

Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks

Nils Reimers and Iryna Gurevych

Ubiquitous Knowledge Processing Lab (UKP-TUD)
Department of Computer Science, Technische Universität Darmstadt
www.ukp.tu-darmstadt.de

Abstract

BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) has set a new state-of-the-art performance on sentence-pair regression tasks like semantic textual similarity (STS). However, it requires that both sentences are fed into the network, which causes a massive computational overhead: Finding the most similar pair in a collection of 10,000 sentences requires about 50 million inference computations (~65 hours) with BERT. The construction of BERT makes it unsuitable for semantic similarity search as well as for unsupervised tasks like clustering.

In this publication, we present Sentence-BERT (SBERT), a modification of the pretrained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. This reduces the effort for finding the most similar pair from 65 hours with BERT / RoBERTa to about 5 seconds with SBERT, while maintaining the accuracy from BERT.

We evaluate SBERT and SRoBERTa on common STS tasks and transfer learning tasks, where it outperforms other state-of-the-art sentence embeddings methods.¹

1 Introduction

In this publication, we present Sentence-BERT (SBERT), a modification of the BERT network using siamese and triplet networks that is able to derive semantically meaningful sentence embeddings². This enables BERT to be used for certain new tasks, which up-to-now were not applicable for BERT. These tasks include large-scale seman-

tic similarity comparison, clustering, and information retrieval via semantic search.

BERT set new state-of-the-art performance on various sentence classification and sentence-pair regression tasks. BERT uses a cross-encoder: Two sentences are passed to the transformer network and the target value is predicted. However, this setup is unsuitable for various pair regression tasks due to too many possible combinations. Finding in a collection of $n = 10\,000$ sentences the pair with the highest similarity requires with BERT $n \cdot (n - 1) / 2 = 49\,995\,000$ inference computations. On a modern V100 GPU, this requires about 65 hours. Similar, finding which of the over 40 million existent questions of Quora is the most similar for a new question could be modeled as a pair-wise comparison with BERT, however, answering a single query would require over 50 hours.

A common method to address clustering and semantic search is to map each sentence to a vector space such that semantically similar sentences are close. Researchers have started to input individual sentences into BERT and to derive fixed-size sentence embeddings. The most commonly used approach is to average the BERT output layer (known as BERT embeddings) or by using the output of the first token (the [CLS] token). As we will show, this common practice yields rather bad sentence embeddings, often worse than averaging GloVe embeddings (Pennington et al., 2014).

To alleviate this issue, we developed SBERT. The siamese network architecture enables that fixed-sized vectors for input sentences can be derived. Using a similarity measure like cosine-similarity or Manhattan / Euclidean distance, semantically similar sentences can be found. These similarity measures can be performed extremely efficient on modern hardware, allowing SBERT to be used for semantic similarity search as well as for clustering. The complexity for finding the

¹Code available: <https://github.com/UKPLab/sentence-transformers>

²With *semantically meaningful* we mean that semantically similar sentences are close in vector space.

most similar sentence pair in a collection of 10,000 sentences is reduced from 65 hours with BERT to the computation of 10,000 sentence embeddings (~5 seconds with SBERT) and computing cosine-similarity (~0.01 seconds). By using optimized index structures, finding the most similar Quora question can be reduced from 50 hours to a few milliseconds (Johnson et al., 2017).

We fine-tune SBERT on NLI data, which creates sentence embeddings that significantly outperform other state-of-the-art sentence embedding methods like InferSent (Conneau et al., 2017) and Universal Sentence Encoder (Cer et al., 2018). On seven Semantic Textual Similarity (STS) tasks, SBERT achieves an improvement of 11.7 points compared to InferSent and 5.5 points compared to Universal Sentence Encoder. On SentEval (Conneau and Kiela, 2018), an evaluation toolkit for sentence embeddings, we achieve an improvement of 2.1 and 2.6 points, respectively.

SBERT can be adapted to a specific task. It sets new state-of-the-art performance on a challenging argument similarity dataset (Misra et al., 2016) and on a triplet dataset to distinguish sentences from different sections of a Wikipedia article (Dor et al., 2018).

The paper is structured in the following way: Section 3 presents SBERT, section 4 evaluates SBERT on common STS tasks and on the challenging argument Facet Similarity (FS) corpus (Misra et al., 2016). Section 5 evaluates SBERT on SentEval. In section 6, we perform an ablation study to test some design aspect of SBERT. In section 7, we compare the computational efficiency of SBERT sentence embeddings in contrast to other state-of-the-art sentence embedding methods.

2 Related Work

We first introduce BERT, then, we discuss state-of-the-art sentence embedding methods.

BERT (Devlin et al., 2018) is a pre-trained transformer network (Vaswani et al., 2017), which set for various NLP tasks new state-of-the-art results, including question answering, sentence classification, and sentence-pair regression. The input for BERT for sentence-pair regression consists of the two sentences, separated by a special [SEP] token. Multi-head attention over 12 (base-model) or 24 layers (large-model) is applied and the output is passed to a simple regression function to derive the final label. Using this setup, BERT set a

new state-of-the-art performance on the Semantic Textual Similarity (STS) benchmark (Cer et al., 2017). RoBERTa (Liu et al., 2019) showed, that the performance of BERT can further improved by small adaptations to the pre-training process. We also tested XLNet (Yang et al., 2019), but it led in general to worse results than BERT.

large disadvantage of the BERT network structure is that no independent sentence embeddings are computed, which makes it difficult to derive sentence embeddings from BERT. To bypass this limitations, researchers passed single sentences through BERT and then derive a fixed sized vector by either averaging the outputs (similar to average word embeddings) or by using the output of the special CLS token (for example: May et al. (2019); Zhang et al. (2019); Qiao et al. (2019)). These two options are also provided by the popular bert-as-a-service-repository³. Up to our knowledge, there is so far no evaluation if these methods lead to useful sentence embeddings.

Sentence embeddings are a well studied area with dozens of proposed methods. Skip-Thought (Kiros et al., 2015) trains an encoder-decoder architecture to predict the surrounding sentences. InferSent (Conneau et al., 2017) uses labeled data of the Stanford Natural Language Inference dataset (Bowman et al., 2015) and the Multi-Genre NLI dataset (Williams et al., 2018) to train a siamese BiLSTM network with max-pooling over the output. Conneau et al. showed, that InferSent consistently outperforms unsupervised methods like SkipThought. Universal Sentence Encoder (Cer et al., 2018) trains a transformer network and augments unsupervised learning with training on SNLI. Hill et al. (2016) showed, that the task on which sentence embeddings are trained significantly impacts their quality. Previous work (Conneau et al., 2017; Cer et al., 2018) found that the SNLI datasets are suitable for training sentence embeddings. Yang et al. (2018) presented a method to train on conversations from Reddit using siamese DNN and siamese transformer networks, which yielded good results on the STS benchmark dataset.

Humeau et al. (2019) addresses the run-time overhead of the cross-encoder from BERT and present a method (poly-encoders) to compute a score between m context vectors and pre-

³<https://github.com/hanxiao/bert-as-service/>