CSCI 544: Applied Natural Language Processing

Contextualized Embeddings & Large-scale Pre-training

Xuezhe Ma (Max)

USC University of Southern California

Outline

- **Large-scale Pre-training**
  - Contextualized Embeddings: Pre-trained Encoder
  - Neural Language Modeling: Pre-trained Decoder
  - Denoising Seq2seq Modeling: Pre-trained Encoder-Decoder
- **Using Pre-trained Models**
  - Fully Fine-tuning
  - Parameter-Efficient Fine-tuning
  - Prompting

Contextualized Embeddings: Pre-trained Encoders

Contextualized Embeddings: Pre-trained Encoders

- ELMo = **E**mbeddings from **L**anguage **Mo**dels
- BERT = **B**idirectional **E**ncoder **R**epresentations from **T**ransformers

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 …
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 …
Cited by 17552 Related articles All 26 versions

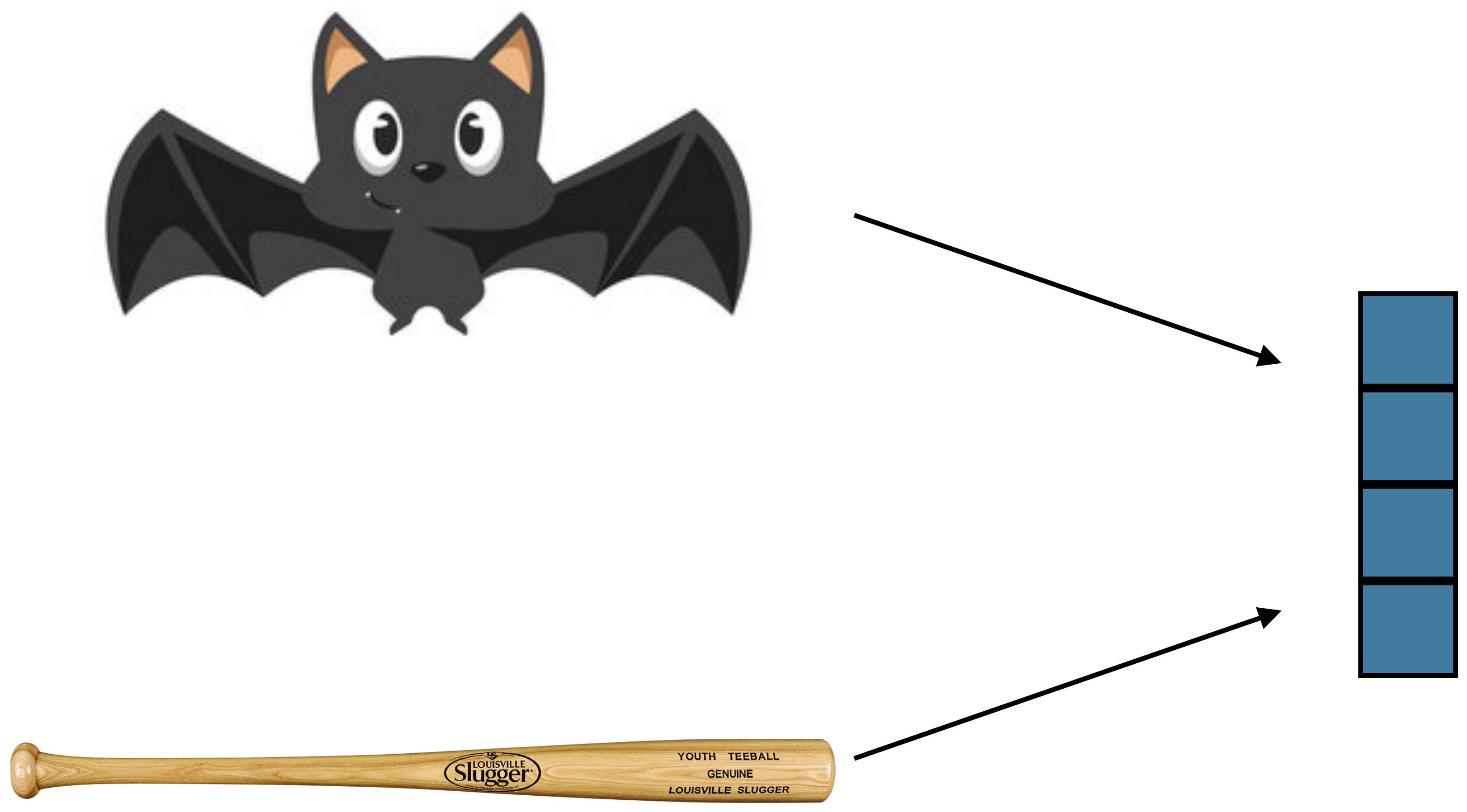
What’s Wrong with Word Embeddings?

- 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

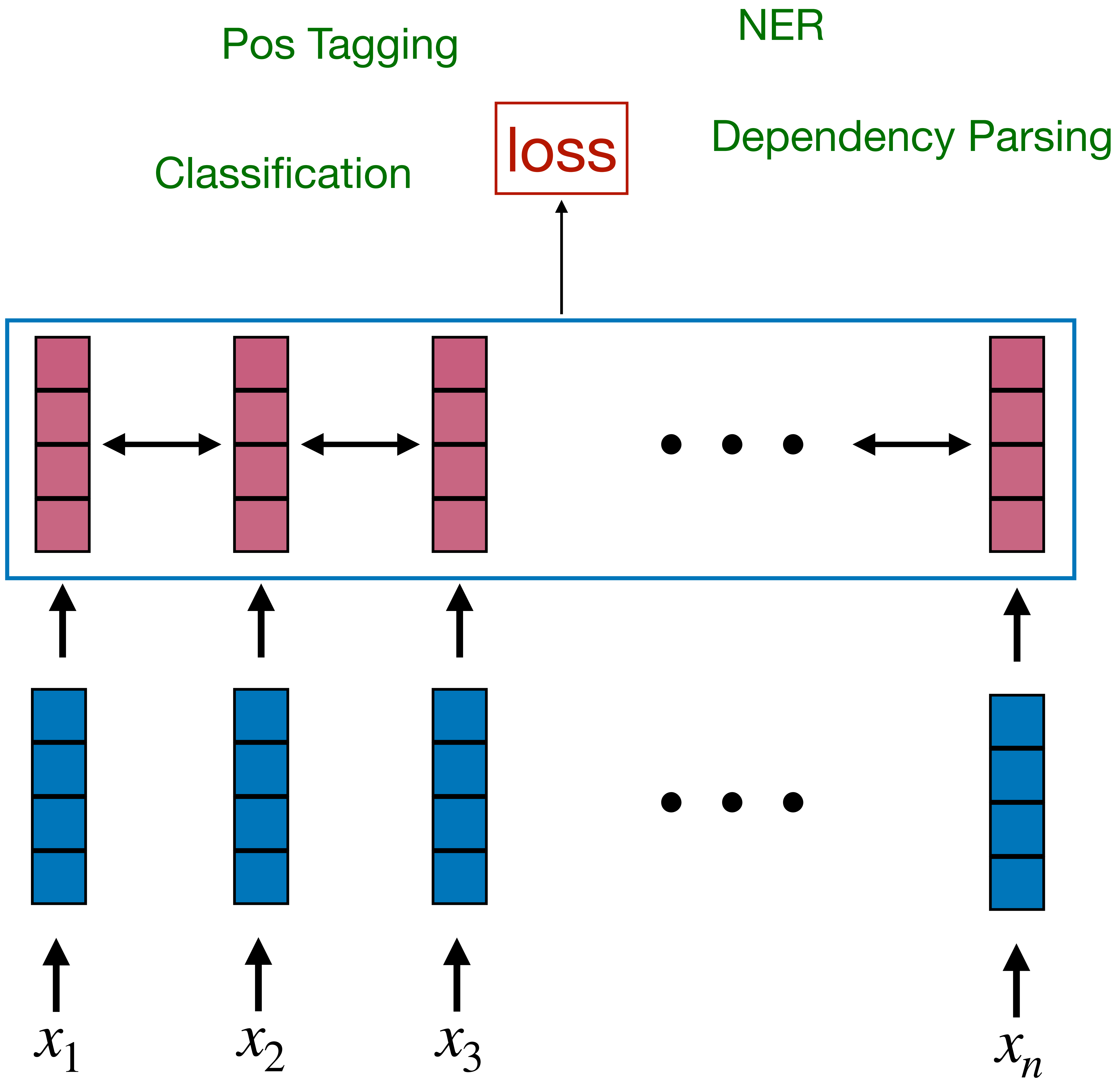
hit with bat

hit the bat

Let’s build a vector for each word conditioned on its context!

What's Wrong with Task-Specific Learning?

- **We have contextualized models!**



Contextualized Embeddings?

general contextualized embeddings!

Contextualized Word Embeddings

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play 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 play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

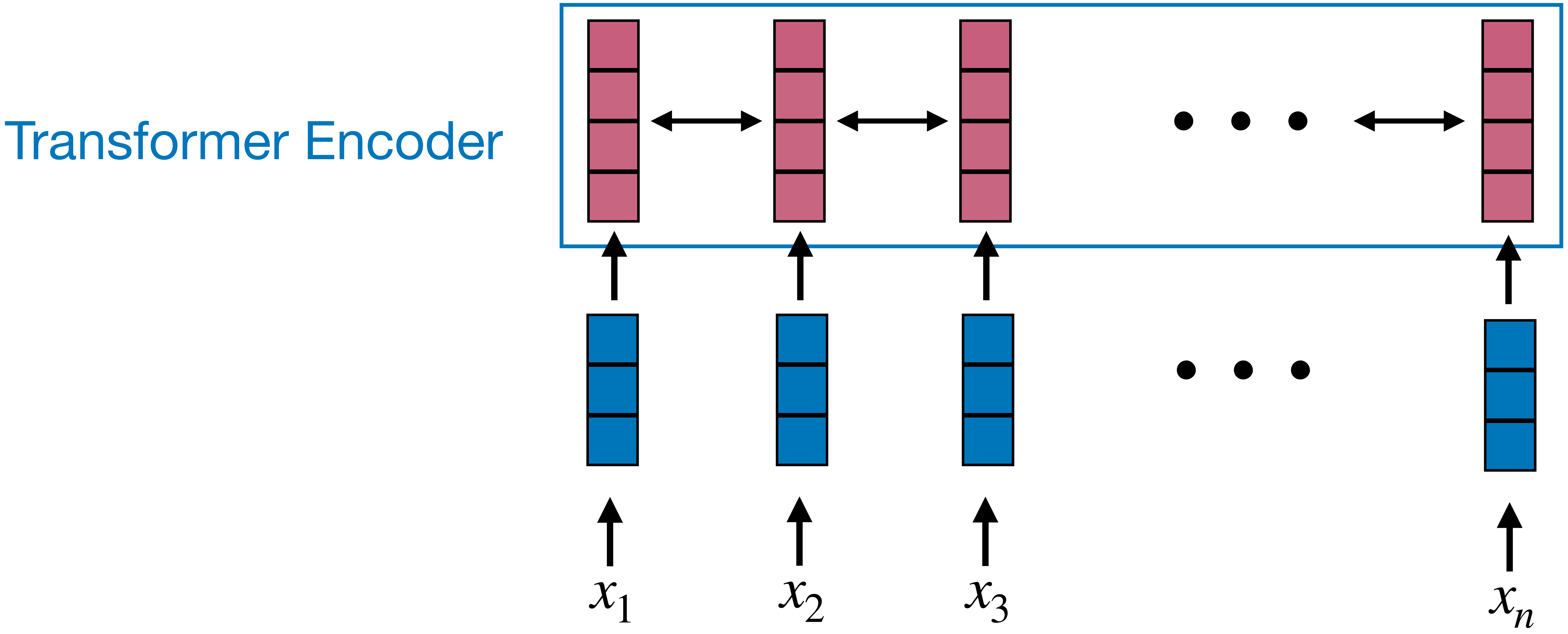
Deep contextualized word representations (Peters et al., 2018)

How can we get these contextualized embeddings?

- **The key idea of BERT:**
  - Train a Transformer encoder on a large corpus
  - Objective: masked language modeling
  - Use the hidden states of the Transformer for each token as contextualized embeddings for each word

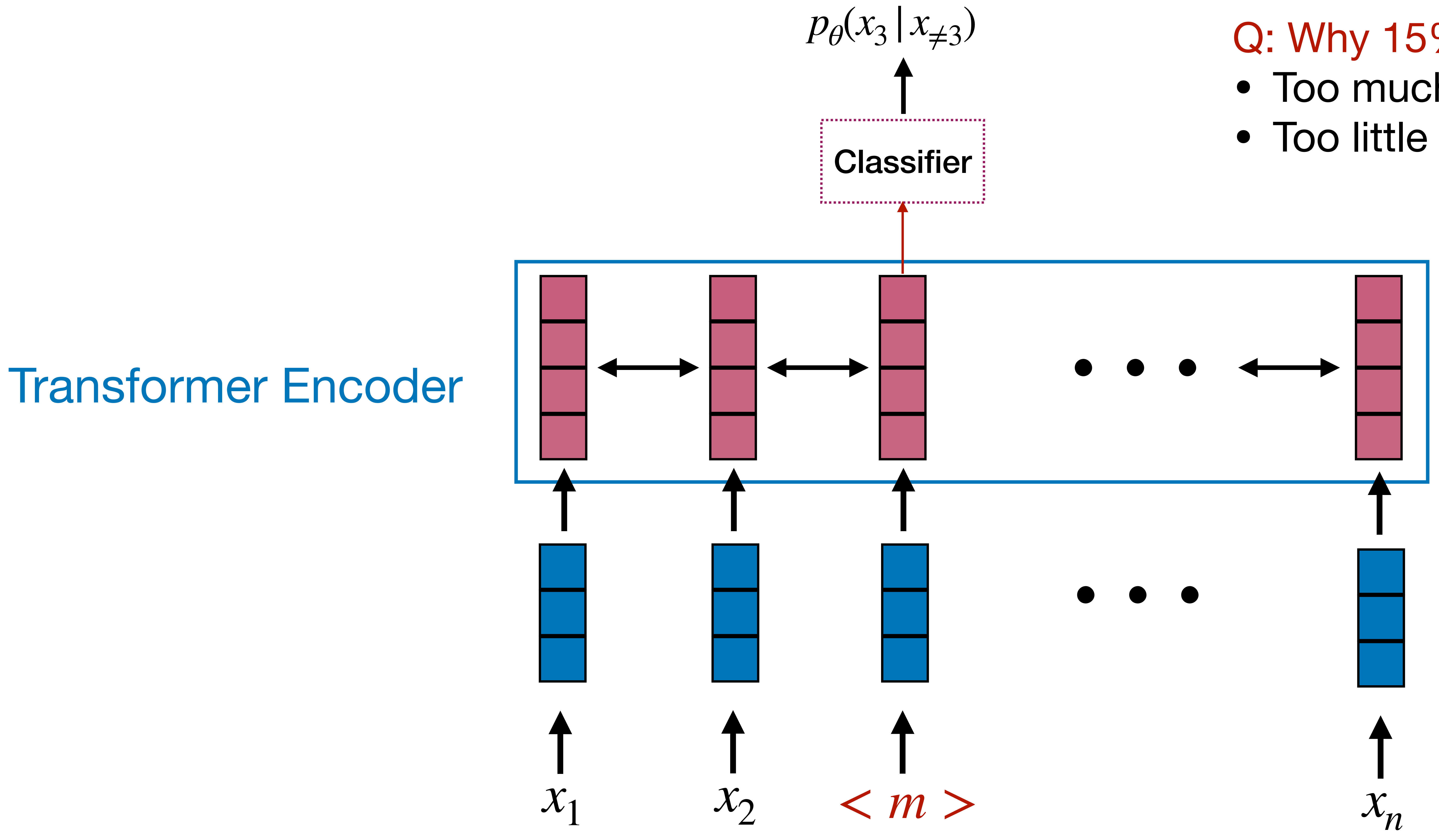
Masked Language Modeling

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



Masked Language Modeling

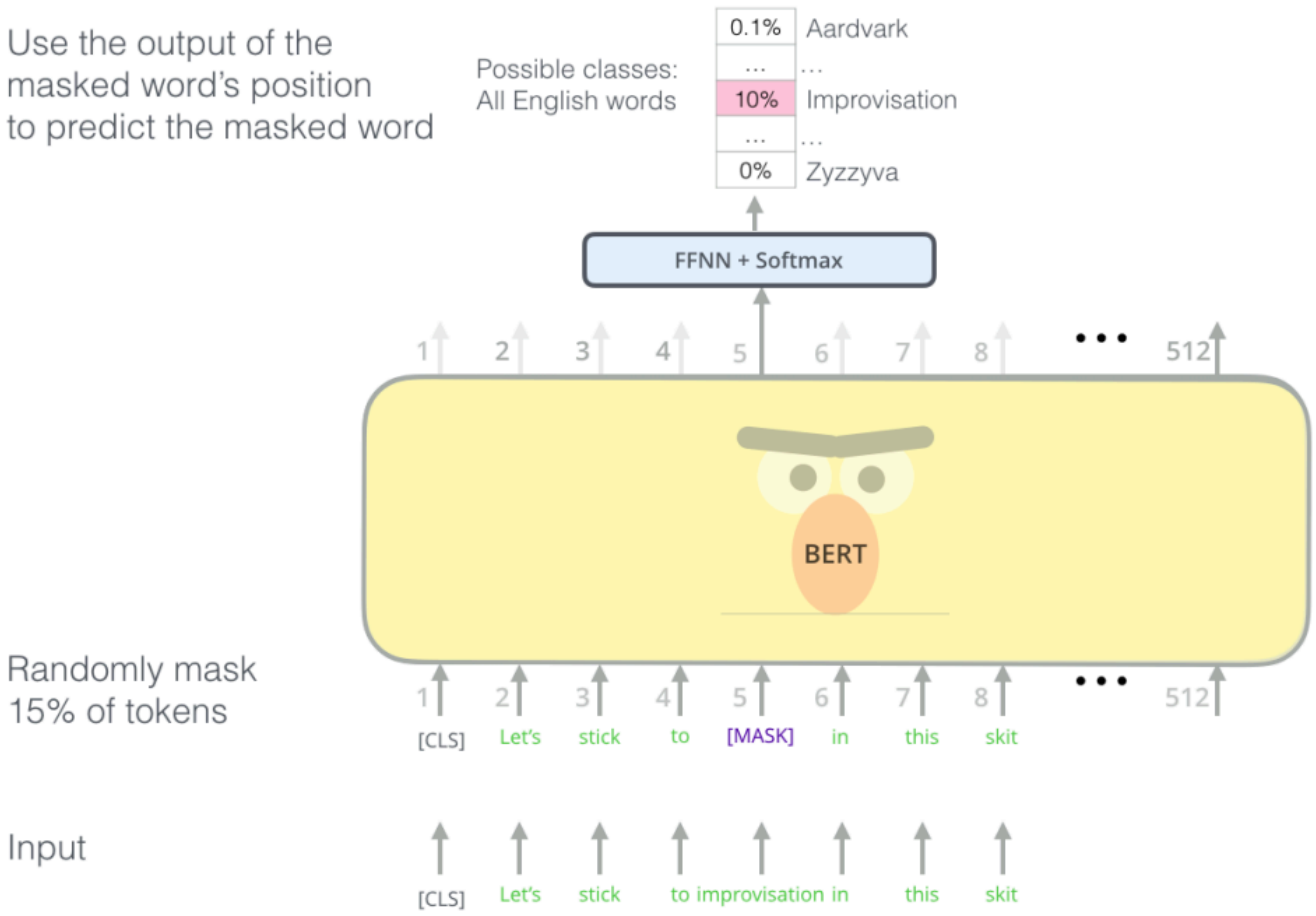
- **Mask out 15% of the input words, and then predict the masked words**



Q: Why 15%

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

Masked Language Modeling (MLM)



A stacked Transformer encoder

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 understanding tasks!

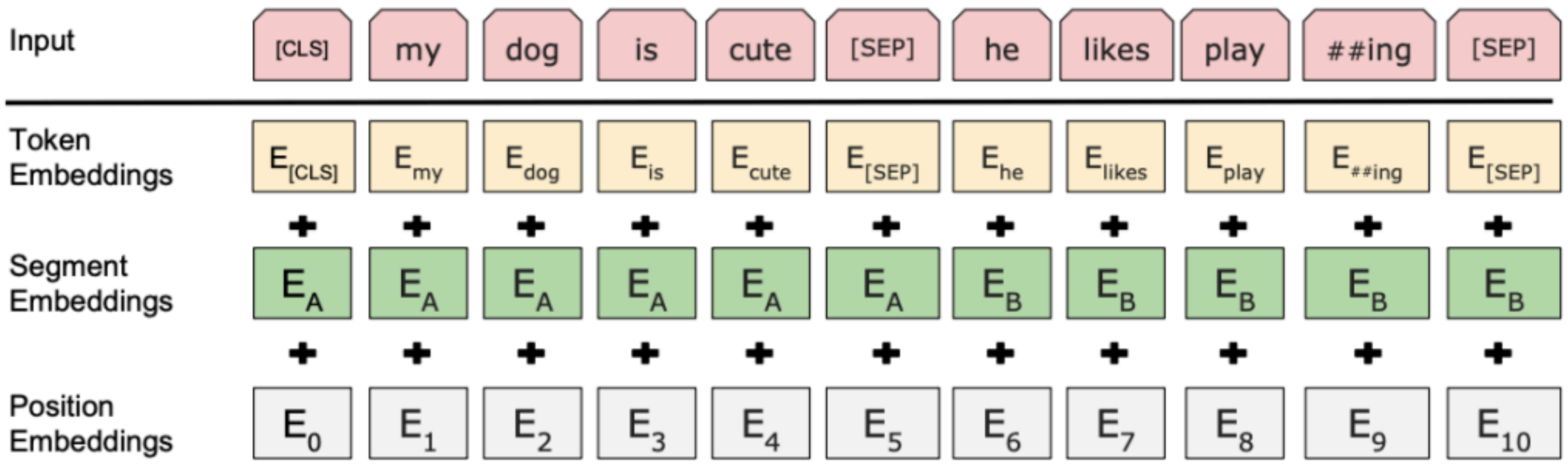
We will see some examples soon!

Training a BERT!

- **Training Data**
  - Wikipedia (2500M 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

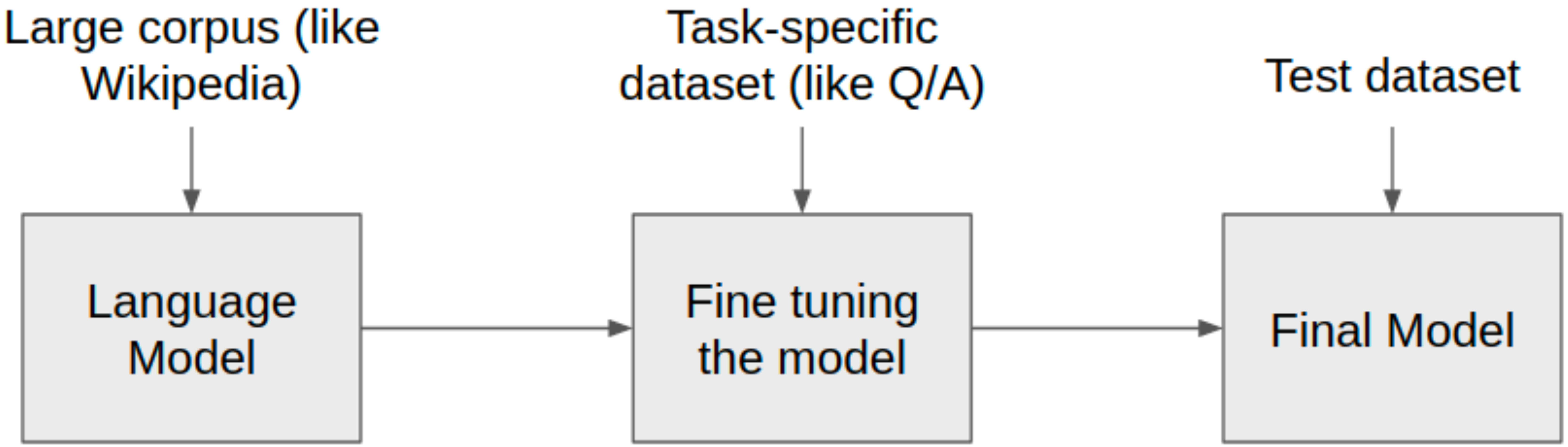
- Input representations



- 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

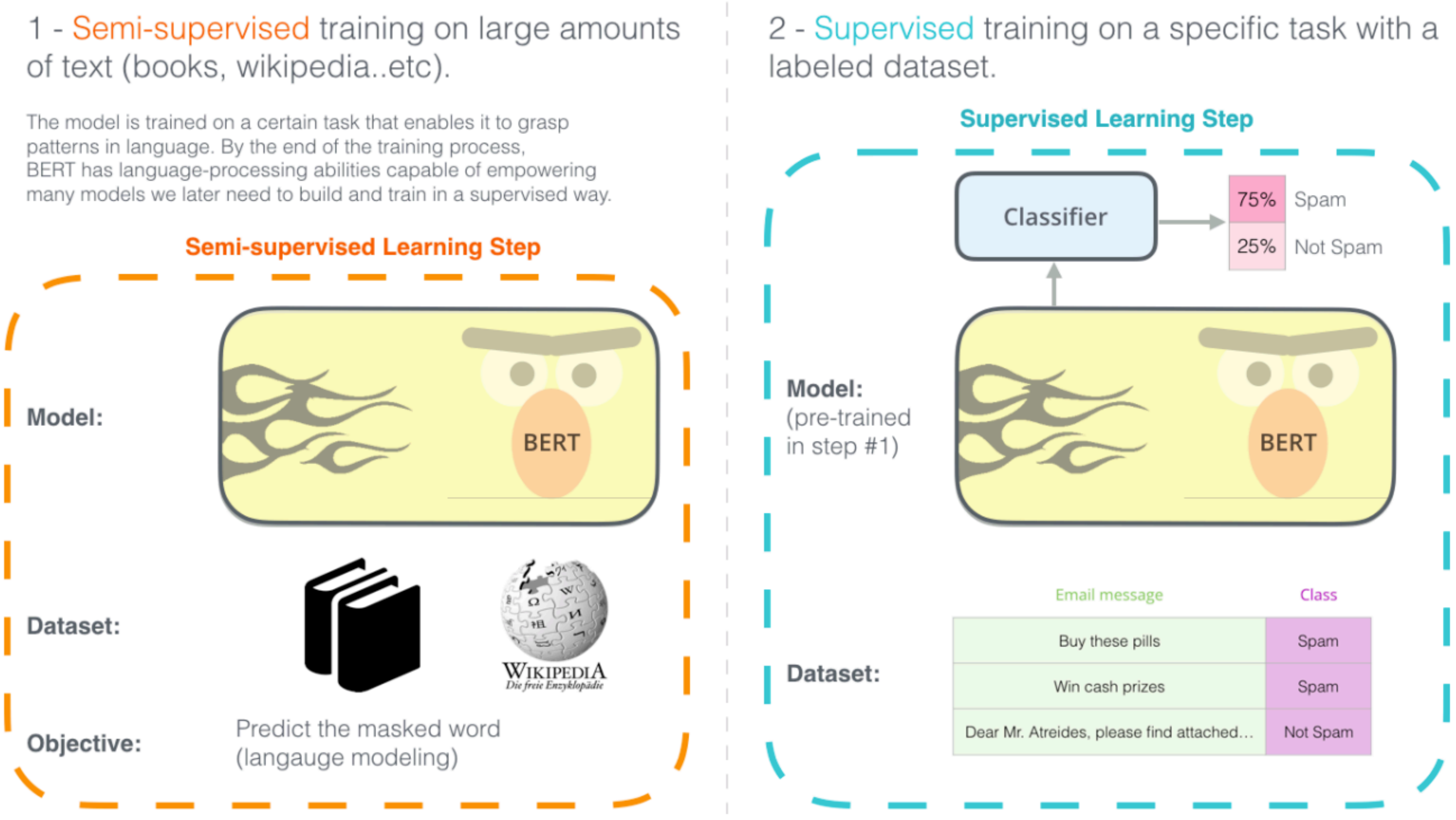
How to use BERT?

- **Fine-tuning BERT for downstream tasks!**

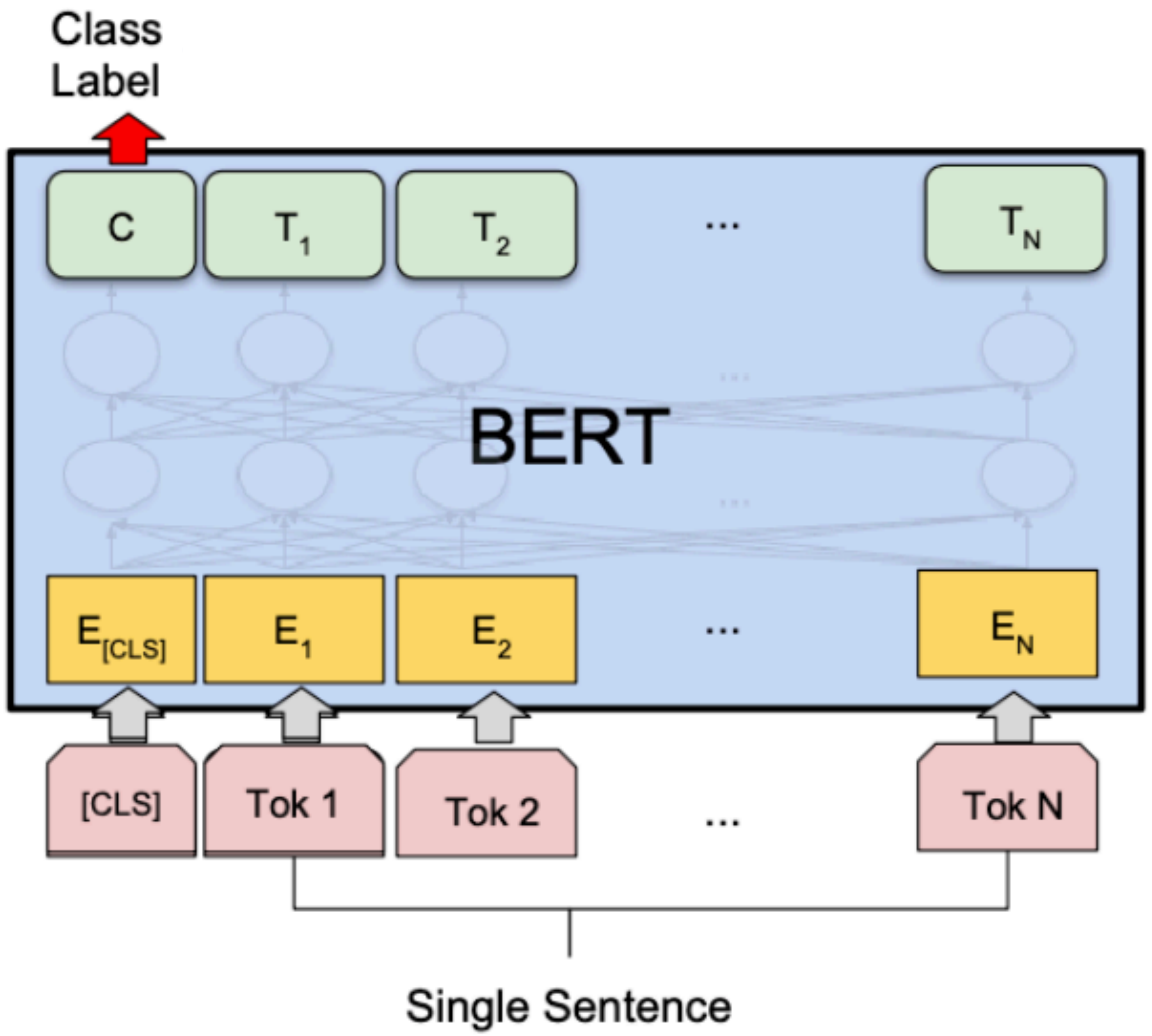


How to use BERT?

- **Fine-tuning BERT for downstream tasks!**



Example: Sentiment Classification



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
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**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 MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$\text{BERT}_{\text{BASE}}$	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

Unidirectional LMs don't work!

Table 5: Ablation over the pre-training tasks using the $\text{BERT}_{\text{BASE}}$ 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.

Hyperparams			Dev Set Accuracy			
#L	#H	#A	LM (ppl)	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: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., 2019)

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!

Neural Language Modeling: Pre-trained Decoders

Auto-regressive Generative Models

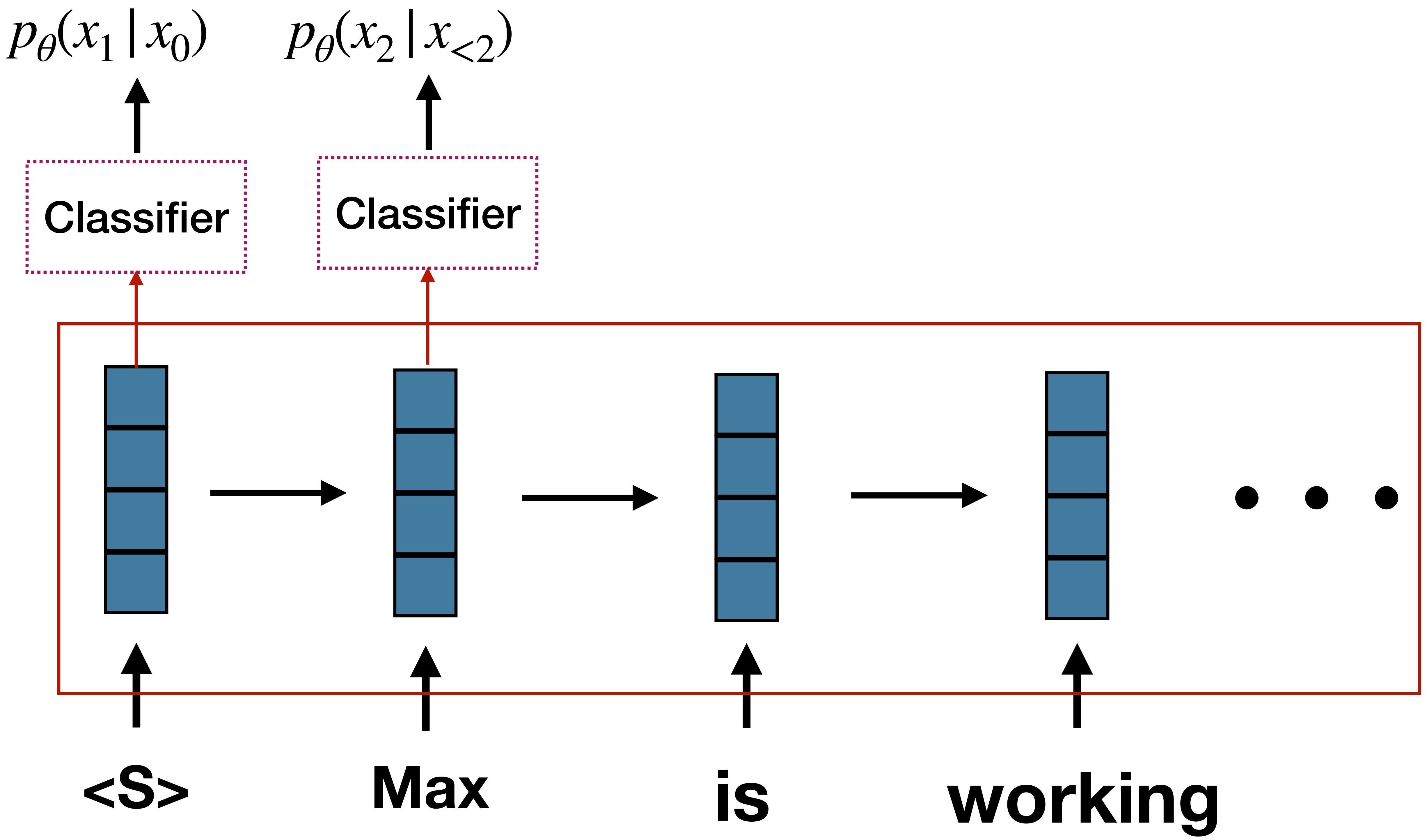
- **Auto-regressive Neural Language Models**

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

Auto-regressive Generative Models

- **Auto-regressive Neural Language Models**

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



Transformer Decoder
Basic Architecture for GPT-2&3

GPT-3

- **Training data**
  - Common Crawl (410B tokens)
  - WebText2 (19B tokens)
  - Books1 & Books2 (12B + 55B tokens)
  - Wikipedia (3B tokens)
- **Transformer Encoder**
  - Medium: 350M parameters
  - Largest: 175B parameters

Q: How to use GPT-3?
- Fine-tuning is too expensive
- Prompting!

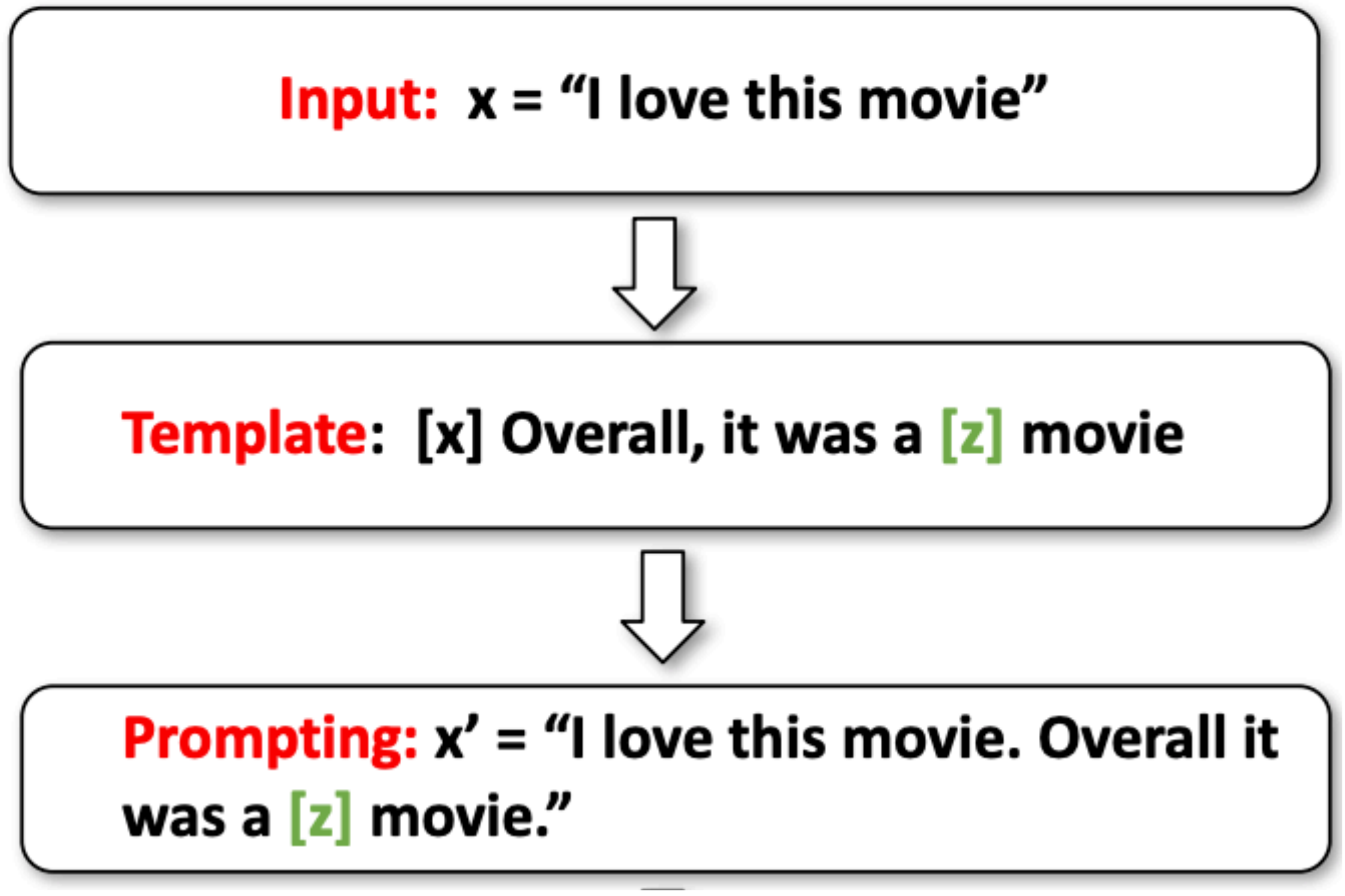
we will see some examples soon!

Prompting

- Prompt Addition
- Answer Prediction
- Answer-Label Mapping

Prompt Addition

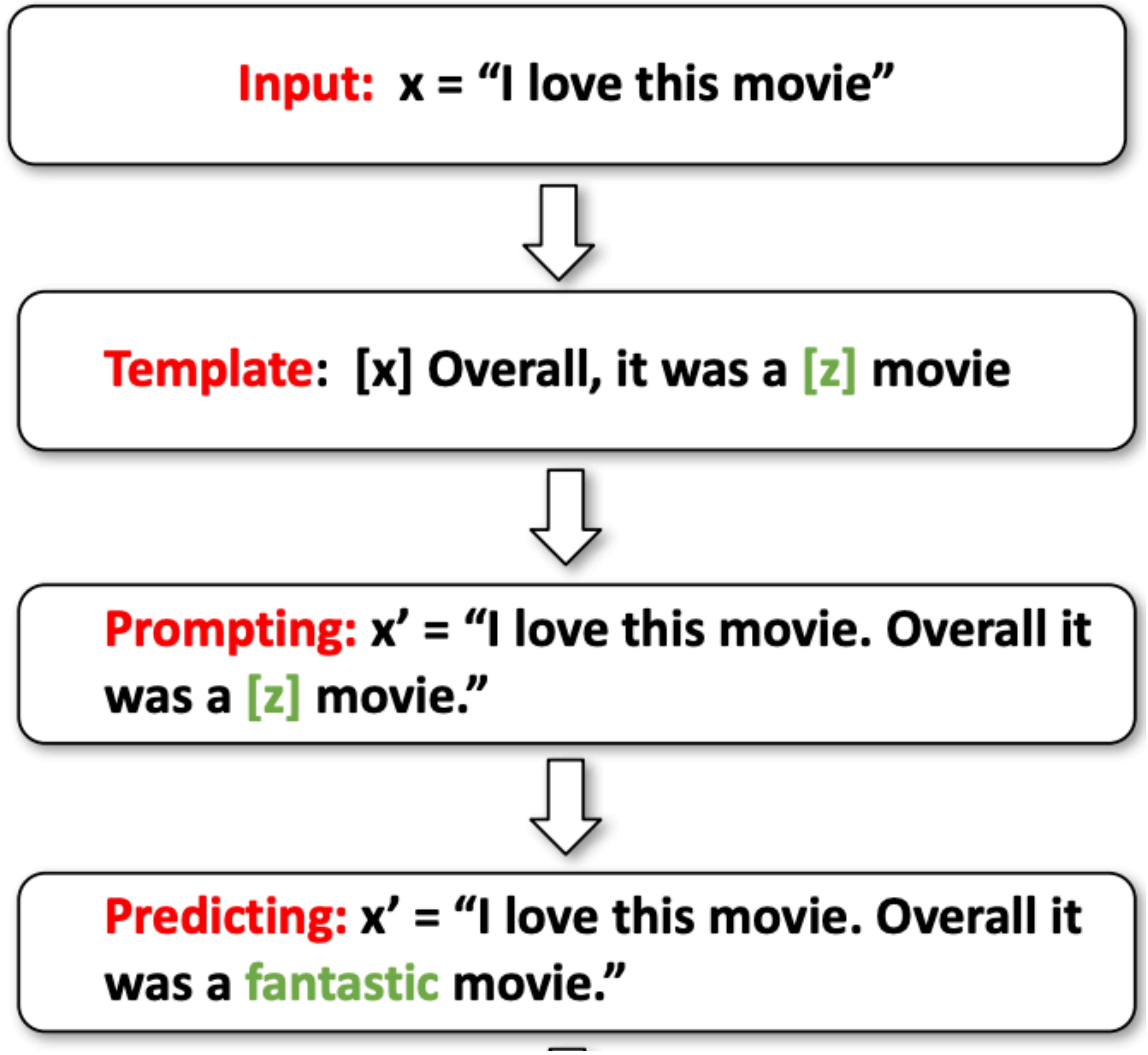
- **Given input x, we create a prompt with two steps:**
  - Define a template with two slots, one for input [x], and one for the answer [z]
  - Fill in the input with slot [x]



Example: sentiment classification

Answer Prediction

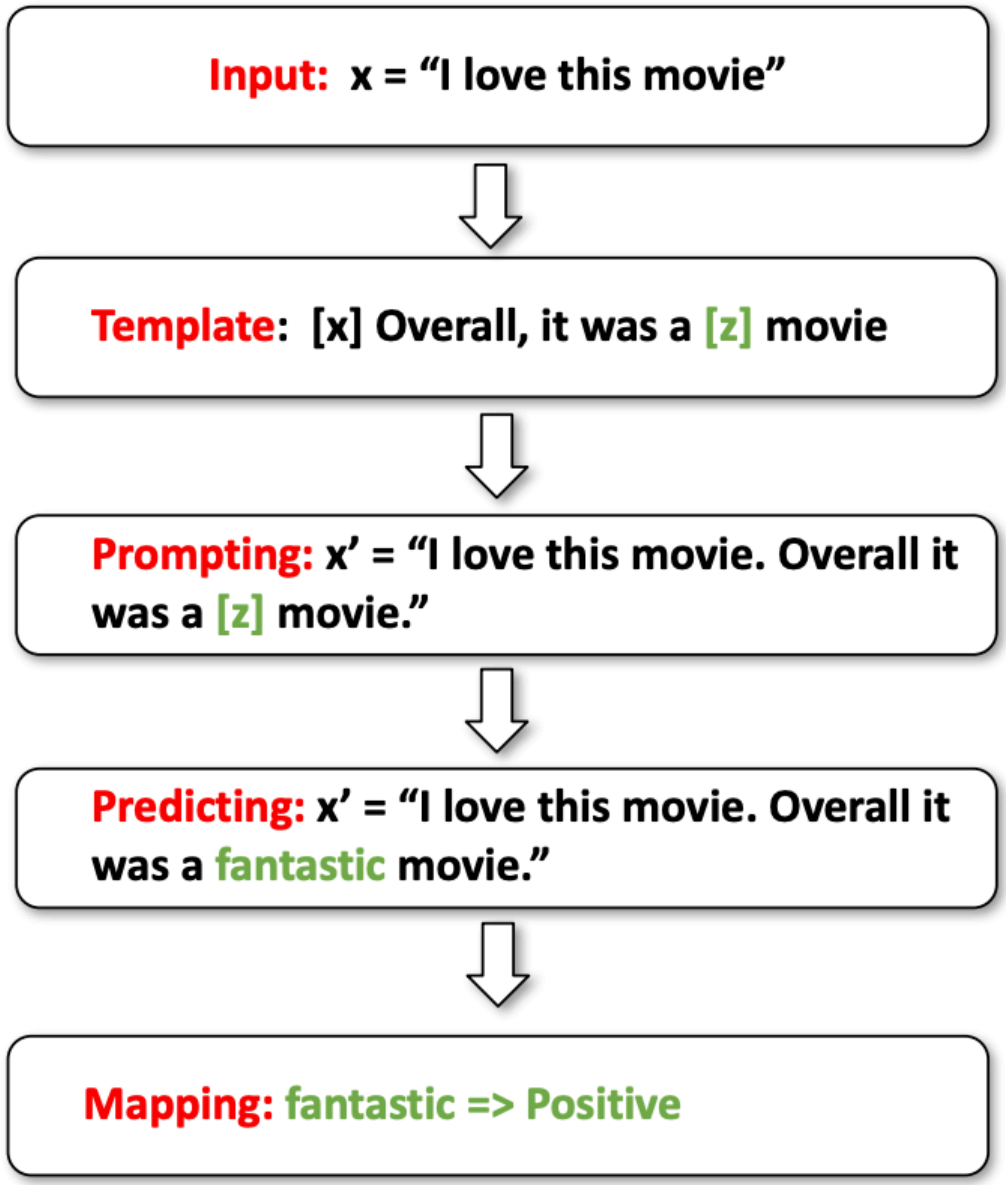
- Given a prompt, predict the answer [z]



Example: sentiment classification

Answer-Labeling Mapping

- Given an answer, map it into a class label



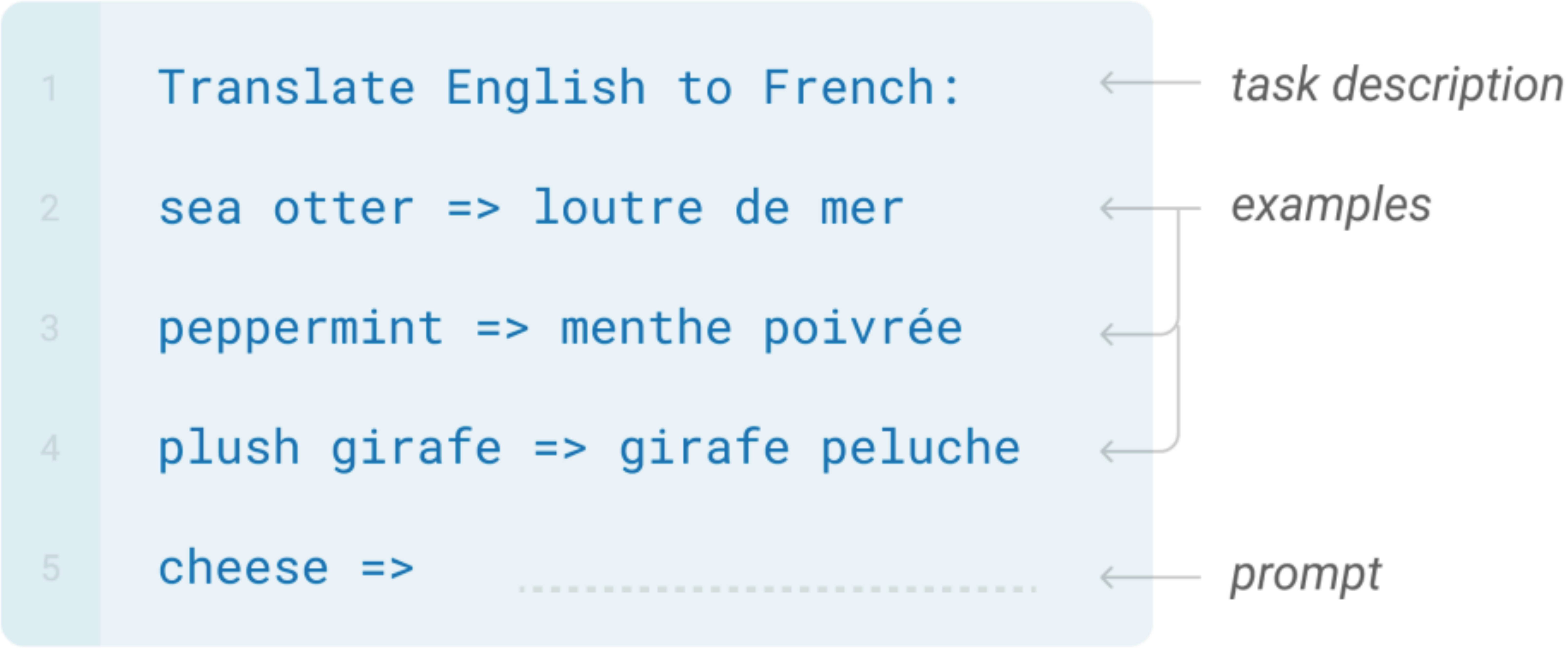
Example: sentiment classification

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

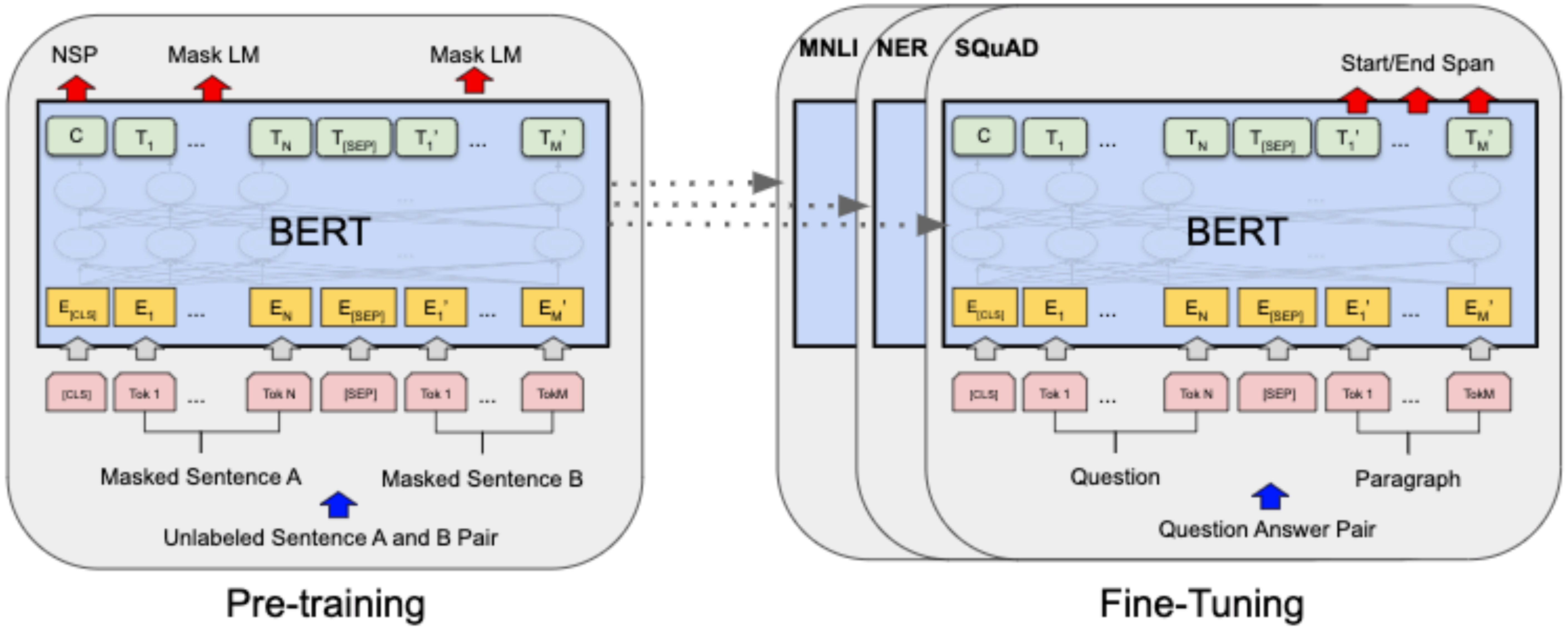
- Suppose we have a few (less than 20) examples of the task



Prompting in GPT-3 for English to French Translation

Parameter-Efficient Fine-tuning

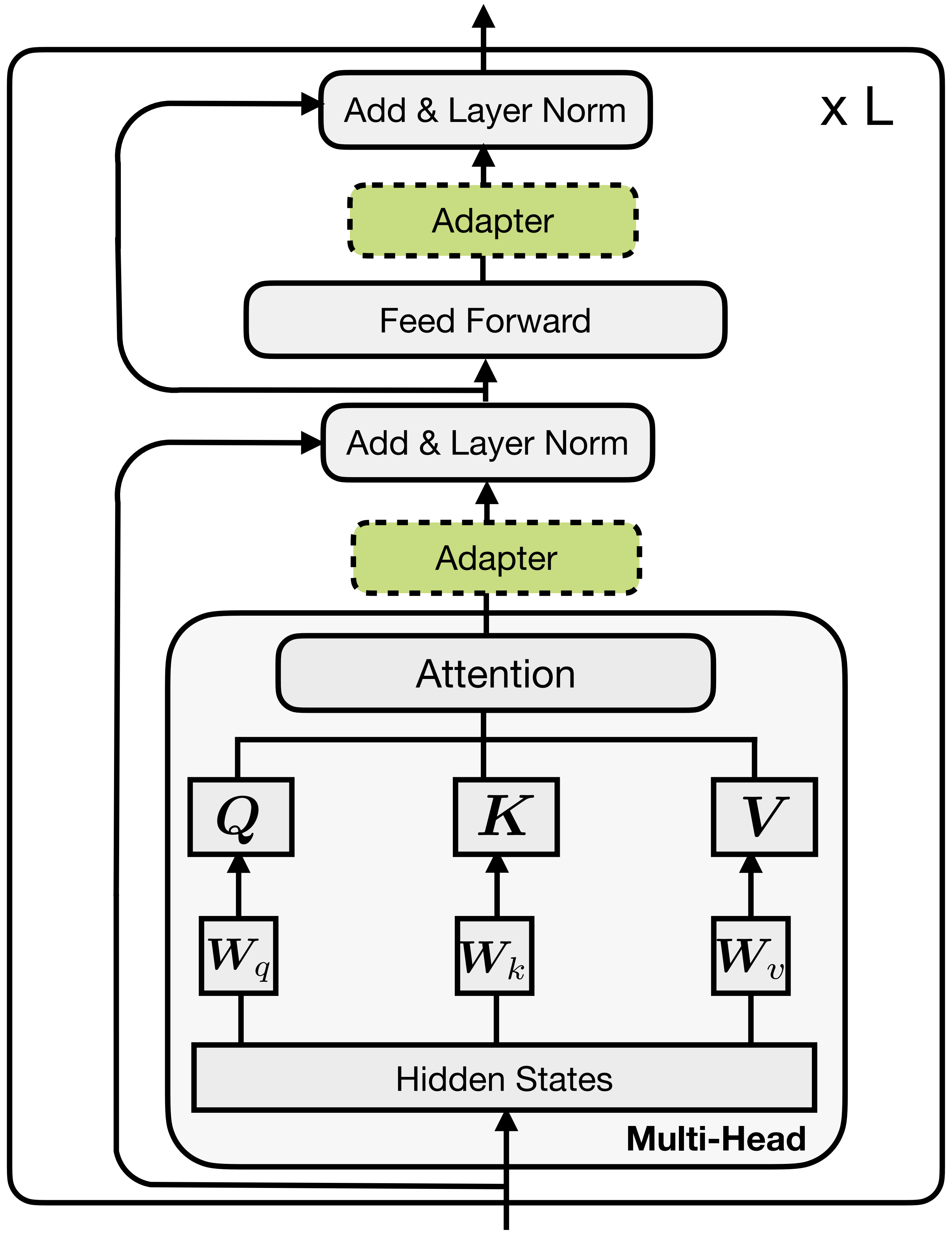
- **Fully Fine-tuning**

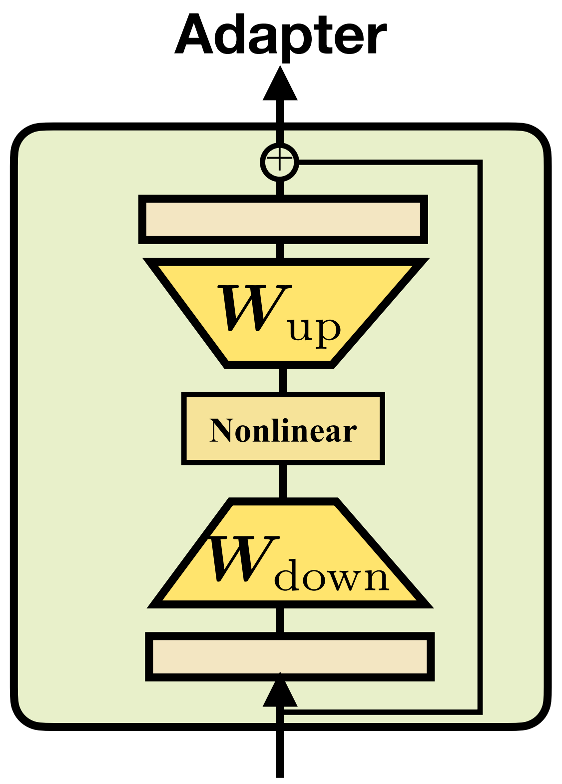


- **Prompting**
  - Performance is not as good as fully fine-tuning

Trade-off between them?

Parameter-Efficient Fine-tuning



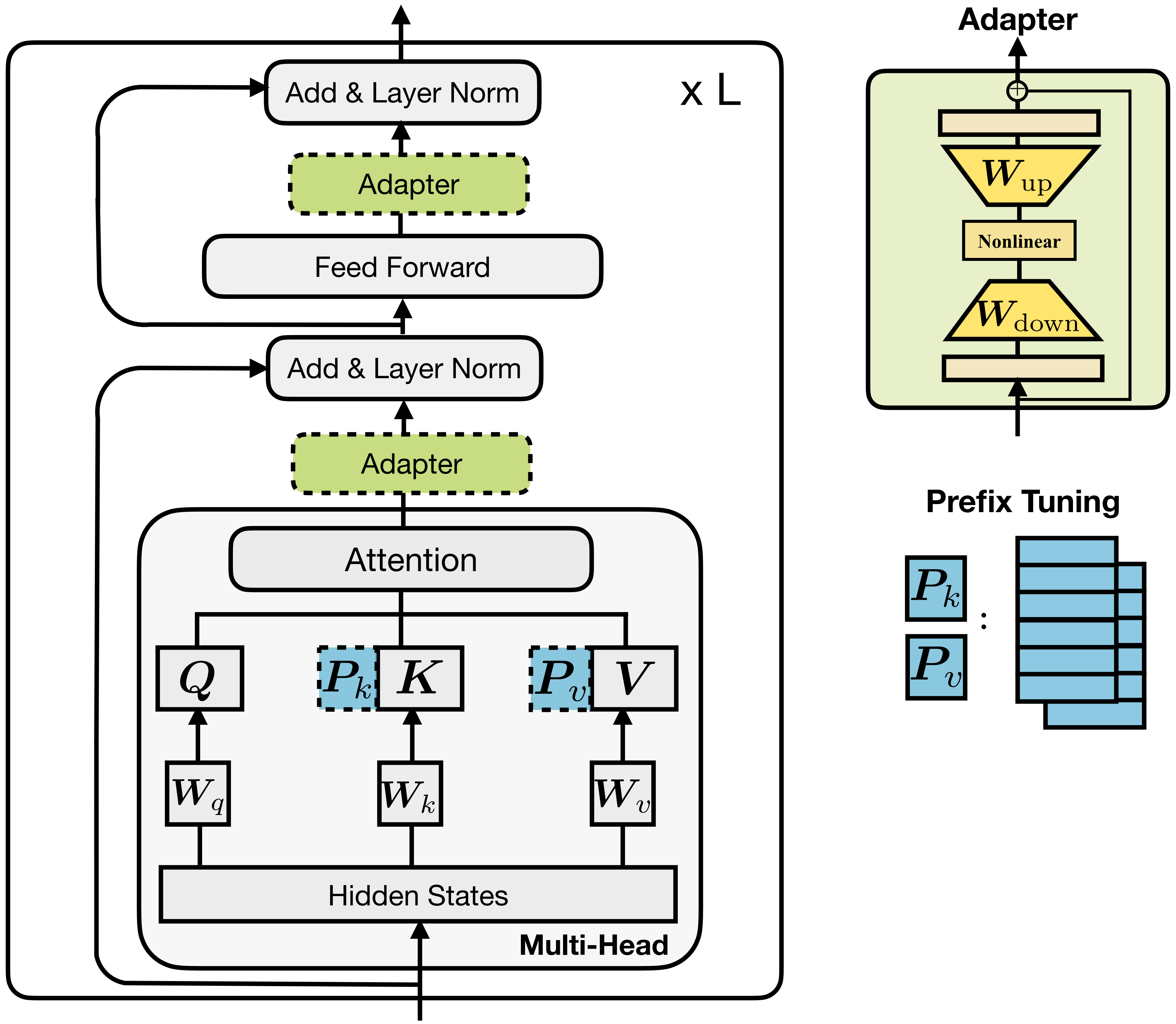


Adapters:

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

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

Parameter-Efficient Fine-tuning



Adapters:

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

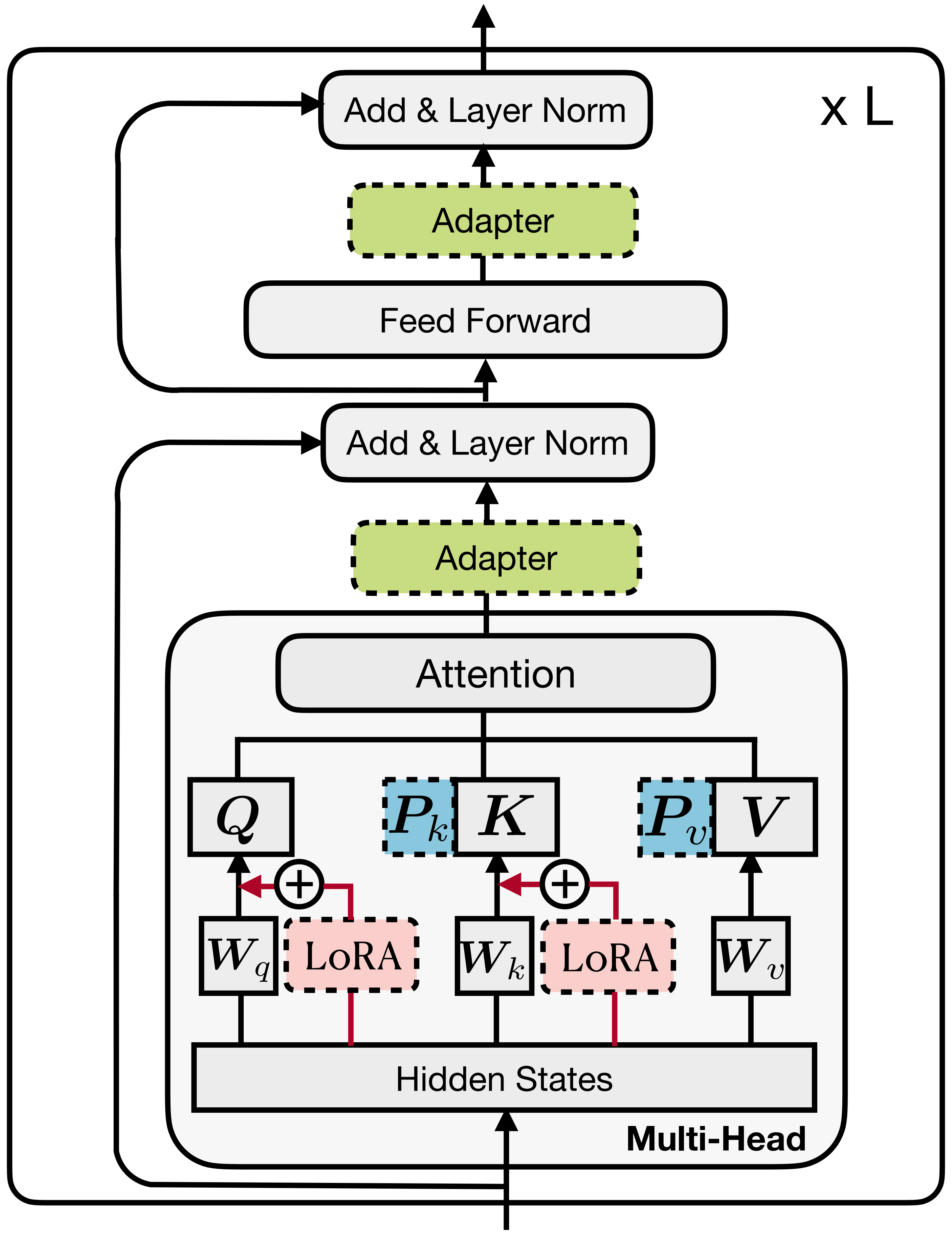
Prefix Tuning:

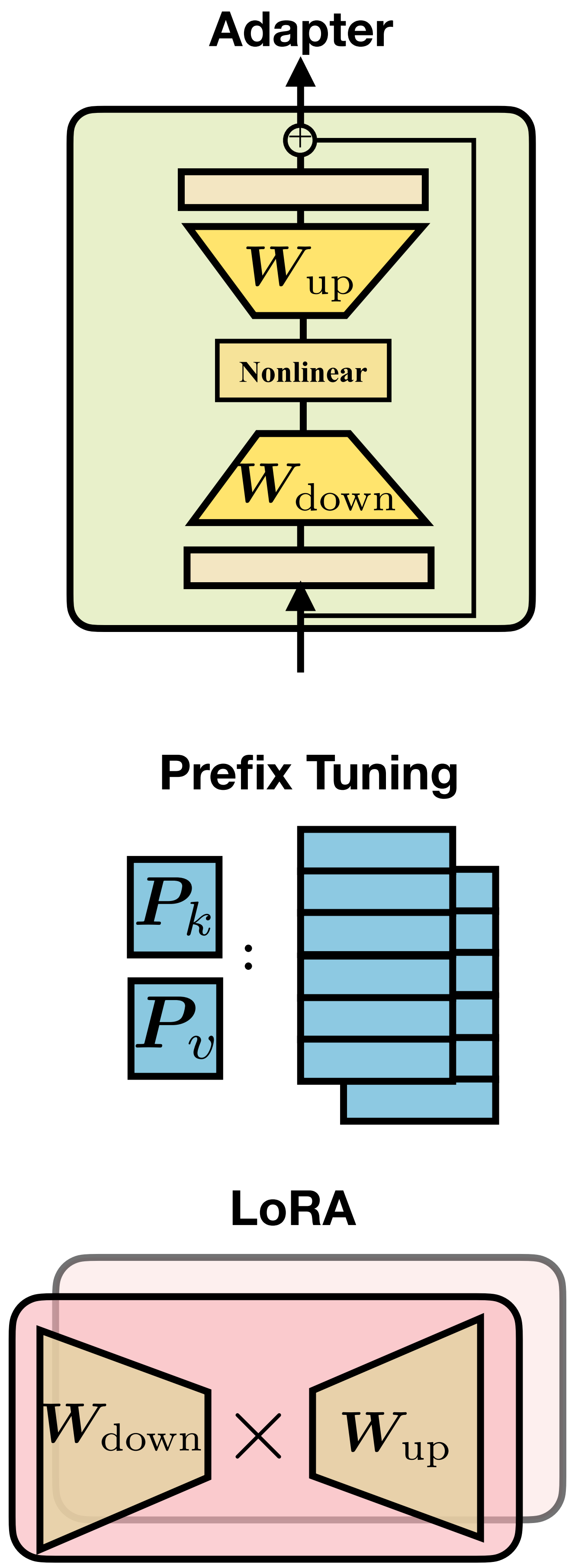
$$\text{head}_i = \text{Attn}(\boldsymbol{x}\boldsymbol{W}_q^{(i)}, \text{concat}(\boldsymbol{P}_k^{(i)}, \boldsymbol{C}\boldsymbol{W}_k^{(i)}), \text{concat}(\boldsymbol{P}_v^{(i)}, \boldsymbol{C}\boldsymbol{W}^{(i)}))$$

[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:

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

Prefix Tuning:

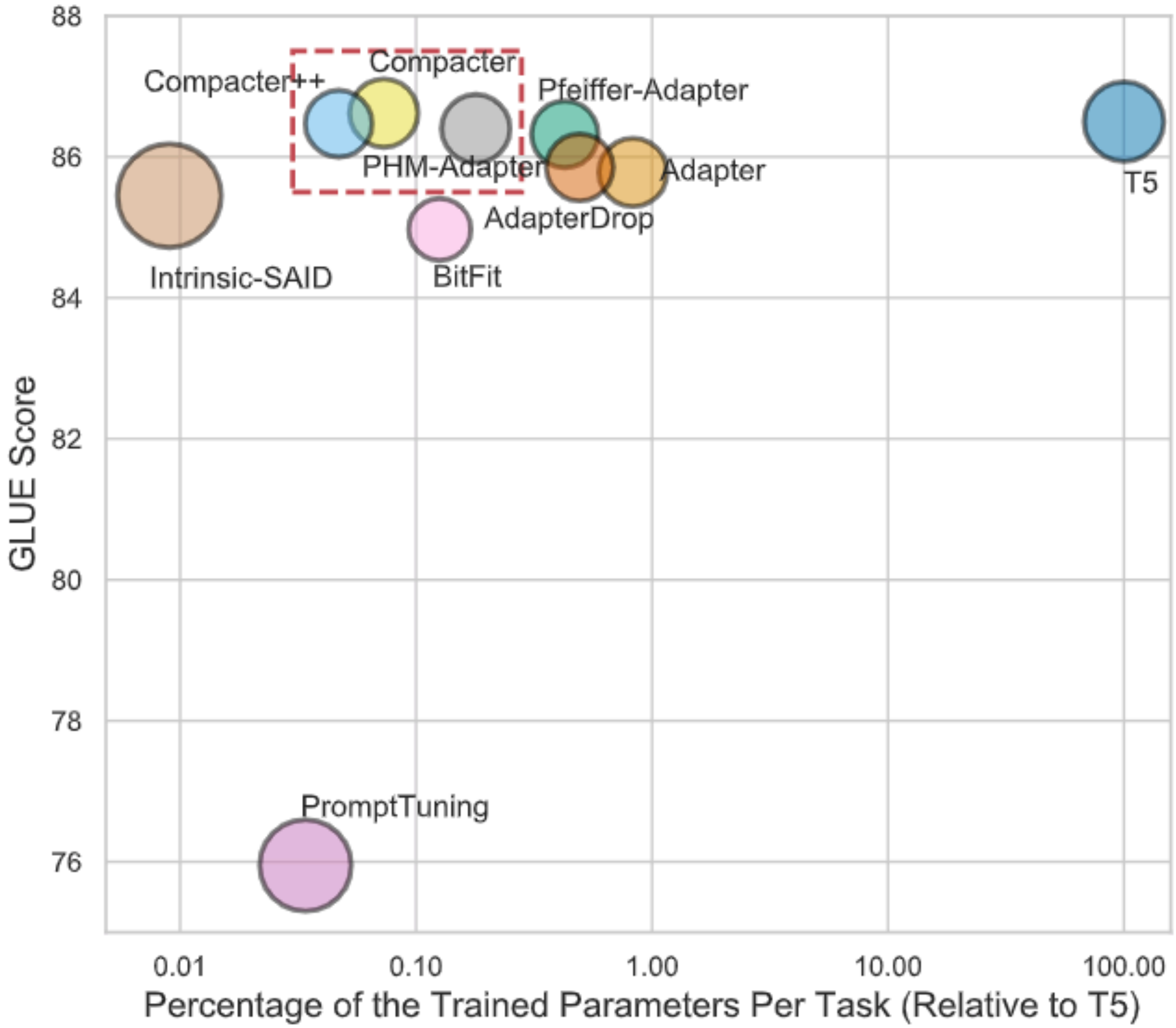
$$\text{head}_i = \text{Attn}(\boldsymbol{x}\boldsymbol{W}_q^{(i)}, \text{concat}(\boldsymbol{P}_k^{(i)}, \boldsymbol{C}\boldsymbol{W}_k^{(i)}), \text{concat}(\boldsymbol{P}_v^{(i)}, \boldsymbol{C}\boldsymbol{W}^{(i)}))$$

LoRA:

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

[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

Parameter-Efficient Fine-tuning

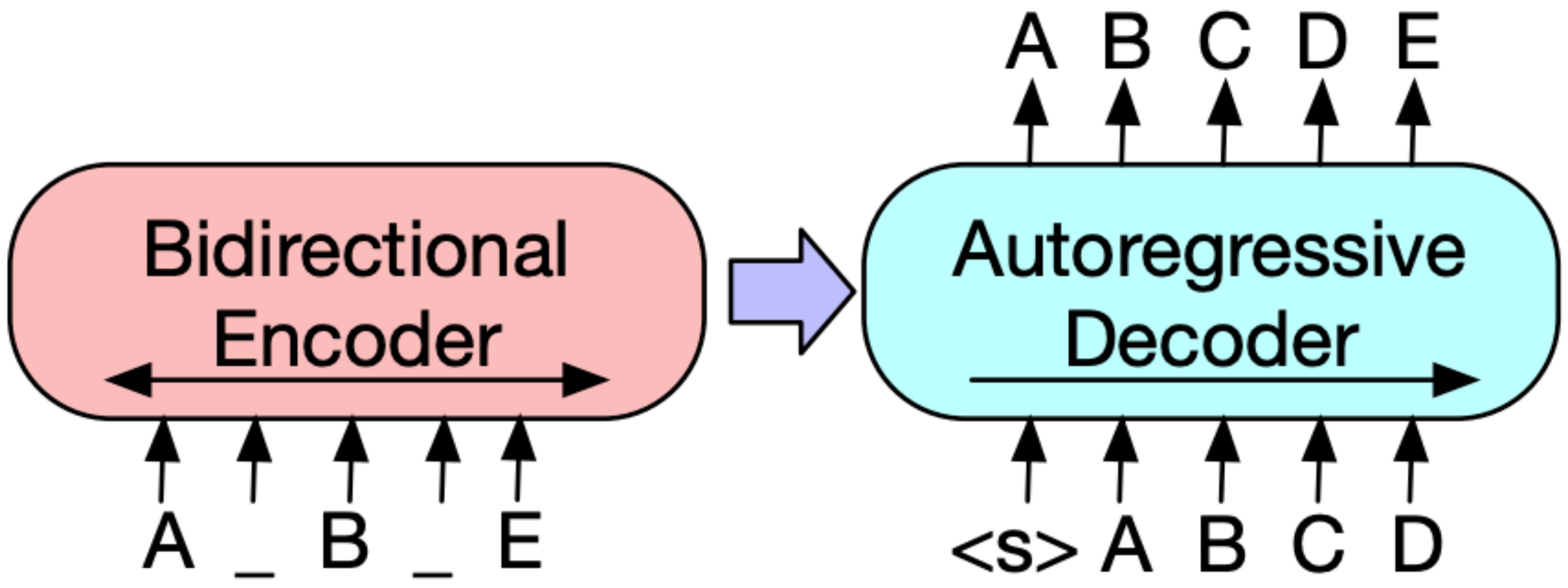


Less than 1% of parameters are tuned to achieve comparable performance to full fine-tuning (He et al., 2021)

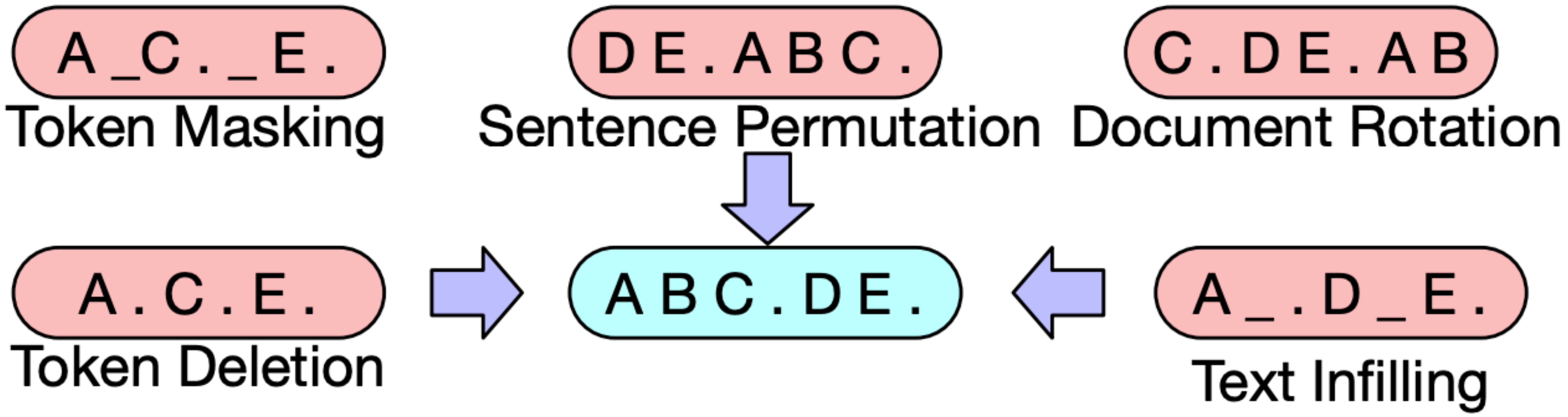
Neural Language Modeling: Pre-trained Decoders

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!



Reading Materials

- **Relavant Papers**
  - BERT
  - GPT-3
  - BART
  - Prompting
  - Parameter-Efficient Fine-tuning