BERT: Pre-training of Deep Bidirectional Transformers for

Language Understanding

Abstract

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models (Peters et al., 2018; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT representations can be fine-tuned with just one additional output layer to create state-of-the-art models 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 benchmark to 80.4% ( 7.6% absolute improvement), MultiNLI accuracy to 86.7% ( 5.6% absolute improvement) and the SQuAD v1.1 question answering Test F1 to 93.2 ( 1.5 absolute improvement), outperforming human performance

by 2.0.

1 Introduction

Language model pre-training has shown to be effective for improving many natural language processing tasks (Dai and Le, 2015; Peters et al., 2017, 2018; Radford et al., 2018; Howard and Ruder, 2018). These tasks 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 (Tjong Kim Sang and De Meulder, 2003) and SQuAD question answering (Rajpurkar et al., 2016), where models are required to produce fine-grained output at the token-level.

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 (Peters et al., 2018), uses tasks-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 parameters, and is trained on the downstream tasks by simply fine-tuning the pretrained parameters. In previous work, both approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

We argue that current techniques severely restrict the power of the pre-trained representations, especially for the fine-tuning approaches. The major limitation is that standard language models 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 leftto-

right architecture, where every token can only attended to previous tokens in the self-attention

layers of the Transformer (Vaswani et al., 2017).

Such restrictions are sub-optimal for sentence-level tasks, and could be devastating when applying fine-tuning based approaches to token-level tasks such as SQuAD question answering (Rajpurkar et al., 2016), where it is crucial to incorporate context from both directions.

In this paper, we improve the fine-tuning based approaches by proposing BERT: Bidirectional Encoder Representations from Transformers.

BERT addresses the previously mentioned unidirectional constraints by proposing a new pre-training objective: the “masked language model” (MLM), inspired by the Cloze task (Taylor,1953). The masked language model 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 model pre-training, the MLM objective allows 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 model, we also introduce 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 models for pretraining, BERT uses masked language models to enable pre-trained deep bidirectional representations. This is also in contrast to Peters et al. (2018), which uses a shallow concatenation of independently trained left-to-right and right-to-left LMs.

• We show that pre-trained representations eliminate the needs of many heavily engineered task-specific architectures. BERT is the first fine-tuning based representation model that achieves state-of-the-art performance on a large suite of sentence-level and token-level tasks, outperforming many systems with task-specific architectures.

• BERT advances the state-of-the-art for eleven NLP tasks. We also report extensive ablations

of BERT, demonstrating that the bidirectional nature of our model is the single most important new contribution. The code and pre-trained model will be available at goo.gl/language/bert.1

2 Related Work

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

2.1 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 (Collobert and Weston, 2008; Mikolov et al., 2013; Pennington et al., 2014) methods. Pretrained word embeddings are considered to be an integral part of modern NLP systems, offering significant improvements over embeddings learned from scratch (Turian et al., 2010).

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). As with traditional word embeddings, these learned representations are also typically used as features in a downstream model.

ELMo (Peters et al., 2017) generalizes traditional word embedding research along a different dimension. They propose to extract context-sensitive features from a language model. When

integrating contextual word embeddings with existing task-specific architectures, ELMo advances the state-of-the-art for several major NLP benchmarks (Peters et al., 2018) including question answering (Rajpurkar et al., 2016) on SQuAD, sentiment analysis (Socher et al., 2013), and named entity recognition (Tjong Kim Sang and De Meulder,2003).

2.2 Fine-tuning Approaches

A recent trend in transfer learning from language models (LMs) is to pre-train some model architecture on a LM objective before fine-tuning that same model 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 parameters need to be learned from scratch. At least partly due this advantage, OpenAI GPT (Radford et al., 2018) achieved previously state-of-the-art results on many sentencelevel tasks from the GLUE benchmark (Wang et al., 2018).

2.3 Transfer Learning from Supervised Data

While the advantage of unsupervised pre-training is that there is a nearly unlimited amount of data available, 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 (Mc-Cann et al., 2017). Outside of NLP, computer vision research has also demonstrated the importance of transfer learning from large pre-trained models, where an effective recipe is to fine-tune models pre-trained on ImageNet (Deng et al., 2009; Yosinski et al., 2014).

3 BERT

We introduce BERT and its detailed implementation in this section. We first cover the model architecture and the input representation for BERT.

We then introduce the pre-training tasks, the core innovation in this paper, in Section 3.3. The pre-training procedures, and fine-tuning procedures are detailed in Section 3.4 and 3.5, respectively.

Finally, the differences between BERT and OpenAI GPT are discussed in Section 3.6.

BERTLARGE: L=24, H=1024, A=16, Total Parameters=340M BERTBASE was chosen to have an identical model 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.

We note that in the literature the bidirectional Transformer is often referred to as a “Transformer encoder” while the left-context-only version is referred to as a “Transformer decoder” since it can be used for text generation. The comparisons between BERT, OpenAI GPT and ELMo are shown visually in Figure 1.

3.2 Input Representation

Our input representation is able to unambiguously represent both a single text sentence or a pair of text sentences (e.g., [Question, Answer]) in one token sequence.4 For a given token, its input representation is constructed by summing the corresponding token, segment and position embeddings.

A visual representation of our input representation is given in Figure 2.

The specifics are:

• We use WordPiece embeddings (Wu et al., 2016) with a 30,000 token vocabulary. We denote split word pieces with ##.

• We use learned positional embeddings with supported sequence lengths up to 512 tokens. The first token of every sequence is always the special classification embedding ([CLS]). The final hidden state (i.e., output of Transformer) corresponding to this token is used as the aggregate sequence representation

for classification tasks. For non-classification tasks, this vector is ignored.

• Sentence pairs are packed together into a single sequence. We differentiate the sentences

in two ways. First, we separate them with a special token ([SEP]). Second, we add a learned sentence A embedding to every token of the first sentence and a sentence B embedding

to every token of the second sentence.

• For single-sentence inputs we only use the sentence A embeddings.