MAdapter: 효율적인 중간 층 도입을 통한 Adapter 구조 개선

김진현°[1], 김태욱†[1]
[1]한양대학교 AI응용학과
{kimjinhye0n,kimtaeuk}@hanyang.ac.kr

MAdapter: A Refinement of Adapters by Augmenting Efficient Middle Layers

Jinhyeon Kim°[1], Taeuk Kim†[1]
[1]Department of Artificial Intelligence Application, Hanyang University

요 약

최근 거대 언어모델의 등장과 동시에, 많은 매개변수를 효과적으로 학습하는 방법인 효율적인 매개변수 미세조정(Parameter Efficient Fine-Tuning) 연구가 활발히 진행되고 있다. 이 중에서 Adapter는 사전학습 언어모델(Pre-trained Language Models)에 몇 개의 추가 병목 구조 모듈을 삽입하여 이를 학습하는 방식으로, 등장한 이후 다양한 연구 영역에서 주목받고 있다. 그러나 몇몇 연구에서는 병목 차원을 증가시켜 미세 조정보다 더 나은 성능을 얻는다는 주장이 나오면서, 원래의 의도와는 다른 방향으로 발전하고 있다는 의견도 있다. 이러한 맥락에서, 본 연구에서는 기존의 Adapter 구조를 개선한 MAdapter를 제안한다. MAdapter는 본래 Adapter에 중간 층을 추가하되 학습 가능한 매개변수의 수는 오히려 줄이는 방법으로, 전체 매개변수 수 대비 1% 내외 만을 학습에 활용하며, Adapter 대비 절반 정도의 매개변수만을 사용하여 기존 결과와 비슷하거나 더 나은 성능을 얻을 수 있는 것을 확인할 수 있다. 또한, 병목 차원 크기 비교와 중간 층 개수 분석을 통한 최적의 MAdapter 구조를 찾고, 이로써 효율적인 매개변수 미세조정 방법을 제시한다.

주제어: PEFT, Adapter, 언어 모델

1. 서론

사전학습 언어모델(Pre-trained Language Models)은 등장과 동시에 다양한 자연어처리 태스크(Task)에서 우수한 성능을 나타내고 있다[1, 2]. 이러한 사전학습된 언어모델을 특정 태스크에 적용하는 일반적인 방법은 모든 매개변수(Parameter)를 미세조정(Fine-tuning)하는 것이다. 그러나 최근 언어모델이 가지는 매개변수의 수가 기하급수적으로 증가함에 따라, 이러한 방식이 요구하는 컴퓨팅 파워의 증가는 점점 더 부담스러워지고 있는 형국이며, 또한 해당 방식은 각 태스크마다 별도의 모델을 관리해야만 하는 한계가 있다.

이러한 문제점을 극복하기 위해, 효율적인 매개변수 미세조정(Parameter Efficient Fine-Tuning) 방법이 개발되었다. 예를 들어, 작은 모듈을 모델의 레이어 사이에 삽입하는 방법(Adapter[3]), 학습 가능한 토큰을 모델의 입력(Input)이나 숨겨진 레이어(Hidden Layer)에 추가하는 방법(Prefix tuning[4], Prompt tuning[5]), 모델의 편향(Bias)만 학습시키는 방법(Bitfit[6]), 또는 모델 가중치에 학습 가능한 행렬을 연결하여 해당 행렬만을 학습하는 방법(LoRA[7]) 등이 있다.

그 중에서 가장 대표적인 형태인 Adpater는 병목 구조(Bottleneck structure)로 구성되어 있으며 최근 연구[3, 8]에서는 미세조정보다 더 나은 결과를 얻기 위해 병목 차원(r)을 증가시키는 방법이 시도되고 있다. 그러나 이러한 방법은 효율적인 매개변수 학습 방식 본래의 목적에 벗어나는 경우를 초래하고 있다. 목적을 만족시키기 위해 Adapter 모듈에 경량화 방법을 적용한 연구도 있지만, 미세조정 이전에 추가적인 작업을 진행해야 하기 때문에 다소 복잡한 과정을 필요로 한다[9].

이에, 본 논문에서는 병목 차원 크기의 중간 층을 추가해 표현력을 증진하면서도 전체적인 매개변수의 수는 줄일 수 있는 MAdapter를 제시한다. 본 연구가 진행한 실험을 통해 도출한 결론은 다음과 같다. MAdapter를 통해 기존 Adapter보다 학습에 활용하는 매개변수의 개수를 줄이면서도 기존의 성능과 비슷하거나 개선된 부분을 보여준다. 추가로 MAdapter의 병목 차원의 변화 및 추가된 중간 층의 개수에 따른 성능 변화를 살펴봄으로써 최적의 MAdapter 구조를 도출한다.

2. 관련 연구

제시한 방법들은 그림 1 (좌)에서 색이 채워진 레이어만 학습하는 것과 동일하다. Transformers 기반 모델에 추가한 모듈을 포함하여 기존 모델의 LayerNorm과 그림에는 보이지 않지만 최종 분류 레이어를 학습한다.

2.1 Adapter

[3]에서 제안된 방법으로, Transformer의 레이어 사이에 작은 모듈(Adapter)을 추가한다. Adapter 레이어는 세 가지 주요 구성요소로 이루어져 있으며 모델의 차원을 d, 병목 차원을 r, 활성화함수를 f, 입력을 h로 표기한다. (1) Feed-forward

† : 교신저자(Corresponding Author)

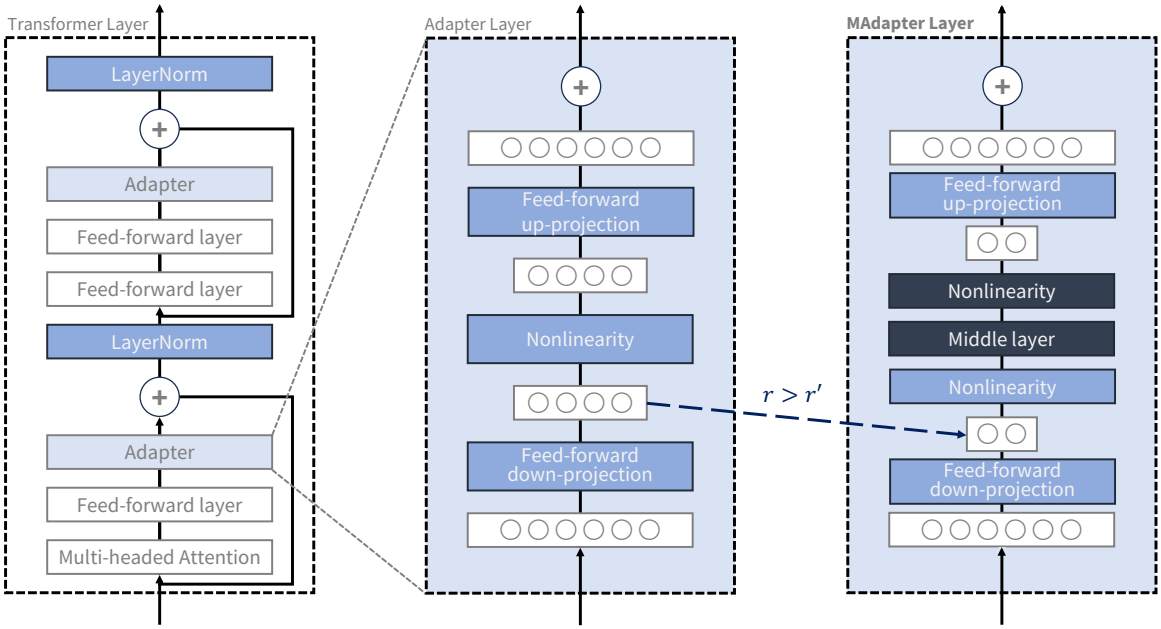

그림 1. (좌) Transfomer Layer (중) Adapter (우) MAdapter (r : 병목 차원, r' : 감소된 병목 차원)

down-projection: 입력을 낮은 차원 공간, 즉 병목 차원으로 투영한다 ($W_{down} \in \mathbb{R}^{d \times r}$). (2) Nonlinearity: 비선형 활성함수이며 본 논문에서는 ReLU를 사용한다($f(\cdot)$). (3) Feed-forward up-projection: 병목 차원에서 다시 입력 크기로 투영하는 레이어이다 ($W_{up} \in \mathbb{R}^{r \times d}$). 잔차 연결(Residual connection)도 포함되어 있는 최종 Adapter의 구성 식은 다음과 같다:

$$h \rightarrow h + f(hW_{down})W_{up}).$$

그림 1의 (좌) 그림과 같이 두개의 Adapter를 Transformer 하나의 레이어 내에 순차적으로 배치하며, 각각 Multi-head Attention, 두 개의 Feed-forward layer 이후에 배치한다. [10]에서는 두 개의 Feed-forward layer 이후에만 배치하는 Adapter를 제시하기도 하였지만, 본 연구에서는 앞서 소개한 [3]의 본래 Adapter 구조를 따른다.

2.2 SparseAdapter

최근 연구[3, 8]에서 병목 차원(r)을 크게 설정한 Adapter를 활용해 미세조정보다 좋은 결과를 제시한다. 하지만 해당 방법은 매개변수의 수가 증가하기 때문에 효율적인 매개변수 학습의 본래 목적에 벗어난다. 본 논문에서는 이 문제에 대한 해결 방법 중 하나인 SparseAdapter[9]를 소개하고 이를 비교 대상으로 삼는다.

SparseAdapter는 Adapter 모듈을 초기화한 후 각 매개변수에 점수를 할당하고, 희소 비율에 맞춰서 가중치에 대한 마스크를 제작한다. 이후 Adapter의 가중치에 마스크를 요소별 곱연산(Element-wise multiplication)을 통해서 가중치를 경량화한다. 학습 과정은 Adapter와 동일하며, 희소 비율에 따라서 매개변수의 수를 조정할 수 있는 장점이 있다. 또한, 병목 차원이 크기가 증가할수록 매개변수 대비 성능이 높게 나타난다. 본 논문에서는 [9]에서 설정한 값과 동일하게 희소 비율 40%로 모듈 경량화 하였으며, 경량화 방법은 5가지 방법(Random, Magnitude, ER, SNIP, GraSP) 중 대부분 태스크에서 좋은 성능을 보여주는 SNIP[11]로 실험을 진행한다.

3. MAdapter

Adapter 구조의 매개변수의 수는 투영 레이어의 크기인 모델의 차원(d) × 병목 차원(r)의 2배 크기($= 2dr$)이다. 본 논문에서 제안하는 MAdapter는 병목 차원을 줄이고 ($r' < r$), 줄인 병목 차원 크기의 중간 층과 활성화 함수 ReLU를 추가한 구조이다. 하나의 MAdapter 모듈에 대한 투영 레이어의 크기는 모델의 차원(d) × 감소한 병목 차원(r')이고, 매개변수는 2개의 투영 레이어 크기에 감소한 병목 차원 크기($r' \times r'$)의 레이어를 더한 값이다($= (2d + r')r'$). 투영 레이어의 크기는 감소했으며($W'_{down} \in \mathbb{R}^{d \times r'}$, $W_{up} \in \mathbb{R}^{r' \times d}$), 추가한 레이어 ($W'_{mid} \in \mathbb{R}^{r' \times r'}$)를 포함한 MAdapter의 구성 식은 다음과 같다:

$$h \rightarrow h + f(f(hW'_{down})W'_{mid})W'_{up}.$$

MAdapter는 이전의 Adapter 대비 매개변수의 수를 줄일 수 있으며, 효율적인 매개변수 미세조정 방법으로 제안할 수 있다. 또한 모델의 깊이를 증가시킴으로써 데이터에 대한 더 나은 적응성을 기대한다. 그림 1의 (우)를 통해서 MAdapter의 구조를 시각적으로 확인할 수 있다.

표 1. GLUE Benchmark에 따른 BERT 검증 데이터 성능

Method	#P	**Model : BERT** / GLUE Benchmark				
		CoLA	MRPC	STS-B	RTE	Avg.
Fine-Tuning	100%	$56.52_{3.0}$	$84.56_{1.5}$	$89.28_{0.3}$	$64.86_{1.1}$	73.81
Adapter	2.0%	$55.03_{1.4}$	$85.7_{0.6}$	$\mathbf{89.88}_{0.4}$	$\mathbf{72.32}_{0.7}$	75.73
SAdapter	1.2%	$\mathbf{57.89}_{1.9}$	$84.39_{1.1}$	$85.77_{0.8}$	$71.96_{1.5}$	75.00
MAdapter	1.0%	$57.84_{1.1}$	$\mathbf{85.94}_{0.6}$	$89.49_{1.2}$	$69.91_{0.7}$	**75.80**

표 2. GLUE Benchmark에 따른 RoBERTa 검증 데이터 성능

Method	#P	**Model : RoBERTa** / GLUE Benchmark				
		CoLA	MRPC	STS-B	RTE	Avg.
Fine-Tuning	100%	$61.35_{2.1}$	$88.73_{1.5}$	$90.65_{0.2}$	$76.30_{2.3}$	79.26
Adapter	2.0%	$\mathbf{62.11}_{0.9}$	$\mathbf{88.56}_{0.6}$	$90.38_{0.1}$	$75.69_{0.5}$	79.19
SAdapter	1.2%	$61.95_{2.7}$	$87.26_{0.7}$	$90.61_{0.3}$	$76.78_{1.1}$	79.15
MAdapter	1.0%	$62.06_{0.8}$	$87.42_{0.8}$	$\mathbf{90.93}_{0.2}$	$\mathbf{79.78}_{0.7}$	**80.05**

4. 실험 환경

4.1 수행 태스크 및 데이터

본 연구는 GLUE Benchmark[12] 데이터셋을 활용한 실험을 진행한다. GLUE Benchmark에서 사용한 태스크는 자연어 추론(Natural language inference), 감정 분석(Sentiment analysis), 문장 유사 평가(Sentence similarity evaluation)이다. 데이터셋은 CoLA, MRPC, STS-B, RTE를 사용한다. 검증(Validation)은 훈련 데이터(Training set)의 10%를 따로 떼어 별도의 데이터셋을 만들어 진행하고, 최종 평가(Evaluation)는 이전 연구[9]와 같이 기존 검증 데이터(Validation set)로 수행한다.

4.2 훈련 세부사항

사전 훈련 언어 모델로 $\text{BERT}_{\text{Base}}$[1]와 $\text{RoBERTa}_{\text{Base}}$[2]를 사용한다. 학습률(Learning rate)은 {1e-5, 5e-5, 1e-4, 5e-4}를 사용하고, 배치(Batch)의 크기는 {8, 16}, 에폭(Epoch)은 20으로 설정하며 학습 조기 종료(Early stopping)는 5로 설정값을 지정한다. Adapter의 병목 차원(r)은 64로 조정하고, MAdapter 감소한 병목 차원(r')은 비교군인 SparseAdapter와 MAdapter 매개변수의 수를 유사한 수준으로 설정하기 위해 기존 병목 차원의 절반($r' = r/2$)인 32로 설정한다. 나머지 하이퍼파라미터에 대해서는 이전 연구[9]와 동일하게 진행하며, 검증을 통해 최적의 하이퍼파라미터를 찾고 해당 하이퍼파라미터로 학습한 모델을 통해 평가를 진행한다.

5. 실험 결과

5.1 방법론 별 성능 결과

표 1, 표 2는 각각 BERT, RoBERTa 모델의 미세조정, Adapter, SparseAdapter, MAdapter 학습의 성능을 나타낸 표이다.[1] 기존 사전학습에 활용된 매개변수의 수 대비 모듈의 매개변수의 수 비율을 #P로 표시하고, 모듈별 점수의 평균은 밑줄로 태스크 기준 가장 좋은 점수는 **굵은 글꼴**로 표시한다.

각 태스크별 평가 지표는 MRPC와 RTE는 정확도(Accuracy)를 사용했고, CoLA는 매튜 상관 계수(Matthew's Correlation Coefficient)[2], STS-B의 경우는 피어슨 상관 계수(Pearson Correlation Coefficient)를 사용한다. 각 모델의 성능은 서로 다른 세 가지 임의 시드(Random seed) 수행한 실험 결과의 평균으로, 각 점수의 아래 첨자는 해당 실험 결과의 표준편차를 나타낸다. 모듈의 매개변수의 수(#P)를 비교했을 때, 미세조정의 전체 매개변수의 수를 100%라고 한다면 Adapter가 약 2%, SparseAdapter와 MAdapter가 약 1% 가량의 수인 것을 확인할 수 있다.

표 1, 표 2의 결과, 평가된 태스크에 대한 평균 수행 능력은 MAdapter가 가장 높은 것을 확인할 수 있다. BERT에서의 평균 점수는 미세조정과 1.99점, Adapter와 0.07점, SparseAdapter와는 0.8점의 차이를 보이며, MRPC에서는 다른 모듈 대비 가장 좋은 성능을 보여준다. RoBERTa에서의 평균 점수는 미세조정과 0.79점, Adapter와 0.86점, SparseAdapter와는 0.9점 차이를 보이며, STS-B, RTE에서 다른 모듈 대비 가장 좋은 성능을 보인다. 특히 미세조정 결과와 비교했을 때, 학습 가능한 매개변수의 수 차이가 상당히 크더라도 모든 태스크의 결과가 유사하거나 더 좋은 것을 표를 통해 확인할 수 있다. 이러한 결과는 모델의 학습 가능한 매개변수의 수를 줄여 계산량을 감소시키면서도 매개변수 전체를 학습하는 것과 성능적으로 큰 차이가 없음을 제시하며, 언어 모델의 매개변수의 수가 증가할수록 Adapter의 매개변수의 수도 증가하는 것을 고려했을 때 더 한정된 매개변수를 사용하면서 효과적인 성능 향상을 이룰 수 있음을 시사한다.

5.2 MAdapter의 구조 변화에 따른 성능 결과

그림 2는 병목 차원에 따른 Adapter와 MAdapter의 MRPC Task 결과 그래프이다. 그래프의 가로 축은 병목 차원(r) 기준이며, 병목 차원은 {32, 64, 128, 256}에서 실험했으며, MAdapter의 그래프의 가로 축은 Adapter의 기준으로 설정한다. 예를 들어 가로 축이 64일 때 MAdapter는 $r' = 32$에 대한

[1] SAdapter는 SparseAdapter의 줄임말을 의미한다.

[2] $MCC = (TP/N - S \times P)/\sqrt{SP(1-S)(1-P)}$
TP : Positive True, N : 전체 관측치의 개수
S : 전체 관측치 중 참인 관측치의 개수의 비율
P : 전체 관측치 중 참이라 예측한 관측치의 개수의 비율

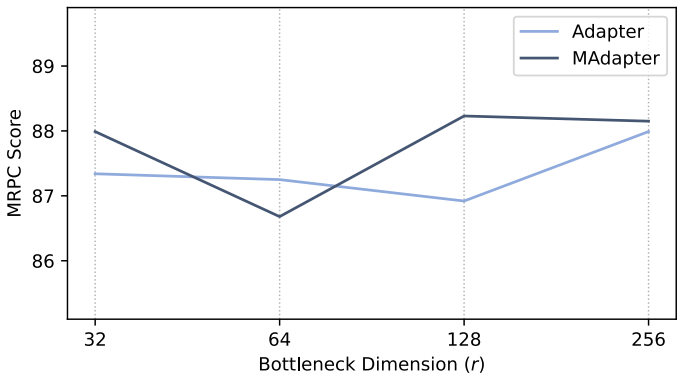

그림 2. 병목 차원에 따른 Adapter와 MAdapter의 성능 그래프

결과이다. 모델은 $\text{RoBERTa}_{\text{Base}}$이며 배치와 학습률를 비롯한 모든 하이퍼파라미터를 하나의 조건으로 실험하였다. 병목 차원이 64일 때는 앞의 실험과 동일하게 Adapter가 MAdpater 보다 좋은 성능을 보인다. 하지만 나머지 차원에서는 Adapter 보다 MAdapter가 성능이 더 좋은 것을 확인할 수 있다. 이를 통해, 최근 SparseAdapter의 경우와 같이 Adapter의 병목 차원을 증가시킴으로써 효율성을 희생하되 성능을 높이는 방식의 비합리성을 지적하고, 매개변수 효율성과 태스크 성능 모두 MAdapter와 같은 대안으로 충족이 가능함을 주장한다. 또한 다양한 병목 차원 및 실험 조건에 대하여 제안된 방식을 더 심층적으로 실험해 볼 필요가 있음을 보인다.

표 3은 MAdapter의 중간 층(Middle layer)의 수가 성능에 미치는 영향을 분석한다. 중간 층을 추가하는 방식은 기존과 같이 감소한 병목 차원(r')과 동일한 크기의 중간 층($r' \times r'$)과 활성화 함수 ReLU를 1개만 고려하는 것이 아닌, 여러 개를 추가하여 구조를 보다 깊게 제작하는 형태로 진행한다. 실험에 사용한 모델은 $\text{RoBERTa}_{\text{Base}}$이고 MAdapter에 최대 3개의 중간 층을 삽입했으며, 층의 개수에 따른 4개의 태스크 결과를 보여준다. 그 결과, 중간 층 수를 증가시킴에 따라 일부 태스크에서는 성능 향상이 나타났지만, 대부분의 태스크에서는 매개변수의 수만 증가하고 성능의 증가는 크게 보이지 않았다. 이로써 MAdapter의 구조에서의 중간 층의 증가에 따른 이점은 거의 없으며 추가되는 매개변수의 수를 고려하였을 때, 최적의 중간 층 수는 1개로 추정된다.

6. 결론 및 향후 연구 방향

본 연구에서는 기존의 Adapter 모듈에 중간 층을 넣어 변형한 MAdapter를 소개한다. GLUE Benchmark[12]의 네 가지 태스크에서 실험을 수행하였으며, 모델 전체를 미세조정하는 대신 MAdapter만을 학습함으로써 매개변수의 수를 기존 Adapter의 절반 가까이 감소시켰음에도 불구하고, 결과는 동등하거나 더 나은 성능을 나타냈다. 또한 다양한 구조 변화를 통해 최적의 MAdapter를 제시한다.

표 3. MAdapter의 중간 층(Middle layer)의 수에 따른 성능

Model	#Layer	CoLA	MRPC	STS-B	RTE
RoBERTa	×1	62.06	87.42	**90.93**	**79.78**
	×2	**62.24**	87.82	90.53	76.06
	×3	59.09	**87.99**	90.61	79.77

향후 연구 방향으로는 대규모 언어 모델(Large Language Model)에 적용 및 다른 유사 방법과의 성능 비교를 통해 MAdapter의 우위성을 확인하고, MAdapter에서 새롭게 추가된 중간 층을 사전 지식을 포함하고 있는 다양한 태스크 및 언어 정보의 가중치로 대체하여 활용 가능성을 파악할 계획이다.

감사의 글

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2022R1F1A1074674).

참고문헌

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Jun. 2019. [Online]. Available: https://aclanthology.org/N19-1423

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[3] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," *International Conference on Machine Learning*, pp. 2790–2799, 2019.

[4] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Aug. 2021. [Online]. Available: https://aclanthology.org/2021.acl-long.353

[5] Y. Gu, X. Han, Z. Liu, and M. Huang, "PPT: Pre-trained prompt tuning for few-shot learning," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8410–8423, May 2022. [Online]. Available: https://aclanthology.org/2022.acl-long.576

[6] E. Ben Zaken, Y. Goldberg, and S. Ravfogel, "BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, May 2022. [Online]. Available: https://aclanthology.org/2022.acl-short.1

[7] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[8] Y. Wang, S. Agarwal, S. Mukherjee, X. Liu, J. Gao, A. H. Awadallah, and J. Gao, "AdaMix: Mixture-of-adaptations for parameter-efficient model tuning," *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5744–5760, Dec. 2022. [Online]. Available: https://aclanthology.org/2022.emnlp-main.388

[9] S. He, L. Ding, D. Dong, J. Zhang, and D. Tao, "SparseAdapter: An easy approach for improving the parameter-efficiency of adapters," *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2184–2190, Dec. 2022. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.160

[10] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, "AdapterFusion: Non-destructive task composition for transfer learning," *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503, Apr. 2021. [Online]. Available: https://aclanthology.org/2021.eacl-main.39

[11] N. Lee, T. Ajanthan, and P. Torr, "SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY," *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=B1VZqjAcYX

[12] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=rJ4km2R5t7