Improving Language Understanding by Generative Pre-Training

Abstract & Introduction

- 대규모의 라벨이 없는 text corpus가 풍부하지만, 작업을 학습하는데 필요한 label이 있는 데이터는 부족한 문제 (데이터 주석 작업의 시간과 비용)
- 본 논문에서는 label이 없는 text data에 대해 generative pre-training한 후, 각 특정 task에 대해 discriminative fine-tuning을 통해 성능 향상을 달성할 수 있음을 보여줌 + 기존에도, supervised learning data가 있음에도, unsupervised learning이 좋은 표현으로 잘 학습하면 성능 향상을 가져올 수 있었음 ex) pre-**trained word embeddings**의 활용
    - 기존의 label이 없는 text로부터 정보를 활용하는 것이 어려운 이유 1) optimization objectives 가 transfer에 유용한 text representation을 학습하는데 효과적인지 명확하지 않다는 문제, 2) 학습된 표현을 target task로 어떻게 효과적으로 전이할 것인지에 대한 문제 (semisupervised learning 접근법을 개발하는 것이 어려운 상황): auxiliary objective (보조적인 학습 목표)를 추가하거나, 아키텍처를 task별로 수정하거나, 복잡한 학습 기법을 사용했었음
- 본 논문에서는 **unsupervised pre-training** (label이 없는 데이터에서 language modeling objective를 사용해서 초기 파라미터를 학습) + **supervised fine-tuning** (supervised objective를 사용하여 목표 task에 적응 위한)을 결합한 semisupervised 접근법을 제한함 → 최소한의 수정으로 다양한 task에 전이 가능한 universal representation을 학습하는 것이 목표임
- 긴 텍스트간 의존성을 효과적으로 처리할 수 있는 Transformer model을 사용
- 이전과 달리, task-aware input transformations을 fine-tuning에 활용하여 효과적인 전이 + 모델 구조 변화 최소화가 가능함
- 전이 과정에서 traversal-style 접근법으로 task-specific input 변환을 적용 : 구조화된 text input을 단일 연속적인 시퀀스로 처리하는 방식을 의미함 → pre-trained된 모델의 아키텍처를 변경하지 않고도 fine-tuning을 효과적으로 수행할 수 있음

Related Work

Semi-supervised Learning for NLP

- 초기에는 unlabeled data를 이용하여 word-level/phrase-level의 통계 정보를 계산하고 이를 supervised model의 feature로 활용하는 방식
- 최근, label이 없는 corpus에서 훈련된 word embedding으로 성능 향상. 하지만 이런 방법들은 주로 word-level information을 전이하는 방식임 → 본 논문에서는 **semantic representation을 학습하는데 초점 → Word2Vec이나 GloVe는 단어 자체의 의미를 벡터로 변환하는 것이 목표였기에 word-level에서 의미를 학습. 하지만 이런 기법은 문장 전체의 의미나 맥락을 충분히 반영하지 않음. language modeling으로 (Transfer model은 다음 단어 예측 또는 마스킹 된 단어 복원 등의 language modeling 목표를 통해 pre-training을 함 → 전체 문장에서 단어가 어떻게 등장하는지 학습할 수 있음 semantic representation을 학습 가능 + 문장 간 관계를 학습 (두 문장이 의미적으로 유사한지, 연결되는지, 모순되는지 등 학습하면))**

Unsupervised pre-training

- supervised learnning의 objective를 변경하는 것이 아닌, 좋은 initialization point를 찾는 것이 목표임
- 기존의 연구에서, pre-training이 regularization 역할을 하여 신경망의 일반화 성능을 향상시킨다는 것을 입증함
- language modeling 목표를 사용하여 신경망을 pre-training 한 후, 지도학습을 통해 목표 task에 fine-tuning 하는 접근법을 따르는 연구들이 우리 연구와 유사한 연구 + 기존의 연구는 LSTM 모델이 짧은 범위에 한정되어 있지만, 우리는 Transformer 네트워크로 더 긴 범위의 언어적 구조를 효과적으로 실험할 수 있음 **: 기존의 연구들은 RNN이나 LSTM을 pre-training 한 후 fine-tuning 하는 방식**
- 또한 기존의 연구들은, pre-training 된 언어 모델에서 얻은 hidden representation을 auxiliary feature (보조 특징)으로 활용하여 지도 학습을 수행하지만, 이 방식은 각각의 task 마다 새로운 파라미터를 많이 추가해야 한다는 단점 **: hidden representation이란 모델이 입력을 처리하면서 각 layer에서 학습한 내부적인 특징을 의미 → 기존 연구에선, pre-training된 모델의 특정 layer에서 나온 hidden state를 추출하여 새로운 supervised model의 입력으로 추가하는 방식 . 이러면 각각의 task 마다 새로운 auxiliary feature를 추가적으로 학습해야 하는 문제가 있음**

→ 우리의 접근법은 transfer 과정에서 모델의 아키텍처 변경을 최소화하도록 설계됨

Auxiliary training objective

- 비지도 학습 목표를 추가하는 것도 반지도 학습의 한 형태가 될 수 있음
- 최근 연구에선, target task objective와 함께 보조학습 목표로 language modeling을 추가하여 sequence labeling의 성능을 향상 시킴
- 우리의 연구에서도 보조 학습 목표를 사용하지만, 비지도 pre-training 만으로도 목표 task에서 필요한 다양한 언어적 특성을 효과적으로 학습할 수 있음을 보여줌

→ 기존의 연구에선, fine-tuning 단계에서 보조 학습 목표로 language modeling을 추가했지만, 우리 연구에선 pre-training 단계에서 이미 충분한 언어적 정보를 학습했다고 판단. fine-tuning 단계에서는 추가적인 보조 학습 목표 없이도 높은 성능을 달성할 수 있음을 보여줌

Method

[1] Unsupervised pre-training : 대규모 코퍼스에서 high-capacity language model을 사용

[2] Supervised Fine-tuning: 학습된 모델을 레이블이 있는 데이터로 discriminative task에 적응시킴

Unsupervisied Pre-training

label 이 없는 token 들의 집합 U 에 대해 standard language modeling의 목표는 다음 확률을 최대화 하는 것

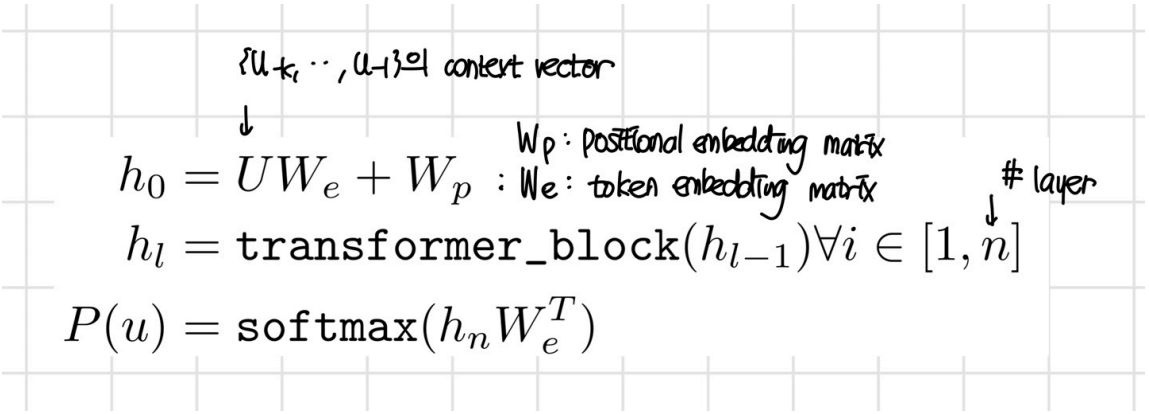
following likelihood:

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

neural network로 modeling된 조건부확률 / maximize 할 목표 / context window의 크기 / neural network의 params

여긴 k개의 token 이 주어졌을 때 u_i (현재 token) 이 나올 확률을 maximize 하는 방향으로 parameter가 update 됨

SGD를 통해 params가 학습됨



Multi-layer Transformer Decoder를 language model로 사용함

: 기존 Transformer의 decoder 구조에서 변경된 것

: 기존의 Transformer 디코더는 다음 단어 예측을 위해, 자기보다 미래 단어를 보지 못하도록 masked 처리, encoder와의 cross-attentino으로 encoder에서 받은 정보를 기반으로 출력을 생성

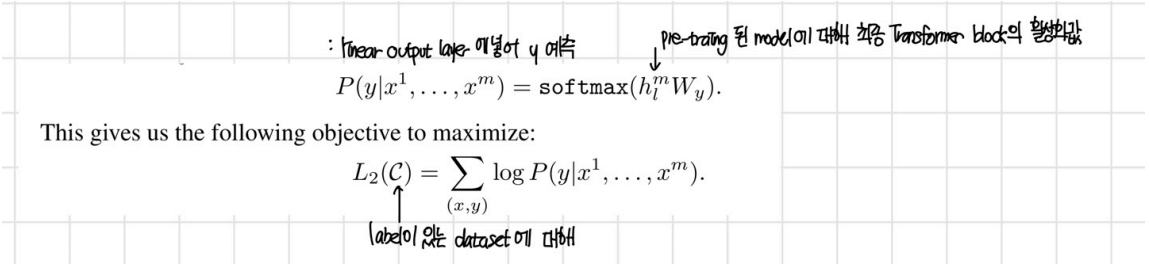
: 본 논문에선, encoder를 사용하지 않고 decoder만 사용

: 이때 완전한 문맥을 활용할 수 있도록 masking 없이 언어 모델을 학습함. 따라서 미래 단어를 예측하는 것이 아닌, 주어진 전체 문맥 정보로 다음 단어를 예측하는 구조

: 기존의 Transformer decoder는 시퀀스를 생성하는 방식이지만, 여기선 판별 모델로 활용하기 때문에 출력 layer를 단순한 linear layer로 처리 → fine-tuning 시 transformer의 최종 벡터를 linear layer에 입력하여 분류 할 수 있도록

: 또한 기존 Transformer은 단일 시퀀스를 처리했지만, 본 연구에서는 질의응답 등 구조화된 입력을 단순 시퀀스로 변환하기 위해 구분자 토큰 (delimiter token)을 추가하여 문장 간 경계를 명확히 함

Supervised Fine-tuning



C는 label 있는 데이터로 각 샘플은 입력 토큰 시퀀스와 label y로 구성

다음의 L2(C)를 최대화 하는 것이 목표

$$L_3(C) = L_2(C) + \lambda * L_1(C)$$

↑ 언어 모델링의 목표 가중치

이때 fine-tuning 단계에서 다음을 보조 학습목표로 추가함

→ 지도 학습 모델의 일반화 성능을 향상, 수렴 속도를 더 빠르게 할 수 있기 때문

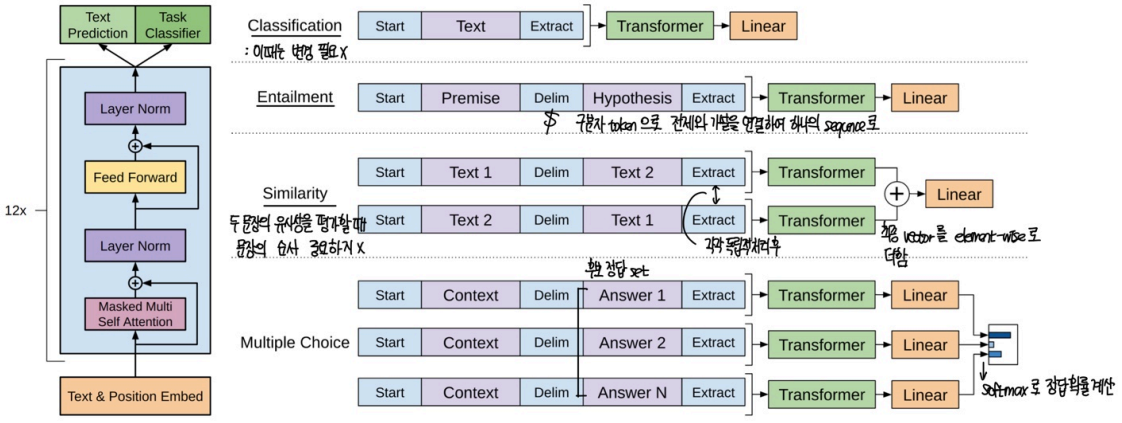
따라서 다음과 같이 새롭게 손실 함수를 최적화한다면, 지도 학습 목표(L2)와 함께 pre-training 된 언어 모델 목표(L1)도 함께 학습이 됨

Task-specific Input Transformation

text classification 과 같은 task는 위의 fine-tuning 방법을 그대로 적용할 수 있으나, 질의응답, 텍스트 entailment(함의) 와 같이 구조화된 입력을 포함하는 특정 작업이 있음. 이때는 모델의 일부 수정이 필요함

이전 연구에서는 transfer learning 이 아닌, task 별로 아키텍처를 변경하는 방법을 사용함 → 과제마다 새로운 모델을 만들어야 한다는 단점이 있음

본 연구에서는, Traversal-style 접근법으로 구조화된 입력을 단순한 시퀀스로 변환하여 모델에 적용



Experiments

- 기존 Transformer 연구를 기반으로 12-layer Transformer Decoder를 사용
- 이때 768개의 hidden state와 12개의 self-attention heads를 사용, position-wise feedforward networks의 inner state dimension은 3072로 설정함

- Adam optimizer with learning rate 2.5e-4 (max) 로 초반 2000번의 업데이트 동안 선형적으로 증가 후 코사인 스케줄링으로 점진적 감소를 할 수 있도록
- 512개의 token 씩 연속된 시퀀스를 sampling 하여 훈련함. 총 학습 횟수는 100 epoch으로
- 이때 residual, embedding, attetion layer에서 dropout prob=0.1 적용 + L2 정규화 사용 (weight decay) 또한 활성화 함수는 **GELU(Gaussian Error Linear Unit)**
- positional embedding에서 기존의 Transformer에서 사용된 것이 아닌, 학습 간으한 positional embedding을 사용함

Unsupervised Pre-training

- BooksCorpus 데이터셋으로 학습함. 여기에는 7000개 이상의 책 장르를 포함. 그리고 연속적인 긴 문장을 포함하고 있어 모델의 장기 의존성 학습에 적합함
- per-token perplexity 18.4 로 매우 낮음. 효과적이 학습이 이루어졌다는 의미

Supervised Fine-tuning

- 비지도 pre-training 단계에서 사용한 하이퍼파라미터를 대부분 재사용 + classifier에 dropout prob = 0.1 추가, learning=6.25e-5, batch size=32 → 이때 대부분의 task에서 3 epochs 만에 빠르게 수렴
- learning rate decay를 전체 학습의 0.2% 동안 warm-up, 이후 선형적 감소
- auxiliary training objective rate(위에서 람다)는 0.5로 설정

Table 1: A list of the different tasks and datasets used in our experiments.

Task	Datasets
Natural language inference	SNLI [5], MultiNLI [66], Question NLI [64], RTE [4], SciTail [25]
Question Answering	RACE [30], Story Cloze [40]
Sentence similarity	MSR Paraphrase Corpus [14], Quora Question Pairs [9], STS Benchmark [6]
Classification	Stanford Sentiment Treebank-2 [54], CoLA [65]

자연어 추론 (Recognizing Textual Entailment, NLI) 란 두개의 문장 (전제 premise, 가설 hypothesis)를 보고 두 문장의 관계가 함의인지, 모순인지, 중립인지를 판단

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

NLI 데이터셋에서 성능

- RTE(뉴스기사)에서는 기존 BiLSTM보다 낮은 성능 → RTE의 데이터셋 크기가 2490개의 샘플로 매우 작아서 multi-task training이 필요할 수도

QA & Commonsense Reasoning (상식적 추론)은 RACE 중학교 고등학교 시험 문제에서 추출된 영어 지문과 관련된 질문 데이터셋을 사용함. CNN, SQuAD 보다 더 많은 추론 유형의 질문을 포함하고 있음. Story Cloze Test는 여러 문장으로 구성된 이야기에서 두개의 선택지 중 올바른 결말을 선택하는 문제

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

QA 데이터셋에서 성능

→ 장기 문맥을 효과적으로 처리할 수 있음을 입증

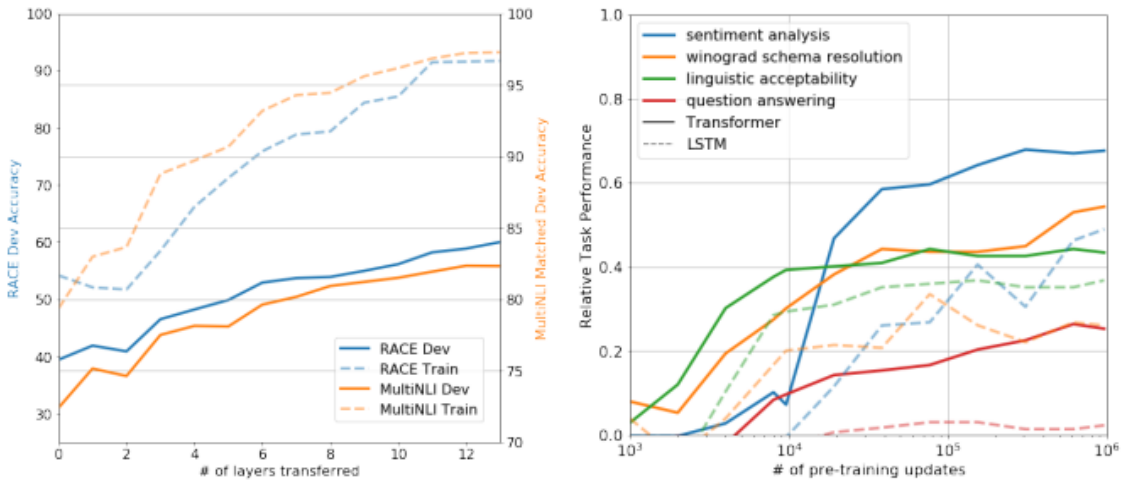
Semantic Similarity (paraphrase detection) 은 두개의 문장이 의미적으로 동일한지를 예측하는 것으로 개념의 다른 표현 방식, 부정어, 구문적 중의성을 인식해야 함

Classification 에서 사용된 CoLA는 전문가가 판단한 문장이 문법적으로 타당한지 여부를 평가, SST2 는 감성 분석으로 문장의 긍부정을 판단

Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

Semantic similarity & classification 에서 성능

- Impact of Number of Layers Transferred (전이되는 layer 수의 영향) : 비지도 pre-training 된 모델에서 몇개의 layer를 전이할 것인지에 따라 성능이 어떻게 변하는지를 확인함
- 또한, Zero-shot 성능 분석을 위해 fine-tuning 없이 pre-training 된 모델 자체가 특정 task를 얼마나 잘 수행할 수 있는지 분석



→ 왼쪽과 같이 전이되는 layer 수가 많을 수록 성능이 향상됨을 확인 meaning pre-trained 모델의 각 layer가 target task를 해결하는데 유용하다는 것을 의미함

→ 오른쪽과 같이 생성 모델의 pre-training 이 다양한 과제와 관련된 기능을 학습하는데 도움이 될 수 있다는 것을 보여줌 + LSTM이 성능에서 변동성이 높은 것을 보아, Transformer의 inductive bias가 전이 학습을 지원하는데 도움이 될 수 있다는 것이 확인됨

- Ablation Study 로 1) 보조 학습 목표 없이 (w/o aux LM) , 2) Transformer vs LSTM 비교 실험, 3) pre-training 없이 Transformer만 학습

Method	Avg. Score	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	MNLI (acc)	QNLI (acc)	RTE (acc)
Transformer w/ aux LM (full)	74.7	45.4	91.3	82.3	82.0	**70.3**	**81.8**	**88.1**	**56.0**
Transformer w/o pre-training	59.9	18.9	84.0	79.4	30.9	65.5	75.7	71.2	53.8
Transformer w/o aux LM	**75.0**	**47.9**	**92.0**	**84.9**	**83.2**	69.8	81.1	86.9	54.4
LSTM w/ aux LM	69.1	30.3	90.5	83.2	71.8	68.1	73.7	81.1	54.6

→ 작은 데이터셋에는 보조학습 목표가 크게 도움이 되지 않았음이 확인됨

→ LSTM을 사용하면 대부분의 과제에서 성능이 감소함

→ 사전 훈련이 없다면 모든 과제에서 성능이 저하

Conclusion

- Generative Pre-training 과 Discriminative Fine-tuning 을 결합하여 task-agnostic 자연어 모델을 위한 새로운 framework를 제안
- 긴 연속적인 텍스트가 포함되어 있는 다양한 corpus에 pre-training 함으로써, 모델이 장기 의존성을 학습하도록 함
- Tranformer 기반 모델과 긴 문맥을 포함하는 데이터가 비지도 사전훈련이 성능이 향상시킬 수 있다는 것을 시사함