Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning

강동규

DeepSync, South Korea

Introduction: In-Context Learning

Deep Sync

- In-context Learning : 모델 자체는 건드리지 않고 추론할 때 질문을 잘 해서 label이 붙어있지 않은 unseen data를 추론하자.
  - 기본적인 모델이 가지고 있는 용량, 지식에 기반하여 labeling이 되어 있지 않은 모델을 추론해보자.
  - 어떻게 잘 추론할 것인가? → 예시를 제공해서 추론하자
- One-Shot, Few-Shot, Zero-Shot → 제공하는 예시(context)의 개수 차이
- Model이 task와 관련된 token에 높은 확률을 주고 다음 단어를 예측하므로 예시를 통해 context를 이해하고 추론할 수 있다.

Introduction: In-Context Learning

Deep Sync

- **과제:** 국가의 수도에 관한 질문에 답하시오.
- **Few-Shot 학습 설정:** 모델에는 작업을 설명하는 소수의 예제(샷)가 제공된다.
  - 이러한 예는 질문-답변 쌍의 형태로 되어 있다.

1. **질문:** 프랑스의 수도는 무엇입니까? **답변:** 파리.
2. **질문:** 일본의 수도는 무엇입니까? **답변:** 도쿄.
3. **질문:** 캐나다의 수도는 무엇입니까? **답변:** 오타와.

답변

- **질문:** 호주의 수도는 무엇입니까?
- <Answer> → <캔버라>

Introduction: In-Context Learning

Deep Sync

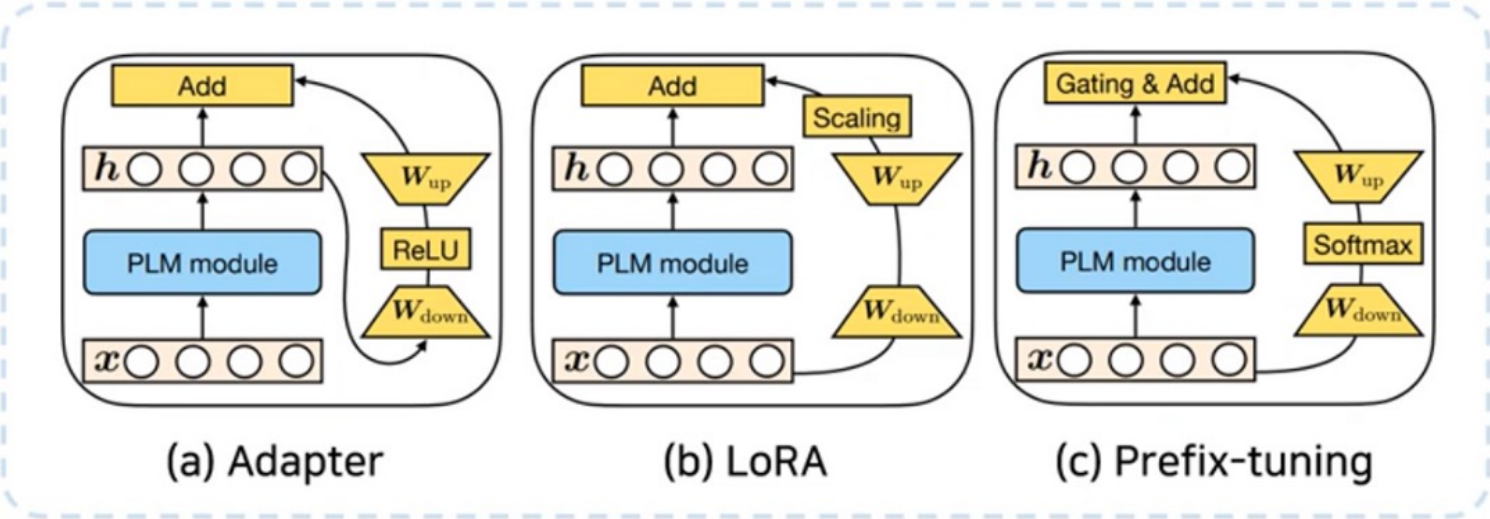
- In-context Learning
  - Min, Sewon, et al. “Rethinking the role of demonstrations: What makes in-context learning work?”, 2022
    - Label이 없는 경우 성능이 크게 하락
    - 부정확한 label, random label을 사용하여도 성능이 크게 저하되지 않음.

  - Input-target pair 예시가 주어져야 하기 때문에 computational cost가 높다. → Key, Value 캐싱해 보완가능
  - Fine-tuning에 비해 성능이 떨어진다.
  - Template에 따라 결과의 변동성이 크다.

Introduction: PEFT (Parameter-Efficient Fine-Tuning)

Deep Sync

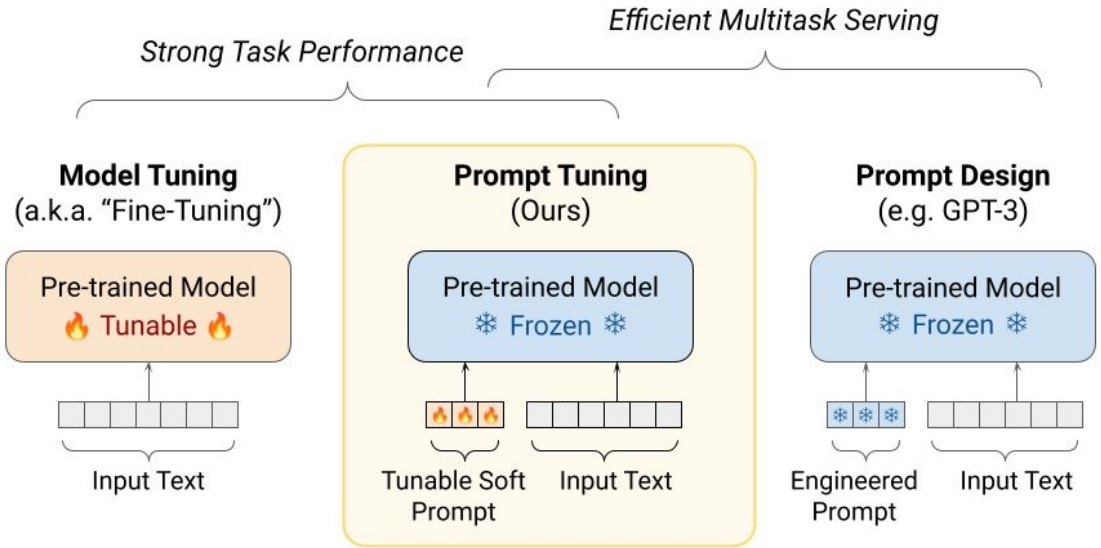
- Parameter-Efficient Fine-Tuning
  - 효과적인 fine-tuning 을 위해, trainable parameter 수를 줄이는 방법



Introduction: PEFT (Parameter-Efficient Fine-Tuning)

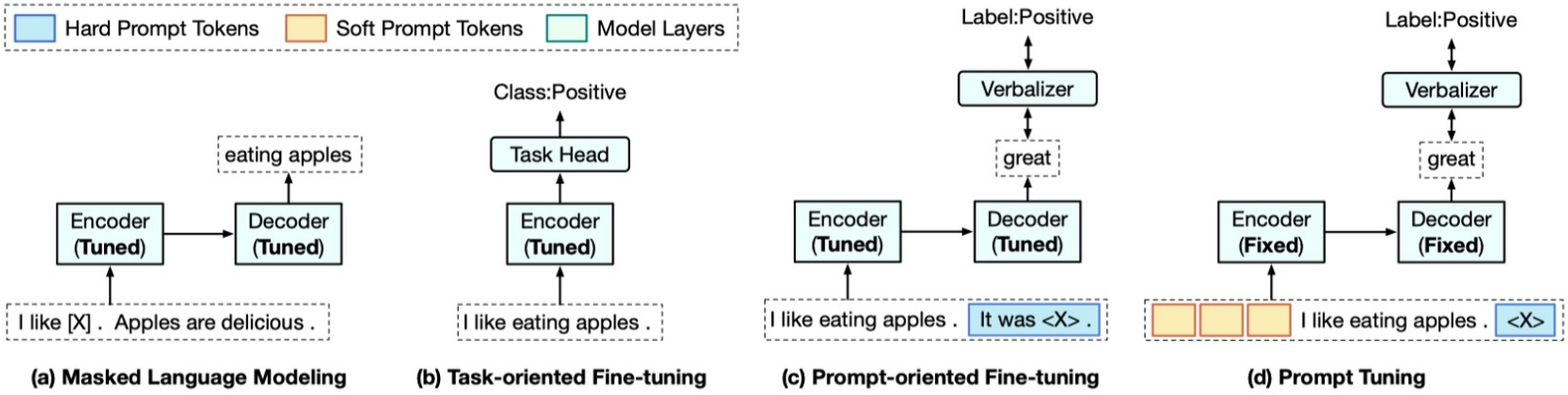
Deep Sync

- Parameter-Efficient Fine-Tuning
  - Prompt tuning : 입력에 k개의 학습 가능한 파라미터인 토큰 임베딩 벡터를 추가하는 방식



Introduction: Few-Shot PEFT

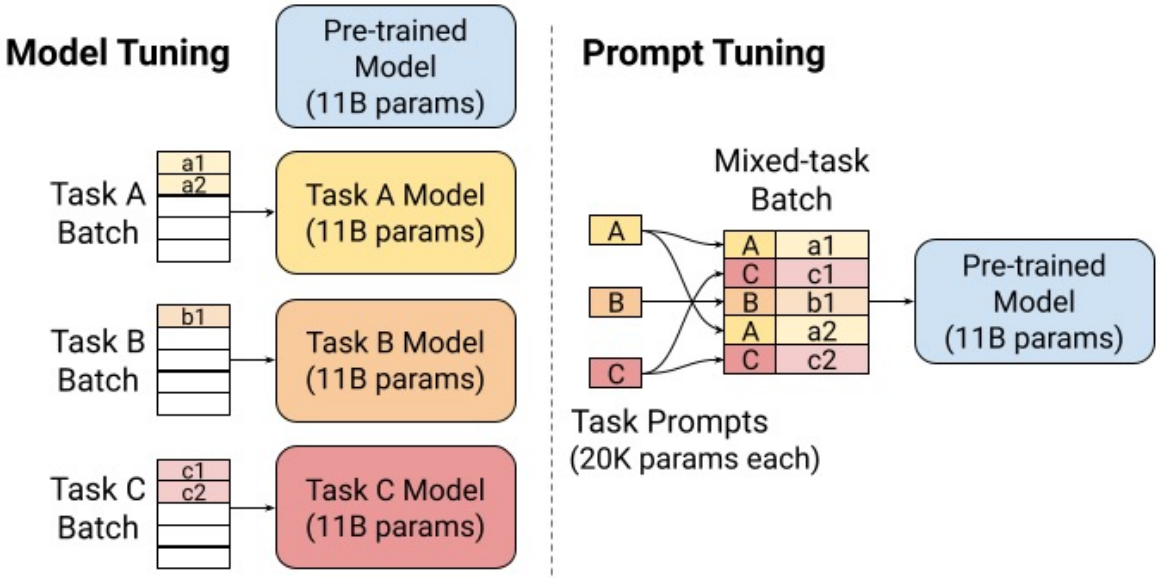
- In-context learning 시 소량의 데이터를 사용
- In-context learning 보다 훨씬 좋은 성능
- Full fine-tuning 보다 훨씬 적은 training cost
- Yuxian Gu, et al. “PPT: Pre-trained Prompt Tuning for Few-shot Learning”
    - Downstream task group을 통해 Soft Prompt를 학습, Prompt Initialization으로 사용하는 방법론 제안



Introduction: PEFT (Parameter-Efficient Fine-Tuning)

Deep Sync

- Prompt Tuning : Mixed-task batch 사용 → Multitask Model
- Labeled data가 매우 적은 경우의 연구가 부족함



Introduction: Few-Shot PEFT

Deep Sync

- Few-Shot ICL에 비해 경쟁력을 얻기 위해서는?

1. Storage, Memory 비용이 들지 않도록 가능한 적은 수의 파라미터 학습
2. Unseen data에 대한 강력한 성능
3. Mixed-task batches가 가능한 모델

➔ IA^3 제시

Idea: IA^3

- IA^3 : **I**nfused **A**daptor by **I**nhibiting and **A**mplifying **I**nner **A**ctivation

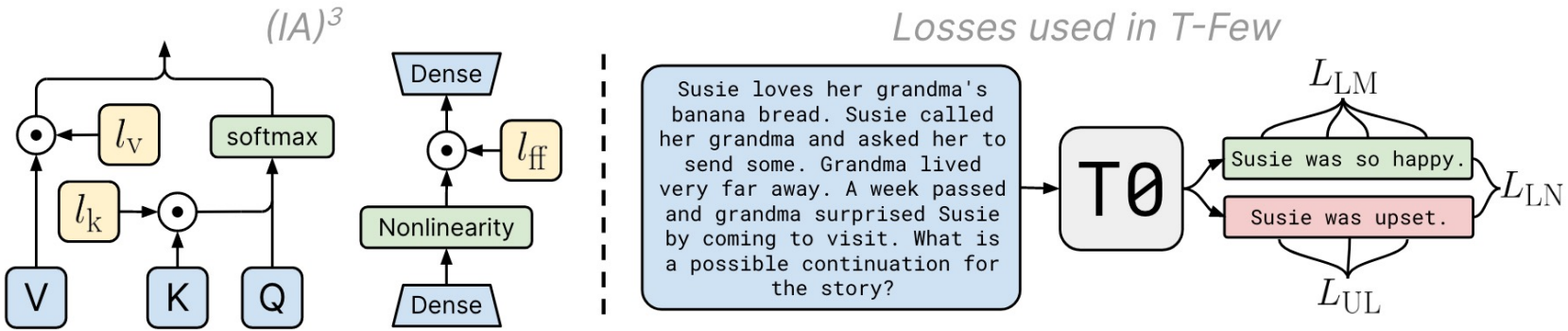
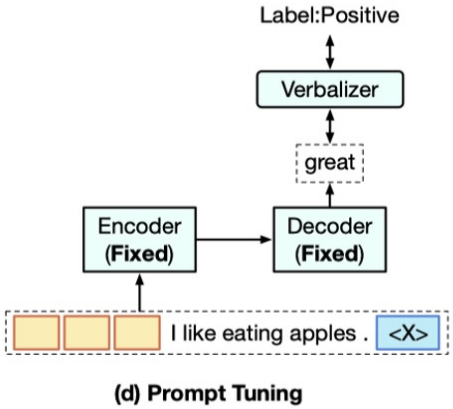


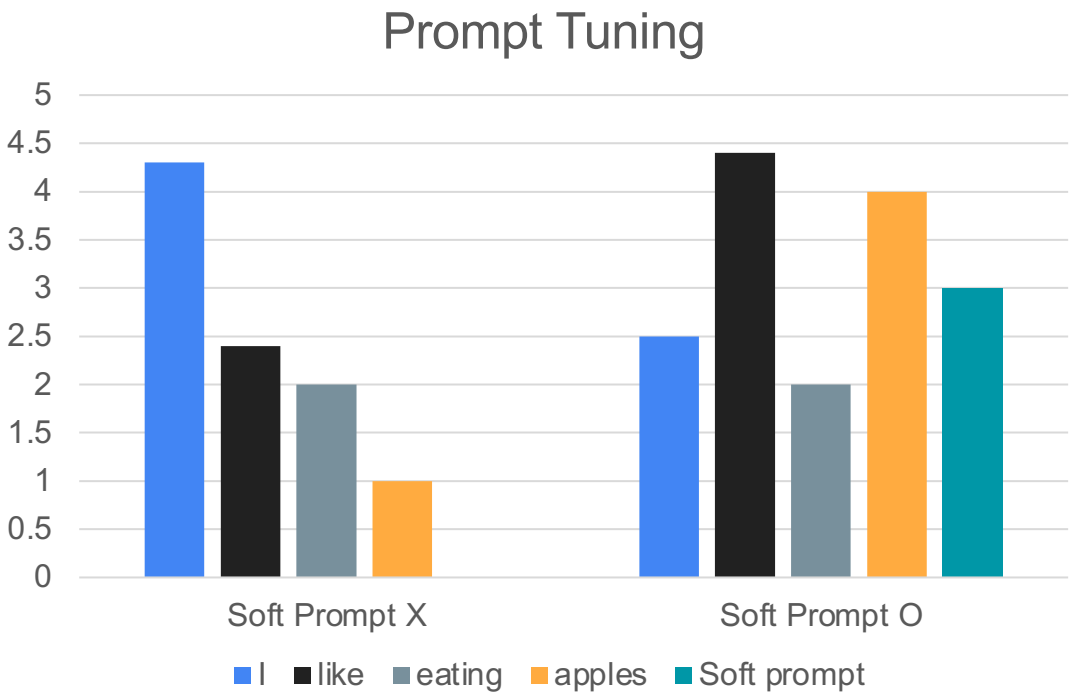
Figure 1: Diagram of `(IA)3` and the loss terms used in the `T-Few` recipe. *Left:* `(IA)3` introduces the learned vectors l_k, l_v, and l_{ff} which respectively rescale (via element-wise multiplication, visualized as $\odot$) the keys and values in attention mechanisms and the inner activations in position-wise feed-forward networks. *Right:* In addition to a standard cross-entropy loss L_{LM}, we introduce an unlikelihood loss L_{UL} that lowers the probability of incorrect outputs and a length-normalized loss L_{LN} that applies a standard softmax cross-entropy loss to length-normalized log-probabilities of all output choices.

Idea: IA^3

- IA^3 : **I**nfused **A**daptor by **I**nhibiting and **A**mplifying **I**nner **A**ctivation
  - Inspired by Prompt Tuning

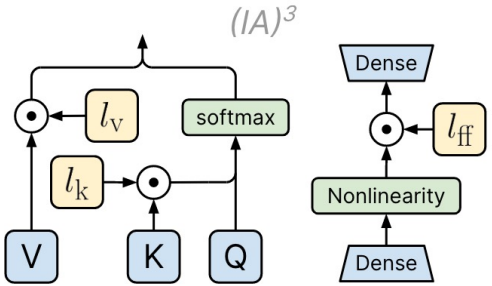


(d) Prompt Tuning

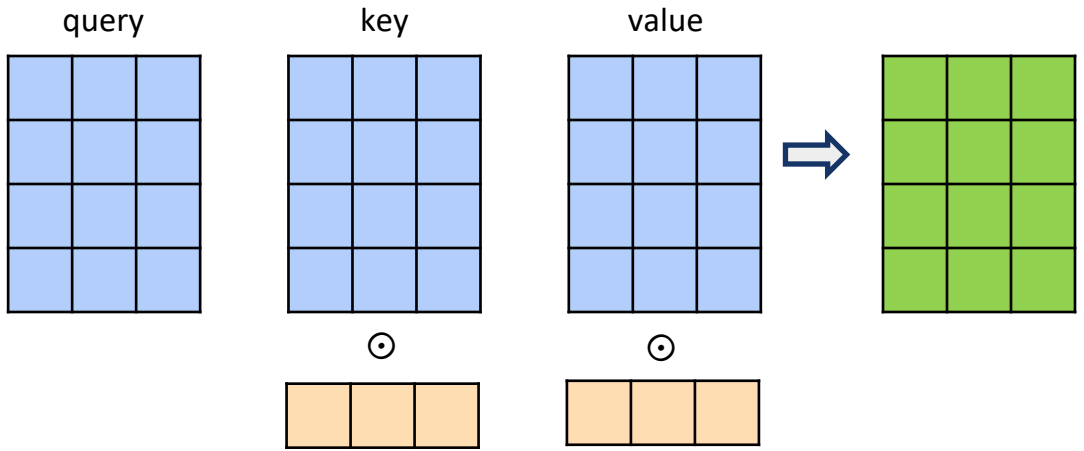


Idea: IA^3

- IA^3 : **I**nfused **A**daptor by **I**nhibiting and **A**mplifying **I**nner **A**ctivation

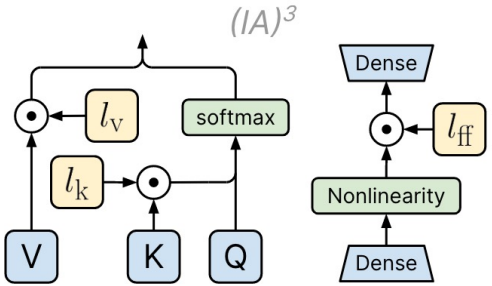


$$\text{softmax}\left(\frac{Q(l_k \odot K^T)}{\sqrt{d_k}}\right)(l_v \odot V)$$

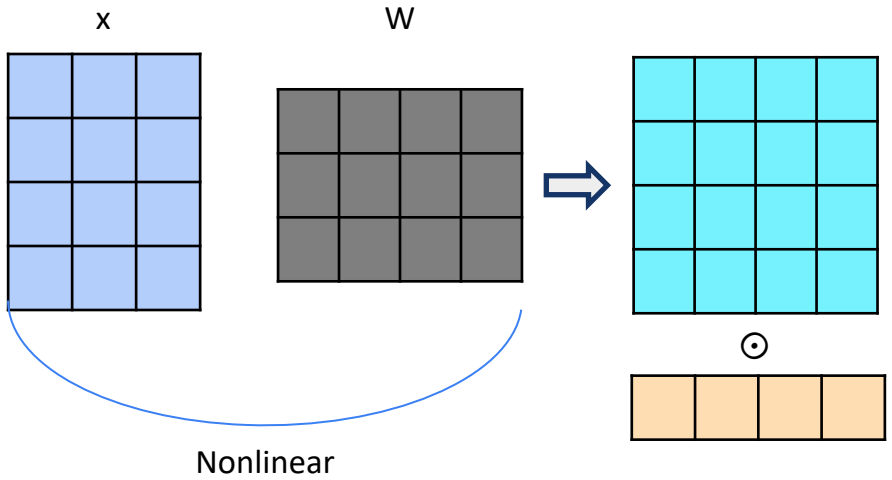


Idea: IA^3

- IA^3 : **I**nfused **A**daptor by **I**nhibiting and **A**mplifying **I**nner **A**ctivation

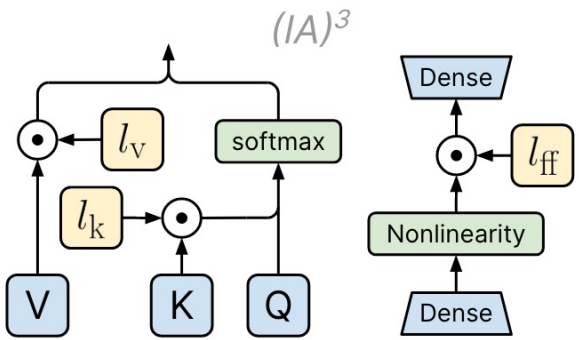


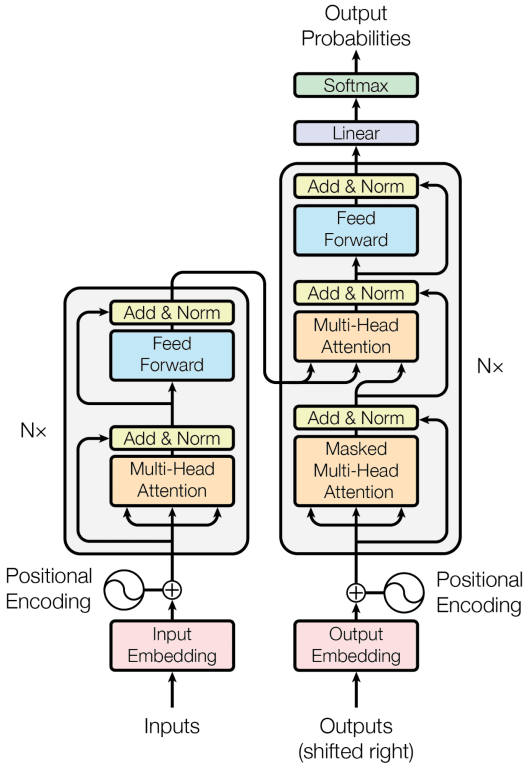
$$l_{ff} \odot \gamma(W_1 x))W_2$$



Idea: IA^3

- IA^3 : **I**nfused **A**daptor by **I**nhibiting and **A**mplifying **I**nner **A**ctivation
  - Numbers of parameters
  - $Encoder * (l_v + l_k + l_{ff}) + Decoder * (2 * (l_v + l_k) + l_{ff})$
- Prompt Tuning은 입력에 포함되는 입력 토큰의 개수를 조정해야 한다.
- 하지만 IA^3 는 조정할 필요가 없다.

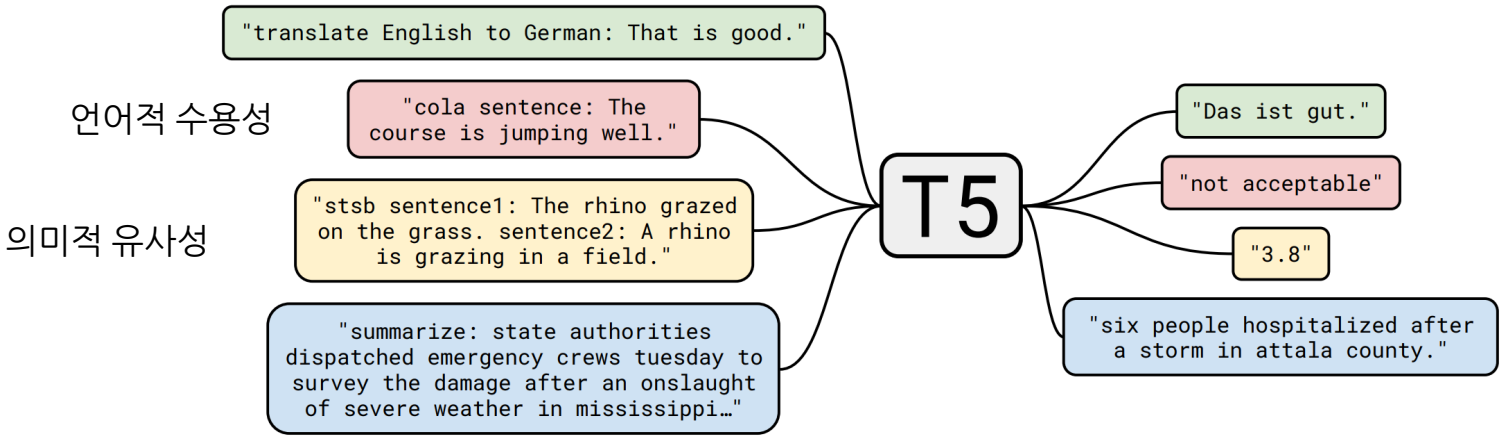




Loss Function

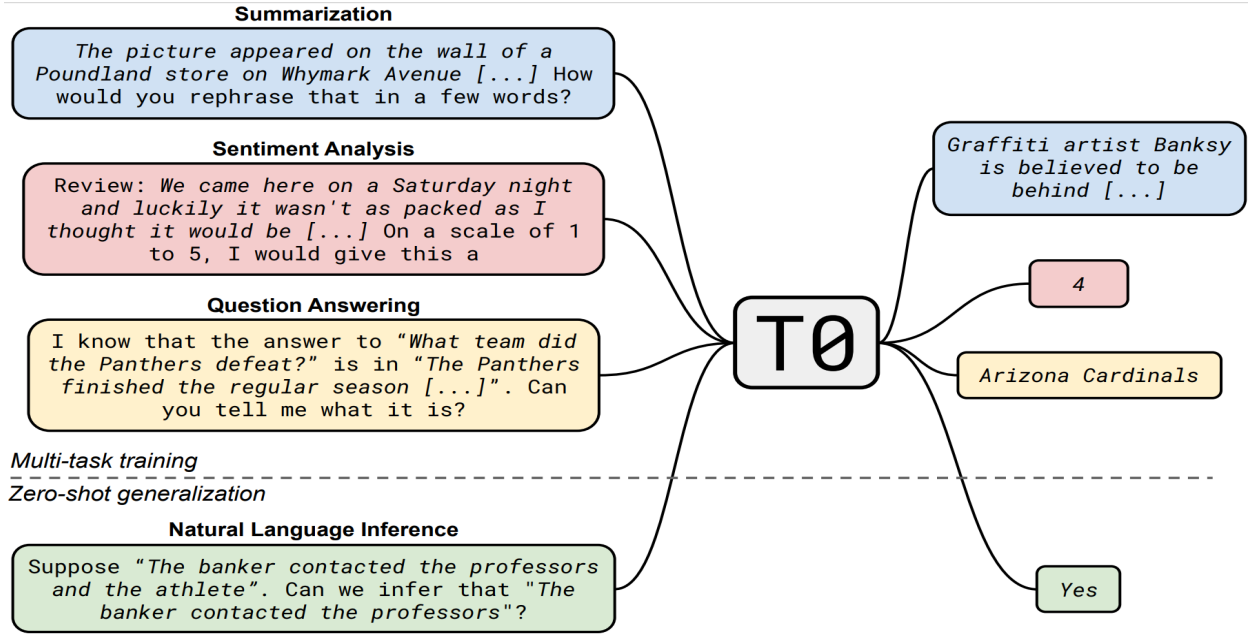
- Base Model : T0 (사전 훈련된 다양한 모델에서 PEFT 방법을 적용한 실험 중 최고의 성능을 보임)
  - T5를 기반으로 Labeling되지 않은 대규모 데이터에 대해 사전학습된 모델
  - Zero-Shot Generalization을 하기 위해 T5를 fine-tuning하여 생성

- T5 : Text-to-Text Transfer Transformer



Loss Function

- Base Model : T0 (사전 훈련된 다양한 모델에서 PEFT 방법을 적용한 실험 중 최고의 성능을 보임)
  - T5를 기반으로 Labeling되지 않은 대규모 데이터에 대해 사전학습된 모델
  - Zero-Shot Generalization을 하기 위해 T5를 fine-tuning하여 생성



Loss Function

Deep Sync

- Unlikelihood Training
- T0 의 평가 : 모든 가능한 label 문자열에 대해 log-likelihood를 이용해 확률을 순위로 매기고 가장 높은 확률을 가진 것이 정답일 경우 모델의 예측이 올바른 것으로 간주
    - ➔ 올바른 선택, 잘못된 선택을 모두 고려
    - ➔ 잘못된 선택은 낮은 확률을 전달하자
    - N : 잘못된 선택의 개수, $T^{(n)}$: 전체 토큰 수

$$L_{\mathrm{UL}} = -\frac{\sum_{n=1}^{N}\sum_{t=1}^{T^{(n)}} \log(1 - p(\hat{y}_i^{(n)}|\mathbf{x}, \hat{y}_{<t}^{(n)}))}{\sum_{n=1}^{N} T^{(n)}}$$

$$L_{LM} = -\frac{1}{T}\sum_{T} \log p(y_t|\mathbb{x}, y_{<t})$$

Loss Function

Deep Sync

- Length Normalization
- 예제에 따라 길이가 다른 경우가 많을 것. 이때, 각 토큰에 할당된 모델의 확률이 <= 1 이기 때문에 짧은 문장이 더 좋다고 생각할 수 있음.

- Ex) Q. 오늘 날씨는 어때?
  - A1: 좋아.(0.7)
  - A2: 하늘이(0.2) * 맑고(0.6) 햇빛이(0.4) * 쨍쨍해서(0.5) * 좋아.(0.7) = 0.0168
  - A3: 하늘이 (0.2) * 맑고(0.6) * 햇빛이(0.4) * 쨍쨍하면서(0.6) * 바람도(0.8) * 잘(0.6) * 불어서(0.4) * 좋아. (0.9)

Loss Function

Deep Sync

- Length Normalization
- 문장 길이를 정규화
- y : 올바른 선택, $\hat{y}$: 잘못된 선택, $\beta(x, y)$: 문장 길이로 정규화한 log-probablility

$$L_{\mathrm{LN}} = -\log \frac{\exp(\beta(\mathbf{x}, \mathbf{y}))}{\exp(\beta(\mathbf{x}, \mathbf{y})) + \sum_{n=1}^{N} \exp(\beta(\mathbf{x}, \hat{\mathbf{y}}^{(n)}))}$$

Loss Function

Deep Sync

$$Loss\ Function = L_{LM} + L_{Ul} + L_{LN}$$

Experiments

Deep Sync

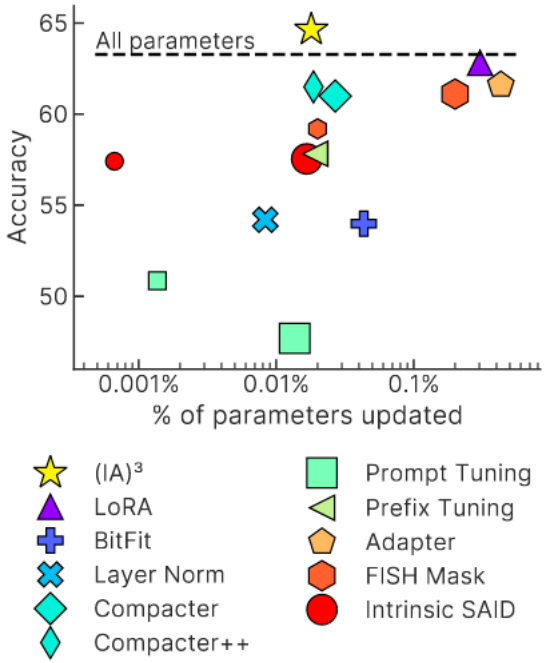
- T-Few Model : T0 + pre-training IA^3
- Loss : $\mathrm{L_{LM}} + \mathrm{L_{Ul}} + \mathrm{L_{LN}}$
- Pre-Training IA^3
  - 100,000 steps, with a batch size of 16
  - T0 학습할 때 사용된 T0 multitask mixture dataset을 사용
- Training T-Few
  - 1,000 steps, with a batch size of 8

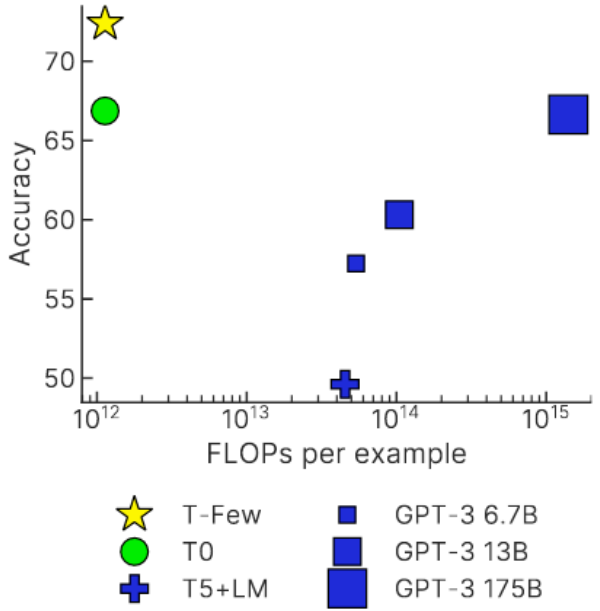
Difference between Model

Deep Sync

T-Few	- T0 + IA^3 + Loss Function - T0 기반 PEFT
T0	- T5 + Multitask Prompt Learning - Zero-Shot Generalization에 좋은 성능
T5	- Text-to-Text로 Multitask 가능
GPT-3	- In-context Learning에 좋은 성능을 보임

Result


All parameters
65
60
55
50
Accuracy
0.001%
0.01%
0.1%
% of parameters updated
(IA)³
LoRA
BitFit
Layer Norm
Compacter
Compacter++
Prompt Tuning
Prefix Tuning
Adapter
FISH Mask
Intrinsic SAID


70
65
60
55
50
Accuracy
10^{12}
10^{13}
10^{14}
10^{15}
FLOPs per example
T-Few
T0
T5+LM
GPT-3 6.7B
GPT-3 13B
GPT-3 175B

Result

Deep Sync

- FLOPs
  - 모델 파라미터 수 * 토큰 수

- T-Few 는 학습 필요하지만 시간 짧고 성능 좋음

Method	Inference FLOPs	Training FLOPs	Disk space	Acc.
T-Few	1.1e12	2.7e16	4.2 MB	72.4%
T0 [1]	1.1e12	0	0 B	66.9%
T5+LM [14]	4.5e13	0	16 kB	49.6%
GPT-3 6.7B [4]	5.4e13	0	16 kB	57.2%
GPT-3 13B [4]	1.0e14	0	16 kB	60.3%
GPT-3 175B [4]	1.4e15	0	16 kB	66.6%

Table 1: Accuracy on held-out T0 tasks and computational costs for different few-shot learning methods and models. T-Few attains the highest accuracy with 1,000× lower computational cost than ICL with GPT-3 175B. Fine-tuning with T-Few costs about as much as ICL on 20 examples with GPT-3 175B.

Conclusion

Deep Sync

- T-Few Model : T0 + pre-training IA^3 + $\mathrm{L_{UI}} + \mathrm{L_{LN}}$
- 작은 파라미터 업데이트하는 방식
- 하이퍼 파라미터 튜닝을 최소화
- Mixed-Task Batch 사용 가능
- Unseen Data에 대해 강력한 성능

Deep Sync

Thank you for your time