Google AI

The latest news from Google AI

Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing

Friday, November 2, 2018

Posted by Jacob Devlin and Ming-Wei Chang, Research Scientists, Google AI Language

One of the biggest challenges in natural language processing (NLP) is the shortage of training data. Because NLP is a diversified field with many distinct tasks, most task-specific datasets contain only a few thousand or a few hundred thousand human-labeled training examples. However, modern deep learning-based NLP models see benefits from much larger amounts of data, improving when trained on millions, or *billions*, of annotated training examples. To help close this gap in data, researchers have developed a variety of techniques for training general purpose language representation models using the enormous amount of unannotated text on the web (known as *pre-training*). The pre-trained model can then be *fine-tuned* on small-data NLP tasks like question answering and sentiment analysis, resulting in substantial accuracy improvements compared to training on these datasets from scratch.

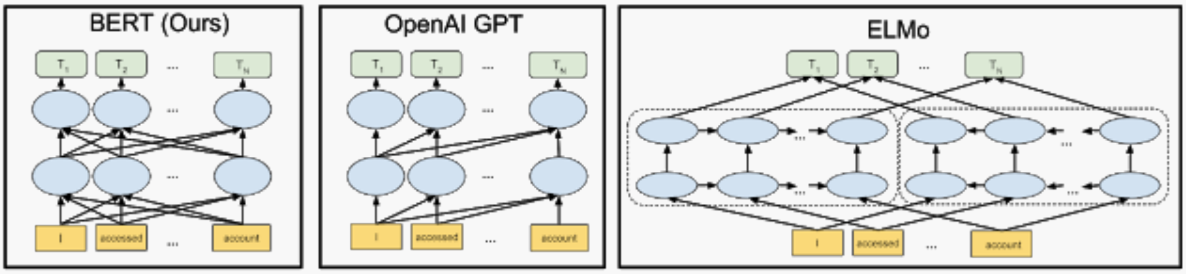
This week, we open sourced a new technique for NLP pre-training called **B**idirectional **E**ncoder **R**epresentations from **T**ransformers, or **BERT**. With this release, anyone in the world can train their own state-of-the-art question answering system (or a variety of other models) in about 30 minutes on a single Cloud TPU, or in a few hours using a single GPU. The release includes source code built on top of TensorFlow and a number of pre-trained language representation models. In our associated paper, we demonstrate state-of-the-art results on 11 NLP tasks, including the very competitive Stanford Question Answering Dataset (SQuAD v1.1).

What Makes BERT Different?

BERT builds upon recent work in pre-training contextual representations — including Semi-supervised Sequence Learning, Generative Pre-Training, ELMo, and ULMFit. However, unlike these previous models, BERT is the first *deeply bidirectional*, *unsupervised* language representation, pre-trained using only a plain text corpus (in this case, Wikipedia).

Why does this matter? Pre-trained representations can either be *context-free* or *contextual*, and *contextual* representations can further be *unidirectional* or *bidirectional*. Context-free models such as word2vec or GloVe generate a single word embedding representation for each word in the vocabulary. For example, the word “*bank*” would have the same context-free representation in “*bank account*” and “*bank of the river.*” Contextual models instead generate a representation of each word that is based on the other words in the sentence. For example, in the sentence “*I accessed the bank account*,” a unidirectional contextual model would represent “*bank*” based on “*I accessed the*” but not “*account*.” However, BERT represents “*bank*” using both its previous and next context — “*I accessed the … account ” — starting from the very bottom of a deep neural network, making it deeply bidirectional.

A visualization of BERT’s neural network architecture compared to previous state-of-the-art contextual pre-training methods is shown below. The arrows indicate the information flow from one layer to the next. The green boxes at the top indicate the final contextualized representation of each input word:

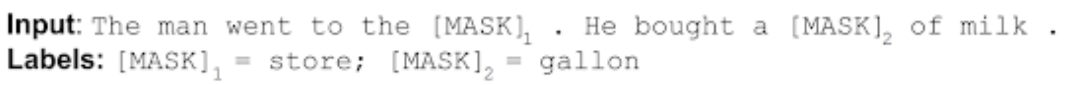


BERT is deeply bidirectional, OpenAI GPT is unidirectional, and ELMo is shallowly bidirectional.

The Strength of Bidirectionality

If bidirectionality is so powerful, why hasn’t it been done before? To understand why, consider that unidirectional models are efficiently trained by predicting each word conditioned on the previous words in the sentence. However, it is not possible to train bidirectional models by simply conditioning each word on its previous *and* next words, since this would allow the word that’s being predicted to indirectly “see itself” in a multi-layer model.

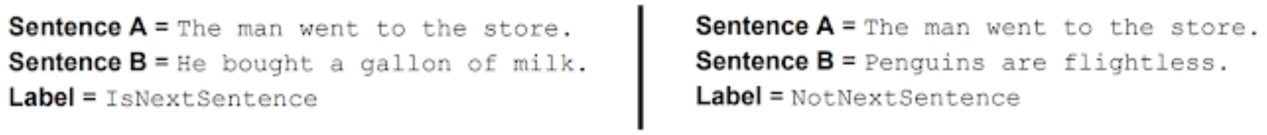
To solve this problem, we use the straightforward technique of masking out some of the words in the input and then condition each word bidirectionally to predict the masked words. For example:



Input: The man went to the [MASK]$_1$. He bought a [MASK]$_2$ of milk .
Labels: [MASK]$_1$ = store; [MASK]$_2$ = gallon

While this idea has been around for a very long time, BERT is the first time it was successfully used to pre-train a deep neural network.

BERT also learns to model relationships between sentences by pre-training on a very simple task that can be generated from any text corpus: Given two sentences A and B, is B the actual next sentence that comes after A in the corpus, or just a random sentence? For example:



Sentence A = The man went to the store.
Sentence B = He bought a gallon of milk.
Label = IsNextSentence

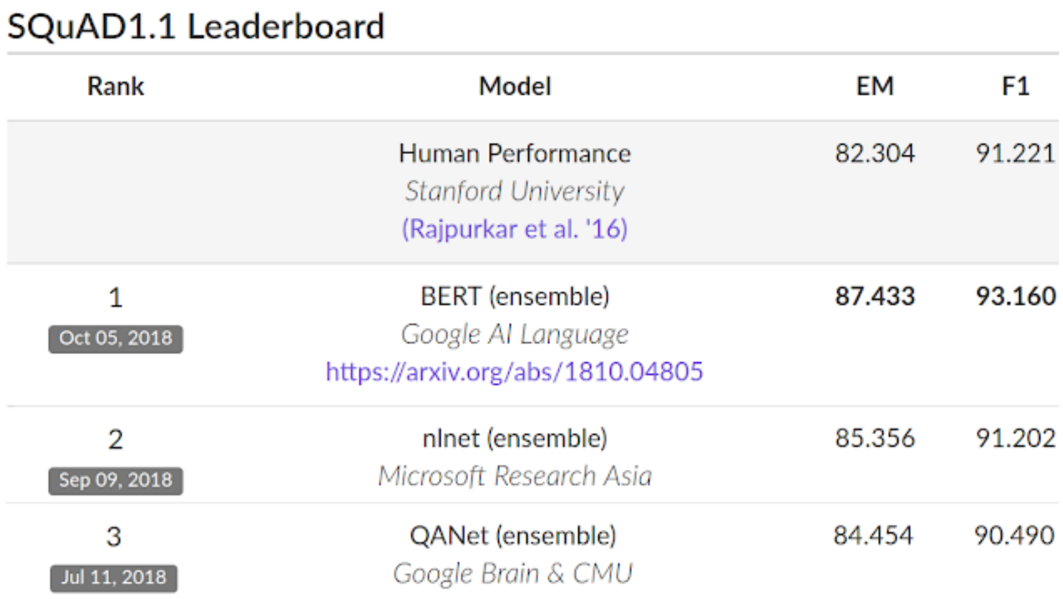
Sentence A = The man went to the store.
Sentence B = Penguins are flightless.
Label = NotNextSentence

Training with Cloud TPUs

Everything that we've described so far might seem fairly straightforward, so what's the missing piece that made it work so well? Cloud TPUs. Cloud TPUs gave us the freedom to quickly experiment, debug, and tweak our models, which was critical in allowing us to move beyond existing pre-training techniques. The Transformer model architecture, developed by researchers at Google in 2017, also gave us the foundation we needed to make BERT successful. The Transformer is implemented in our open source release, as well as the tensor2tensor library.

Results with BERT

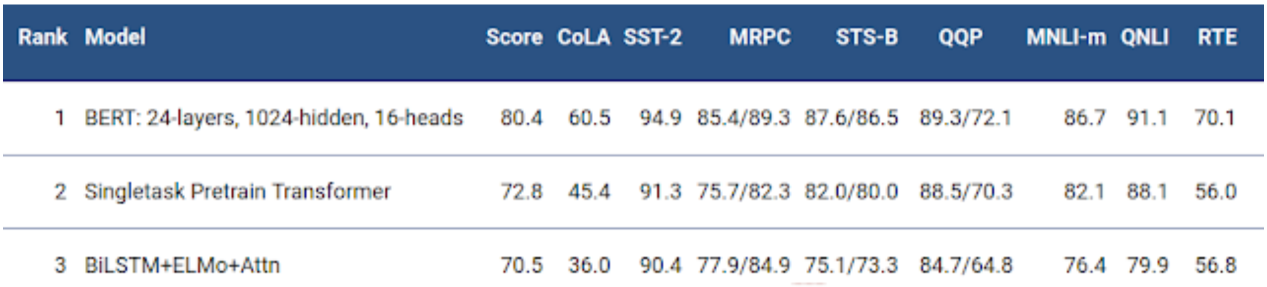
To evaluate performance, we compared BERT to other state-of-the-art NLP systems. Importantly, BERT achieved all of its results with almost no task-specific changes to the neural network architecture. On SQuAD v1.1, BERT achieves 93.2% F1 score (a measure of accuracy), surpassing the previous state-of-the-art score of 91.6% and human-level score of 91.2%:



SQuAD1.1 Leaderboard

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	**87.433**	**93.160**
2 Sep 09, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.356	91.202
3 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490

BERT also improves the state-of-the-art by 7.6% absolute on the very challenging GLUE benchmark, a set of 9 diverse Natural Language Understanding (NLU) tasks. The amount of human-labeled training data in these tasks ranges from 2,500 examples to 400,000 examples, and BERT substantially improves upon the state-of-the-art accuracy on all of them:



Rank	Model	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	QNLI	RTE
1	BERT: 24-layers, 1024-hidden, 16-heads	80.4	60.5	94.9	85.4/89.3	87.6/86.5	89.3/72.1	86.7	91.1	70.1
2	Singletask Pretrain Transformer	72.8	45.4	91.3	75.7/82.3	82.0/80.0	88.5/70.3	82.1	88.1	56.0
3	BiLSTM+ELMo+Attn	70.5	36.0	90.4	77.9/84.9	75.1/73.3	84.7/64.8	76.4	79.9	56.8

Making BERT Work for You

The models that we are releasing can be fine-tuned on a wide variety of NLP tasks in a few hours or less. The open source release also includes code to run pre-training, although we believe the majority of NLP researchers who use BERT will never need to pre-train their own models from scratch. The BERT models that we are releasing today are English-only, but we hope to release models which have been pre-trained on a variety of languages in the near future.

The open source TensorFlow implementation and pointers to pre-trained BERT models can be found at http://goo.gl/language/bert . Alternatively, you can get started using BERT through Colab with the notebook “BERT FineTuning with Cloud TPUs.”

You can also read our paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" for more details.