B.E.R.T: Bidirectional Encoder Representations from Transformers

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova

Background

Before BERT, language models majorly processed text in a unidirectional manner that is either from left to right or right to left.

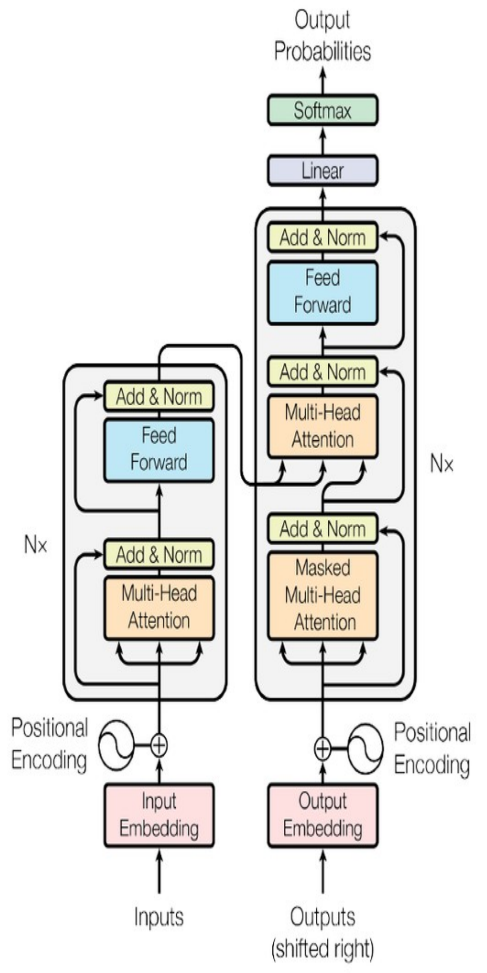
- Transformer
- GPT (Generative Pre-Trained Transformer)

Transformer Architecture

1. Self-Attention Mechanism: Self-attention allows the model to weigh the importance of different words in a sequence relative to each other, capturing dependencies regardless of their distance in the text.
2. Encoder-Decoder Structure: The Transformer consists of an encoder that is designed to process the input and distill it down to its meaning (independent of the language) and a decoder that generates the output sequence, based on the representation receeived.

Left-to-Right Approach in Early Models

3. Unidirectional Context: Models like GPT process input text in a left-to-right fashion. At each position, the model can only attend to the tokens that come before it.



Overview

BERT, a novel language representation model that revolutionized the Natural Language Processing (NLP) domain. Unlike previous models that read text input in one direction, BERT reads the entire sequence of words simultaneously in both directions (left-to-right and right-to-left). This bidirectional approach allows BERT to have a deeper sense of language context and flow than single-direction models.

1. Input/Output Representations: To make BERT handle a variety of down-stream tasks, our input representation is able to unambiguously represent both a single sentence and a pair of sentences (e.g., h Question, Answer) in one token sequence.

2. Masked Language Modeling (MLM): A task where random words are masked, and the model predicts them based on context.

3. Next Sentence Prediction (NSP): A binary classification task where the model predicts if a sentence follows another in a text.

Working

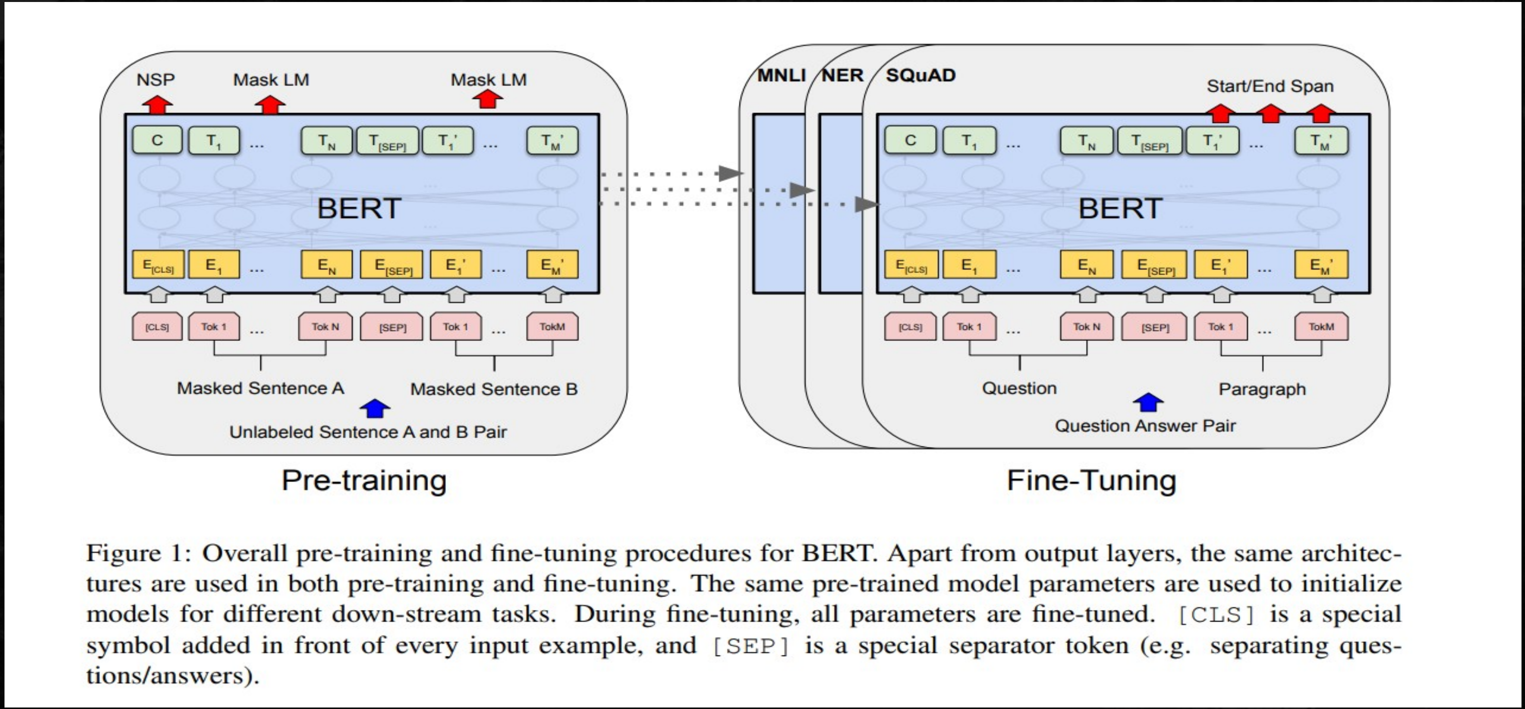


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are 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).

STRENGTHS

- ***Innovative Bidirectional Training:*** By reading text in both directions, BERT is able to understand the semantic of the context on larger scale.

- ***Versatility Across Tasks***: The model is designed to allow, it, to be fine-tuned for a wide range of NLP tasks with minimal architecture modifications.

- ***Language Representation***: Semantic search uses embeddings to compare results to queries, rather than doing simple word matching.

- Parameter Sharing: BERT share parameters between all the layers, allowing it to transfer information between layers during Pre-Training and Fine-Tunnning.

- ***SOTA Results:*** Demonstrated significant performance improvements over previous models on multiple benchmarks such as SQuad, MNLI.

WEAKNESS

- ***Computational Intensity:*** BERT requires large computational resources for pre-training, which act as a barrier for researchers with limited access to high-performance hardware.
- ***Large Model Size:*** The extensive parameters make deployment on edge devices or devices with limited memory very difficult.
- ***Next Sentence Prediction Critique:*** Subsequent research suggested that the NSP task may not significantly contribute to downstream performance and could be optimized or replaced.
- ***Context Length Limitation:*** BERT has a maximum input length (typically 512 tokens), which can be restrictive for longer texts such as big stories or long documents.
- ***Biases:*** BERT model can be biased, if the training of the model is done on the biased data. So, one has to take care of that too.

Relation to Story Generation and Interactive Fiction

- ***Enhanced Language Understanding:*** BERT's deep understanding of context can help in creating and improving the narrative logic in story generation.

- ***Character and Dialogue Consistency:*** In interactive fiction, BERT can help maintain the balance between the character and dialogues by making sure that they both are contextually appropriate.

- ***User Input Interpretation:*** BERT can help in understanding the player commands and inputs more clearly, leading to more accurate and satisfying responses.

- ***Story Generation:*** BERT can help in creating plausible side quests for the player, making the game more engaging and meaningful to the user.

- **Hybrid Model:** Since BERT can’t directly be used as the generative model, but we can use the BERT for understanding the context ad guiding the generation model to produce actual text.

Thank You!!!!

Any Questions