Text Classification

When the dataset D contains sequences of text examples"

D=a, b, c, d, e, f

Where a is "I" example and c is i+1; N is the number of documents; accordingly, f is the N-1 item. We want to label them from the set of all labels k. There are two ways of classification as below:

1. **Single-label classification or multiclass**
2. **Multi-label classification**

Application: news categories, film review sentiment and tagging content.

Word embedding: set of language modeling and feature learning techniques. Words from vocabularies are mapped into vectors to capture relationships etc. Word2Vec, Bert, etc.

Language Model: a language model computes a probability for a sequence of words.
P (w1, w2 …., wn)
Neural net language model: Learn to predict next word in the sequence based on the context.

P (w |context)

Modern approaches to NLP task: RNN (Vanishing Gradient Problem), LSTM (Capturing long context), Bi-LSTM (processing context in both direction), Transformers (using attention).

Transformers:

- Attention layer see entire sequence as a whole.
- Much easier to train in parallel
- Unsupervised pretraining then transfer learning
- Text Classification, Question Answering, Machine Translation etc.
- GPT, BERT, GPT-2, XLNet, Megtron, Turing-NLG

Bert:

- Bidirectional Encoder Representations from Transformers
- Method of pretraining language representation
- Transformer based architecture (with slight differences)
- Word-Piece embedding
- You can fine-tune such model on a specific task
- Classification, Name Entity Recognition, Question Answering
- State of the art results on a number of NLP tasks at that time

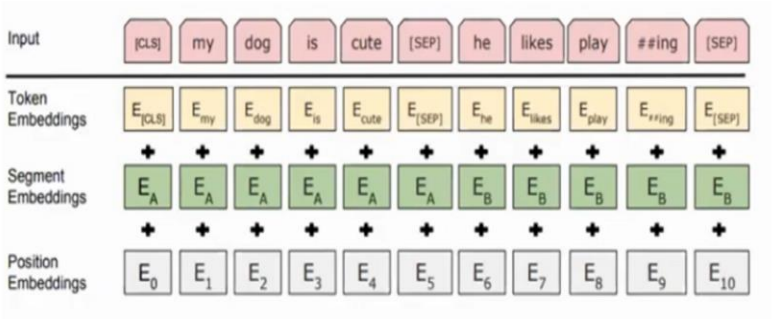
Bert Tokenizer

- Word-Piece embedding (Sub-word Tokenization)
- Bert input is constraint to 512 tokens

- Special tokens [CLS], [SEP], [PAD] tokens
- Positional Embeddings, Segment Embeddings, Token Embeddings; it is worth mentioning that the input embeddings are the sum of the Token, Segment and positional embeddings.

Bert leverage transfer learning happens at two levels; first is pretraining and second is fine-tuning. For fine running stage, Bert load the pretrained model and add task specific layer. Fine tuning may happens for the tasks below:

- Sentence Pair Classification Task
- Single Sentence Classification Task
- Question Answering Task
- Single Sentence Tagging Task



Let's Start with Bert:

- Installation (We will use google Collab)
- Load your Data using Tensor Flow Dataset
- BERT Tokenizer
- Load pretrained BERT model (Transformers library)
- Compile model choose loss, optimizer etc.
- Fine-tunning the model

Follow the instruction in PDF " BERT in action".

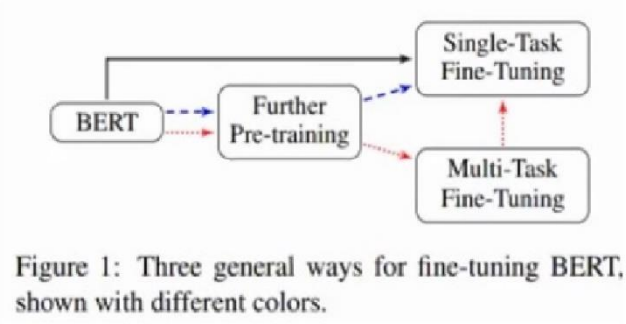


Figure 1: Three general ways for fine-tuning BERT, shown with different colors.