# BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

The stated purpose of BERT model, is to train deep bi-directional representations of unlabelled text, conditioned on **left and right** context. It can be further fine-tuned by adding an additional layer.
There are few datasets with good quality annotations, however, there is an abundance of unlabelled text. The BERT models are pre-trained on this unsupervised data using a **Masked Language Modelling** objective. This model can then be applied to other tasks by **transfer learning**.
There are two approaches to apply pre-trained embeddings for down-stream tasks, *feature based* and *fine-tuning*. The major limitations of these approaches are that standard language models are uni-directional which limit the choice of architectures that can be used during pre-training.

## Related Work

**ELMo**: ELMo uses deep bi-directional LSTMs, to generate contextualized word embeddings. At each layer, the output of the left-to-right hidden representation and right-to-left representations are concatenated to obtain the contextualized embedding. ELMo advanced the state-of-the-art on multiple NLP tasks.
**Open AI GPT**: Fine-tuning approach — ALl the model parameters are fine-tuned using uni-directional language models on the down-stream tasks.

## Training

1. **Masked Language Modelling**: 15% for the words chosen randomly in the input sentence is masked using the `[MASK]` token. The model then aims to predice the vocabulary id of the masked token. The MLM objective enables the representation to use both the left and right context. Additionally, BERT uses a next sentence prediction task. BERT is trained in two stages, *pre-training* and *fine-tuining*. In *pre-training*, the model is trained on un-labeled data, while in *fine-tuning*, all of the pre-trained parameters are fine-tuned using labeled data. BERT has a unified architecture across all the tasks.

2. **Tokenization**: A sentence is first tokenized, a `[CLS]` token is added to the start of the sentence. Multipel sentences are separated by the `[SEP]` token. BERT uses a concatenation of WordPiece, Positional and Segment embeddings. The segment embeddings (attenstion mask) represents which sentence the particular token belongs to.

3. **Architecture**: BERT uses stacked encoder-decoder Transformer blocks. where each block has multiple self-attention heads. For *Sequence classification* tasks, the hidden state corresponding to the `[CLS]` token is used.