NUDT
National University of
Defense Technology

论文阅读

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

杨森

October 26, 2018

论文概述

- 发布时间：2018.10.11
- 作者：Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova
    - Google AI Language

Contents

What is BERT?

- **BERT** is a new language representation model, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
- Transformer 架构由 Google 在论文 Attention is all you need 中首次提出，最初用于机器翻译。

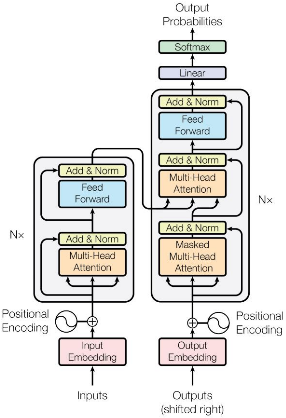


Figure 1: The Transformer - model architecture.

Introduction

How to use BERT?

- BERT adopts the **various embeddings** of token as input
- Pre-train BERT using two unsupervised tasks
  - Masked LM
  - Next Sentence Prediction
- Incorporating BERT with one additional output layer to solve the tasks.
  - sequence-level
  - token-level

Introduction

Motivation

- Language model pre-training is effective for improving NLP tasks
  - natural language inference
  - paraphrasing
  - NER, QA
- Two strategies for applying pre-trained language representations
  - feature-based: specific architectures, additional feature
  - fine-tuning: minimal task-specific parameters, fine-tuning the pre-trained parameters

Introduction

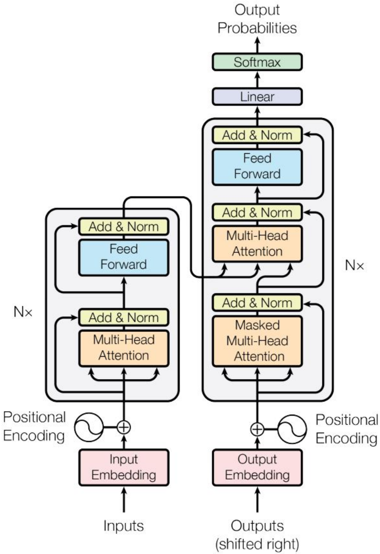
Motivation

- Drawbacks: the power of pre-trained representations is restricted
  - standard (unidirectional, $P(w_i|w_1 \cdots w_{i-1})$) language models limits the choice of architectures for pre-training
  - can't capture the full context ($P(w_i|w_1 \cdots w_{i-1}, w_{i+1} \cdots w_n)$ is better)

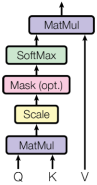
Contribution:

- Demonstrate the importance of bidirectional pre-training
- Introduce the BERT and **eliminate the needs of many heavily engineered task-specific architectures**
- BERT advances the state of the art for eleven NLP tasks

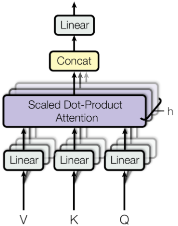
Transformer Architecture



Scaled Dot-Product Attention



Multi-Head Attention



Attention in transformer

- Scaled dot-product attention:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})\,V$$

- Multi-head attention:

$$MultiHead(Q, K, V) = Concat(head_1, \cdots, head_h)$$
$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Transformer

Position embedding

- To make use of the order of the sequence

$$PE(pos, 2i) = sin(pos/10000^{2i/d_{model}})$$
$$PE(pos, 2i+1) = cos(pos/10000^{2i/d_{model}})$$

Position-wise Feed forward

- 包含连个线性变换和一个非线性函数 (ReLU)

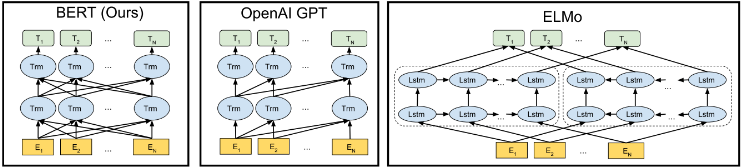
$$FFN(X) = \max(0, xW_1 + b_1)W_2 + b_2$$

BERT architecture

- Num of layers(i.e.,Transformer blocks): L, Hidden size: H, Num of Heads: A, Filter size in FFN: $4H$

Two model size

- **BERT$_{\text{BASE}}$**
  - $L = 12, H = 768, A = 12$
  - total parameters is about $110M$
- **BERT$_{\text{LARGE}}$**
  - $L = 24, H = 1024, A = 16$
  - total parameters is about $340M$



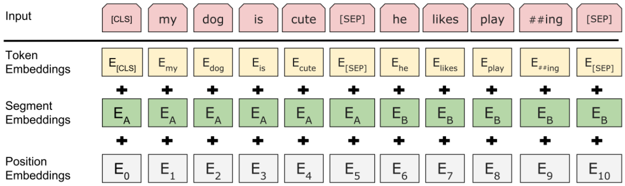
Input Representation

Input

- Single or pair sentences
  - sentence can be an arbitrary span of contiguous text

Embedding

- WordPiece embedding, positional embedding
- The special classification embedding: [CLS]
- Differentiate the sentence in two way:
  - a special token [SEQ]
  - segment embedding



Pre-training Tasks

Task #1: Masked LM

- Mask 15% tokens in each sequence at random
- The final hidden vectors of mask token is used to prediction

How to mask this sentence: my dog is *hairy*

- 80% replace with token [MASK], e.g.,

  my dog is hairy→my dog is [MASK]
- 10% replace with random word, e.g.,

  my dog is hairy→my dog is [apple]
- 10% keep unchanged (**to bias the representation towards the actual word**), e.g.,

  my dog is hairy→my dog is hairy

Pre-training Tasks

Task #2: Next sentence prediction

- To train a model that understands sentence relationships

How to choose sentence pairs <A, B>:

- 50% B is actual next sentence that follows A, e.g.,
  - Input=**[CLS] the man went to [MASK] store [SEQ] he bought a gallon [MASK] milk [SEP]**
  - Label = **IsNext**
- 50% B is a random sentence, e.g.,
  - Input = **[CLS] the man went to [MASK] store [SEQ] he bought a gallon [MASK] milk [SEP]**
  - Label = **NotNext**

Pre-training Procedure

Concatenate two Corpus

- BooksCorpus (800M words)
- English Wikipedia(2,500M words)

Generate training input

- Sample two spans of text as a sentence(typically longer than single sentences)
- The combined length is $\leq$ 512 tokens
- Mask 15% tokens

Loss

- Sum of the mean masked LM likehood and mean next sentence prediction likehood

Pre-training Procedure

Train

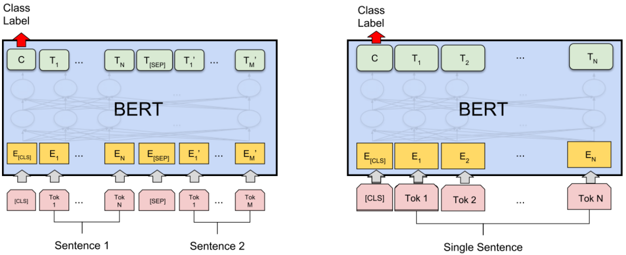
- Batch size: 256 sequences
- Steps: 1,000,000 (about 40 epocs over the 3.3 billion word)
- Adam lr 1e-4, $\beta_1 = 0.9, \beta_2 = 0.999$, L2 weight decay of 0.01, dropout 0.1
- Learning rate warmup over first 10,000 steps, and linear decay
- Activation: gelu
  - $GELU(x) = xP(X \leq x), x \sim N(\mu, \sigma^2)$
- **BERT$_{\text{BASE}}$ is trained on 16 TPU, BERT$_{\text{LARGE}}$ is on 64 TPU, each pre-train 4 days**

Tasks

The 11 tasks in the paper：

- Single sentence tasks
  - CoLA, SST-2
- Similarity and paraphrase tasks
  - MRPC, QQP, STS-B
- Inference tasks
  - MNLI, QNLI, RTE, WNLI, SWAG
- Question answering
  - SQuAD v1.1
- Named entity recognition
  - CoNLL 2003

Fine-tuning Procedure



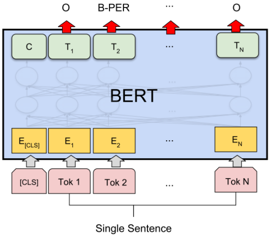
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

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

For sequence-level classification task

- Take the final hidden state for the [CLS] token
- New parameters: $W \in \mathbb{R}^{K \times H}$, K is num of labels
- The aim is to maximize the log-probability

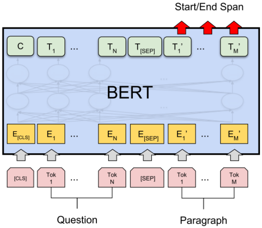
Fine-tuning Procedure



NER task（CoNLL 2003）

- Feed the final hidden representation $T_i \in R^H$ into a classification layer

Fine-tuning Procedure



QA task (SQuAD)

- New parameters: start vector $S \in \mathbb{R}^H$ and end vector $E \in \mathbb{R}^H$
- The prob of word i being the start of answer span (same for end of span)

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

- The training objective is the log-likehood of corrent start and end positions

Fine-tuning Procedure

Hyperparameters in fine-tuning

- Most model hyperparameters are same as pre-training
- Dropout probability is always kept at 0.1
- The optimal hyperparameter values are task-specific
  - Batch size: 16, 32
  - Learning rate(Adam): 5e-5, 3e-5, 2e-5
  - Number of epochs: 3, 4

Experiments Results

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
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

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				
$\text{BERT}_{\text{BASE}}$ (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 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

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

Table 3: CoNLL-2003 Named Entity Recognition results. The hyperparameters were selected using the Dev set, and the reported Dev and Test scores are averaged over 5 random restarts using those hyperparameters.

Ablation Studies

Effect of pre-training tasks

- No "next sentence prediction(NSP)"
- Left-to-Right(LTR)
- + BiLSTM : adds a randomly initialized BiLSTM on top of the “LTR + No NSP” model during fine-tuning

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

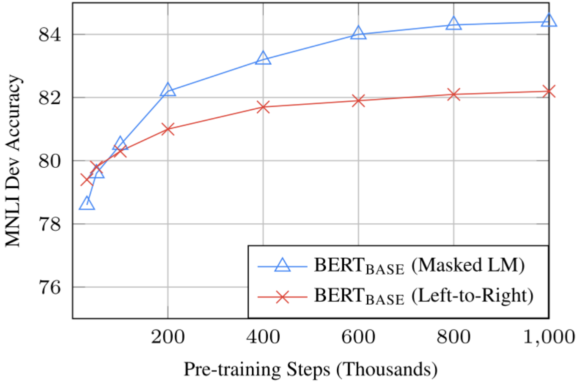
Ablation Studies

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

Ablation Studies

Effect of Number of Training Steps



- BERT need large amount of pre-training
- MLM outperforms the LTR model while it converge slightly slower

Ablation Studies

Feature-based Approach with BERT

- Test on CoNLL-2013 NER task
- Use BERT representation without fine-tuning
- The classification model is a two-layer 768-dimensional BiLSTM

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

- 一些评价

地位类似于resnet在图像，里程碑式的工作，宣告着nlp范式的改变。以后研究工作估计很多都要使用他初始化，就像之前大家使用word2vec一样自然。

这肯定是NLP领域近期最重要的进展。

这两天被这篇BERT的paper刷屏了，目测接下来会出现一系列"pre-training is all you need"的paper（开玩笑）。BERT是一个语言表征模型（language representation model），通过超大数据、巨大模型、和极大的计算开销训练而成，在11个自然语言处理的任务中取得了最优（state-of-

全文一个公式都没有，有啥好嗨的

发布于 2018-10-17

- 难以复现
  - 强大算力，大量数据
- 我们该如何做

参考资料

论文:

- Attention is all you need

网页:

- Transformer 模型的实现
- BERT 模型解读