Paper Review

Prompt-Based Learning

Myeongsup Kim

Master Student
Data Science & Business Analytics Lab.
School of Industrial Management Engineering
Korea University

Myeongsup_kim@korea.ac.kr

Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference

Schick and Schütze, 2021, EACL

It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners

Schick and Schütze, 2021, NAACL

Outstanding Long Paper Award at 2021 NAACL

GPT Understands, Too

Liu et al., 2021, arXiv

<What Are Not Covered in This Presentation>

- **Details of Transformer**

  Vaswani et al., Attention is All You Need, NIPS, 2017

- **Details of Transformer-Based Language Models (BERT, ALBERT, GPTs, …)**

  Devlin et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NAACL, 2019

  Lan et al., ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, ICLR, 2020

  Radford et al., Improving Language Understanding by Generative Pre-Training, 2018

  Radford et al., Language Models Are Unsupervised Multitask Learners, 2019

  Brown et al., Language Models Are Few-Shot Learners, NeurIPS, 2020

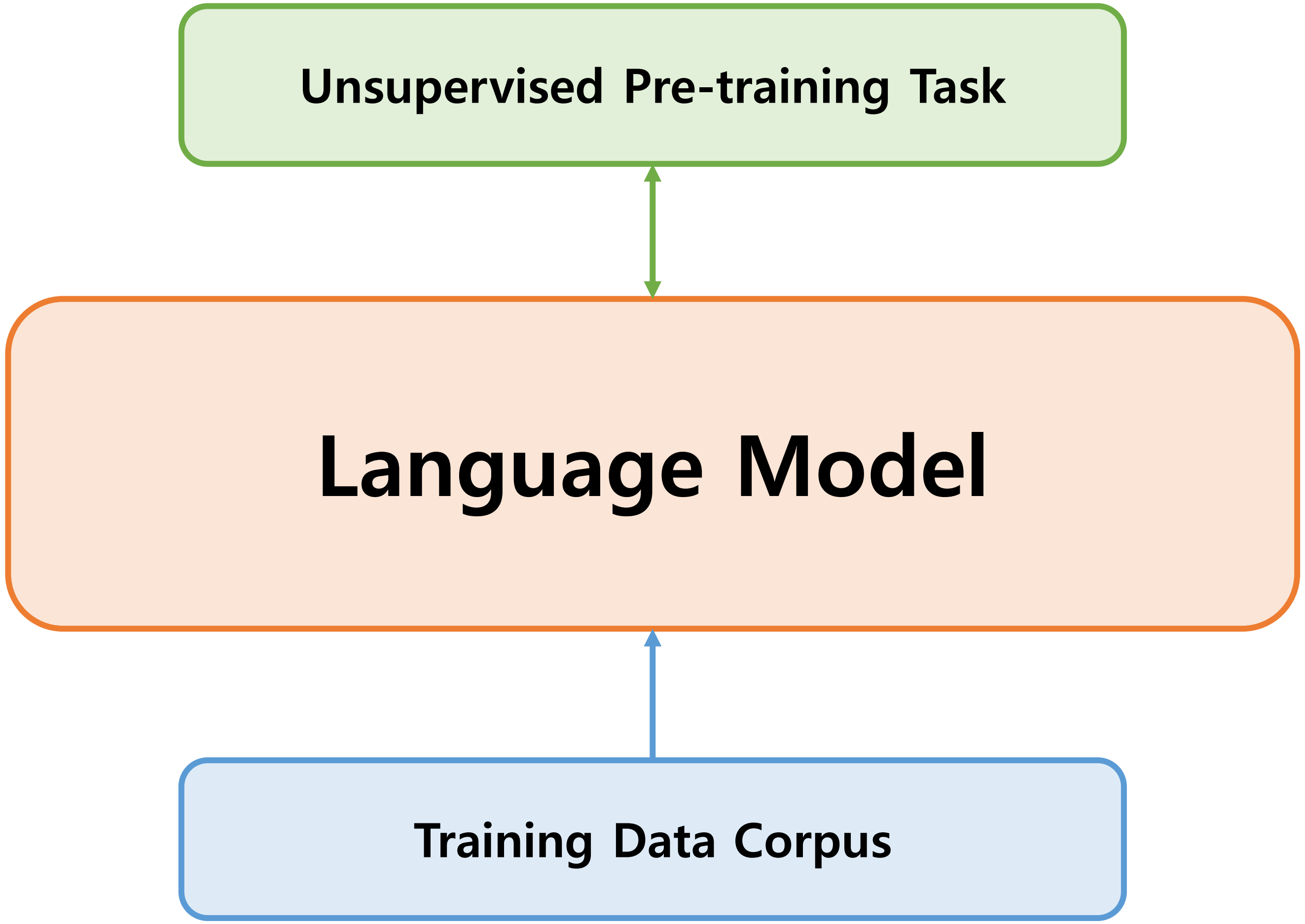
Pre-Requisites

- Hyper Scale Language Model
- Few Shot Learning for Language Model
- In-Context Learning

Introduction

-Hyper Scale Language Model

<Language Model>


Unsupervised Pre-training Task

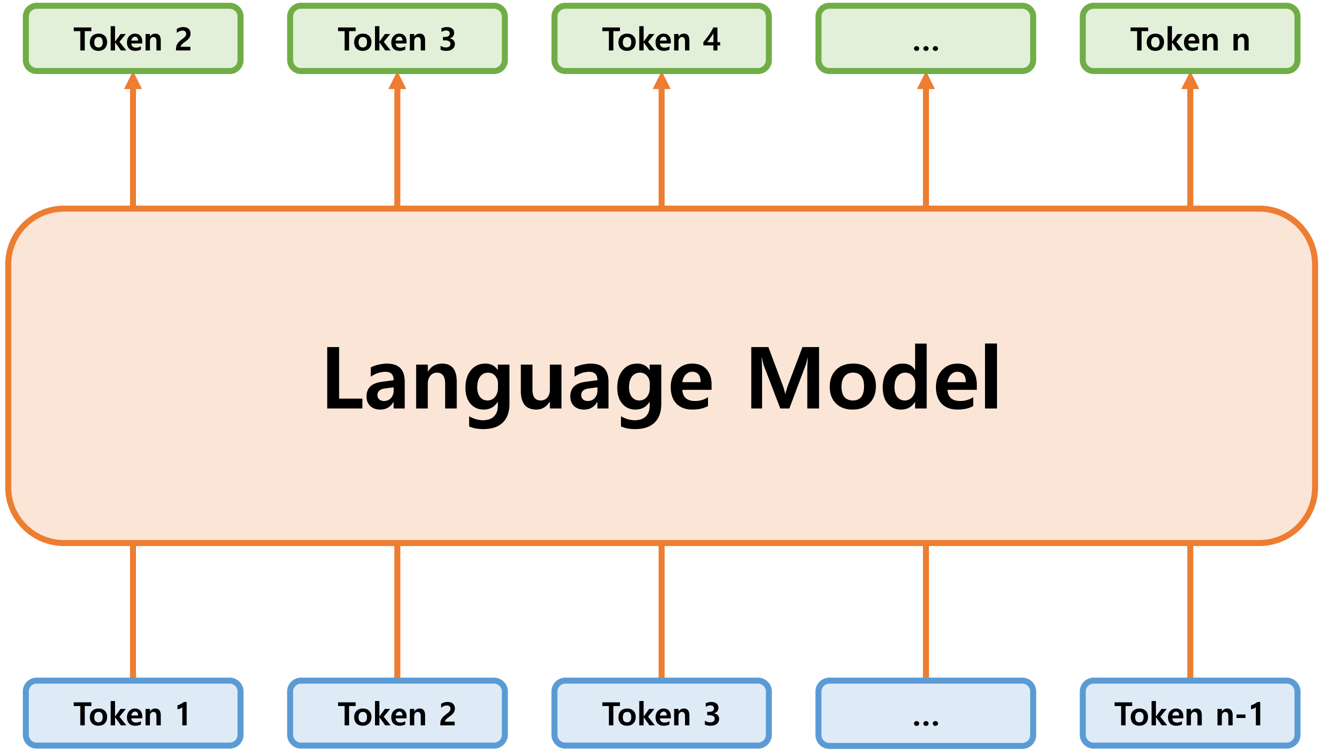
Language Model

Training Data Corpus

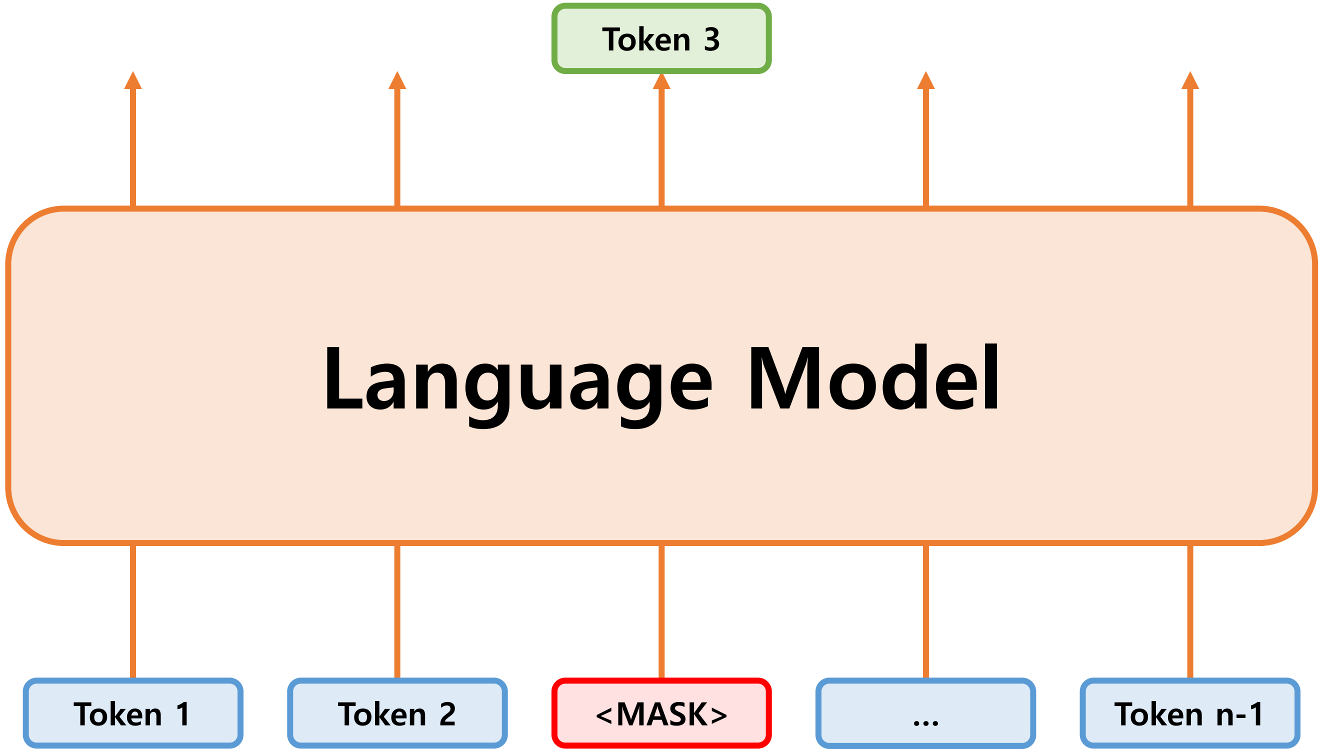
Introduction

-Hyper Scale Language Model

<Language Modeling>



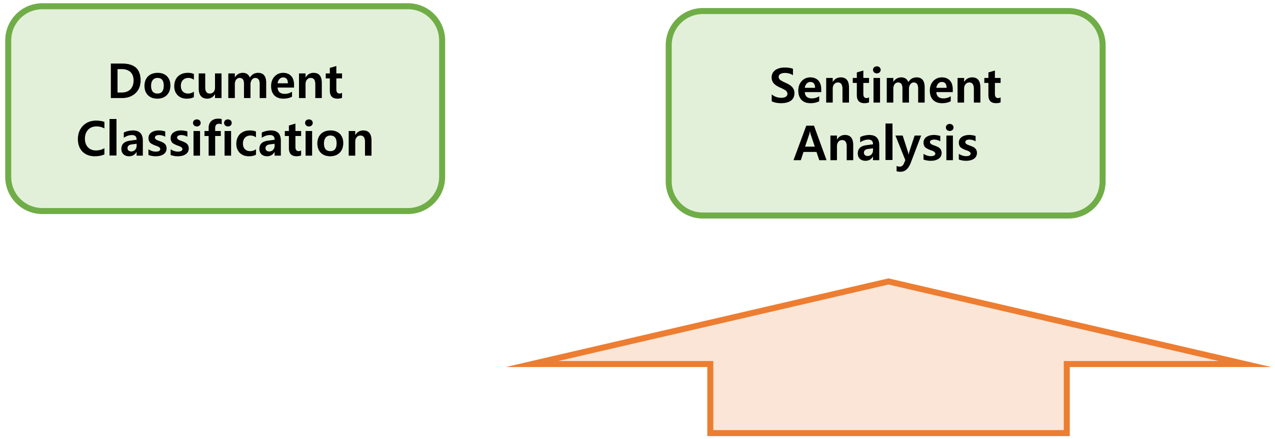
<Masked Language Modeling>


Token 3
Language Model
Token 1
Token 2
<MASK>
...
Token n-1

Introduction

-Hyper Scale Language Model

<Language Model>



Document Classification

Sentiment Analysis

...

ELMO

Embeddings from Language Model

BERT

Bidirectional Encoder Representation from Transformer

GPT

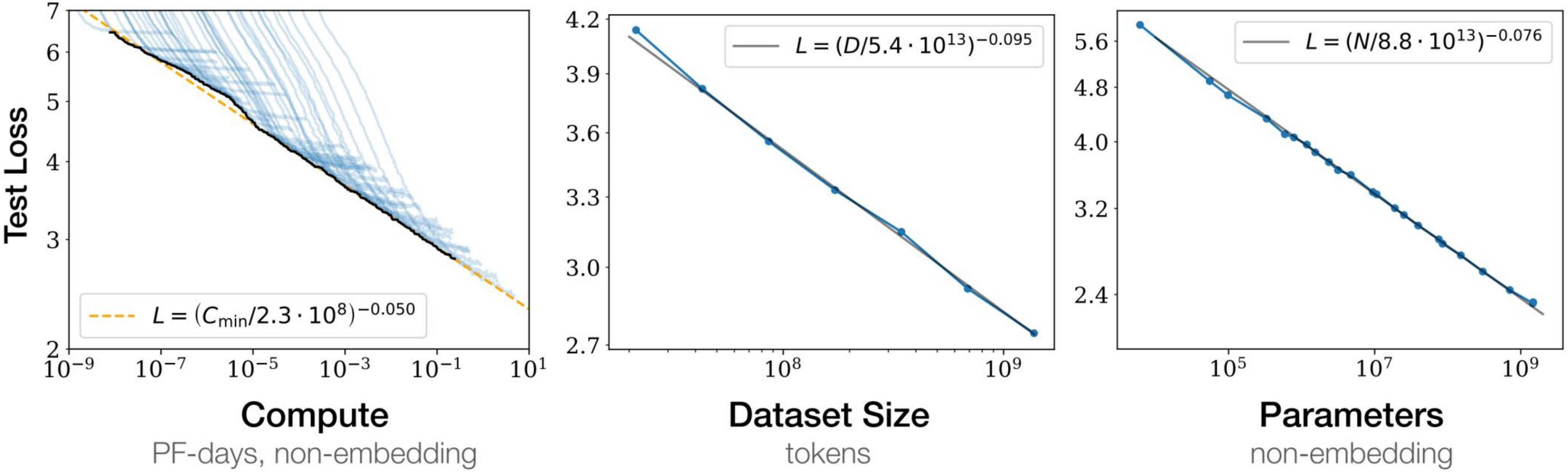
Generative Pre-trained Transformer

Kaplan et al., Scaling Laws for Neural Language Models, arXiv, 2020

Introduction

-Hyper Scale Language Model

<The Scaling Laws for LMs>

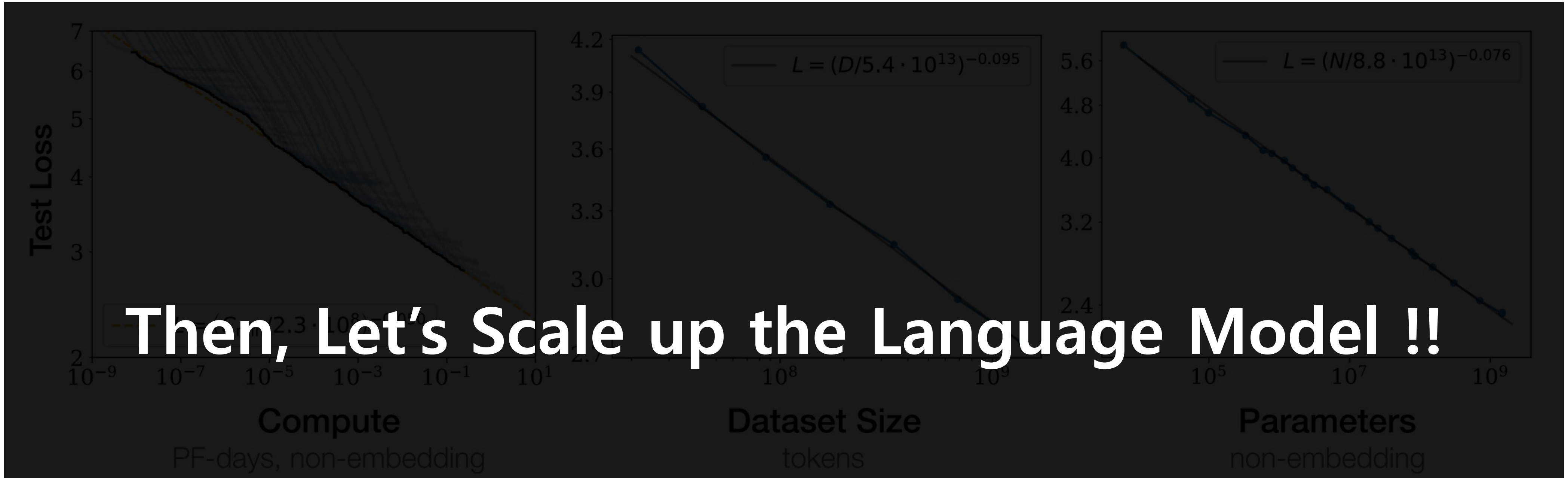


"**Language modeling performance improves** smoothly as **we increase <u>the model size</u>**, **<u>dataset size</u>**, and **<u>amount of compute</u> used for training**." (Kaplan et al., 2020)

Introduction

-Hyper Scale Language Model

<The Scaling Laws for LMs>

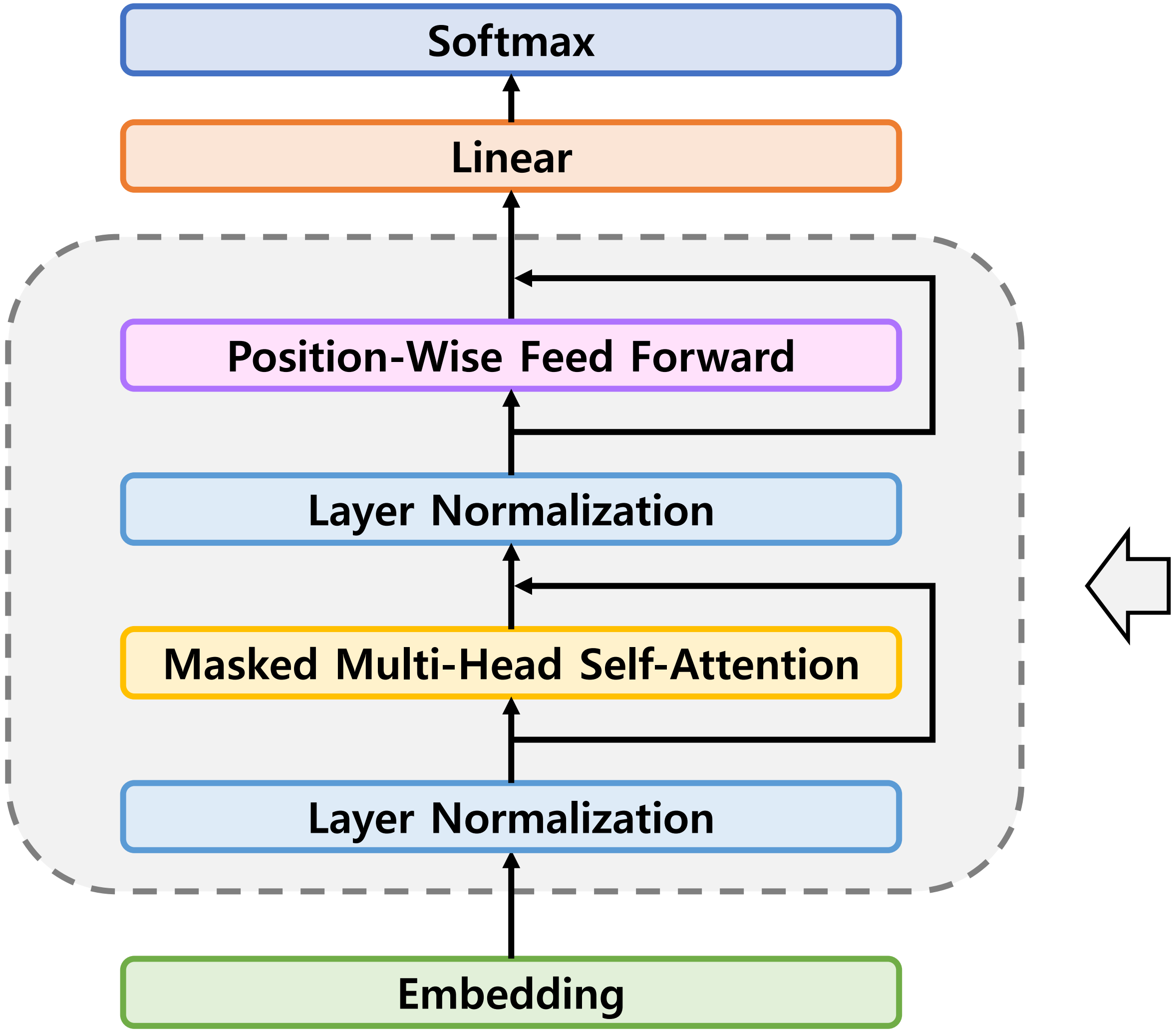


Then, Let’s Scale up the Language Model !!

“**Language modeling performance improves** smoothly as **we increase <u>the model size</u>**, **<u>dataset size</u>**, and **<u>amount of compute</u> used for training**.” (Kaplan et al., 2020)

Brown et al., Language Models Are Few-Shot Learners, NeurIPS, 2020

<Generative Pre-trained Transformer-3>



$\boldsymbol{n_{params} = 175B}$

$n_{layers} = 96$

$d_{model} = 12{,}288$

$n_{heads} = 96$

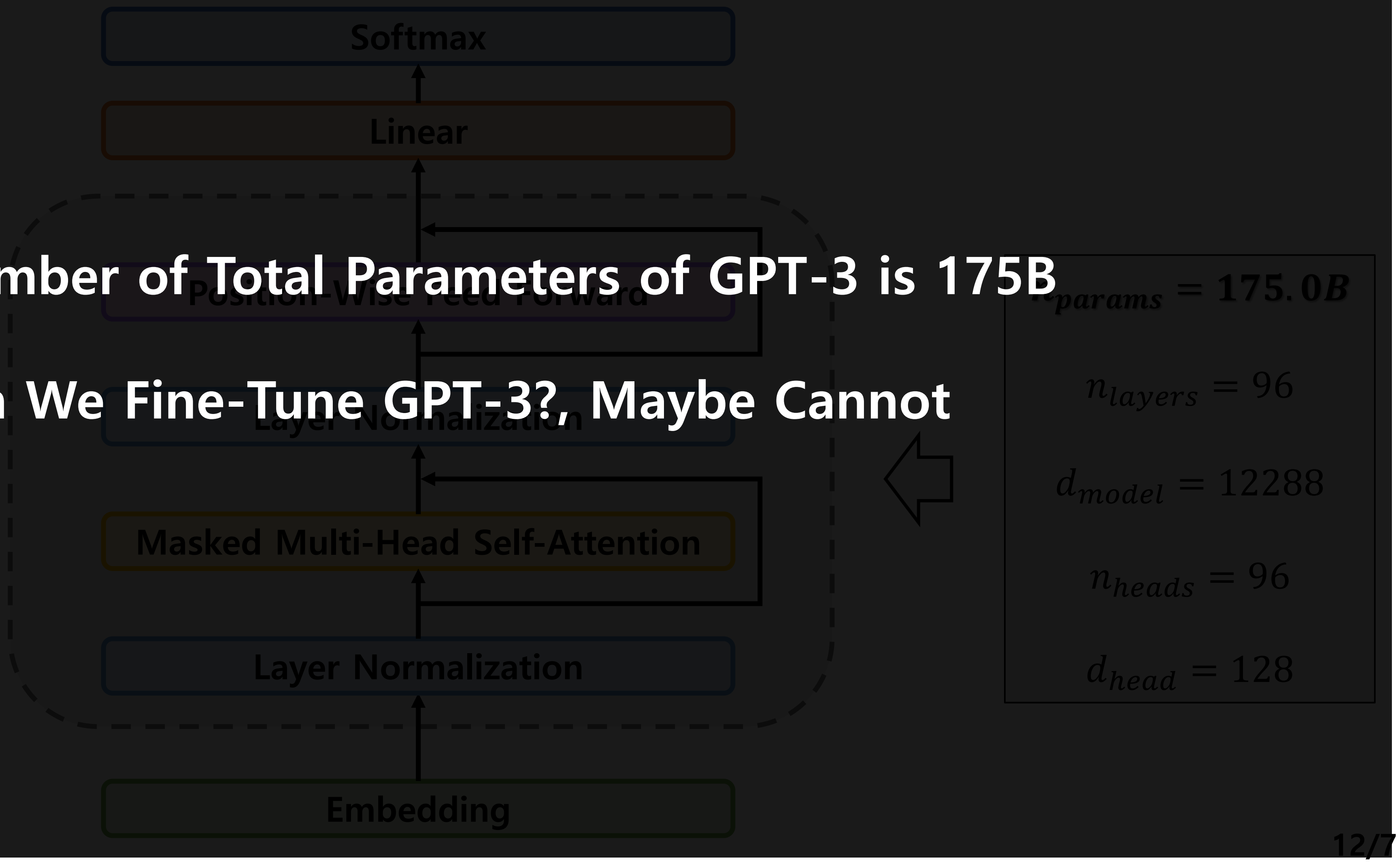
$d_{head} = 128$

Brown et al., Language Models Are Few-Shot Learners, NeurIPS, 2020

Introduction

-Hyper Scale Language Model

<Generative Pre-trained Transformer-3>



The Number of Total Parameters of GPT-3 is 175B

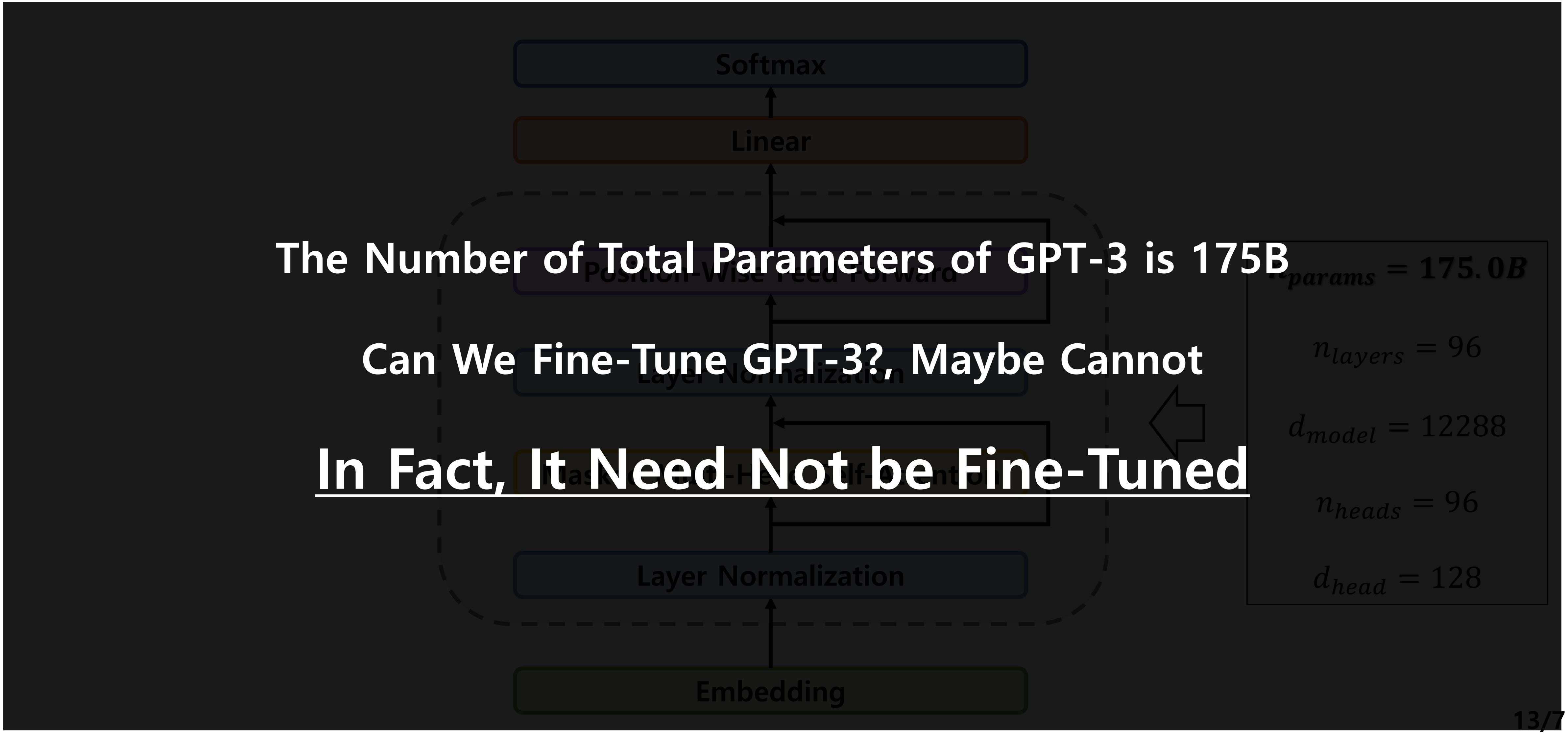
Can We Fine-Tune GPT-3?, Maybe Cannot

Brown et al., Language Models Are Few-Shot Learners, NeurIPS, 2020

Introduction

-Hyper Scale Language Model

<Generative Pre-trained Transformer-3>



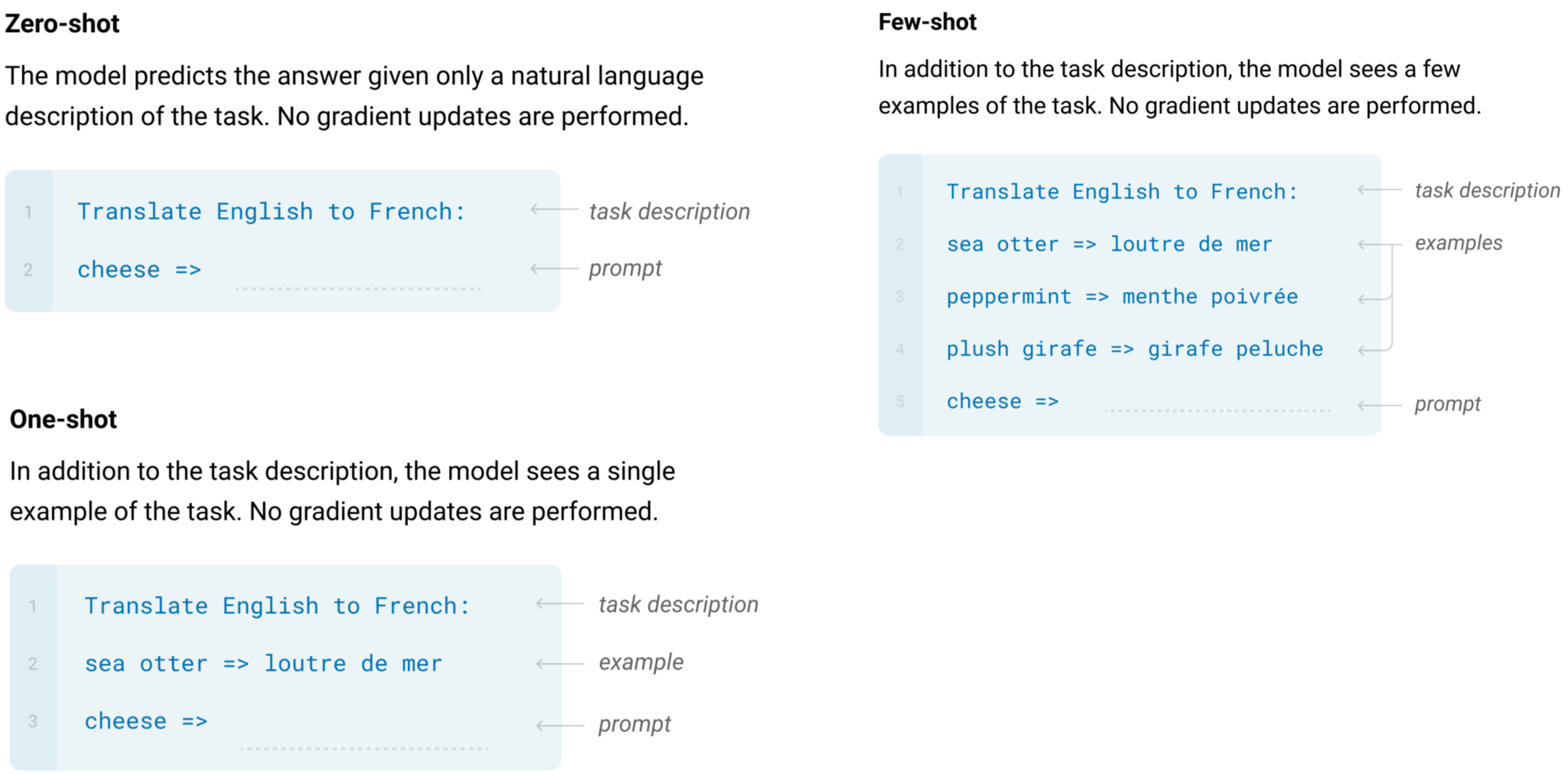
The Number of Total Parameters of GPT-3 is 175B

Can We Fine-Tune GPT-3?, Maybe Cannot

In Fact, It Need Not be Fine-Tuned

Brown et al., Language Models Are Few-Shot Learners, NeurIPS, 2020

<Few Shot Learning for LM>



Zero-shot

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

```
Translate English to French:
cheese => ..............................
```

One-shot

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

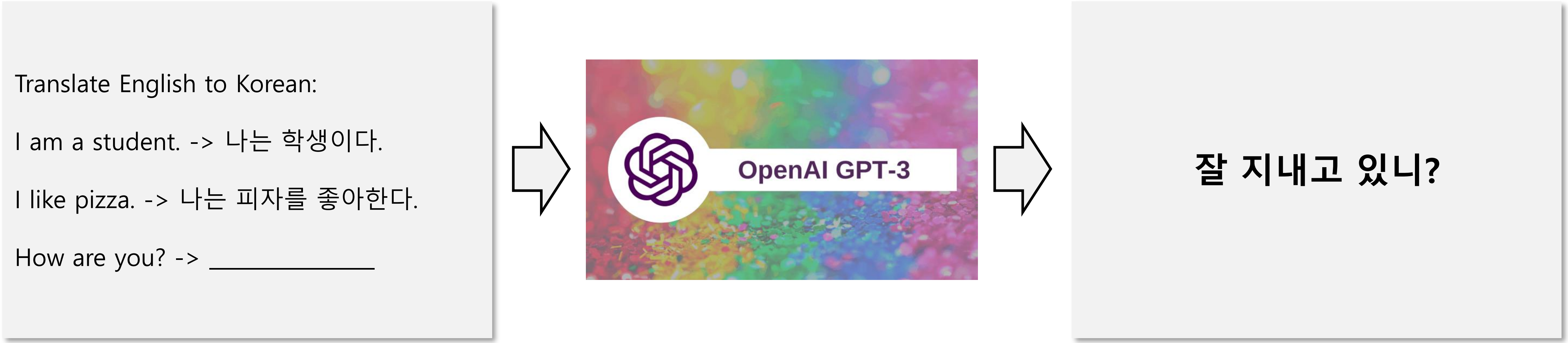
```
Translate English to French:
sea otter => loutre de mer
cheese => ..............................
```

Few-shot

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

```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese => ..............................
```

<Few Shot Learning for LM>



Translate English to Korean:

I am a student. -> 나는 학생이다.

I like pizza. -> 나는 피자를 좋아한다.

How are you? -> ______________

잘 지내고 있니?

Introduction

-Few Shot Learning for LM

<Few Shot Learning for LM>

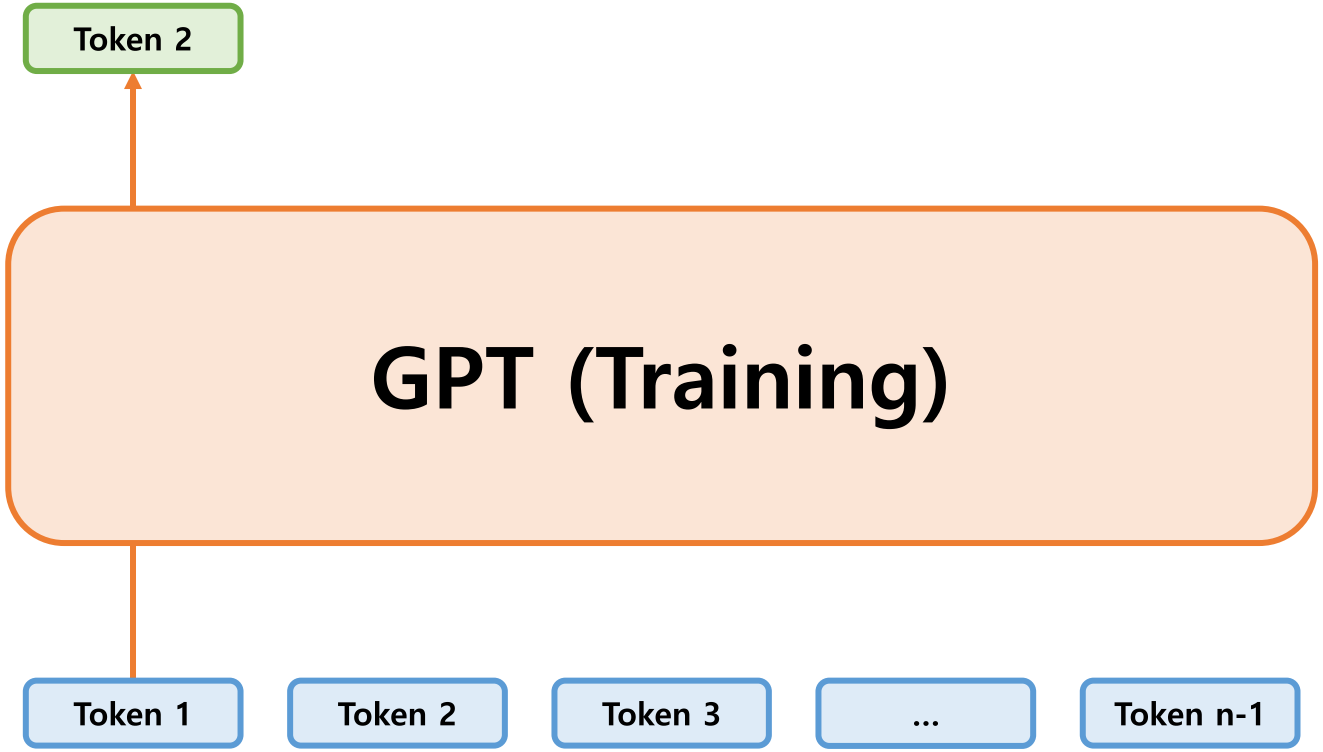
Answer the question:

Where is the capital of UK? -> London

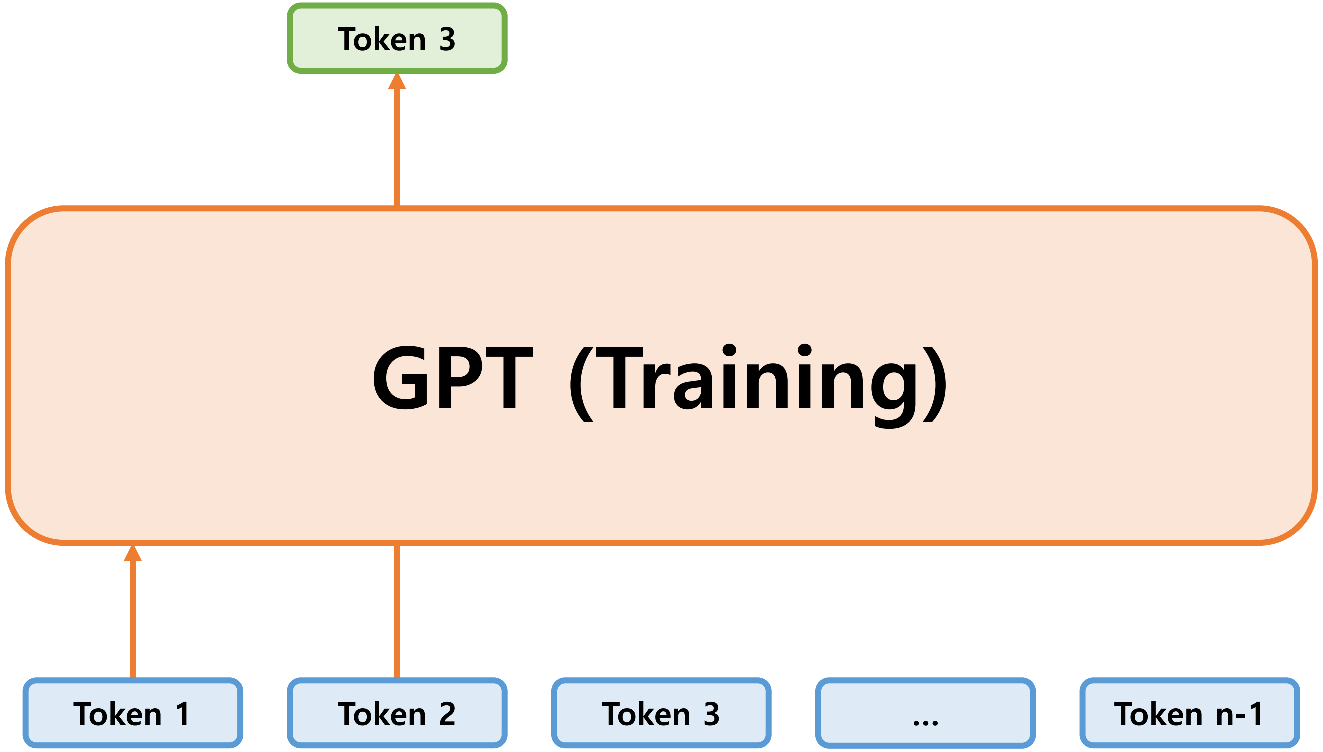
Who founded Apple? -> __________

Steve Jobs

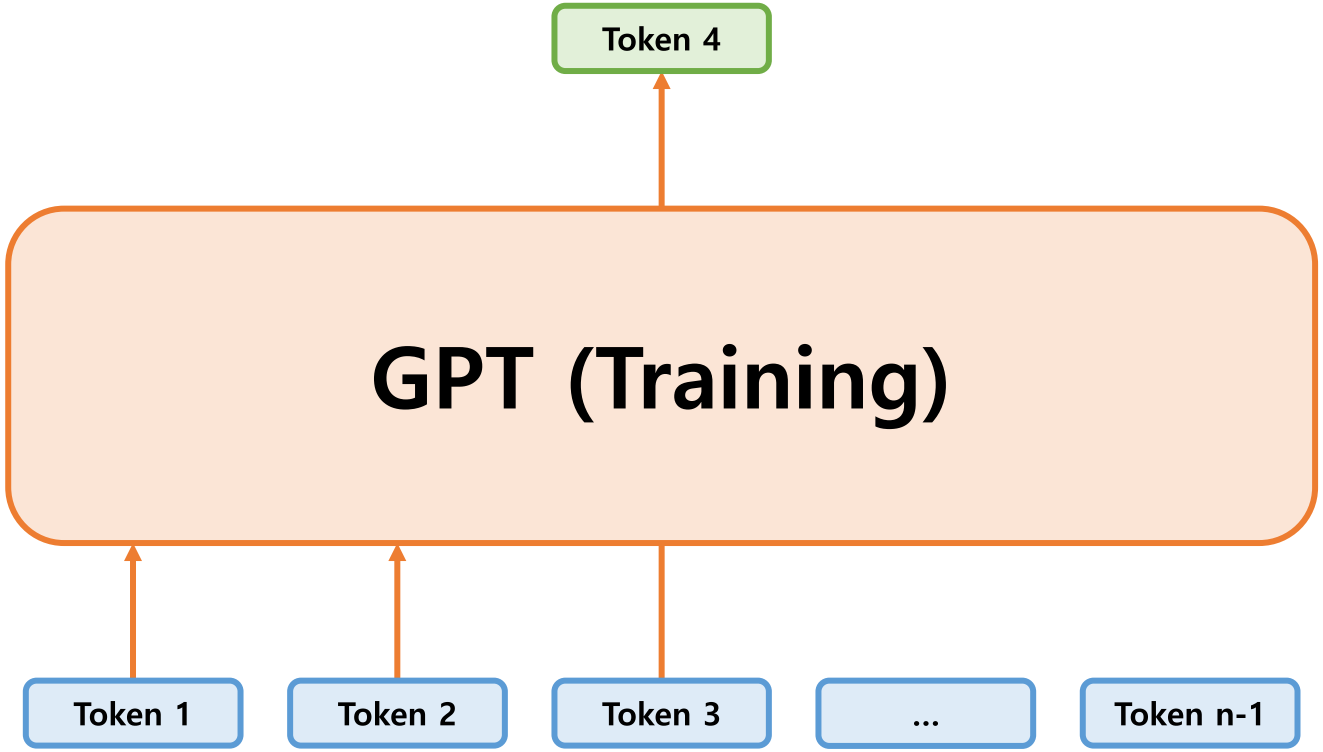
<In-Context Learning>


Token 2
GPT (Training)
Token 1
Token 2
Token 3
...
Token n-1

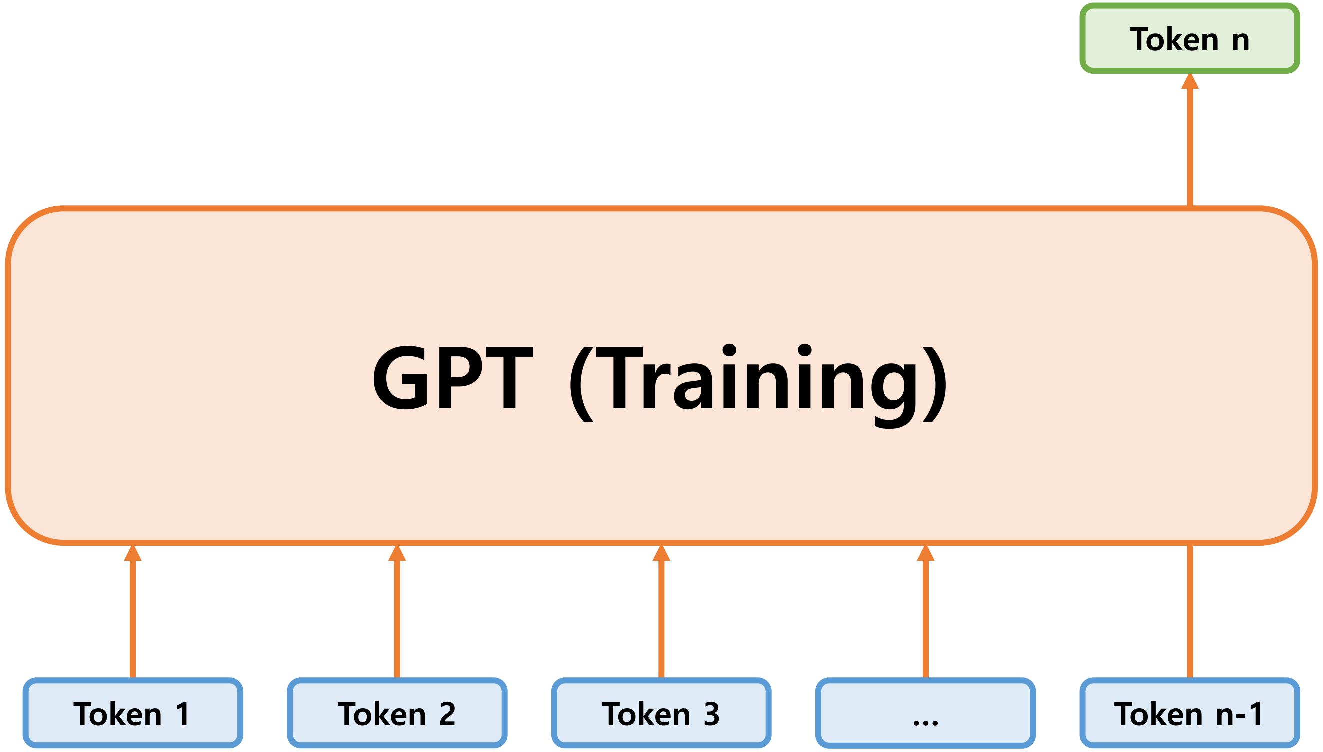
<In-Context Learning>


Token 3
GPT (Training)
Token 1
Token 2
Token 3
...
Token n-1

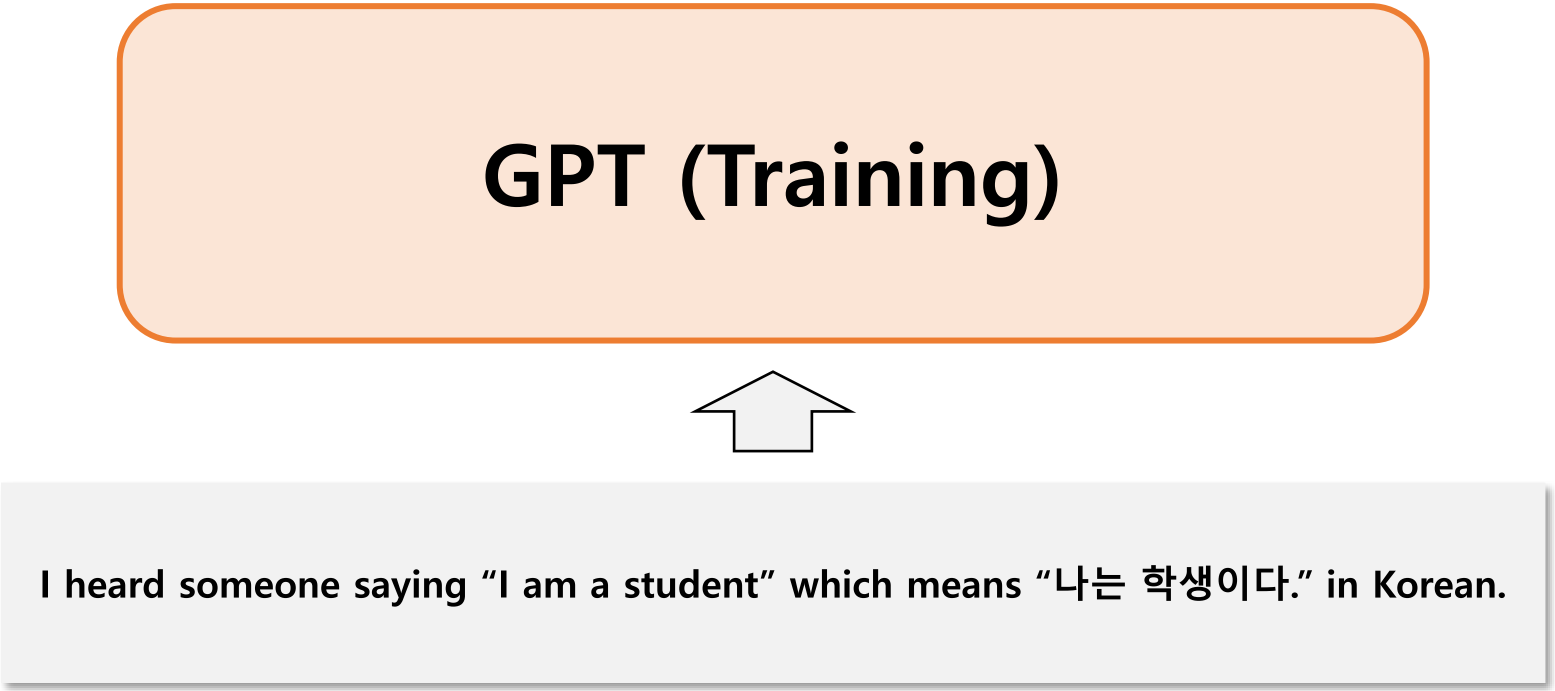
<In-Context Learning>


Token 4
GPT (Training)
Token 1
Token 2
Token 3
...
Token n-1

<In-Context Learning>


Token n
GPT (Training)
Token 1
Token 2
Token 3
...
Token n-1

<In-Context Learning>



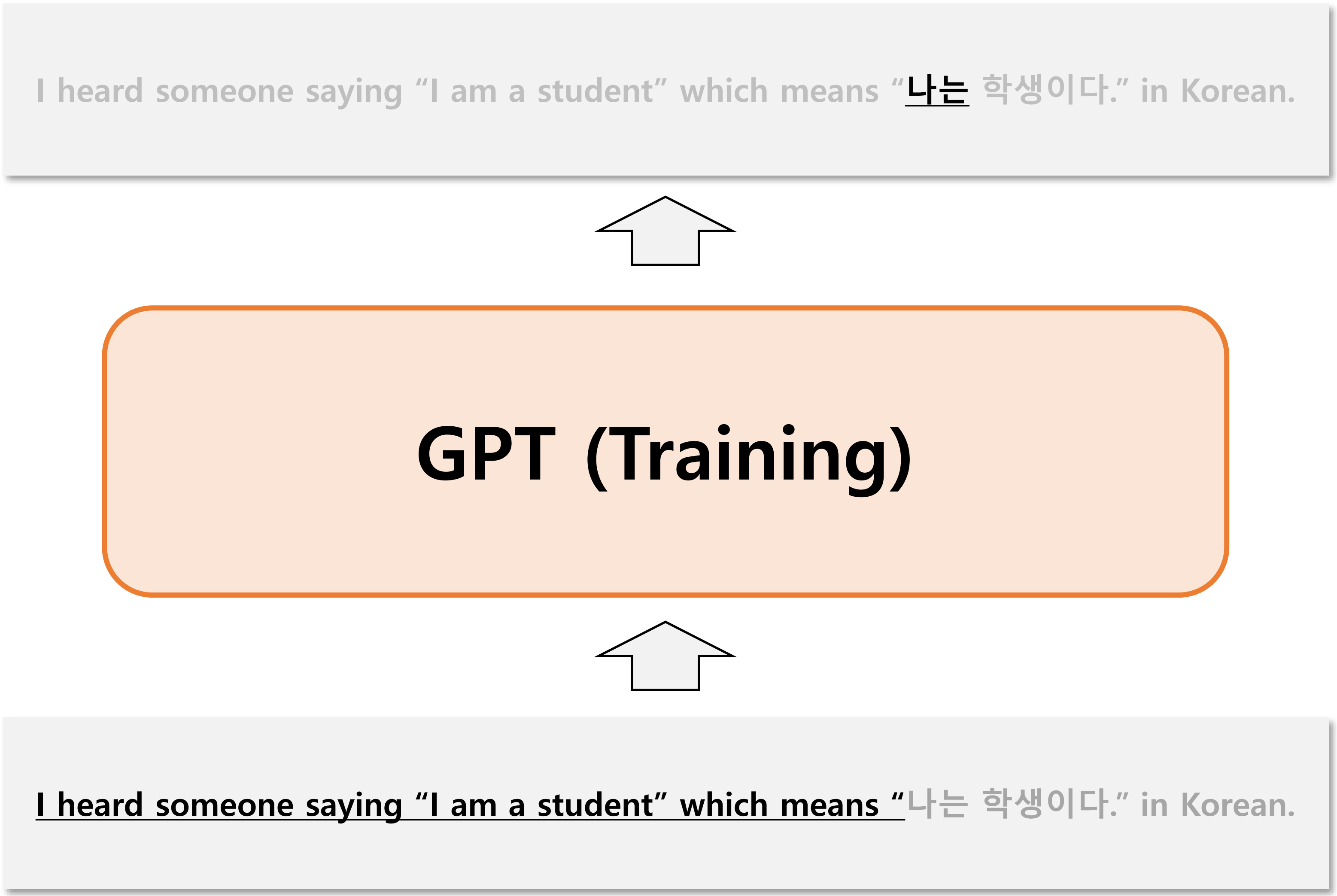
GPT (Training)

I heard someone saying “I am a student” which means “나는 학생이다.” in Korean.

Introduction

-In-Context Learning

<In-Context Learning>



I heard someone saying “I am a student” which means “나는 학생이다.” in Korean.

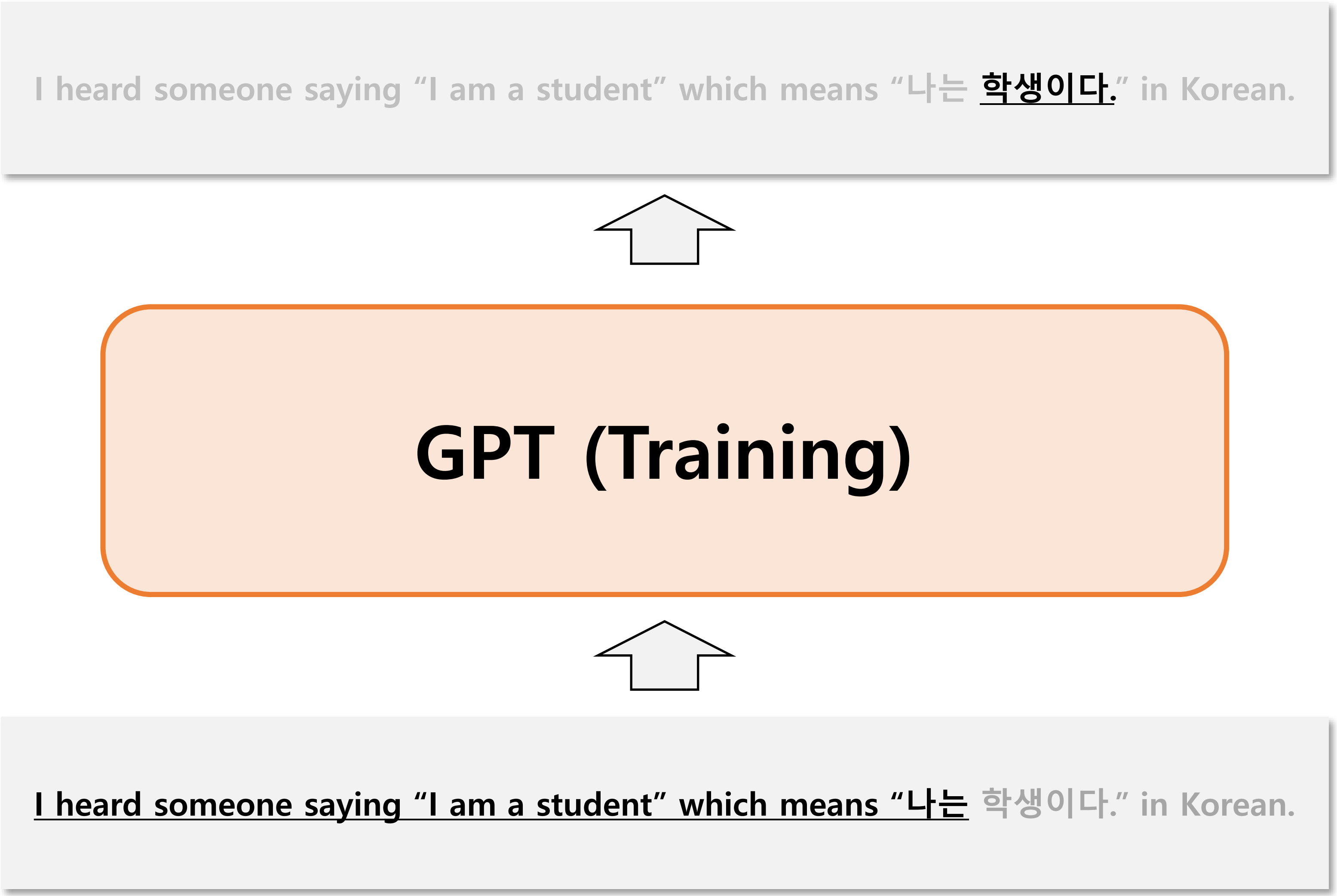
GPT (Training)

I heard someone saying “I am a student” which means “나는 학생이다.” in Korean.

Introduction

-In-Context Learning

<In-Context Learning>



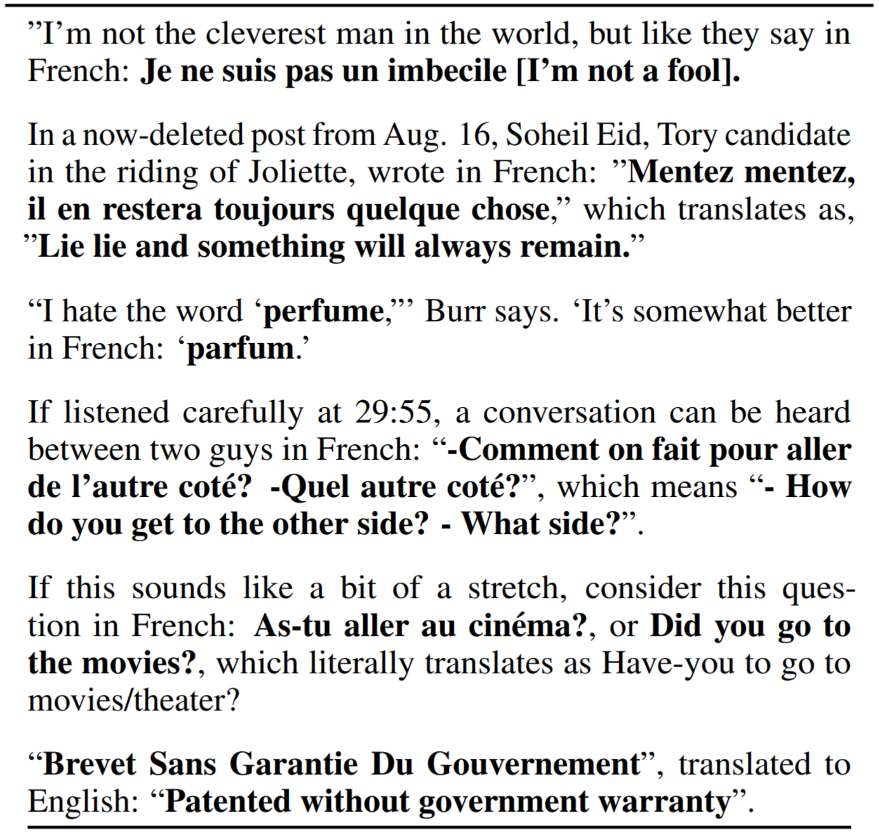
I heard someone saying “I am a student” which means “나는 학생이다.” in Korean.

GPT (Training)

I heard someone saying “I am a student” which means “나는 학생이다.” in Korean.

Radford et al., Language Models Are Unsupervised Multitask Learners, 2019

<In-Context Learning>



”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: ”**Mentez mentez, il en restera toujours quelque chose**,” which translates as, ”**Lie lie and something will always remain.**”

“I hate the word ‘**perfume**,”’ Burr says. ‘It’s somewhat better in French: ‘**parfum**.’

If listened carefully at 29:55, a conversation can be heard between two guys in French: “**-Comment on fait pour aller de l’autre coté? -Quel autre coté?**”, which means “**- How do you get to the other side? - What side?**”.

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

“**Brevet Sans Garantie Du Gouvernement**”, translated to English: “**Patented without government warranty**”.

“Examples of **naturally occurring demonstrations of English to French and French to English translation** found throughout the WebText training set.” (Radford et al., 2019)

Brown et al., Language Models Are Few Shot Learners, NeurIPS, 2020

Introduction

-In-Context Learning

<In-Context Few Shot Learning>

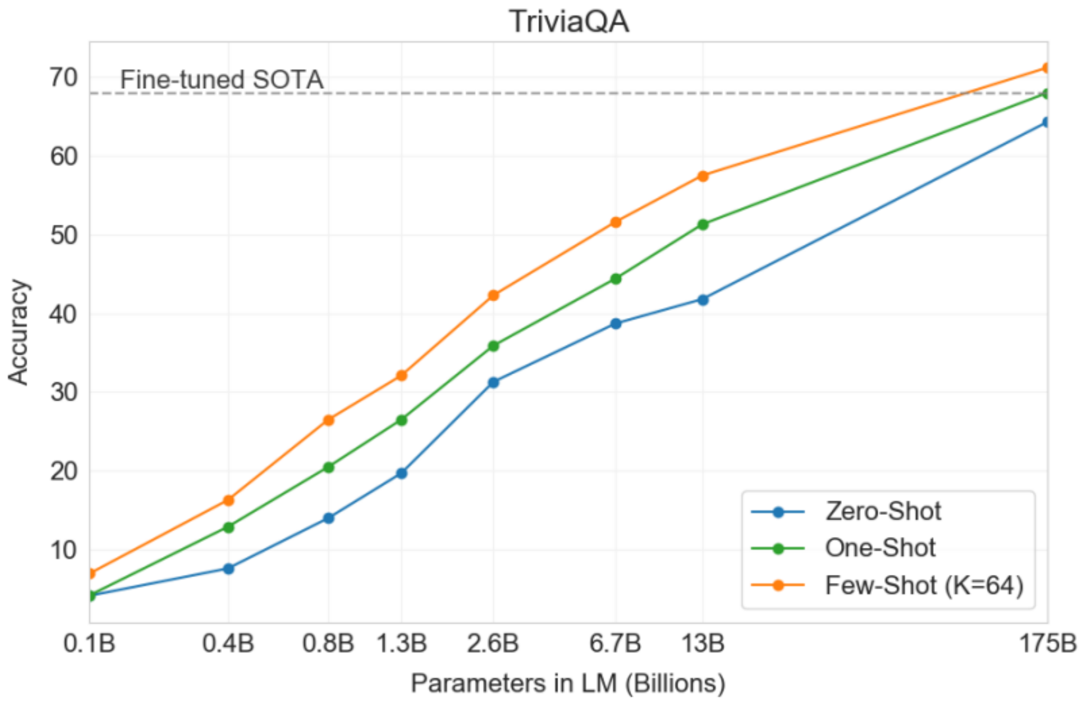


Figure 3.3: On TriviaQA GPT3's performance grows smoothly with model size, suggesting that language models continue to absorb knowledge as their capacity increases. One-shot and few-shot performance make significant gains over zero-shot behavior, matching and exceeding the performance of the SOTA fine-tuned open-domain model, RAG [LPP+20]

Introduction

-In-Context Learning

Brown et al., Language Models Are Few Shot Learners, NeurIPS, 2020

<In-Context Few Shot Learning>

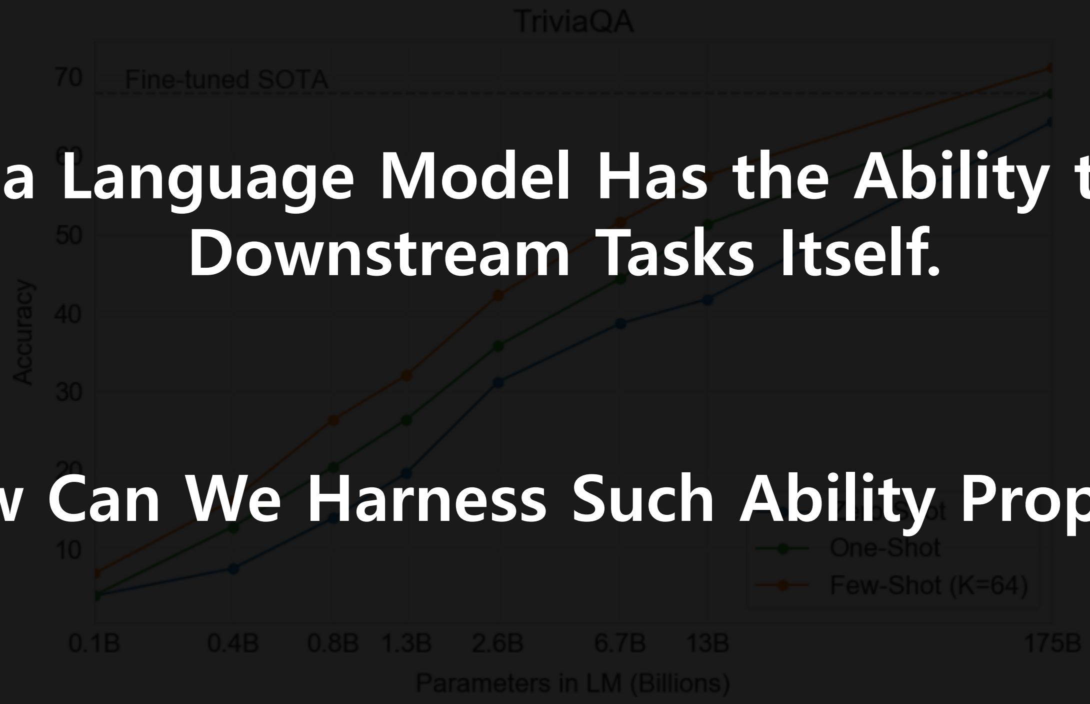


Figure 3.3: On TriviaQA GPT3's performance grows smoothly with model size, suggesting that language models continue to absorb knowledge as their capacity increases. One-shot and few-shot performance make significant gains over zero-shot behavior, matching and exceeding the performance of the SOTA fine-tuned open-domain model, RAG [LPP$^+$20]

Pre-training of a Language Model Has the Ability to Perform Many Downstream Tasks Itself.

How Can We Harness Such Ability Properly?

Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference

Schick and Schütze, 2021, EACL

It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners

Schick and Schütze, 2021, NAACL

Outstanding Long Paper Award at 2021 NAACL

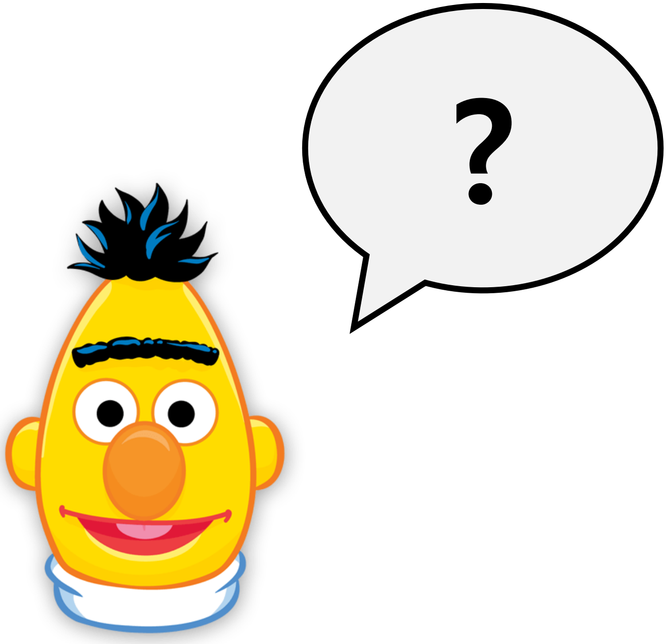
Pattern-Exploiting Training

- Pattern-Verbalizer Pair
- PVP Training and Inference
- PET with Multiple Masks

<Pattern-Verbalizer Pair>

Question Answering

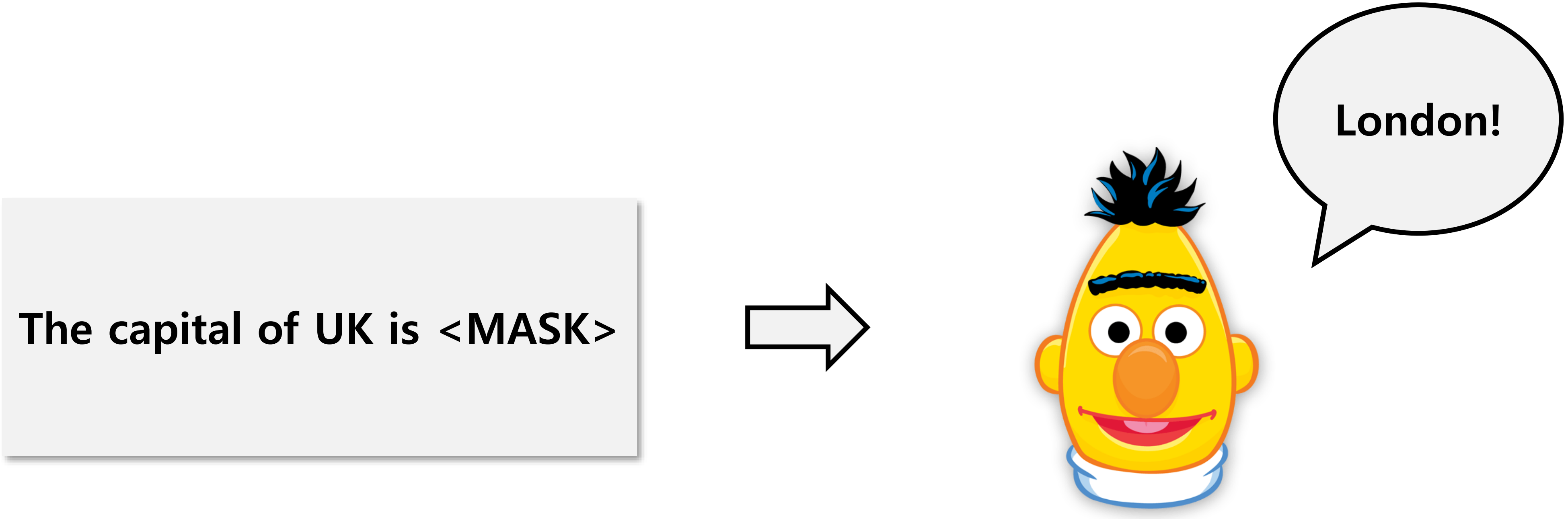
Where is the capital of UK?



Pattern-Exploiting Training

-Pattern-Verbalizer Pair

<Pattern-Verbalizer Pair>



The capital of UK is <MASK>

London!

Pattern-Exploiting Training

-Pattern-Verbalizer Pair

<Pattern-Verbalizer Pair>

Question Answering

Where is the capital of UK? → The capital of UK is <MASK>

Sentiment Classification

My phone doesn't work → My phone doesn't work.
So, I feel <MASK>

Pattern-Exploiting Training

-Pattern-Verbalizer Pair

<Pattern-Verbalizer Pair>

<Notation>

$M: Masked\ Language\ Model$

$V: Vocabulary$

$\_\_\_\_ \in V\ :\ Mask\ Token$

$\mathcal{L}, A: Labels, Target\ Classification\ Task$

$\mathbf{x} = (s_1, \ldots, s_k):\ Input\ for\ Task\ A$

$s_i \in V^*: Phrase$

$\boldsymbol{P}: \boldsymbol{Pattern}, where\ P(\mathbf{x}) \in V^*$

$\boldsymbol{v}: \boldsymbol{Verbalizer}, \mathcal{L} \rightarrow V$

$(\boldsymbol{P}, \boldsymbol{v}): \boldsymbol{Pattern}\text{ - }\boldsymbol{Verbalizer\ Pair}\ (\mathbf{PVP})$

<Pattern-Verbalizer Pair>

<Notation>

$M: Masked\ Language\ Model$

$V: Vocabulary$

$\_\_\_\_ \in V : Mask\ Token$

$\mathcal{L}, A: Labels, Target\ Classification\ Task$

$\mathbf{x} = (s_1, \ldots, s_k):\ Input\ for\ Task\ A$

$s_i \in V^*: Phrase$

$\boldsymbol{P}: \boldsymbol{Pattern}, where\ P(\mathbf{x}) \in V^*$

$\boldsymbol{v}: \boldsymbol{Verbalizer}, \mathcal{L} \rightarrow V$

$(\boldsymbol{P}, \boldsymbol{v}): \boldsymbol{Pattern\text{-}Verbalizer\ Pair}\ (\mathbf{PVP})$

Task: RTE **(Recognizing Textual Entailment)**

Sen1: Oil prices rise.

Sen2: Oil prices fall back.

Label: Not Entailed

Pattern-Exploiting Training

-Pattern-Verbalizer Pair

<Pattern-Verbalizer Pair>

<Notation>

$M: Masked\ Language\ Model$

$V: Vocabulary$

$\underline{\quad\quad} \in V\ :\ Mask\ Token$

$\mathcal{L}, \boldsymbol{A}: Labels, \boldsymbol{Target\ Classification\ Task}$

$\mathbf{x} = (s_1, \ldots, s_k):\ Input\ for\ Task\ A$

$s_i \in V^*: Phrase$

$\boldsymbol{P}: \boldsymbol{Pattern}, where\ P(\mathbf{x}) \in V^*$

$\boldsymbol{v}: \boldsymbol{Verbalizer}, \mathcal{L} \rightarrow V$

$(\boldsymbol{P}, \boldsymbol{v}): \boldsymbol{Pattern\text{-}Verbalizer\ Pair}\ (\mathbf{PVP})$

Task: RTE **(Recognizing Textual Entailment)**

Sen1: Oil prices rise.

Sen2: Oil prices fall back.

Label: Not Entailed

Pattern-Exploiting Training

-Pattern-Verbalizer Pair

<Pattern-Verbalizer Pair>

<Notation>

$M: Masked\ Language\ Model$

$V: Vocabulary$

$\_\_\_ \in V : Mask\ Token$

$\mathcal{L}, \boldsymbol{A}: Labels, \boldsymbol{Target\ Classification\ Task}$

$\mathbf{x} = (\boldsymbol{s_1}, \dots, \boldsymbol{s_k}): Input\ for\ \boldsymbol{Task\ A}$

$\boldsymbol{s_i} \in \boldsymbol{V^*}: \boldsymbol{Phrase}$

$\boldsymbol{P}: \boldsymbol{Pattern}, where\ P(\mathbf{x}) \in V^*$

$\boldsymbol{v}: \boldsymbol{Verbalizer}, \mathcal{L} \to V$

$(\boldsymbol{P}, \boldsymbol{v}): \boldsymbol{Pattern\text{ - }Verbalizer\ Pair}\ (\mathbf{PVP})$

Task: RTE **(Recognizing Textual Entailment)**

Sen1: Oil prices rise.

Sen2: Oil prices fall back.

Label: Not Entailed

<Pattern-Verbalizer Pair>

<Notation>

$M: Masked\ Language\ Model$

$V: Vocabulary$

$\underline{\quad\quad} \in V\ :\ Mask\ Token$

$\mathcal{L}, A: \boldsymbol{Labels}, \boldsymbol{Target\ Classification\ Task}$

$\mathbf{x} = (\boldsymbol{s_1}, \ldots, \boldsymbol{s_k}): Input\ for\ \boldsymbol{Task\ A}$

$\boldsymbol{s_i} \in \boldsymbol{V^*}: \boldsymbol{Phrase}$

$\boldsymbol{P}: \boldsymbol{Pattern}, where\ P(\mathbf{x}) \in V^*$

$\boldsymbol{v}: \boldsymbol{Verbalizer}, \mathcal{L} \rightarrow V$

$(\boldsymbol{P}, \boldsymbol{v}): \boldsymbol{Pattern\text{ - }Verbalizer\ Pair}\ (\mathrm{PVP})$

Task: RTE **(Recognizing Textual Entailment)**

Sen1: **Oil prices rise.**

Sen2: **Oil prices fall back.**

Label: Not Entailed

Pattern-Exploiting Training

-Pattern-Verbalizer Pair

<Pattern-Verbalizer Pair>

<Notation>

$M: Masked\ Language\ Model$

$V: Vocabulary$

$\_\_\_ \in V\ :\ Mask\ Token$

$\boldsymbol{\mathcal{L}}, \boldsymbol{A}: \boldsymbol{Labels}, \boldsymbol{Target\ Classification\ Task}$

$\mathbf{x} = (\boldsymbol{s}_{\mathbf{1}}, \ldots, \boldsymbol{s}_{\boldsymbol{k}}): Input\ for\ \boldsymbol{Task\ A}$

$\boldsymbol{s}_{\boldsymbol{i}} \in \boldsymbol{V}^*: \boldsymbol{Phrase}$

$\boldsymbol{P}: \boldsymbol{Pattern}, where\ P(\mathbf{x}) \in V^*$

$\boldsymbol{v}: \boldsymbol{Verbalizer}, \boldsymbol{\mathcal{L}} \to V$

$(\boldsymbol{P}, \boldsymbol{v}): \boldsymbol{Pattern\text{-}Verbalizer\ Pair}\ (\mathbf{PVP})$

Task: RTE **(Recognizing Textual Entailment)**

Sen1: **Oil prices rise.**

Sen2: **Oil prices fall back.**

Label: Not Entailed -> No

Entailed -> Yes

Pattern-Exploiting Training

-Pattern-Verbalizer Pair

<Pattern-Verbalizer Pair>

<Notation>

$M: \mathit{Masked\ Language\ Model}$

$\boldsymbol{V}: \boldsymbol{Vocabulary}$

$\_\_\_ \in V\ : \mathit{Mask\ Token}$

$\mathcal{L}, A: \boldsymbol{Labels}, \boldsymbol{Target\ Classification\ Task}$

$\mathbf{x} = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_k): \mathit{Input\ for}\ \boldsymbol{Task\ A}$

$\boldsymbol{s}_i \in \boldsymbol{V}^*: \boldsymbol{Phrase}$

$\boldsymbol{P}: \boldsymbol{Pattern}, \mathit{where}\ P(\mathbf{x}) \in V^*$

$v: \boldsymbol{Verbalizer}, \mathcal{L} \rightarrow \boldsymbol{V}$

$(\boldsymbol{P}, \boldsymbol{v}): \boldsymbol{Pattern\text{-}Verbalizer\ Pair}\ (\mathbf{PVP})$

Task: RTE (Recognizing Textual Entailment)

Sen1: Oil prices rise.

Sen2: Oil prices fall back.

Label: Not Entailed -> No

Entailed -> Yes

<Pattern-Verbalizer Pair>

<Notation>

$M: Masked\ Language\ Model$

$\boldsymbol{V}: \boldsymbol{Vocabulary}$

$\underline{\qquad} \in V : Mask\ Token$

$\boldsymbol{\mathcal{L}}, A: \boldsymbol{Labels}, \boldsymbol{Target\ Classification\ Task}$

$\mathbf{x} = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_k): Input\ for\ \boldsymbol{Task\ A}$

$\boldsymbol{s}_i \in \boldsymbol{V}^*: \boldsymbol{Phrase}$

$\boldsymbol{P}: \boldsymbol{Pattern}, where\ P(\mathbf{x}) \in V^*$

$\boldsymbol{v}: \boldsymbol{Verbalizer}, \boldsymbol{\mathcal{L}} \rightarrow \boldsymbol{V}$

$(\boldsymbol{P}, \boldsymbol{v}): \boldsymbol{Pattern\text{ - }Verbalizer\ Pair}\ (\mathbf{PVP})$

Task: RTE (Recognizing Textual Entailment)

Sen1: Oil prices rise.

Sen2: Oil prices fall back.

Label: Not Entailed -> No

Entailed -> Yes

$\boldsymbol{P}(\mathbf{x})$:

Oil prices rise? ____, Oil prices fall back.

<Pattern-Verbalizer Pair>

<Notation>

M: *Masked Language Model*

$\boldsymbol{V}$: ***Vocabulary***

$\_\_\_ \in V$: *Mask Token*

$\boldsymbol{\mathcal{L}}, \boldsymbol{A}$: ***Labels***, ***Target Classification Task***

$\mathbf{x} = (\boldsymbol{s_1}, \ldots, \boldsymbol{s_k})$: *Input for* ***Task A***

$\boldsymbol{s_i} \in \boldsymbol{V}^*$: ***Phrase***

$\boldsymbol{P}$: ***Pattern***, *where* $\boldsymbol{P}(\mathbf{x}) \in \boldsymbol{V}^*$

$\boldsymbol{v}$: ***Verbalizer***, $\boldsymbol{\mathcal{L}} \rightarrow \boldsymbol{V}$

$(\boldsymbol{P}, \boldsymbol{v})$: ***Pattern - Verbalizer Pair*** (PVP)

Task: RTE (Recognizing Textual Entailment)

Sen1: Oil prices rise.

Sen2: Oil prices fall back.

Label: Not Entailed -> No

Entailed -> Yes

$P(\mathbf{x})$:

Oil prices rise? ___, Oil prices fall back.

Pattern-Exploiting Training

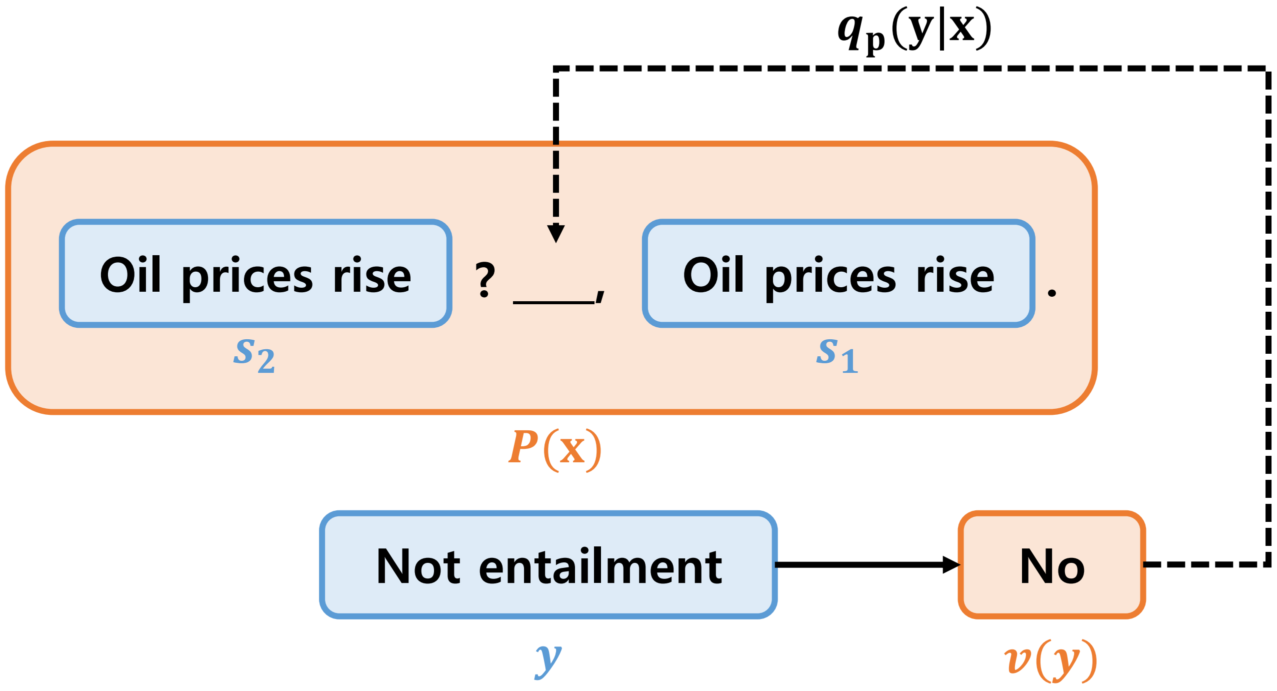
-Pattern-Verbalizer Pair

<Pattern-Verbalizer Pair>

$$\boldsymbol{P}: \boldsymbol{Pattern}, where\ \boldsymbol{P}(\mathbf{x}) \in \boldsymbol{V}^*$$

$$\boldsymbol{v}: \boldsymbol{Verbalizer}, \mathcal{L} \rightarrow V: \boldsymbol{v}(\boldsymbol{y}) \in \boldsymbol{V}$$

$$(\boldsymbol{P}, \boldsymbol{v}): \boldsymbol{Pattern\text{ - }Verbalizer\ Pair}\ (\mathbf{PVP})$$

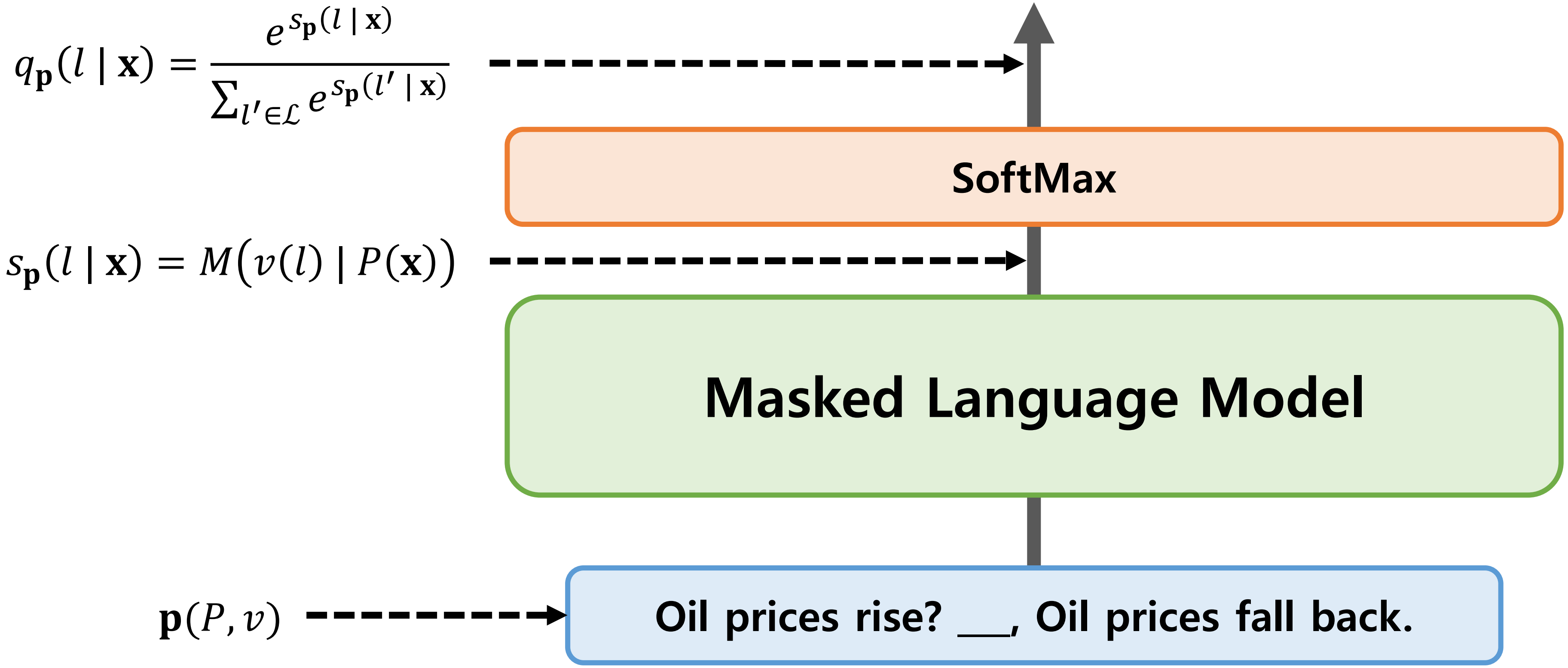


$\boldsymbol{q}_{\mathbf{p}}(\mathbf{y}|\mathbf{x})$
Oil prices rise
? ____,
Oil prices rise
.
$\boldsymbol{s}_2$
$\boldsymbol{s}_1$
$\boldsymbol{P}(\mathbf{x})$
Not entailment
No
$\boldsymbol{y}$
$\boldsymbol{v}(\boldsymbol{y})$

<PVP Training and Inference>

$$\mathbf{p} = (P, v)\text{: PVP}, \qquad s_{\mathbf{p}}(l \mid \mathbf{x}) = M\big(v(l) \mid P(\mathbf{x})\big)$$

$$q_{\mathbf{p}}(l \mid \mathbf{x}) = \frac{e^{s_{\mathbf{p}}(l \mid \mathbf{x})}}{\sum_{l' \in \mathcal{L}} e^{s_{\mathbf{p}}(l' \mid \mathbf{x})}}$$



$$q_{\mathbf{p}}(l \mid \mathbf{x}) = \frac{e^{s_{\mathbf{p}}(l \mid \mathbf{x})}}{\sum_{l' \in \mathcal{L}} e^{s_{\mathbf{p}}(l' \mid \mathbf{x})}}$$

$$s_{\mathbf{p}}(l \mid \mathbf{x}) = M\big(v(l) \mid P(\mathbf{x})\big)$$

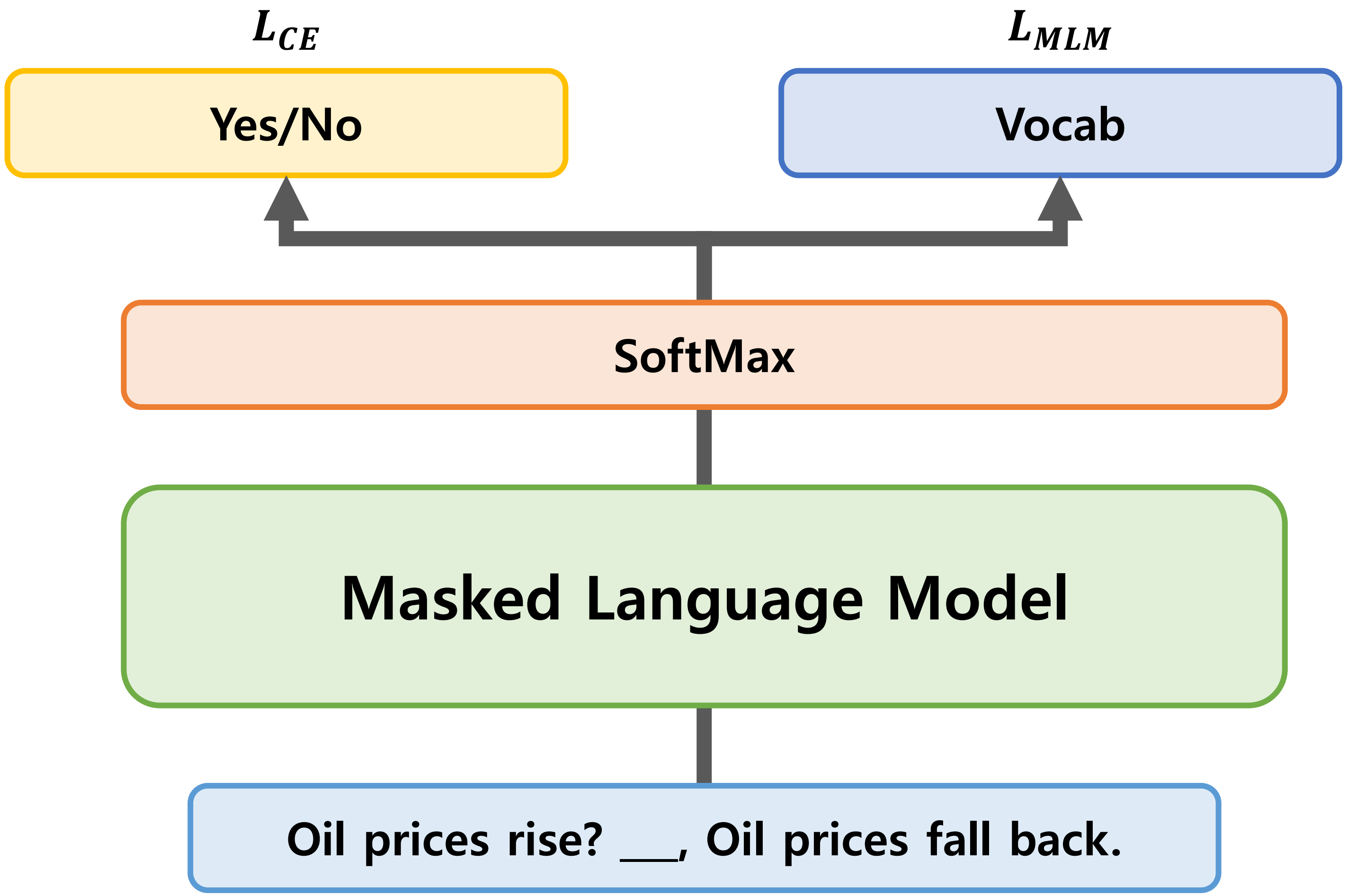
$$\mathbf{p}(P, v)$$

Pattern-Exploiting Training

-PVP Training and Inference

<Auxiliary Language Modeling>

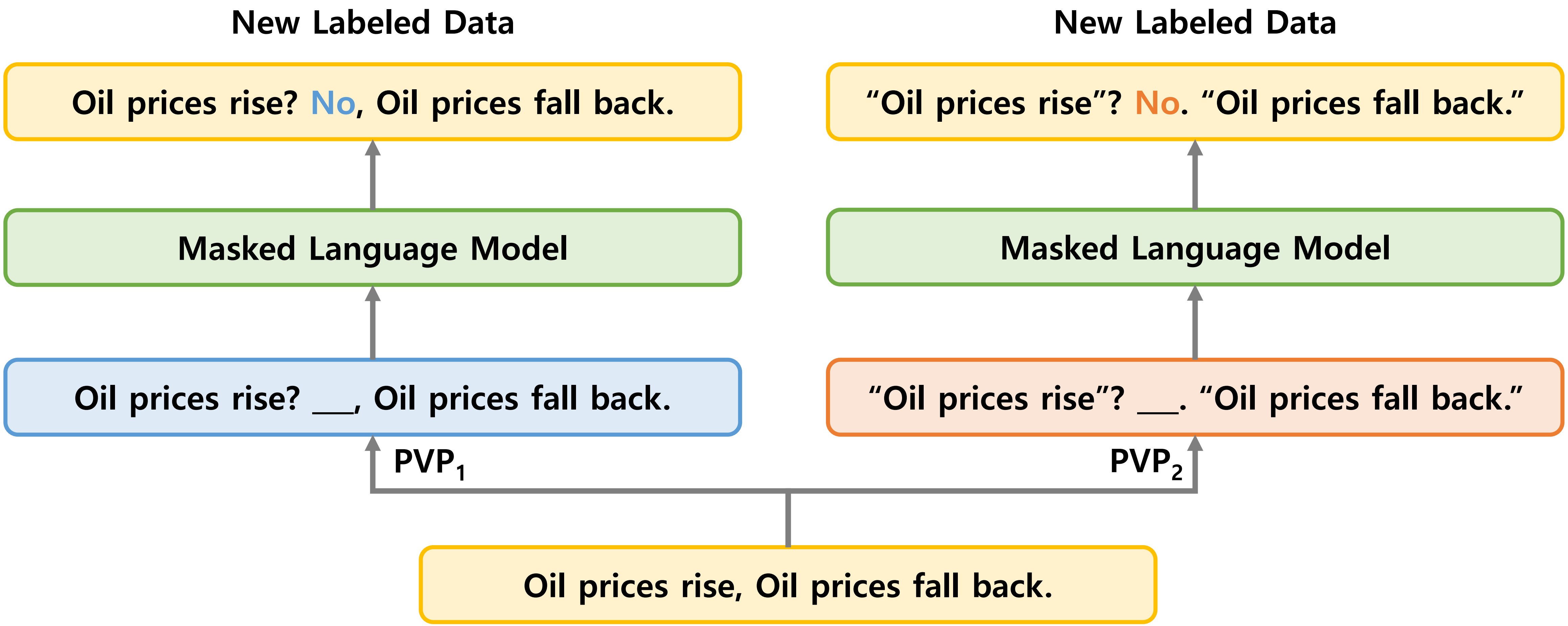
$$L = (1 - \alpha) \cdot L_{CE} + \alpha \cdot L_{MLM}$$



Pattern-Exploiting Training

-PVP Training and Inference

<Combining PVPs>


New Labeled Data
Oil prices rise? No, Oil prices fall back.
Masked Language Model
Oil prices rise? ___, Oil prices fall back.
PVP₁
New Labeled Data
"Oil prices rise"? No. "Oil prices fall back."
Masked Language Model
"Oil prices rise"? ___. "Oil prices fall back."
PVP₂
Oil prices rise, Oil prices fall back.

Pattern-Exploiting Training

-PVP Training and Inference

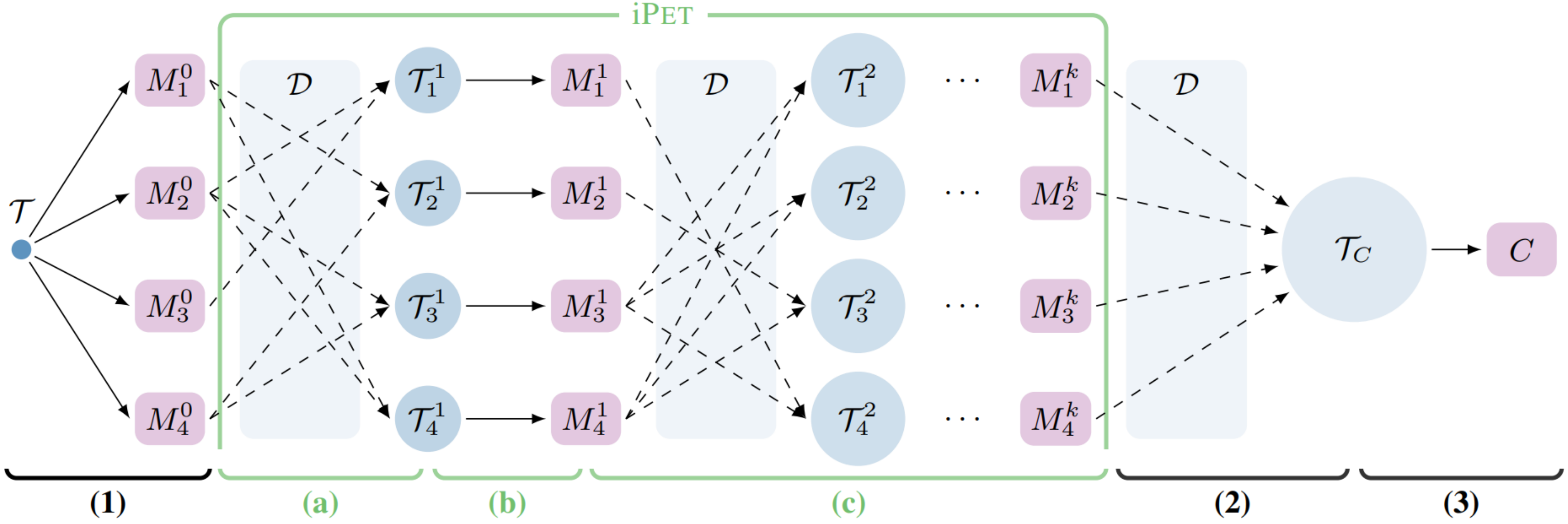
<Combining PVPs>

$\mathcal{T}$: *Training Dataset*, $\mathcal{D}$: *Unlabeled Data*

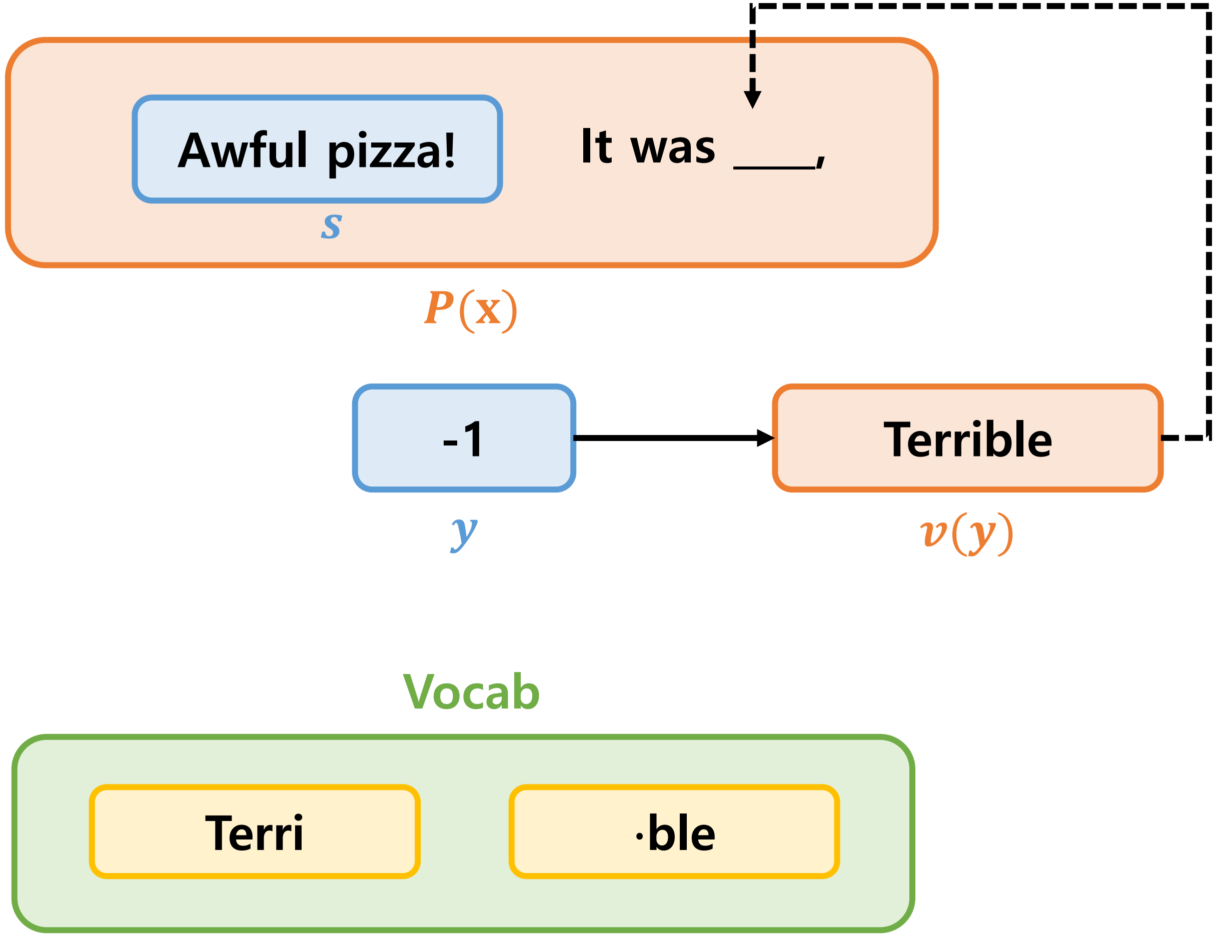
$\mathcal{M} = \{M_{\mathbf{p}} \mid \mathbf{p} \in \mathcal{P}\}$: *Ensemble of Fine-tuned Model*

$$s_{\mathcal{M}}(l \mid \mathbf{x}) = \frac{1}{Z} \sum_{\mathbf{p} \in \mathcal{P}} w(\mathbf{p}) \cdot s_{\mathbf{p}}(l \mid \mathbf{x})$$

$Z = \sum_{\mathbf{p} \in \mathcal{P}} w(\mathbf{p})$, $w(\mathbf{p})$: *Weight (1 or Accuracy before Training)*



<PET with Multiple Tasks>


Awful pizza!
It was ____,
s
$P(\mathbf{x})$
-1
Terrible
y
$v(y)$
Vocab
Terri
·ble

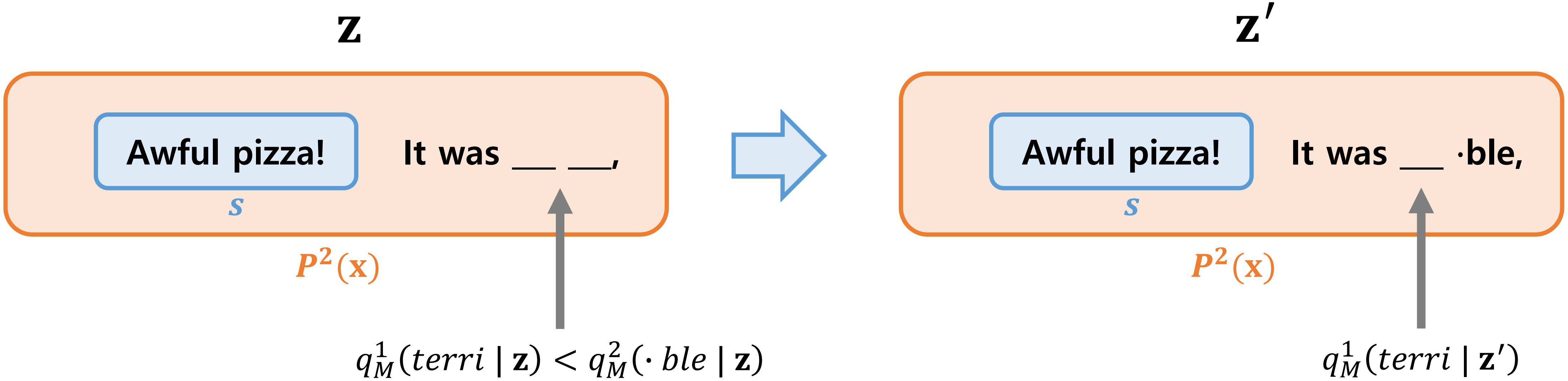
Pattern-Exploiting Training

-PVP Training and Inference

<Combining PVPs>

$$q_{\mathbf{p}}(y \mid x) = q\left(v(y) \mid P^{k}(x)\right)$$

$$q(t_1 \dots t_k \mid \mathbf{z}) = \begin{cases} 1 & if\ k = 0 \\ q_M^{j}(t_j \mid \mathbf{z}) \cdot q(t' \mid \mathbf{z}') & if\ k \geq 1 \end{cases}$$



$\mathbf{z}$

Awful pizza! It was ___ ___,

s

$P^2(\mathbf{x})$

$q_M^1(terri \mid \mathbf{z}) < q_M^2(\cdot\, ble \mid \mathbf{z})$

$\mathbf{z}'$

Awful pizza! It was ___ ·ble,

s

$P^2(\mathbf{x})$

$q_M^1(terri \mid \mathbf{z}')$

Experiments

- Tasks and Patterns

- Results

<Examples of Tasks and Patterns>

Task: WiC (Word in Context)

W: bed

S1: There's a log of trash on the bed of the river

S2: I keep a glass of water next to my bed

Label: Similar

Pattern

"s1" / "s2". Similar sense of "w"? ___. // yes, no

s1 s2 Does w have the same meaning in both sentences? ___ // yes, no

w. Sense (1) (a) "s1" (___) "s2". // b, 2

Task: COPA (Choice of Plausible Alternatives)

P: The man broke his toe. What was the cause of this?

C1: He got a hole in his sock.

C2: He dropped a hammer on his foot

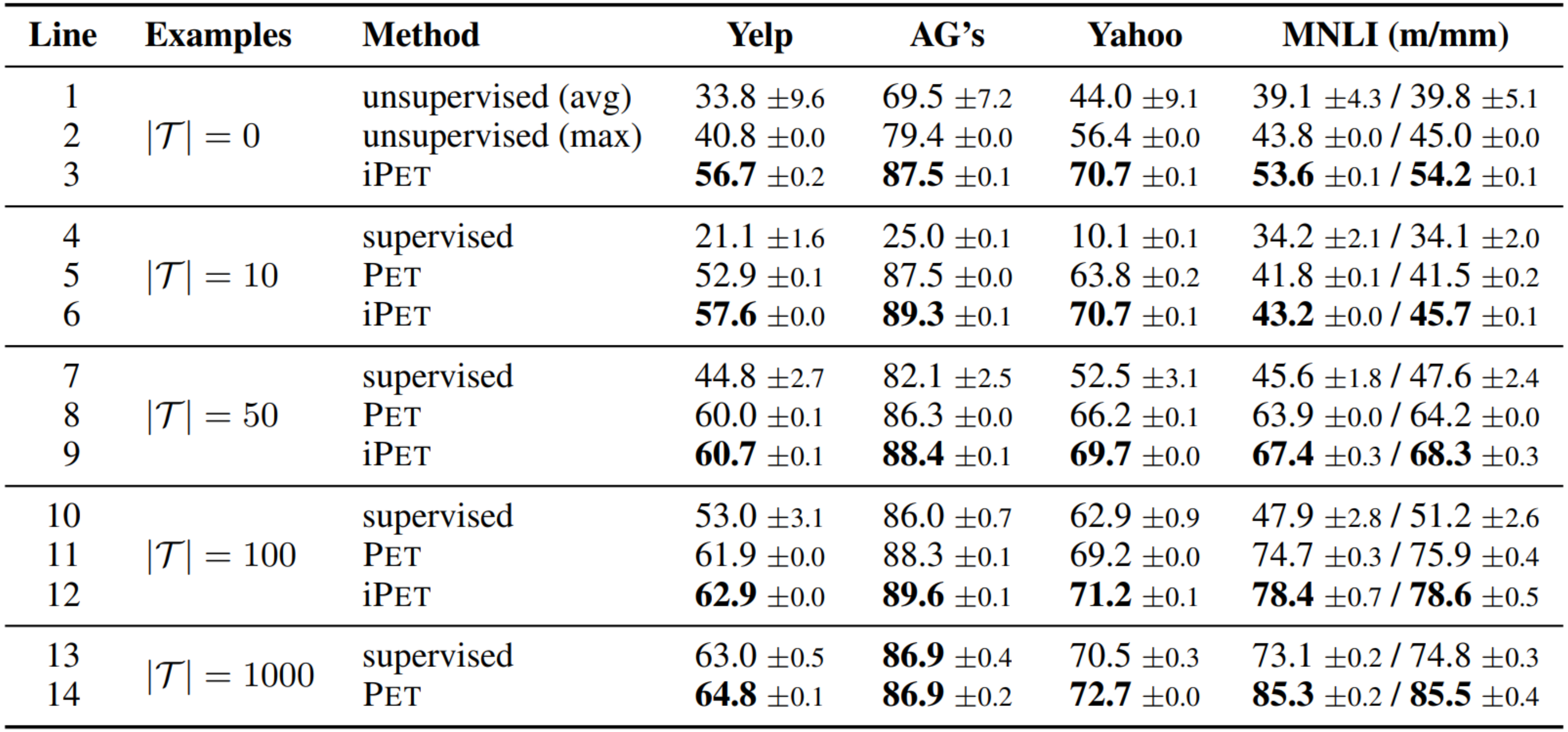
Label: C2

Pattern

"C1" or "C2"? p, so ___. // C1, C2

C1 or C2? p, so ___. // C1, C2

<Results>



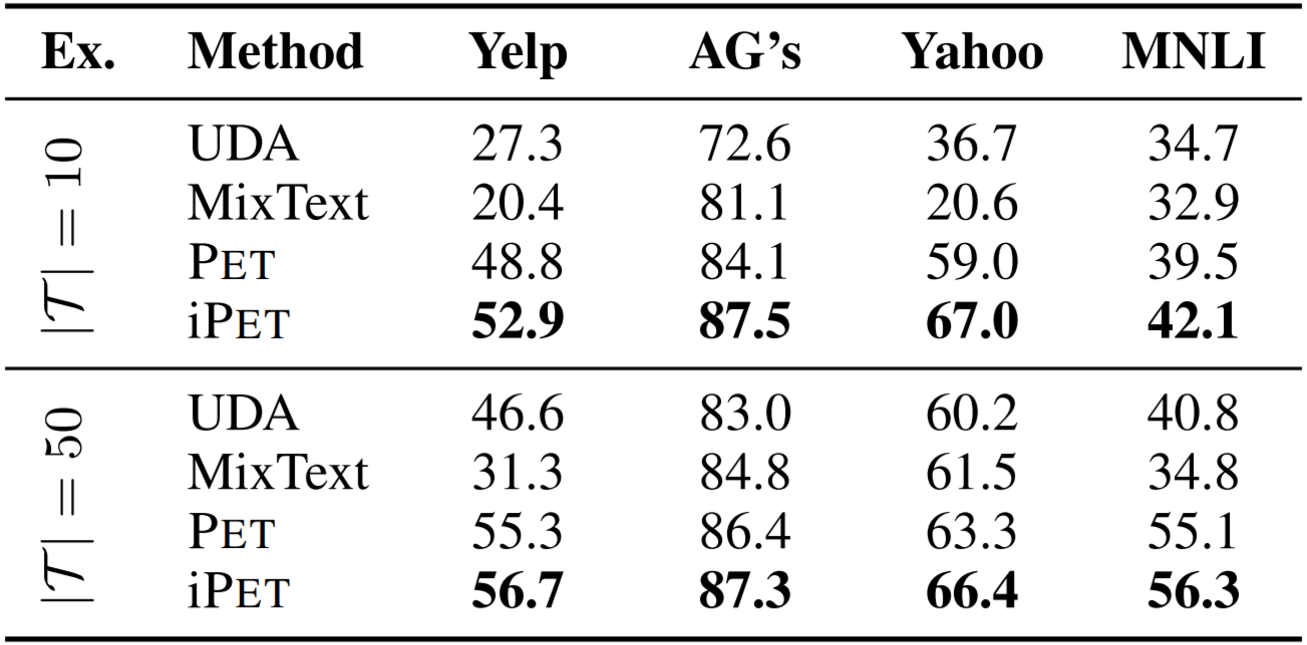
Line	Examples	Method	Yelp	AG's	Yahoo	MNLI (m/mm)
1	$\lvert\mathcal{T}\rvert = 0$	unsupervised (avg)	33.8 ±9.6	69.5 ±7.2	44.0 ±9.1	39.1 ±4.3 / 39.8 ±5.1
2		unsupervised (max)	40.8 ±0.0	79.4 ±0.0	56.4 ±0.0	43.8 ±0.0 / 45.0 ±0.0
3		iPET	**56.7** ±0.2	**87.5** ±0.1	**70.7** ±0.1	**53.6** ±0.1 / **54.2** ±0.1
4	$\lvert\mathcal{T}\rvert = 10$	supervised	21.1 ±1.6	25.0 ±0.1	10.1 ±0.1	34.2 ±2.1 / 34.1 ±2.0
5		PET	52.9 ±0.1	87.5 ±0.0	63.8 ±0.2	41.8 ±0.1 / 41.5 ±0.2
6		iPET	**57.6** ±0.0	**89.3** ±0.1	**70.7** ±0.1	**43.2** ±0.0 / **45.7** ±0.1
7	$\lvert\mathcal{T}\rvert = 50$	supervised	44.8 ±2.7	82.1 ±2.5	52.5 ±3.1	45.6 ±1.8 / 47.6 ±2.4
8		PET	60.0 ±0.1	86.3 ±0.0	66.2 ±0.1	63.9 ±0.0 / 64.2 ±0.0
9		iPET	**60.7** ±0.1	**88.4** ±0.1	**69.7** ±0.0	**67.4** ±0.3 / **68.3** ±0.3
10	$\lvert\mathcal{T}\rvert = 100$	supervised	53.0 ±3.1	86.0 ±0.7	62.9 ±0.9	47.9 ±2.8 / 51.2 ±2.6
11		PET	61.9 ±0.0	88.3 ±0.1	69.2 ±0.0	74.7 ±0.3 / 75.9 ±0.4
12		iPET	**62.9** ±0.0	**89.6** ±0.1	**71.2** ±0.1	**78.4** ±0.7 / **78.6** ±0.5
13	$\lvert\mathcal{T}\rvert = 1000$	supervised	63.0 ±0.5	**86.9** ±0.4	70.5 ±0.3	73.1 ±0.2 / 74.8 ±0.3
14		PET	**64.8** ±0.1	**86.9** ±0.2	**72.7** ±0.0	**85.3** ±0.2 / **85.5** ±0.4

<Average accuracy and standard deviation for RoBERTa (large)>

Experiments

- Results

<Results>



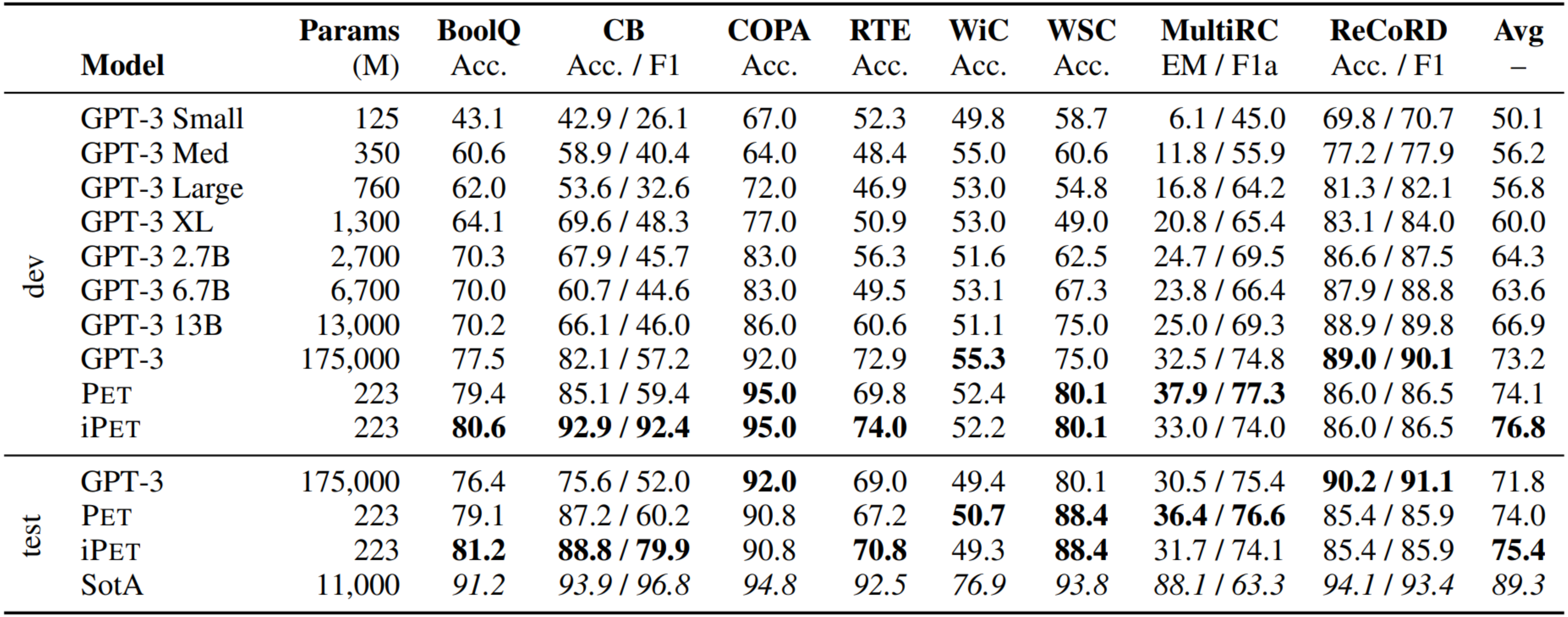
Ex.	Method	Yelp	AG's	Yahoo	MNLI
$\|\mathcal{T}\| = 10$	UDA	27.3	72.6	36.7	34.7
	MixText	20.4	81.1	20.6	32.9
	PET	48.8	84.1	59.0	39.5
	iPET	**52.9**	**87.5**	**67.0**	**42.1**
$\|\mathcal{T}\| = 50$	UDA	46.6	83.0	60.2	40.8
	MixText	31.3	84.8	61.5	34.8
	PET	55.3	86.4	63.3	55.1
	iPET	**56.7**	**87.3**	**66.4**	**56.3**

<Comparison of PET with two state-of-the-art semi-supervised method using RoBERTa (base)>

Experiments

- Results

<Results>



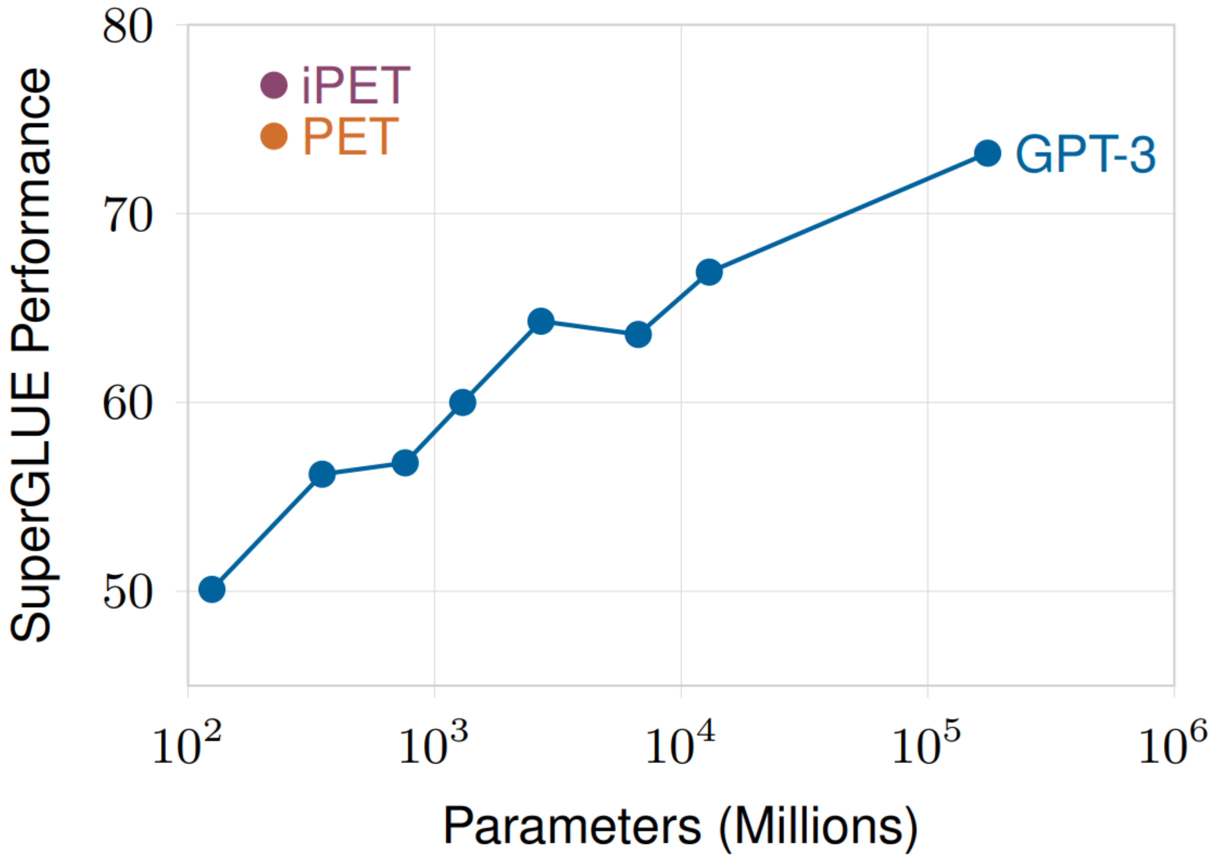
		Params	BoolQ	CB	COPA	RTE	WiC	WSC	MultiRC	ReCoRD	Avg
	Model	(M)	Acc.	Acc. / F1	Acc.	Acc.	Acc.	Acc.	EM / F1a	Acc. / F1	–
dev	GPT-3 Small	125	43.1	42.9 / 26.1	67.0	52.3	49.8	58.7	6.1 / 45.0	69.8 / 70.7	50.1
	GPT-3 Med	350	60.6	58.9 / 40.4	64.0	48.4	55.0	60.6	11.8 / 55.9	77.2 / 77.9	56.2
	GPT-3 Large	760	62.0	53.6 / 32.6	72.0	46.9	53.0	54.8	16.8 / 64.2	81.3 / 82.1	56.8
	GPT-3 XL	1,300	64.1	69.6 / 48.3	77.0	50.9	53.0	49.0	20.8 / 65.4	83.1 / 84.0	60.0
	GPT-3 2.7B	2,700	70.3	67.9 / 45.7	83.0	56.3	51.6	62.5	24.7 / 69.5	86.6 / 87.5	64.3
	GPT-3 6.7B	6,700	70.0	60.7 / 44.6	83.0	49.5	53.1	67.3	23.8 / 66.4	87.9 / 88.8	63.6
	GPT-3 13B	13,000	70.2	66.1 / 46.0	86.0	60.6	51.1	75.0	25.0 / 69.3	88.9 / 89.8	66.9
	GPT-3	175,000	77.5	82.1 / 57.2	92.0	72.9	**55.3**	75.0	32.5 / 74.8	**89.0 / 90.1**	73.2
	PET	223	79.4	85.1 / 59.4	**95.0**	69.8	52.4	**80.1**	**37.9 / 77.3**	86.0 / 86.5	74.1
	iPET	223	**80.6**	**92.9 / 92.4**	**95.0**	**74.0**	52.2	**80.1**	33.0 / 74.0	86.0 / 86.5	**76.8**
test	GPT-3	175,000	76.4	75.6 / 52.0	**92.0**	69.0	49.4	80.1	30.5 / 75.4	**90.2 / 91.1**	71.8
	PET	223	79.1	87.2 / 60.2	90.8	67.2	**50.7**	**88.4**	**36.4 / 76.6**	85.4 / 85.9	74.0
	iPET	223	**81.2**	**88.8 / 79.9**	90.8	**70.8**	49.3	**88.4**	31.7 / 74.1	85.4 / 85.9	**75.4**
	SotA	11,000	*91.2*	*93.9 / 96.8*	*94.8*	*92.5*	*76.9*	*93.8*	*88.1 / 63.3*	*94.1 / 93.4*	*89.3*

<Results on SuperGLUE for GPT-3 primed with 32 examples and for PET/iPET with ALBERT-xxlarge-v2>

Experiments

- Results

<Results>



<Performance on SuperGLUE with 32 training examples>

Conclusion

<Conclusion>

- Proposed Pattern-Exploiting Training that consists of defining pairs of cloze question patterns and verbalizers that help leverage the knowledge contained within pretrained language models for downstream tasks.

- Proposed modified PET enabling to be used for tasks that require predicting multiple tokens.

- Shown that using PET, it is possible to achieve few-shot text classification performance similar to GPT-3 on SuperGLUE with LMs that have much fewer parameters.

<Conclusion>

- Proposed Pattern-Exploiting Training that consists of defining pairs of cloze question patterns and verbalizers that help leverage the knowledge contained within pretrained language models for downstream tasks.
- Proposed modified PET enabling to be used for tasks that require predicting multiple tokens.
- Shown that using PET, it is possible to achieve few-shot text classification performance similar to GPT-3 on SuperGLUE with LMs that have much fewer parameters.

PET has achieved remarkable performance, but it requires thousands of unlabeled data, and hand-crafted patterns.

Additionally, since discrete prompts are used, the results may be sub-optimal to continuous neural network.

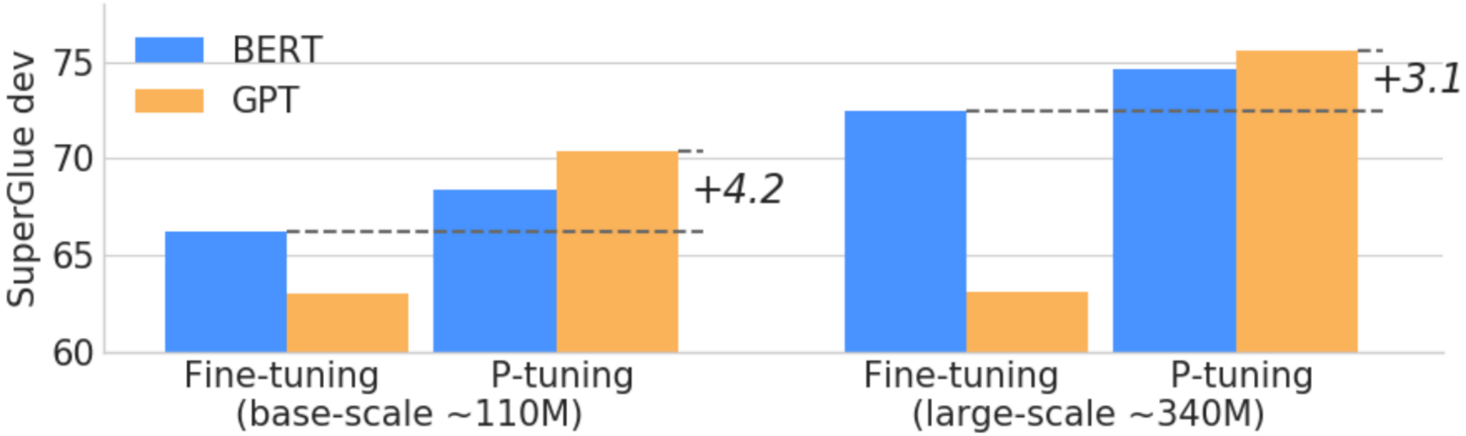
How can PET be further improved?

GPT Understands, Too

Liu et al., 2021, arXiv

P-Tuning

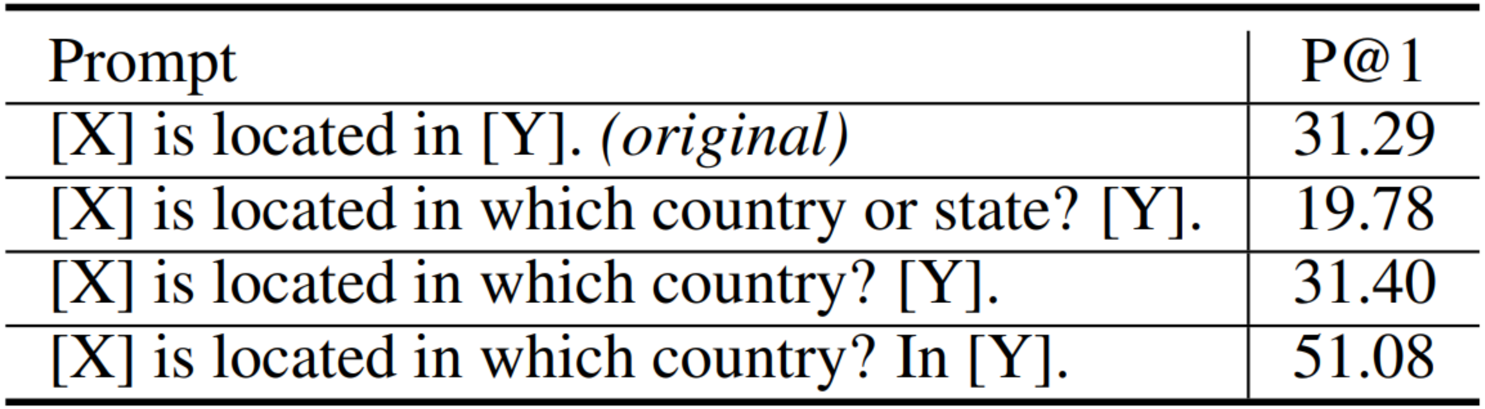
<Overview>



<Average scores on 7 dev datasets of SuperGLUE>

“GPTs can be better than similar-sized BERTs on NLU with P-tuning.”

<Overview>

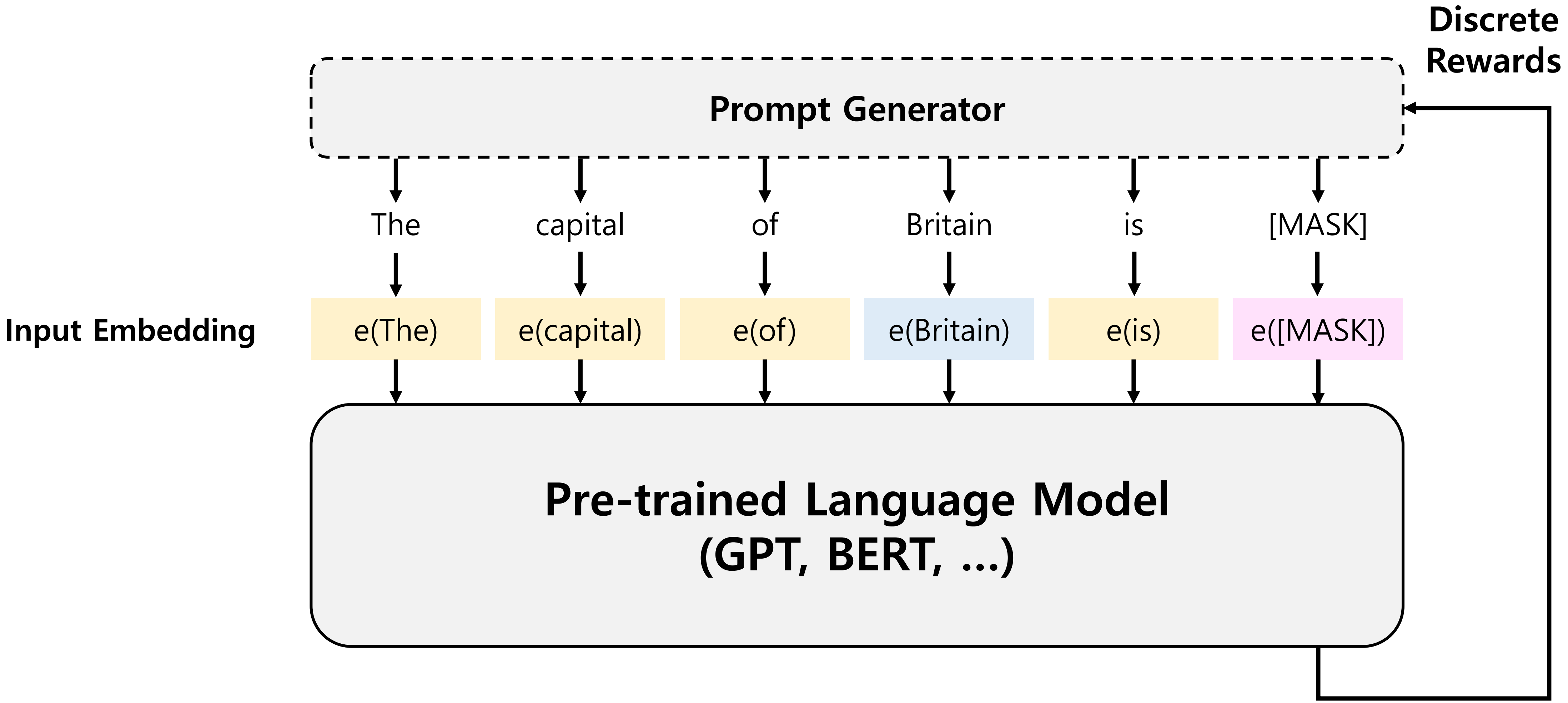


Prompt	P@1
[X] is located in [Y]. *(original)*	31.29
[X] is located in which country or state? [Y].	19.78
[X] is located in which country? [Y].	31.40
[X] is located in which country? In [Y].	51.08

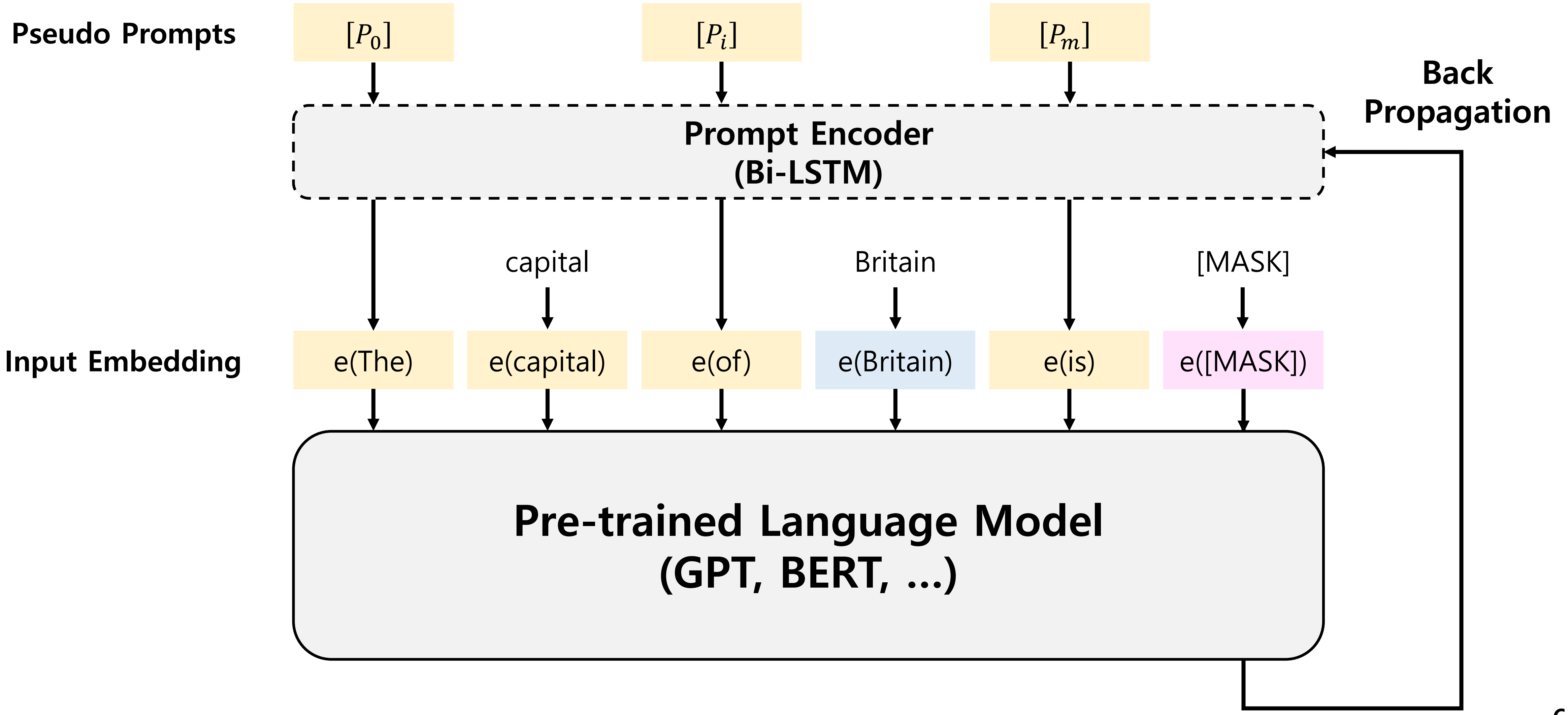
<Case study on LAMA-TREx P17 with bert-base-cased>

“A single-word change in prompts could yield a drastic difference.”

<Discrete Prompt Search>

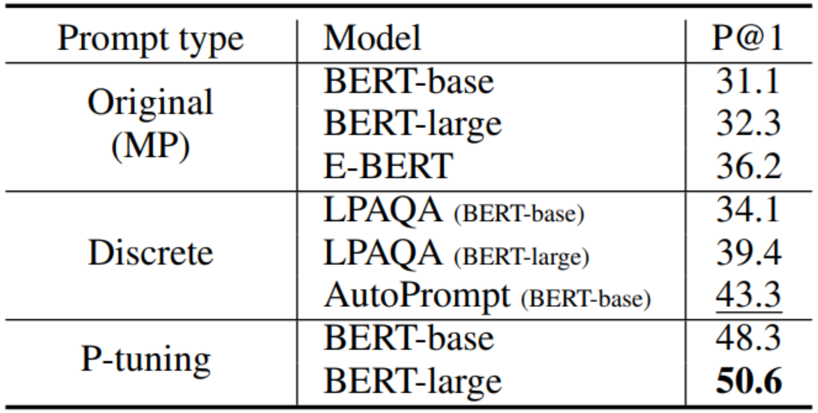

Discrete
Rewards
Prompt Generator
The
capital
of
Britain
is
[MASK]
Input Embedding
e(The)
e(capital)
e(of)
e(Britain)
e(is)
e([MASK])
Pre-trained Language Model
(GPT, BERT, …)

<P-Tuning>

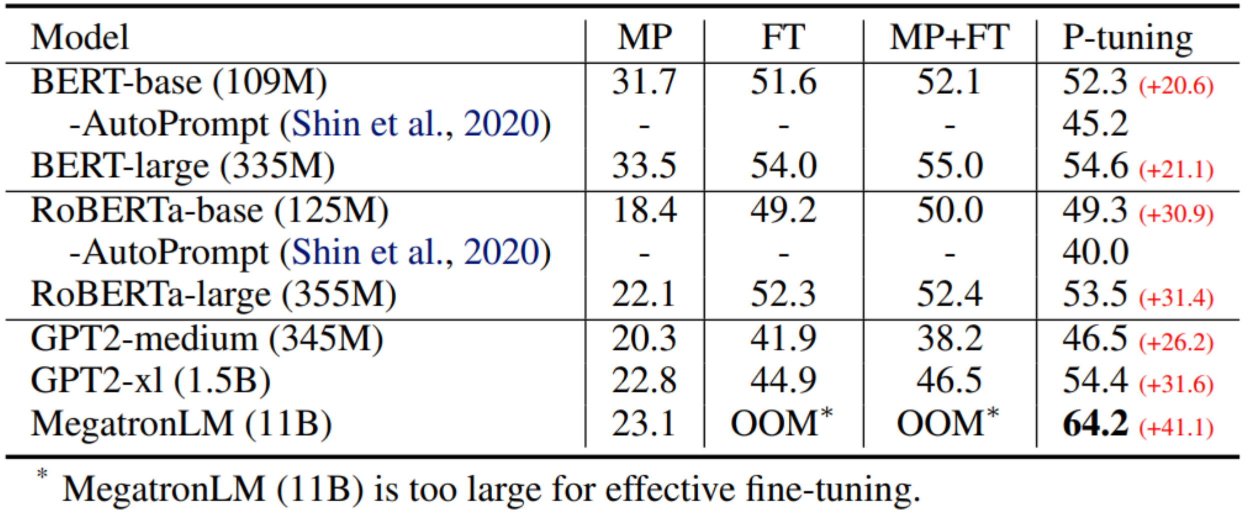

Pseudo Prompts
$[P_0]$
$[P_i]$
$[P_m]$
Back Propagation
Prompt Encoder (Bi-LSTM)
capital
Britain
[MASK]
Input Embedding
e(The)
e(capital)
e(of)
e(Britain)
e(is)
e([MASK])
Pre-trained Language Model (GPT, BERT, …)

Experiments

<Results>



Prompt type	Model	P@1
Original (MP)	BERT-base	31.1
	BERT-large	32.3
	E-BERT	36.2
Discrete	LPAQA (BERT-base)	34.1
	LPAQA (BERT-large)	39.4
	AutoPrompt (BERT-base)	43.3
P-tuning	BERT-base	48.3
	BERT-large	**50.6**



Model	MP	FT	MP+FT	P-tuning
BERT-base (109M)	31.7	51.6	52.1	52.3 (+20.6)
-AutoPrompt (Shin et al., 2020)	-	-	-	45.2
BERT-large (335M)	33.5	54.0	55.0	54.6 (+21.1)
RoBERTa-base (125M)	18.4	49.2	50.0	49.3 (+30.9)
-AutoPrompt (Shin et al., 2020)	-	-	-	40.0
RoBERTa-large (355M)	22.1	52.3	52.4	53.5 (+31.4)
GPT2-medium (345M)	20.3	41.9	38.2	46.5 (+26.2)
GPT2-xl (1.5B)	22.8	44.9	46.5	54.4 (+31.6)
MegatronLM (11B)	23.1	OOM*	OOM*	**64.2** (+41.1)

* MegatronLM (11B) is too large for effective fine-tuning.

<Knowledge probing Precision@1 on LAMA-34k (left) and LAMA-29k (right)>

"P-tuning outperforms all the discrete prompt searching baseline"

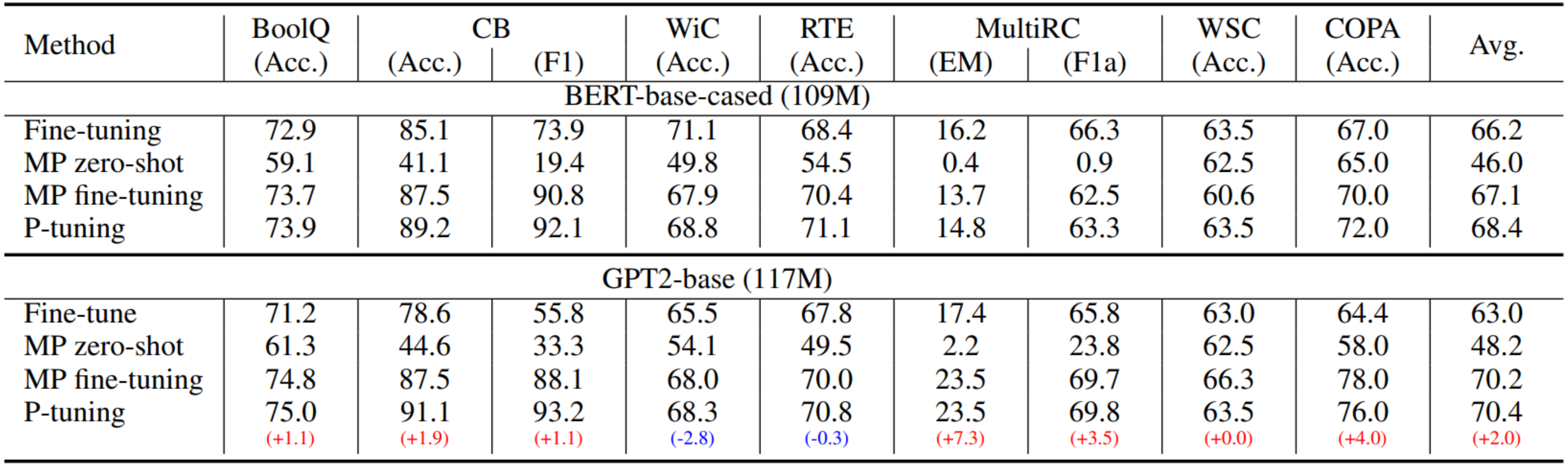
"Despite fixed pre-trained model parameters, P-tuning overwhelms the fine-tuning GPTs"

(MP: Manual Prompt, FT: Fine-tuning, MP+FT: Manual Prompt Augmented Fine-tuning, PT: P-tuning)

Experiments

- Results

<Results>



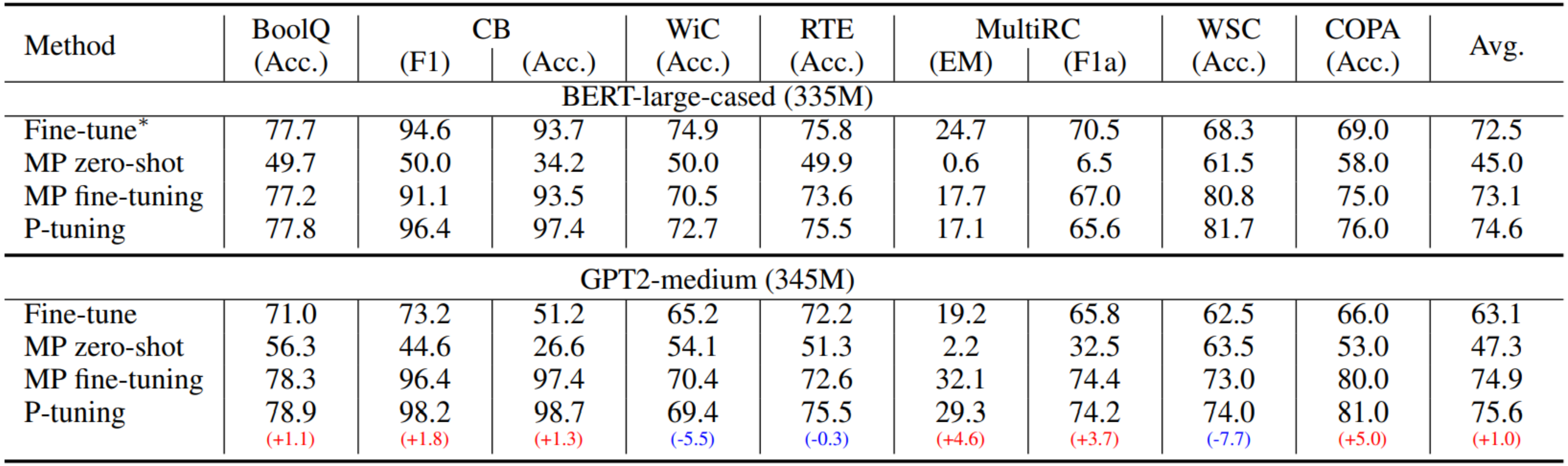
Method	BoolQ	CB		WiC	RTE	MultiRC		WSC	COPA	Avg.
	(Acc.)	(Acc.)	(F1)	(Acc.)	(Acc.)	(EM)	(F1a)	(Acc.)	(Acc.)	
BERT-base-cased (109M)										
Fine-tuning	72.9	85.1	73.9	71.1	68.4	16.2	66.3	63.5	67.0	66.2
MP zero-shot	59.1	41.1	19.4	49.8	54.5	0.4	0.9	62.5	65.0	46.0
MP fine-tuning	73.7	87.5	90.8	67.9	70.4	13.7	62.5	60.6	70.0	67.1
P-tuning	73.9	89.2	92.1	68.8	71.1	14.8	63.3	63.5	72.0	68.4
GPT2-base (117M)										
Fine-tune	71.2	78.6	55.8	65.5	67.8	17.4	65.8	63.0	64.4	63.0
MP zero-shot	61.3	44.6	33.3	54.1	49.5	2.2	23.8	62.5	58.0	48.2
MP fine-tuning	74.8	87.5	88.1	68.0	70.0	23.5	69.7	66.3	78.0	70.2
P-tuning	75.0 (+1.1)	91.1 (+1.9)	93.2 (+1.1)	68.3 (-2.8)	70.8 (-0.3)	23.5 (+7.3)	69.8 (+3.5)	63.5 (+0.0)	76.0 (+4.0)	70.4 (+2.0)

<Fully-supervised learning on SuperGLUE dev with base-scale models>

Experiments

- Results

<Results>



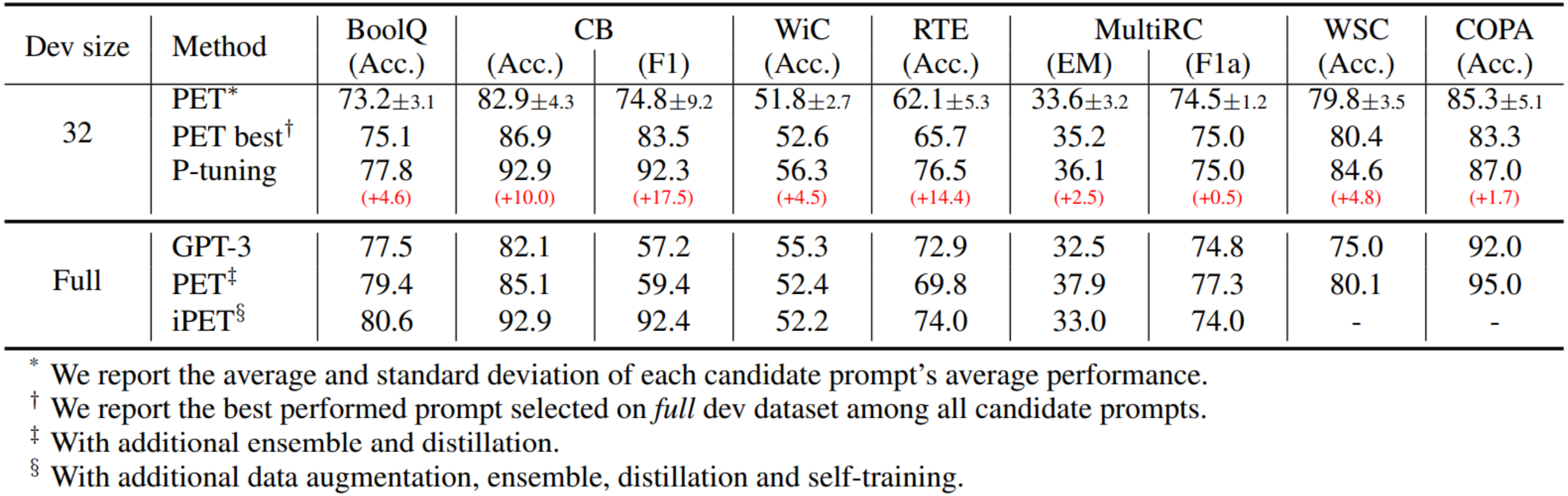
Method	BoolQ	CB		WiC	RTE	MultiRC		WSC	COPA	Avg.
	(Acc.)	(F1)	(Acc.)	(Acc.)	(Acc.)	(EM)	(F1a)	(Acc.)	(Acc.)	
BERT-large-cased (335M)										
Fine-tune*	77.7	94.6	93.7	74.9	75.8	24.7	70.5	68.3	69.0	72.5
MP zero-shot	49.7	50.0	34.2	50.0	49.9	0.6	6.5	61.5	58.0	45.0
MP fine-tuning	77.2	91.1	93.5	70.5	73.6	17.7	67.0	80.8	75.0	73.1
P-tuning	77.8	96.4	97.4	72.7	75.5	17.1	65.6	81.7	76.0	74.6
GPT2-medium (345M)										
Fine-tune	71.0	73.2	51.2	65.2	72.2	19.2	65.8	62.5	66.0	63.1
MP zero-shot	56.3	44.6	26.6	54.1	51.3	2.2	32.5	63.5	53.0	47.3
MP fine-tuning	78.3	96.4	97.4	70.4	72.6	32.1	74.4	73.0	80.0	74.9
P-tuning	78.9 (+1.1)	98.2 (+1.8)	98.7 (+1.3)	69.4 (-5.5)	75.5 (-0.3)	29.3 (+4.6)	74.2 (+3.7)	74.0 (-7.7)	81.0 (+5.0)	75.6 (+1.0)

<Fully-supervised learning on SuperGLUE dev with large-scale models>

Experiments

- Results

<Results>



Dev size	Method	BoolQ	CB		WiC	RTE	MultiRC		WSC	COPA
		(Acc.)	(Acc.)	(F1)	(Acc.)	(Acc.)	(EM)	(F1a)	(Acc.)	(Acc.)
32	PET*	73.2±3.1	82.9±4.3	74.8±9.2	51.8±2.7	62.1±5.3	33.6±3.2	74.5±1.2	79.8±3.5	85.3±5.1
	PET best†	75.1	86.9	83.5	52.6	65.7	35.2	75.0	80.4	83.3
	P-tuning	77.8 (+4.6)	92.9 (+10.0)	92.3 (+17.5)	56.3 (+4.5)	76.5 (+14.4)	36.1 (+2.5)	75.0 (+0.5)	84.6 (+4.8)	87.0 (+1.7)
Full	GPT-3	77.5	82.1	57.2	55.3	72.9	32.5	74.8	75.0	92.0
	PET‡	79.4	85.1	59.4	52.4	69.8	37.9	77.3	80.1	95.0
	iPET§	80.6	92.9	92.4	52.2	74.0	33.0	74.0	-	-

* We report the average and standard deviation of each candidate prompt's average performance.

† We report the best performed prompt selected on *full* dev dataset among all candidate prompts.

‡ With additional ensemble and distillation.

§ With additional data augmentation, ensemble, distillation and self-training.

<Few-shot learning (32 train samples) on SuperGLUE dev>

Conclusion

<Conclusion>

- Proposed P-tuning which augments pre-trained model’s ability in natural language understanding by automatically searching better prompts in the continuous space.
- On the SuperGLUE benchmark, P-tuning endows GPT-style models to show competitive performance with similar-size BERTs in natural language understanding, which is assumed impossible in the past.
- P-tuning also helps on bidirectional models and consequently outperforms state-of-the-art methods in the few-shot SuperGLUE benchmark.

Any Questions?

Thank You