CS 410: Tech Review

Nisarg Bipinchandra Mistry
nmistry2@illinois.edu

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

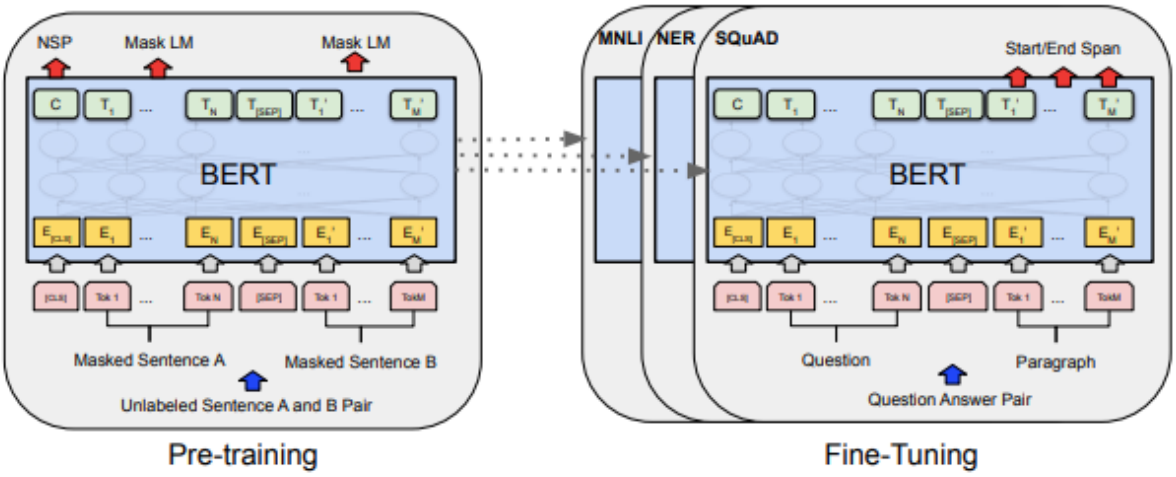
This technology review provides a summary of the "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" paper and also provides the BERT model's applications in information retrieval and information systems.

Before BERT(Bidirectional Encoder Representations from Transformers), the pre-trained models had limitations like parameter tuning as the model architecture and the model being unidirectional (feed-forward). However, the BERT model uses a novel architecture for language representation that trains on unlabeled text data by performing computations on left context and right context in all layers of the architecture. The model can also be fine-tuned with new output layers based on use cases, and hence, the model can be used for various different NLP (Natural language processing) tasks. In technology review explains the model architecture and the steps for performing the pre-training and parameter tuning for this model.

The main distinction between BERT and other NLP model architectures is the unified architecture which is used across many different tasks. The difference between its pre-trained architecture and its final downstream architecture is very small. BERT is first pre-trained over unlabeled datasets over multiple pre-training tasks. Subsequently, for any task for which model needs to be used, it is fine-tuned by initializing the BERT model using the weights/parameters using the pre-trained parameters for each task. This pre-training of model and then using the learned weights as initialization to varied tasks makes the model perform very well on variety of problem statements and the final model is then targeted towards the problem statement. The input for pre-training consists of a single sentence and a pair of sentences forming a question and answer relationship. A sentence is composed of tokenization for each token vocabulary calculated from the WordPiece embeddings So, the input is the sum of these tokenizations of (sentence, answer sentences). This will further assist us when undergoing pre-training which combines left to right as well as right to left language models for unsupervised tasks.

For pre-training use two unsupervised tasks to complete this step, Masked-LM and Next Sentence Prediction(NSP). Most models either take a left to right or a right to left model approach so the way they circumvent this is using Masked LM. This allows us to mask some percentage of input tokens in order to trivially predict the target word. We don't always replace the masked word with a mask token, instead we designate the mask token 80% of the time with 10% being a random token and the remaining percentage left unchanged. Second task we use for pre-training is NSP. NSP allows us to understand the relationship between two sentences by 50% of the time we label the actual next sentence and the other 50% we label a random sentence from the corpus. Each BERT implementation

takes 4 arguments; L is the number of layers, H is the hidden size, A is the self attention heads and the last argument is the total parameter count. The paper specifies two BERT instantiation that they tested against similar models in this subject space, BERT_BASE (L=12, H=768, A=12, Total Parameters=110M) and BERT_LARGE (L=24, H=1024, A=16, Total Parameters=340M).



Fine-tuning is simpler compared to other language models because BERT allows us to model several downstream tasks. BERT includes “bidirectional cross attention” between two sentences instead of handling these tasks independently. For each downstream task, we plug in the input and output into BERT to fine tune the parameters end-to-end. This allows fine-tuning to be relatively inexpensive time wise.

Each BERT implementation takes 4 arguments; L is the number of layers, H is the hidden size, A is the self attention heads and the last argument is the total parameter count. The paper specifies two BERT instantiation that they tested against similar models in this subject space, BERT_BASE (L=12, H=768, A=12, Total Parameters=110M) and BERT_LARGE (L=24, H=1024, A=16, Total Parameters=340M). Both BERT_BASE and BERT_Large were put up against multiple earlier language models and evaluated against 11 tasks; MNLI-, QQP, QNLI, SST-2, CoLA, STS-B, MRPC, RTE, SQuAD v1.1, SQuAD v2.0, and SWAG. They performed significantly better than the competition with BERT_LARGE actually performing the best amongst the collective of models. From this analysis we see that we BERT is a logical choice for many NLP tasks such as Question Answering and language inference. Also from this we see that increasing BERTs model size is actually very positively correlated to the success of the models analysis.

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

The paper also talks about multiple ablation experiments over a number of facets of BERT in order to better understand their relative importance. The paper talks about the following ablations made to the experiment however we won't go into details of these in this technology review.

1. Effect of pre-training tasks
2. Effect of model size
3. Feature-based approach with BERT

All in all, this paper shows us the development of BERT as a language model and the importance/significance of its pre training and fine tuning for model preparation. We see that this model is highly scalable for many NLP task usage and performed well in comparison to other models previously used. We also see that this model is relatively inexpensive and more efficient in order to achieve state of the art performances.