BERT: Pre-training of Deep Bi[redacted]ctional Transform[redacted] for Language Und[redacted]tanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

arXiv:1810.04805v2 [cs.CL] 24 May 2019

Abstract

We introduce a new language representation [redacted]l called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransform[redacted]. Unlike recent language representation [redacted]ls (Pet[redacted] et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all lay[redacted]. As a result, the pre-trained BERT [redacted]l can be fine-tuned with just one additional output layer to create state-of-the-art [redacted]ls for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

1 Introduction

Language [redacted]l pre-training has been shown to be effective for improving many natural language processing tasks (Dai and Le, 2015; Pet[redacted] et al., 2018a; Radford et al., 2018; Howard and Ruder, 2018). These include sentence-level tasks such as natural language inference (Bowman et al., 2015; Williams et al., 2018) and paraphrasing (Dolan and Brockett, 2005), which aim to predict the relationships between sentences by analyzing them holistically, as well as token-level tasks such as named entity recognition and question answering, where [redacted]ls are required to produce fine-grained output at the token level (Tjong Kim Sang and De Meulder, 2003; Rajpurkar et al., 2016).

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Pet[redacted] et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific paramet[redacted], and is trained on the downstream tasks by simply fine-tuning *all* pre-trained paramet[redacted]. The two approaches share the same objective function during pre-training, where they use unidirectional language [redacted]ls to learn general language representations.

We argue that current techniques restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language [redacted]ls are unidirectional, and this limits the choice of architectures that can be used during pre-training. For example, in OpenAI GPT, the authors use a left-to-right architecture, where every token can only attend to previous tokens in the self-attention lay[redacted] of the Transformer (Vaswani et al., 2017). Such restrictions are sub-optimal for sentence-level tasks, and could be very harmful when applying fine-tuning based approaches to token-level tasks such as question answering, where it is crucial to incorporate context from both directions.

In this paper, we improve the fine-tuning based approaches by proposing BERT: **B**idirectional **E**ncoder **R**epresentations from **T**ransform[redacted]. BERT alleviates the previously mentioned unidirectionality constraint by using a "masked language [redacted]l" (MLM) pre-training objective, inspired by the Cloze task (Taylor, 1953). The masked language [redacted]l randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary id of the masked word based only on its context. Unlike left-to-right language [redacted]l pre-training, the MLM objective enables the representation to fuse the left and the right context, which allows us to pre-train a deep bidirectional Transformer. In addition to the masked language [redacted]l, we also use a "next sentence prediction" task that jointly pre-trains text-pair representations. The contributions of our paper are as follows:

- We demonstrate the importance of bidirectional pre-training for language representations. Unlike Radford et al. (2018), which uses unidirectional language [redacted]ls for pre-training, BERT uses masked language [redacted]ls to enable pre-trained deep bidirectional representations. This is also in contrast to Pet[redacted] et al. (2018a), which uses a shallow concatenation of independently trained left-to-right and right-to-left LMs.

- We show that pre-trained representations reduce the need for many heavily-engineered task-specific architectures. BERT is the first fine-tuning based representation [redacted]l that achieves state-of-the-art performance on a large suite of sentence-level *and* token-level tasks, outperforming many task-specific architectures.

- BERT advances the state of the art for eleven NLP tasks. The code and pre-trained models are available at `https://github.com/google-research/bert`.

2 Related Work

There is a long history of pre-training general language representations, and we briefly review the most widely-used approaches in this section.

2.1 Unsupervised Feature-based Approaches

Learning widely applicable representations of words has been an active area of research for decades, including non-neural (Brown et al., 1992; Ando and Zhang, 2005; Blitzer et al., 2006) and neural (Mikolov et al., 2013; Pennington et al., 2014) methods. Pre-trained word embeddings are an integral part of [redacted]rn NLP systems, offering significant improvements over embeddings learned from scratch (Turian et al., 2010). To pre-train word embedding vectors, left-to-right language [redacted]ling objectives have been used (Mnih and Hinton, 2009), as well as objectives to discriminate correct from incorrect words in left and right context (Mikolov et al., 2013).

These approaches have been generalized to coarser granularities, such as sentence embeddings (Kiros et al., 2015; Logeswaran and Lee, 2018) or paragraph embeddings (Le and Mikolov, 2014). To train sentence representations, prior work has used objectives to rank candidate next sentences (Jernite et al., 2017; Logeswaran and Lee, 2018), left-to-right generation of next sentence words given a representation of the previous sentence (Kiros et al., 2015), or denoising auto-encoder derived objectives (Hill et al., 2016).

ELMo and its predecessor (Pet[redacted] et al., 2017, 2018a) generalize traditional word embedding research along a different dimension. They extract *context-sensitive* features from a left-to-right and a right-to-left language [redacted]l. The contextual representation of each token is the concatenation of the left-to-right and right-to-left representations. When integrating contextual word embeddings with existing task-specific architectures, ELMo advances the state of the art for several major NLP benchmarks (Pet[redacted] et al., 2018a) including question answering (Rajpurkar et al., 2016), sentiment analysis (Socher et al., 2013), and named entity recognition (Tjong Kim Sang and De Meulder, 2003). Melamud et al. (2016) proposed learning contextual representations through a task to predict a single word from both left and right context using LSTMs. Similar to ELMo, their [redacted]l is feature-based and not deeply bidirectional. Fedus et al. (2018) shows that the cloze task can be used to improve the robustness of text generation [redacted]-[redacted]ls.

2.2 Unsupervised Fine-tuning Approaches

As with the feature-based approaches, the first works in this direction only pre-trained word embedding paramet[redacted] from unlabeled text (Collobert and Weston, 2008).

More recently, sentence or document encod[redacted] which produce contextual token representations have been pre-trained from unlabeled text and fine-tuned for a supervised downstream task (Dai and Le, 2015; Howard and Ruder, 2018; Radford et al., 2018). The advantage of these approaches is that few paramet[redacted] need to be learned from scratch. At least partly due to this advantage, OpenAI GPT (Radford et al., 2018) achieved previously state-of-the-art results on many sentence-level tasks from the GLUE benchmark (Wang et al., 2018a). Left-to-right language [redacted]l-

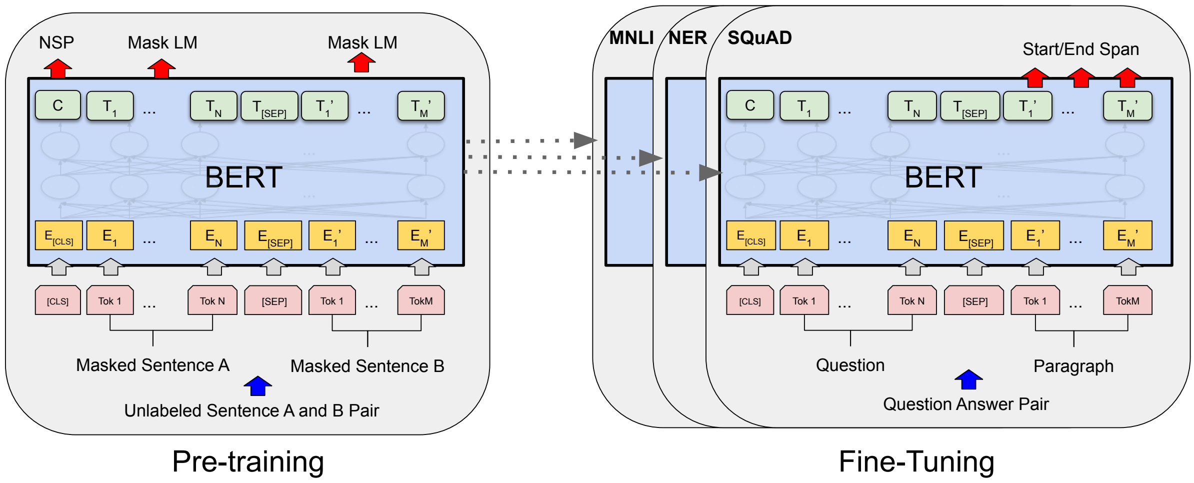

Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output lay[redacted], the same architectures are used in both pre-training and fine-tuning. The same pre-trained [redacted]l paramet[redacted] are used to initialize [redacted]ls for different down-stream tasks. During fine-tuning, all paramet[redacted] are fine-tuned. `[CLS]` is a special symbol added in front of every input example, and `[SEP]` is a special separator token (e.g. separating questions/answ[redacted]).

ing and auto-encoder objectives have been used for pre-training such [redacted]ls (Howard and Ruder, 2018; Radford et al., 2018; Dai and Le, 2015).

2.3 Transfer Learning from Supervised Data

There has also been work showing effective transfer from supervised tasks with large datasets, such as natural language inference (Conneau et al., 2017) and machine translation (McCann et al., 2017). Computer vision research has also demonstrated the importance of transfer learning from large pre-trained [redacted]ls, where an effective recipe is to fine-tune [redacted]ls pre-trained with ImageNet (Deng et al., 2009; Yosinski et al., 2014).

3 BERT

We introduce BERT and its detailed implementation in this section. There are two steps in our framework: *pre-training* and *fine-tuning*. During pre-training, the [redacted]l is trained on unlabeled data over different pre-training tasks. For fine-tuning, the BERT [redacted]l is first initialized with the pre-trained paramet[redacted], and all of the paramet[redacted] are fine-tuned using labeled data from the downstream tasks. Each downstream task has separate fine-tuned [redacted]ls, even though they are initialized with the same pre-trained paramet[redacted]. The question-answering example in Figure 1 will serve as a running example for this section.

A distinctive feature of BERT is its unified architecture across different tasks. There is minimal difference between the pre-trained architecture and the final downstream architecture.

[redacted]l Architecture BERT's [redacted]l architecture is a multi-layer bidirectional Transformer encoder based on the original implementation described in Vaswani et al. (2017) and released in the `tensor2tensor` library.[1] Because the use of Transform[redacted] has become common and our implementation is almost identical to the original, we will omit an exhaustive background description of the [redacted]l architecture and refer read[redacted] to Vaswani et al. (2017) as well as excellent guides such as "The Annotated Transformer."[2]

In this work, we denote the number of lay[redacted] (i.e., Transformer blocks) as L, the hidden size as H, and the number of self-attention heads as A.[3] We primarily report results on two [redacted]l sizes: $\textbf{BERT}_{\textbf{BASE}}$ (L=12, H=768, A=12, Total Paramet[redacted]=110M) and $\textbf{BERT}_{\textbf{LARGE}}$ (L=24, H=1024, A=16, Total Paramet[redacted]=340M).

$\text{BERT}_{\text{BASE}}$ was chosen to have the same [redacted]l size as OpenAI GPT for comparison purposes. Critically, however, the BERT Transformer uses bidirectional self-attention, while the GPT Transformer uses constrained self-attention where every token can only attend to context to its left.[4]

[1] https://github.com/tensorflow/tensor2tensor
[2] http://nlp.seas.harvard.edu/2018/04/03/attention.html
[3] In all cases we set the feed-forward/filter size to be $4H$, i.e., 3072 for the $H = 768$ and 4096 for the $H = 1024$.
[4] We note that in the literature the bidirectional Trans-