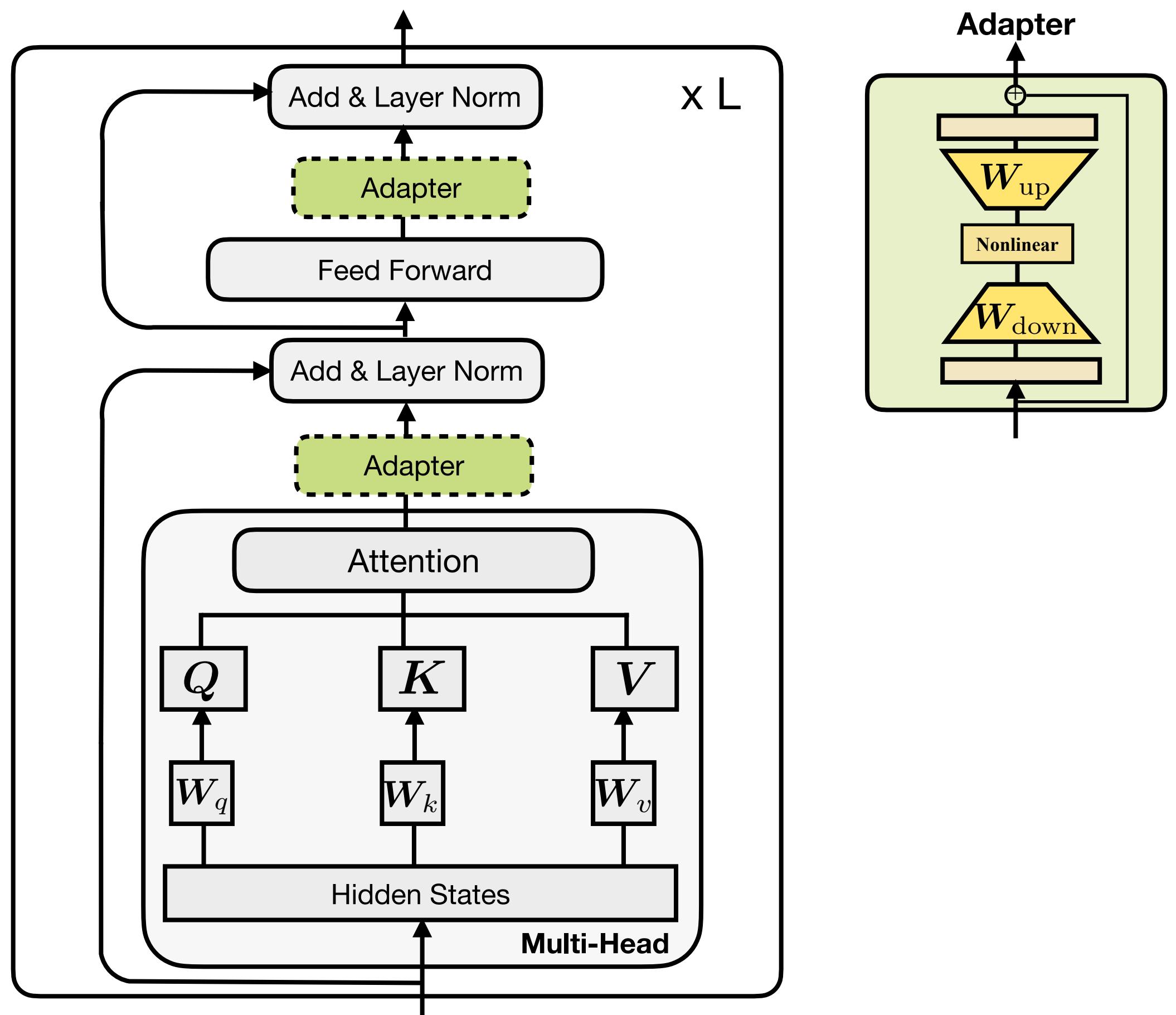
Parameter-Efficient Fine-tuning

- Fully Fine-tuning



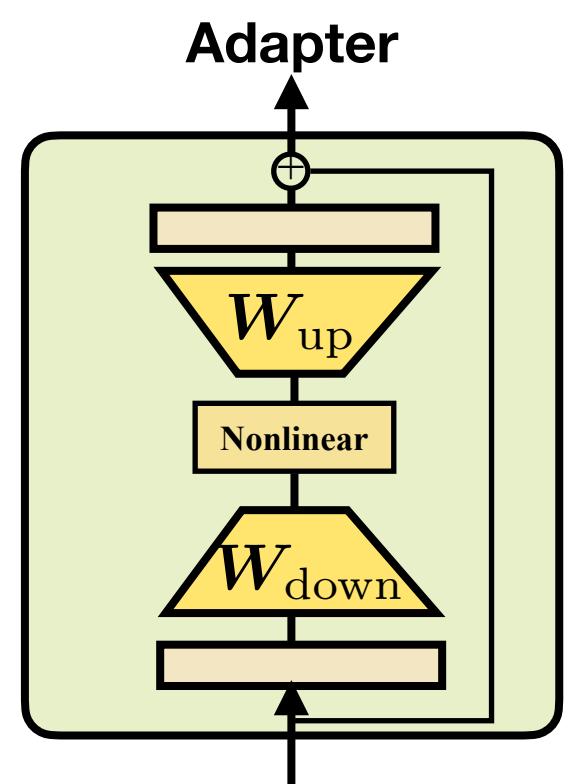
Each task needs a separate model copy
→ expensive as the number of tasks and model size grow

Parameter-Efficient Fine-tuning



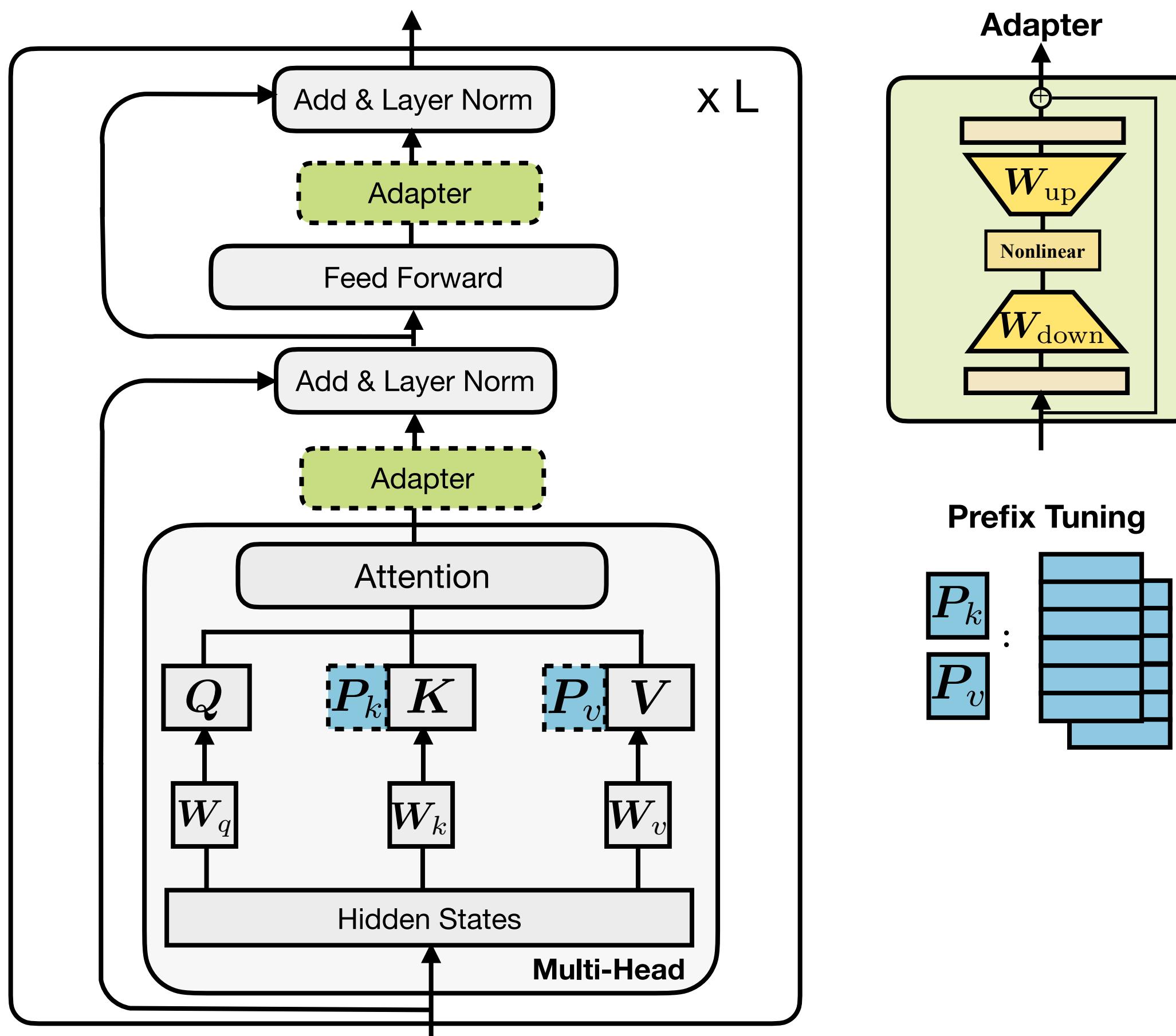
Adapters:

$$h \leftarrow h + f(hW_{\text{down}})W_{\text{up}}$$



[1] Houlsby et al. Parameter-Efficient Transfer Learning for NLP. ICML 2019

Parameter-Efficient Fine-tuning



Adapters:

$$\mathbf{h} \leftarrow \mathbf{h} + f(\mathbf{h}\mathbf{W}_{\text{down}})\mathbf{W}_{\text{up}}$$

Prefix Tuning:

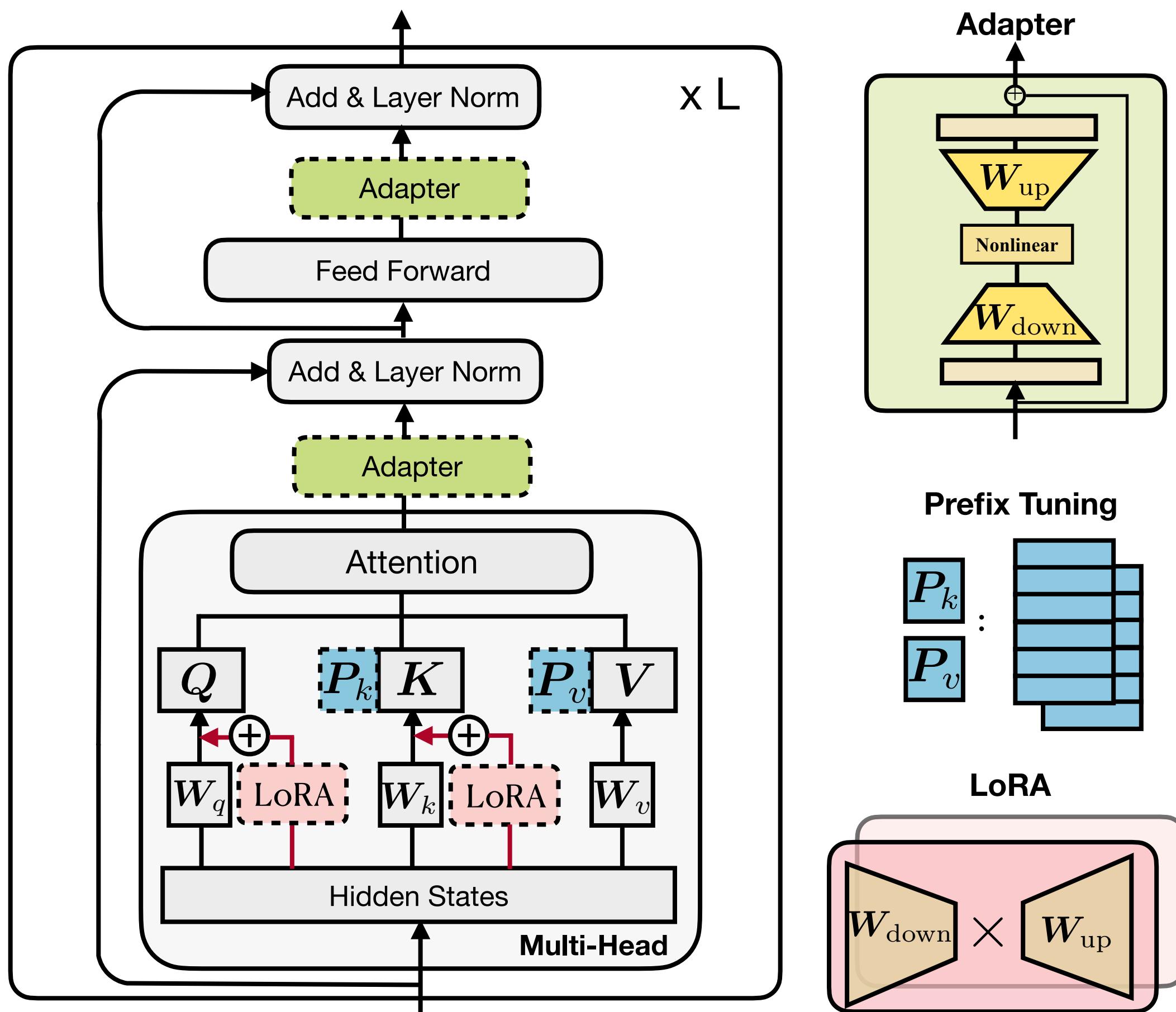
head_i

$$= \text{Attn}(\mathbf{x}\mathbf{W}_q^{(i)}, \text{concat}(\mathbf{P}_k^{(i)}, \mathbf{C}\mathbf{W}_k^{(i)}), \text{concat}(\mathbf{P}_v^{(i)}, \mathbf{C}\mathbf{W}^{(i)}))$$

[1] Houlsby et al. Parameter-Efficient Transfer Learning for NLP. ICML 2019

[2] Li et al. Prefix-Tuning: Optimizing Continuous Prompts for Generation. ACL 2021

Parameter-Efficient Fine-tuning



Adapters:

$$\mathbf{h} \leftarrow \mathbf{h} + f(\mathbf{h} \mathbf{W}_{\text{down}}) \mathbf{W}_{\text{up}}$$

Prefix Tuning:

head_i

$$= \text{Attn}(\mathbf{x} \mathbf{W}_q^{(i)}, \text{concat}(\mathbf{P}_k^{(i)}, \mathbf{C} \mathbf{W}_k^{(i)}), \text{concat}(\mathbf{P}_v^{(i)}, \mathbf{C} \mathbf{W}^{(i)}))$$

LoRA:

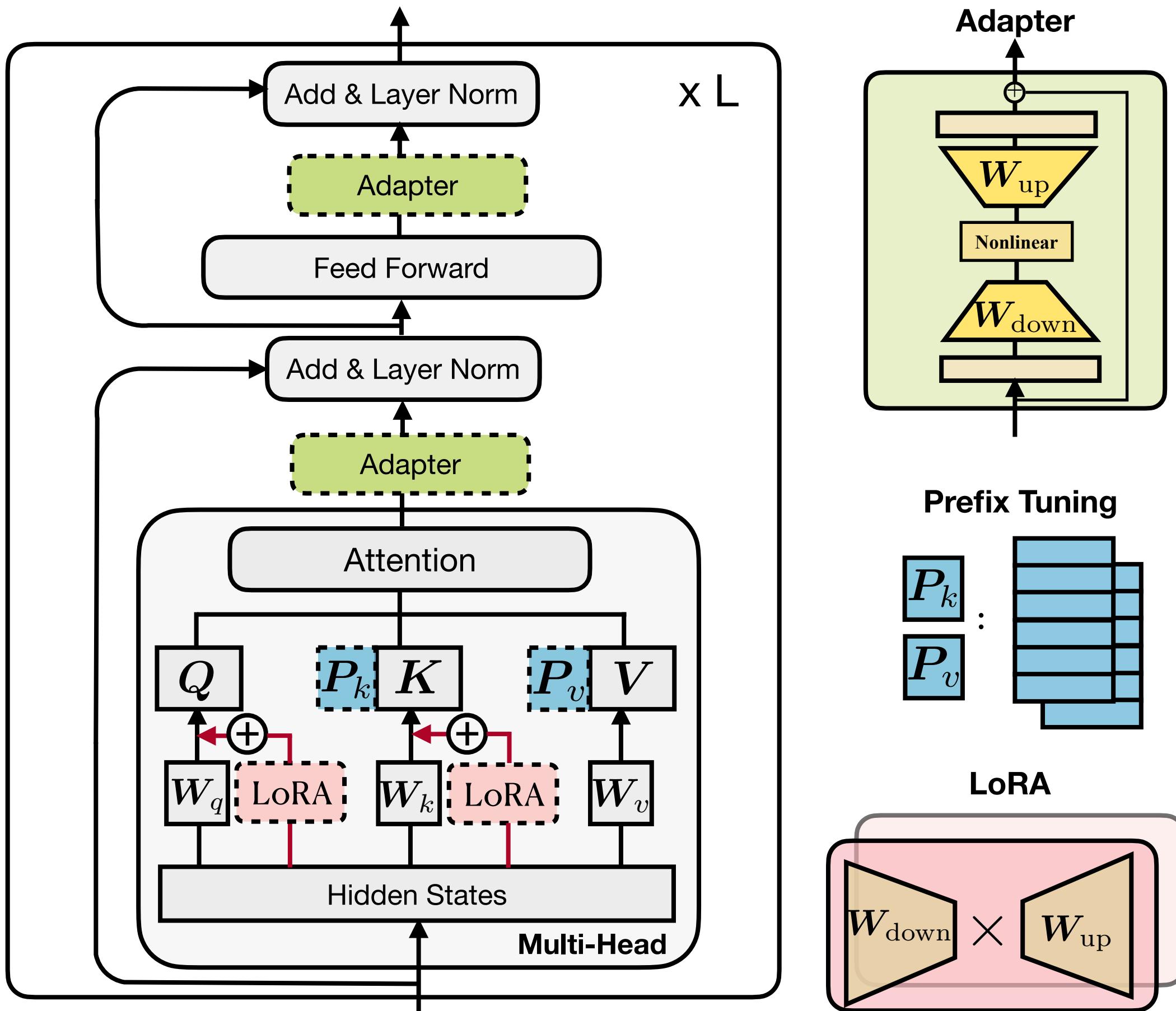
$$\mathbf{h} \leftarrow \mathbf{h} + s \cdot \mathbf{x} \mathbf{W}_{\text{down}} \mathbf{W}_{\text{up}}$$

[1] Houlsby et al. Parameter-Efficient Transfer Learning for NLP. ICML 2019

[2] Li et al. Prefix-Tuning: Optimizing Continuous Prompts for Generation. ACL 2021

[3] Hu et al. LoRA: Low-Rank Adaptation of Large Language Models. Preprint 2021

Existing Methods



He et al., 2022:

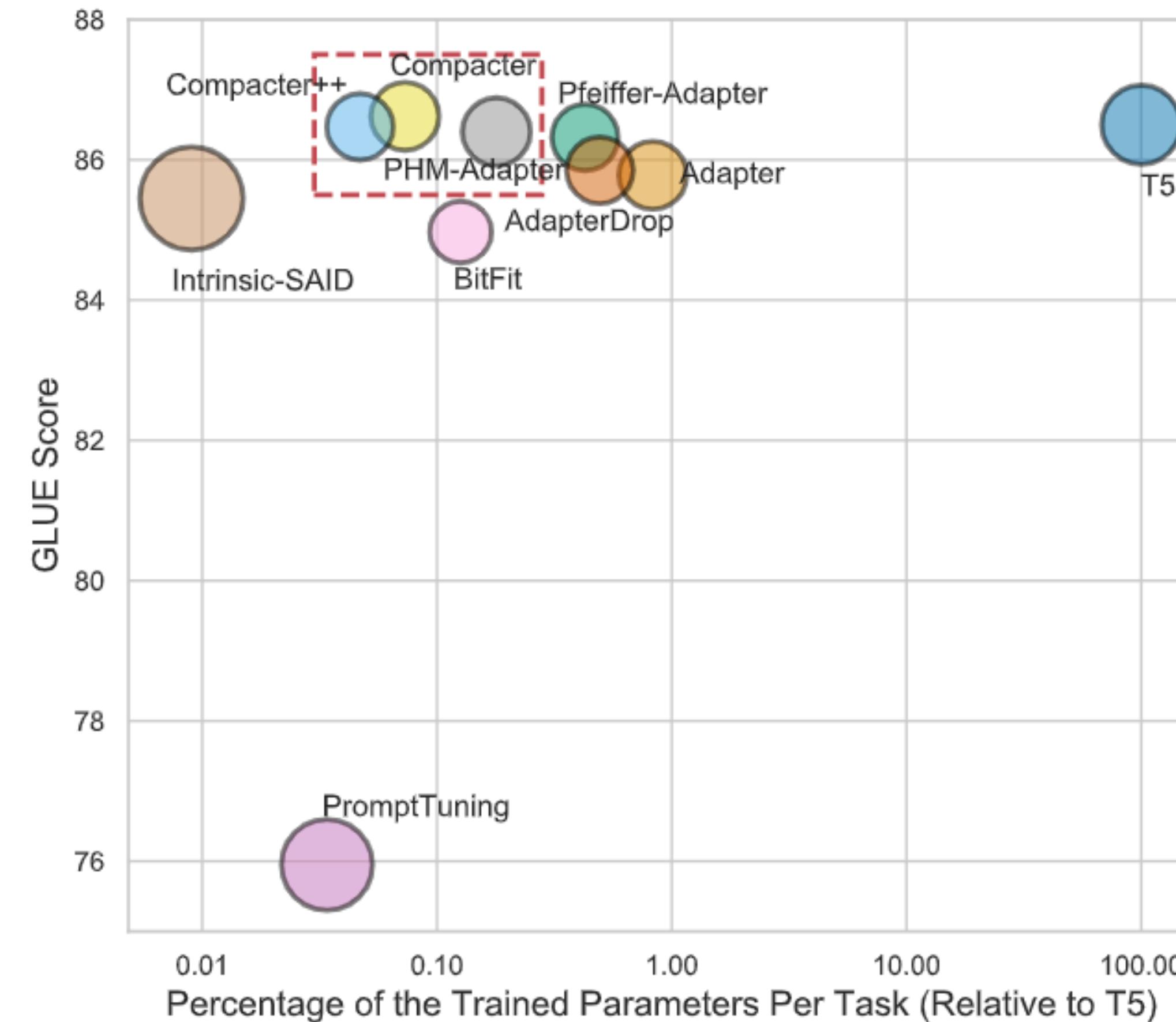
- A unified framework for all the three existing methods
- Enabling to design better adapters

[1] Houlsby et al. Parameter-Efficient Transfer Learning for NLP. ICML 2019

[2] Li et al. Prefix-Tuning: Optimizing Continuous Prompts for Generation. ACL 2021

[3] Hu et al. LoRA: Low-Rank Adaptation of Large Language Models. Preprint 2021

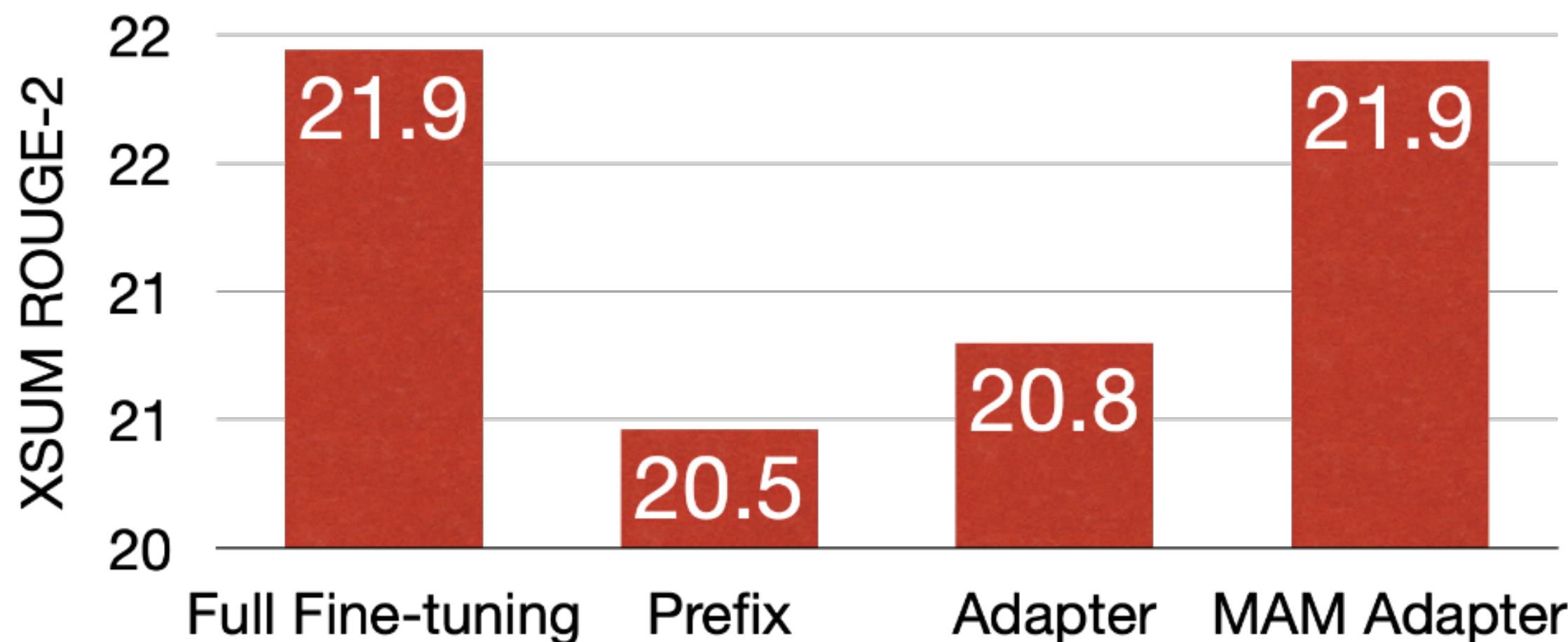
Parameter-Efficient Fine-tuning



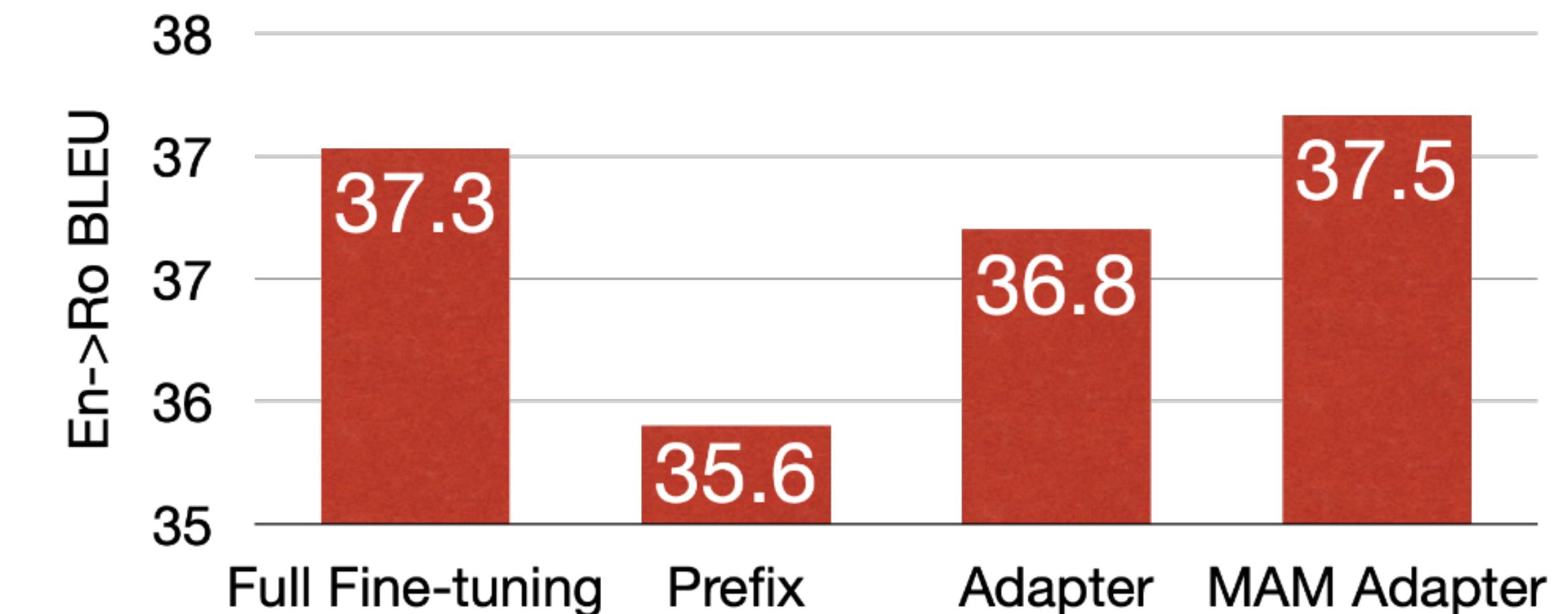
Less than 1% of parameters are tuned to achieve comparable performance to full fine-tuning (He et al., 2022)

Mix-and-Match (MAM) Adapter

XSum: Abstractive Summarization, BART-large



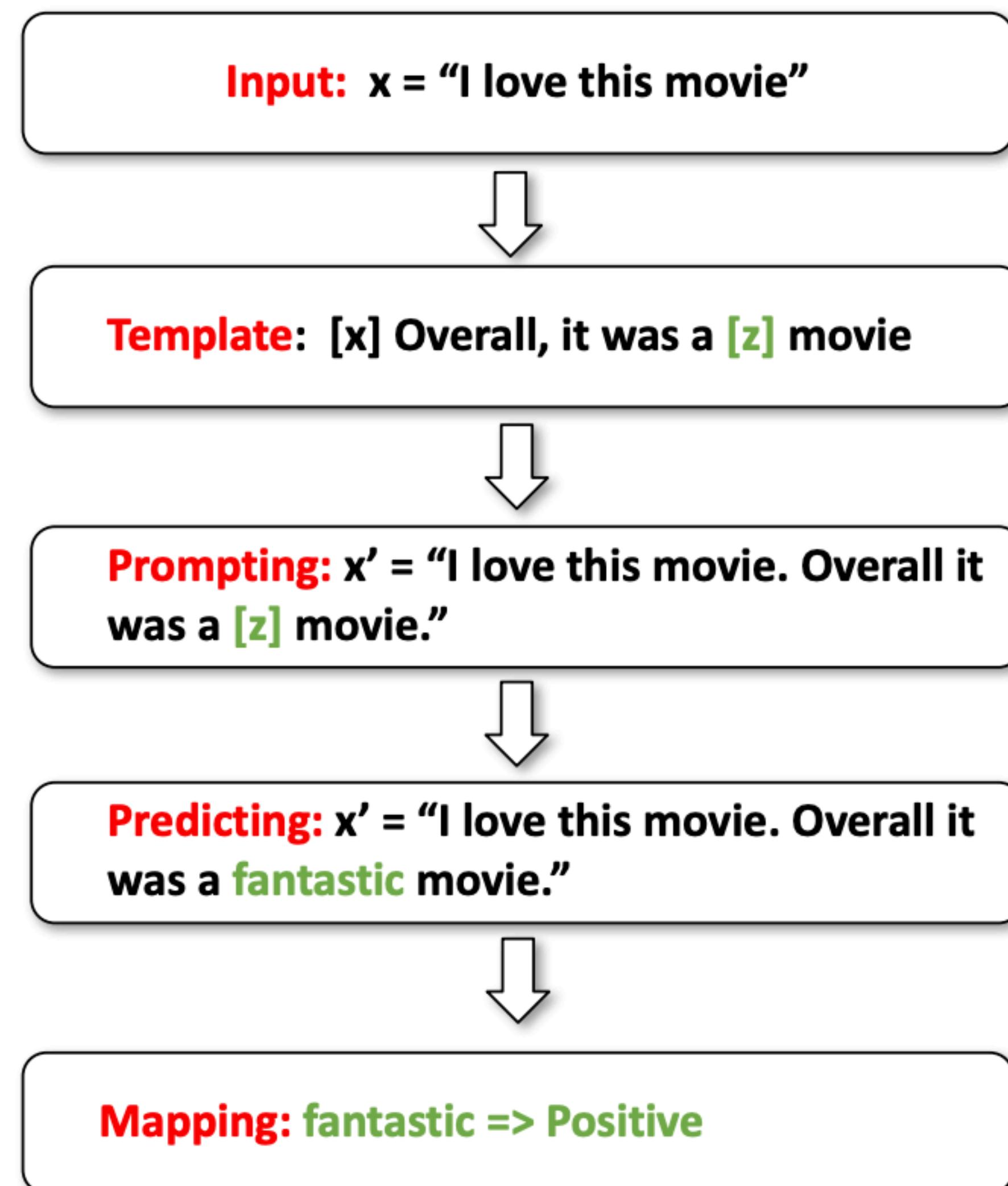
WMT16 En-Ro Translation, mBART-large



Comparable performance to full fine-tuning while tuning 6.7% relative size of parameters

Fine-tuning via Human Instructions

- Prompting/In context Learning



Example: sentiment classification

Fine-tuning via Human Instructions

- Can we directly communicate with LLMs via instructions in human languages

What is the sentiment of “I love this movie”

The sentiment of this sentence is **positive**,
because ...

Fine-tuning via Human Instructions

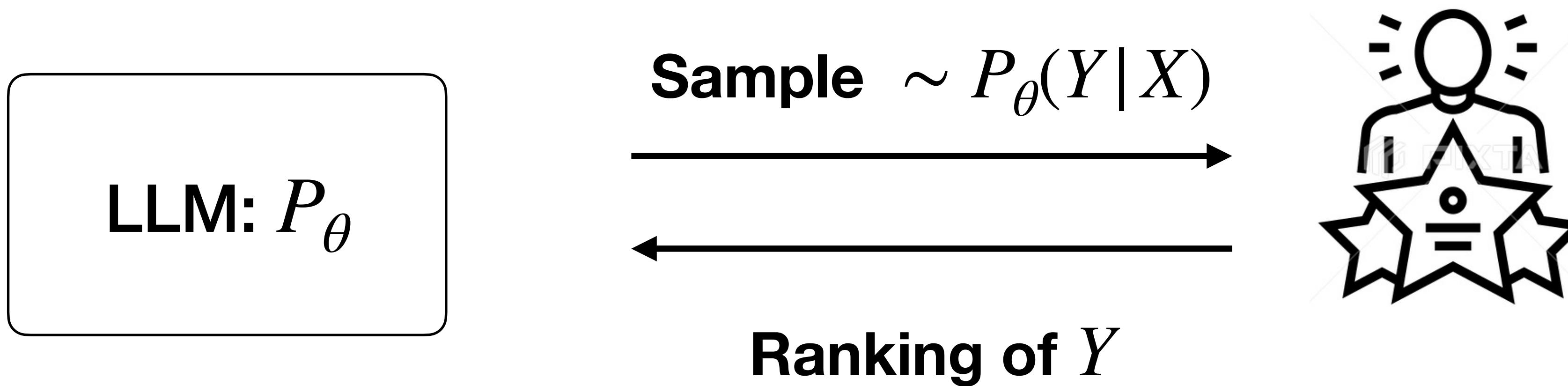
- Collect a set of prompts/instructions
 - Different categories
- Create a dataset with demonstration data
 - Demonstrating the desired output behavior
- Fine-tune the model with the demonstration data

Demonstration Data

Use-case	(%)	Use-case	Prompt
Generation	45.6%	Brainstorming	List five ideas for how to regain enthusiasm for my career
Open QA	12.4%		
Brainstorming	11.2%	Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Chat	8.4%		
Rewrite	6.6%	Rewrite	This is the summary of a Broadway play: """ {summary} """
Summarization	4.2%		
Classification	3.5%		
Other	3.5%		
Closed QA	2.6%		
Extract	1.9%		

The ability of following human-instruction is generalizable to general cases

Reinforcement Learning with Human Feedbacks



Reading Materials

- Relevant Papers

- BERT
- GPT-3
- BART
- Prompting
- Parameter-Efficient Fine-tuning