Prompt Tuning 기법 리뷰
(The Power of Scale for Parameter-Efficient Prompt Tuning)

에이아이스쿨(AISchool) 대표
양진호 (솔라리스)
http://aischool.ai
http://solarisailab.com

Prompt Tuning Paper

- Lester, Brian, Rami Al-Rfou, and Noah Constant. "The power of scale for parameter-efficient prompt tuning." arXiv preprint arXiv:2104.08691 (2021).
- https://arxiv.org/pdf/2104.08691.pdf

The Power of Scale for Parameter-Efficient Prompt Tuning

Brian Lester* **Rami Al-Rfou** **Noah Constant**
Google Research
{brianlester,rmyeid,nconstant}@google.com

Abstract

In this work, we explore "prompt tuning," a simple yet effective mechanism for learning "soft prompts" to condition frozen language models to perform specific downstream tasks. Unlike the discrete text prompts used by GPT-3, soft prompts are learned through backpropagation and can be tuned to incorporate signals from any number of labeled examples. Our end-to-end learned approach outperforms GPT-3's few-shot learning by a large margin. More remarkably, through ablations on model size using T5, we show that prompt tuning becomes more competitive with scale: as models exceed billions of parameters, our method "closes the gap" and matches the strong performance of model tuning (where all model

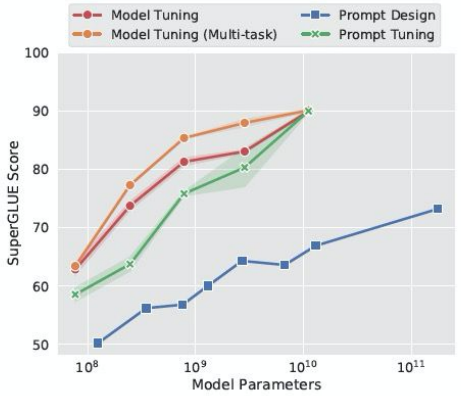
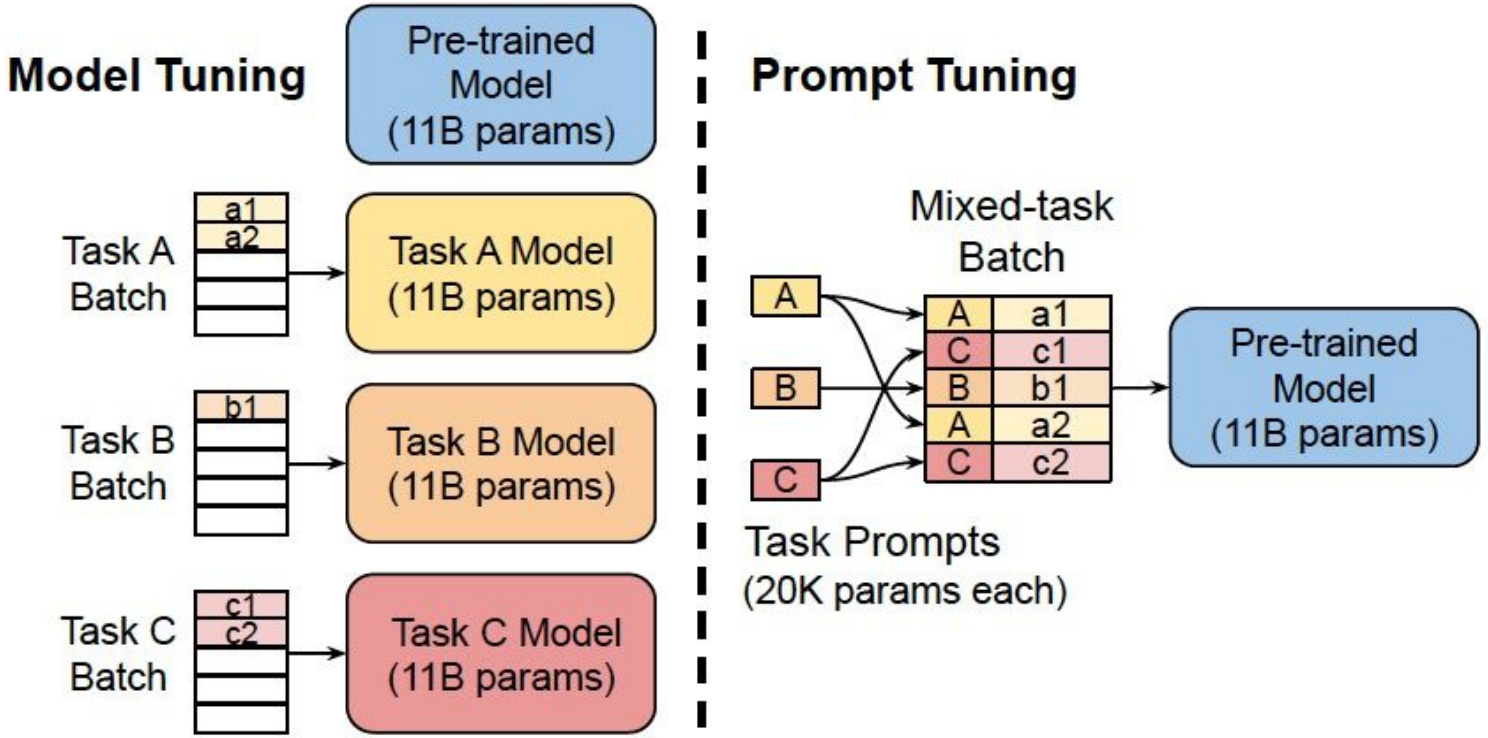


Figure 1: Standard **model tuning** of T5 achieves strong performance, but requires storing separate copies of the model for each end task. Our **prompt tuning** of T5

Overview

- Prompt Tuning의 핵심 idea : fine-tuning 과정시 pre-train이 끝난 파라미터 w_0를 고정하고 각 Task에 적합한 정보로 Fine-Tuning 가능한 Embedding을 input 앞에 추가함



Prompt Tuning Paper

- Prompt Tuning Paper를 같이 살펴보면서 Prompt Tuning의 디테일한 내용들을 살펴봅시다!

Abstract

- 이 연구에서는 "프롬프트 튜닝"이라는 간단하면서도 효과적인 메커니즘을 탐구합니다.
- 이는 특정 하위 작업을 수행하도록 고정된 언어 모델을 조건화하기 위해 "소프트 프롬프트"를 학습하는 방법입니다.
- GPT-3에서 사용된 이산 텍스트 프롬프트와 달리, 소프트 프롬프트는 역전파를 통해 학습되며, 레이블이 붙은 예시의 수와 상관없이 신호를 통합하여 조정할 수 있습니다.
- 우리의 종단 간 학습 접근법은 GPT-3의 few-shot learning보다 큰 차이로 우월합니다. 더 놀라운 것은, T5를 사용한 모델 크기에 대한 제거를 통해, 프롬프트 튜닝이 규모와 경쟁력을 가지게 됩니다: 모델이 수십억 개의 매개변수를 초과하면, 우리의 방법은 "격차를 좁히고" 모델 튜닝의 강력한 성능과 일치합니다(모든 모델 가중치가 조정됩니다).
- 이러한 발견은 큰 모델들이 공유하고 제공하기 cost가 많이 들기 때문에 여러 하위 작업에 하나의 고정된 모델을 재사용하는 능력이 이 부담을 줄일 수 있다는 점에서 특히 관련이 있습니다. 우리의 방법은 최근 제안된 "접두사 튜닝"(prefix-tuning)에 대한 간소화로 볼 수 있으며, 이와 다른 유사한 접근법과의 비교를 제공합니다. 마지막으로, 소프트 프롬프트로 고정된 모델을 조건화하면 도메인 이전에 대한 견고성의 이점을 얻을 수 있으며, "프롬프트 앙상블링"을 효과적으로 가능하게 합니다.

1. Introduction

- 사전 훈련된 대형 언어 모델의 광범위한 성공으로 인해 이러한 일반적인 목적의 모델을 하위 작업에 적용하기 위한 다양한 기술이 등장하게 되었습니다.
- ELMo (Peters et al., 2018)는 사전 훈련된 모델을 고정하고 그것의 계층별 표현에 대한 작업 특정 가중치를 학습하는 것을 제안하였습니다.
- 그러나 GPT (Radford et al., 2018)와 BERT (Devlin et al., 2019) 이후로 주요한 적용 기술은 모델 튜닝(또는 "fine tuning")이 되었으며, 이는 Howard와 Ruder (2018)에 의해 제안된 것처럼, 모든 모델 파라미터가 적응 과정에서 조정되는 방식입니다.

1. Introduction

- 불행하게도, 프롬프트 기반 적응에는 몇 가지 주요한 단점이 있습니다.
- 작업 설명은 오류가 발생하기 쉽고 인간의 개입이 필요하며, 프롬프트의 효과는 모델 입력에 들어갈 수 있는 조절 텍스트의 양으로 제한됩니다.
- 결과적으로, 하위 작업의 품질은 아직도 조정된 모델의 품질보다 훨씬 뒤쳐집니다.
- 예를 들어, GPT-3 175B의 SuperGLUE에서의 퓨샷 성능은 T5-XXL(Raffel 등, 2020)이 미세 조정된 성능(89.3)에 비해 17.5 포인트 떨어진 71.8이며, 이는 파라미터가 16배 더 많음에도 불구하고 그렇습니다.

1. Introduction

- 최근에는 프롬프트 디자인을 자동화하기 위한 여러 노력이 제안되었습니다. Shin 등(2020)은 하위 응용 프로그램 훈련 데이터에 의해 안내되는 단어의 이산 공간을 검색하는 알고리즘을 제안합니다. 이 기술은 수동 프롬프트 디자인을 능가하지만, 모델 튜닝에 대한 차이는 여전히 있습니다.
- Li와 Liang(2021)은 "prefix tuning"을 제안하고 생성적 작업에서 강력한 결과를 보여줍니다. 이 방법은 모델 파라미터를 동결하고, 튜닝 중에 오류를 인코더 스택의 각 계층에, 입력 계층을 포함하여 앞에 추가된 prefix 활성화로 역전파합니다.

1. Introduction

- 이 논문에서는 언어 모델을 적응시키기 위한 더욱 간소화된 방법으로 프롬프트 튜닝을 제안합니다.
- 우리는 전체 사전 훈련된 모델을 동결하고 입력 텍스트 앞에 추가로 k개의 조정 가능한 토큰만을 허용합니다.
- 이 "소프트 프롬프트"는 종단간(end-to-end)으로 훈련되며, 전체 레이블이 지정된 데이터셋의 신호를 집약할 수 있습니다.
- 이를 통해 우리의 방법은 퓨샷 프롬프트를 능가하고 모델 튜닝과의 품질 격차를 줄일 수 있습니다(그림 1).
- 동시에, 단일 사전 훈련된 모델이 모든 하위 작업에 재사용되므로, 동결된 모델의 효율적인 제공 이점을 유지합니다(그림 2).

1. Introduction

- 그림 1이 보여주는 것처럼, 프롬프트 튜닝은 규모와 함께 경쟁력을 더욱 갖추게 됩니다.

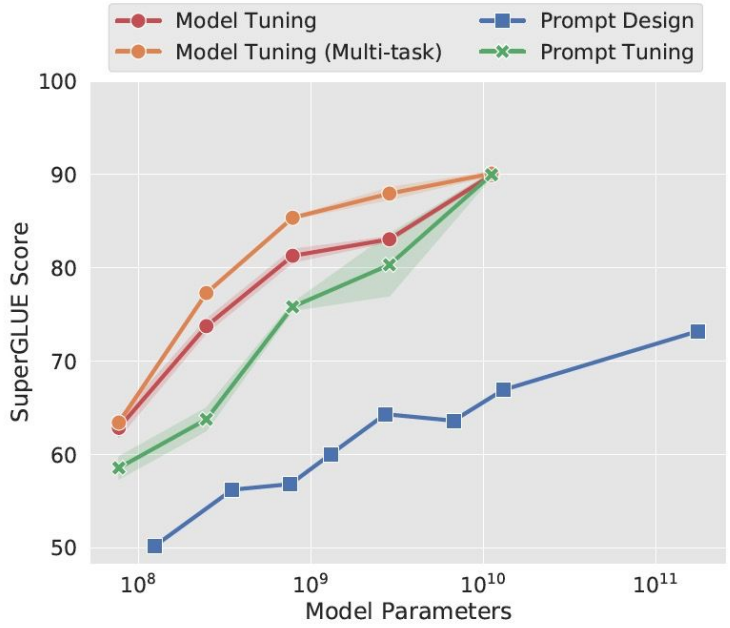


Figure 1: Standard **model tuning** of T5 achieves strong performance, but requires storing separate copies of the model for each end task. Our **prompt tuning** of T5 matches the quality of model tuning as size increases, while enabling the reuse of a single frozen model for all tasks. Our approach significantly outperforms few-shot **prompt design** using GPT-3. We show mean and standard deviation across 3 runs for tuning methods.

2. Prompt Tuning

- T5(Raffel 등, 2020)의 "텍스트에서 텍스트로" 접근법을 따라, 우리는 모든 작업을 텍스트 생성으로 변환합니다.
- X가 토큰의 시리즈이고 y가 단일 클래스 레이블인 경우, 입력에 대한 출력 클래스의 확률로 분류를 모델링하는 대신, Pr(y|X) 이제 우리는 조건부 생성으로 그것을 모델링합니다.
- 여기서 Y는 클래스 레이블을 나타내는 토큰의 시퀀스입니다.
- T5는 분류를 Pr(Y|X)로 모델링하며, 이는 그것의 인코더와 디코더를 구성하는 트랜스포머(Vaswani 등, 2017)의 가중치로 매개변수화됩니다.

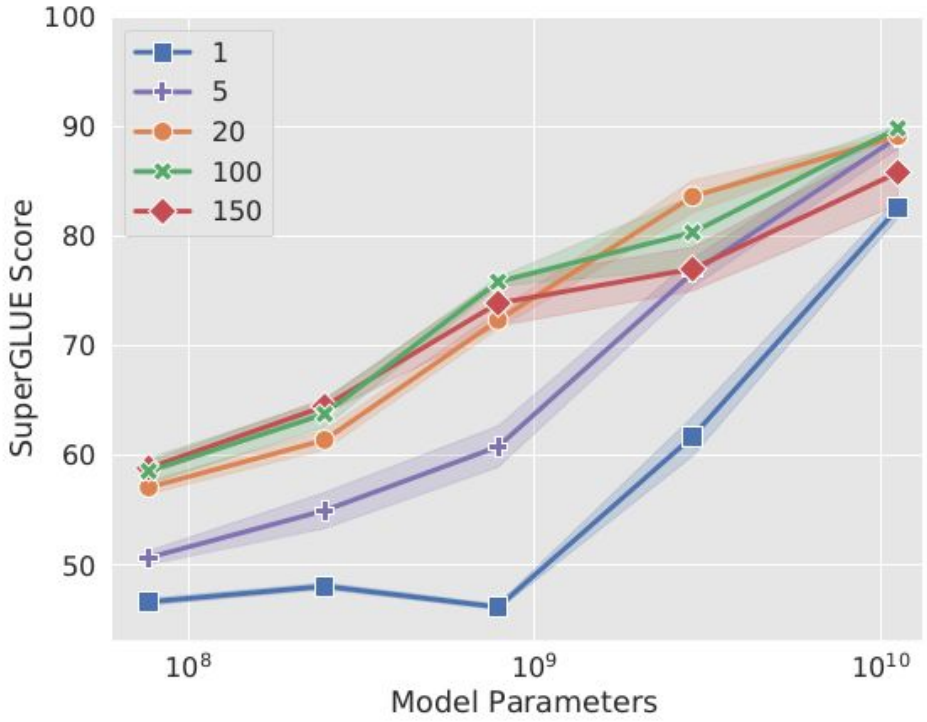
2.1 Design Decisions

- 프롬프트 표현을 초기화하는 데는 여러 가지 방법이 있습니다.
- 가장 간단한 방법은 무작위 초기화를 사용하여 처음부터 훈련하는 것입니다.
- 더 정교한 옵션은 각 프롬프트 토큰을 모델의 어휘에서 추출된 임베딩으로 초기화하는 것입니다.
- 개념적으로, 우리의 소프트 프롬프트는 입력 앞의 텍스트와 같은 방식으로 동결된 네트워크의 동작을 조절하므로, 단어와 같은 표현이 좋은 초기화 지점으로 사용될 수 있다는 것을 알 수 있습니다.
- 분류 작업의 경우, 세 번째 옵션은 출력 클래스를 열거하는 임베딩으로 프롬프트를 초기화하는 것입니다. 이는 Schick과 Schütze (2021)의 "verbalizers"와 유사합니다. 우리는 모델이 출력에서 이 토큰들을 생성하기를 원하므로, 유효한 대상 토큰의 임베딩으로 프롬프트를 초기화하면 모델이 적절한 출력 클래스로 출력을 제한하도록 기울이게 됩니다.

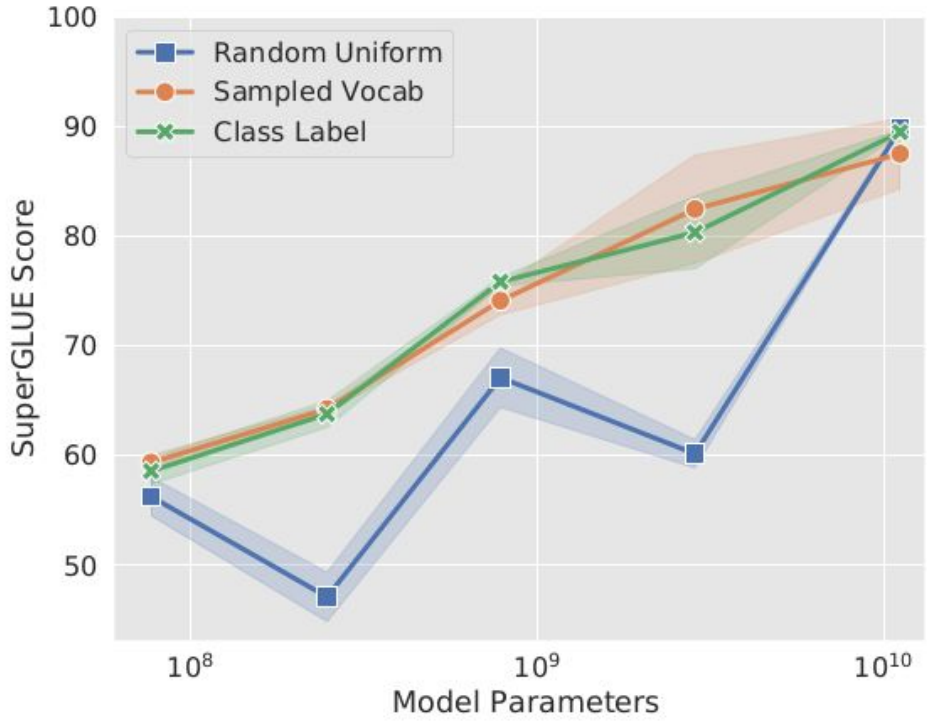
2.1 Design Decisions

- 또 다른 디자인 고려사항은 프롬프트의 길이입니다.
- 우리 방법의 파라미터 비용은 **EP**이며, 여기서 **E**는 토큰 임베딩 차원이고 **P**는 프롬프트의 길이입니다.
- 프롬프트가 짧을수록 조정해야 하는 새로운 파라미터가 적으므로, 여전히 잘 수행되는 최소 길이를 찾으려고 합니다.

Results



(a) Prompt length



(b) Prompt initialization

4. Comparison to Similar Approaches

- Li와 Liang (2021)은 "접두사 튜닝"(prefix-tuning)을 제안합니다:
- 모든 트랜스포머 계층에서 앞에 추가되는 일련의 접두사를 학습하는 것입니다. 이는 모든 네트워크 계층에서 예시를 걸쳐 고정된 트랜스포머 활성화를 학습하는 것과 유사합니다.
- 반면, 프롬프트 튜닝은 임베디드 입력에 앞에 추가되는 단일 프롬프트 표현을 사용합니다. 더 적은 매개변수를 필요로 하는 것을 넘어, 우리의 접근법은 입력 예제에 의해 문맥화된 중간 계층의 작업 표현을 트랜스포머가 업데이트하도록 허용합니다.
- 그들의 연구는 GPT-2 (Radford 등, 2019)와 BART (Lewis 등, 2020)를 기반으로 하며, 반면에 우리의 연구는 T5에 중점을 둔다. 또한 모델 크기가 증가함에 따라 성능과 디자인 선택에 대한 견고성의 변화를 조사한다. BART를 사용할 때, 접두사 튜닝은 인코더와 디코더 네트워크 양쪽에 접두사를 포함하며, 프롬프트 튜닝은 인코더에만 프롬프트를 필요로 합니다. Li와 Liang (2021)은 또한 학습을 안정화하기 위해 접두사의 재파라미터화에 의존하며, 이는 훈련 중에 많은 수의 매개변수를 추가합니다. 반면에 우리의 구성은 이 재파라미터화를 필요로 하지 않으며, SuperGLUE 작업과 모델 크기에 걸쳐 견고합니다.

Overview

- Prefix-tuning의 핵심 idea : fine-tuning 과정시 pre-train이 끝난 파라미터 w_0를 고정하고 Prefix-tuning 세팅의 새로운 파라미터를 학습시킴

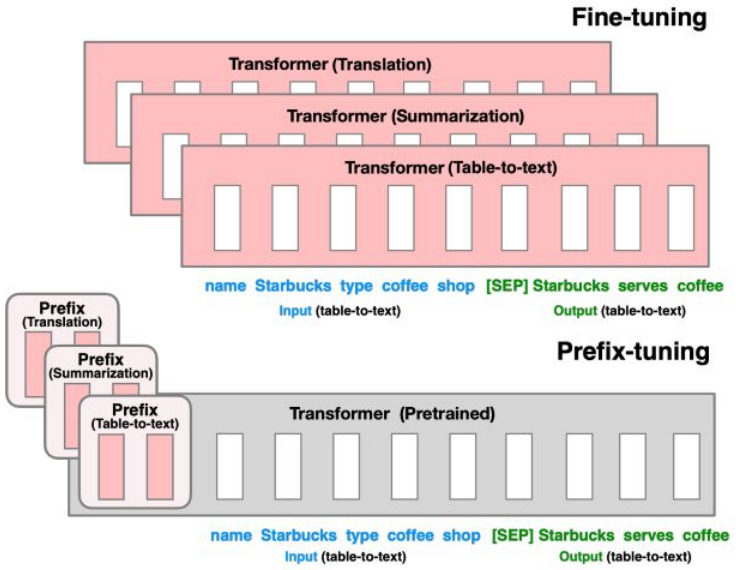


Figure 1: Fine-tuning (top) updates all Transformer parameters (the red Transformer box) and requires storing a full model copy for each task. We propose prefix-tuning (bottom), which freezes the Transformer parameters and only optimizes the prefix (the red prefix blocks). Consequently, we only need to store the prefix for each task, making prefix-tuning modular and space-efficient. Note that each vertical block denote transformer activations at one time step.

4. Comparison to Similar Approaches

- Liu 등(2021)은 "P-Tuning"을 제안합니다.
- 이 방법에서는 학습 가능한 연속적인 프롬프트가 사람의 디자인을 기반으로 한 패턴을 사용하여 임베디드 입력 전체에 교차되게 됩니다.
- 우리의 접근법은 이 복잡함을 제거하고 단순히 프롬프트를 입력에 앞에 추가합니다. SuperGLUE 결과를 높이기 위해서는 P-튜닝을 모델 튜닝과 함께 사용해야 합니다.
- 즉, 모델은 프롬프트와 주요 모델 매개변수 모두를 공동으로 업데이트해야 합니다. 반면, 우리의 접근법은 원래의 언어 모델을 고정시켜 둡니다.

Overview

- P-Tuning의 핵심 idea : fine-tuning 과정시 pre-train이 끝난 파라미터 w_0를 고정하고 Template 생성을 위한 Prompt Encoder만을 학습시킴

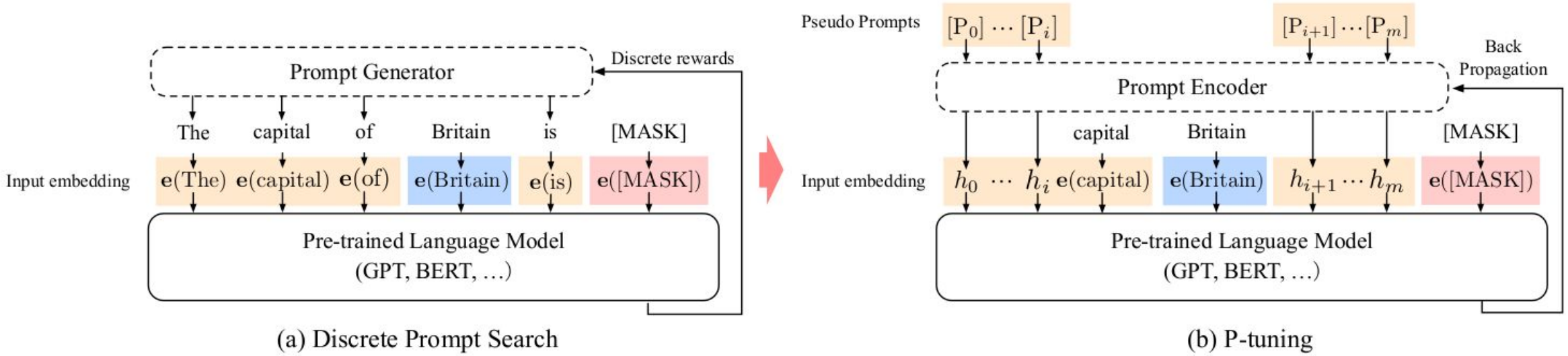


Figure 2. An example of prompt search for "The capital of Britain is [MASK]". Given the context (blue zone, "Britain") and target (red zone, "[MASK]"), the orange zone refer to the prompt tokens. In (a), the prompt generator only receives discrete rewards; on the contrary, in (b) the pseudo prompts and prompt encoder can be optimized in a differentiable way. Sometimes, adding few task-related anchor tokens (such as "capital" in (b)) will bring further improvement.

Results

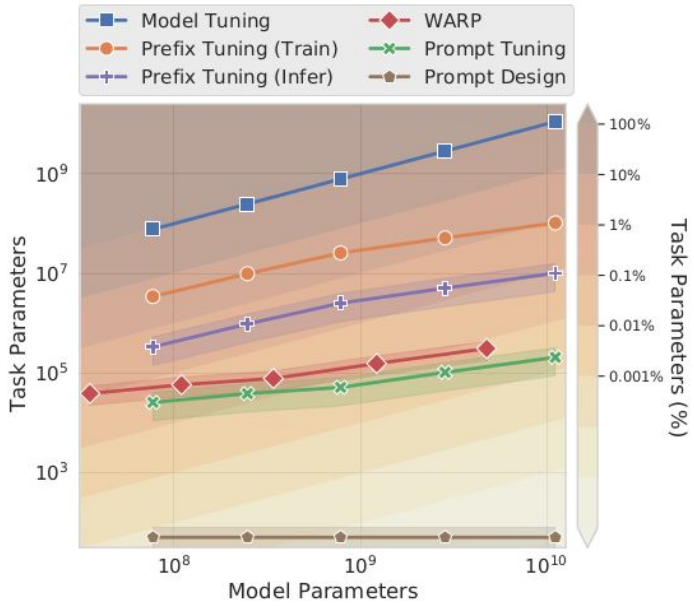


Figure 4: Parameter usage of various adaptation techniques, fixing architecture to T5.1.1 and prompt/prefix length to 1–100 tokens (bands show mean and stddev). **Model Tuning**: All parameters are task-specific. **Prefix Tuning**: Activations are tuned in the prefix of each layer, requiring 0.1–1% task-specific parameters for inference, but more are used for training. **WARP**: Task parameters are reduced to under 0.1% by only tuning input and output layers. **Prompt Tuning**: Only prompt embeddings are tuned, reaching under 0.01% for most model sizes. **Prompt Design**: Only a sequence of prompt IDs (500–2000 tokens) is required.

8. Conclusion

- 이 논문에서는 프롬프트 튜닝(Prompt Tuning)이 동결된 사전 훈련된 언어 모델을 하위 작업에 적응시키는 유용한 기술임을 보였습니다.
- 인기 있는 SuperGLUE 벤치마크에서, 그 작업 성능은 전통적인 모델 튜닝과 견줄만하며, 모델 크기가 증가함에 따라 그 차이가 사라집니다.
- 제로샷 도메인 전송에서는 프롬프트 튜닝이 일반화에 향상된 결과를 가져온다는 것을 발견했습니다. 이는 일반적인 언어 이해 파라미터를 동결하고 하위 학습을 가벼운 파라미터 발자국으로 제한함으로써 특정 도메인에 과적합하는 것을 피할 수 있음을 합리적으로 나타냅니다.
- 작업 품질 지표를 넘어, 우리는 저장 및 제공 비용 측면에서 동결된 사전 훈련된 모델로의 이동의 매력에 대해 논의했습니다.
- 이런 이동은 효율적인 다중 작업 제공 뿐만 아니라, 효율적이고 높은 성능의 프롬프트 앙상블을 가능하게 합니다.
- 앞으로, 작업을 정의하는 파라미터와 일반 언어 모델링 파라미터를 구분하는 것은 새로운 연구의 많은 방향을 열어놓는 흥미로운 단계라고 생각합니다.