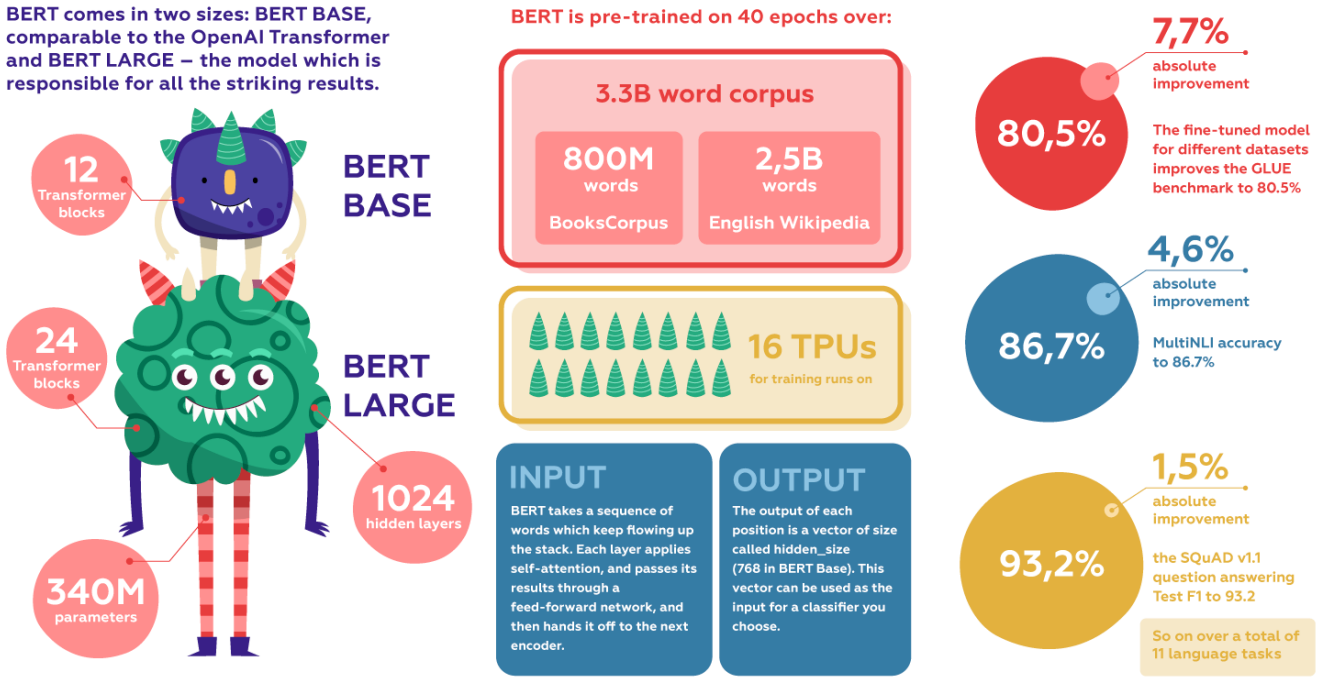
Google’s BERT changing the NLP Landscape

Sciforce Follow
Nov 21, 2019 · 7 min read ★

We write a lot about open problems in Natural Language Processing. We complain a lot when working on NLP projects. We pick on inaccuracies and blatant errors of different models. But what we need to admit is that NLP has already changed and new models have solved the problems that may still linger in our memory. One of such drastic developments is the launch of Google’s Bidirectional Encoder Representations from Transformers, or BERT model — the model that is called the best NLP model ever based on its superior performance over a wide variety of tasks.

When Google researchers presented a deep bidirectional Transformer model that addresses 11 NLP tasks and surpassed even human performance in the challenging area of question answering, it was seen as a game-changer in NLP/NLU.

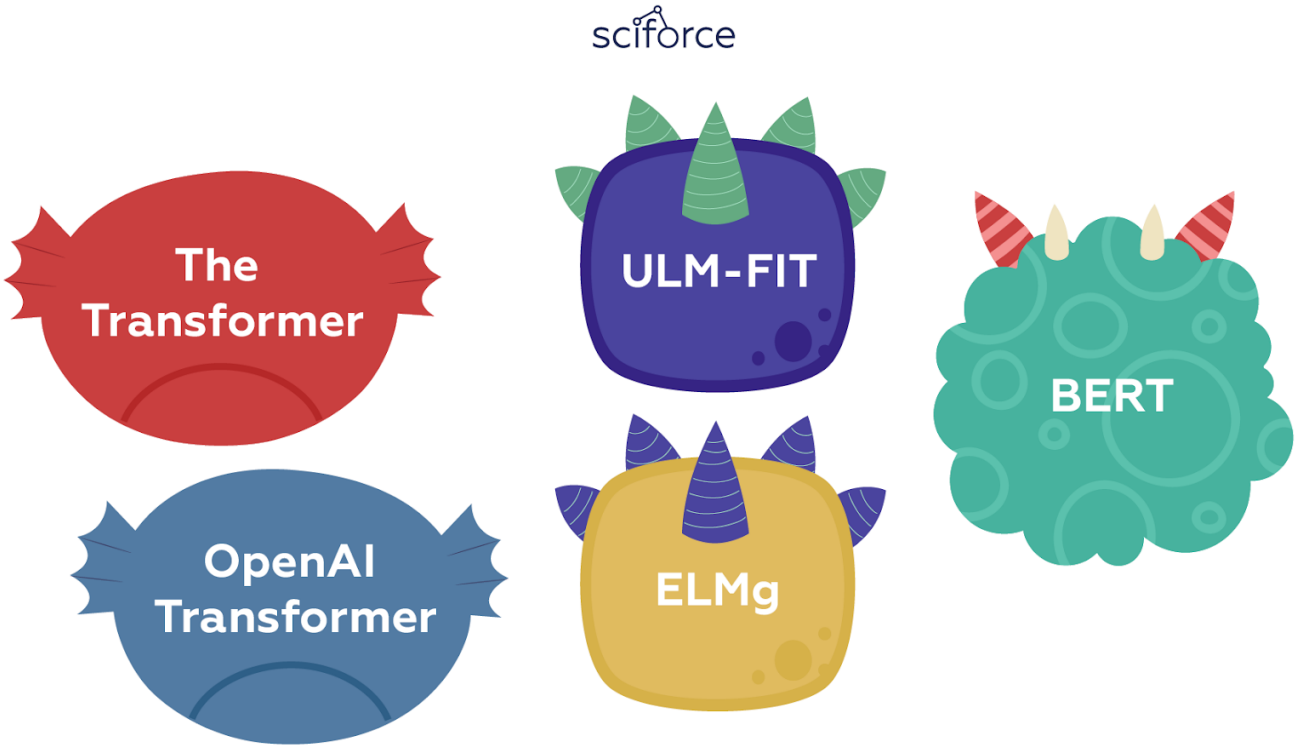
sciforce
BERT model at a glance



- BERT comes in two sizes: BERT BASE, comparable to the OpenAI Transformer and BERT LARGE — the model which is responsible for all the striking results.
- BERT is huge, with 24 Transformer blocks, 1024 hidden layers, and 340M parameters.
- BERT is pre-trained on 40 epochs over a 3.3 billion word corpus, including BooksCorpus (800 million words) and English Wikipedia (2.5 billion words).
- BERT pre-training runs on 16 TPUs for training.
- As input, BERT takes a sequence of words which keep flowing up the stack. Each layer applies self-attention, and passes its results through a feed-forward network, and then hands it off to the next encoder.
- The output of each position is a vector of size called *hidden_size* (768 in BERT Base). This vector can be used as the input for a classifier you choose.
- The fine-tuned model for different datasets improves the GLUE benchmark to 80.5 percent (7.7 percent absolute improvement), MultiNLI accuracy to 86.7 percent (4.6 percent absolute improvement), the SQuAD v1.1 question answering Test F1 to 93.2 (1.5 absolute improvement), and so on over a total of 11 language tasks.

Theories underneath

BERT builds on top of a number of clever ideas that have been bubbling up in the NLP community recently — including but not limited to Semi-supervised Sequence Learning (by Andrew Dai and Quoc Le), Generative Pre-Training, ELMo (by Matthew Peters and researchers from AI2 and UW CSE), ULMFiT (by fast.ai founder Jeremy Howard and Sebastian Ruder), the OpenAI transformer (by OpenAI researchers Radford, Narasimhan, Salimans, and Sutskever), and the Transformer (by Vaswani et al). However, unlike previous models, BERT is the first *deeply bidirectional*, *unsupervised* language representation, pre-trained using only a plain text corpus.



Two Pillars of BERT

BERT builds on two key ideas that paved the way for many of the recent advances in NLP:

- the transformer architecture, and
- unsupervised pre-training.

Transformer Architecture

The Transformer is a sequence model that forgoes the sequential structure of RNN's for a fully attention-based approach. Transformers boast both training efficiency and superior performance in capturing long-distance dependencies compared to the recurrent neural network architecture that falls short on long sequences. What makes BERT different from OpenAI GPT (a left-to-right Transformer) and ELMo (a concatenation of independently trained left-to-right and right- to-left LSTM), is that the model's architecture is a deep bidirectional Transformer encoder.

A bidirectional encoder consists of two independent encoders: one encoding the normal sequence and the other the reversed sequence. The output and final states are concatenated or summed. The deep bidirectional encoder is an alternative bidirectional encoder where the outputs of every layer are summed (or concatenated) before feeding them to the next layer. 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 — the problems that prevented researchers from introducing bidirectional encoders to their models. BERT's solution to overcome the barrier is to 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.

Unsupervised pre-training

It is virtually impossible to separate the two sides of BERT. Apart from being bidirectional, BERT is also pre-trained. A model architecture is first trained on one language modeling objective, and then fine-tuned for a supervised downstream task. The model's weights are learned in advance through two unsupervised tasks: masked language modeling (predicting a missing word given the left and right context in the ***Masked Language Model (MLM)*** method) and the binarized next sentence prediction (predicting whether one sentence follows another). Therefore, BERT doesn't need to be trained from scratch for each new task; rather, its weights are fine-tuned.

Why does this combination matter?

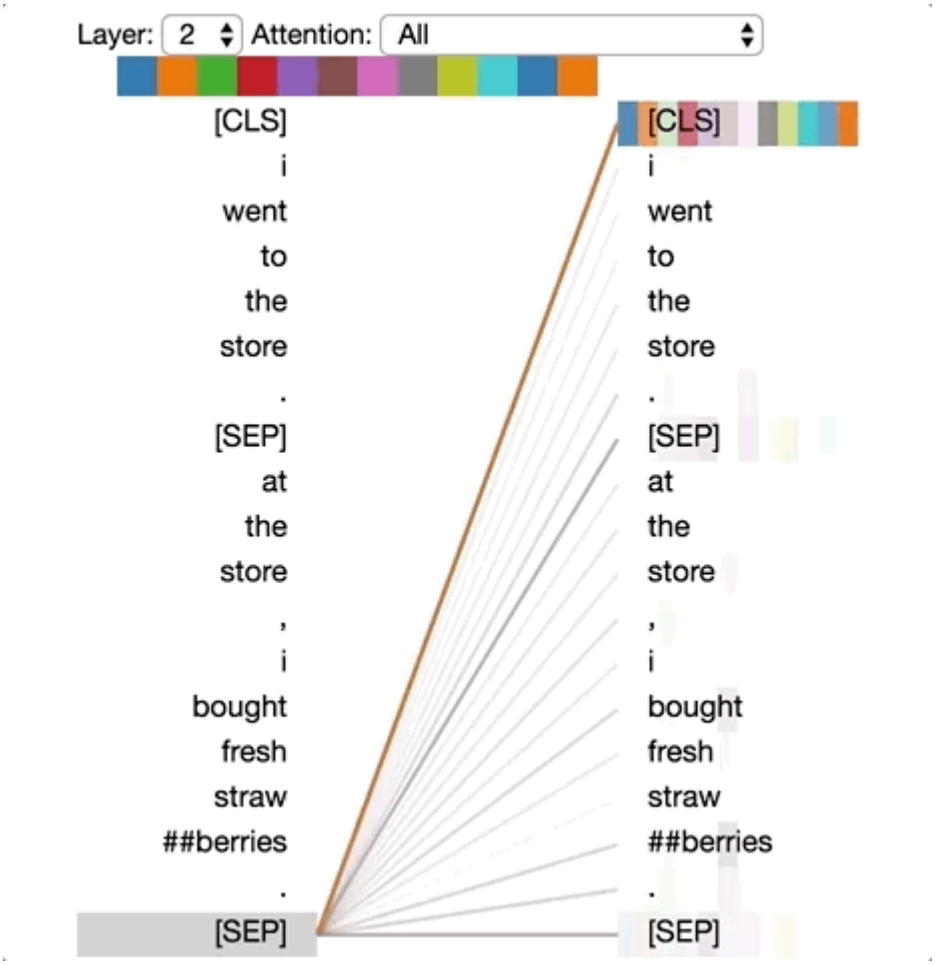
Aylien Research Scientist Sebastian Ruder says in his blog that pre-trained models may have "the same wide-ranging impact on NLP as pretrained ImageNet models had on computer vision." However, pre-trained representations are not homogeneous: they can either be *context-free* or *contextual*, and *contextual* representations can further be

unidirectional or *bidirectional*. While context-free models such as word2vec or GloVe generate a single word embedding representation for each word in the vocabulary, contextual models generate a representation of each word that is based on the other words in the sentence. The bidirectional approach in BERT represents each word using both its previous and next context starting from the very bottom of a deep neural network, making it deeply bidirectional.

The pre-trained model can then be fine-tuned on small-data NLP tasks like question answering and sentiment analysis, and significantly improve the accuracy compared to training from scratch.

Visualizing BERT

Deep-learning models in general are notoriously opaque, and various visualization tools have been developed to help make sense of them. To understand how BERT works, it is possible to visualize attention with the help of Tensor2Tensor.



The tool visualizes attention as lines connecting the position being updated (left) with the position being attended to (right). Colors identify the corresponding attention head(s), while line thickness reflects the attention score. At the top of the tool, the user can select the model layer, as well as one or more attention heads (by clicking on the color patches at the top, representing the 12 heads).

Open-source

Soon after the release of the paper describing the model, the team also open-sourced the code of the model, and made available for download versions of the model that were already pre-trained on massive datasets. These span BERT Base and BERT Large, as well as languages such as English, Chinese, and a multilingual model covering 102 languages trained on wikipedia. Thanks to this invaluable gift, anyone can now build a machine learning model involving language processing to use this powerhouse as a readily-available component — saving time, energy, knowledge, and resources.

The best way to try out BERT directly is through the BERT FineTuning with Cloud TPUs notebook hosted on Google Colab. Besides, it is a good starting point to try Cloud TPUs.

Afterwards you can proceed to the BERT repo and the PyTorch implementation of BERT. On top of it, the AllenNLP library uses this implementation to allow using BERT embeddings with any model.

BERT in practice

BERT was one of our top choices in CALLv3 shared task (the text subtask of which we have actually won). The Spoken CALL Shared Task is an initiative to create an open challenge dataset for speech-enabled CALL(computer-assisted language learning) systems. It is based on data collected from a speech-enabled online tool that helps Swiss German teens practice skills in English conversation.The task is to label pairs as "accept" or "reject", accepting responses which are grammatically and linguistically correct.

We used BERT embeddings to classify the students' phrases as correct or incorrect. More specifically, we used its multi_cased_L-12_H-768_A-12 model trained on Wikipedia and the BookCorpus.

From BERT, we obtained a 768-dimensional vector for each phrase from the dataset. We used German prompts translated using the Google Translate service and the corresponding English answers concatenated via ’|||’ as inputs. This approach turned out to work well in our case. Used in combination with the nnlm model, BERT showed the second best result in our experiments. Besides, we did not perform finetuning because of the scarcity of the data set. However, we believe that with the sufficient amount of data, finetuning of BERT can yield even better results.

Our experiments reconfirmed that the BERT model is a powerful tool that can be used in such a sentence pair tasks as question answering and entailment.

Epilogue: the future is exciting

While we were writing this post, a news came that the Facebook AI team released their code for the XLM/mBERT pretrained models that cover over 100 languages. All code is built on top of PyTorch and you can directly start playing around with the models with a provided ipython notebook. The new method called XLM, published in this year’s paper, provides a technique to pretrain cross-lingual language models based on the popular technique of Transformers. The recent release, therefore, means you can now use pretrained models or train your own to perform machine translation and cross-lingual classification using the above languages and transfer it to low-resource languages, addressing the long-standing problem.

)

Machine Learning NLP Artificial Intelligence Deep Learning Data Science

About Help Legal