GPT Understand, Too

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding,
Yujie Qian, Zhilin Yang, Jie Tang (2021)

단국대학교 모바일시스템공학과 양윤성

Contents

1-(1). Background

- Pre-trained Language Model can be categorized into 3
  - Unidirectional Language Models (GPT)
  - Bidirectional Language Models (BERT)
  - Hybrid Language Models (XLNet, UniLM)

1-(1). Background

- Originally, GPT has been perceived as unsuitable for NLU tasks
  - Because GPT is a left-to-right unidirectional model
- Some people think GPT's NLU task ability is underestimated
- How about in-context learning method using prompt for GPT?

1-(2). in-Context Learning

1. Enter a description or some examples of task in the input of LM
2. Then LM generates output that fits the task

- For in-context learning, prompting is required
  - User fits some template into the input of LM
  - To accurately understand and generate natural language

➢ Original method : Hand-crafted prompt

1-(3). Limitations

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

Table 1. Case study on LAMA-TREx P17 with bert-base-cased. A single-word change in prompts could yield a drastic difference.

- Large validation set required
- Instability of Hand-crafted prompts

✓ Find automatic prompt searching method!

1-(3). Limitations

How about AUTOPROMPT?

- First suggestion about way to find prompt automatically (2020)
- Hand-crafted prompt → Discrete prompt
- But neural networks are inherently continuous

  → Discrete prompt must be sub-optimal

1-(4). Main Idea

- Fine-tuning : Update weights by supervised learning using dataset specialized in tasks → Inefficient method in LLM
- P-tuning : Automatically search prompts in the continuous space
  - Only continuous prompts are updated weights
  - No need to adjust the model parameters

1-(5). Overview

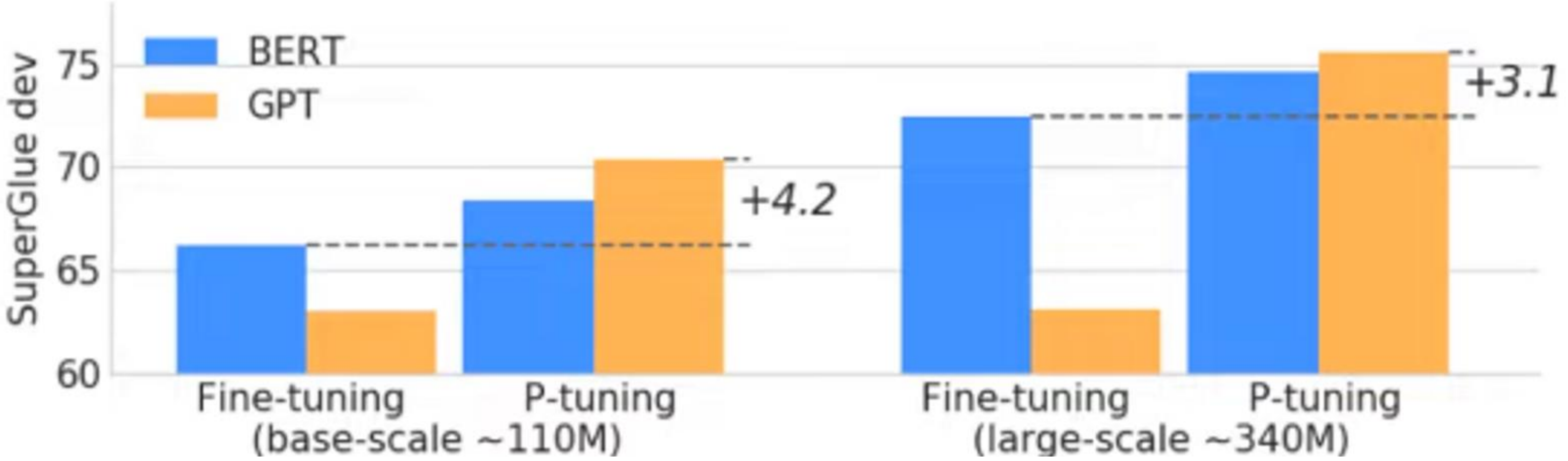


Figure 1. Average scores on 7 dev datasets of SuperGlue. GPTs can be better than similar-sized BERTs on NLU with P-tuning.

- P-tuning performance is better
- Authors want to break stereotype that GPT can only generate but do not understand

Contents

2-(1). Notation

- T : template — The capital of Britain is [MASK]
- X : context — Britain
- Y : target — [MASK]
- P : prompt — The capital of ⋯ is ⋯
- P_i : i'th prompt token — P_1 : The, P_2 : capital, ⋯
- E : pretrained embedding

2-(2). Discrete Prompt Search

- Input tokens of pre-trained LM T = {$[P_{0:i}]$, X, $[P_{i+1:\mathrm{m}}]$, Y} are mapped to input embeddings {e($[P_{0:i}]$), e(X), e($[P_{i+1:\mathrm{m}}]$), e(Y)}.

$[\mathrm{P}_i] \in \mathcal{V}$

$\mathcal{M}$: pretrained model

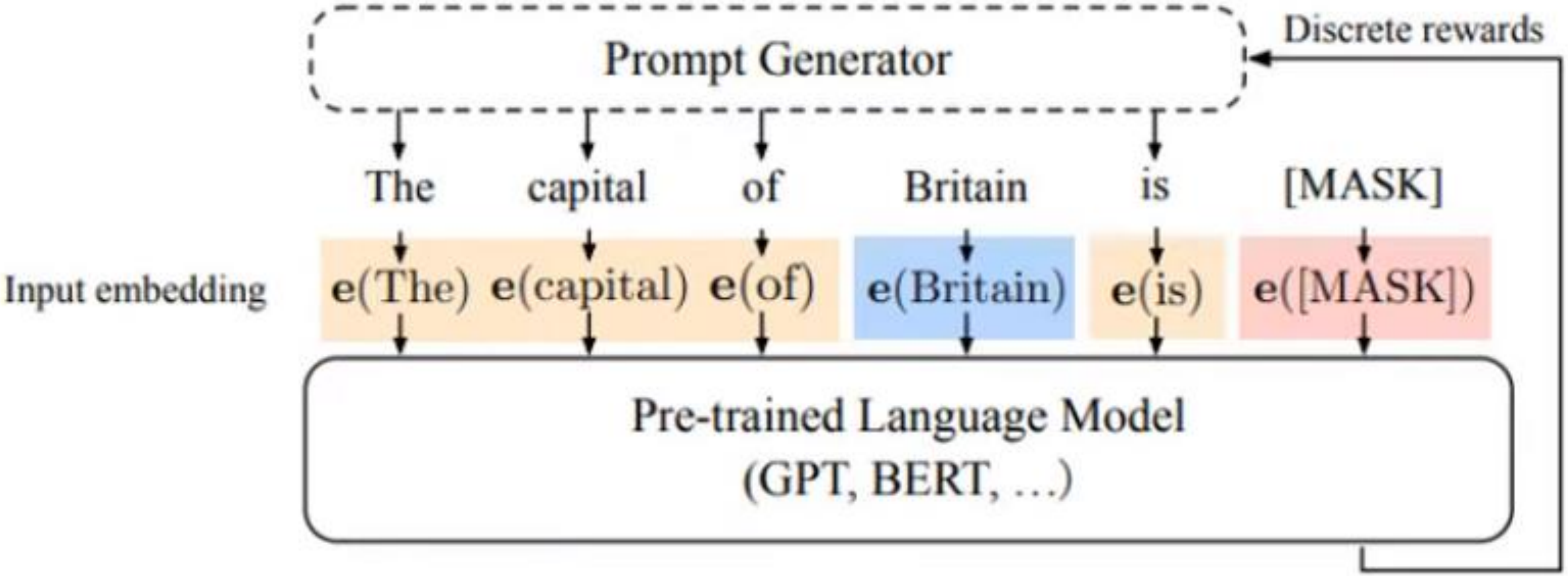
$\mathcal{V}$: vocabulary of $\mathcal{M}$

The capital of Britain is [MASK]

⇩

$$\{\mathbf{e}(\mathrm{The}), \mathbf{e}(\mathrm{capital}), \mathbf{e}(\mathrm{of}), \mathbf{e}(\mathrm{Britain}), \mathbf{e}(\mathrm{is}), \mathbf{e}([\mathrm{Mask}])\}$$

2-(2). Discrete Prompt Search



2-(3). P-tuning

- Prompt is replaced by pseudo token

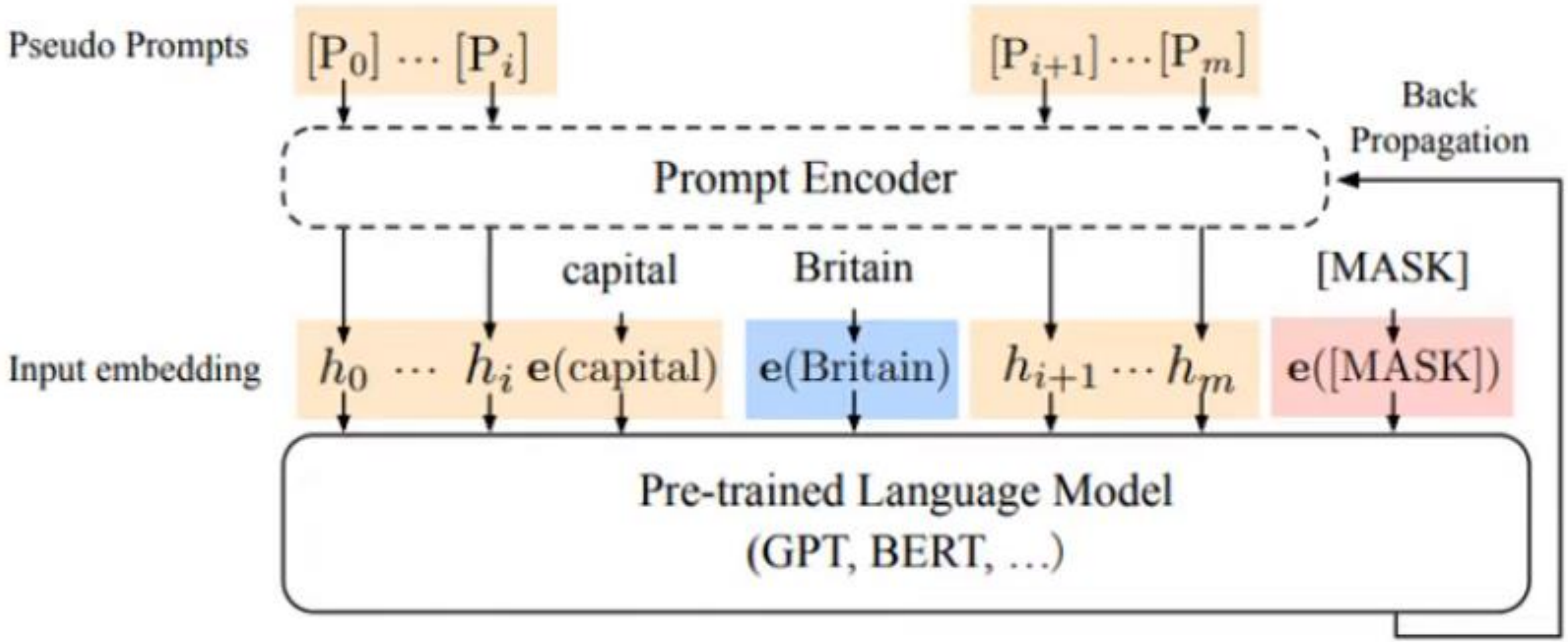
$$\{\mathbf{h}([\mathrm{PROMPT}]), \mathbf{h}([\mathrm{PROMPT}]), \ldots, \mathbf{e}(\mathrm{Britain}), \mathbf{h}([\mathrm{PROMPT}]), \ldots, \mathbf{h}([\mathrm{PROMPT}]), \mathbf{e}([\mathrm{Mask}])\}$$

- h : Pseudo token embedding model (prompt encoder)
  - 2 MLP layers are used by bidirectional LSTM

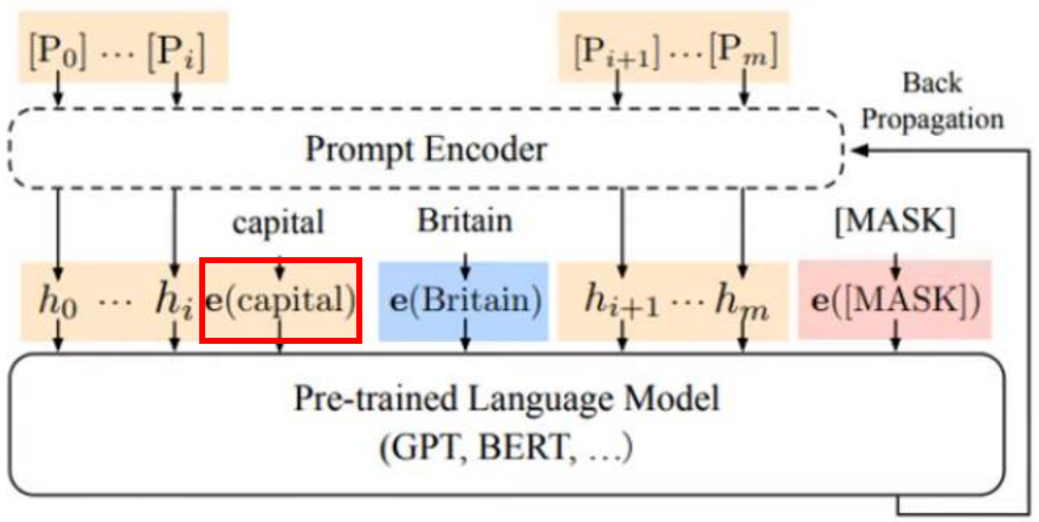
$$h_i = \mathrm{MLP}([\mathrm{LSTM}(h_{0:i}); \mathrm{LSTM}(h_{i:m})])$$

$$\hat{h}_{0:m} = \arg\min_{h} \mathcal{L}(\mathcal{M}(\mathbf{x}, \mathbf{y}))$$

2-(3). P-tuning


Pseudo Prompts
[P0] ··· [Pi]
[Pi+1] ··· [Pm]
Back Propagation
Prompt Encoder
capital
Britain
[MASK]
Input embedding
h0 ··· hi e(capital)
e(Britain)
hi+1 ··· hm
e([MASK])
Pre-trained Language Model
(GPT, BERT, ...)

2-(4). Anchor Token



- Use certain word as anchor token
- Anchor token : capital

  → Predicting a country's capital
- It helps some NLU tasks in the SuperGLUE benchmark

Contents

3-(1). LAMA Dataset

- LAMA Dataset : Estimate knowledge probing task ability
- LAMA-34k : Cover all BERT vocab
- LAMA-29k : Intersection of BERT and GPT vocab
- Baseline
    - MP(Manual Prompt) : Original handcraft prompts from LAMA
    - FT(fine-tuning)

3-(2). Knowledge Probing Precision@1

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**

- Test dataset : LAMA-34
- P-tuning outperforms all the discrete prompt searching baselines

3-(2). Knowledge Probing Precision@1

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)

- Test dataset : LAMA-29
- P-tuning overwhelms the fine-tuning GPT

3-(3). SuperGLUE Dataset

- Using 7 NLU task datasets for SuperGLUE benchmark
- Setting : Fully-supervised and few-shot (train/dev set size : 32)
  - BoolQ, MultiRC : Question and Answering
  - CB, RTE : Textual entailment
  - WiC : Co-Reference resolution
  - COPA : Causal Reasoning
  - WSC : Word Sense Disambiguation

3-(4). Fully-Supervised Setting (base)

Method	BoolQ (Acc.)	CB (Acc.)	CB (F1)	WiC (Acc.)	RTE (Acc.)	MultiRC (EM)	MultiRC (F1a)	WSC (Acc.)	COPA (Acc.)	Avg.
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)

3-(4). Fully-Supervised Setting (large)

Method	BoolQ (Acc.)	CB (F1)	CB (Acc.)	WiC (Acc.)	RTE (Acc.)	MultiRC (EM)	MultiRC (F1a)	WSC (Acc.)	COPA (Acc.)	Avg.
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)

* We report the same results taken from SuperGLUE (Wang et al., 2019b).

3-(5). Few-Shot Setting

Dev size	Method	BoolQ (Acc.)	CB (Acc.)	CB (F1)	WiC (Acc.)	RTE (Acc.)	MultiRC (EM)	MultiRC (F1a)	WSC (Acc.)	COPA (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.

- P-tuning outperforms PET(Dev32) and PET best(Dev32) on all tasks
- Even outperforms SOTA (GPT, PET, iPET) on 4 out of 7 tasks

Contents

4. Conclusion

- P-tuning solves the problems of manual prompts

(large validation set, adversarial prompts, overfitting)

- Also demonstrates that GPT-style performs NLU tasks as well as BERT-style
- General method to increase not only GPT but also BERT performance

Thank You!

GPT Understand, Too