

BERT:

Pre-training of Deep Bidirectional
Transformers for Language Understanding

BERT

- **Bidirectional Encoder Representations from Transformers**
- Language Representation Model
- arXiv Preprint (Google AI Language)
- 대부분의 NLP Task의 SOTA를 갈아치움

Pre-training for Transfer Learning

- 최근 자연어처리에서 가장 큰 화두는 Pre-training !!
- 비전(Vision)에서는 이미 오래 전부터 ImageNet 데이터로 미리 잘 학습된 Pre-trained 네트워크를 이용해서 weight를 초기화하고 Transfer Learning하는 것이 너무나도 당연한 일이 되어버림
- 이제 자연어처리에서도 잘 학습된 네트워크를 이용해 weight를 초기화하고 Specific Task에 대해서 Transfer Learning하는 방향으로 연구가 활발함

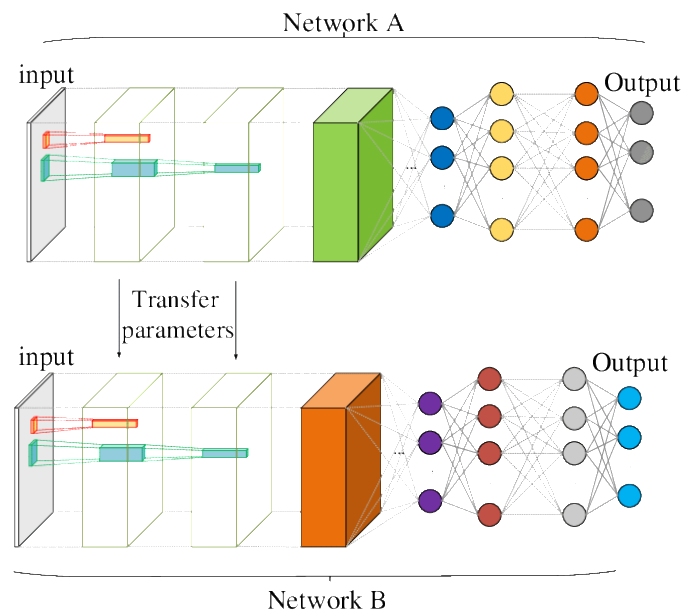
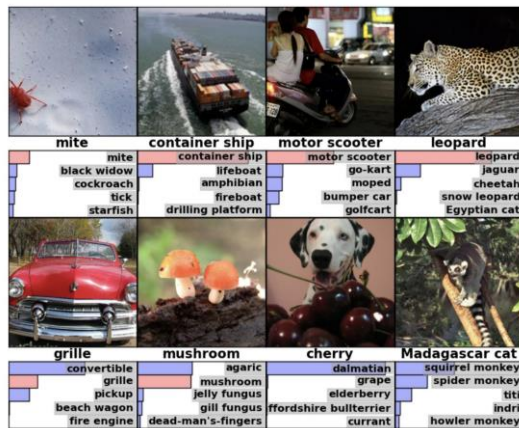
Transfer Learning

- Transfer Learning이란, 기존에 잘 학습된 모델을 사용하여 새로운 모델을 만들어서 학습을 빠르게 하고 예측 성능을 높이는 방법
- ImageNet과 같은 대용량 데이터를 이용해서 VGGNet, GoogLeNet, ResNet 등 엄청난 GPU와 시간을 들여 잘 학습시켜놓은 모델의 weight를 그대로 들고와서 초기값으로 사용하는 것
- 초기화를 한 뒤에는 자신의 테스트(classification, detection, tracking 등)에 맞춰서 Fine-tuning하는 방식

ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.



Transformer-based Language Model for Pre-training

- Open AI에서 2018년 여름에 **Transformer** 기반의 **Language Model (LM)**을 이용해서 Pre-training을 하고, 학습된 weight를 이용해서 Specific Task에 대해서 Transfer Learning하는 방법을 제안함
- 설명에 앞서 Transformer 리뷰
 - "Attention is All You Need", Vaswani et al. (Google), NIPS 2017
 - CNN, RNN 없이 Attention만으로 네트워크를 쌓아올린 모델
 - NMT를 목적으로한 Encoder-Decoder 구조
 - Decoder = Language Model

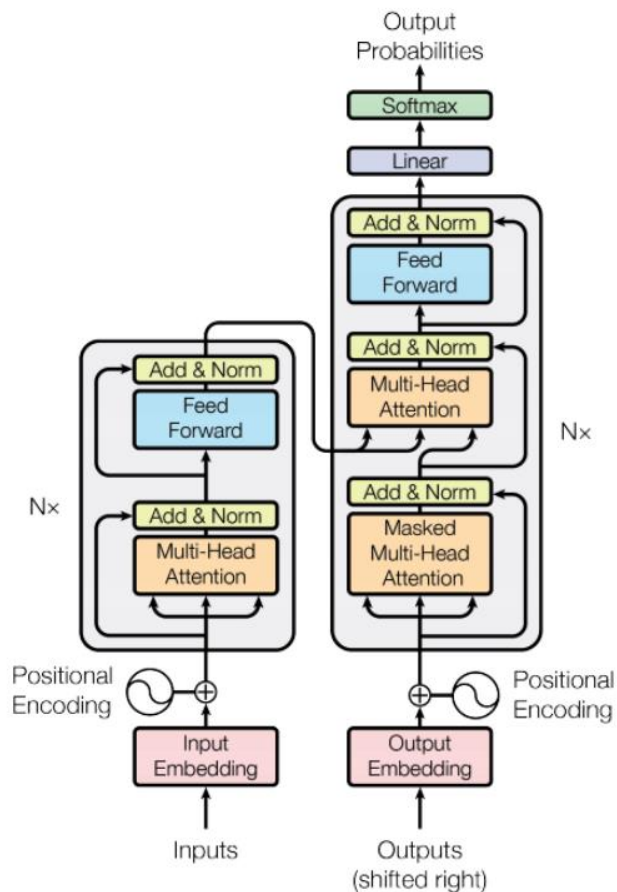


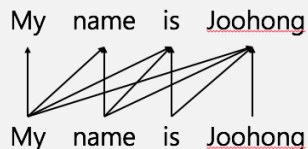
Figure 1: The Transformer - model architecture.

Transformer Review

- Multi-Head Attention이 핵심
- Query, Key, Value를 입력 받고 Q, K의 유사도 만큼 V를 가중합
- 특히 Decoder에는 Masked Multi-Head Attention이 사용되는데, 이는 나중 단어를 고려하지 않는 Language Model의 특성을 살리기 위한 것

- Masked Self Attention

My name is Joohong



- Auto regressive 하게

앞의 단어들에 대해서만 Attention 하겠다!

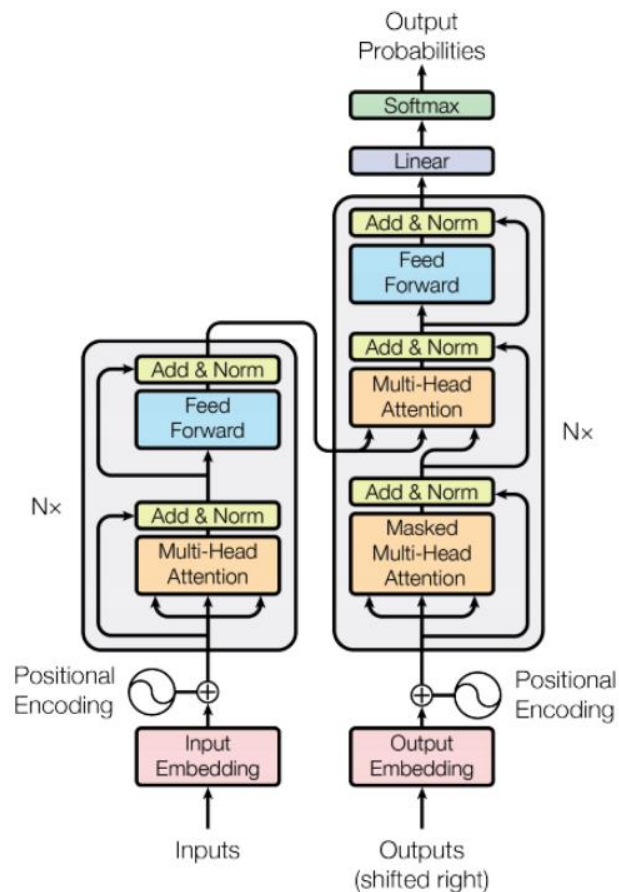
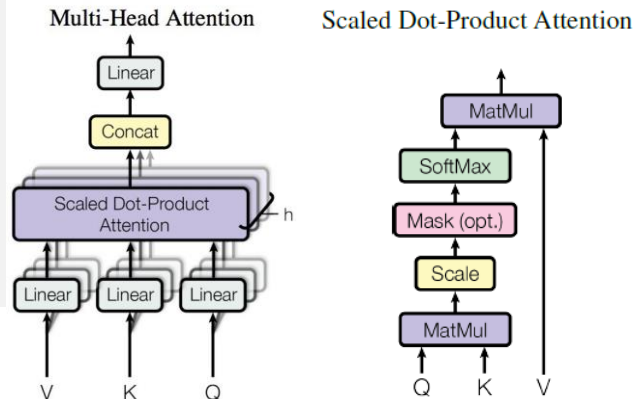


Figure 1: The Transformer - model architecture.

Transformer-based Language Model for Pre-training

- Open AI에서 2018년 여름에 발표한 **GPT** (Generative Pre-Training; **Transformer** 기반의 **Language Model (LM)**)을 이용해서 Pre-training을 하고, 학습된 weight를 이용해서 Specific Task에 대해서 Transfer Learning하는 방법을 제안함
- LM과 같은 Transformer의 Decoder만을 사용해서 Wiki 같은 큰 코퍼스를 넣어서 LM 학습을 진행
- 잘 학습된 (=언어를 잘 이해하고 있는) LM의 구조 및 weight를 그대로 가져다가 사용하여 각 테스트에 대한 Transfer Learning을 진행

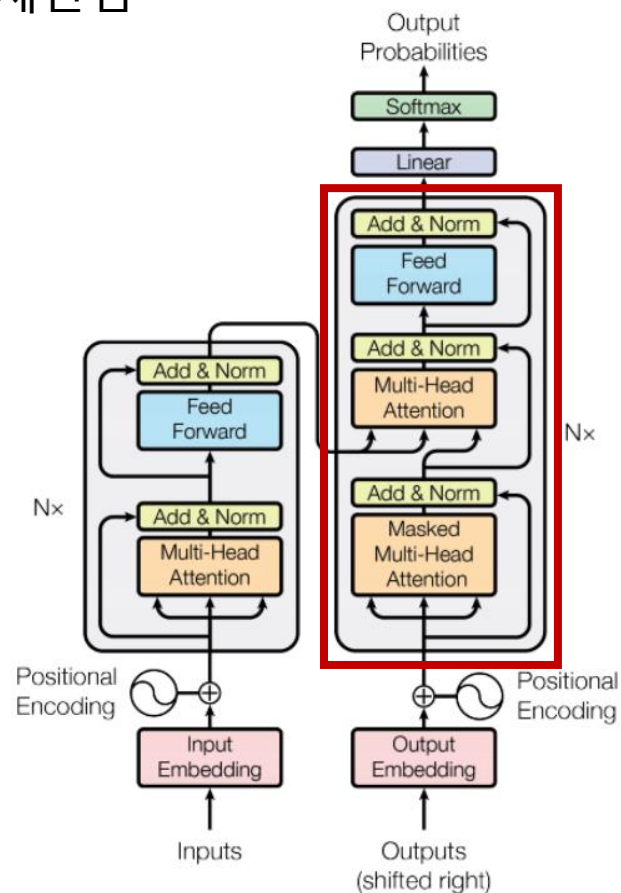


Figure 1: The Transformer - model architecture.

Transformer-based Language Model for Pre-training

- 왼쪽과 같이 Transformer의 Decoder로 LM 학습을 하고
- 오른쪽과 같이 다양한 NLP 테스트에 대해서 Transfer Learning
- Input의 생김새가 조금씩 다름

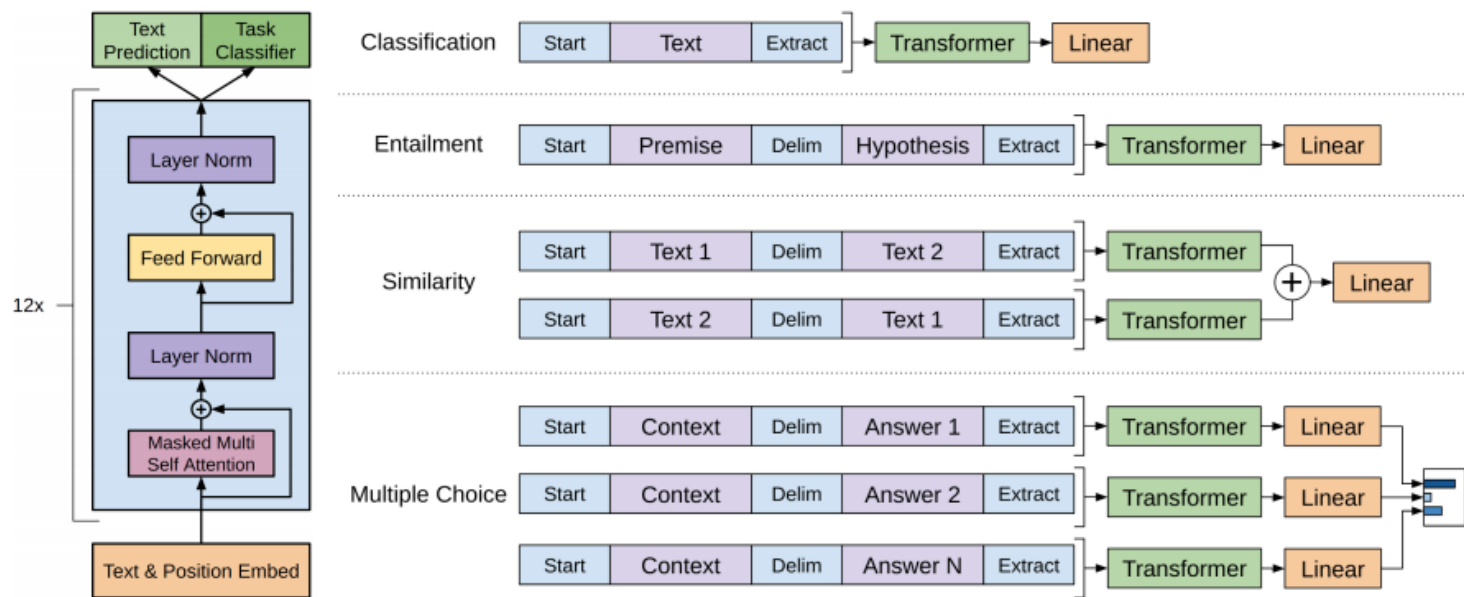


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

BERT vs Open AI GPT

- 전반적인 모델의 구조와 매커니즘은 거의 유사

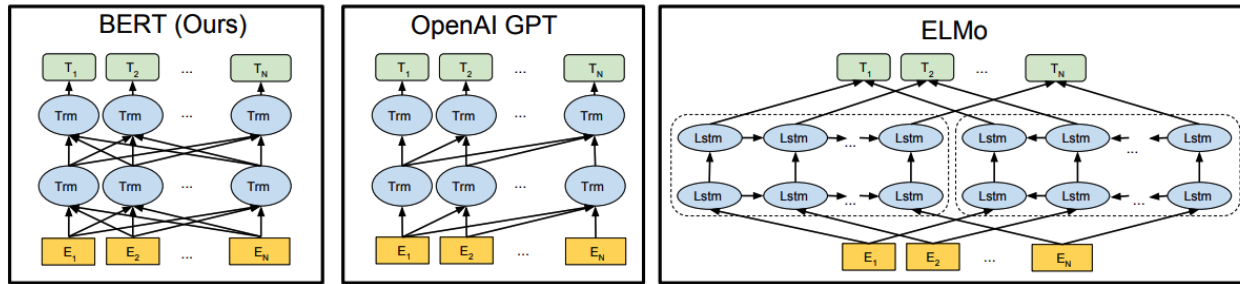
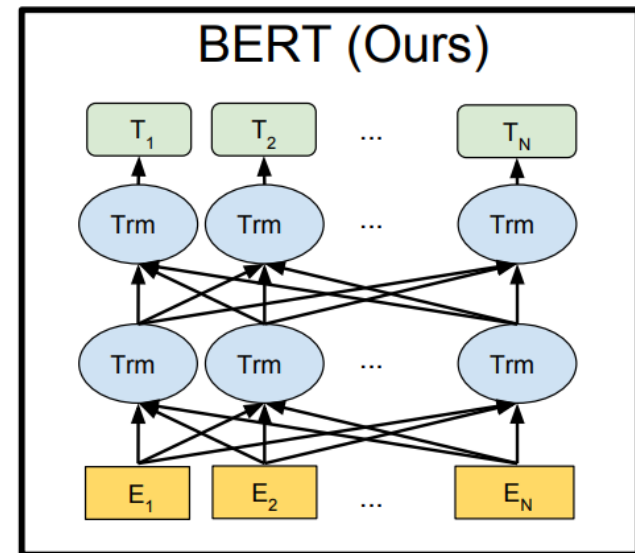


Figure 1: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM to generate features for downstream tasks. Among three, only BERT representations are jointly conditioned on both left and right context in all layers.

- Open AI GPT에 대한 이슈로는 아래와 같은 것들이 있는데,
 - Input을 어떻게 넣을 것인가?
 - Pre-training을 단순 LM 학습처럼 하는게 효과적인가?
 - 여러가지 테스트에 대한 Transfer Learning을 어떻게 진행할 것인가?
- 이런 이슈에 대해서 BERT는 새로운 방법을 제안하였고,
다양한 NLP 테스트에 대해서 기존 성능을 훌쩍 뛰어넘는 결과를 보임

BERT - Large Scale Model

- 상당히 거대한 모델
 - **BERT_{BASE}**: L=12, H=768, A=12, Total Parameters=110M
 - **BERT_{LARGE}**: L=24, H=1024, A=16, Total Parameters=340M
- Pre-Training Configuration
 - 4 Cloud TPUs (16 TPU chips) for BASE
 - 16 Cloud TPUs (64 TPU chips) for LARGE
 - 40 Epoch & 256 mini-batch
 - 256 sequence * 512 tokens = 128,000 tokens/batch
 - 3.3 billion word corpus
 - 4 days to training



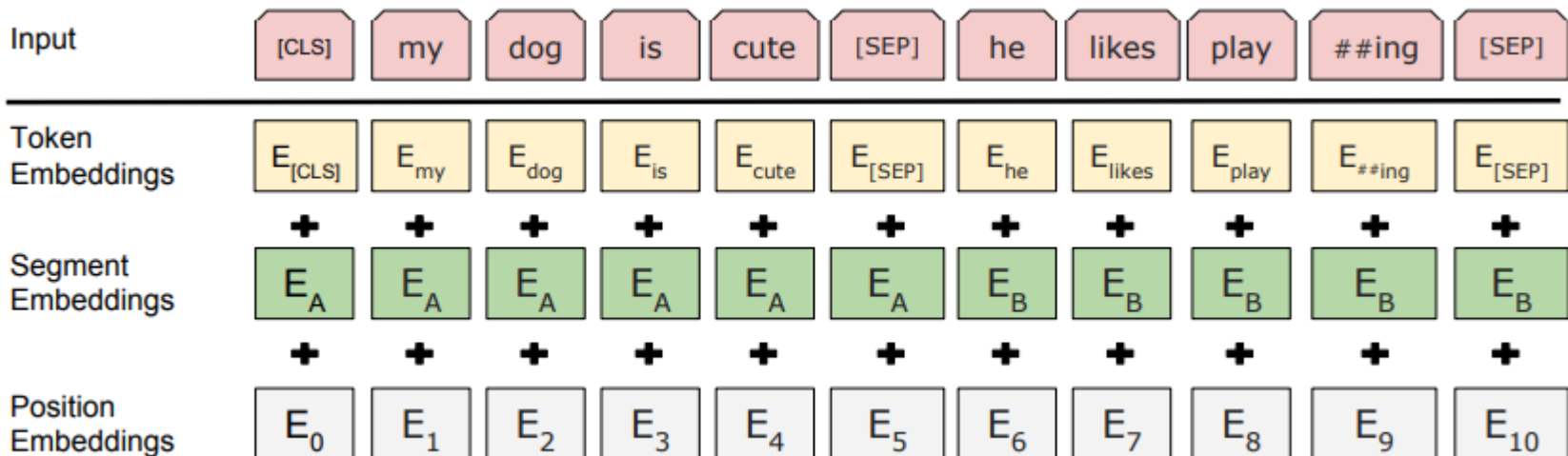
Input Representation

- Input token sentence의 형태

- Single sentence or a pair of sentences(e.g., [Question, Answer])

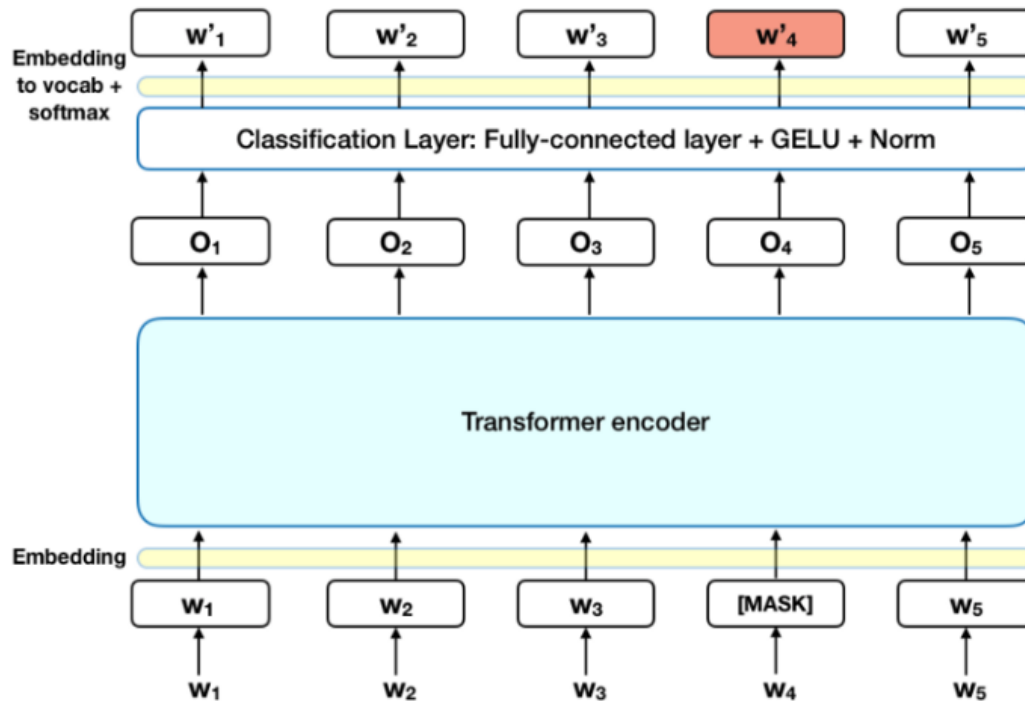
- Input representation의 형태

- Token embedding+ segment embedding + position embedding
- Token embedding : WordPiece embedding (30,000 token vocab)
- position embedding : up to 512 tokens(=max_length = 512)
- First token은 항상 [CLS]로 시작, non-classification task에서는 무시됨
- 문장의 끝에 [SEP] Token이 삽입되고, pair sentence의 경우 [SEP]토큰을 포함한 앞 문장은 E_A 를, segment embedding으로 사용하고 뒷 문장을 E_B 로 사용
- Single sentence의 경우 모든 segment embedding을 E_A 로 사용



Pre-training tasks #1: Masked LM

- Input token의 일부를 랜덤하게 Masking하고, Masked Token이 무엇인지를 찾는 방법으로 LM을 학습시키자! (= Cloze task)
 - 기존의 LM과 동일하게 진행되지만 Masking되지 않은 token은 loss 계산에 참여하지 않음



Pre-training tasks #1: Masked LM

- Input token의 일부를 랜덤하게 Masking하고, Masked Token이 무엇인지를 찾는 방법으로 LM을 학습시키자! (= Cloze task)
 - 기존의 LM과 동일하게 진행되지만 Masking되지 않은 token은 loss 계산에 참여하지 않음
- output layer의 범위는 Vocabulary가 되고 softmax function을 통해 정답을 예측하자
- 발생할 수 있는 2가지 단점!
 - pre-training과 fine-tuning간의 간극 발생 (실제로 특정 semantic token의 입력이 들어오는 fine-tuning과 [MASK] token을 이용하는 pre-training과의 간극)
 - 각 Batch의 15%만 학습이 진행
- pre-training과 fine-tuning간의 간극을 최소화하기 위해 다음과 같은 룰에 따라 Masking한다.
 - Input sequence의 15%는 다음 3가지 방법에 의해 replace된다. replace될 token의
 - 1) 80%는 [MASK] token으로 replace된다.
 - 2) 10%는 random word로 replace된다.
 - 3) 10%는 기존 단어로 replace된다. 이때 기존 단어를 predict를 해야 한다.
- Converge하는데 오래걸리는 단점이 있지만 absolute accuracy가 더 높으니까 괜찮!

Pre-training tasks #2: Next Sentence Prediction

- Pair Sentence에 대해서 앞 문장과 뒤 문장의 관계가 IsNext인지 NotNext인지 판별하는 task를 넣고 loss에 추가하자!
- Why?
 - QNA NLI에서 필요한 sentence 간의 relation을 학습
- 실제 데이터셋에서 50%는 pair로 묶고, 50%는 다른 Corpus의 random한 sentence랑 묶자
- 실제 fine-tuning에서는 single-sentence를 허용하지만 pre-training에서는 pair sentence만 사용

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

Fine-tuning

- Batch size : 16(BASE), 32(LARGE)
- Number of epochs : 3(BASE), 4(LARGE)
- Time to Complete Fine-tuning : 30 minute !!

GLUE Benchmark(General Language Understanding Evaluation)

-MNLI

Crowdsourced entailment classification

(input – 두 문장, 문장 간의 관계를 predict(entailment, contradiction, neutral))

-QQP

Quora Question Pairs

(input – 두 질문, 두 질문이 semantically equivalent한가를 predict)

-QNLI

Question Natural Language Inference

(input – (질문, 문장), Stanford QA dataset의 binary classification 버전, pair의 관계가 올바른가를 predict)

-SST-2

Stanford Sentiment Treebank

(input – 한 문장, Movie Review에 대한 binary classification)

-CoLA

Corpus of Linguistic Acceptability

(input – 한 문장, 영어 문장이 linguistically acceptable한지 predict)

-STS-B

Semantic Textual Similarity Benchmark

(input – 문장 pair, 두 문장이 얼마나 similar한지 1~5)

-MRPC

Microsoft Research Paraphrase Corpus

(input – 문장 pair, 온라인 뉴스에서 추출한 pair가 semantically equivalent한지 predict)

-RTE

Recognizing Textual entailment

(MNLI랑 비슷한데 data가 더 적음)

-WNLI

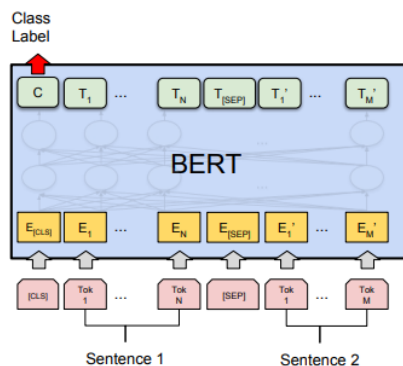
Winograd NLI

(small natural language inference,

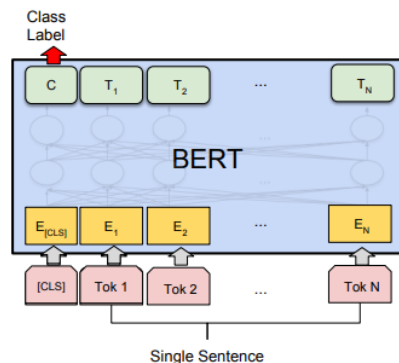
Dataset에 문제가 있어서 성능이 항상 65.1 이하 -> 제외)

GLUE는 TEST LABEL을 포함하지 않음!

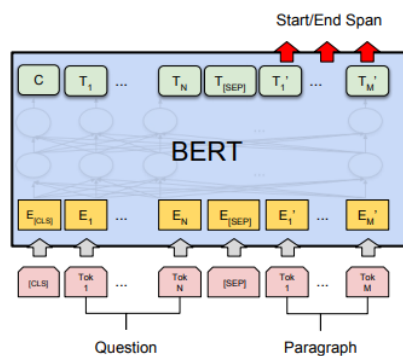
(website에 업로드하면 채점)



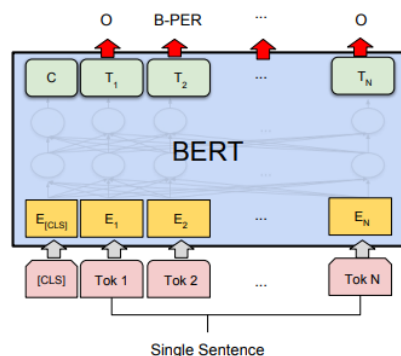
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, CoLA



(c) Question Answering Tasks:
SQuAD v1.1



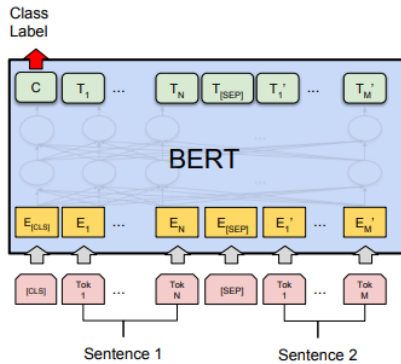
(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

| System | MNLI-(m/mm) | QQP | QNLI | SST-2 | CoLA | STS-B | MRPC | RTE | Average |
|-----------------------|------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | 392k | 363k | 108k | 67k | 8.5k | 5.7k | 3.5k | 2.5k | - |
| 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 | 79.9 | 90.4 | 36.0 | 73.3 | 84.9 | 56.8 | 71.0 |
| OpenAI GPT | 82.1/81.4 | 70.3 | 88.1 | 91.3 | 45.4 | 80.0 | 82.3 | 56.0 | 75.2 |
| BERT _{BASE} | 84.6/83.4 | 71.2 | 90.1 | 93.5 | 52.1 | 85.8 | 88.9 | 66.4 | 79.6 |
| BERT _{LARGE} | 86.7/85.9 | 72.1 | 91.1 | 94.9 | 60.5 | 86.5 | 89.3 | 70.1 | 81.9 |

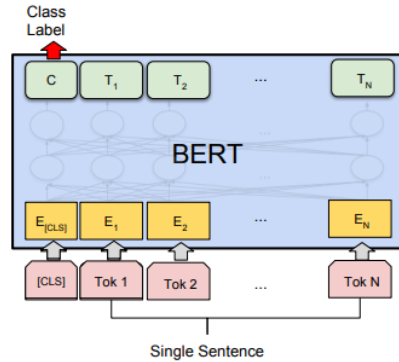
Table 1: GLUE Test results, scored by the GLUE evaluation server. The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set. OpenAI GPT = (L=12, H=768, A=12); BERT_{BASE} = (L=12, H=768, A=12); BERT_{LARGE} = (L=24, H=1024, A=16). BERT and OpenAI GPT are single-model, single task. All results obtained from <https://gluebenchmark.com/leaderboard> and <https://blog.openai.com/language-unsupervised/>.

<<처럼 data에 따라 다르게 두고 fine-tuning함

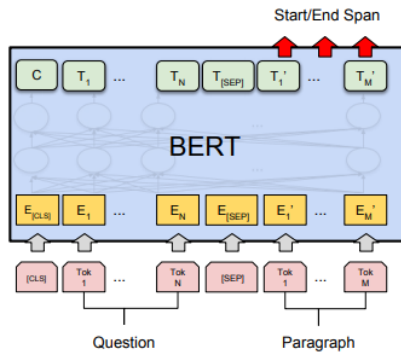
(BERT(Large)는 dataset이 작으면 finetuning이 가끔 불안정해서 몇번 돌려보고 젤좋은거 하나고름)



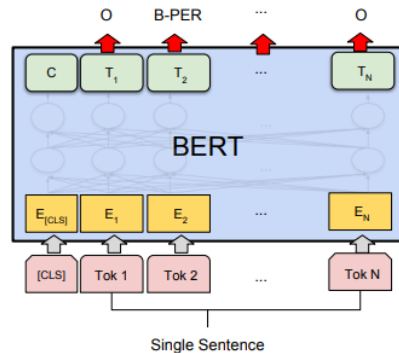
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, CoLA



(c) Question Answering Tasks:
SQuAD v1.1



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

SQuAD

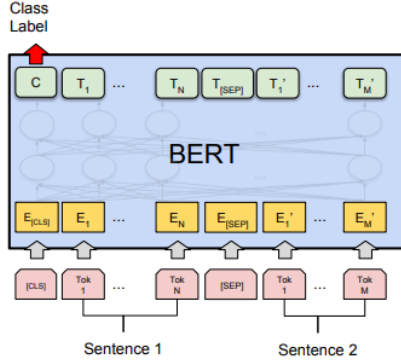
Stanford Question Answering Dataset
(Input – Question / 답이 포함된 Paragraph(Wiki)
Paragraph 내의 answer text span을 predict)

Parameter 추가

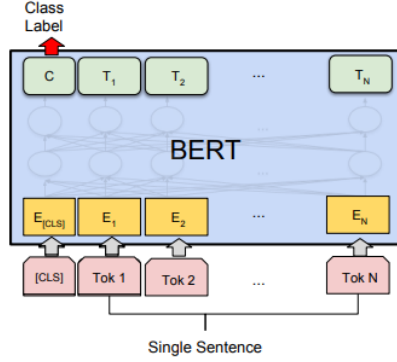
-Start vector, End vector from final hidden token

$T_i \rightarrow i$ 번째 input token

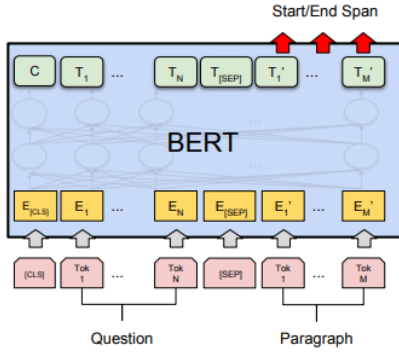
T_i 와 Start ~ End span 사이를 전부 softmax -> 가장 높은 scored span을 사용한다



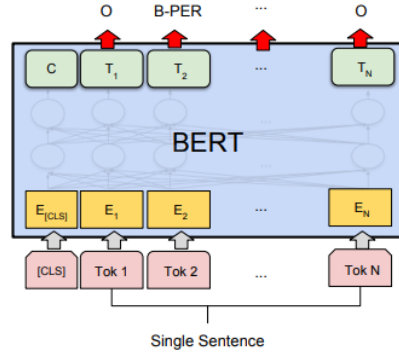
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, CoLA



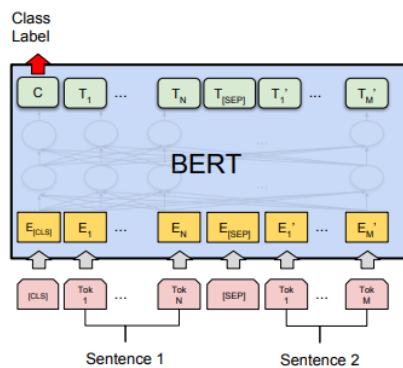
(c) Question Answering Tasks:
SQuAD v1.1



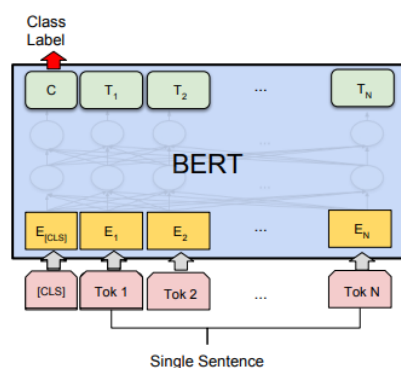
(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

| System | Dev | | Test | |
|---------------------------------------|-------------|-------------|-------------|-------------|
| | EM | F1 | EM | F1 |
| Leaderboard (Oct 8th, 2018) | | | | |
| Human | - | - | 82.3 | 91.2 |
| #1 Ensemble - nlnet | - | - | 86.0 | 91.7 |
| #2 Ensemble - QANet | - | - | 84.5 | 90.5 |
| #1 Single - nlnet | - | - | 83.5 | 90.1 |
| #2 Single - QANet | - | - | 82.5 | 89.3 |
| Published | | | | |
| BiDAF+ELMo (Single) | - | 85.8 | - | - |
| R.M. Reader (Single) | 78.9 | 86.3 | 79.5 | 86.6 |
| R.M. Reader (Ensemble) | 81.2 | 87.9 | 82.3 | 88.5 |
| Ours | | | | |
| BERT _{BASE} (Single) | 80.8 | 88.5 | - | - |
| BERT _{LARGE} (Single) | 84.1 | 90.9 | - | - |
| BERT _{LARGE} (Ensemble) | 85.8 | 91.8 | - | - |
| BERT _{LARGE} (Sgl.+TriviaQA) | 84.2 | 91.1 | 85.1 | 91.8 |
| BERT _{LARGE} (Ens.+TriviaQA) | 86.2 | 92.2 | 87.4 | 93.2 |

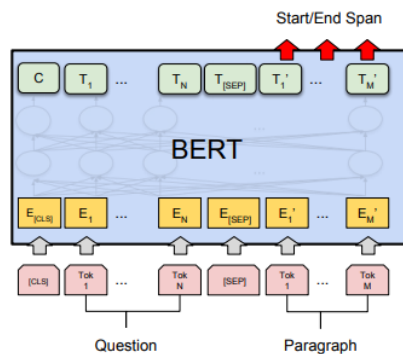
Table 2: SQuAD results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.



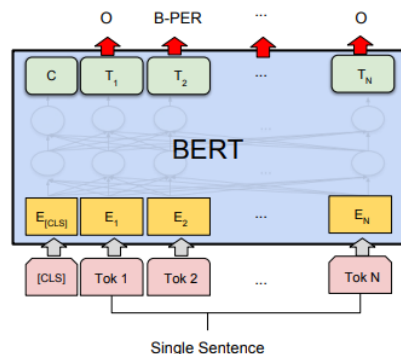
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, CoLA



(c) Question Answering Tasks:
SQuAD v1.1



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

Named Entity Recognition

-CoNLL 2003

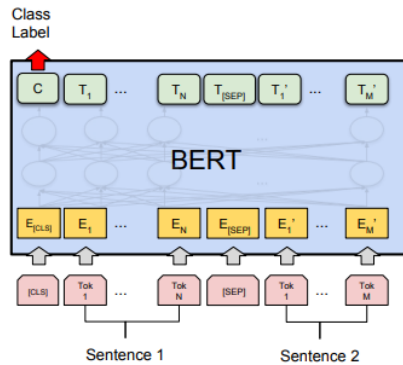
NER Dataset

200k training words(Person, Organization, Location, Miscellaneous, Other)

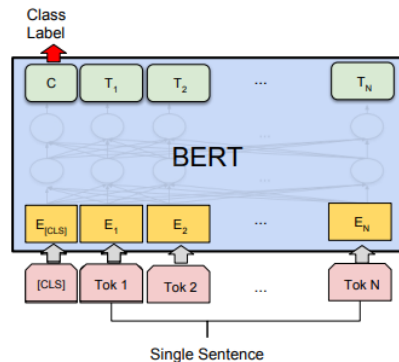
Jim Hen ##son was a puppet ##eer
I-PER I-PER X O O O X

WordPiece tokenizer로 문장을 tokenizing

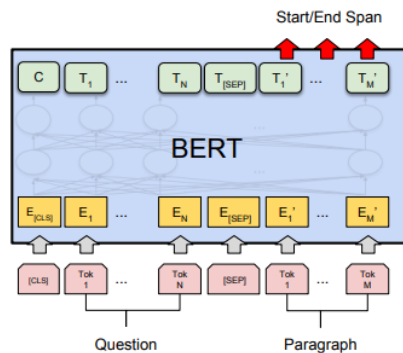
| System | Dev F1 | Test F1 |
|--------------------------------|-------------|-------------|
| ELMo+BiLSTM+CRF | 95.7 | 92.2 |
| CVT+Multi (Clark et al., 2018) | - | 92.6 |
| BERT _{BASE} | 96.4 | 92.4 |
| BERT _{LARGE} | 96.6 | 92.8 |



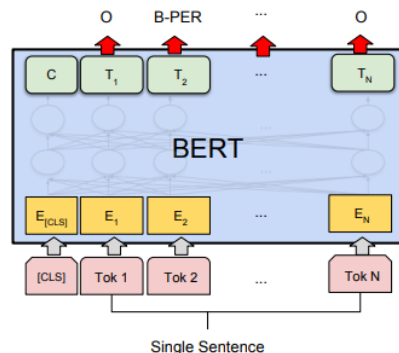
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, CoLA



(c) Question Answering Tasks:
SQuAD v1.1



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

A girl is going across a set of monkey bars. She
 (i) jumps up across the monkey bars.
 (ii) struggles onto the bars to grab her head.
 (iii) gets to the end and stands on a wooden plank.
 (iv) jumps up and does a back flip.

SWAG

The Situations With Adversarial Generations
 Sentence from a video captioning dataset

| System | Dev | Test |
|------------------------------------|-------------|-------------|
| ESIM+GloVe | 51.9 | 52.7 |
| ESIM+ELMo | 59.1 | 59.2 |
| BERT _{BASE} | 81.6 | - |
| BERT _{LARGE} | 86.6 | 86.3 |
| Human (expert) [†] | - | 85.0 |
| Human (5 annotations) [†] | - | 88.0 |

Ablation Studies

1. Effect of Pretraining Tasks

- bidirectionality가 유용한가를 증명하는 부분

No NSP : Masked LM을 사용해 training, Next Sentence Prediction은 빼고

LTR & No NSP : Left to Right만 사용, Masking X (OpenAI GPT와 comparable, but dataset은 더 큼)

(SQuAD : token level hidden states have no right-side context -> 성능 떡락)

MRPC : unclear, but full hyperparameter sweep with many random restarts)

| Tasks | Dev Set | | | | |
|----------------------|-----------------|---------------|---------------|----------------|---------------|
| | MNLI-m (Acc) | QNLI (Acc) | MRPC (Acc) | SST-2 (Acc) | SQuAD (F1) |
| BERT _{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 |

Ablation Studies

2. Effect of Model Size

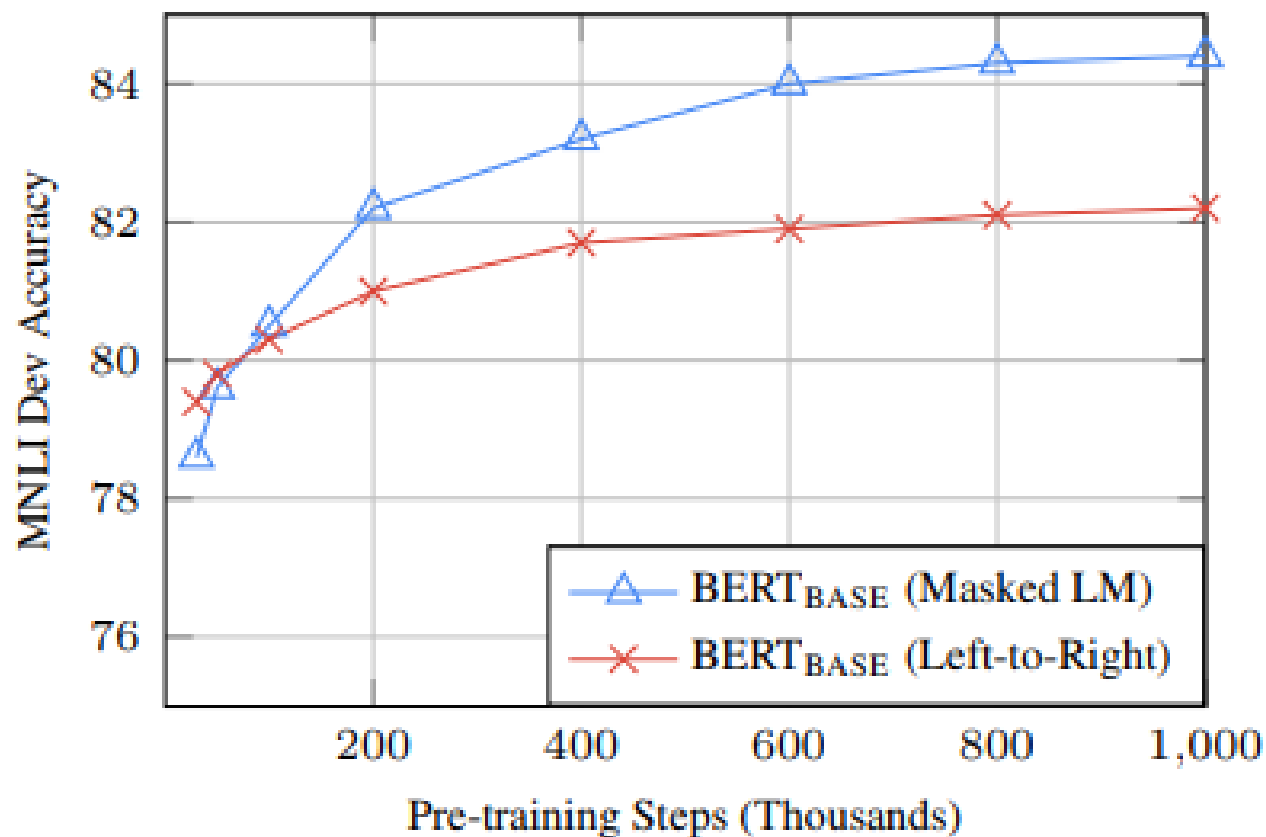
- Model Size가 fine-tuning에 미치는 영향을 증명하는 부분
(Number of layers, Hidden units, Attention heads, 나머지는 똑같이)

| Hyperparams | | | | Dev Set Accuracy | | |
|-------------|------|----|----------|------------------|------|-------|
| #L | #H | #A | LM (ppl) | MNLI-m | MRPC | SST-2 |
| 3 | 768 | 12 | 5.84 | 77.9 | 79.8 | 88.4 |
| 6 | 768 | 3 | 5.24 | 80.6 | 82.2 | 90.7 |
| 6 | 768 | 12 | 4.68 | 81.9 | 84.8 | 91.3 |
| 12 | 768 | 12 | 3.99 | 84.4 | 86.7 | 92.9 |
| 12 | 1024 | 16 | 3.54 | 85.7 | 86.9 | 93.3 |
| 24 | 1024 | 16 | 3.23 | 86.6 | 87.8 | 93.7 |

Ablation Studies

3. Effect of Number of Training Steps

- BERT가 fine-tuning accuracy를 위해서 엄청난 수의 pre-training을 해야 함(128,000 words/batch * 1,000,000 steps)
- LTR보다 수렴 속도가 더 느리지만 이미 절대적인 accuracy가 outperform



Ablation Studies

4. Feature-based Approach with BERT

- Fine Tuning approach가 아닌 feature-based approach에서도 잘 동작하는가를 증명하는 부분

: NER 실험에서 activation layer를 빼와서 fine-tuning 없이 사용해보자!

(contextual embedding = random initialized 2-layer BiLSTM before classification layer -> ELMo-like)

| System | Dev F1 | Test F1 |
|--------------------------------|--------|---------|
| ELMo+BiLSTM+CRF | 95.7 | 92.2 |
| CVT+Multi (Clark et al., 2018) | - | 92.6 |
| BERT _{BASE} | 96.4 | 92.4 |
| BERT _{LARGE} | 96.6 | 92.8 |

| Layers | Dev F1 |
|--------------------------|--------|
| Finetune All | 96.4 |
| First Layer (Embeddings) | 91.0 |
| Second-to-Last Hidden | 95.6 |
| Last Hidden | 94.9 |
| Sum Last Four Hidden | 95.9 |
| Concat Last Four Hidden | 96.1 |
| Sum All 12 Layers | 95.5 |

Conclusion

- Demonstrate empirical improvement due to transfer learning
- Contrast to OpenAI GPT, very deep bidirectional architectures
- Powerful performance surpassing human
- (사족) 다만, 저걸 학습시킬 수 있는 하드웨어 자원이 과연 일반 연구실이나 개인에게 있는가를 생각해보면 아쉬울 따름