[논문리뷰] GPT1 (Improving Language Understanding by Generative Pre-Training)

Abstract

자연어 이해

- 문장삽입(textual entailment)
- 질문, 대답(question and answering)
- 의미유사성 평가(semantic similarity assessment)
- 문서 분류(document classification)

레이블링이 되지 않은 코퍼스들은 풍부하지만, 구체적인 task들의 학습을 위한 레이블링 된 데이터는 부족하다. 따라서 모델이 적절히 수행되는 것이 어렵다.

레이블링 되지 않은 다양한 말뭉치들을 generative pre-training을 시킨 후, 각 특정 task에 맞게 discriminative fine-tuning을 진행함으로써 큰 성능 향상을 얻을 수 있다.

fine-tuning을 진행함에 있어서 모델구조를 최소한으로 변화시키고 효과적으로 전이를 달성하기 위하여 task-aware input transformations를 사용한다.

general task-agnostic모델은 12개의 task 중에서 9개가 SOTA를 달성하였다.

- commonsense reasoning : 8.9% 향상
- question answering : 5.7% 향상
- textual entailment : 1.5% 향상

1. Introduction

날것의 데이터에서 효과적으로 학습하는 능력은 자연어 처리에서 지도학습의 의존성을 낮추는 데 있어서 중요하다. 대부분의 딥러닝 방법들은 레이블링 된 데이터를 필요로 한다. 하지만 레이블링 된 데이터는 부족하기 때문에 많은 도메인에서의 적용이 어렵게 된다.

이러한 상황에서 레이블링이 되지 않은 데이터로 부터 언어적인 정보를 활용하는 모델은 시간과 돈이 많이 드는 레이블링 된 데이터를 대체할 수 있다. 또한 이런 비지도 학습은 지도학습보다 더 좋은 결과를 주기도 한다.

그러나 레이블링이 되지 않은 text로 부터 단어 수준 이상의 정보를 얻는 것이 어려운 이유 두 가지가 있다.

- 전이에 유용한 text representations를 학습하는 데 있어서 어떤 optimizer objectives가 가장 효과적인지가 불분명하다.

- 학습된 representations를 특정 Task로 효과적으로 전이하는 데 있어서 정확히 정해진 방법이 없다.

이러한 부정확한 것들이 semi-supervised 학습의 개발을 어렵게 했다.이 논문에서 우리는 비지도 pre-training과 지도 fine-tuning을 결합한 semi-supervised를 연구할 것 이다.

training procedure

- 뉴럴 네트워크 모델의 초기 파라미터들을 학습하기 위해서 레이블링이 되지 않은 데이터에 대한 언어 모델 목적함수를 사용한다.
- 이러한 파라미터를 지도 목적함수를 사용하여 task에 적용시킨다.

모델을 구성하는 데 있어서 transformer를 사용하였다.

- text의 long-term dependencies를 다루는 데 있어서 더 구조화된 메모리를 제공한다.
- 다양한 작업에 걸쳐 강력한 전이 성능을 제공한다.

task-specific input adaptations 사용

- pre-trained모델에 최소한의 변화를 주면서 효과적으로 fine-tuning을 가능하게 해준다.

자연어 추론, 질의응답, 의미유사성, 분류 총 4가지 분야에서 실험 결과 12개의 task들 중 9개가 SOTA를 달성할 수 있었다.

2. Related Work

Semi-supervised learning for NLP

semi-supervised 학습은 sequence labeling이나 text classification의 적용에 있어서 큰 관심을 가져왔다. 초기의 접근법들은 supervied 모델의 특징으로 쓰이게 될 통계(단어 수준이나 구 단위 수준의)를 계산하기 위하여 레이블링이 되지 않은 데이터를 사용했다. 지난 몇년동안 연구자들은 다양한 과제에서의 성능 향상을 위해 레이블링이 되지 않은 말뭉치로 학습이된 워드 임베딩 사용의 효과를 입증해왔다. 그러나 이러한 방법들은 단어수준의 정보밖에 학습하지 못했다.

최근 연구들은 레이블링 되지 않은 데이터로 부터 단어 수준 이상의 정보를 학습하기 위해 노력하고있다. 레이블링이 되지 않은 데이터로 학습이 된, 구단위 혹은 문장 단위의 임베딩은 다양한 과제를 text에서 벡터 표현으로 적합하게 바꾸기 위해 쓰여왔다.

Unsupervised pre-training

지도학습목적함수를 수정하는 것 대신 좋은 초깃값을 찾는 것이 목표라면 비지도학습은 semi-supervised 학습의 특별한 경우가 된다. pre-training 단계에서는 정규화를 더 잘 되게 도와준다. 따라서 이러한 방식은 image classification, speech recognition, entity disambiguation, machine translation 등 다양한 deep neural networks 과제에서 쓰이고 있다.

비슷한 연구로 언어 모델링 목적함수를 사용하여 pre-training을 하고 지도학습을 통해 목적 과제에 맞게 fine-tuning을 하는 방식이 있다. LSTM을 통해 실험을 한 결과 좁은 범위에 한정이 됐지만

transformer networks를 사용해본 결과 넓은 범위의 언어적 구조까지 가능하였다. 그리고 더 나아가 다른 많은 과제들에도 효과적으로 사용이 가능하였다.

다른 접근법으로는 hidden 표현을 사용하는 것인데, 목표 과제에 대하여 지도학습을 진행할 때 auxiliary features로 사용하는 것이다. 이는 각 과제마다 상당한 양의 새로운 파라미터들을 제공하는데, 반면에 우리는 모델구조의 최소한의 변화만을 필요로 한다 .

Auxiliary training objectives

보조 비지도학습 목적함수를 더하는 것은 semi-supervised 학습의 대안이다. 우리의 실험에도 보조 목적함수를 쓰지만, 우리는 비지도 pre-training이 목표 task에 대한 여러 언어적 측면을 이미 학습했다는 것을 보여줄 것이다.

3. Framework

training procedure

- text의 말뭉치에 대하여 high-capacity 언어모델을 학습한다.
- fine-tuning 단계에서 레이블링 된 데이터를 이용하여 모델을 특정 task에 적용한다.

3.1 Unsupervised pre-training

토큰의 비지도 말뭉치가 주어졌을 때(U={*u*1,⋯,*un*}), 우리는 following likelihood를 최대화 하기 위하여 표준언어 모델링 목적함수를 사용한다.

> L1(U)=∑ilogP(ui ∣ ui−k,⋯,ui−1;Θ)

*k*는 context window의 크기이고, *P*는 파라미터 Θ를 사용하도록 모델링 된다. 이러한 파라미터들은 SGD를 이용하여 학습이 된다. 이 모델은 입력 문맥 토큰에 multi-headed self-attention operation을 적용한 다음 position-wise feedforward layers를 적용하여 대상 토큰에 대한 출력 분포를 생성한다.

실험에서 언어 모델로 multi-layer Transformer decoder를 사용한다.

> h0=UWe+Wphl=transformer_block(hl−1)∀l∈[1,n]P(u)=softmax(hnWeT)

- *U*=(*u*−*k*,⋯,*u*−1) `토큰의 문맥벡터`
- *n* `layer의 수`
- *We* `토큰 임베딩 행렬`
- *Wp* `위치 임베딩 행렬`

3.2 Supervised fine-tuning

모델을 학습시킨 후에, 파라미터를 supervised 목적 과제에 적용한다.

- C `레이블링 된 데이터셋`

- $x1,\cdots,xm$ `입력 토큰들의 sequence`
- y `label`

입력값은 최종 트랜스포머 블록의 activation hlm을 얻기위해 pre-trained 모델을 지나친다. 그리고 나서 y를 예측하기 위해 파라미터 Wy와 함께 선형 출력 층으로 가진다.

> $P(y \mid x1,\cdots,xm)=softmax(hlmWy)$

following objective to maximize

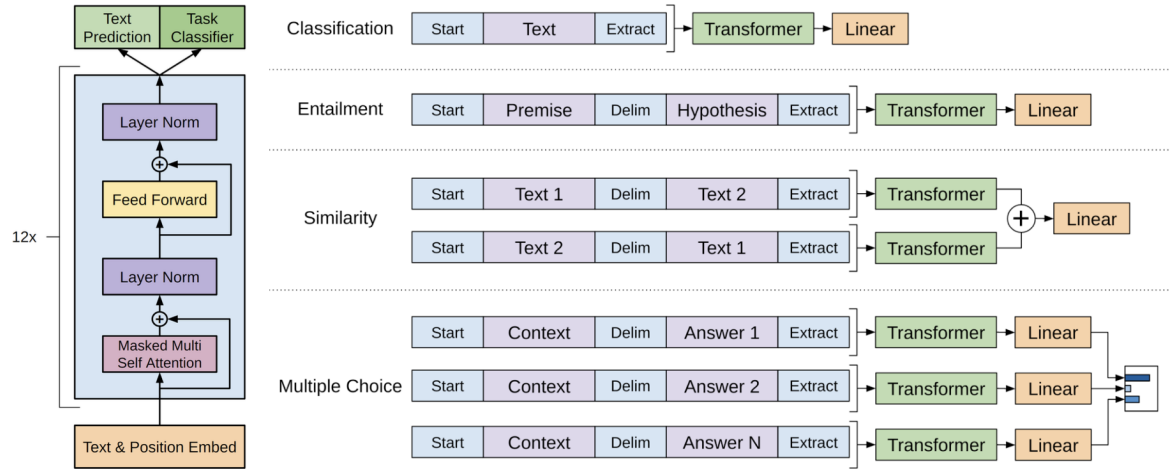
> $L2(C)=\sum(x,y)logP(y \mid x1,\cdots,xm)$

추가적으로 fine-tuning을 위해 auxiliary objective로써 언어 모델을 포함하는 것은 지도모델의 일반화를 향상 시켰으며, 수렴을 가속화했다.

> $L3(C)=L2(C)+\lambda*L1(C)$

결론적으로, fine-tuning 단계에서 추가된 파라미터는 Wy와 구분기호 토큰들을 위한 임베딩이다.

3.3 Task-specific input transformations



pre-trained 모델은 연속적인 text를 통하여 학습이 되었기때문에, fine-tuning을 함에 있어서도 pre-trained 모델이 잘 작동할 수 있도록 input데이터의 약간의 수정이 필요하다.

- structed input -> an rodered sequence

이러한 변화는 기존 모델의 구조를 과도하게 바꾸는 것을 예방해준다.(사진참고) 모든 transformation들은 임의로 초기화된 start <*s*>, end <*e*> 토큰들을 포함하고 있다

Textual entailment

전제 p 와 가정 h 를 구별자 토큰 $ 로 연결한다.

Similarity

딱히 순서에 연관성이 없는 두 문장을 구별자 토큰을 사이에 두고 이어 붙인다.

Question Answering and Commonsense Reasoning

문맥문서 *z*, 질문 *q*, 가능한 답변들 {*ak*}이라고 하면, 문맥문서와 질문, 그리고 각각 가능한 답변들을 구별자 토큰을 사이에 두고 이어붙인다. [*z*;*q*;$;*ak*]이런 sequence들은 모델을 통해 독립적으로 처리가 된 후, softmax 레이어를 통해 정규화가 되어 가능한 답변에 대한 출력 분포를 생성한다.

4. Experiments

4.1 Setup

Unsupervised pre-training

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]

- 사용 데이터셋 : BooksCorpus
- 대안 데이터셋 : 1B Word Benchmark (ELMo에서 사용됐다)크기는 비슷하지만 문장 수준에서 섞여있기 때문에 long-range structure가 파괴되어있다.

Model specifications

- Decoder : 768 dimensional state
- Inner start: 3072 dimension
- Optimizer : Adam
- Learning rate : 2.5e-4 (2000 update 까지는 0에서 부터 서서히 증가) -> cosine schedule로 0으로 감소.
- Epoch : 100
- Batch_size : 64 (sequence : 512 tokens)
- Activation function : Gaussian Error Linear Unit(GELU)
- Attention dropouts : p = 0.1
- Encoding : Bytepair encoding(BPE)
- Tokenizer : spaCy

Fine-tuning details

- 비지도 pre-train에서 쓰인 하이퍼파라미터를 그대로 사용
- dropout(p = 0.1) 추가
- Learning rate : 6.25e-5
- Batch_size : 32
- Epoch : 3
- learning rate decay schedule with warmup over 0.2%
- λ : 0.5

4.2 Supervised fine-tuning

Natural Language Inference

Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

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

다섯개의 데이터셋을 사용하였다.

- Image captions (SNLI)
- Transcribed speech, popular fiction, and government reports (MNLI)
- Wikipedia articles (QNLI)
- Science exams (SciTail)
- News articles (RTE)

성능향상은 아래와 같다.

- SNLI `0.6%`
- MNLI `1.5%`
- QNLI `5.8%`

이는 여러 문장에 대해 더 잘 추론하고 언어적 모호성의 측면을 처리할 수 있는 모델의 능력을 보여준다.

Question answering and commonsense reasoning

Table 3: Results on question answering and commonsense reasoning, comparing our model with current state-of-the-art methods.. 9x means an ensemble of 9 models.

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**

11페이지로 이동

다음 데이터셋을 사용하였다.

- RACE (중고등학교 시험문제)
- Story Cloze Test

성능향상은 아래와 같다.

- Race 5.7%
- Story Cloze Test 8.9%

이는 long-range contexts를 효과적으로 처리할 수 있는 모델의 능력을 보여준다.

Senmantic Similarity

Table 4: Semantic similarity and classification results, comparing our model with current state-of-the-art methods. All task evaluations in this table were done using the GLUE benchmark. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

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**

다음 데이터셋을 사용하였다.

- Microsoft Paraphrase corpus (MRPC)
- Quora Question Pairs (QQP)
- Semantic Textual Similarity benchmark (STS-B)

QQP에서는 BiLSTM + ELMo + Attn 모델보다도 `4.2%` 향상했다.

Classification

- CoLA `35.0 -> 45.4` 성능 향상
- SST-2 `91.3%` accuracy 달성
- GLUE benchmark `68.9 -> 72.8` 성능 향상

> 총 12개의 데이터셋에서 9개가 SOTA를 달성했다. 이러한 결과는 우리의 접근법이 STS-B와 같은 작은 데이터 세트(5.7k)에서 가장 큰 SNI(550550k)에 이르기까지 다양한 크기의 데이터 세트에서 잘 작동한다는 것을 보여준다.

5. Analysis

impact of number of layers transterred

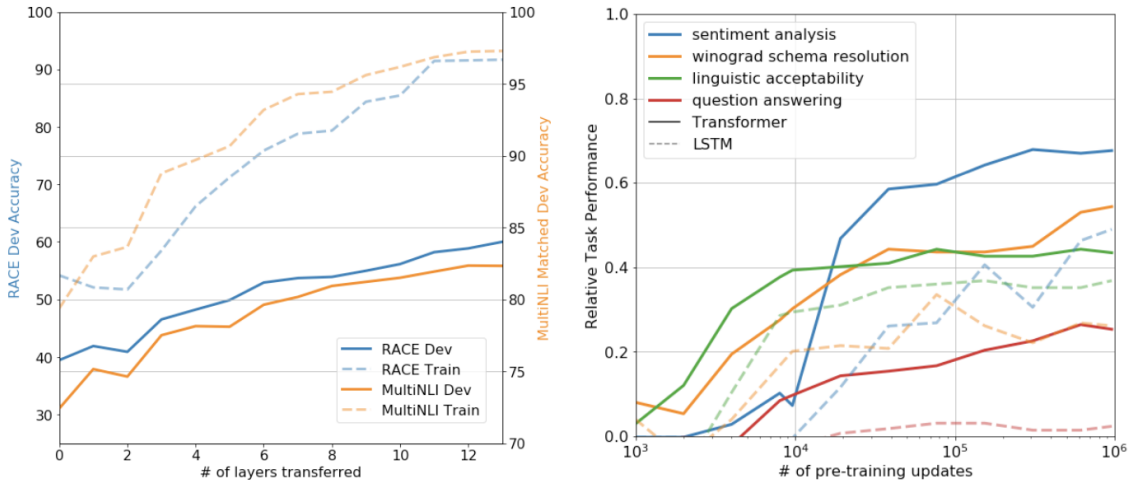


Figure 2: (**left**) Effect of transferring increasing number of layers from the pre-trained language model on RACE and MultiNLI. (**right**) Plot showing the evolution of zero-shot performance on different tasks as a function of LM pre-training updates. Performance per task is normalized between a random guess baseline and the current state-of-the-art with a single model.

Zero-shot Behaviors

우리는 transformer의 언어 모델 사전 교육이 효과적인 이유를 더 잘 이해하고자 한다. 가설은 기본 generative 모델이 언어 모델링 능력을 향상시키기 위해 우리가 평가하는 많은 작업을 수행하는 방법

을 배우고, LSTM과 비교했을 때 transformer의 더 구조화된 attentional 메모리가 transfer을 돕는다는 것이다.

기본 generative 모델을 사용하여 supervised fine-tuning 없이 작업을 수행하는 일련의 heuristic solution을 설계했다. 우리는 이러한 heuristic solution의 성능이 안정적이고 꾸준히 증가하는 것을 보았고, 이는 generative pre-training이 다양한 task의 학습을 지원함을 나타낸다. 우리는 또한 LSTM이 제로샷 성능에서 더 큰 분산을 보인다는 것을 관찰했는데, 이는 트랜스포머 아키텍처의 inductive 바이어스가 transfer에서 도움이 된다는 것을 나타낸다.

Ablation studies

Table 5: Analysis of various model ablations on different tasks. Avg. score is a unweighted average of all the results. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

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

- auxiliary LM objective없이 fine-tuning을 진행하였다.큰 데이터셋에서는 auxiliary objective가 도움을 주지만, 작은 데이터셋에서는 그렇지 않음을 알 수 있었다.
- 같은 프레임워크를 사용하여 transformer와 2048 unit LSTM의 효과를 분석하였다.LSTM은 MRPC에서만 transformer를 뛰어넘는 성능을 보여주었다.
- transformer를 pre-training없이 바로 target task에 지도학습하여 비교하였다.14.8%의 성능하락이 있었다.

6. Conclusion

generative pre-training과 discriminative fine-tuning을 통해 자연어 이해가 뛰어난 프레임워크를 소개했다. 다양한 말뭉치로의 pre-training은 질의응답, 의미 유사성 평가, 문장삽입, 분류 등 다양한 분야에서도 성공적으로 전이가 되었으며 12개의 데이터셋에서 9개가 SOTA를 달성하였다.