Improving Language Understanding by Generative Pre-Training

Alec Radford et al.

TABLE OF CONTENTS

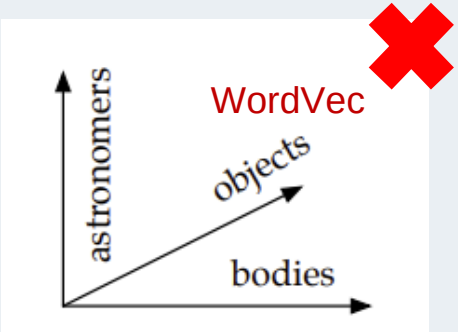
BACKGROUND

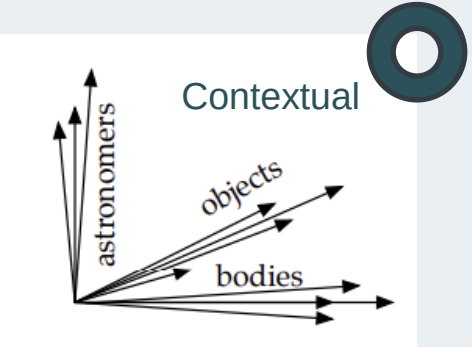
Downstream task에 대해 학습하기 위해서 많은 Labeled data를 사용해야 한다.

WordVec, GloVe처럼 unsupervised 방법으로 **pre-trained word embedding(언어 학습)**을 얻고
Downstream task에 적용하면 성능 향상

BACKGROUND

하지만, WordVec, GloVe 모두 word level information만 담고있다.





BACKGROUND

Unlabeled된 Text 데이터에서 단어 단위보다 더 복잡한 정보를 통해 성능 향상 사례 존재

ex) Cbow, Skip gram

문제

1. 전이 학습을 위한 Text representation (embedding matrix)의 학습 방향 모른다

2. Embedding matrix을 downstream task model에 전이(transfer)하는 효과적인 방법 모른다.

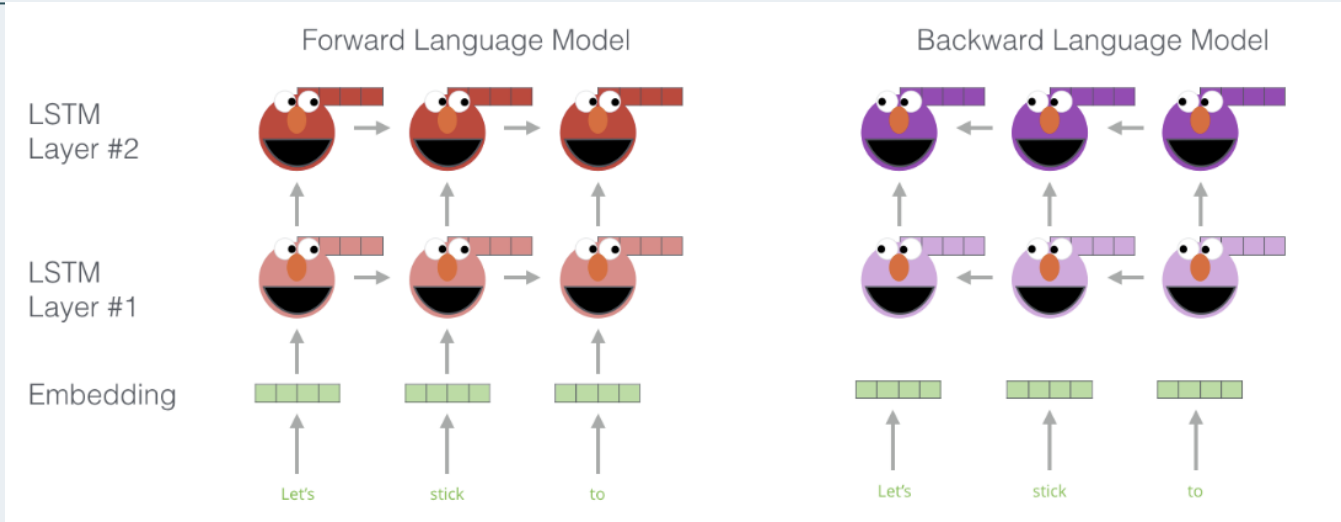
연구 목표

Our goal is to learn a universal representation that transfers with little adaptation to a wide range of tasks.

Unsupervised 방법으로 Unlabeled된 방대한 Text Data + Supervised 방법으로 Labeled data 학습

참고

ELMO: 사전학습된 모델 사용, 처음으로 문맥을 고려한 Word Embedding



다른 점: ELMO는 LSTM 사용

GPT = 최초의 사전 학습된 **Transformer** 모델

Transformer를 사용한 이유

Attention이 LSTM와 같은 RNN보다 Long term dependencies 고려하는데 도움

참고: 기존 연구

Semi supervised Learning for NLP
기존에도 Unlabeled data 학습을 통해 얻은 word representation → Specific task에 적용(But 다 word-level)

Unsupervised Pre-training
Supervised 학습 방법의 시작점을 효율적으로 초기화
+ 정규화, 일반화 능력 향상

<-> 기존에는 LSTM 사용
<-> downstream에 적용할 때 구조 변화 필요

Auxiliary unsupervised training objectives
추가적인 비지도 학습 목표를 사용하여 성능 향상
(ex: Sequence labeling을 하기 위한 supervised 학습 방법에 language modeling task 넣기)

→ **GPT에서 사용**

GPT 모델 Unsupervised pre-training

Standard Language Modeling 학습 방법

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

최대화

Input sequence의 토큰들을 순서대로 옳게 예측하는 확률을 최대화

= generative pre-training

$$P(u) = \texttt{softmax}(h_n W_e^T)$$

$$h_l = \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n]$$

.
.
.

$$h_l = \texttt{transformer\_block}(h_{l-1}) \forall i \in [1, n]$$

12

$$h_0 = UW_e + W_p$$

단어 임베딩 + 위치 임베딩

GPT 모델 Supervised Fine-tuning

labeled dataset $\mathcal{C}$ 안 하나의 데이터 = $x^1, \ldots, x^m$, along with a label y

Linear + softmax

$$P(y|x^1, \ldots, x^m) = \texttt{softmax}(h_l^m W_y)$$

Fine tuning model의 parameter

Pretrained model output

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m)$$

최대화

GPT 모델 Supervised Fine-tuning

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m)$$

labeled dataset $\mathcal{C}$

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

Hyperparameter

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

최대화

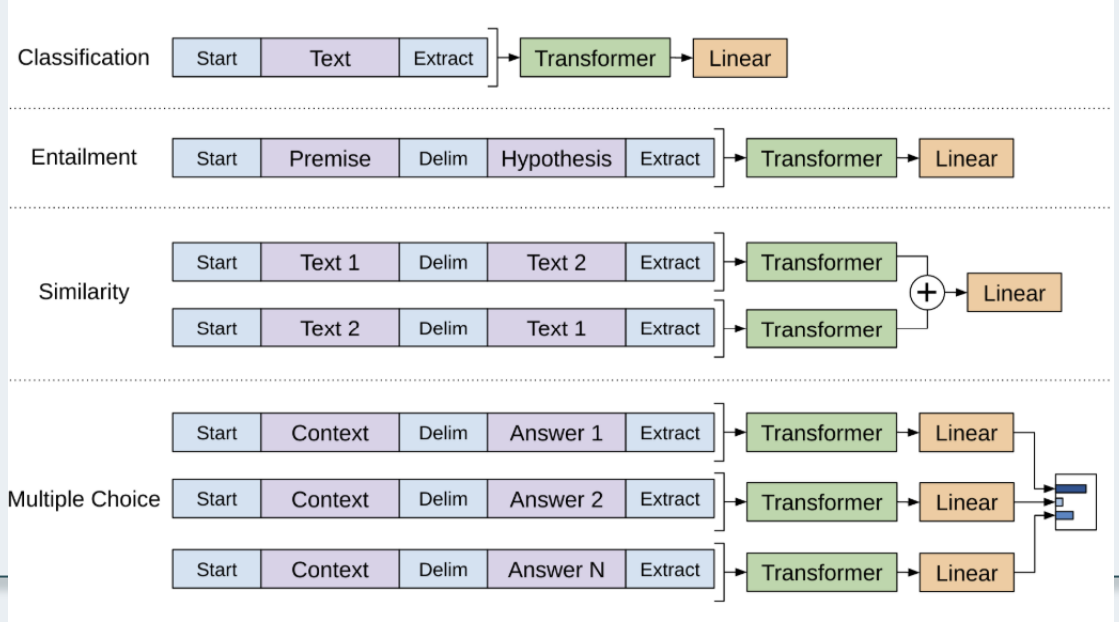
Auxiliary objective 추가: Language Modeling task 넣기
+ improves model's generalization
+ accelerates convergence

Task specific input transformation : Traversal-style approach

기존
Pretrained와 task specific model 구조적 차이 컸다

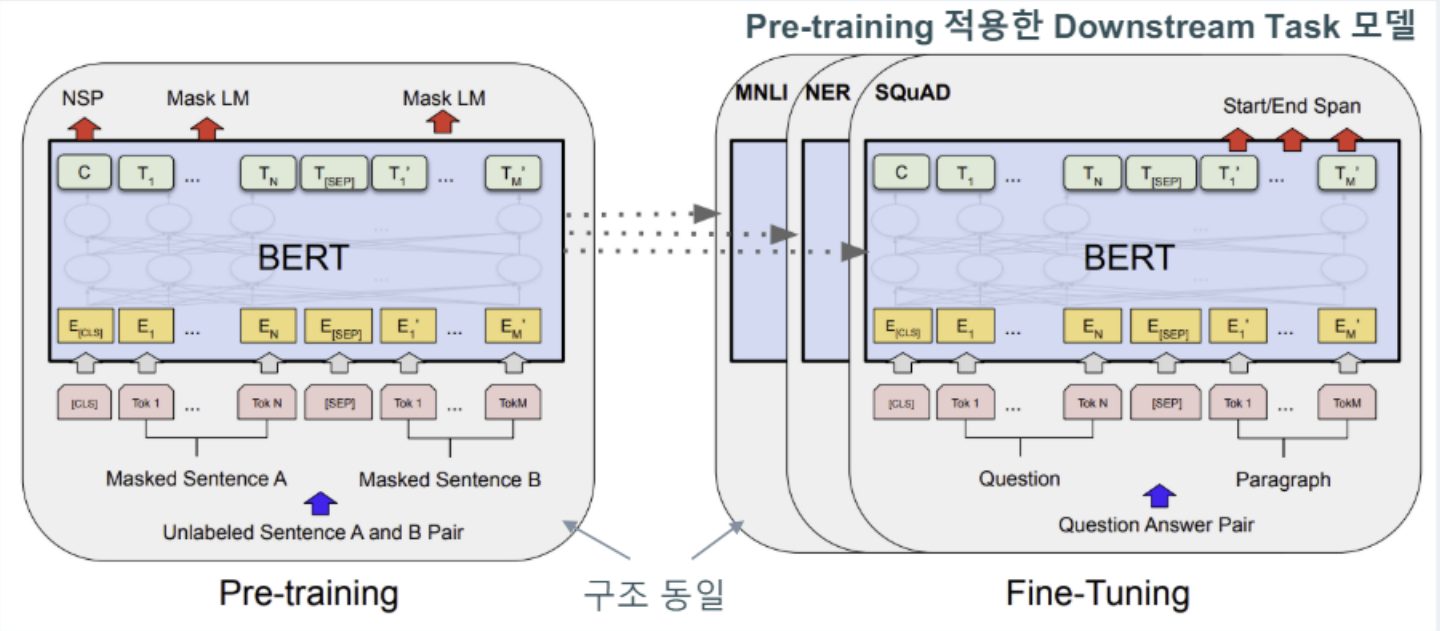
•Task-aware input transformations during fine-tuning achieve effective transfer while requiring minimal changes to the model architecture.

현재
Pretrained와 task specific model 구조적 차이 적다



BERT Bidirectional Encoder Representations from Transformers

Multi layer bidirectional Transformer Encoder



Experiments: 입력 데이터

Unsupervised Pre-Training

다양한 장르의 책 7000권

→ 문장들이 쭉 이어지므로 standard language modeling을 사용하는 Decoder based 구조와 어울린다.

→ Learn to condition on long range information

ELMO → 문장 level 데이터로 학습했지만 문장들 간의 순서를 고려하지 않은 shuffling 했음
(long range structure x)

Experiments: Model Specification

Original Transformer와 비슷하다. (12 decoder layers, 768 dimensional states …)

+ BPE (byte pair encoding)
+ Gaussian Error Linear Unit (GELU) 활성화 함수
+ positional embeddings instead of sinusoidal version

Experiments: Fine tuning details

Dropout rate 0.1, learning rate 6.25e-5, …
+ linear learning rate decay schedule
+warmup over 0.2% of training

Experiments: Supervised Tuning on NLP Tasks

NLI(추론) Task

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

Data 수가 적은 Task에서 성능 향상 x

Experiments: Supervised Tuning on NLP Tasks

Question Answering and Commonsense Reasoning(추론) Task

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**

= Long range context 잘 이해할 수 있다

Experiments: Supervised Tuning on NLP Tasks

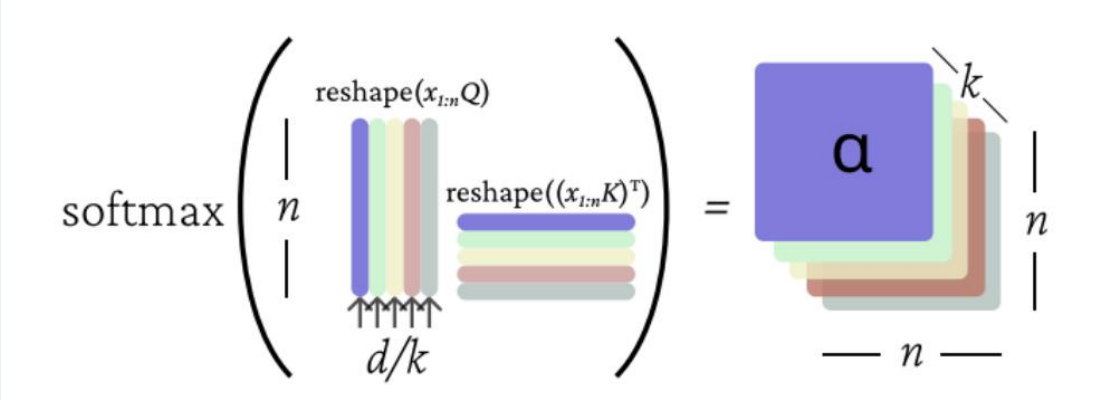
Semantic Similarity and Classification Task

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**

결과적으로 9/12 개의 task에 대해 SOTA

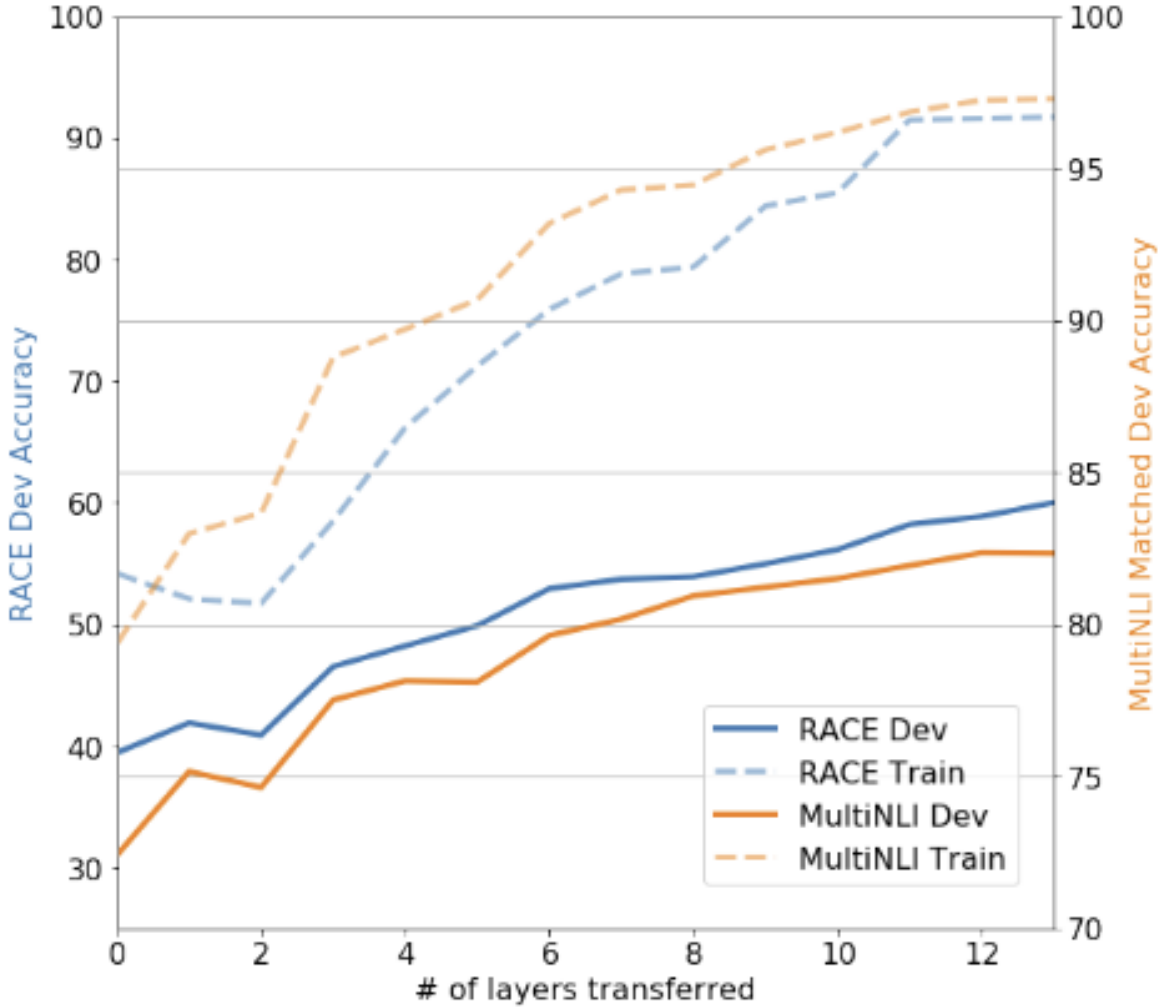
Analysis: Impact of number of layers transferred

Transformer Decoder based Pre-training을 통해 여러 embedding layers 생긴다



Multihead attention

Head 수만큼 다른 관점에서 input을 해석. 즉, layer마다 input을 보는 관점이 다름



This indicates that each layer in the pre-trained model contains useful functionality for solving target tasks.

모델이 특정 작업에 대한 지도 학습을 진행하지 않았음에도
처음 보는 작업에 대해 예측을 수행할 때 모델이 보여주는 행동

Analysis: Zero-shot behaviors

왜 language modeling을 통한 transformer pre-training이 효과적일까?

"A hypothesis is that the underlying generative model learns to perform many of the tasks we evaluate on in order to improve its language modeling capability and that the more structured attentional memory of the transformer assists in transfer compared to LSTMs"

Language modeling을 통해 transformer based pre-training 하는 과정에서 이미 우리가 Test할 때 사용하는 Task를 배운다.
== Test가 모델이 언어를 잘 이해했는지 확인하는 것인데 pre-training을 통해서 언어를 학습하므로 Pre-training에서 이미 NLP task를 학습했을 것이다

Transformer는 LSTM보다 long term dependency 고려 잘 하므로 효과적이다.

Ex) I work in ____. → 다음에 올 것이 기업/단체 이름임을 학습 == **Named Entity Recognition**

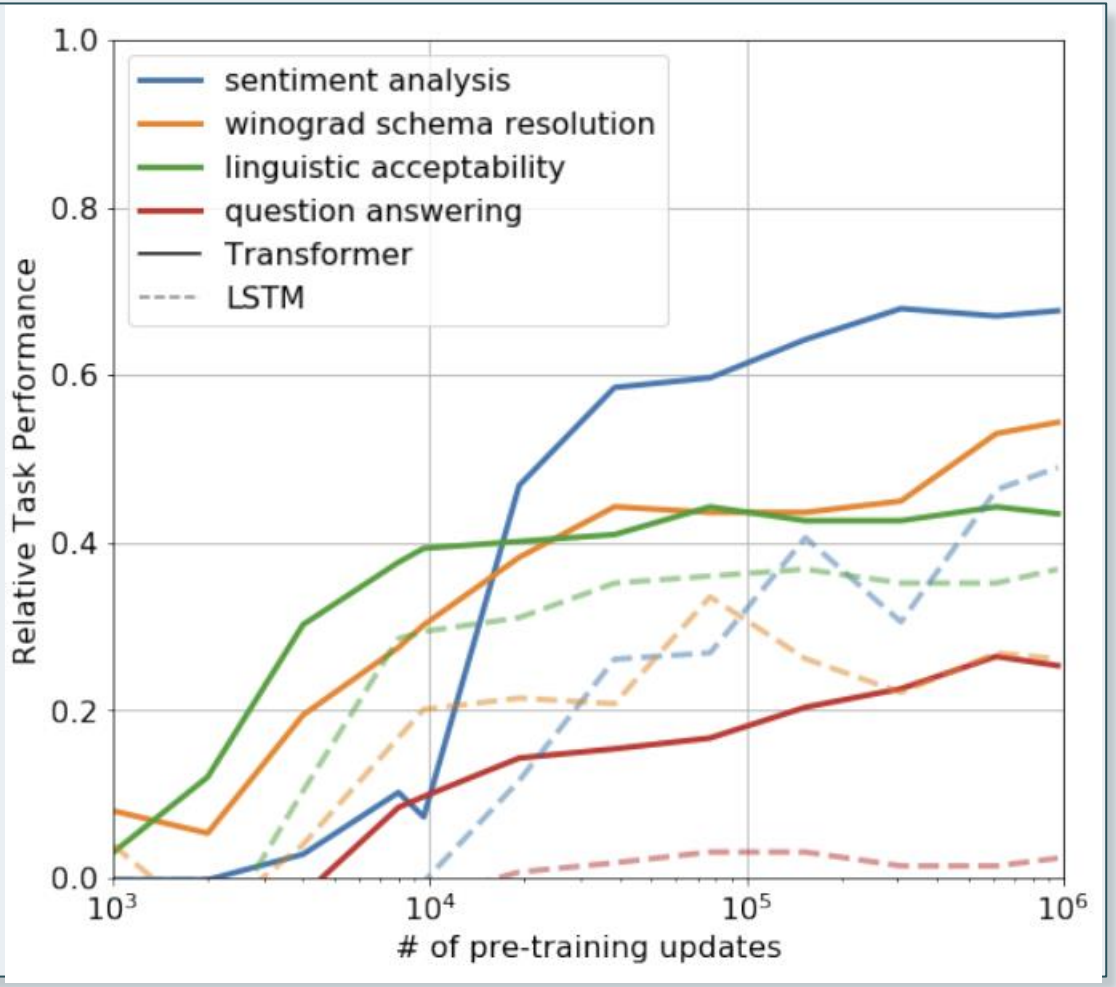
Analysis: Zero-shot behaviors

Supervised fine-tuning을 하지 않고 **transformer**와 **LSTM** 각각 **unsupervised pre-training**만 한 결과를 통해 NLP task test

+ Pre-training을 할수록 성능 올라감

+ LSTM으로 pre-training 할 때는 단순하게 증가 안 하지만 (high variance)

+ transformer로 pretraining할 때는 계속 증가하는 추세



Analysis: Ablation studies

Auxiliary language modeling objective: NLI에 효과적
+ data 수 큰 거에 효과적

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 language modeling objective 별로다

Analysis: Ablation studies

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

Transformer 사용하는 것이 맞다

Analysis: Ablation studies

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

Transformer에 Supervised learning만 한 것

Pre-training의 성능

Analysis: Ablation studies

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

Pre-training의 성능

GPT-2 ?

Language Models are Unsupervised Multitask Learners

https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf#page=1&zoom=100,0,0

NLI(추론) Task

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

Data 수가 많은 Task에서 성능 향상 큼

Given the strong performance of our approach on larger NLI datasets, it is likely our model will benefit from multi-task training as well but we have not explored this currently.