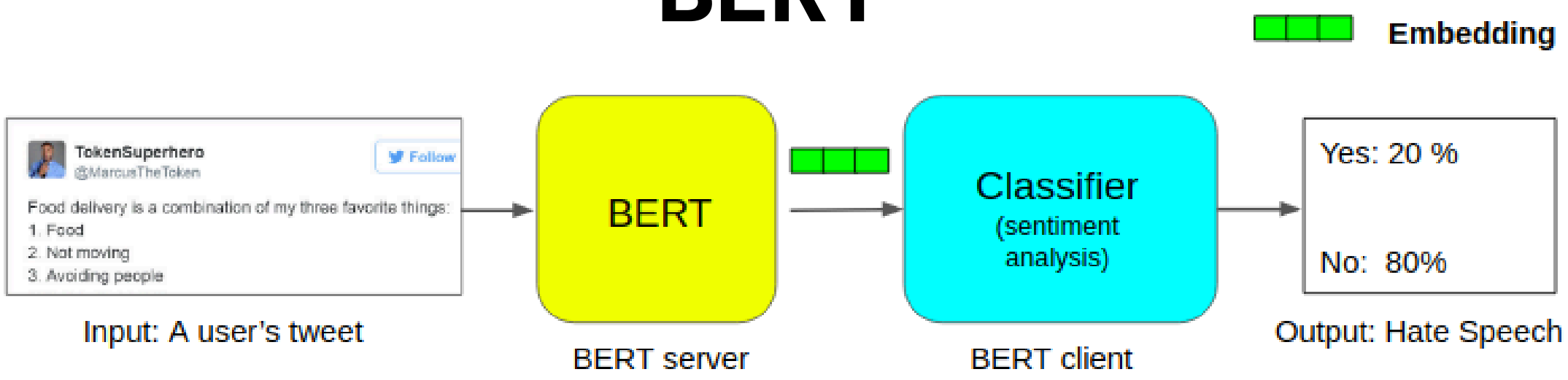
Analytics Vidhya

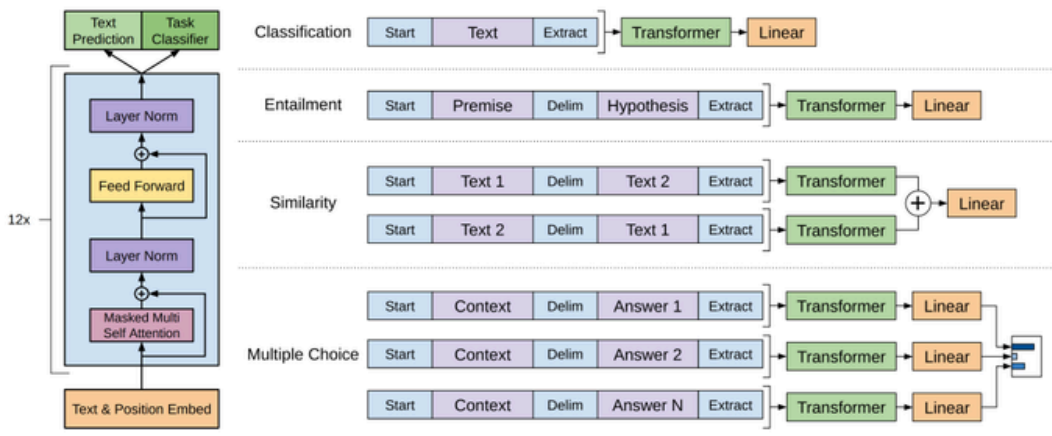
Mastering LLMs

Day 6: Introduction to Popular Transformer Models

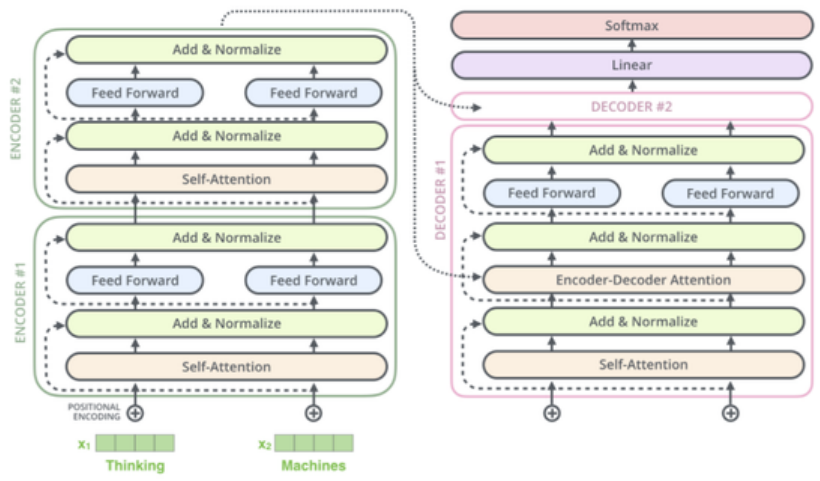
BERT



GPT

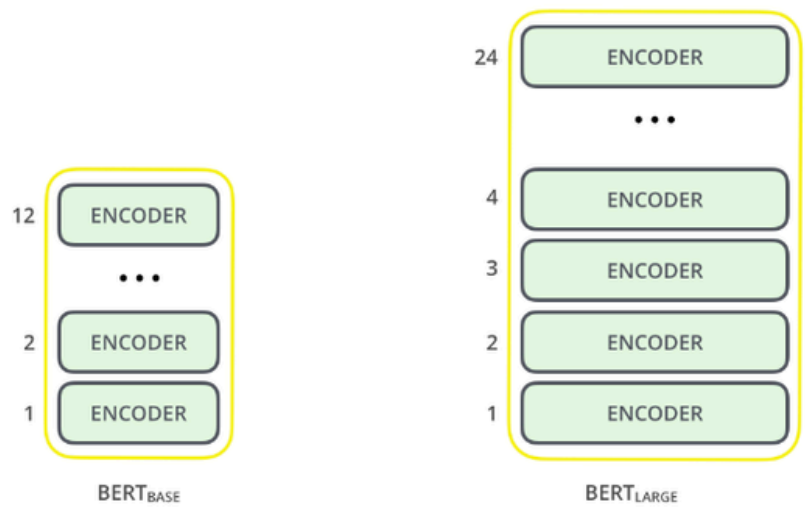


T5



ModernBERT

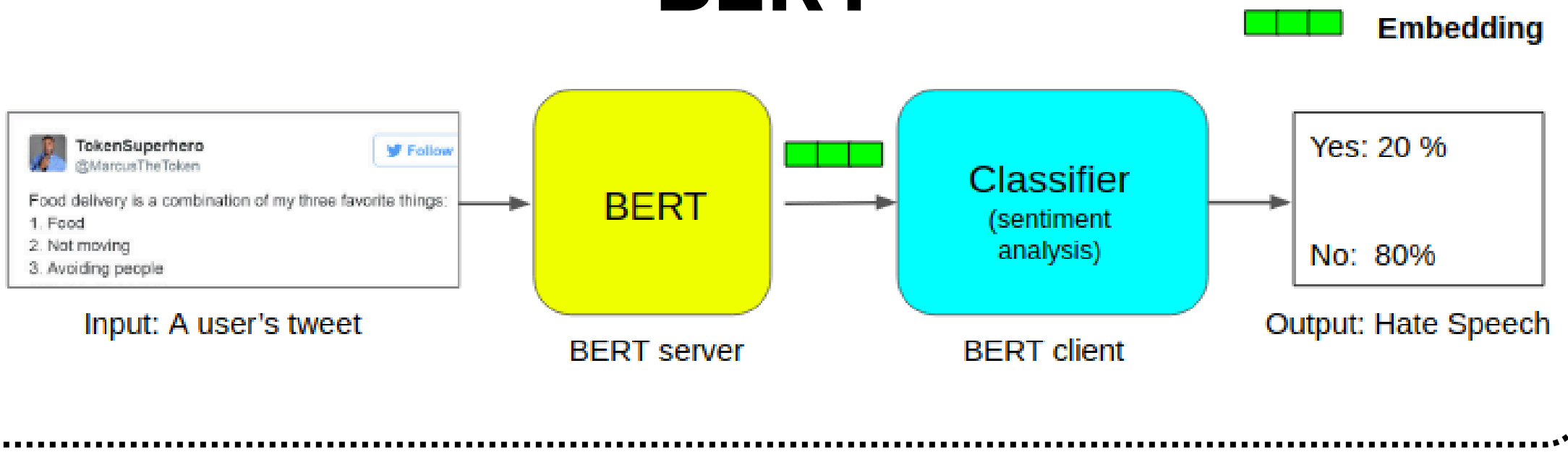
Answer.AI + Lighton +



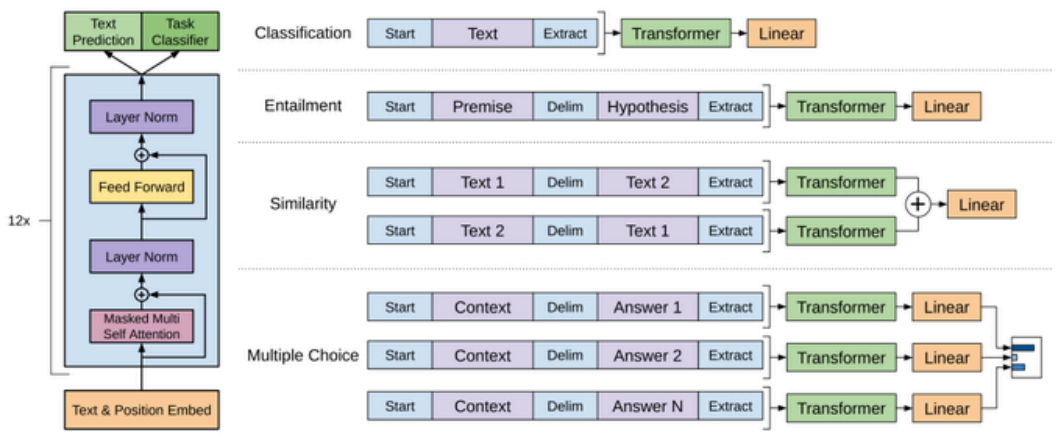
Analytics Vidhya

Now we have a clear understanding of Transformer, how it works from the previous posts. Now, it's time to learn about the popular transformer models

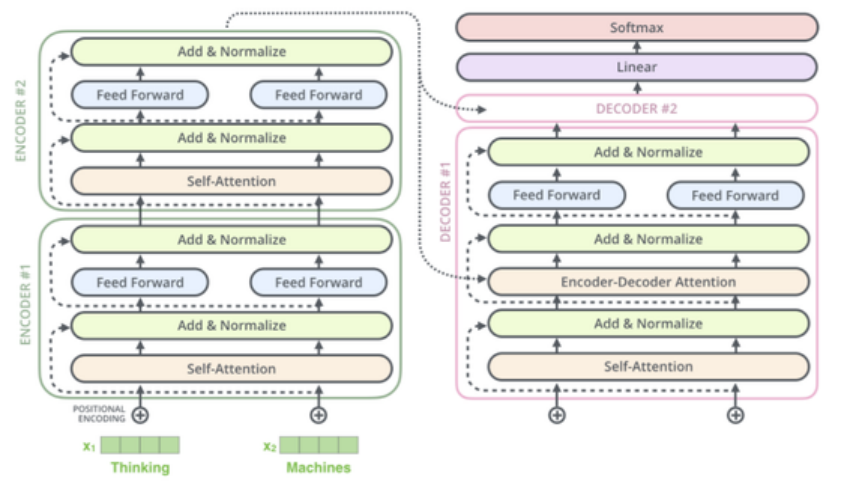
BERT



GPT



T5



ModernBERT

Answer.AI + Lighton +

Today, we will learn about

BERT ➡

What is BERT?

- BERT stands for **Bidirectional Encoder Representations from Transformers**. It is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context.

- As a result, the pre-trained **BERT model** can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of NLP tasks.

- That sounds way too complex as a starting point. But it does summarize what BERT does pretty well so let's break it down:
    - It's easy to get that BERT stands for Bidirectional Encoder Representations from Transformers. For now, the key takeaway from this line is – BERT is based on the Transformer architecture.

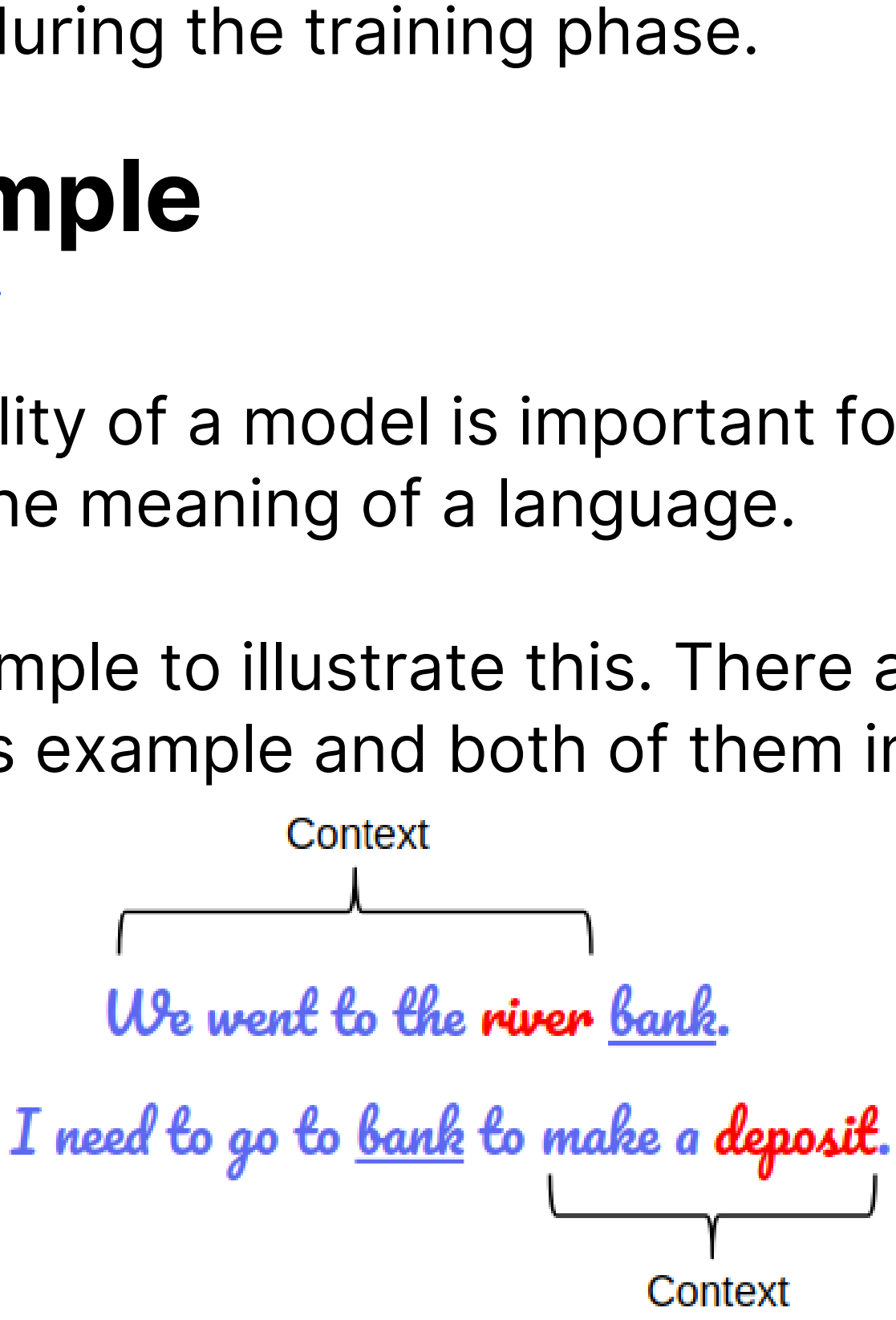
- BERT is pre-trained on a large corpus of unlabelled text including the entire Wikipedia(that's 2,500 million words!) and Book Corpus (800 million words). This pre-training step is half the magic behind BERT's success.

- BERT is a "deeply bidirectional" model. Bidirectional means that BERT learns information from both the left and the right side of a token's context during the training phase.

BERT Example

The bidirectionality of a model is important for truly understanding the meaning of a language.

Let's see an example to illustrate this. There are two sentences in this example and both of them involve the word "bank":



If we try to predict the nature of the word “bank” by only taking either the left or the right context, then we will be making an error in at least one of the two given examples.

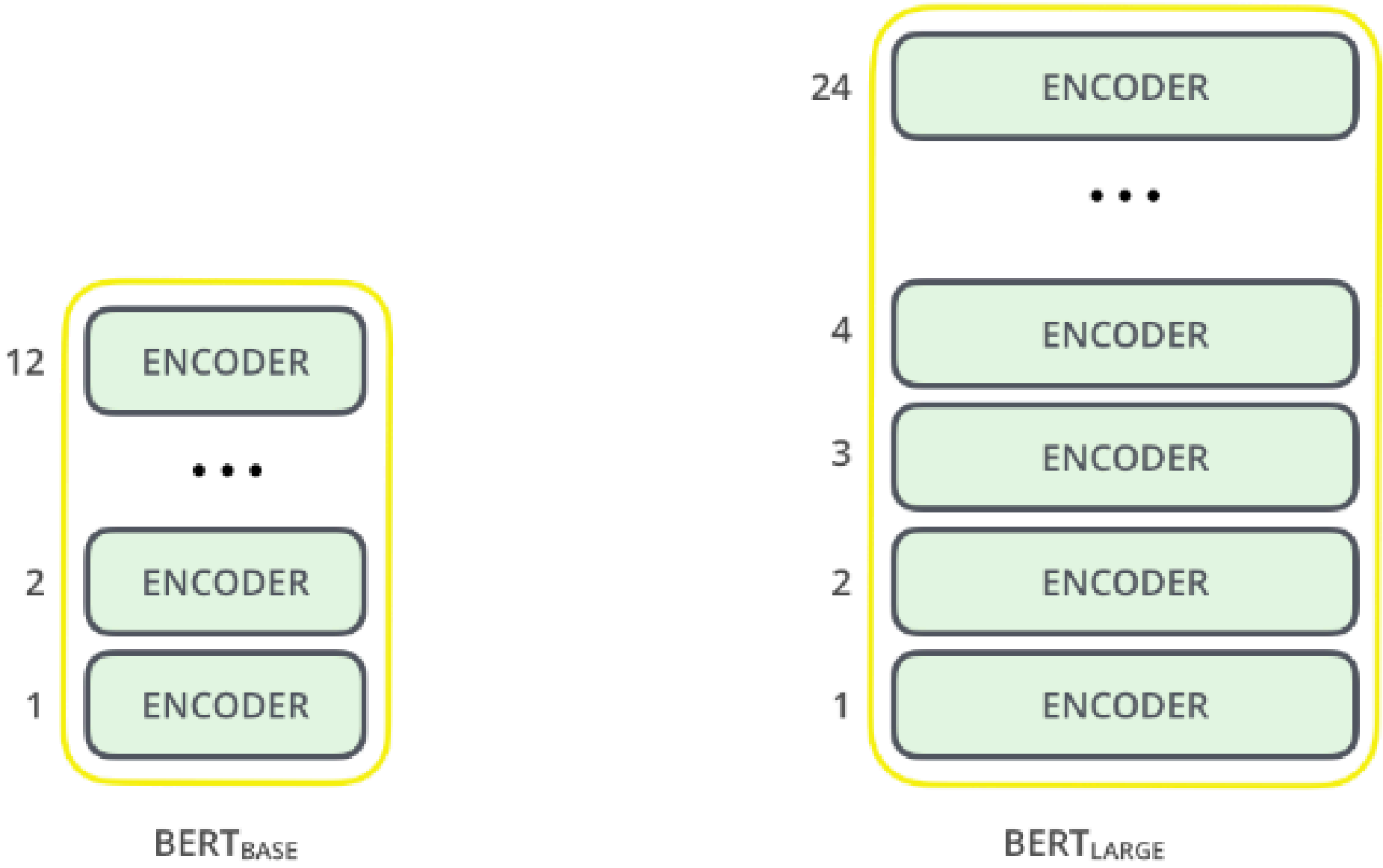
One way to deal with this is to consider both the left and the right context before making a prediction. That’s exactly what BERT does! We will see later in the article how this is achieved.

BERT’s Architecture

The BERT architecture builds on top of Transformer. Two popular variants include:

- **BERT Base**: 12 layers (transformer blocks), 12 attention heads, and 110 million parameters

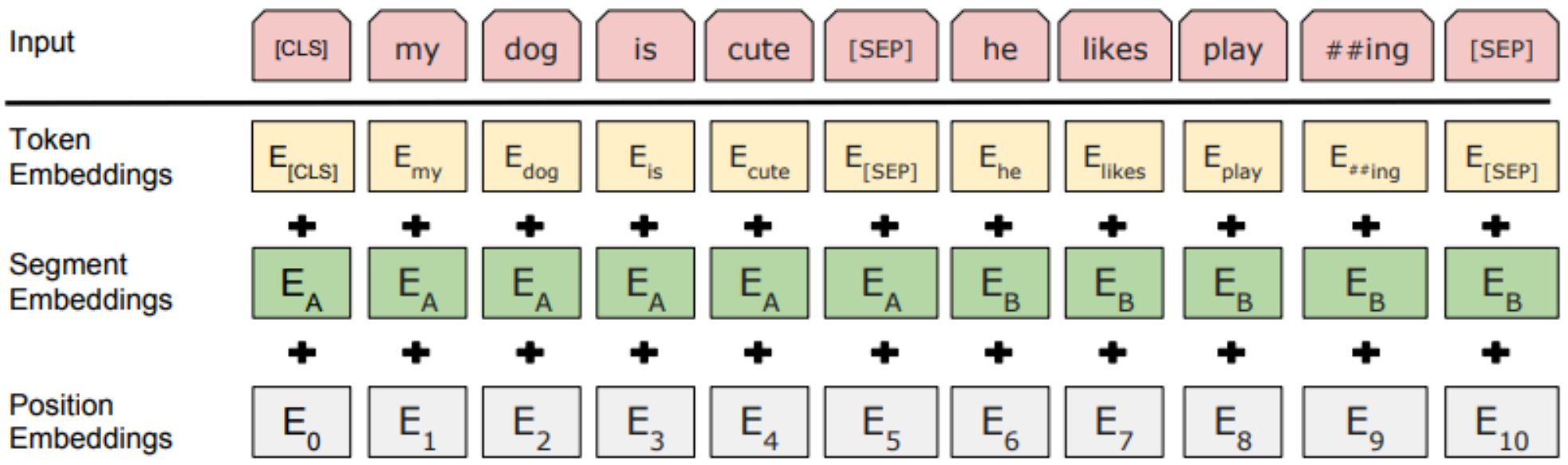
- **BERT Large**: 24 layers (transformer blocks), 16 attention heads and, 340 million parameters



The BERT Base architecture has the same model size as OpenAI's GPT-1 for comparison purposes. All of these Transformer layers are Encoder-only blocks.

Now that we know the overall architecture of BERT, let's see what kind of text processing steps are required before we get to the model building phase.

Text Preprocessing



The developers behind BERT have added a specific set of rules to represent the input text for the model. Many of these are creative design choices that make the model even better.

For starters, every input embedding is a combination of 3 embeddings:

1. **Position Embeddings**: BERT learns and uses positional embeddings to express the position of words in a sentence. These are added to overcome the limitation of Transformer which, unlike an RNN, is not able to capture “sequence” or “order” information

2. **Segment Embeddings**: BERT can also take sentence pairs as inputs for tasks (Question-Answering). That's why it learns a unique embedding for the first and the second sentences to help the model distinguish between them. In the above example, all the tokens marked as EA belong to sentence A (and similarly for EB)

3. **Token Embeddings**: These are the embeddings learned for the specific token from the WordPiece token vocabulary.

Pre-training Tasks

BERT is pre-trained on two NLP tasks:

MLM

Masked Language Model

NSP

Next Sentence Prediction

Masked Language Model (MLM):

- **Objective**: Predict randomly masked words in a sentence based on context.

- **Example**:
  - Input: "The [MASK] is near the river."
  - Model predicts: "bank."

- This forces the model to understand the relationships between all words in a sentence.

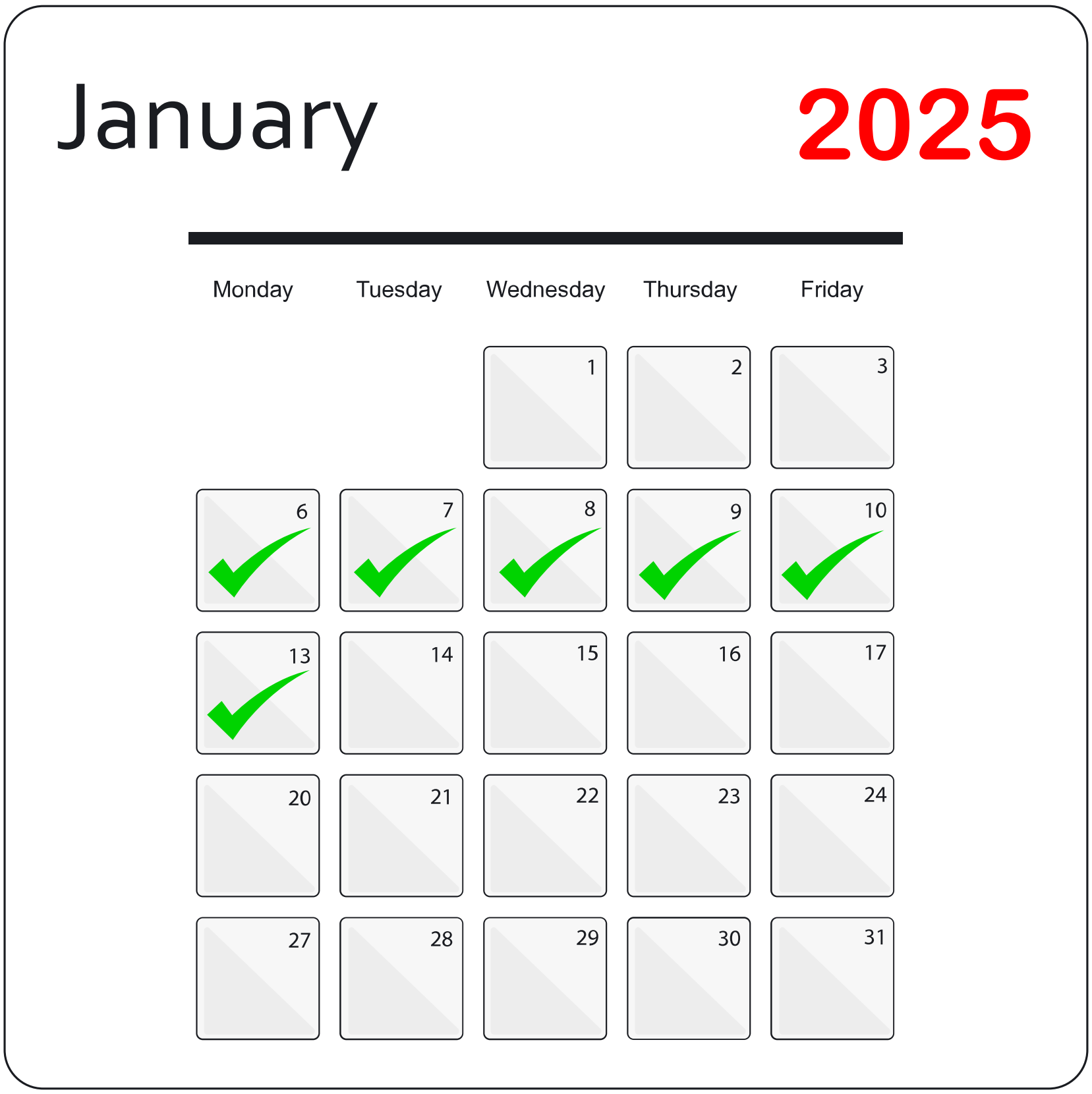
Next Sentence Prediction (NSP):

- **Objective**: Predict whether one sentence logically follows another.

- **Example**:
  - Sentence A: "I love reading books."
  - Sentence B: "Libraries are quiet places."
  - Output: "Is Sentence B the next sentence for Sentence A? (Yes/No)."

- This helps BERT understand sentence-level relationships.

Analytics Vidhya



Stay Tuned for **Day 7** of

Mastering LLMs