Improving Language Understanding by Generative Pre-Training

김충현

Introduction

GTP는 여러 태스크로 효과적인 transfer가 가능한 universal representation을 학습하는 것이 목표

- 어떤 Architecture를 사용할 것인가?

  -> Transformer

- 어떤 방식으로 representation을 학습할 것인가?

  -> Pre-train Language model with unsupervised learning

- 어떤 방식으로 representation을 task-specific 하게 transfer 할 것인가?

  -> Very small modification to pre-trained network

Introduction

Raw data로부터 unsupervised learning으로 효율적으로 학습하는 건 매우 중요하다
Why?

- Supervised Learning in NLP is a hard task
    - Low-resource NLP problem
        - Not enough labeled data
        - Low-resource languages
        - Spoken language domain <-> Written language domain
        - Mismatch of corpora and actual meaning

Introduction

Raw data로부터 unsupervised learning으로 효율적으로 학습하는 건 매우 중요하다
Why?

- Enough data?
    - Pretrained word-embedding improves performance

Introduction

하지만, 효과적인 semi-supervised learning 을 NLP 에 적용하는 건 쉽지 않다
Why?

- Unclear transfer methods
- Unclear de facto standard optimization method for embedding
  - ELMO (language modeling) ?
  - CoVe (machine translation) ?
  - Discourse coherence ?

Background : Unsupervised pre-training

Word-level embedding에서 Context-level embedding으로 발전

Traditional word vectors

- Bag of Words
- TF-IDF
- Distributional Embeddings
- …

Word Embeddings

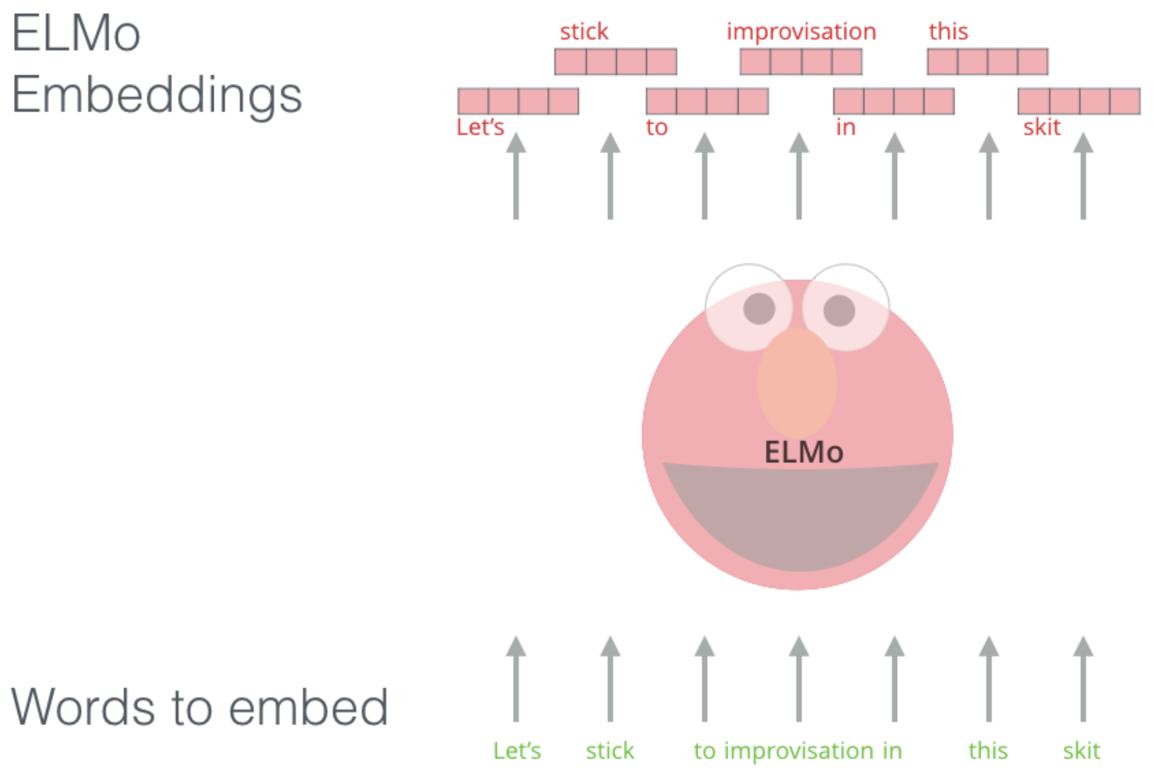
- Word2Vec
- GloVe
- FastText

More than word-level semantics

- ELMo
- CoVe
- …

Background : Unsupervised pre-training

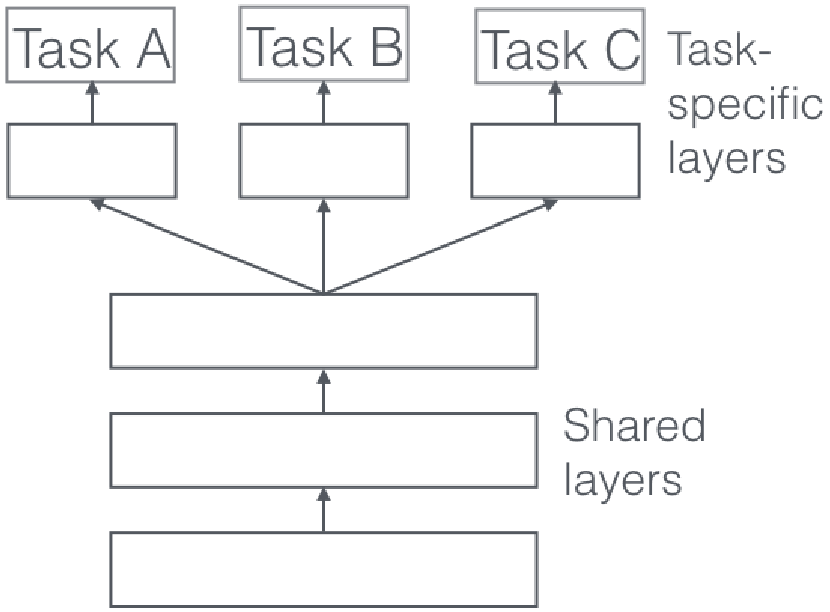
Word-level embedding에서 Context-level embedding으로 발전



http://jalammar.github.io/illustrated-bert/

Background : Auxiliary training objectives

Language modeling objective 등 unsupervised training objective를 두면 성능 향상 효과



http://ruder.io/multi-task/index.html#auxiliarytasks

$L_1(C)$: some task-specific objective

$$L_1(C) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m)$$

$L_2(C)$: unsupervised training objective

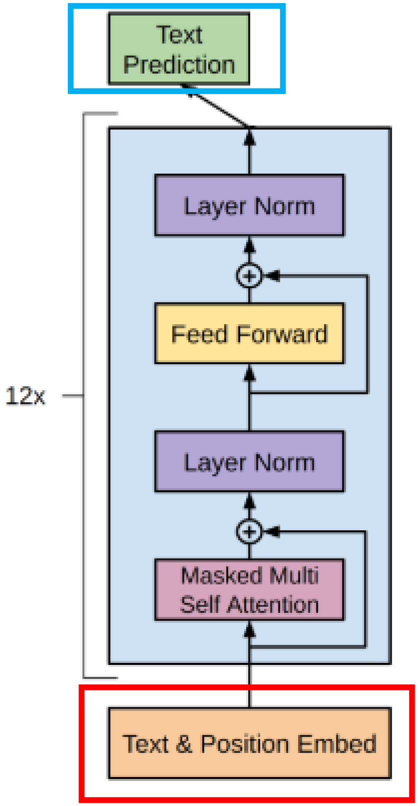
$$L_2(C) = \sum_{i} \log P(u_i|u_{i-k}, \ldots, u_{i-1}; \Theta)$$

GTP

Unsupervised training 후, Supervised training이라는 두 단계로 나뉜다

Unsupervised pre-training → Supervised fine-tuning

GTP Unsupervised pre-training



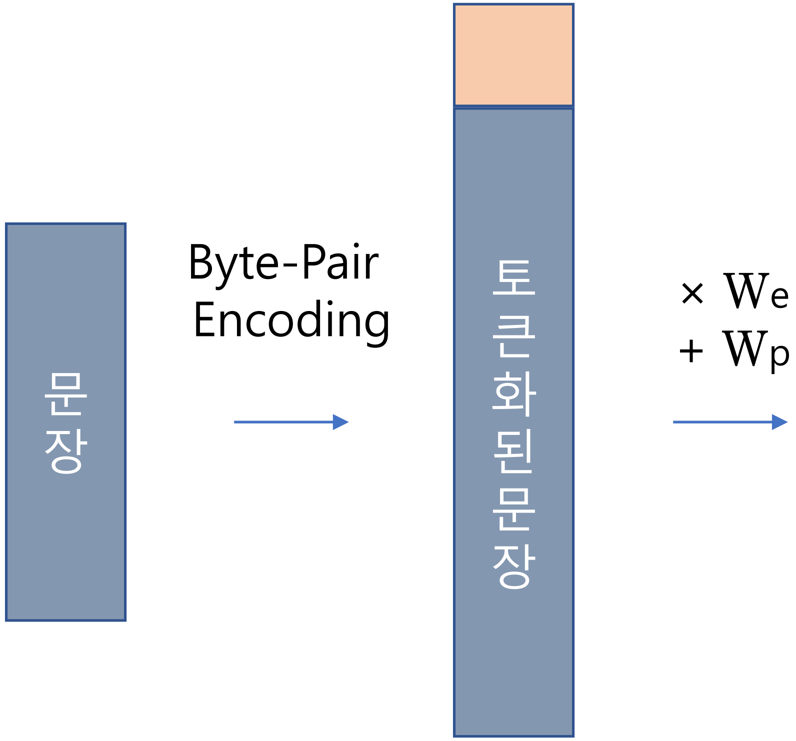
$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

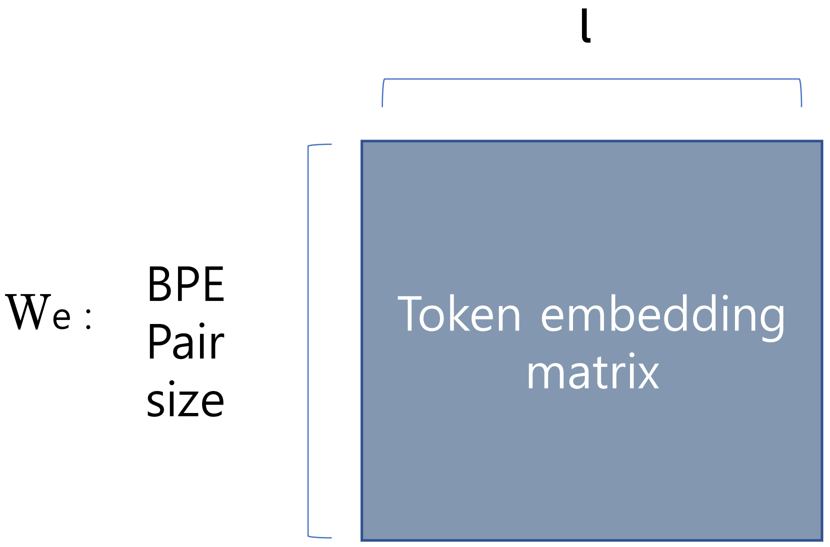
$$\boxed{h_0 = UW_e + W_p}$$

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

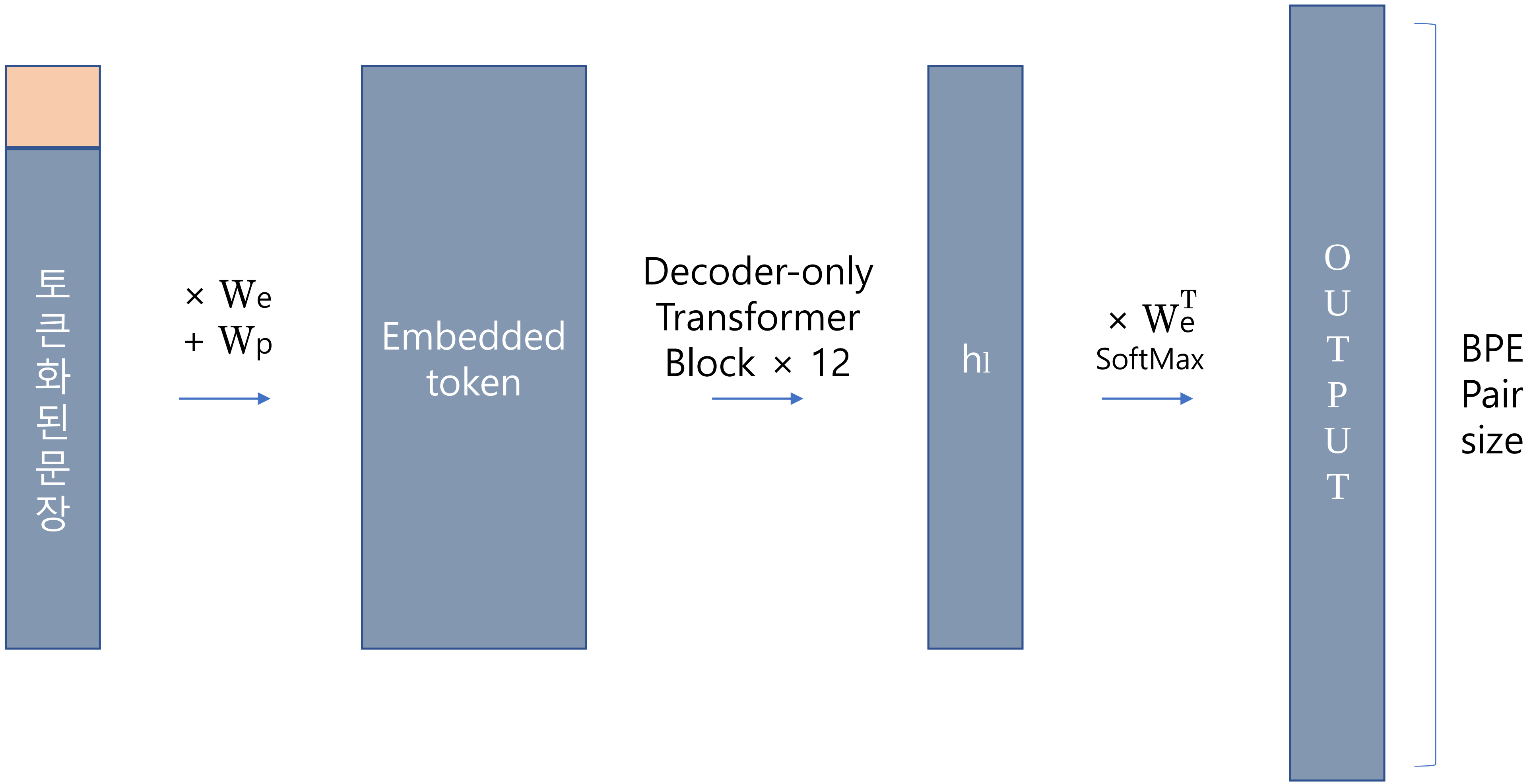
$$\boxed{P(u) = \texttt{softmax}(h_n W_e^T)}$$

GTP Unsupervised pre-training

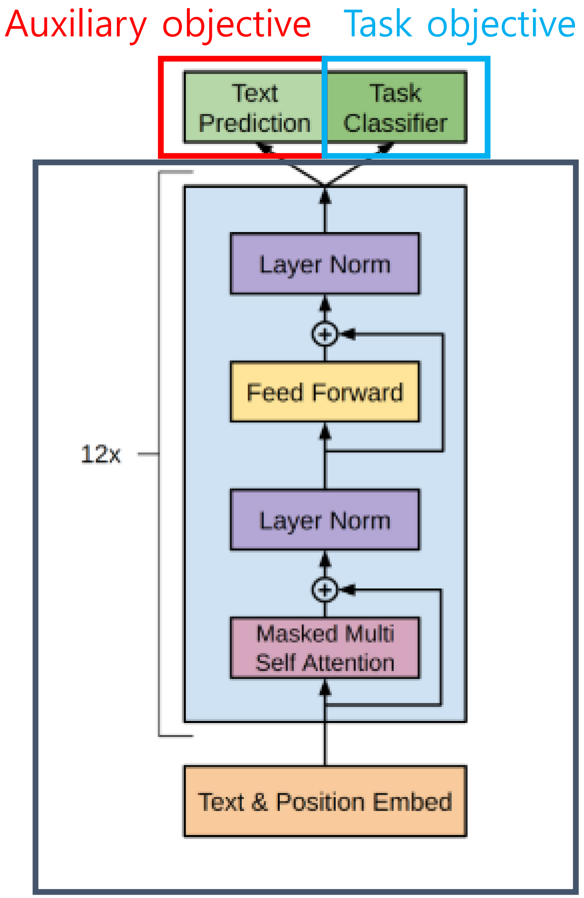




GTP Unsupervised pre-training



GTP Supervised training



Pre-trained network

단 한 레이어의 linear layer만 추가됨

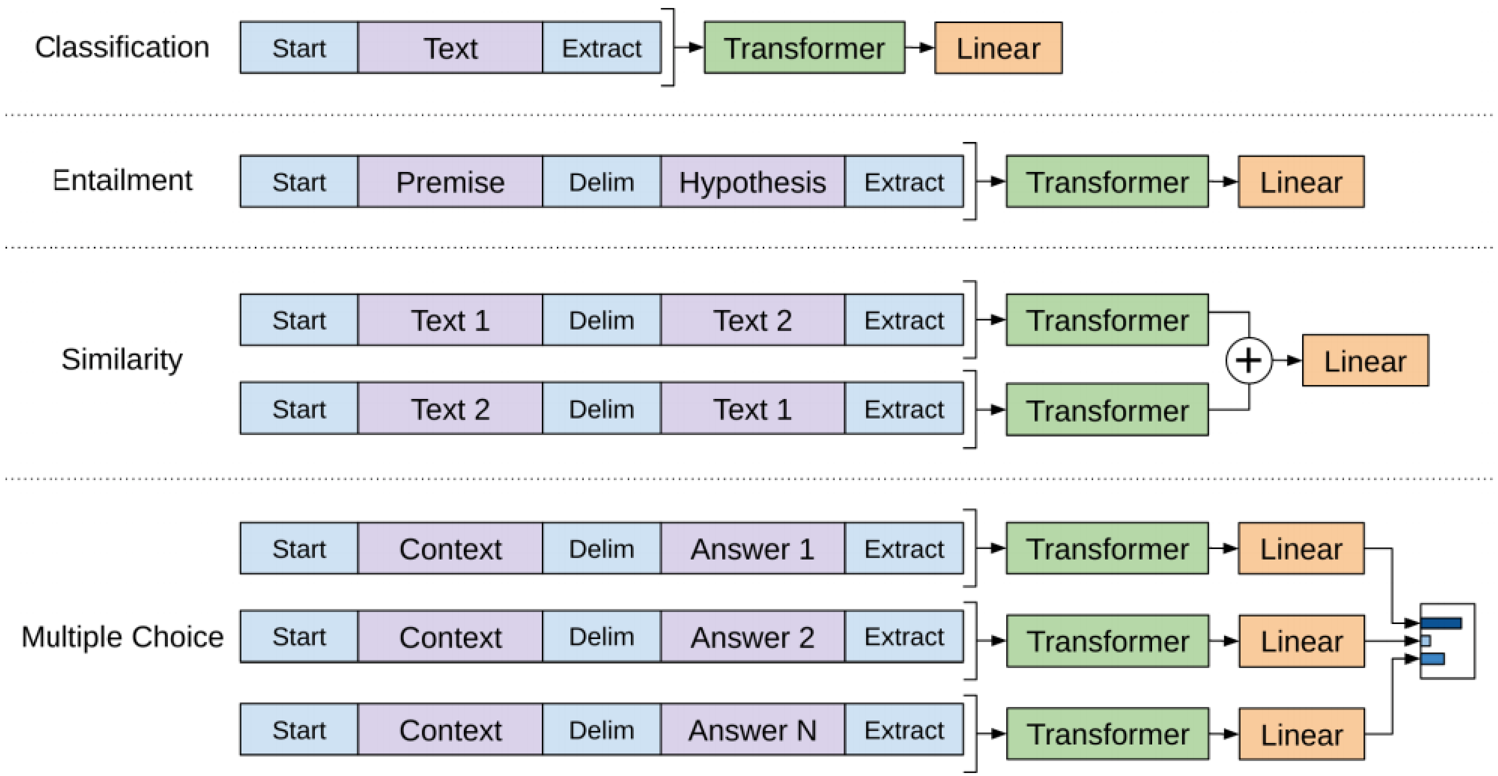
$$P(y|x^1,\dots,x^m) = \texttt{softmax}(h_l^m \boxed{W_y}). \quad (3)$$

This gives us the following objective to maximize:

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1,\dots,x^m). \quad (4)$$

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

GTP Supervised inference



Structured input은 delimiter로 구분되는 ordered sequence로 변환

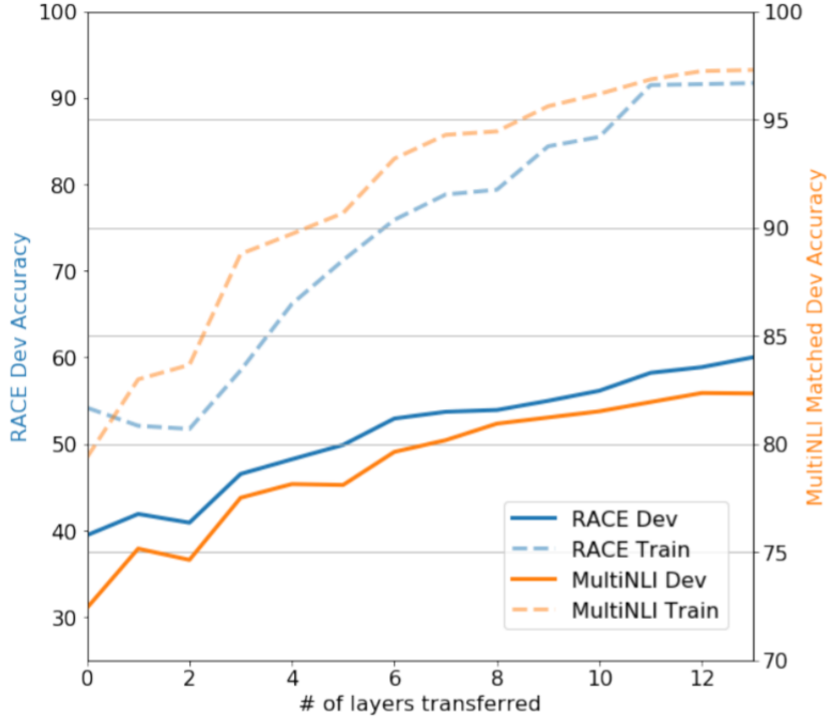
Inference 때는 auxiliary task head 사용하지 않아도 됨

Model specifications

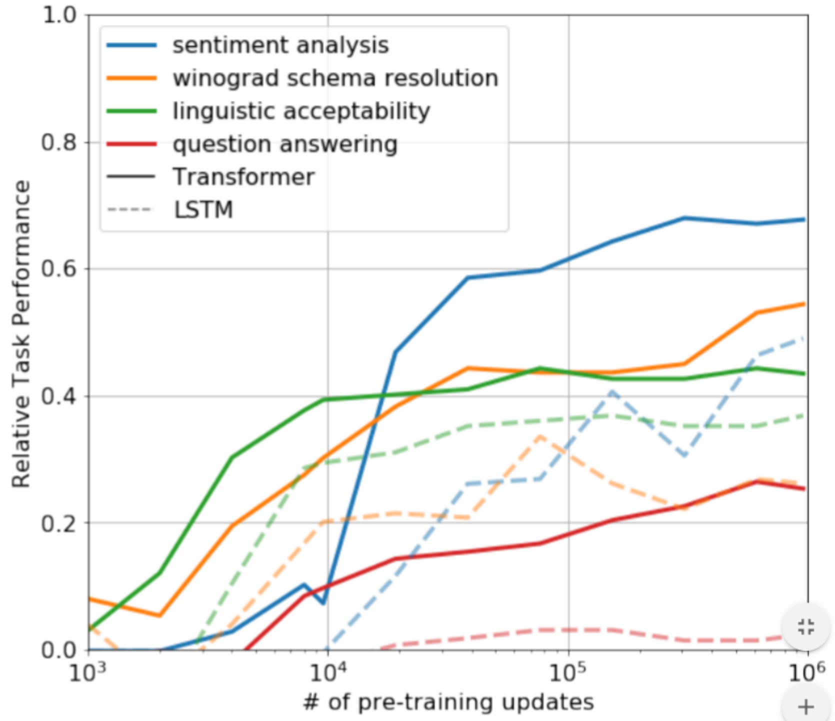
- 12 decoder-only transformer
- Adam optimization
- Cosine annealing : learning rate schedules with restart
- Input : Contiguous sequences of 512 tokens
- Weight initialization of N(0, 0.02)
- BPE with 40,000 merges

Analysis

Impact of number of layers transferred



Zero-shot behaviors



Analysis

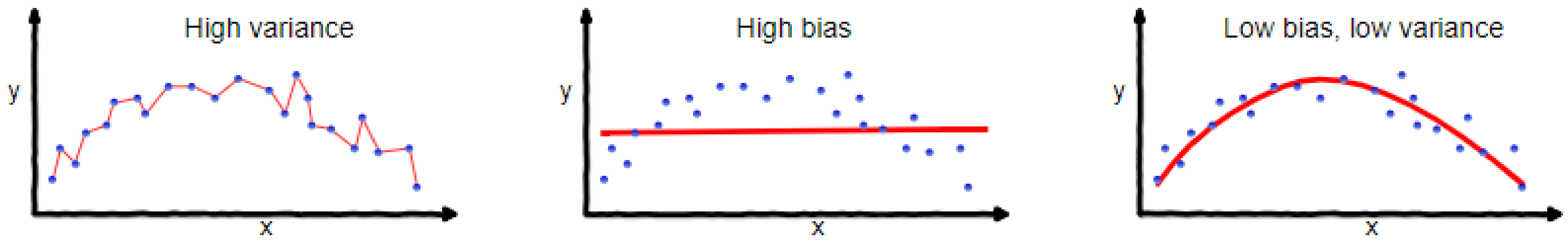
Table 5: Analysis of various model ablations on different tasks. Avg. score is a unweighted average of all the results. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

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

1. Larger dataset benefit from auxiliary tasks
2. Transformer helps
3. Pre-training helps

Analysis

1. Larger dataset benefit from auxiliary tasks
   -> auxiliary tasks introduce regularization effect and parameter noise
   -> 모델의 bias가 증가



overfitting underfitting Good balance

Analysis

Drawbacks

1. Compute requirements : 1 month on 8 GPUs
2. Dataset의 한계 : Books, text available on internet do not contain complete information about world