Universal Sentence Encoder

Daniel Cer[a], Yinfei Yang[a], Sheng-yi Kong[a], Nan Hua[a], Nicole Limtiaco[b], Rhomni St. John[a], Noah Constant[a], Mario Guajardo-Céspedes[a], Steve Yuan[c], Chris Tar[a], Yun-Hsuan Sung[a], Brian Strope[a], Ray Kurzweil[a]

[a]Google Research
Mountain View, CA

[b]Google Research
New York, NY

[c]Google
Cambridge, MA

Abstract

We present models for encoding sentences into embedding vectors that specifically target transfer learning to other NLP tasks. The models are efficient and result in accurate performance on diverse transfer tasks. Two variants of the encoding models allow for trade-offs between accuracy and compute resources. For both variants, we investigate and report the relationship between model complexity, resource consumption, the availability of transfer task training data, and task performance. Comparisons are made with baselines that use word level transfer learning via pretrained word embeddings as well as baselines do not use any transfer learning. We find that transfer learning using sentence embeddings tends to outperform word level transfer. With transfer learning via sentence embeddings, we observe surprisingly good performance with minimal amounts of supervised training data for a transfer task. We obtain encouraging results on Word Embedding Association Tests (WEAT) targeted at detecting model bias. Our pre-trained sentence encoding models are made freely available for download and on TF Hub.

1 Introduction

Limited amounts of training data are available for many NLP tasks. This presents a challenge for data hungry deep learning methods. Given the high cost of annotating supervised training data, very large training sets are usually not available for most research or industry NLP tasks. Many models address the problem by implicitly performing limited transfer learning through the use of pre-trained word embeddings such as those produced by word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). However, recent work has demonstrated strong transfer task performance using pre-trained sentence level embeddings (Conneau et al., 2017).

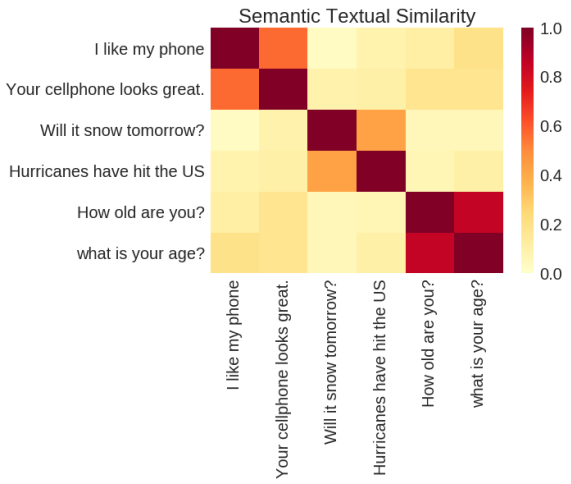

Figure 1: Sentence similarity scores using embeddings from the universal sentence encoder.

In this paper, we present two models for producing sentence embeddings that demonstrate good transfer to a number of other of other NLP tasks. We include experiments with varying amounts of transfer task training data to illustrate the relationship between transfer task performance and training set size. We find that our sentence embeddings can be used to obtain surprisingly good task performance with remarkably little task specific training data. The sentence encoding models are made publicly available on TF Hub.

Engineering characteristics of models used for transfer learning are an important consideration. We discuss modeling trade-offs regarding memory requirements as well as compute time on CPU and GPU. Resource consumption comparisons are made for sentences of varying lengths.

```
import tensorflow_hub as hub

embed = hub.Module("https://tfhub.dev/google/"
    "universal-sentence-encoder/1")

embedding = embed([
    "The quick brown fox jumps over the lazy dog."])
```

Listing 1: Python example code for using the universal sentence encoder.

2 Model Toolkit

We make available two new models for encoding sentences into embedding vectors. One makes use of the transformer (Vaswani et al., 2017) architecture, while the other is formulated as a deep averaging network (DAN) (Iyyer et al., 2015). Both models are implemented in TensorFlow (Abadi et al., 2016) and are available to download from TF Hub:[1]

https://tfhub.dev/google/universal-sentence-encoder/1

The models take as input English strings and produce as output a fixed dimensional embedding representation of the string. Listing 1 provides a minimal code snippet to convert a sentence into a tensor containing its sentence embedding. The `embedding` tensor can be used directly or incorporated into larger model graphs for specific tasks.[2]

As illustrated in Figure 1, the sentence embeddings can be trivially used to compute sentence level semantic similarity scores that achieve excellent performance on the semantic textual similarity (STS) Benchmark (Cer et al., 2017). When included within larger models, the sentence encoding models can be fine tuned for specific tasks using gradient based updates.

3 Encoders

We introduce the model architecture for our two encoding models in this section. Our two encoders have different design goals. One based on the transformer architecture targets high accuracy at the cost of greater model complexity and resource consumption. The other targets efficient inference with slightly reduced accuracy.

[1]The encoding model for the DAN based encoder is already available. The transformer based encoder will be made available at a later point.

[2]Visit https://colab.research.google.com/ to try the code snippet in Listing 1. Example code and documentation is available on the universal encoder website provided above.

3.1 Transformer

The transformer based sentence encoding model constructs sentence embeddings using the encoding sub-graph of the transformer architecture (Vaswani et al., 2017). This sub-graph uses attention to compute context aware representations of words in a sentence that take into account both the ordering and identity of all the other words. The context aware word representations are converted to a fixed length sentence encoding vector by computing the element-wise sum of the representations at each word position.[3] The encoder takes as input a lowercased PTB tokenized string and outputs a 512 dimensional vector as the sentence embedding.

The encoding model is designed to be as general purpose as possible. This is accomplished by using multi-task learning whereby a single encoding model is used to feed multiple downstream tasks. The supported tasks include: a Skip-Thought like task (Kiros et al., 2015) for the unsupervised learning from arbitrary running text; a conversational input-response task for the inclusion of parsed conversational data (Henderson et al., 2017); and classification tasks for training on supervised data. The Skip-Thought task replaces the LSTM (Hochreiter and Schmidhuber, 1997) used in the original formulation with a model based on the Transformer architecture.

As will be shown in the experimental results below, the transformer based encoder achieves the best overall transfer task performance. However, this comes at the cost of compute time and memory usage scaling dramatically with sentence length.

3.2 Deep Averaging Network (DAN)

The second encoding model makes use of a deep averaging network (DAN) (Iyyer et al., 2015) whereby input embeddings for words and bi-grams are first averaged together and then passed through a feedforward deep neural network (DNN) to produce sentence embeddings. Similar to the Transformer encoder, the DAN encoder takes as input a lowercased PTB tokenized string and outputs a 512 dimensional sentence embedding. The DAN encoder is trained similarly to the Transformer based encoder. We make use of mul-

[3]We then divide by the square root of the length of the sentence so that the differences between short sentences are not dominated by sentence length effects