BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

2019.11.06

박승일

CONTENTS

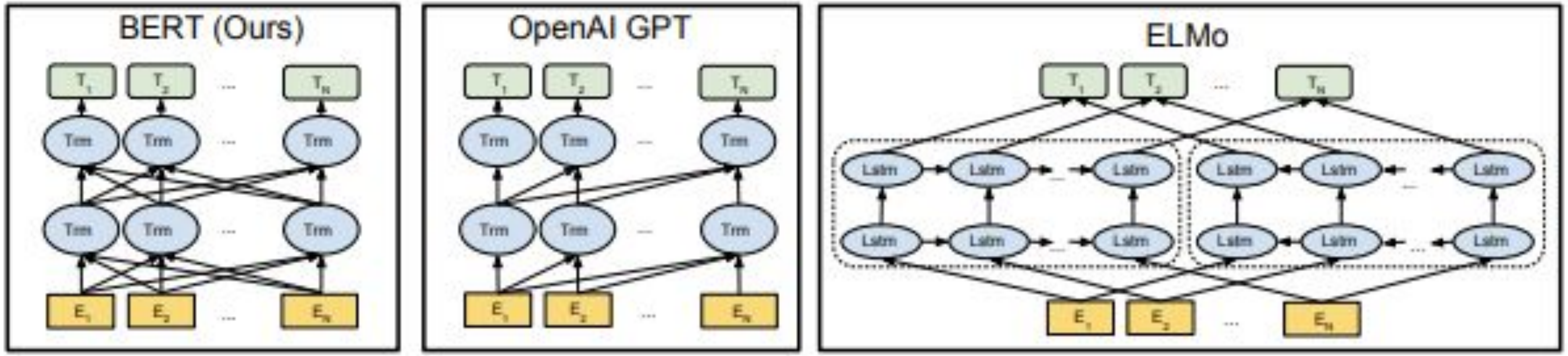
1. Introduction

- Bidirectional Encoder Representations from Transformer
- wiki + BooksCorpus(total 3300M words) unlabeled data pre-training and labeled data transfer learning
- masked language models(MLM), next sentence prediction(NSP) on pre-training
- BERT advances the state of the art for eleven NLP tasks
- with few architecture change, small fine-tuning data and epochs

2. Related work

- **Unsupervised Feature-based Approaches**
  - ELMo
  - Universal Sentence Encoder

- **Unsupervised Fine-tuning Appoaches**
  - OpenAI GPT

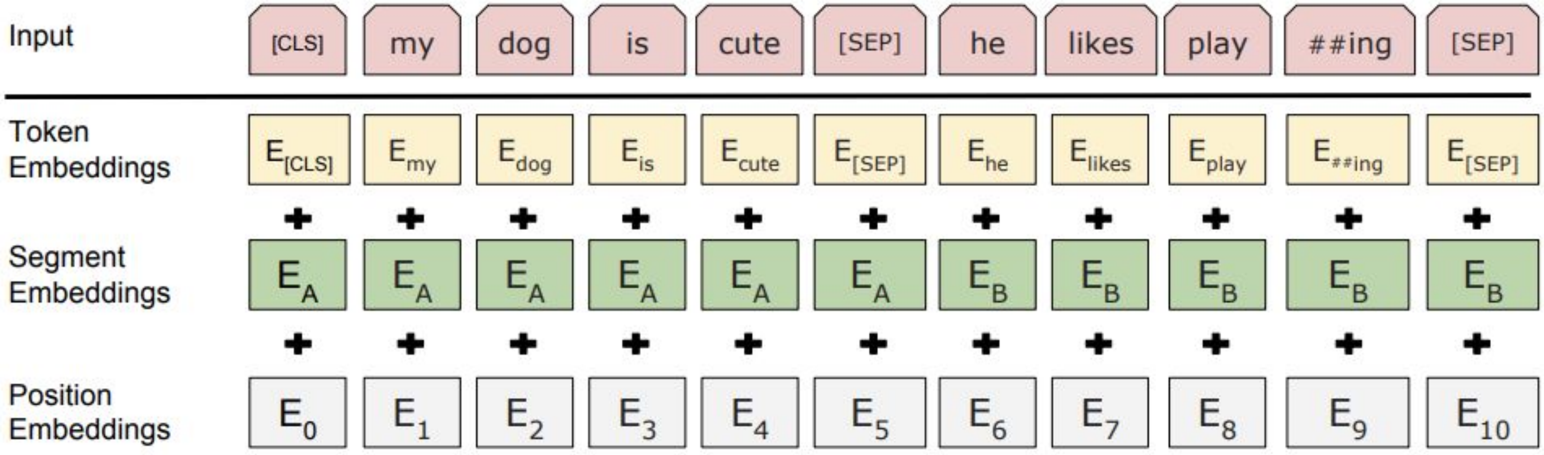


3. BERT

- BERT_BASE (L=12, H=768, A=12, Total Parameters=110M)
  same model size as OpenAI GPT for comparison

- BERT_LARGE (L=24, H=1024, A=16, Total Parameters=340M)

- Input Output Representations



3-1. Pre-training BERT(Masked LM:Cloze)

1. masked 15% of all wordpiece at random
2. replace “masked” words with [MASK] token 80%, random token 10%, unchanged token 10% for fine-tuning. Because on fine-tuning there is no [MASK] token.

   original input -> 나는 나를 사랑한다 (‘사랑한다'라는 wordpiece를 masking한다 가정)
   전체 시간의 80%-> 나는 나를 [MASK]
   전체 시간의 10%-> 나는 나를 싫어한다
   전체 시간의 10%-> 나는 나를 사랑한다

3. predict the masked words

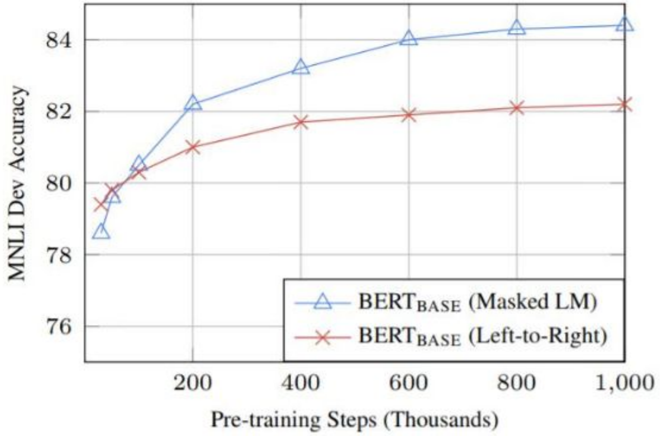
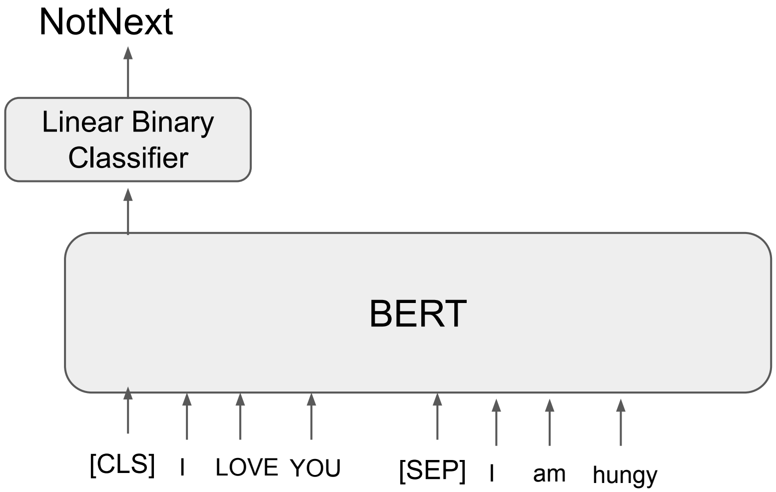


Figure 4: Ablation over number of training steps. This shows the MNLI accuracy after fine-tuning, starting from model parameters that have been pre-trained for k steps. The x-axis is the value of k.

3-1. Pre-training BERT(Next Sentence Prediction)

for understanding the relationship between two sentences, which is not directly captured by language modeling.



Next Sentence Prediction The next sentence prediction task can be illustrated in the following examples.

Input = `[CLS] the man went to [MASK] store [SEP]`

`he bought a gallon [MASK] milk [SEP]`

Label = `IsNext`

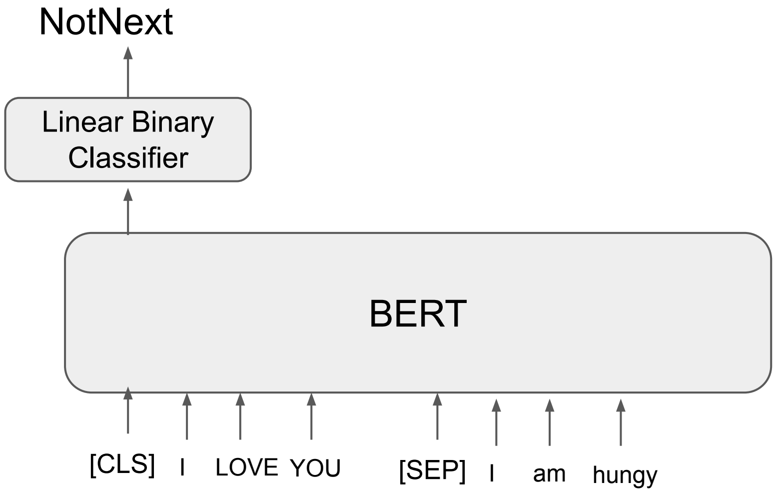
Input = `[CLS] the man [MASK] to the store [SEP]`

`penguin [MASK] are flight ##less birds [SEP]`

Label = `NotNext`

3-1. Pre-training BERT(Next Sentence Prediction)

for understanding the relationship between two sentences, which is not directly captured by language modeling.



Next Sentence Prediction The next sentence prediction task can be illustrated in the following examples.

Input = `[CLS] the man went to [MASK] store [SEP]`

`he bought a gallon [MASK] milk [SEP]`

Label = `IsNext`

Input = `[CLS] the man [MASK] to the store [SEP]`

`penguin [MASK] are flight ##less birds [SEP]`

Label = `NotNext`

3-1. Pre-training BERT(Pre-training data)

Use **document level corpus** rather than shuffled sentence-level corpus in order to extract **long contiguous sequences**.

- BooksCorpus(800M words)
- English Wikipedia(2500M words)

3-1. Fine-tuning BERT

Compared to pre-training, fine-tuning is relatively **inexpensive**
Using **same** pre-trained model

1. **Using pair sentence**
- sentence pairs in paraphrasing
- hypothesis-premise pairs in entailment
- question-passage pairs in question answering
- degenerate text-∅ pair in text classification or sequence tagging

2. **Using CLS representations**
- entailment
- sentiment analysis

3-1. Fine-tuning BERT

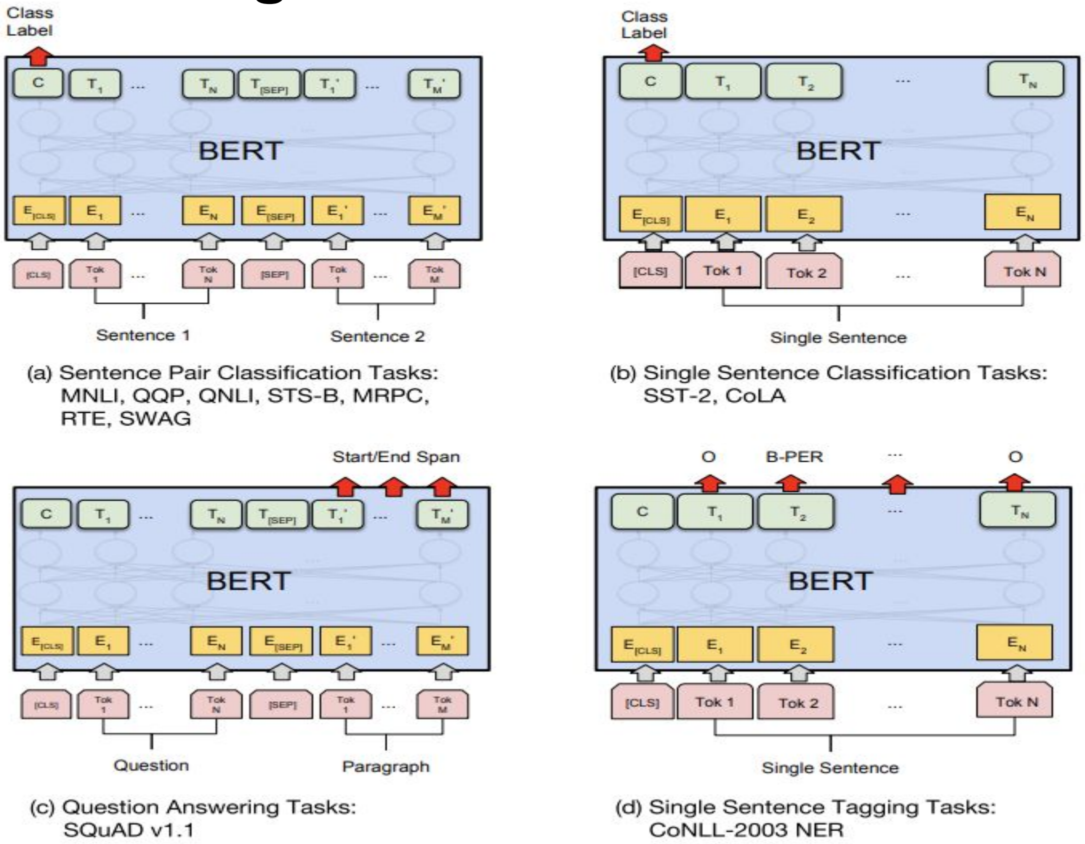


Figure 4: Illustrations of Fine-tuning BERT on Different Tasks.

4-1.GLUE

BERT IS REALLY GOOD for all GLUE tasks!!

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

BERT_LARGE unstable on **small datasets**, so we ran several **random restarts** and selected the best model on the Dev set.

BERT_LARGE > BERT_BASE with very little training data.

4-3. SQuAD

- Stanford Question Answering Dataset
- Given a question and a passage from wikipedia containing the answer.
- Task is to predict answer text span in passage
- SQuAD 2.0 include distinguish ability to abstain question which cannot be answered

4-4. SWAG

- situations with adversarial generations
- the task is to choose the most plausible continuation among four choices.
- make four sequences which concat question and answer, and get score of each answers

5.1 Effect of Pre-training Tasks

Tasks	Dev Set MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$\text{BERT}_{\text{BASE}}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

Table 5: Ablation over the pre-training tasks using the $\text{BERT}_{\text{BASE}}$ architecture. "No NSP" is trained without the next sentence prediction task. "LTR & No NSP" is trained as a left-to-right LM without the next sentence prediction, like OpenAI GPT. "+ BiLSTM" adds a randomly initialized BiLSTM on top of the "LTR + No NSP" model during fine-tuning.

Using Next Sentence Prediction, Masked Language Model both is Excellent!!

5.3 Feature-based Approach with BERT

System	Dev F1	Test F1
ELMo (Peters et al., 2018a)	95.7	92.2
CVT (Clark et al., 2018)	-	92.6
CSE (Akbik et al., 2018)	-	**93.1**
Fine-tuning approach		
$\text{BERT}_{\text{LARGE}}$	96.6	92.8
$\text{BERT}_{\text{BASE}}$	96.4	92.4
Feature-based approach ($\text{BERT}_{\text{BASE}}$)		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

BERT can be used as Feature based Approach.
Using last four hidden sum for features is best performing.

6. Conclusion

- Unsupervised pre-training을 통한 language model transfer learning은 높은 성능향상과 작은 데이터로 fine-tuning을 진행할 수 있도록 해준다.

- Masked language model을 이용한 bidirection architecture을 제안, sentence context understanding을 위한 next sentence prediction Pre-training을 제안

참고문헌

- http://jalammar.github.io/illustrated-bert/
- https://arxiv.org/pdf/1810.04805.pdf