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

Bidirectional Encoder Representations from Transformers

Maya Angelova

Recent NLP Highlights
Potsdam University

mangelova@uni-potsdam.de

July 12, 2019

Contents

Overview

- won the best paper award at North American Chapter of the Association for Computational Linguistics 2019 **naacl2019**
- takes both the previous and next tokens into account when predicting
- is trained on a next sentence prediction task
- uses the Transformer architecture for encoding
- has minimal difference between the pre-trained architecture and the final downstream architecture
- performs better when given more parameters, even on small datasets **blogml**

Core Ideas

- Semi-supervised Sequence Learning **dai15**
- ELMo **elmo**
- ULMFiT **ulmfit**
- OpenAI transformer **openai**
- Transformer aka *Attention is all you need* **attention**

Transfer Learning

- limitations of methods such as word2vec and GloVe: no context, very shallow language modeling tasks
- therefore ELMo, ULMFiT
- BERT
- next *XLNet: Generalized Autoregressive Pretraining for Language Understanding* (outperforms BERT on 20 tasks), submitted on 19 June 2019

Word2Vec and GloVe

- words as a numeric representation
- word2Vec/GloVe train a vector to capture semantic and syntactic relationships

-0.34	-0.84	0.20	-0.26	-0.12	0.2

Figure 1: The GloVe word embedding of the word "stick": a vector of 200 floats, rounded to two decimals**illustrated**

- the word "stick" would be represented by this vector no-matter what the context

ELMo: Embeddings from Language Models

- uses a bi-directional LSTM trained on a specific task
- encompasses previous and next words in the context
- looks at the entire sentence before assigning each word an embedding
- represents a certain word as a linear combination of corresponding hidden layers including its embedding
- trained to predict the next word in a sequence of words
- ELMO embeddings can be integrated as a simple concatenation to the embedding layer
- can make use of small datasets more efficiently

ELMO Embedding

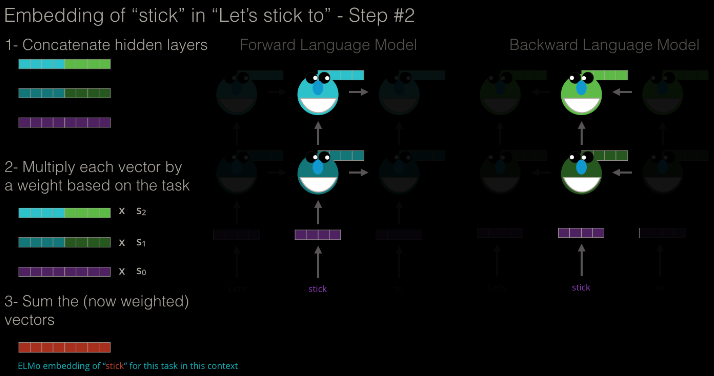

Figure 2: ELMo’s contextualized embedding through grouping together hidden states concatenation followed by weighted summation **illustrated**

ULM-FiT: Universal Language Model Fine-tuning

- based on multi-layer bi-LSTM network without attention
- enables transfer learning for NLP tasks
  1. general LM pre-training
  2. target task LM fine-tuning
  3. target task classifier fine-tuning

Figure 3: Fine-tune a pre-trained model and use it for text classification on a new dataset **ulmfitblog**

OpenAI Transformer

- a fine-tunable unidirectional pre-trained model based on the Transformer
- stacked 12 decoder layers with self-attention layer
- transfer learning
  1. unsupervised training on a large collection of free text corpora
  2. supervised fine-tuning the pre-trained layers for downstream tasks **illustrated**

Differences

- generally language models are unidirectional or left-to-right/right-to-left
- ELMO and ULMFit: bidirectional LSTM based standard L2R and R2L language model
- BERT is also bidirectional but uses the Transformer instead of LSTM
- BERT randomly masks words in the sentence and predicts them

Architecture comparison

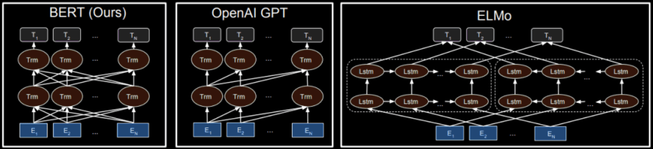

Figure 4: BERT is deeply bidirectional, OpenAI GPT is unidirectional, and ELMo is shallowly bidirectional **bert**

2-Phase BERT

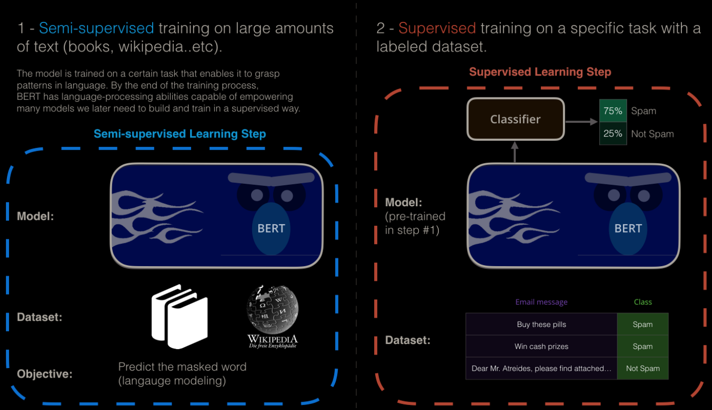

Figure 5: Two phase BERT **bert**

Example: Sentence Classification

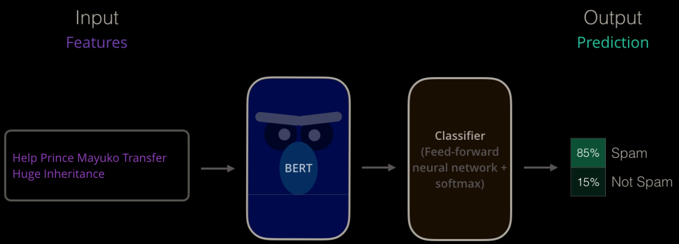

Figure 6: Sentence Classification with BERT **illustrated**

BERT architecture

- BERT base: 110M params, has 12 Transformer blocks (encoder layers) consisting of:
  - FFNN with 768 hidden units
  - 12 attention heads
- BERT large: 340M params, 24 Transformer blocks (encoder layers) consisting of:
  - FFNN with 1024 hidden units
  - 16 attention heads

2-phase BERT

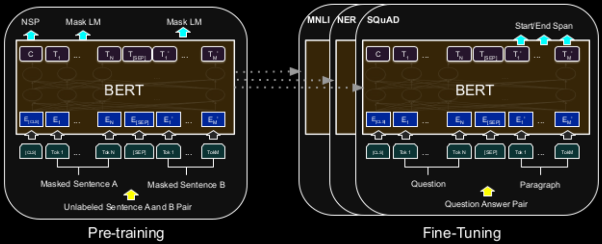

Figure 7: The same pre-trained model parameters are used to initialize models for different down-stream tasks and during fine-tuning all parameters are fine-tuned **bert**

BERT specifics

- BERT uses wordpieces (e.g. playing becomes *play* + *##ing*)
- a 30,000 token vocabulary with the first token of every sequence is a classification token marked with [CLS]
- pre-training corpus: BooksCorpus (800M words) and English Wikipedia (2,500M words)
- uses masked language modeling
- uses next sentence prediction

BERT model input

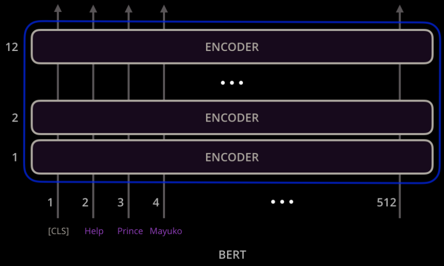

Figure 8: Input schema **blogtds**

BERT model input

Figure 9: Input schema: input embeddings are sum of the token, segmentation and position embeddings **bert**

Training task 1: Masked language modeling

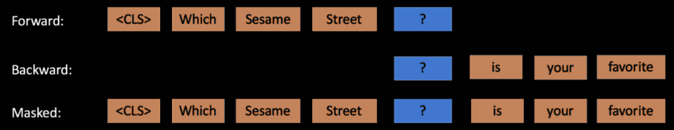

Figure 10: Based on the Cloze task **tdsblog**

Training task 1: Masked language modeling

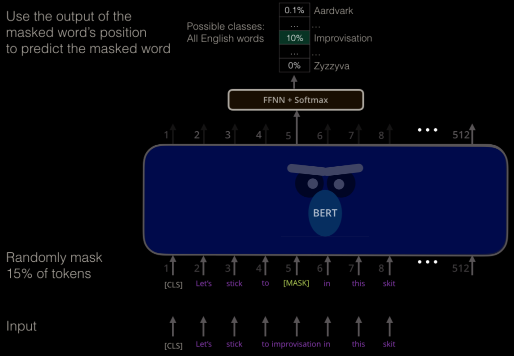

Figure 11: BERT masks 15% of words in the input and asks the model to predict the missing word **illustrated**

Training task 2: next sentence prediction

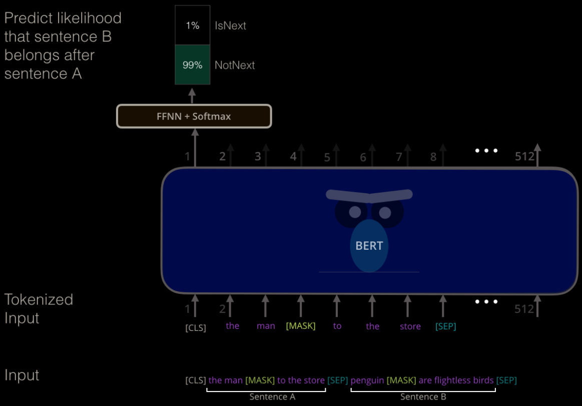

Figure 12: Given two sentences (A and B), is B likely to be the sentence that follows A, or not? **illustrated**

Training task 2: next sentence prediction

- uses a NSP task to pretrain the model for tasks that require an understanding of the relationship between two sentences
- separates the sentences with [SEP] token
- during training 50% of the time label *isNext* is *true*

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: sentence pair classification tasks

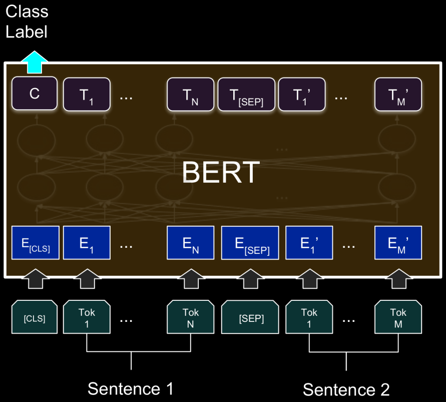

Figure 13: For each downstream task train on the task-specific inputs and outputs with BERT and fine-tune all the parameters end-to-end **bert**

Fine-Tuning: single sentence classification tasks

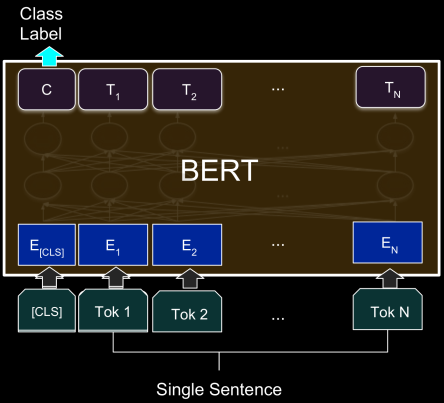

Figure 14: For each downstream task train on the task-specific inputs and outputs with BERT and fine-tune all the parameters end-to-end **bert**

Fine-Tuning: question answering tasks

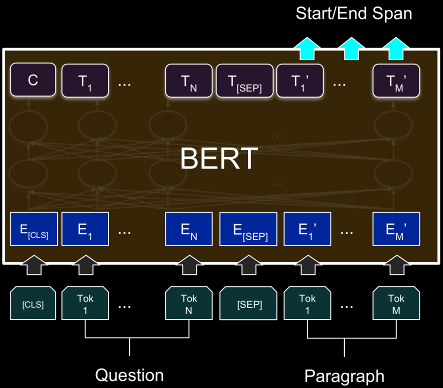

Figure 15: For each downstream task train on the task-specific inputs and outputs with BERT and fine-tune all the parameters end-to-end **bert**

Fine-Tuning: single sentence tagging tasks

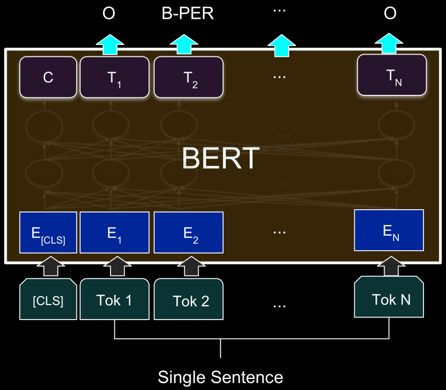

Figure 16: For each downstream task train on the task-specific inputs and outputs with BERT and fine-tune all the parameters end-to-end **bert**

Hyperparameters pretraining

- For pretraining, BERT uses the following hyperparameters:
  - Sequence length (single example): 256
  - Batch size: 512
  - Training steps: 1,000,000 (Approximately 40 epochs)
  - Optimizer: Adam
  - Learning rate: 1e-4
  - Learning rate schedule: Warmup for 10,000 steps, then linear decay
  - Dropout: 0.1
  - Activation function: gelu (Gaussian Error Linear Unit)

Experiments

- General Language Understanding Evaluation (GLUE) benchmark: collection containing diverse natural language understanding tasks
- The Stanford Question Answering Dataset (SQuAD v1.1) is a collection of 100k crowd-sourced question/answer pairs
- SQuAD 2.0
- Situations With Adversarial Generations (SWAG)

NLP Tasks

- Language understanding
- Natural language inference
- Paraphrase detection
- Sentiment analysis
- Linguistic acceptability analysis
- Semantic similarity analysis
- Textual entailment

Findings

1. masked language modeling is more effective than sequential language modeling
2. the next sentence prediction task is necessary

Tasks	Dev Set MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$\mathrm{BERT}_{\mathrm{BASE}}$	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

Figure 17: The results for the ablation study for various pretraining tasks **bert**

Findings

- BERT large model performs better

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

Figure 18: Ablation over BERT model size. #L = number od layers, #H = hidden size, #A = nr of attention heads; LM(ppl) - masked LM perplexity of held-out training data **bert**

Findings

- fine-tuning the entire BERT leads to the best performance
- can be used also as a fixed feature extractor

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

Figure 19: The accuracy on the MNLI dataset vs the number of pretraining steps for left-to-right and masked language modeling **bert**

Thank you for your attention!

- 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics, https://naacl2019.org/
- Paper Dissected: “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding” Explained, https://mlexplained.com/2019/01/07/paper-dissected-bert-pre-training-of-deep-bidirectional-transformers-for-language-understanding-explained/
- *The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)*, https://jalammar.github.io/illustrated-bert/
- A. M. Dai, Q. V. Le, *Semi-supervised Sequence Learning*, CoRR, 2015, http://arxiv.org/abs/1511.01432
- M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, *Deep contextualized word representations*, CoRR, 2018, http://arxiv.org/abs/1802.05365

J. Howard, S. Ruder, *Fine-tuned Language Models for Text Classification*, CoRR, 2018, http://arxiv.org/abs/1801.06146

A.Radford, K. Narashimhan, T. Salimans, I. Sutskever, *Improving Language Understandingby Generative Pre-Training*, https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, CoRR, 2017, http://arxiv.org/abs/1706.03762

Tutorial on Text Classification (NLP) using ULMFiT and fastai Library in Python https://www.analyticsvidhya.com/blog/2018/11/tutorial-text-classification-ulmfit-fastai-library/

- J. Devlin, M. Chang, K. Lee, K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, CoRR, 2018, http://arxiv.org/abs/1810.04805
- NLP: Contextualized word embeddings from BERT, https://towardsdatascience.com/nlp-extract-contextualized-word-embeddings-from-bert-keras-tf-67ef29f60a7b?gi=592f6804340a
- Building a Multi-label Text Classifier using BERT and TensorFlow, https://towardsdatascience.com/building-a-multi-label-text-classifier-using-bert-and-tensorflow-f188e0ecdc5d