Text classification with transformers in TensorFlow 2

David Mraz

Atheros.ai

david@atheros.ai

April 15, 2020

Overview

1. Text Classification
2. Why transformers?
3. BERT
4. Practical Part

Problem formulation - Text Classification

Dataset D contains sequences of text examples:

$$D = X_1, X_2, \ldots, X_N, \tag{1}$$

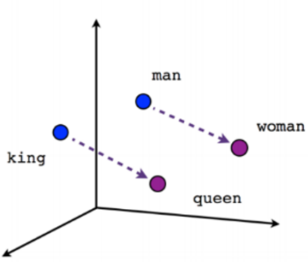
where X_i is i*th example (i.e., document, text segment) and N is the number of documents. We want to label them from the set of all labels k.

- single-label (multi-class) classification
- multi-label classification

Applications: news categories, film review sentiment, tagging content

What are word embeddings?

Word embedding - set of language modeling and feature learning techniques. Words from vocabulary are mapped to vectors to capture relationships etc. e.g. Word2Vec, Glove, ELMO, BERT embeddings



Male-Female

What is language model?

A **language model** is a probability distribution over sequences of words. It assigns a probability for a sequence of words

$$P(w_1, \cdots .w_n)$$

Neural net language models
Learn to predict next word in the sequence based on the context

$$P(w_i|\mathrm{context})$$

Modern approaches to NLP tasks

- RNN (Recurrent neural network) (vanishing gradient problem)
- LSTM (Long short-term memory) (Hochreiter *et al.*, 1997) (capture longer context)
- Bi-LSTM (process context in both directions)
- Transformers (drop LSTM $\rightarrow$ attention (Vaswani *et al.*, 2017))

Transformer approach

- attention seeing entire sequence as a whole
- much easier to train in paralell
- unsupervised pretraining then transfer learning
- text classification, question answering, machine translation etc.
- GPT, BERT, GPT-2, XLNet, Megatron, Turing-NLG

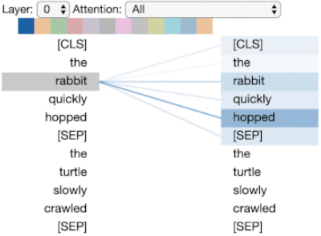


Figure: https://github.com/jessevig/bertviz

What is BERT?

- Bidirectional Encoder Representations from Transformers (Devlin *et al.*, 2018)
- method of pretraining language representation
- transformer based architecture (with slight differences)
- WordPiece embeddings
- you can fine-tune such model on a specific task
- classification, named-entity recognition, question answering etc.
- state of the art results on a number of NLP tasks at that time

BERT Tokenizer

- WordPiece embeddings (subword tokenization)(Schuster *et al.*, 2012)
- BERT input is constrained to 512 tokens
- special tokens [CLS], [SEP], [PAD] tokens
- positional embeddings, segment embeddings, token embeddings

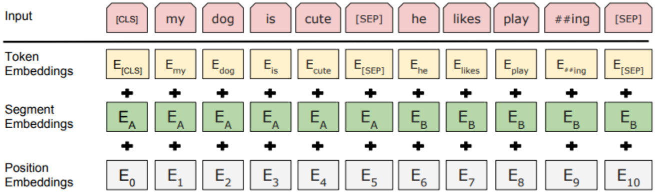
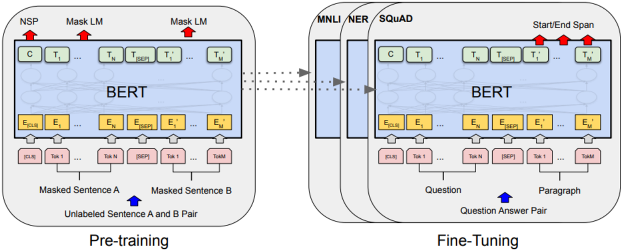


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

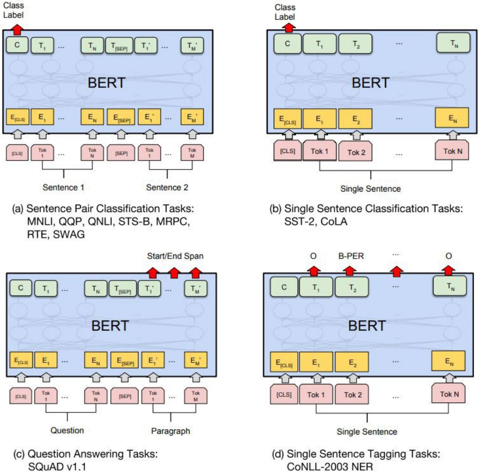
Pretraining (BERT)

- BooksCorpus (800m words), English Wikipedia (2500M words)
- masked language modelling (MLM), next sentence prediction (NSP)
- pretraining is expensive, load already pretrained models - BERT (base), BERT (large)
- leverage transfer learning by pretraining just once and then fine-tune on specific tasks



BERT fine-tunning

Load the pretrained model and add task specific layer



IMDB examples

Review	**Sentiment**
Probably my all-time favorite movie, a story of selflessness, sacrifice and dedication to a noble cause,	Positive
Encouraged by the positive comments about this film on here I was looking forward to watching this film. Bad mistake. I've seen 950+ films and this is truly one of the worst of them...	Negative

Table: Examples in IMDB dataset

State of the art results on IMDB dataset

Model	Accuracy	Paper/Source
XLNet	96.21	(Yang *et al.*, 2019)
$BERT_{large}ITPT - FiT$	95.79	(Sun *et al.*, 2019)
$BERT_{base}ITPT - FiT$	95.63	(Sun *et al.*, 2019)
ULMFiT	95.4	(Howard *et al.*, 2018)
Block-sparse LSTM	94.99	(Gray *et al.*, 2017)

Table: Performance on IMDB review dataset (nlpprogress.com)

IMDB sentiment analysis with BERT (practical part)

- installation (we will use Google Colab)
- load IMDB using TensorFlow datasets
- BERT tokenizer
- load pretrained BERT model (transformers library)
- compile model choose loss, optimizer etc.
- fine-tunning the model
- evaluate the model

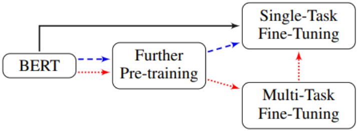


Figure 1: Three general ways for fine-tuning BERT, shown with different colors.

Google Colab

References I

- S. Hochreiter *et al.*, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- A. Vaswani *et al.*, *Attention is all you need*, 2017. arXiv: 1706.03762 [cs.CL].
- J. Devlin *et al.*, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2018. arXiv: 1810.04805 [cs.CL].
- M. Schuster *et al.*, "Japanese and korean voice search," in *International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 5149–5152.
- Z. Yang *et al.*, *Xlnet: Generalized autoregressive pretraining for language understanding*, 2019. arXiv: 1906.08237 [cs.CL].
- C. Sun *et al.*, *How to fine-tune bert for text classification?* 2019. arXiv: 1905.05583 [cs.CL].

References II

J. Howard *et al.*, *Universal language model fine-tuning for text classification*, 2018. arXiv: 1801.06146 [cs.CL].

S. Gray *et al.*, "Gpu kernels for block-sparse weights," , 2017.