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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova

단국대학교 모바일시스템공학과 양윤성

Contents

1-(1). Background

What is problem of GPT?

- Use only transformer decoder → Left to right (one direction)
- Unable to learn 2 sentences → Can’t know relationship
  - Difficulty in understanding exact context
  - Weakness in tasks like QA, NLI …
- High learning costs and time-consuming

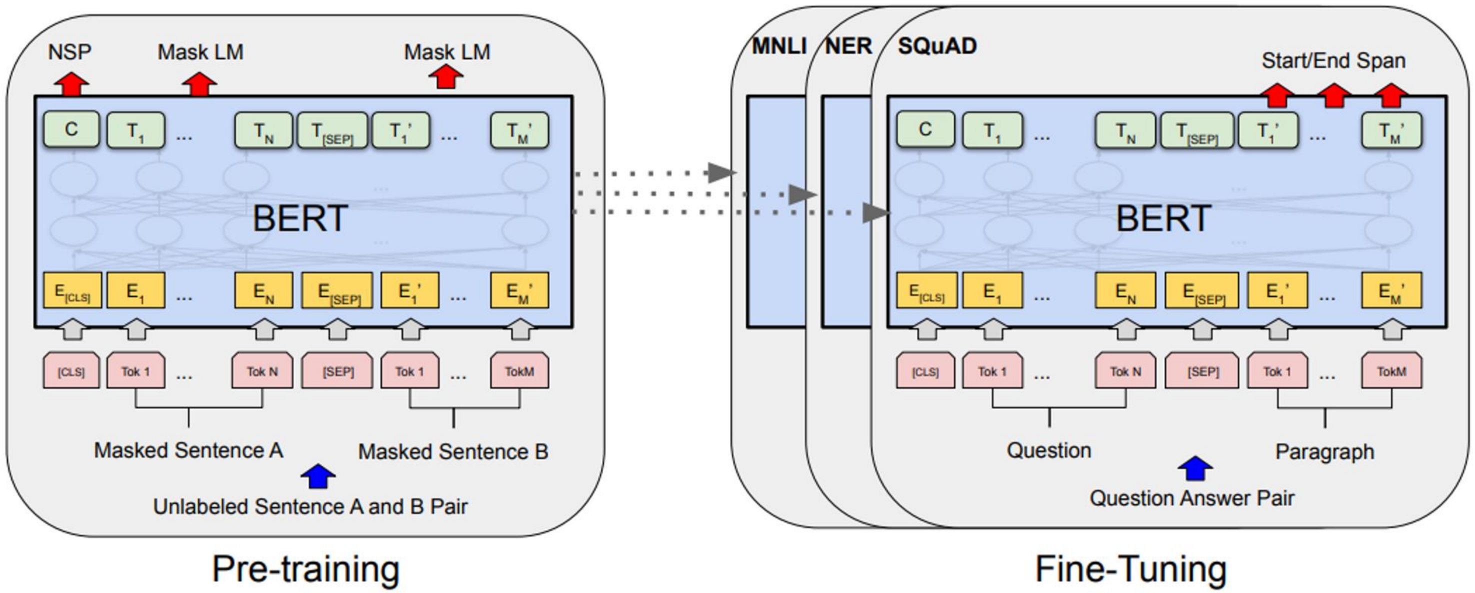
1-(2). Main Idea

BERT : Bidirectional Encoder Representations from Transformer

- Use only transformer encoder → Because of bidirectional feature
- Two or more sentences can be input sequence
- Two pre-training processes (MLM, NSP)
- Apply to tasks by adding only 1 simple layer (Fine-Tuning)

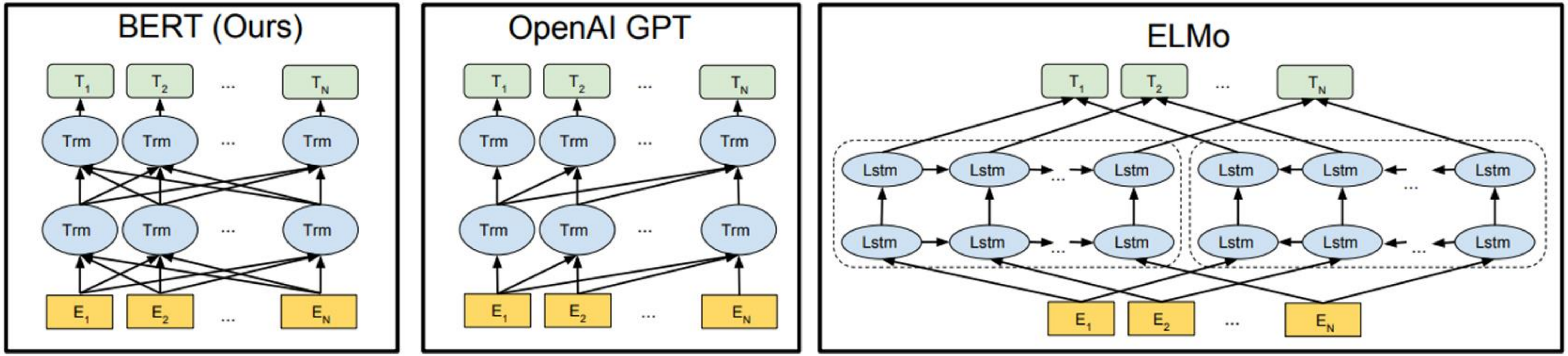
Contents

2-(1). Model Architecture



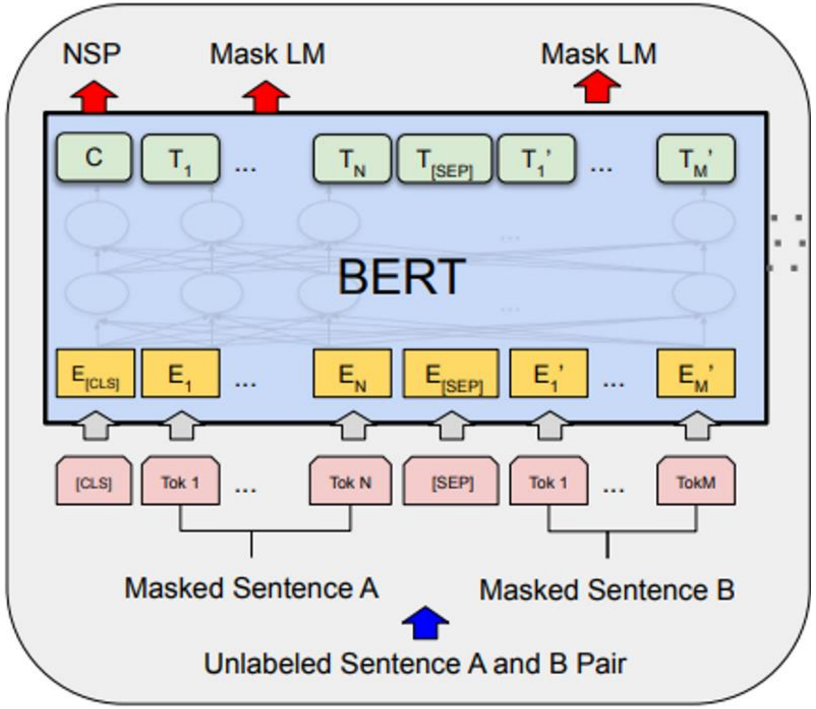
- The overall structure of BERT
- Two pre-training processes (MLM, NSP)
- Add layer by specific task (Fine-Tuning)

2-(1). Model Architecture



- ELMo : Bidirectional LSTM
- GPT : Transformer decoder (Left to right)
- BERT : Transformer encoder (Bidirectional)

2-(2). Input Representation



Pre-training

1. Masked Language Model (MLM)

2. Use CLS, SEP Token to separate inputs

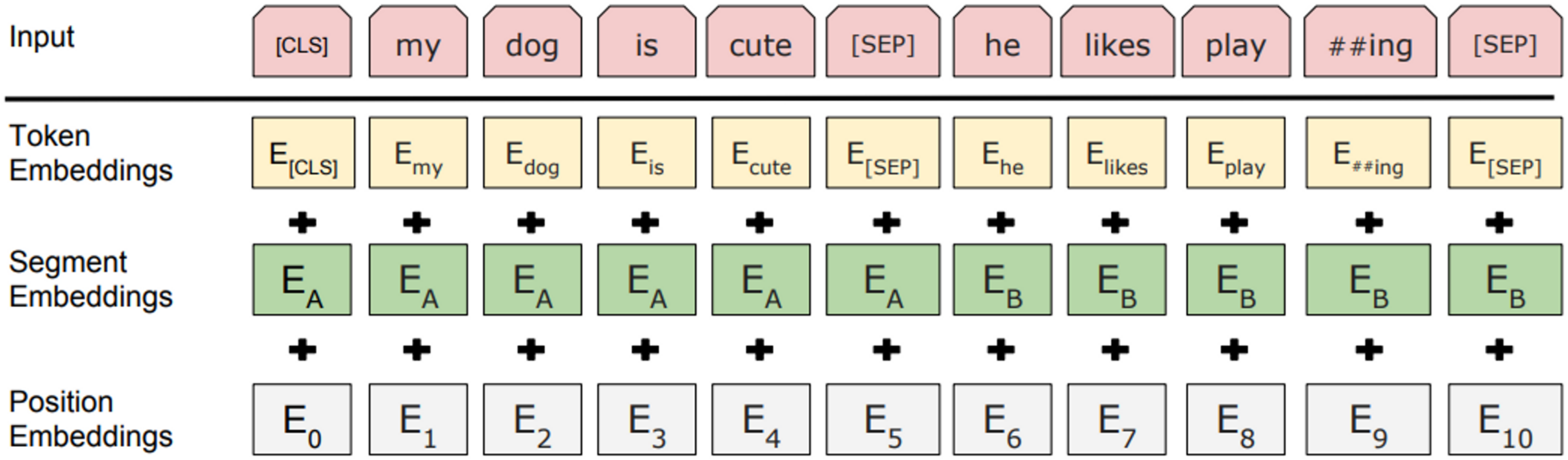
3. Same work as Transformer encoder

4. Extract final hidden state vector

   (Training Goal : T_n = Masked E_n)

5. Next-Sentence Prediction (NSP)

2-(2). Input Representation



Sum of three embedding vectors

- Position Embeddings : Add location info, same as Transformer
- Segment Embeddings : Add sequence order info

2-(2). Input Representation

Token Embedding : Word Piece

- Distinguish by space → Use Word Piece
  - ✓ More effective method to distinguish tokens (Ex. play and -ing)
- Make word meaning clearer
- Be good at new words or typing errors

2-(3). Masked Language Model

- Randomly change 15% of the total
- Model learns to accurately predict masked tokens
- Problems of MLM
  - Fine-tuning doesn't require masked tokens
  - Miss-match occurs between fine tuning and pre-training

2-(3). Masked Language Model

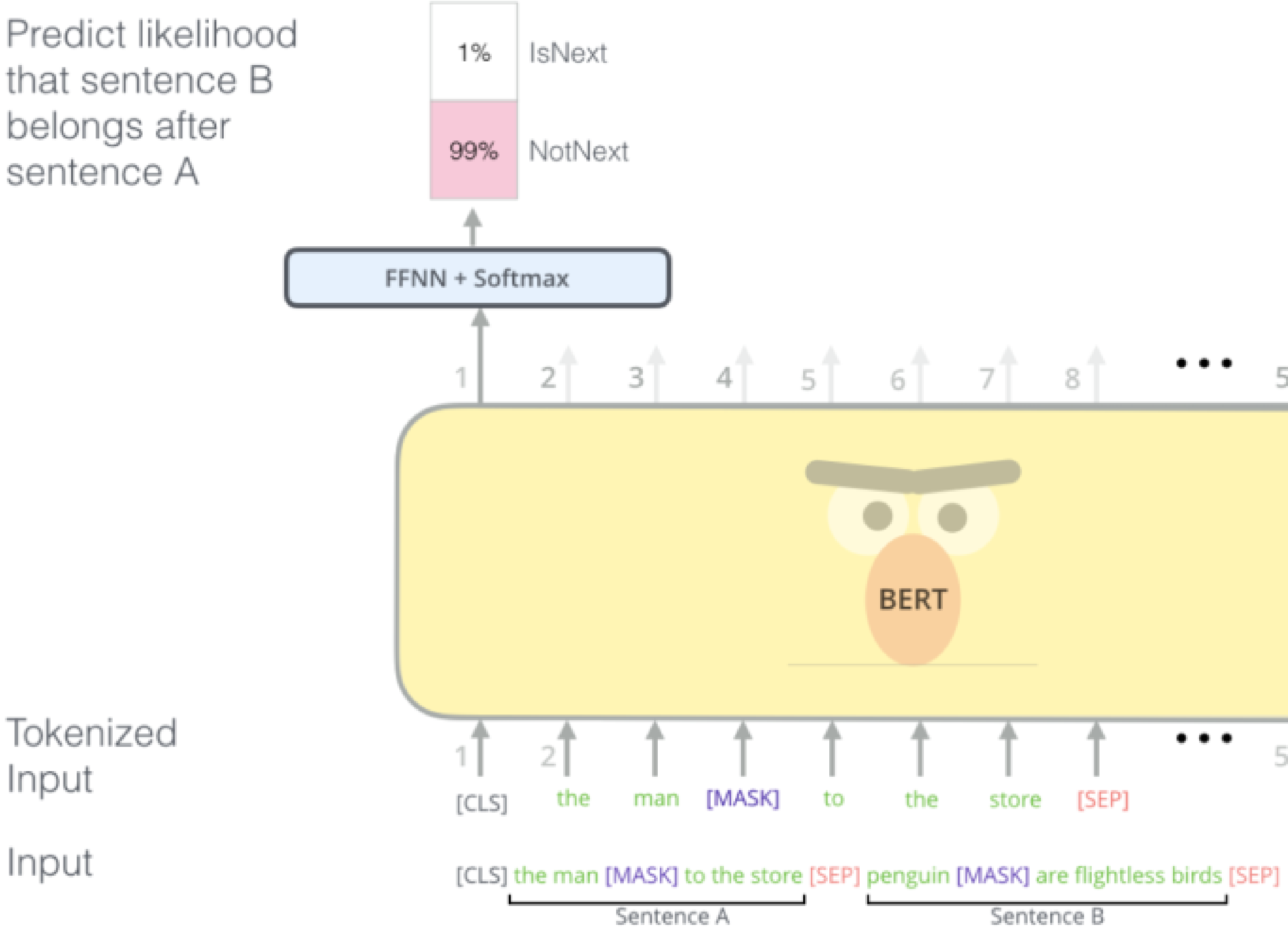
Masking Rates			Dev Set Results		
MASK	SAME	RND	MNLI	NER	
			Fine-tune	Fine-tune	Feature-based
80%	10%	10%	84.2	95.4	94.9
100%	0%	0%	84.3	94.9	94.0
80%	0%	20%	84.1	95.2	94.6
80%	20%	0%	84.4	95.2	94.7
0%	20%	80%	83.7	94.8	94.6
0%	0%	100%	83.6	94.9	94.6

- Solution : [MASK] token 80%, Random token 10%, Unchanged 10%
- Minimize miss-match problem

2-(4). Next-Sentence Prediction

- To understand the relationship between sentences (such as QA, NLI)
- Predict whether sentence A and B appear continuously in actual corpus
- Dataset : 50% is continuous sentences, 50% is chosen randomly
- Apply IsText/NotText label to the final output of [CLS]
  - IsText : B is continuous sentence that follows A
  - NotText : B is randomly selected sentence

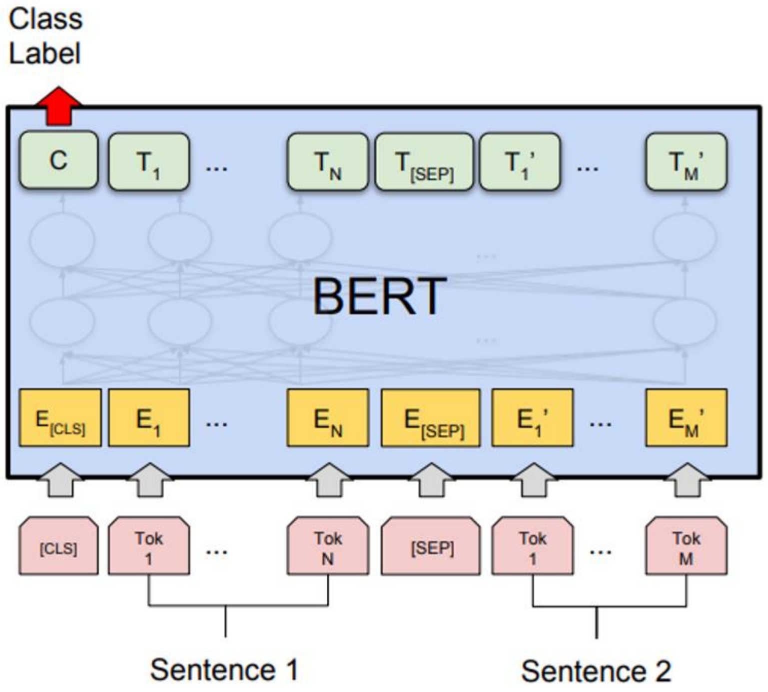
2-(4). Next-Sentence Prediction



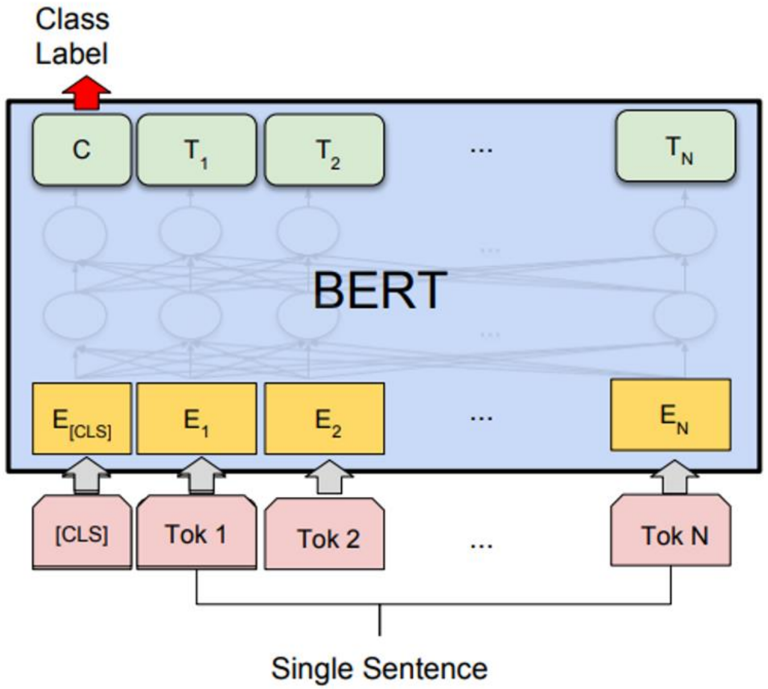
2-(5). Fine-Tuning

- 2-(2) ~ 2-(4) : Learning model of BERT with unsupervised corpus
- Fine-Tuning : Plug in the task specific inputs and outputs into BERT
- Use weight, dataset, objective functions for each task
- Provide the flexibility to apply to a variety of NLP task

2-(5). Fine-Tuning

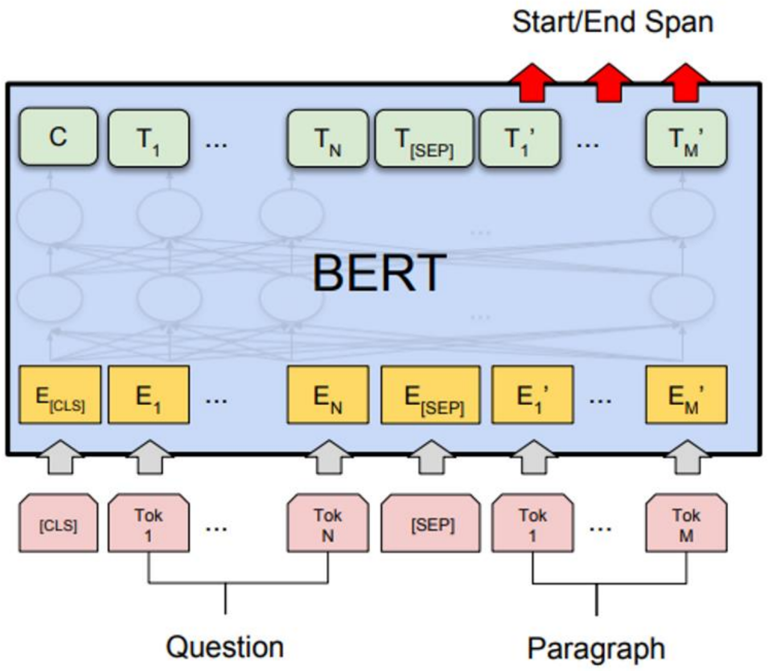


(a) Sentence pair classification tasks

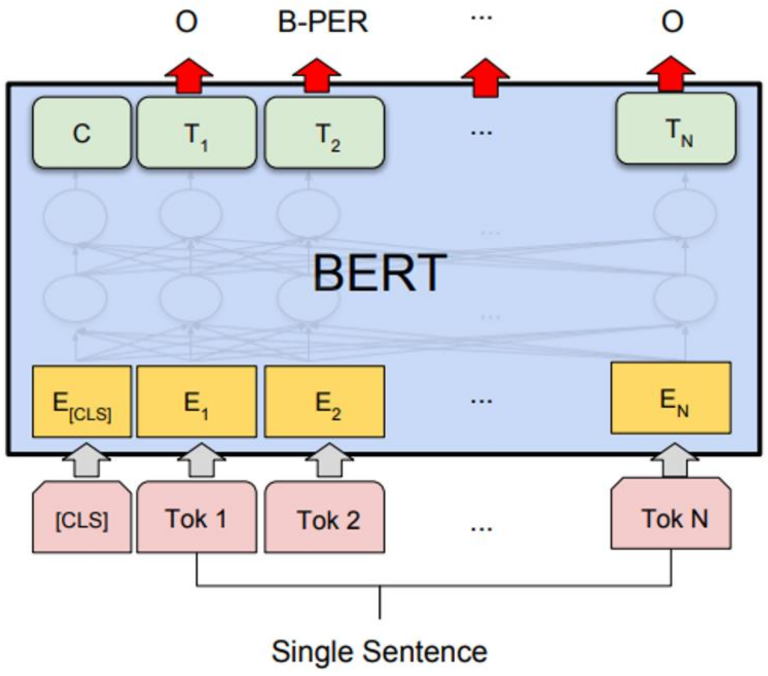


(b) Single sentence classification tasks

2-(5). Fine-Tuning



(c) Question answering tasks



(d) Single sentence tagging tasks

Contents

3-(1). GLEU Score

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**

- Performance improved for all NLP tasks
- Effective for even small datasets

3-(2). SQuAD and SWAG

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD 1.1 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

Table 3: SQuAD 2.0 results. We exclude entries that use BERT as one of their components.

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

Table 4: SWAG Dev and Test accuracies. †Human performance is measured with 100 samples, as reported in the SWAG paper.

3-(3). 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

- Importance of pre-training → Including NSP performs better
  - No NSP : Trained without NSP
  - LTR & No NSP : Trained as a left-to-right LM without NSP, like GPT

3-(4). 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

Table 6: Ablation over BERT model size. #L = the number of layers; #H = hidden size; #A = number of attention heads. "LM (ppl)" is the masked LM perplexity of held-out training data.

- Larger structure show better performance
  - Base → #L: 12, #H: 768, #A: 12
  - Large → #L: 24, #H: 1024, #A: 16

3-(5). 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		
BERT$_{LARGE}$	96.6	92.8
BERT$_{BASE}$	96.4	92.4
Feature-based approach (BERT$_{BASE}$)		
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	-

- Named Entity Recognition results
- Best performance(96.1) is only 0.3 difference from the result of fine-tuning(96.4)

✓ Demonstrate performance as a feature-based approach to BERT

Contents

4. Conclusion

- Active use of unsupervised learning
  - Use unlabeled data → More data available
  - Pre-Training and Fine-Tuning → Increased learning accuracy
- Complement the limitations of GPT
  - Bidirectional → Great contextualization
  - Low cost and time

Thank You!

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