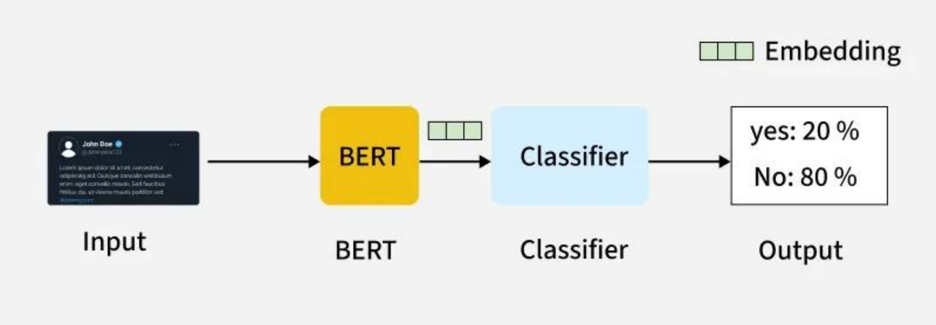
BERT

BERT **(Bidirectional Encoder Representations from Transformers)** stands as an open-source machine learning framework designed for the natural language processing (NLP).



What is BERT?

BERT (Bidirectional Encoder Representations from Transformers) leverages a transformer-based neural network to understand and generate human-like language. BERT employs an encoder-only architecture. In the original **Transformer architecture**, there are both encoder and decoder modules. The decision to use an encoder-only architecture in BERT suggests a primary emphasis on understanding input sequences rather than generating output sequences.

Traditional language models process text sequentially, either from left to right or right to left. This method limits the model's awareness to the immediate context preceding the target word. BERT uses a bi-directional approach considering both the left and right context of words in a sentence, instead of analyzing the text sequentially, BERT looks at all the words in a sentence simultaneously.

Pre-training BERT Model

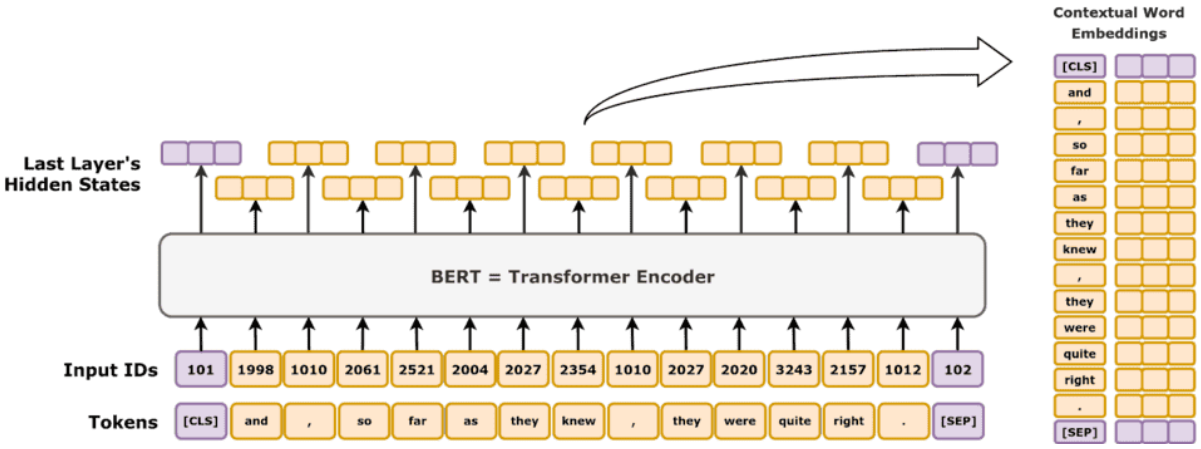
The BERT model undergoes Pre-training on Large amounts of unlabeled text to learn contextual embeddings.

BERT is pre-trained on large amount of unlabeled text data. The model learns contextual embeddings, which are the representations of words that take into account their surrounding context in a sentence.

BERT engages in various unsupervised pre-training tasks. For instance, it might learn to predict missing words in a sentence (Masked Language Model or MLM task), understand the relationship between two sentences, or predict the next sentence in a pair.

Workflow of BERT

BERT is designed to generate a language model so, only the encoder mechanism is used. Sequence of tokens are fed to the Transformer encoder. These tokens are first embedded into vectors and then processed in the neural network. The output is a sequence of vectors, each corresponding to an input token, providing contextualized representations. When training language models, defining a prediction goal is a challenge. Many models predict the next word in a sequence, which is a directional approach and may limit context learning.



BERT addresses this challenge with two innovative training strategies:

Masked Language Model (MLM)

Next Sentence Prediction (NSP)

1. Masked Language Model (MLM)

In BERT's pre-training process, a portion of words in each input sequence is masked and the model is trained to predict the original values of these masked words based on the context provided by the surrounding words.

BERT adds a classification layer on top of the output from the encoder. This layer is important for predicting the masked words.

The output vectors from the classification layer are multiplied by the embedding matrix, transforming them into the vocabulary dimension. This step helps align the predicted representations with the vocabulary space.

The probability of each word in the vocabulary is calculated using the SoftMax activation function. This step generates a probability distribution over the entire vocabulary for each masked position.

The loss function used during training considers only the prediction of the masked values. The model is penalized for the deviation between its predictions and the actual values of the masked words.

The model converges slower than directional models because during training, BERT is only concerned with predicting the masked values, ignoring the prediction of the non-masked words. The increased context awareness achieved through this strategy compensates for the slower convergence.

2. Next Sentence Prediction (NSP)

BERT predicts if the second sentence is connected to the first. This is done by transforming the output of the [CLS] token into a 2×1 shaped vector using a classification layer, and then calculating the probability of whether the second sentence follows the first using SoftMax.

In the training process, BERT learns to understand the relationship between pairs of sentences, predicting if the second sentence follows the first in the original document.

50% of the input pairs have the second sentence as the subsequent sentence in the original document, and the other 50% have a randomly chosen sentence.

To help the model distinguish between connected and disconnected sentence pairs. The input is processed before entering the model.

BERT predicts if the second sentence is connected to the first. This is done by transforming the output of the [CLS] token into a 2×1 shaped vector using a classification layer, and then calculating the probability of whether the second sentence follows the first using SoftMax.

During the training of BERT model, the Masked LM and Next Sentence Prediction are trained together. The model aims to minimize the combined loss function of the Masked LM and Next Sentence Prediction, leading to a robust language model with enhanced capabilities in understanding context within sentences and relationships between sentences.

Why to train Masked LM and Next Sentence Prediction together?

Masked LM helps BERT to understand the context within a sentence and Next Sentence Prediction helps BERT grasp the connection or relationship between pairs of sentences. Hence, training both the strategies together ensures that BERT learns a broad and comprehensive understanding of language, capturing both details within sentences and the flow between sentences.

Fine-Tuning on Labeled Data

We perform Fine-tuning on labeled data for specific NLP tasks.

After the pre-training phase, the BERT model, armed with its contextual embeddings, is fine-tuned for specific natural language processing (NLP) tasks. This step tailors the model to more targeted applications by adapting its general language understanding to the nuances of the particular task.

BERT is fine-tuned using labeled data specific to the downstream tasks of interest. These tasks could include sentiment analysis, question-answering, named entity recognition, or any other NLP application. The model's parameters are adjusted to optimize its performance for the particular requirements of the task at hand.

BERT's unified architecture allows it to adapt to various downstream tasks with minimal modifications, making it a versatile and highly effective tool in natural language understanding and processing.

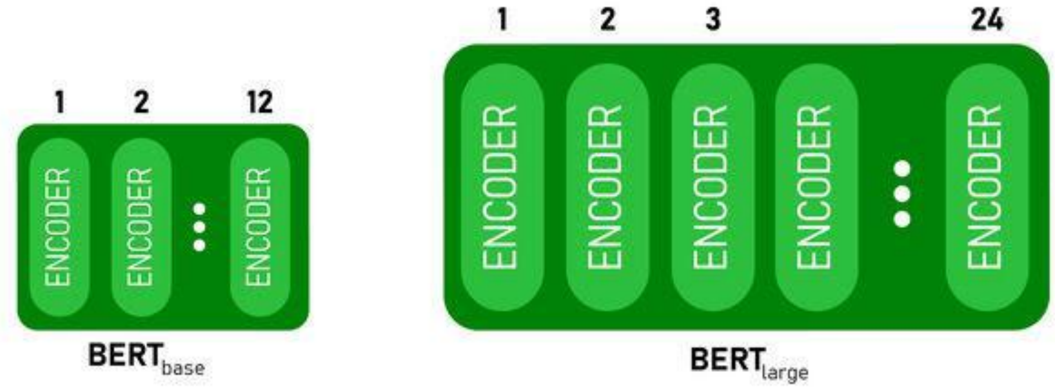
BERT Architecture

The architecture of BERT is a multilayer bidirectional transformer encoder which is quite similar to the transformer model. A transformer architecture is an encoder-decoder network that uses self-attention on the encoder side and attention on the decoder side.

BERTBASE has 12 layers in the Encoder stack while BERTLARGE has 24 layers in the Encoder stack. These are more than the Transformer architecture described in the original paper (6 encoder layers).

BERT architectures (BASE and LARGE) also have larger feedforward networks (768 and 1024 hidden units respectively), and more attention heads (12 and 16 respectively) than the Transformer architecture suggested in the original paper. It contains 512 hidden units and 8 attention heads.

BERTBASE contains 110M parameters while BERTLARGE has 340M parameters.



How to use BERT model in NLP?

BERT can be used for various natural language processing (NLP) tasks such as:

1. Classification Task

- BERT can be used for classification task like sentiment analysis, the goal is to classify the text into different categories (positive/ negative/ neutral), BERT can be employed by adding a classification layer on the top of the Transformer output for the [CLS] token.
- The [CLS] token represents the aggregated information from the entire input sequence. This pooled representation can then be used as input for a classification layer to make predictions for the specific task.

2. Question Answering

- In question answering tasks, where the model is required to locate and mark the answer within a given text sequence, BERT can be trained for this purpose.
- BERT is trained for question answering by learning two additional vectors that mark the beginning and end of the answer. During training, the model is provided with questions and corresponding passages, and it learns to predict the start and end positions of the answer within the passage.

3. Named Entity Recognition (NER)

- BERT can be utilized for NER, where the goal is to identify and classify entities (e.g., Person, Organization, Date) in a text sequence.
- A BERT-based NER model is trained by taking the output vector of each token form the Transformer and feeding it into a classification layer. The layer predicts the named entity label for each token, indicating the type of entity it represents.

Application of BERT

BERT is used for various applications. Some of these are:

1. **Text Representation:** BERT is used to generate word embeddings or representation for words in a sentence.
2. **Named Entity Recognition (NER)**: BERT can be fine-tuned for named entity recognition tasks, where the goal is to identify entities such as names of people, organizations, locations, etc., in a given text.
3. **Text Classification:** BERT is widely used for text classification tasks, including sentiment analysis, spam detection, and topic categorization. It has demonstrated excellent performance in understanding and classifying the context of textual data.

4. **Question-Answering Systems:** BERT has been applied to question-answering systems, where the model is trained to understand the context of a question and provide relevant answers. This is particularly useful for tasks like reading comprehension.

5. **Machine Translation:** BERT's contextual embeddings can be leveraged for improving machine translation systems. The model captures the nuances of language that are crucial for accurate translation.

6. **Text Summarization:** BERT can be used for abstractive text summarization, where the model generates concise and meaningful summaries of longer texts by understanding the context and semantics.

7. **Conversational AI:** BERT is employed in building conversational AI systems, such as chatbots, virtual assistants, and dialogue systems. Its ability to grasp context makes it effective for understanding and generating natural language responses.

8. **Semantic Similarity:** BERT embeddings can be used to measure semantic similarity between sentences or documents. This is valuable in tasks like duplicate detection, paraphrase identification, and information retrieval.