ELECTRA

Efficiently **L**earning an **E**ncoder that **C**lassifies **T**oken **R**eplacements **A**ccurately

Index

Introduction

논문 기준, 학습에 필요한 가격

DEVIEW 2019

* TPU는 1 device -> 4 chips -> 8 cores 입니다.

Model	Size	TPU ($ per hour)	TPU Count (device)	Training Time	Cost (USD)	CO2 emissions (lbs)
BERT	24 Layers (340M)	v2 ($4.5)	16	4 days	$6,912 (약 850만원)	1428
GPT-2	48 Layers (1542M)	v3 ($8)	32	7 days	$43,008 (약 5,100만원)	2516
XLNet	24 Layers (365M)	v3 ($8)	128	2.5 days	$61,440 (약 7,300만원)	-

CO_2 * NY ↔ SF Air Travel: 1924 (lbs)

참고: https://syncedreview.com/2019/06/27/the-staggering-cost-of-training-sota-ai-models/,
Energy and Policy Considerations for Deep Learning in NLP, 2019 E Strubell

출처: Clova AI Deview 2019

Introduction

MLM(Masked Language Modeling)

입력 시퀀스의 토큰 중 15%를 마스킹하여 이를 복원하는 테스크로, 자동 회귀 언어 모델링 학습 (autoregressive language modeling)과 달리 양방향 정보를 고려하여 전체적인 문맥을 이해하는 방향으로 학습한다.

전체 토큰 중 15%에 대해서만 loss가 발생한다.

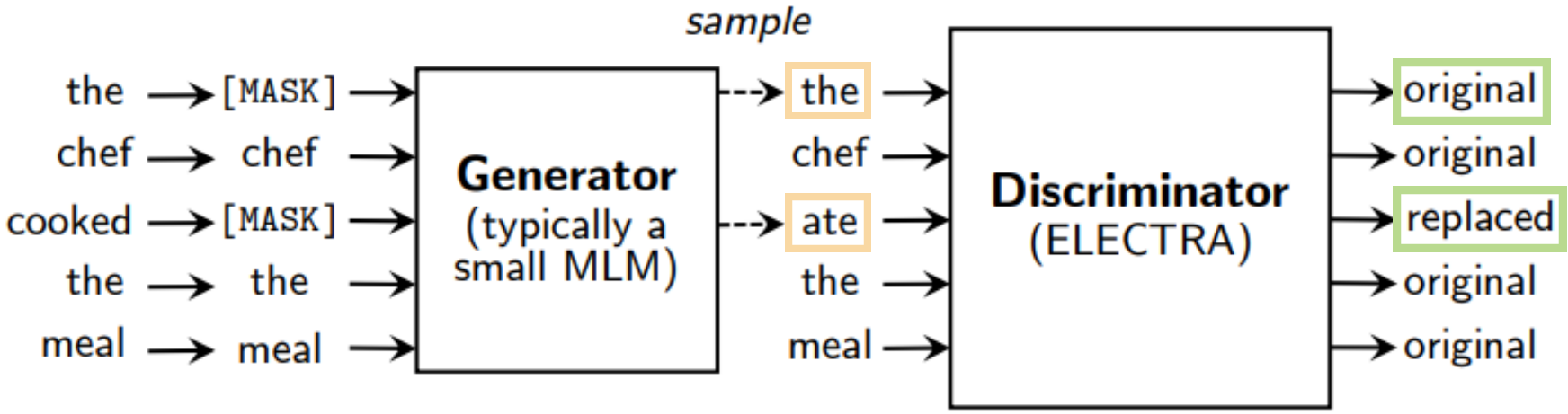
- (!) 데이터가 많이 필요하다
- (!!) 학습하는데 비용이 많이 발생한다

Fine-tuning task에서는 [MASK] 토큰이 사용되지 않는다.

- (!) 사전 학습과 파인 튜닝 사이 토큰에 대한 불일치가 생길 수 있다.
- (!!) 성능 손실이 발생할 수 있다

ELECTRA

Replaced Token Detection



[MASK] Recognition

Original vs Replaced

$$\min_{\theta_G,\theta_D} \sum_{\boldsymbol{x}\in\mathcal{X}} \mathcal{L}_{\text{MLM}}(\boldsymbol{x},\theta_G) + \lambda\mathcal{L}_{\text{Disc}}(\boldsymbol{x},\theta_D)$$

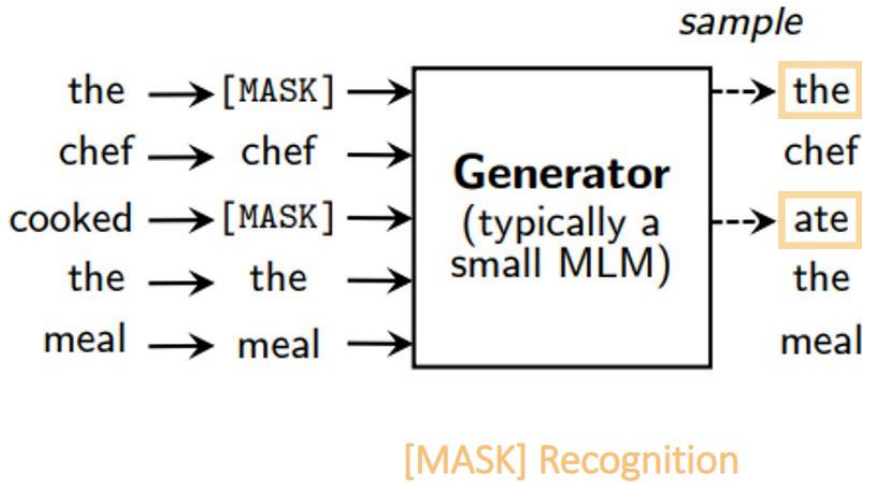
ELECTRA

Generator

[MASK]가 포함된 문장을 받아 [MASK]에 해당하는 단어를 생성 => **MLM**

Softmax layer를 사용해서 masked된 sequence token의 분포에 대한 MLE 계산 후 loss function으로 사용

$$\mathcal{L}_{\mathrm{MLM}}(\boldsymbol{x}, \theta_G) = \mathbb{E}\left(\sum_{i \in \boldsymbol{m}} -\log p_G(x_i | \boldsymbol{x}^{\mathrm{masked}})\right)$$



[MASK] Recognition

ELECTRA

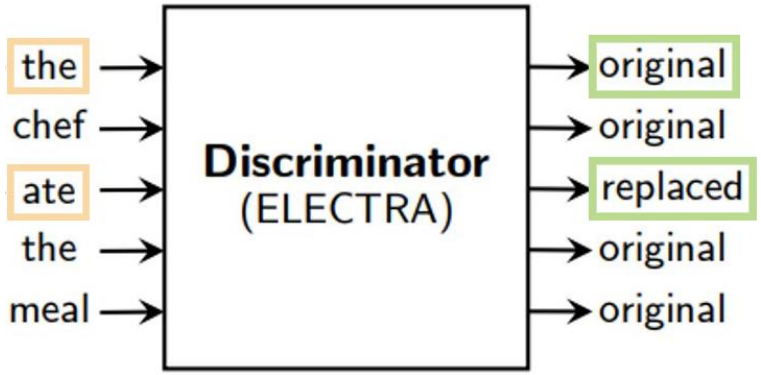
Discriminator

Generator가 만든 sequence token에 대해 Original or Replaced binary classification

Loss Function은 cross-entropy + sigmoid 사용

$$\mathcal{L}_{\text{Disc}}(\boldsymbol{x}, \theta_D) = \mathbb{E}\left(\sum_{t=1}^{n} -\mathbb{1}(x_t^{\text{corrupt}} = x_t)\log D(\boldsymbol{x}^{\text{corrupt}}, t) - \mathbb{1}(x_t^{\text{corrupt}} \neq x_t)\log(1 - D(\boldsymbol{x}^{\text{corrupt}}, t))\right)$$

판별기의 loss function -> loss가 모든 입력 토큰에 대해 계산될 수 있음



Original vs Replaced

ELECTRA

Objective Function

Generator와 Discriminator의 loss function을 lambda로 가중합한 loss function을 최소화 한다.

$$\min_{\theta_G,\theta_D} \sum_{\boldsymbol{x}\in\mathcal{X}} \mathcal{L}_{\text{MLM}}(\boldsymbol{x},\theta_G) + \lambda\mathcal{L}_{\text{Disc}}(\boldsymbol{x},\theta_D)$$

- MLM Loss가 Disc Loss보다 크기 때문에 lambda를 50 정도로 설정 (scale 맞추기)

- Disc Loss는 generator로 back-prob 되지는 않는다. (Generator의 sampling 과정에서 argmax를 사용하는데 loss가 전파될 수 없음)

Experiments

Weight Sharing

가중치를 서로 공유하면 얻는 이점은??

Global minimum 수렴 속도 up + Parameter 수를 줄이는 것으로 normalized!

"BUT" generator, discriminator의 크기를 맞춰 주어야지 모든 weight을 공유 가능... + memory 적으로 부담됨...

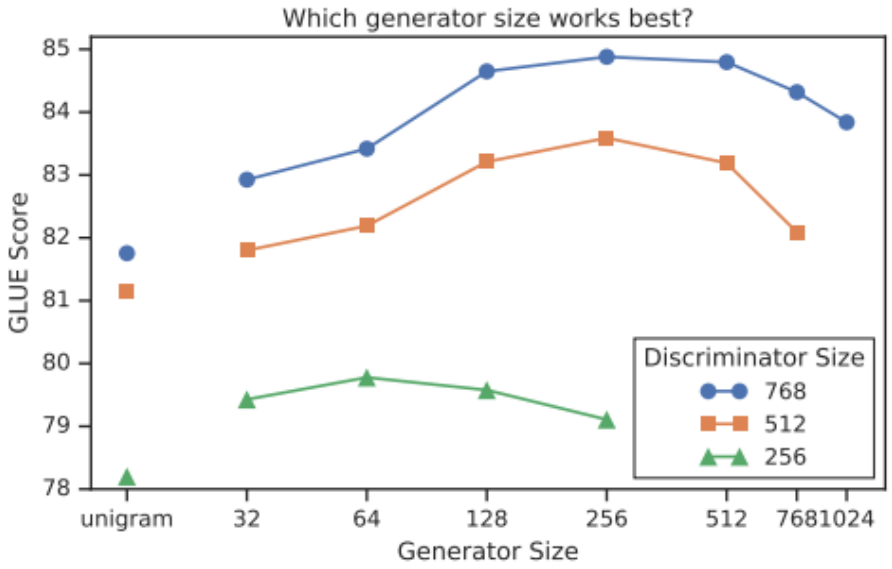
Embedding만 sharing

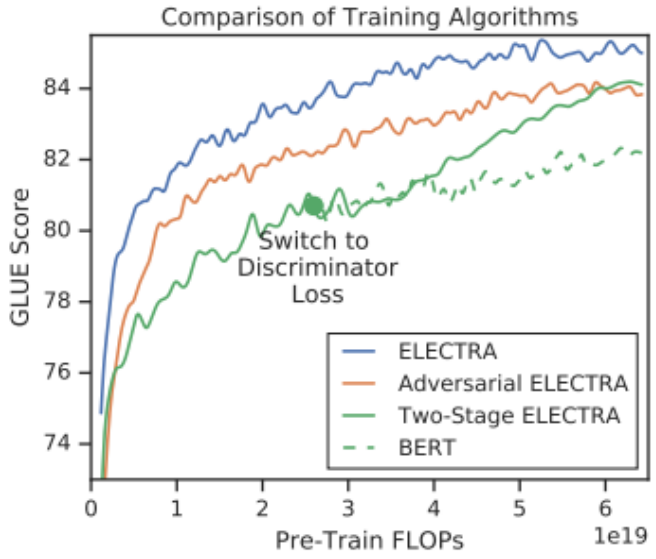
Input tokens의 모든 후보(corpus)에 대해 튜닝할 수 있는 것은 MLM밖에 없다. Embedding layer를 sharing함으로써 discriminator도 튜닝된 값 사용 가능!

Experiments

Smaller Generator

Generator의 크기를 Discriminator보다 작게 하는게 성능 향상에 도움된다.





GLUE scores

Experiments

Performance

Model	Train / Infer FLOPs	Speedup	Params	Train Time + Hardware	GLUE
ELMo	3.3e18 / 2.6e10	19x / 1.2x	96M	14d on 3 GTX 1080 GPUs	71.2
GPT	4.0e19 / 3.0e10	1.6x / 0.97x	117M	25d on 8 P6000 GPUs	78.8
BERT-Small	1.4e18 / 3.7e9	45x / 8x	14M	4d on 1 V100 GPU	75.1
BERT-Base	6.4e19 / 2.9e10	1x / 1x	110M	4d on 16 TPUv3s	82.2
ELECTRA-Small	1.4e18 / 3.7e9	45x / 8x	14M	4d on 1 V100 GPU	79.9
50% trained	7.1e17 / 3.7e9	90x / 8x	14M	2d on 1 V100 GPU	79.0
25% trained	3.6e17 / 3.7e9	181x / 8x	14M	1d on 1 V100 GPU	77.7
12.5% trained	1.8e17 / 3.7e9	361x / 8x	14M	12h on 1 V100 GPU	76.0
6.25% trained	8.9e16 / 3.7e9	722x / 8x	14M	6h on 1 V100 GPU	74.1
ELECTRA-Base	6.4e19 / 2.9e10	1x / 1x	110M	4d on 16 TPUv3s	85.1

Model	Train FLOPs	Params	SQuAD 1.1 dev		SQuAD 2.0 dev		SQuAD 2.0 test	
			EM	F1	EM	F1	EM	F1
BERT-Base	6.4e19 (0.09x)	110M	80.8	88.5	–	–	–	–
BERT	1.9e20 (0.27x)	335M	84.1	90.9	79.0	81.8	80.0	83.0
SpanBERT	7.1e20 (1x)	335M	88.8	94.6	85.7	88.7	85.7	88.7
XLNet-Base	6.6e19 (0.09x)	117M	81.3	–	78.5	–	–	–
XLNet	3.9e21 (5.4x)	360M	**89.7**	**95.1**	87.9	**90.6**	87.9	90.7
RoBERTa-100K	6.4e20 (0.90x)	356M	–	94.0	–	87.7	–	–
RoBERTa-500K	3.2e21 (4.5x)	356M	88.9	94.6	86.5	89.4	86.8	89.8
ALBERT	3.1e22 (44x)	235M	89.3	94.8	87.4	90.2	88.1	90.9
BERT (ours)	7.1e20 (1x)	335M	88.0	93.7	84.7	87.5	–	–
ELECTRA-Base	6.4e19 (0.09x)	110M	84.5	90.8	80.5	83.3	–	–
ELECTRA-400K	7.1e20 (1x)	335M	88.7	94.2	86.9	89.6	–	–
ELECTRA-1.75M	3.1e21 (4.4x)	335M	**89.7**	94.9	**88.0**	**90.6**	**88.7**	**91.4**

Table 4: Results on the SQuAD for non-ensemble models.

Conclusion

ELECTRA는 작은 Generator가 만든 negative token과 입력 token을 구별하는 테스크인 Replaced Token Detection을 통해 다른 모델에 비해서 상대적으로 적은 계산량을 사용하여 훨씬 효율적이고 성능이 높은 모델을 만들 수 있었다.