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

Contents

1 Introduction

Language Model Pre-training

- Language model pre-training has been shown to be effective for improving many NLP tasks

- Two existing strategies

  -> *feature-based* (ELMo): *pre-trained representations as additional features*

  -> *fine-tuning* (GPT): *minimal task-specific parameters, simply fine-tuning*

Limitations of Current Techniques

- Standard language models are unidirectional

  -> ELMo: *independently trained left-to-right and right-to-left LMs*

  -> GPT: *left-to-right LMs*

- Unidirectional model restrict the power of the pre-trained representation

Goal

- Improve the fine-tuning based approaches

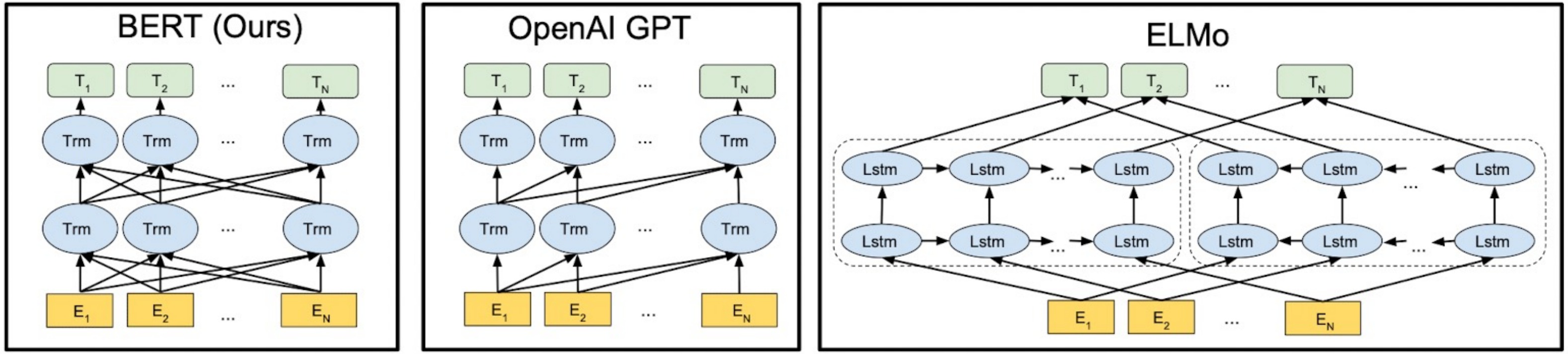
- Masked Language Model (MLM)

  -> *Randomly masks some of the input tokens and predict based on context*

  -> *Allow to pre-train a deep bidirectional Transformer*

- Next Sentence Prediction (NSP)

Differences in Pre-training Model Architectures



- BERT: Bidirectional Transformer
- GPT: Left-to-right Transformer
- ELMo: Concatenation of left-to-right and right-to- left LSTM

Contributions

- Demonstrate the importance of bidirectional pre-training

  -> *MLM enable pre-trained deep bidirectional representations*

- First fine-tuning based model that achieves SOTA performance on a large suite of sentence-level and token-level tasks

  -> *Pre-trained representations reduce the need for heavy engineering*

2

Related Work

Unsupervised Feature-based Approaches

- Extracted feature is used as input to supervised training

- Word embeddings

  -> *Integral part of NLP, offering significant improvements*

- ELMo

  -> *Extract context-sensitive features*

  -> *Not deeply bidirectional(concatenation of left-to-right and right-to-left)*

Unsupervised Fine-tuning Approaches

- Use pre-train model and fine-tuning it

  -> *Few parameters need to be learned from scratch*

- GPT

  -> *Left-to-right language modeling*

  -> *Unidirectional*

3

Methods

BERT Training Steps

- Pre-training

  -> *Training on unlabeled data*

- Fine-tuning

  -> *BERT model is first initialized with the pre-trained parameters*

  -> *Parameters are fine-tuned using labeled data from the downstream tasks*

BERT Model Architecture

- Multi-layer bidirectional Transformer encoder based on Transformer

- Results on two model sizes

    -> $BERT_{BASE}$ (*L*=12, *H*=768, *A*=12, *Parameters*=110M)

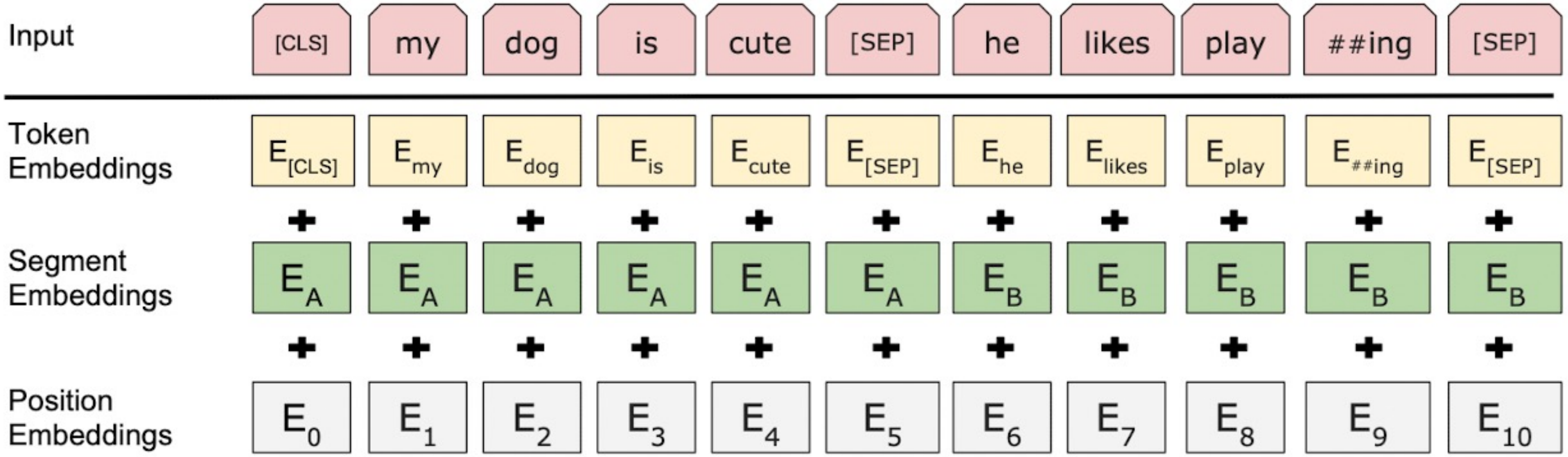
    -> $BERT_{LARGE}$ (*L*=24, *H*=1024, *A*=16, *Parameters*=340M)

    -> *L: number of layers, H: hidden size, A: number of self-attention heads*

Input Representations

- Input representation is able to represent both a single sentence and a pair of sentences in one token sequence
- A ‘sequence’ refers to the input token sequence to BERT, which may be a single sentence or two sentences packed together

Input Representations



- Input embeddings are the sum of the token, segment, position embeddings.

Input Embeddings

- Token Embeddings

  -> *Represent each individual token in the input sentence*

- Segment Embeddings

  -> *Differentiate between the two sentences in pair of sentences*

- Position Embeddings

  -> *Represent the position of each word in input sentence*

Token Embeddings

- WordPiece embeddings with a 30,000 token vocabulary

- First token of sequence is a special classification token [CLS]

  -> *Represent the entire sentence in a single vector*

- Special token [SEP] separate a single sequence into sentence pairs

  -> *Represent the position of each word in input sentence*

Training

- Pre-training

  -> *MLM, NSP*

  -> *BooksCorpus, English Wikipedia*

- Fine-tuning

  -> *Simply plug in the task-specific inputs into BERT*

  -> *Only new parameters are classification layer weights*

Masked Language Model

- Some percentage of the input tokens is masked randomly

- Only predict the masked words

  -> *Final hidden vectors are fed into an output softmax*

- Train a deep bidirectional representation

Masked Language Model

- 15% of tokens are masked at random in each sequence

  -> *80% of token is replaced with [MASK]*

  -> *10% of token is replaced with random*

  -> *10% of token is unchanged*

- Train a deep bidirectional representation

Masked Language Model

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

- Reduce the mismatch between pre-training and fine-tuning
- Introducing noise and forcing the model to generalize well
- Allow the model to learn about the context without difficulty of having to predict missing token

Next Sentence Prediction

- Train a model to understand the relationship between two sentences

  -> *Can't directly trained by language modeling*

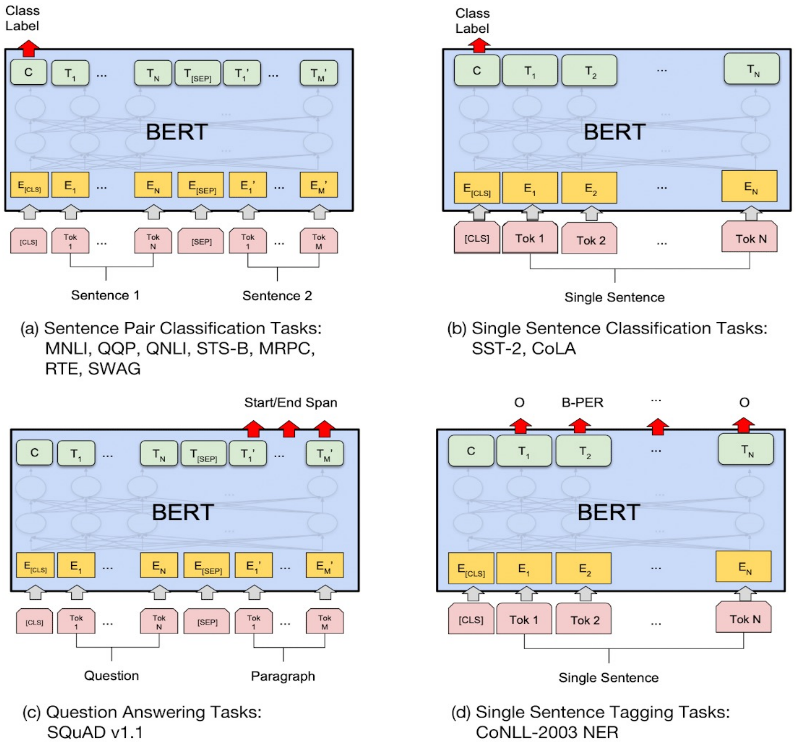
- Pe-train for a binarized next sentence prediction task

  -> *Sentence* A *and* B *for each pre-training example*

  -> 50% *of* B *is actual next sentence, and* 50% *of* B *is random sentence*

- [CLS] token is used for next sentence prediction

Fine-tuning



- The type of input of the model and the output token depend on the task
- Single Sentence vs Sentence Pair
- [CLS] vs Word Tokens

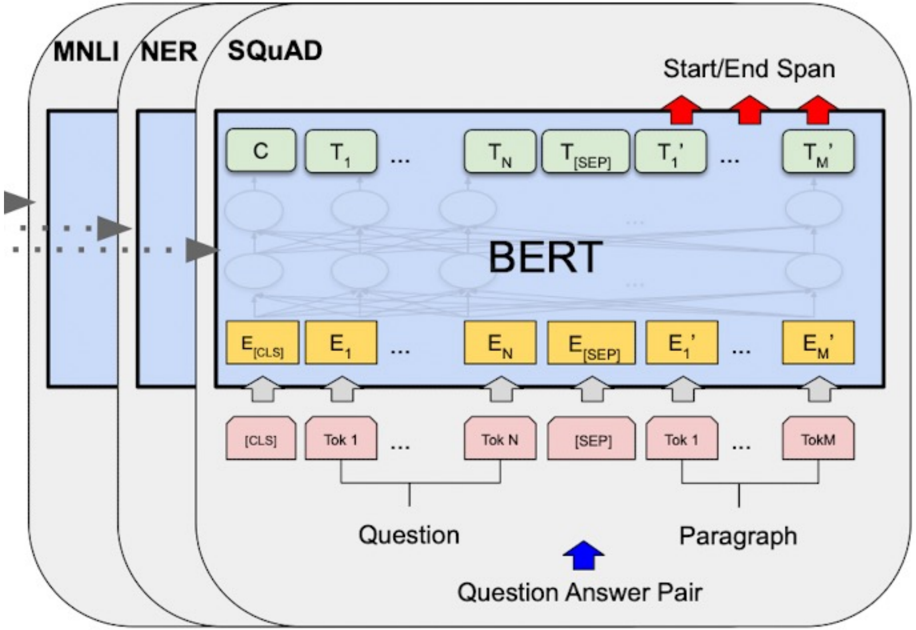
4 Experiments

General Language Understanding Evaluation (GLUE)

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**

- Collection of diverse natural language understanding tasks
- $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$ outperform all systems on all tasks

Question Answering



- Start vector: S, End vector: *E*
- Probability of word *i* being the start of the answer span is computed as dot-product between *Ti* and S followed by a softmax

$$P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}$$

- Score of candidate span from *i to j*

$$S \cdot T_i + E \cdot T_j$$

SQuAD v1.1

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**

- Collection of 100k crowdsourced question/answer pairs
- Input question and passage are represented as singled sequence

SQuAD v2.0

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				
BERT$_{LARGE}$ (Single)	78.7	81.9	80.0	83.1

- Allowing for the possibility that no answer exist
- Probability space for the start and end answer span positions include the position of the [CLS]
- Compare the score of the no-answer span (with threshold) with the best non-null span

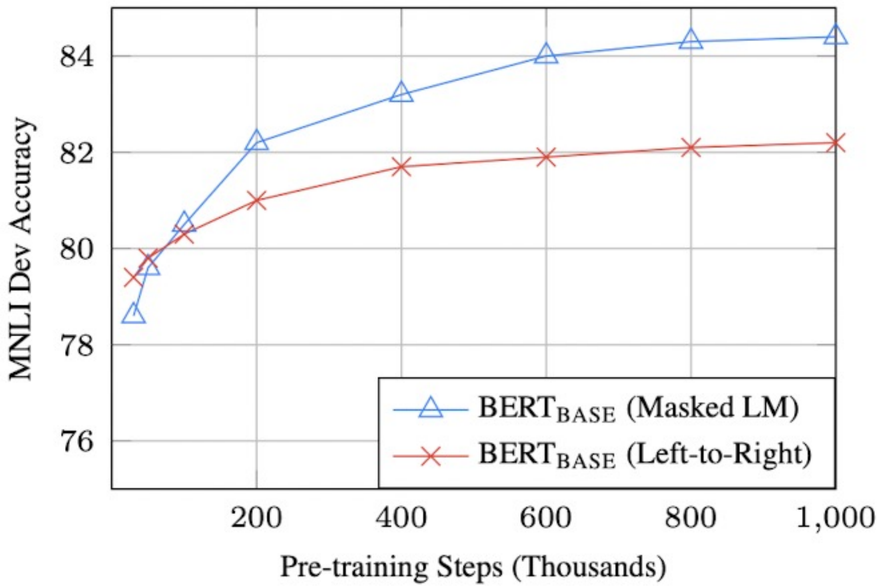
SWAG

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

- Given a sentence, task is to choose the most plausible continuation among four choices
- For fine-tuning, construct four input sequences, each containing the concatenation of given sentence and a possible continuation

5 Ablation Studies

Masked Language Model



- MNLI Dev accuracy after fine-tuning from a checkpoint that has been pre-trained for k steps

Next Sentence Prediction

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

- No NSP

  -> *Only MLM*

- LTR & NO NSP

  -> *Only Left-to-Right LM (GPT)*

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

- Demonstrate that scaling to extreme model sizes also leads to large improvements on very small scale tasks, provided that the model has been sufficiently pre-trained

6 Conclusion

Summary

- Unsupervised pre-training is an integral part of many language understanding systems
- Major contribution is generalizing these findings to deep bidirectional architectures, allowing the same pre-trained model to successfully tackle a broad set of NLP tasks.

Thanks!

Any questions ?