**1. Explain the architecture of BERT**

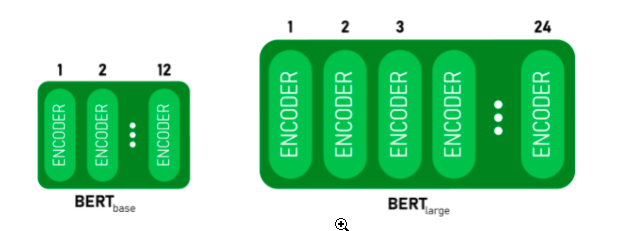
**Ans**: BERT is an open-source machine learning framework for natural language processing (NLP). BERT is designed to help computers understand the meaning of ambiguous language in text by using surrounding text to establish context. The BERT framework was pre-trained using text from Wikipedia and can be fine-tuned with question-and-answer datasets.

BERT, which stands for Bidirectional Encoder Representations from Transformers, is based on Transformers, a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection.

BERT is released in two sizes BERTBASE and BERTLARGE. The BASE model is used to measure the performance of the architecture comparable to another architecture and the LARGE model produces state-of-the-art results that were reported in the research paper.

One of the main reasons for the good performance of BERT on different NLP tasks was the use of Semi-Supervised Learning. This means the model is trained for a specific task that enables it to understand the patterns of the language. After training the model (BERT) has language processing capabilities that can be used to empower other models that we build and train using supervised learning.

BERT is basically an Encoder stack of transformer architecture. 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.

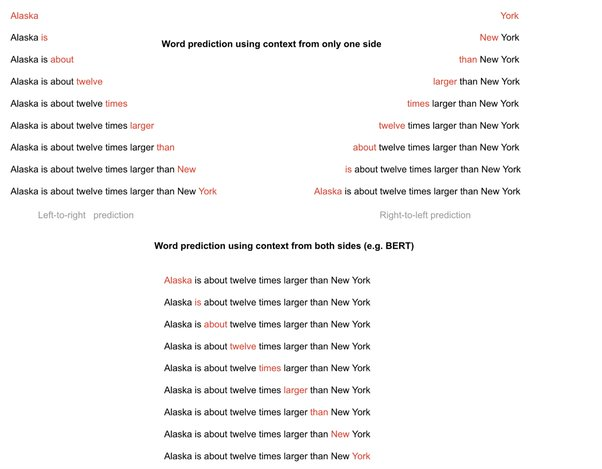


**2. Explain Masked Language Modeling (MLM)?**

**Ans:** Masked language modeling is an example of autoencoding language modeling (the output is reconstructed from corrupted input) - we typically mask one or more of words in a sentence and have the model predict those masked words given the other words in sentence. By training the model with such an objective, it can essentially learn certain (but not all) statistical properties of word sequences.

BERT is a model that is trained on a masked language modeling objective.

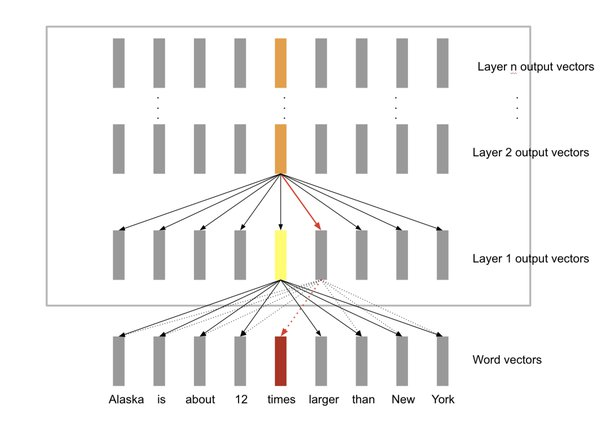
Language modeling approaches shown in figure below.



Language modeling approaches - Autoregressive approach (e.g. left to right prediction, right to left prediction). Masked language approach - Using prediction of a word using all other words in a sentence (the words in red in BERT case is masked out - replaced with a special token [MASK]). BERT masks about 15% of words in a sentence and using the context words to predict it.

Masked language modeling is useful when trying to learn deep representations (that is learning multiple representations of a word using a deep model - these representations have shown to improve performance in downstream tasks. For example lower layer representations of certain models being useful for syntactic tasks whereas higher layer representations for semantic tasks) for a word using words from either side of a word in a sentence (deep and bidirectional representations).

A masked language model is particularly useful for learning deep bidirectional representations because the standard language modeling approach (autoregressive modeling) wont work in a deep model with bidirectional context - the prediction of a word would indirectly see itself making the prediction trivial as shown below (the word “times” can be used in its own prediction from layer 2 onwards. BERT addresses this problem by replacing the word being predicted with a mask token) . However, we could also learn deep bidirectional representations without having to resort to masked language modeling by using the permutation approach of a more recent model - XLNet.



**3. Explain Next Sentence Prediction (NSP)?**

**Ans:** Next sentence prediction (NSP) is one-half of the training process behind the BERT model (the other being masked-language modeling — MLM). Where MLM teaches BERT to understand relationships between words — NSP teaches BERT to understand longer-term dependencies across sentences.

**4. What is Matthews evaluation?**

**Ans:** Matthews defined VM, known as the Matthews coefficient, as the crystal volume per unit of protein molecular weight, and showed that VM bears a straightforward relationship to the fractional volume of solvent in the crystal.

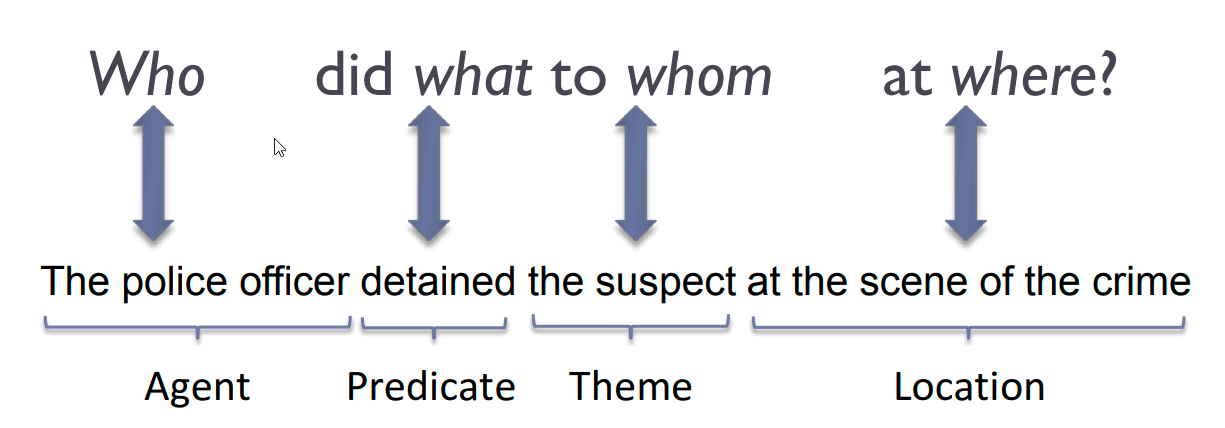
**5. What is Matthews Correlation Coefficient (MCC)?**

**Ans:** Matthew’s correlation coefficient, also abbreviated as MCC was invented by Brian Matthews in 1975. MCC is a statistical tool used for model evaluation. Its job is to gauge or measure the difference between the predicted values and actual values and is equivalent to chi-square statistics for a 2 x 2 contingency table.

**6. Explain Semantic Role Labeling?**

**Ans:** In natural language processing, semantic role labeling (also called shallow semantic parsing or slot-filling) is the process that assigns labels to words or phrases in a sentence that indicates their semantic role in the sentence, such as that of an agent, goal, or result.

It serves to find the meaning of the sentence. To do this, it detects the arguments associated with the predicate or verb of a sentence and how they are classified into their specific roles. A common example is the sentence "Mary sold the book to John." The agent is "Mary," the predicate is "sold" (or rather, "to sell,") the theme is "the book," and the recipient is "John." Another example is how "the book belongs to me" would need two labels such as "possessed" and "possessor" and "the book was sold to John" would need two other labels such as theme and recipient, despite these two clauses being similar to "subject" and "object" functions.



**7. Why Fine-tuning a BERT model takes less time than pretraining?**

**Ans**: During pre-training, the model is trained on unlabeled data over different pre-training tasks. For fine-tuning, the BERT model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the downstream tasks.

**8. Recognizing Textual Entailment (RTE)?**

**Ans:** Textual Entailment Recognition has been proposed recently as a generic task that captures major semantic inference needs across many NLP applications, such as Question Answering, Information Retrieval, Information Extraction, and Text Summarization. This task requires to recognize, given two text fragments, whether the meaning of one text is entailed (can be inferred) from the other text.

**9. Explain the decoder stack of GPT models?**

**Ans:** GPT-2 does not require the encoder part of the original transformer architecture as it is decoder-only, and there are no encoder attention blocks, so the decoder is equivalent to the encoder, except for the MASKING in the multi-head attention block, the decoder is only allowed to glean information from the prior words