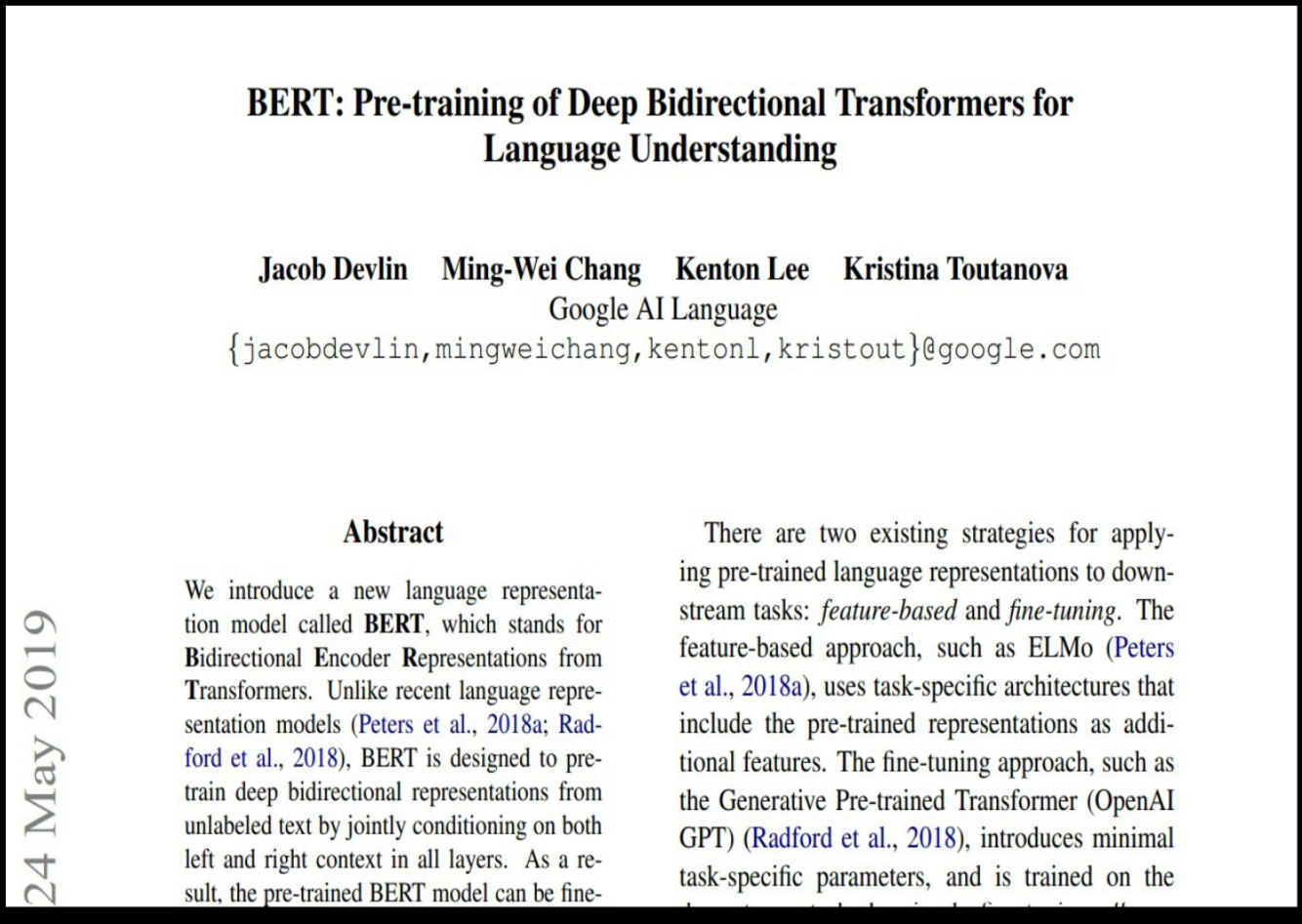
Bidirectional Encoder Representations from Transformers(BERT)

BERT is a pre-trained transformer model developed by Google that is designed to understand the context of a word in search queries and other text.



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

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

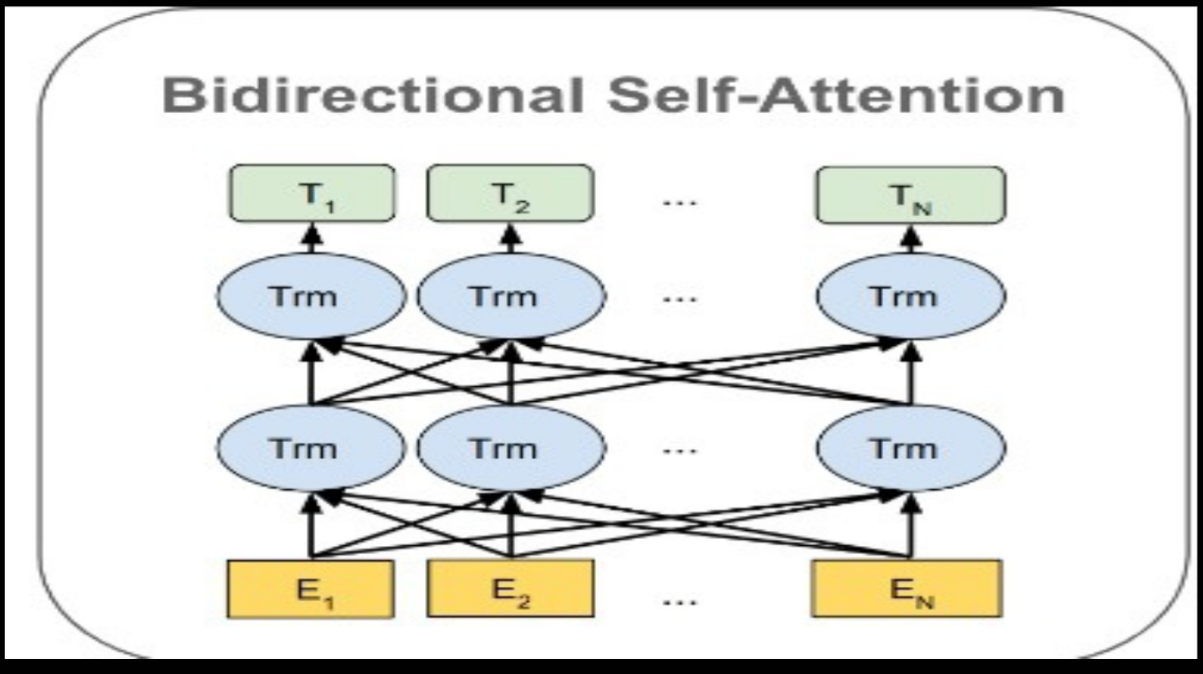
We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the

Key Feartures of BERT

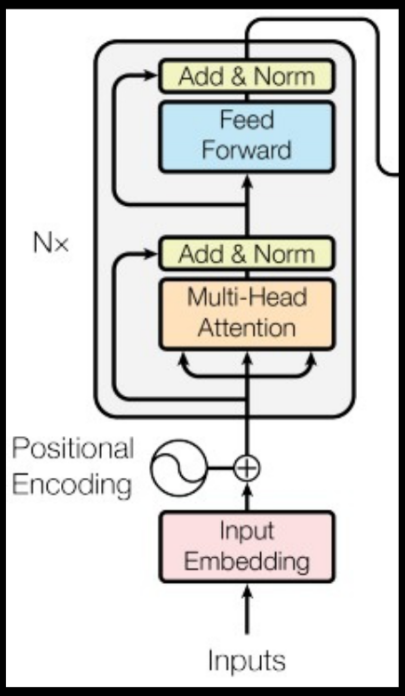
1. Bidirectional Contextual Understanding:

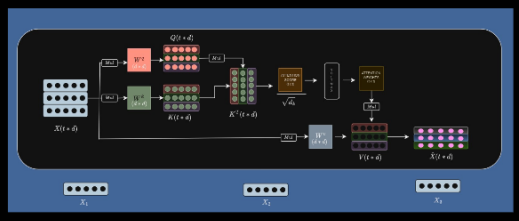
BERT reads text bidirectionally, meaning it looks at both the left and right context when processing a word. This approach allows for a deeper understanding of word meaning based on its context.

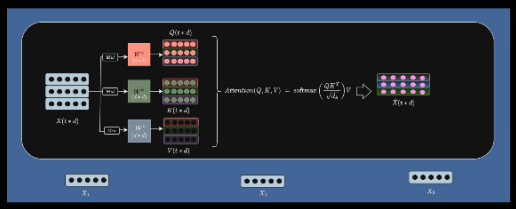


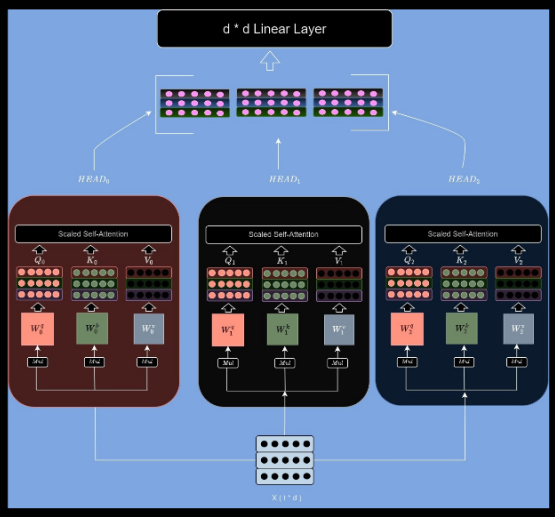
2. Transformer Architecture:

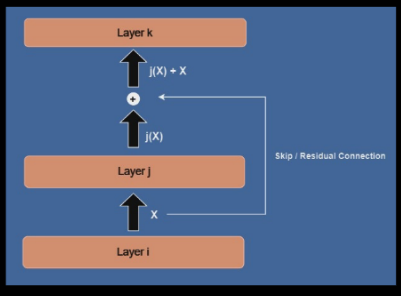
BERT is based on the Transformer architecture, which uses self-attention mechanisms to process text. This allows the model to capture complex dependencies between words in a sentence.

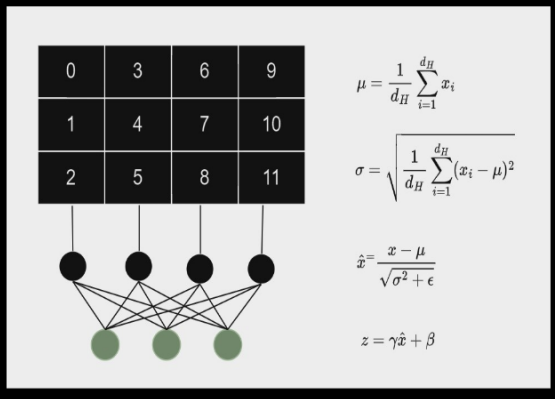


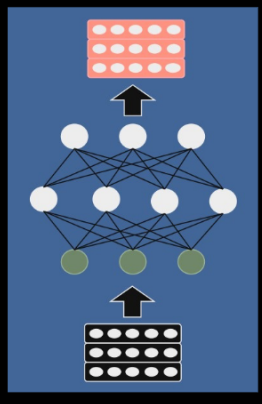


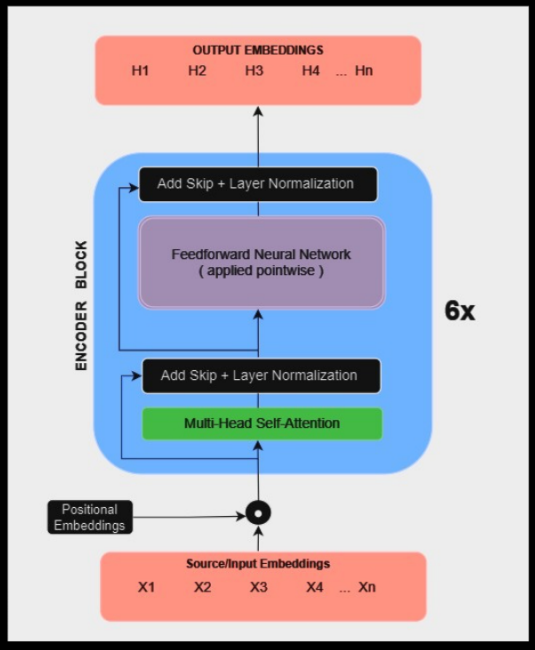












3. Pre-training and Fine-tuning:

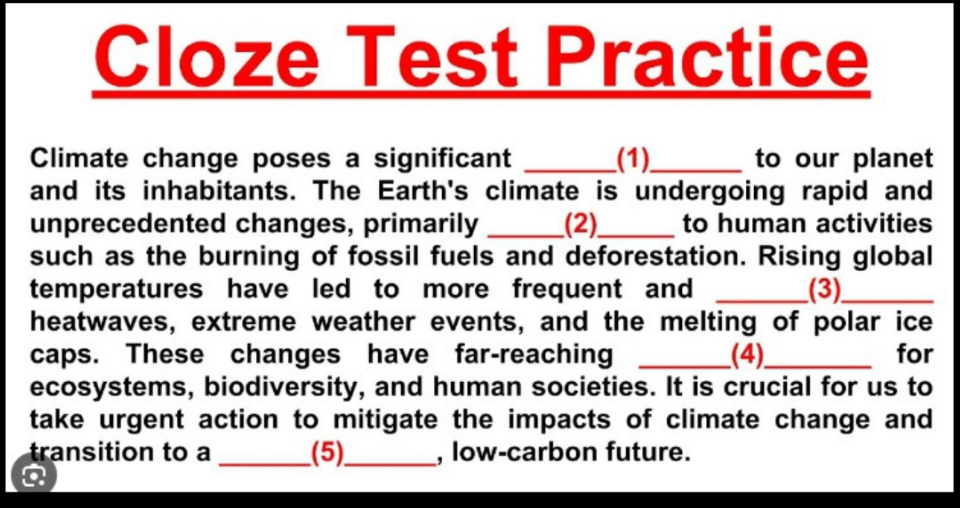
Pre-training: BERT is initially pre-trained on a large corpus of text, such as the BooksCorpus and English Wikipedia, using unsupervised learning tasks.

Fine-tuning: After pre-training, BERT can be fine-tuned on specific downstream tasks (e.g., question answering, sentiment analysis) with labeled data.

Pre-training Tasks

1. Masked Language Model (MLM):

During pre-training, some percentage of the input tokens are masked at random, and BERT is trained to predict these masked tokens based on the surrounding context.This task helps BERT learn bidirectional representations of text.



Cloze Test Practice

Climate change poses a significant _____(1)_____ to our planet and its inhabitants. The Earth's climate is undergoing rapid and unprecedented changes, primarily _____(2)_____ to human activities such as the burning of fossil fuels and deforestation. Rising global temperatures have led to more frequent and _____(3)_____ heatwaves, extreme weather events, and the melting of polar ice caps. These changes have far-reaching _____(4)_____ for ecosystems, biodiversity, and human societies. It is crucial for us to take urgent action to mitigate the impacts of climate change and transition to a _____(5)_____, low-carbon future.

MLM is used to train deep bidirectional representations. Known as the Cloze task in the literature.

Training Procedure:

1. Mask a random percentage of input tokens.
Predict the masked tokens rather than reconstructing the entire input.
2. Typically, 15% of WordPiece tokens in each sequence are masked at random.

Prediction Mechanism:

Final hidden vectors corresponding to the masked tokens are fed into an output softmax over the vocabulary.

Core Problem:

The key issue is that the model's exposure to [MASK] tokens during pre-training does not align with its exposure to real-world text during fine-tuning, where such tokens are absent. This discrepancy can cause the model to perform sub-optimally because the patterns it learned during pre-training (predicting masked tokens) are not directly applicable during fine-tuning (no masked tokens).

Mitigation Strategy:

- Random Selection:

  Randomly select some tokens in the input sequence for. potential masking.Typically, 15% of WordPiece tokens in.

each sequence are masked at random.

- Token Replacement:

- 80% of the time: Replace the selected token with the [MASK] token.
- 10% of the time: Replace the selected token with a random token from the vocabulary.
- 10% of the time: Keep the original token unchanged.

Use the modified sequence (with [MASK] and random tokens) to predict the original tokens. and Compute the loss only for the positions where the tokens were modified.

Example:

Original Sentence: "The quick brown fox jumps over the lazy dog"
Masked Sentence (80% mask, 10% random, 10% unchanged):

- Masked Token: "The [MASK] brown fox jumps over the lazy dog" (80% chance for "quick")
- Random Token: "The xyz brown fox jumps over the lazy dog" (10% chance for "quick")
- Unchanged Token: "The quick brown fox jumps over the lazy dog" (10% chance for "quick")

Next Sentence Prediction (NSP):

BERT is trained to predict whether a given sentence B follows sentence A in the original text. This task helps BERT understand the relationship between sentences.

For each pair of sentences, 50% of the time, B is the actual next sentence (labeled as IsNext), and 50% of the time, B is a random sentence (labeled as NotNext).

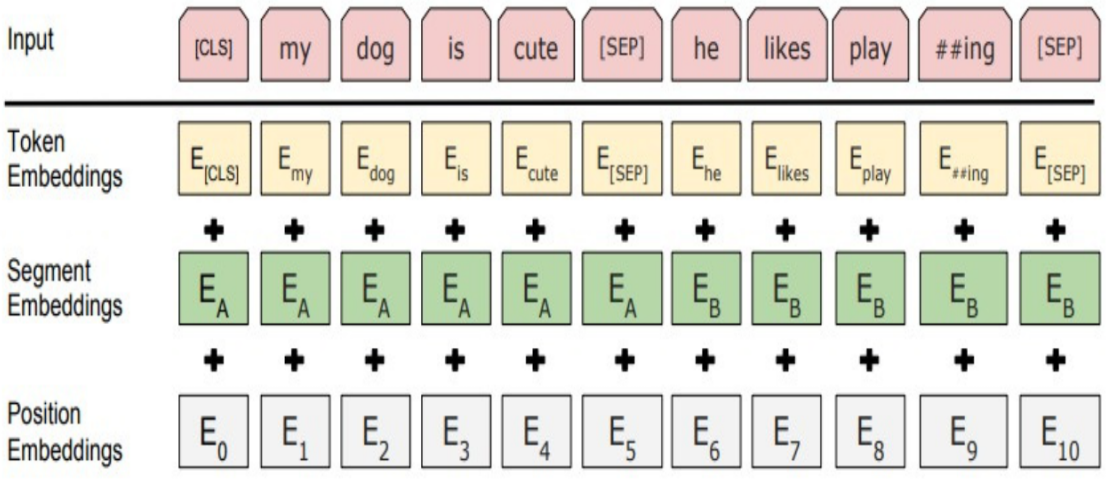


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

Fine-tuning BERT

Task-Specific Data:

Fine-tuning involves training BERT on a specific task with labeled data relevant to that task.

Model Adaptation:

Task-specific layers are added on top of BERT to adapt it to the specific requirements of the downstream task.For example, for a classification task, a classification layer is added on top of the [CLS] token representation.

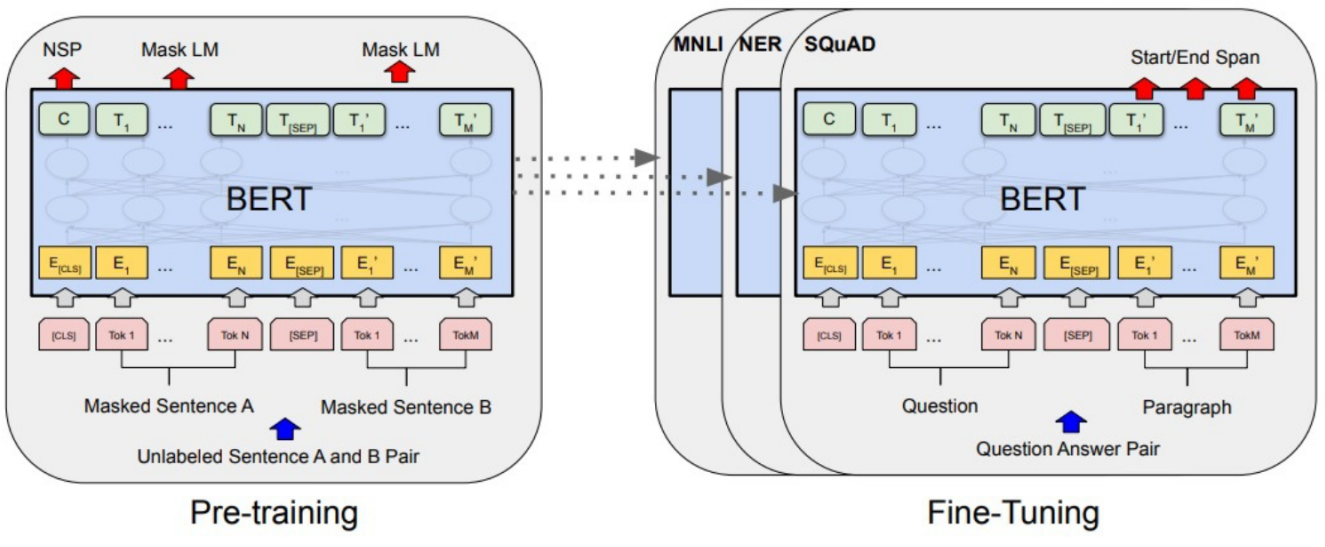


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are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

Training:

BERT is fine-tuned end-to-end, updating the pre-trained weights along with the new task-specific layers. This allows BERT to adapt its learned language representations to the nuances of the specific task.

Applications of BERT

Question Answering: Extracting answers from a passage of text based on a given question.
Text Classification: Categorizing text into predefined labels (e.g., sentiment analysis).
Named Entity Recognition (NER): Identifying and classifying entities (e.g., names, dates) within text.