ALBERT : A Lite BERT for self-supervised Learning of language representations

Zhenzhong Lan, Mingda Chen, Sebastian Goodman,
Kevin Gimpel, Piyush Sharma, Radu Soricut

정영석

Contents

Introduction

▪ 연구 배경

- **대다수의 NLP task는 Training dataset이 부족하여 Pre-trained model을 활용함**

  ✓ 따라서 성능이 좋은 Pre-trained model을 만들기 위해 모델의 size가 점점 커지게 됨

  → 과연 모델의 크기(size)를 무한정으로 늘리는 것이 성능에 좋을까?

- **실제 환경에서 모델의 크기는 하드웨어 자원에 따라 한정적일 수 밖에 없음**

  ✓ 실제로 많은 word representation model 의 크기 ↑ : 학습 시간 ↑,
  word representation 모델의 overhead ↑

  ✓ 기존의 해결 방법 : parallelization (Shazzer et al.; Shoeybi et al.)
  clever memory management (Chen et al.; Gomez et al)

Introduction

▪ 연구 목적

- BERT 모델의 구조적 변화를 통해 parameter 수를 획기적으로 줄임

  ✓ Factorized embedding parameterization (embedding vector 인수분해)

  ✓ cross-layer parameter sharing (layer간 parameter 공유)

  → **파라미터 사용의 효율성** ↑

- 문장 사이의 일관성 정보를 학습하기 위한 Loss 함수 설계

  ✓ Sentence-order-prediction (SOP)

  → **Next Sentence Prediction (NSP)의 단점을 보완함**

▪ Related Work

- Scaling up representation learning for natural language

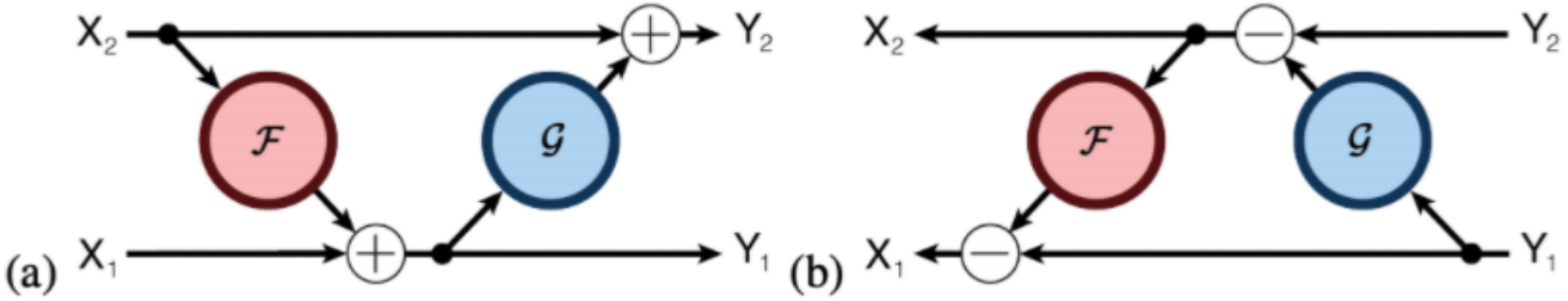
  ✓ 자연어 처리의 많은 Task에서 좋은 성능을 달성한 모델들이 fine-tuning 기법을 사용함 (패러다임 변화)

  ✓ 많은 task에서 모델의 크기가 커질수록 좋은 성능을 달성

  → **단순히 모델의 크기만 키우는 것이 모델의 성능 향상에 도움이 되는 것은 아님 (Devlin et al.)**

  ✓ 모델의 크기에 따른 메모리 사용량을 조절하기 위한 알고리즘들이 탄생함

  - Gradient checkpointing (역전파 전달에 불필요한 노드의 값을 메모리에 저장하지 않고 제거함)
  - RevNet (입력된 값을 메모리에 저장하지 않고 계산을 통해 추론함)



Related Work

▪ Related Work

- Cross-layer parameter sharing
  - ✓ Transformer에서 파라미터 공유 층이 주어-술어의 관계를 표현하는데 적합한 것이 실험을 통해 검증 됨 (Universal Transformer)
  - ✓ Transformer model 에서 Cross-layer parameter 층을 사용하면 특정 층 이후에는 입력된 벡터와 비슷한 값으로 수렴하는 것을 보임 (Deep Equilibrium Model)

- Sentence Ordering Objectives
  - ✓ 많은 이전 연구에서 담화의 일관성에 대한 성능을 평가하기 위한 목적함수를 만들기 위한 연구를 진행함
  - ✓ 대부분의 연구에서 꽤 단순한 방법을 통해 목적함수를 설계함
    - Skip-thought, FastSent
    - Discourse markers prediction (예고, 강조, 요약 등)

Methodology

▪ Model Architecture Choice

- Factorized embedding parameterization

  ✓ 모델링의 관점 : WordPiece Embedding 방법은 context-independent representation을 의미
  Hidden-layer embedding 방법은 context-dependent representation을 의미

  ✓ BERT-Like 모델의 경우 context-dependent representation 이 성능향상에 주요한 역할을 함
  → WordPiece Embedding과정에서 Parameter를 감소해 학습 parameters 수를 줄일 수 있음

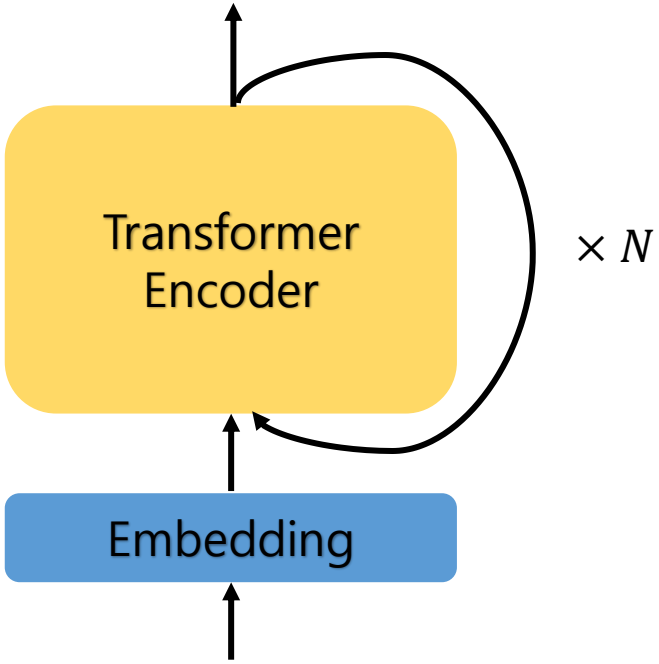
  ✓ $O(V \times H)$ 보다 $O(V \times E + E \times H)$의 과정을 통해 학습 파라미터를 효과적으로 줄일 수 있음
  $Ex)$ V : 30,000, H : 768 , E : 128

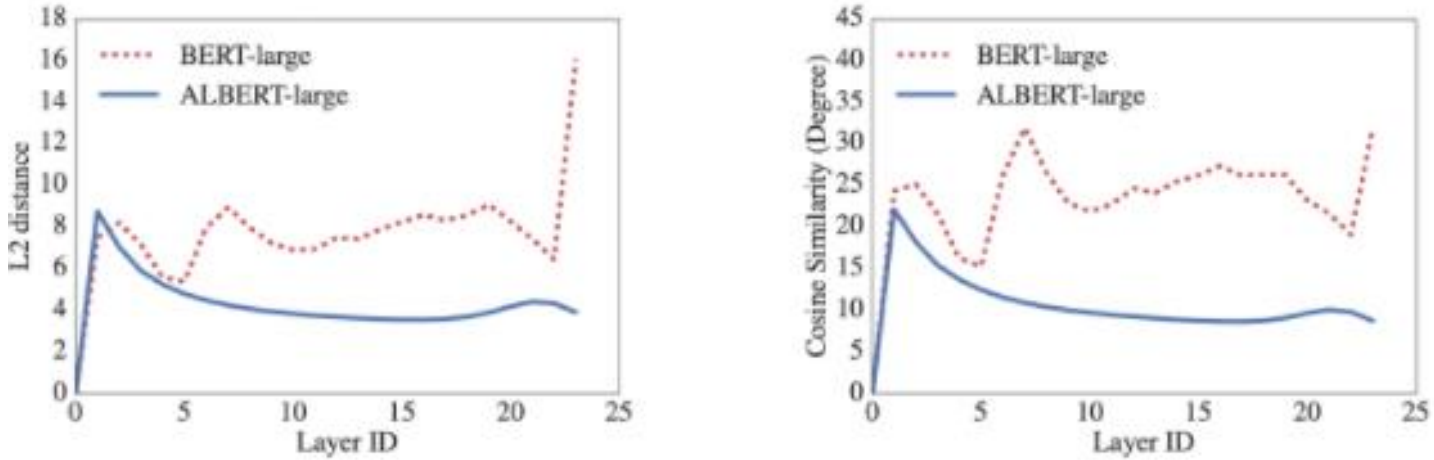
  $$V \times H = 23{,}040{,}000 \gg V \times E + E \times H = 3{,}938{,}304$$

Methodology

▪ Model Architecture Choice

- Cross-layer parameter sharing





[그림] 각 Layer 의 결과와 입력 임베딩 값의 유사도 비교

Methodology

Model Architecture Choice

- Configurations of the BERT and ALBERT

Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
	xlarge	1270M	24	2048	2048	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

Table 2: The configurations of the main BERT and ALBERT models analyzed in this paper.

Methodology

▪ Model Architecture Choice

- Inter-sentence coherence loss

  ✓ 몇몇 연구에서 BERT 에서 사용했던 Next Sentence Prediction (NSP) Loss는 신뢰성이 떨어지는 것으로 나타남

  → NLP task에서 NSP는 Masked Language Model (MLM) 에 비해 너무 간단한 task로 추측됨

  ✓ $ALBERT$ 에서는 NSP의 단점을 보완하고 문장간 일관성 정보를 표현하기 위해 새로운 loss 설계

  → Sentence-Order-Prediction (SOP) : 문장의 순서를 뒤바꾸면서 문맥을 이해하도록 설계

Methodology

▪ Model Architecture Choice

- Factorized embedding parameterization

  ✓ Transformer에서 파라미터 공유 층이 주어-술어의 관계를 표현하는데 적합한 것이 실험을 통해 검증 됨 (Universal Transformer)

  ✓ Transformer model 에서 Cross-layer parameter 층을 사용하면 특정 층 이후에는 입력된 벡터와 비슷한 값으로 수렴하는 것을 보임 (Deep Equilibrium Model)

- Sentence Ordering Objectives

  ✓ 많은 이전 연구에서 담화의 일관성에 대한 성능을 평가하기 위한 목적함수를 만들기 위한 연구를 진행함

  ✓ 대부분의 연구에서 꽤 단순한 방법을 통해 목적함수를 설계함

    - Skip-thought, FastSent
    - Discourse markers prediction (예고, 강조, 요약 등)

Experiments

Experimental Setup

- Dataset
  - ✓ Train Set : BOOK Corpus, English Wikipedia
  - ✓ Dev Set : SQuAD, RACE dataset
- Pre-processing
  - ✓ Input 문장 변환 : $[CLS]\ x_1\ [SEP]\ x_2\ [SEP]$
  - ✓ MLM Loss를 위한 n-gram Masking

$$p(n) = \frac{1/n}{\sum_k^N 1/k}, (\max(n) = 3)$$

Experiments

Downstream Evaluation

- Comparison BERT with ALBERT

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	4.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	5.6x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	1.7x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	0.6x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	0.3x

- Factorized Embedding Parameterization

Model	E	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base not-shared	64	87M	89.9/82.9	80.1/77.8	82.9	91.5	66.7	81.3
	128	89M	89.9/82.8	80.3/77.3	83.7	91.5	67.9	81.7
	256	93M	90.2/83.2	80.3/77.4	84.1	91.9	67.3	81.8
	768	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base all-shared	64	10M	88.7/81.4	77.5/74.8	80.8	89.4	63.5	79.0
	128	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1
	256	16M	88.8/81.5	79.1/76.3	81.5	90.3	63.4	79.6
	768	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8

Experiments

Downstream Evaluation

- Cross-Layer Parameter Sharing

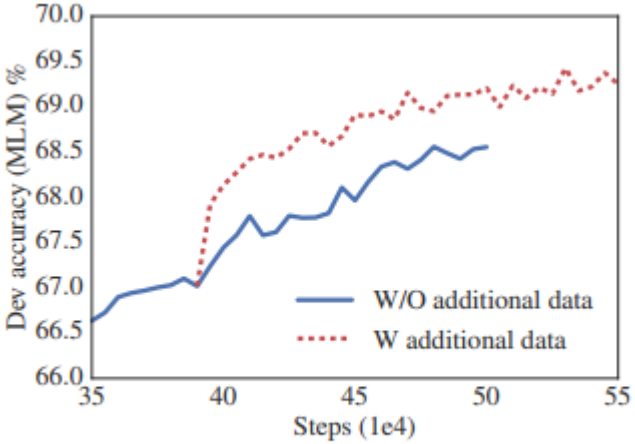
Model	E	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base not-shared	64	87M	89.9/82.9	80.1/77.8	82.9	91.5	66.7	81.3
	128	89M	89.9/82.8	80.3/77.3	83.7	91.5	67.9	81.7
	256	93M	90.2/83.2	80.3/77.4	84.1	91.9	67.3	81.8
	768	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base all-shared	64	10M	88.7/81.4	77.5/74.8	80.8	89.4	63.5	79.0
	128	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1
	256	16M	88.8/81.5	79.1/76.3	81.5	90.3	63.4	79.6
	768	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8

- Sentence Order Prediction (SOP)

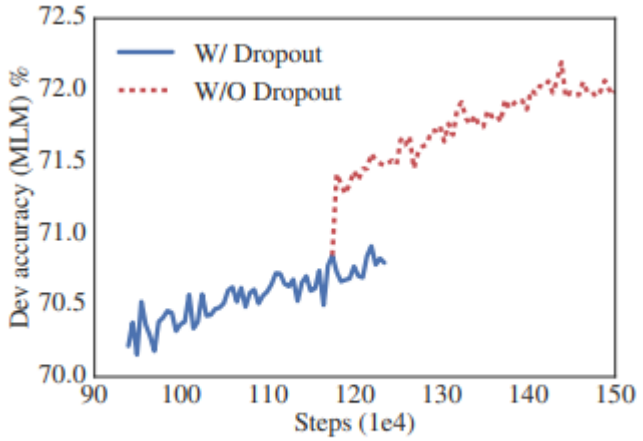
	Intrinsic Tasks			Downstream Tasks					
SP tasks	MLM	NSP	SOP	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
None	54.9	52.4	53.3	88.6/81.5	78.1/75.3	81.5	89.9	61.7	79.0
NSP	54.5	90.5	52.0	88.4/81.5	77.2/74.6	81.6	**91.1**	62.3	79.2
SOP	54.0	78.9	86.5	**89.3/82.3**	**80.0/77.1**	**82.0**	90.3	**64.0**	**80.1**

Experiments

- Additional Training Data and Dropout Effects



(a) Adding data



(b) Removing dropout

	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
No additional data	**89.3/82.3**	**80.0/77.1**	81.6	90.3	64.0	80.1
With additional data	88.8/81.7	79.1/76.3	**82.4**	**92.8**	**66.0**	**80.8**

	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
With dropout	94.7/89.2	89.6/86.9	90.0	96.3	85.7	90.4
Without dropout	**94.8/89.5**	**89.9/87.2**	**90.4**	**96.5**	**86.1**	**90.7**

THANK YOU