**Paper Link**

<https://arxiv.org/pdf/1810.04805.pdf>

**Video Summary**

Link to summary paper here

**Pre-requisite papers (Roots)**

Attention Is All You need

BERT pretrains bidirectional representations from unlabeled text using both the left and right context, so that it can be fine-tuned with just one additional layer for multiple output tasks

**Notes**

BERT has a pre-training and fine-tuning stage, with pre-training done in an unsupervised manner and fine-tuning down on a downstream (focused, supervised) task

BERT uses a multi-layer bidirectional Transformer architecture, nearly identical to the original

The # of layers is denoted as L, hidden size as H, # of self attention heads as A

BERT transformer uses bidirectional attention (except when generating output) while GPT transformer uses constrained self-attention where every token can only attend to context on its left

For a token, its input representation is created by summing the token, position, and segment embeddings

Standard conditional language models for unsupervised tasks cannot be trained bidirectionally since that would allow each word to "see itself"

For the unsupervised task, 15% of the input tokens are randomly masked and their hidden vectors are fed into an output layer to generate the predictions for those words

We create a mismatch between pre-training and fine-tuning, since the fine-tuning supervised stage will not have [MASK] token

* To mitigate this, 10% of the time [MASK] is replaced with a random token and 10% of the time [MASK] is replaced with the unchanged i-th token
* We then use the input representation (a function of the embedding, position, and context) and get the final hidden state Ti which we use to predict the original token
* The parameters and how they work with the context from the previous layers help determine whether the correct output is predicted — if it is not, then the parameters are changed so the information flowing in from the previous layer when predicting the [MASK] token is useful

The NSP task pre-trains for a next sentence task prediction, where 50% of the time B is the actual next sentence which follows A and 50% of the time it is a random sentence from the corpus (binary classification task)

We generate C for next sentence prediction which is a meaningful representation without fine-tuning

BERT transfers all parameters to initialize end-task model parameters

When fine-tuning BERT on tasks involving pairs (e.g. QA), a common practice is to independently encode text pairs before applying bidirectional cross attention

BERT uses the self-attention mechanism to unify these two stages, since encoding a concatenated text pair w/ self attention effectively includes bidirectional cross attention

**Branches**

How information from this paper may be used in other research papers

This is responsible for the bottom up top down architecture with soft attention

The context and hidden state from both the front and back are used

This paper makes it seem like a bidirectional architecture can be used for the second pair

BERT uses encoded concatenated text pairs with self-attention, which effectively includes bidirectional cross attention. The reason we can use bidirectional cross attention is because we are pre-training the model and we are not actually generating the second text pair, but trying to understand the relationship between the first and second text pair (whether the second text pair follows from the first). Therefore, since we are given the second text pair, it is okay to use it bidirectionally in order to get a stronger representation for our output task.