BERT model

26 December 2023 09:30

BERT - Bi-directional encoder representation from Transformers

i) Tokenization and encoding
   a. Split the words into sub-words/workpieces
   b. Adds Tokens like
      i. [CLS] - Used for classification
      ii. [SEP] - separator that makes boundaries between inputs
      iii. [MASK] - to hide certain tokens

ii) Key steps
   a. Tokenization
   b. Positional Embedding - Tokens assigned numerical representation
   c. Bi-directional Encoding - Tokens fed into multiple layers of transformer encoding
   d. Attention Mechanism - Used to selectively focus on different parts of the i/p
   e. Simultaneous processing - A sentence is processed from both sides
   f. Refined Representation

iii) Benefits of Bi-directional
   a. Better Accuracy
   b. Lesser ambiguity
   c. Long distance relationships - If the words are far apart they can learn the relationships

iv) What is the goal of the transform encoder
   a. To get a rich contextual representation of the sentence.

v) What is attention mechanism??

BERT - https://arxiv.org/pdf/1810.04805.pdf (Paper)

- 2 strategies to apply language representation to down-stream tasks
  - Feature based - ELMo uses a specific architecture related to the task being accomplished and uses pre-trained representation as additional feature
  - Fine tuning - Such as the OpenAI GPT introduces task specific parameters and is trained on the downstream tasks by fine tuning all the pre-trained parameters
  - **Limitation of the both is that they use Unidirectional language models**
  - Unidirectional models can perform bad in scenarios where there question answering and we need to understand the long distance relationships

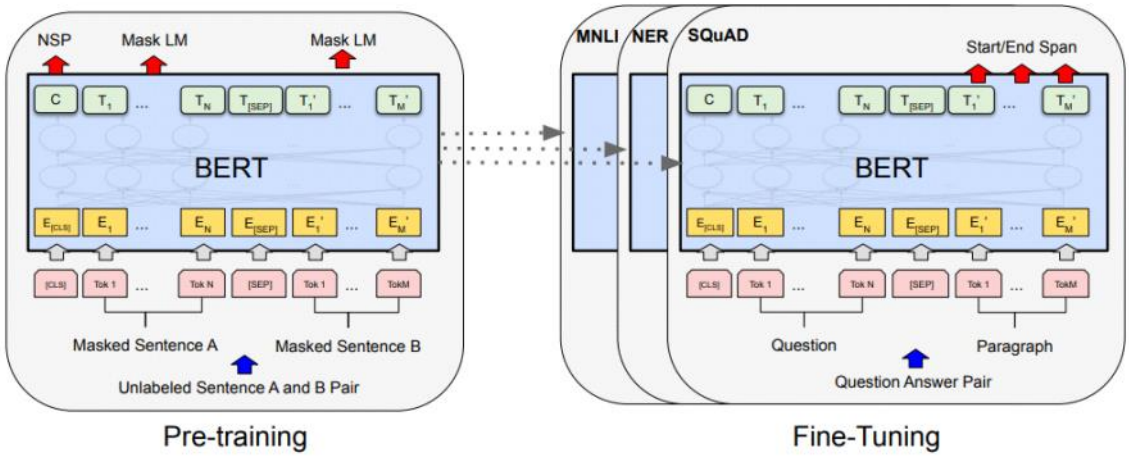

Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

USE of MLM

- Randomly masks some of the tokens from the input and the objective is to predict the original vocabulary based only off of its context
- MLM allows it to fuse both left and right context where as other language models used only shallow concatenation of independently trained left to right and right to left LMs
- Pre-trained representation reduces the need for heavily engineered task specific architecture

Pre-training BERT

- **Task #1 : MaskedLM**
  - We mask some percentage of the input tokens at random and then predict those masked tokens
  - the final hidden vectors corresponding to the mask tokens are fed into **softmax** over the vocabulary (probabilities of the final hidden tokens are considered)
  - to use the masks during the fine tuning process as well we do not always replace the masks with the mask tokens
  - 15% of the positions are masked
    - for ith token we replace it with the mask token 80% of the time
    - a random token 10% of the time
    - the same unchanged token 10% of the time
- **Task #2: Next Sentence Prediction**
  - Understanding the relation between 2 different sentences which is directly not captured by language modelling
  - To accomplish this we pre-train for a **binarized** next sentence prediction
  - When choosing 2 examples for pre-training (A and B)
    - 50% of the time B is the actual next statement of A **(IsNext)**
    - 50% of the time B is the not next statement of A **(NotNext)**

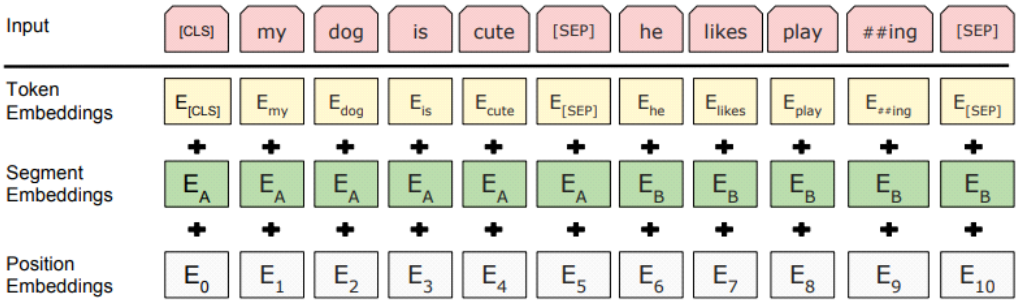

Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

Fine Tuning BERT

- Due to the presence of the attention mechanism in the transformer it allows BERT to model many downstream tasks . by swapping out appropriate inputs and outputs
- Depending on the task we pass task specific inputs and outputs in the BERT and fine tune all the parameters end to end
- Compared to pre-training, fine tuning is inexpensive

Even though MLM's converge slowly when compared to a left-to-right model (which predicts every token) but the improvements in the MLM model outweigh the increased training cost