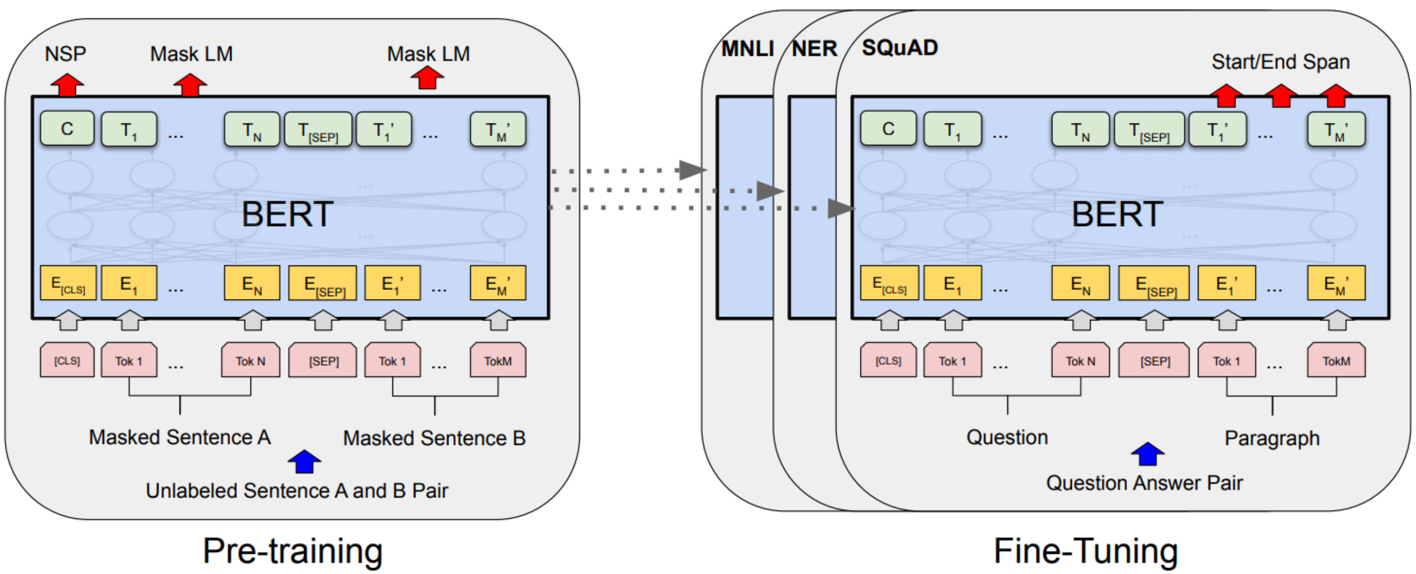
Lecture 8-5: BERT

Pilsung Kang
School of Industrial Management Engineering
Korea University

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- BERT
  - ✓ Designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers
    - Masked language model (MLM): bidirectional pre-training for language representations
    - Next sentence prediction (NSP)

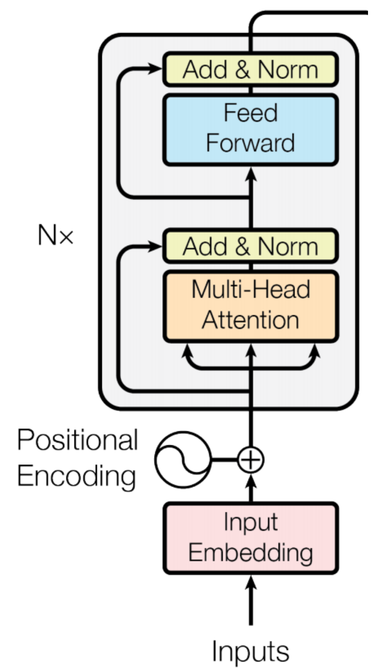


    - Pre-trained BERT model can be fine-tunes with just one additional output layer to create SOTA models for a wide range of NLP tasks (QA, NER, Sentiment Analysis, etc.)

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- BERT: Model Architecture
  - ✓ Multi-layer bidirectional Transformer encoder
    - L: number of layers (Transformer block)
    - H: hidden size
    - A: number of self attention heads
  - ✓ $\text{BERT}_{\text{BASE}}$
    - L = 12, H=768, A = 12
    - Total parameters = 110M
    - Same model size as OpenAI GPT
  - ✓ $\text{BERT}_{\text{LARGE}}$
    - L = 24, H=1,024, A = 16
    - Total parameters = 340M



BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- BERT: Input/Output Representations
  - ✓ To make BERT handle a variety of down-stream tasks, the input representation is able to unambiguously represent both a single sentence and a pair of sentences (ex: Question-Answer)
    - **Sentence**: an arbitrary span of contiguous text, rather than an actual linguistic sentence
    - **Sequence**: the input token sequences to BERT, which may be a single sentence or two sentences packed together

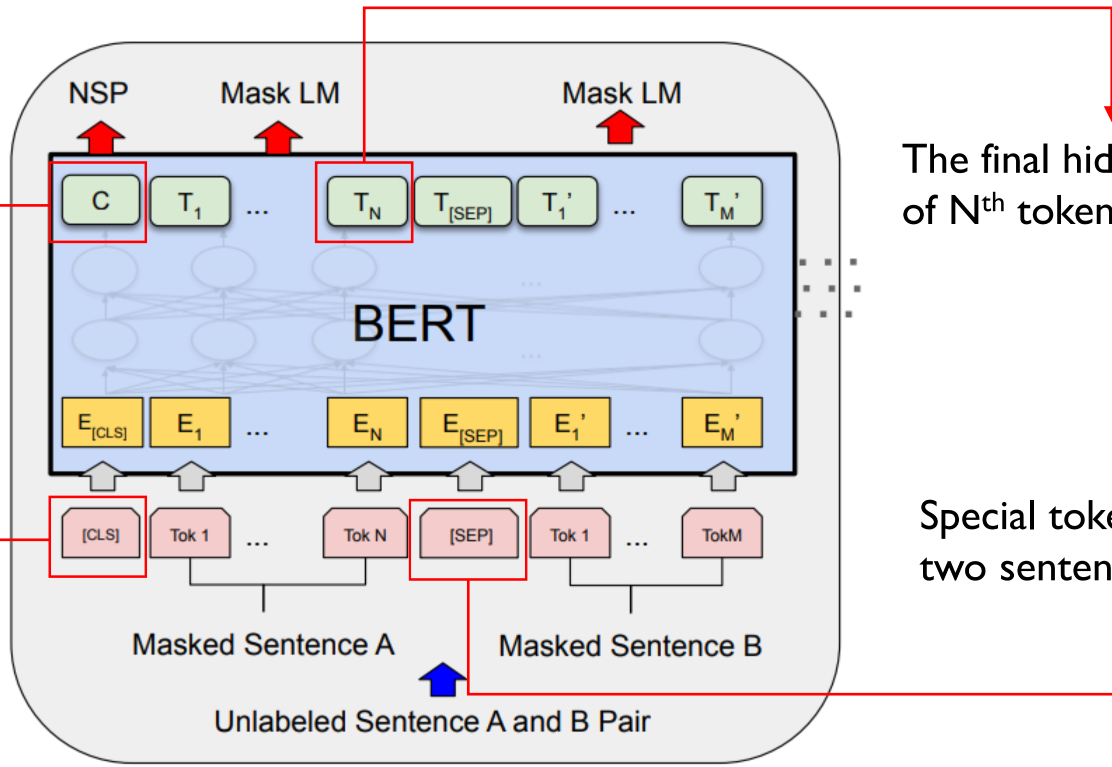
BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- BERT: Input/Output Representations

The final hidden vector of special [CLS] token

The first token of every sequence



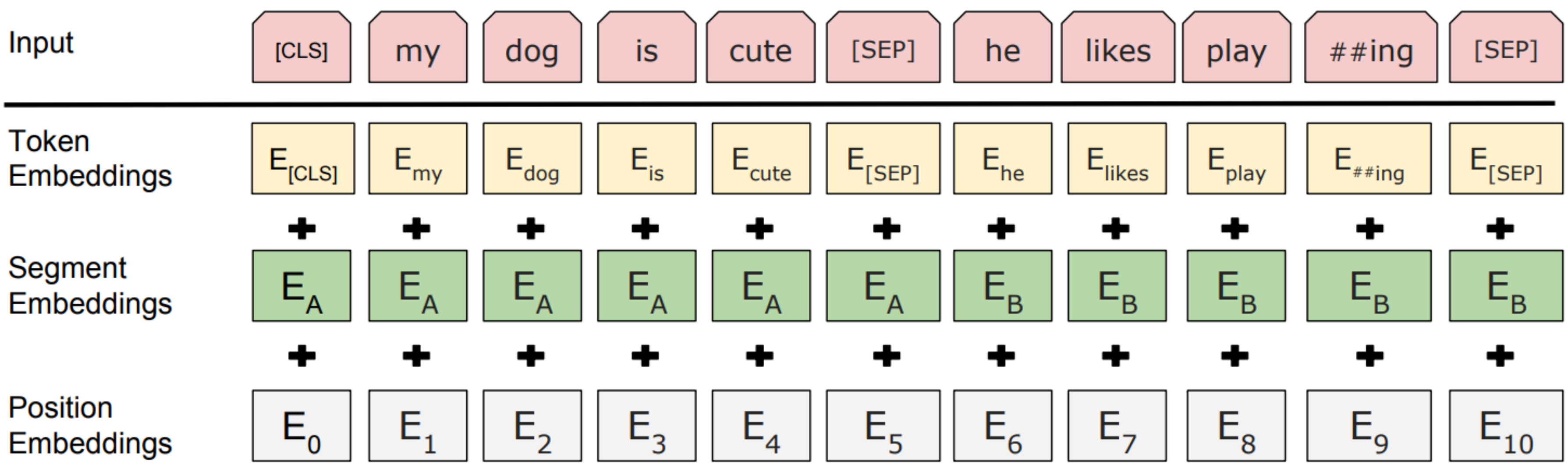
The final hidden vector of N^{th} token

Special token separating two sentences

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

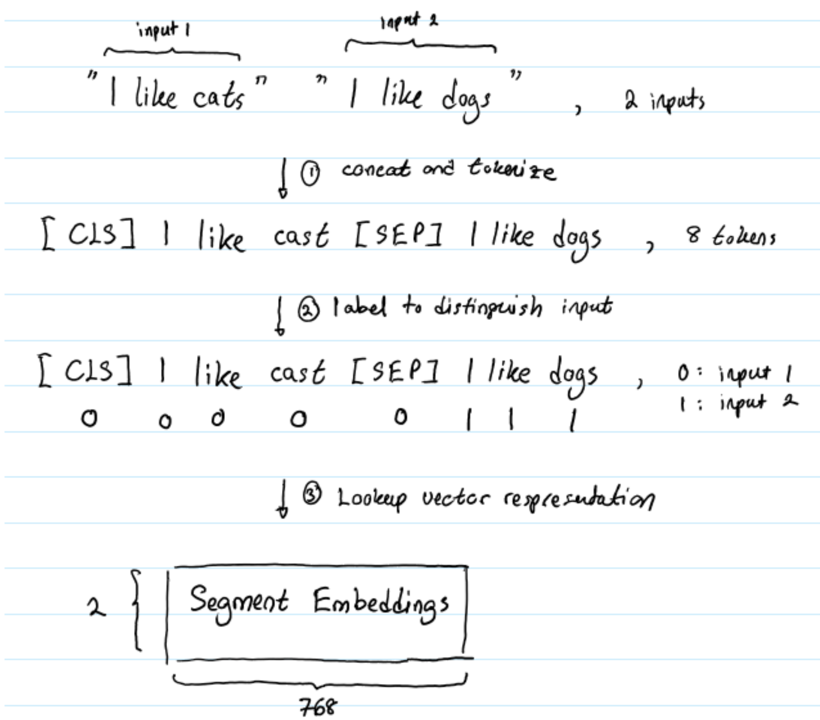
- BERT: Input/Output Representations
  - ✓ Input representation is the sum of
    - (1) Token embedding: WordPiece embeddings with a 30,000 token vocabulary
    - (2) Segment embedding
    - (3) Position embedding: same as in the Transformer



BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- BERT: Input/Output Representations
  - ✓ (2) Segment embedding



Layer-wise accounting:

Going through layers from top to bottom, we can see following:

1. Inputs — Token and segment do not have any trainable parameters, as expected.
2. Token embeddings parameters= 23040000 (H * T) — because each of 30k (T) tokens needs a representation in dimension 768 (H)
3. Segment Embeddings parameters = 1536 (2*H) because we need two vectors each of length (H). The vectors represent Segment A and Segment B respectively
4. Token embeddings and segment embeddings are added to Position Embedding. Parameters = 393216 (H*P). This is because it needs to generate P vectors, each of length H, for the tokens starting 1 to 512 (P). The position embeddings in BERT are trained and not fixed as in *Attention is all you need;* There's a dropout applied, and then Layer Normalization is done
5. Layer Normalization parameters = 1536 (2*H). Normalization has two parameters to learn — mean and standard deviation of each of the embedding position, hence 2*H
6. Encoder: MultiheadSelfAttention: MultiHeadAttention = 2362368

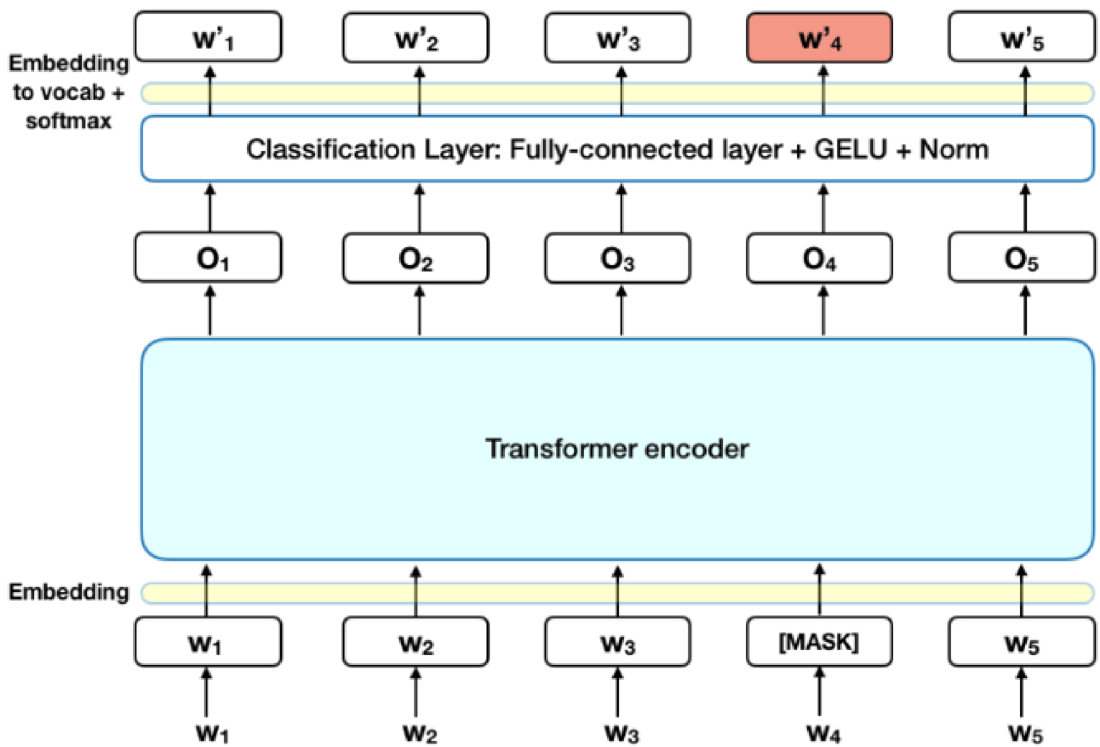
https://medium.com/@_init_/why-bert-has-3-embedding-layers-and-their-implementation-details-9c261108e28a

https://mc.ai/understanding-bert-architecture/

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 1: Masked Language Model (MLM)
    - 15% of each sequence are replaced with a [MASK] token
    - Predict the masked words rather tan reconstructing the entire input in denoising encoder



https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 1: Masked Language Model (MLM)
    - (Caution!) A mismatch occurs between pre-training and fine-tuning, since the [`MASK`] token does not appear during fine-tuning
    - (Solution) If the i-th token is chosen to be masked, it is replaced by the [`MASK`] token 80% of the time, a random toke 10% of the time, and unchanged 10% of the time
      - (80%) `my dog is hairy → my dog is [MASK]`
      - (10%) `my dog is hairy → my dog is apple`
      - (10%) `my dog is hairy → my dog is hairy`

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 1: Masked Language Model (MLM)
    - (Caution!) A mismatch occurs between pre-training and fine-tuning, since the [MASK] token does not appear during fine-tuning
    - (Solution) If the i-th token is chosen to be masked, it is replaced by the [MASK] token 80% of the time, a random toke 10% of the time, and unchanged 10% of the time

Masking Rates			Dev Set Results		
MASK	SAME	RND	MNLI Fine-tune	NER Fine-tune	NER Feature-based
80%	10%	10%	84.2	95.4	94.9
100%	0%	0%	84.3	94.9	94.0
80%	0%	20%	84.1	95.2	94.6
80%	20%	0%	84.4	95.2	94.7
0%	20%	80%	83.7	94.8	94.6
0%	0%	100%	83.6	94.9	94.6

BERT: Bidirectional Encoder Representations from Transformer

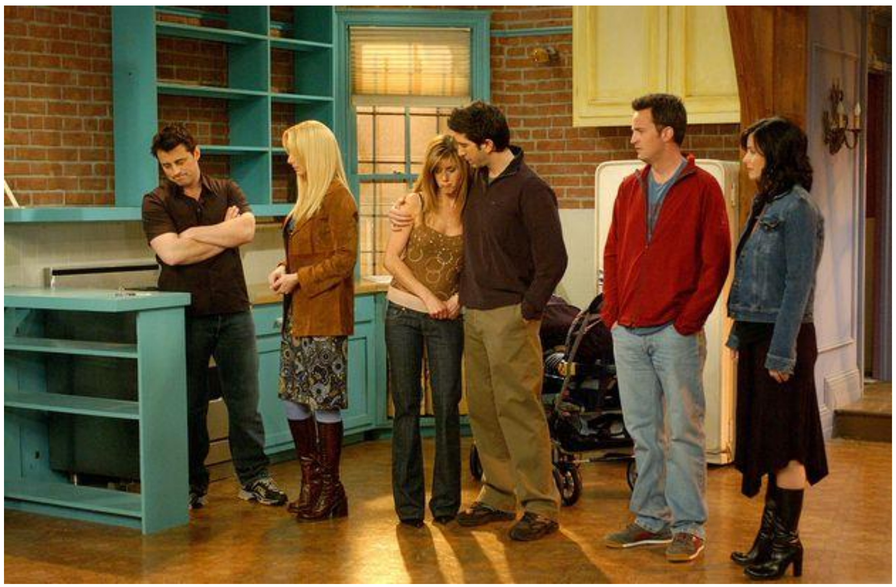
Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 2: Next Sentence Prediction (NSP)
    - Many important downstream tasks such as QA and NLI are based on understanding the relationship between two sentences, which is not directly captured by language modeling
    - A Binarized next sentence prediction task that can be trivially generated from any monolingual corpus is trained
      - 50% of the time B is the actual next sentence that follows A (IsNext)
      - 50% of the time it is a random sentence from the corpus (NotNext)
      - C is used for next sentence prediction
    - Despite its simplicity, pre-training towards this task is very beneficial both QA and NLI

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 2: Next Sentence Prediction (NSP)



Monica: This is harder than I thought it would be.

Chandler: Oh, it is gonna be okay.

Rachel: Do you guys have to go to the new house right away, or do you have some time?

Monica: We got some time.

Rachel: Okay, should we get some coffee?

Chandler: Sure. Where?

https://fangj.github.io/friends/season/1017-1018.html

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 2: Next Sentence Prediction (NSP)

Monica: This is harder than I thought it would be.

Chandler: Oh, it is gonna be okay.

Rachel: Do you guys have to go to the new house right away, or do you have some time?

Monica: We got some time.

Rachel: Okay, should we get some coffee?

Chandler: Sure. Where?

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Task 2: Next Sentence Prediction (NSP)

Monica: This is harder than I thought it would be.

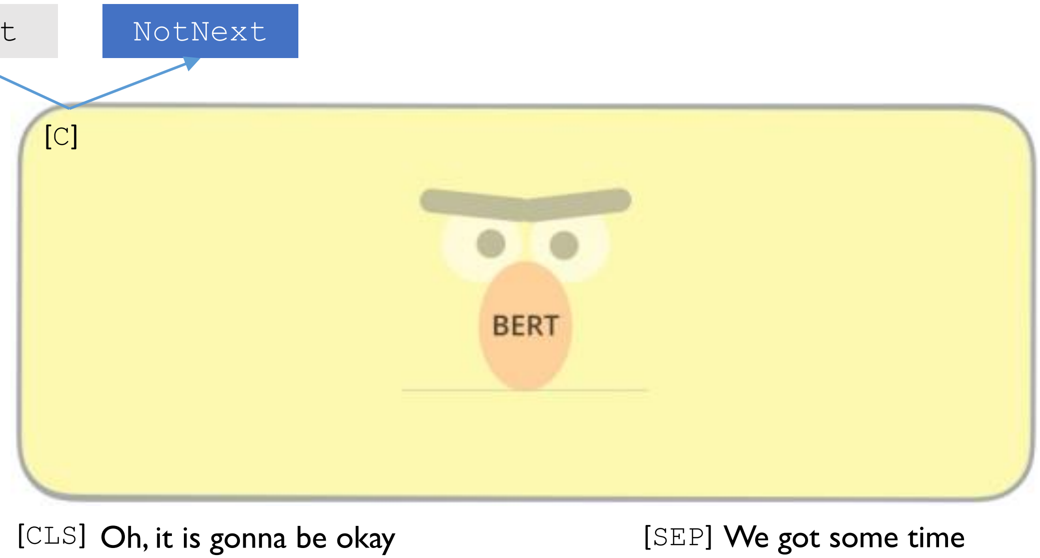
Chandler: Oh, it is gonna be okay.

Rachel: Do you guys have to go to the new house right away, or do you have some time?

Monica: We got some time.

Rachel: Okay, should we get some coffee?

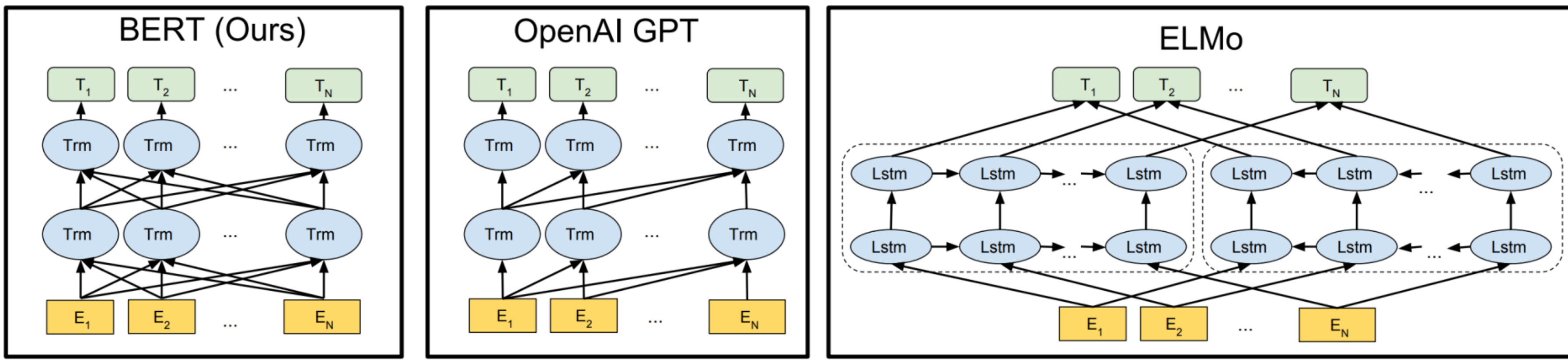
Chandler: Sure. Where?



BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

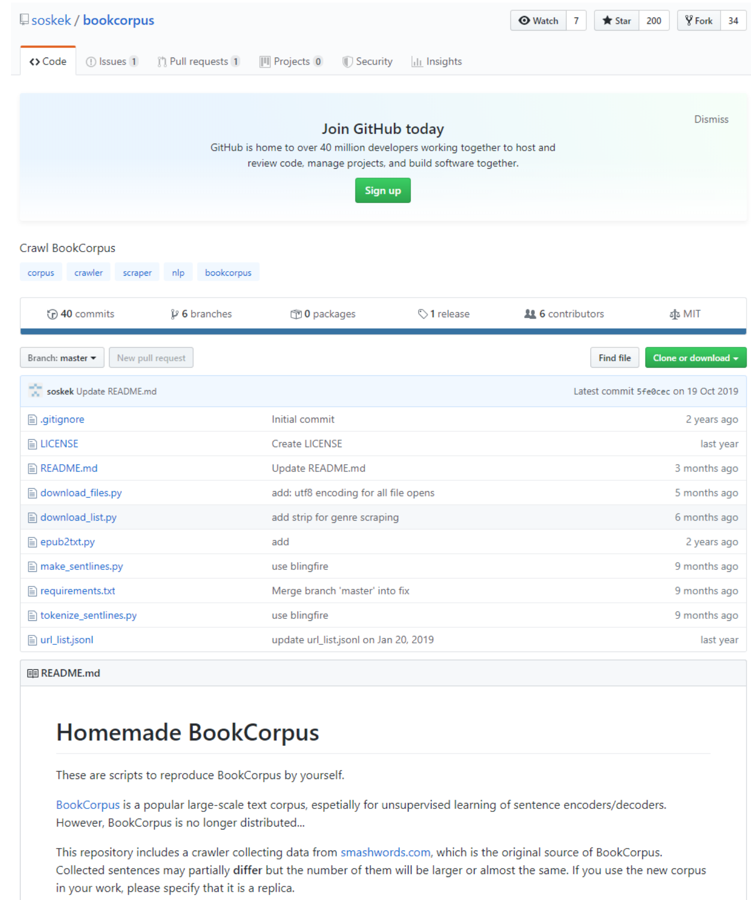
- Differences in pre-training model architectures



BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Datasets for pre-training
    - BooksCorpus (800M words) (Zhu et al., 2015)

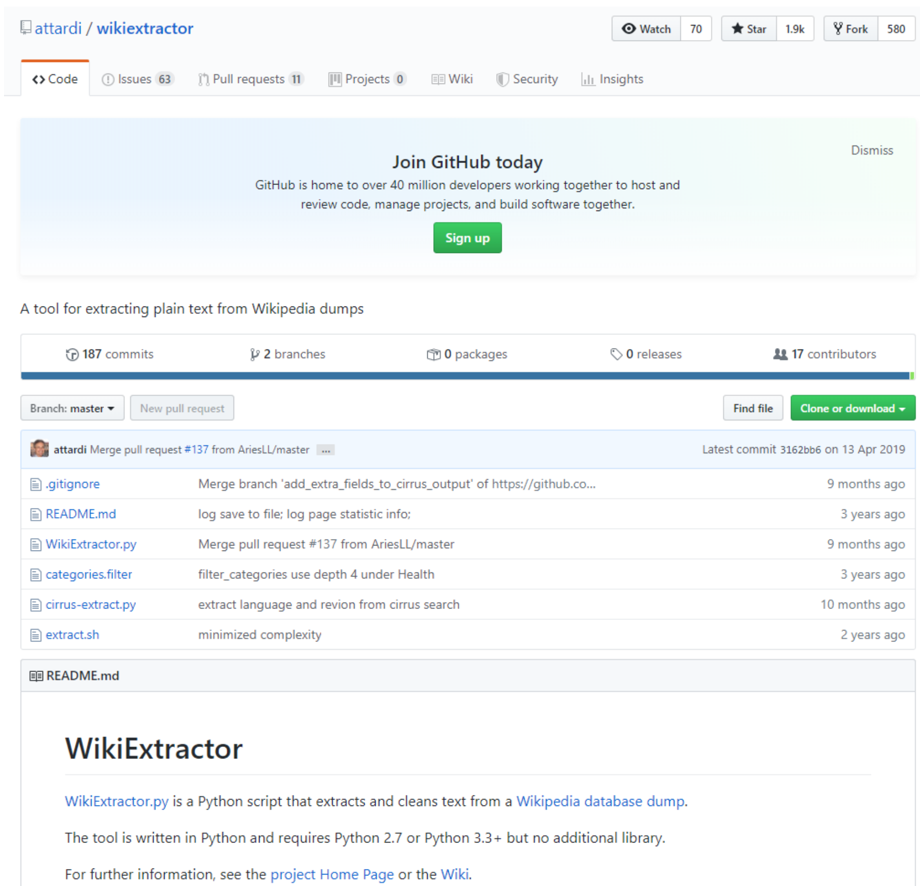


https://github.com/soskek/bookcorpus

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Pre-training BERT
  - ✓ Datasets for pre-training
    - ▪ English Wikipedia (2,500M words)



https://github.com/attardi/wikiextractor

BERT: Bidirectional Encoder Representations from Transformer

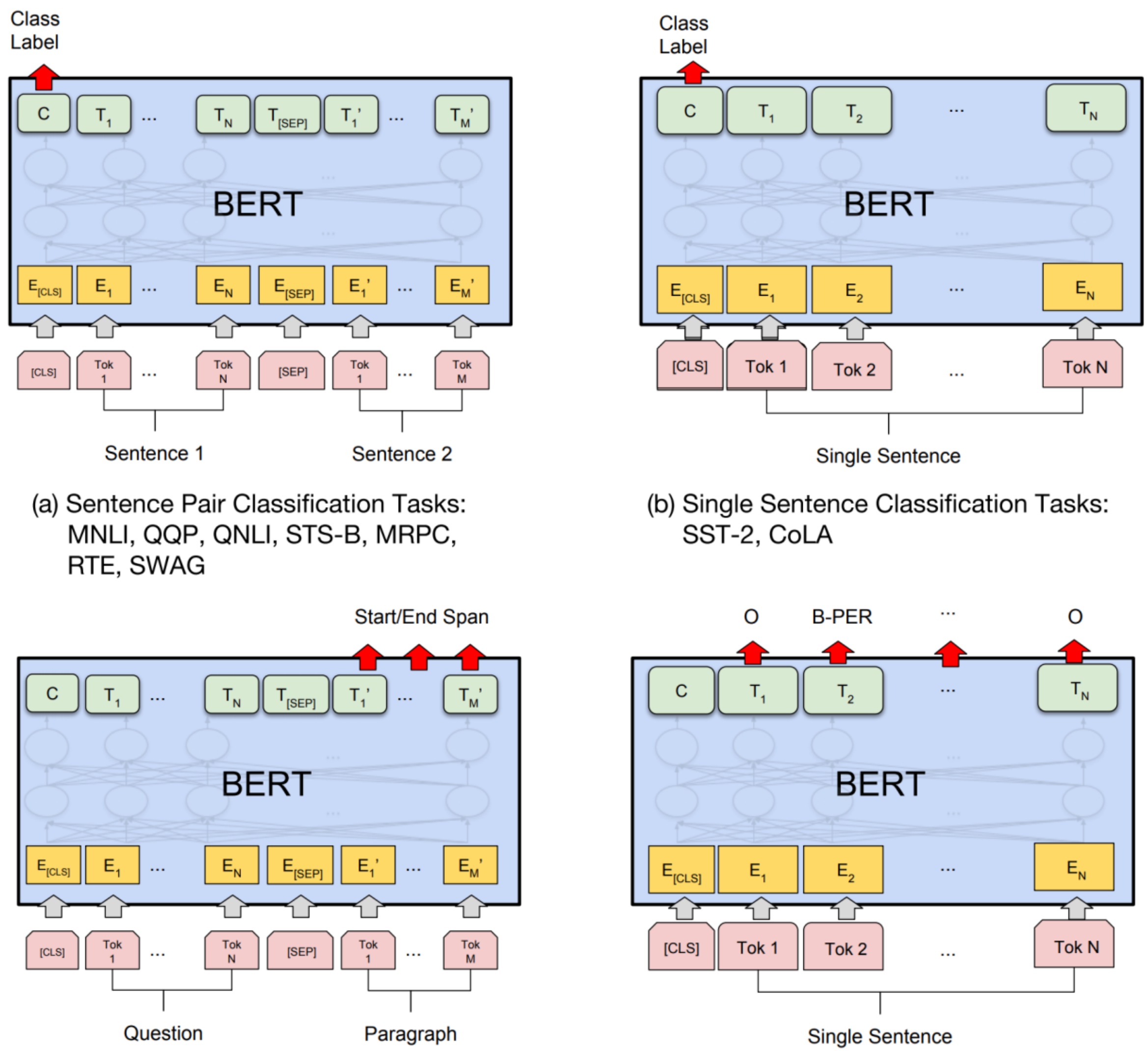
Devlin et. al (2018)

- Pre-training BERT
  - ✓ Hyper-parameter settings
    - Maximum token length: 512
    - Batch size: 256
    - Adam with learning rate of 1e-4, beta1 = 0.9 beta2 = 0.999
    - L2 weight decay of 0.01
    - Learning rate warmup over the first 10,000 steps, linear decay of the learning rate
    - Dropout probability of 0.1 on all layers
    - GeLU activation function rather than standard ReLU
    - $\text{BERT}_{\text{BASE}}$ took 4 days with 16 TPUs and $\text{BERT}_{\text{LARGE}}$ took 4 days with 64 TPUs
    - Pre-train the model with sequence length of 128 for 90% of the steps
    - The rest 10% of the steps are trained with sequence length of 512

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Fine-tuning BERT



(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

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Experiments
  - ✓ A collection of diverse NLU tasks

Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
1	ERNIE Team - Baidu	ERNIE		90.2	72.2	97.5	93.0/90.7	92.9/92.5	75.2/90.8	91.2	90.6	98.0	90.9	94.5	49.4
2	王玮	ALICE v2 large ensemble (Alibaba DAMO NLP)		90.1	73.2	97.1	93.9/91.9	93.0/92.5	74.8/91.0	90.8	90.6	99.2	87.4	94.5	48.7
3	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5	50.2
4	T5 Team - Google	T5		89.7	70.8	97.1	91.9/89.2	92.5/92.1	74.6/90.4	92.0	91.7	96.7	92.5	93.2	53.1
5	XLNet Team	XLNet (ensemble)		89.5	70.2	97.1	92.9/90.5	93.0/92.6	74.7/90.4	90.9	90.9	99.0	88.5	92.5	48.4
6	ALBERT-Team Google Language	ALBERT (Ensemble)		89.4	69.1	97.1	93.4/91.2	92.5/92.0	74.2/90.5	91.3	91.0	99.2	89.2	91.8	50.2
7	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)		88.8	68.0	96.8	93.1/90.8	92.4/92.2	74.8/90.3	91.1	90.7	98.8	88.7	89.0	50.1
8	Facebook AI	RoBERTa		88.5	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	98.9	88.2	89.0	48.7
9	Junjie Yang	HIRE-RoBERTa		88.3	68.6	97.1	93.0/90.7	92.4/92.0	74.3/90.2	90.7	90.4	95.5	87.9	89.0	49.3
10	Microsoft D365 AI & MSR AI	MT-DNN-ensemble		87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	42.8

https://gluebenchmark.com/leaderboard

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	79.8	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: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Experiments

✓ Ablation study 1: Effect of Pre-training Tasks

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 study 2: Effect of Model Size

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

BERT: Bidirectional Encoder Representations from Transformer

Devlin et. al (2018)

- Experiments
  - ✓ Ablation study 3: Feature-based Approach with BERT
    - CoNLL-2003 NER task

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	-


ANY
questions?