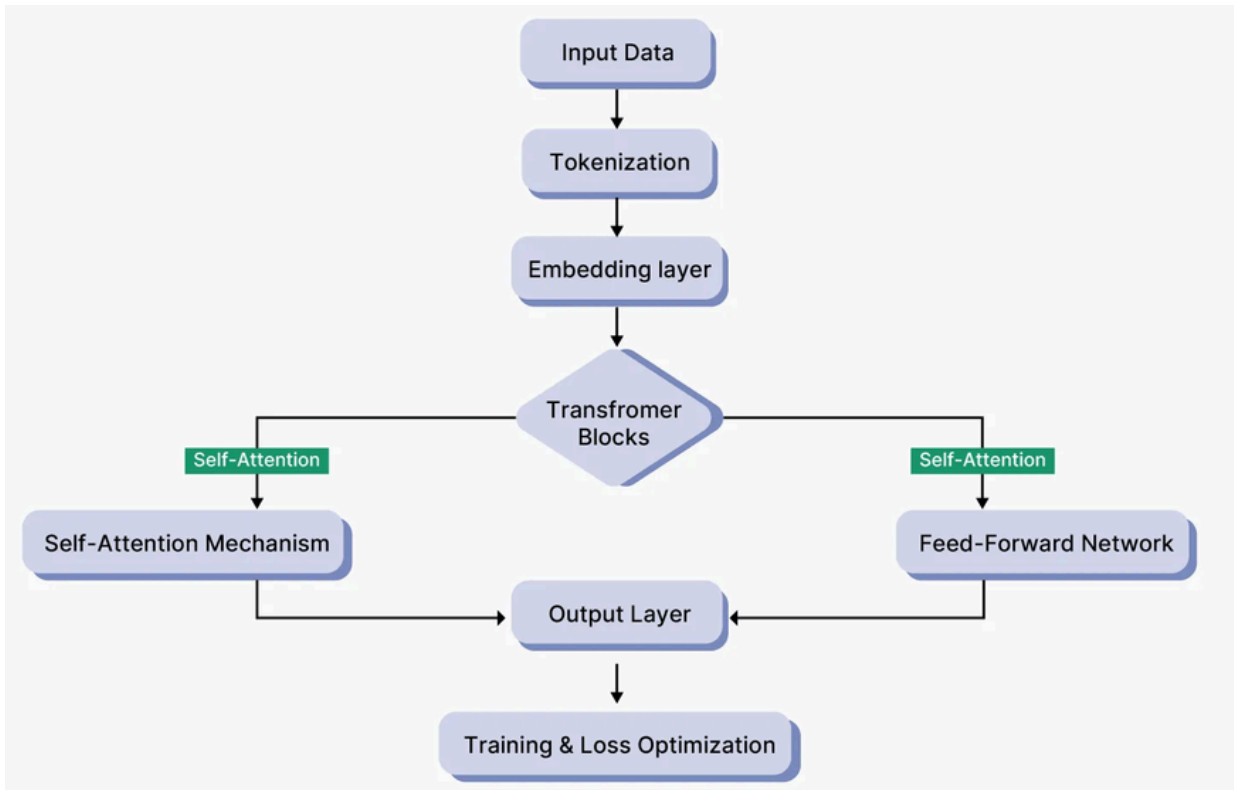
Analytics Vidhya

Top 8 Specialized AI Models



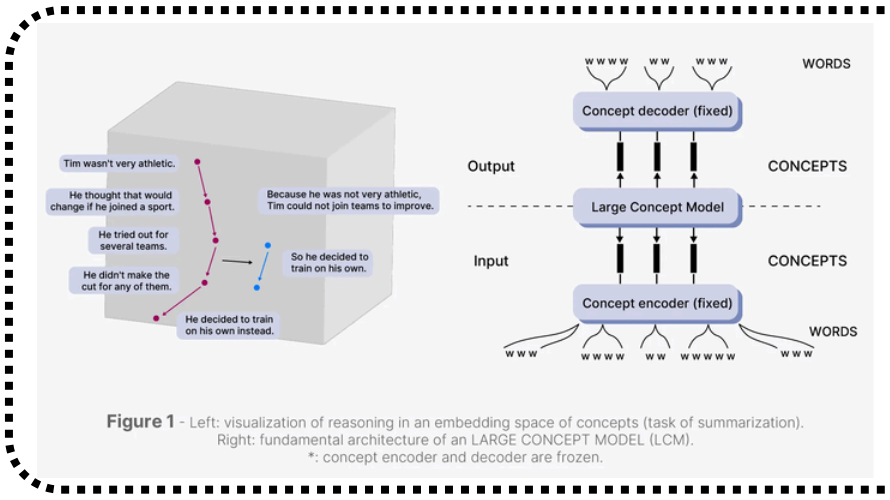
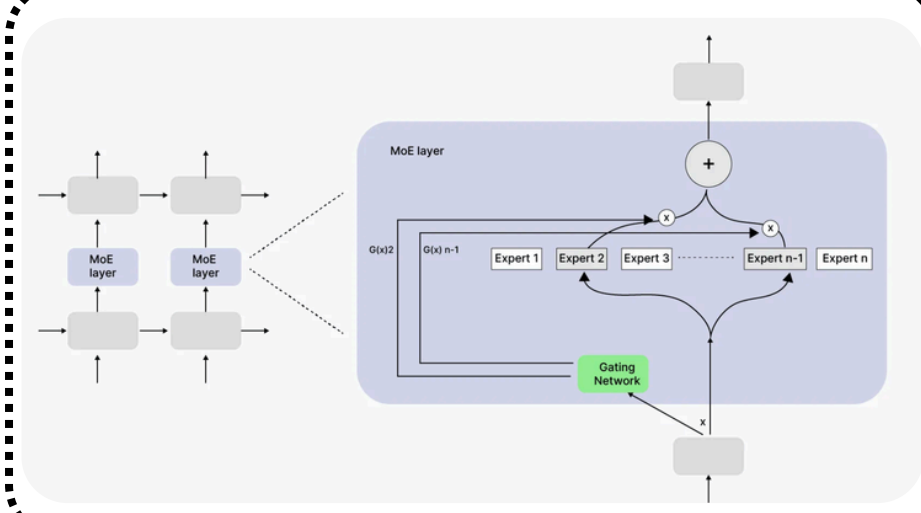
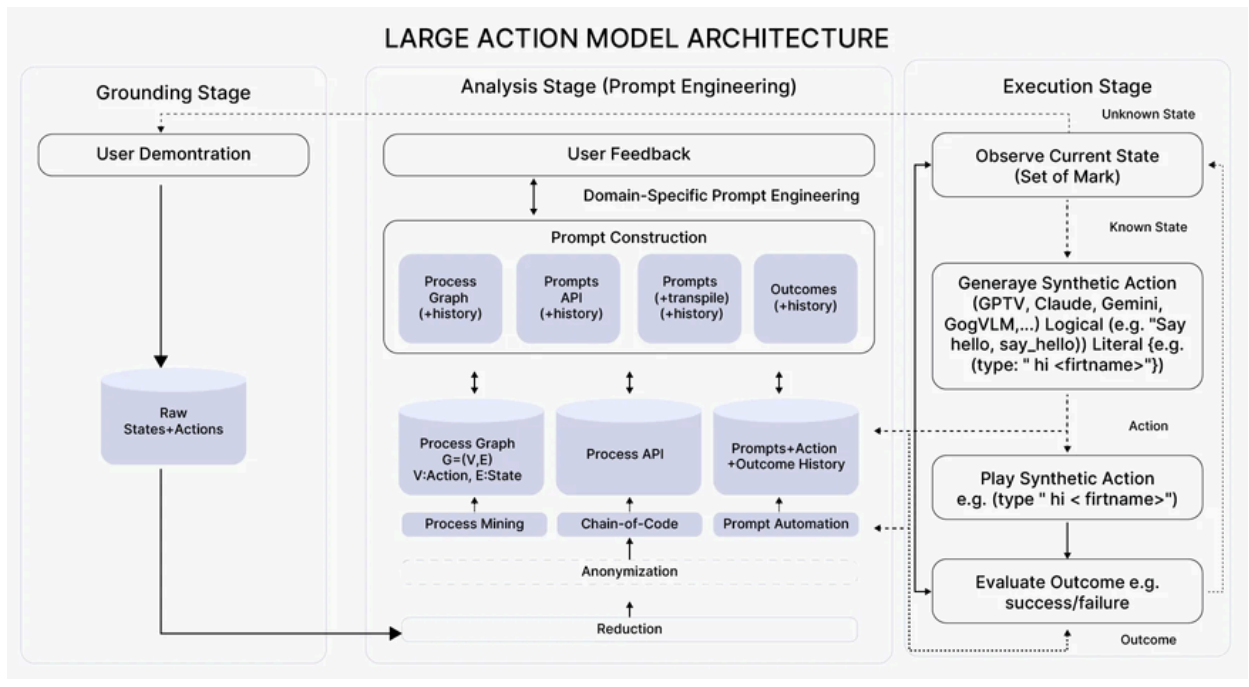
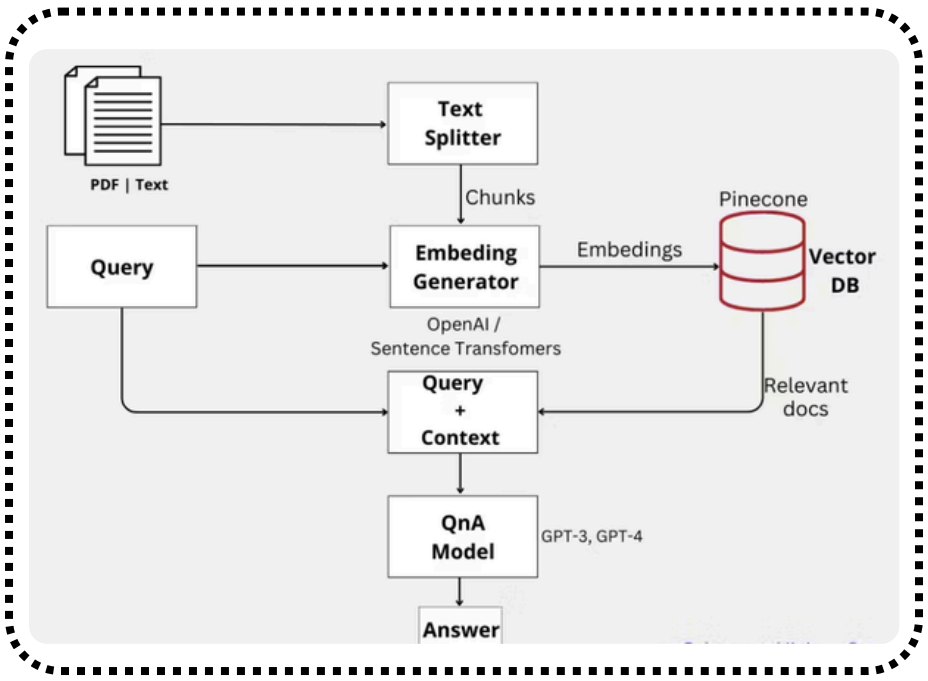
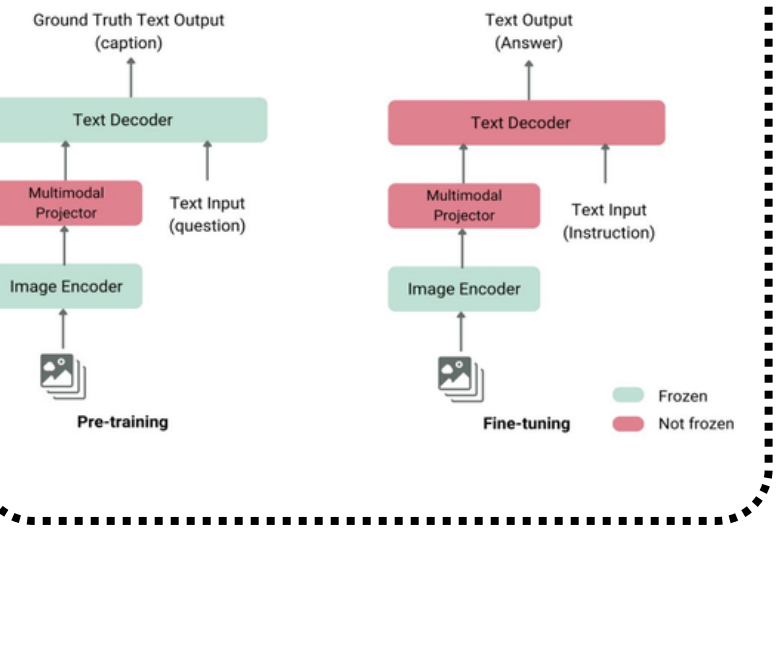


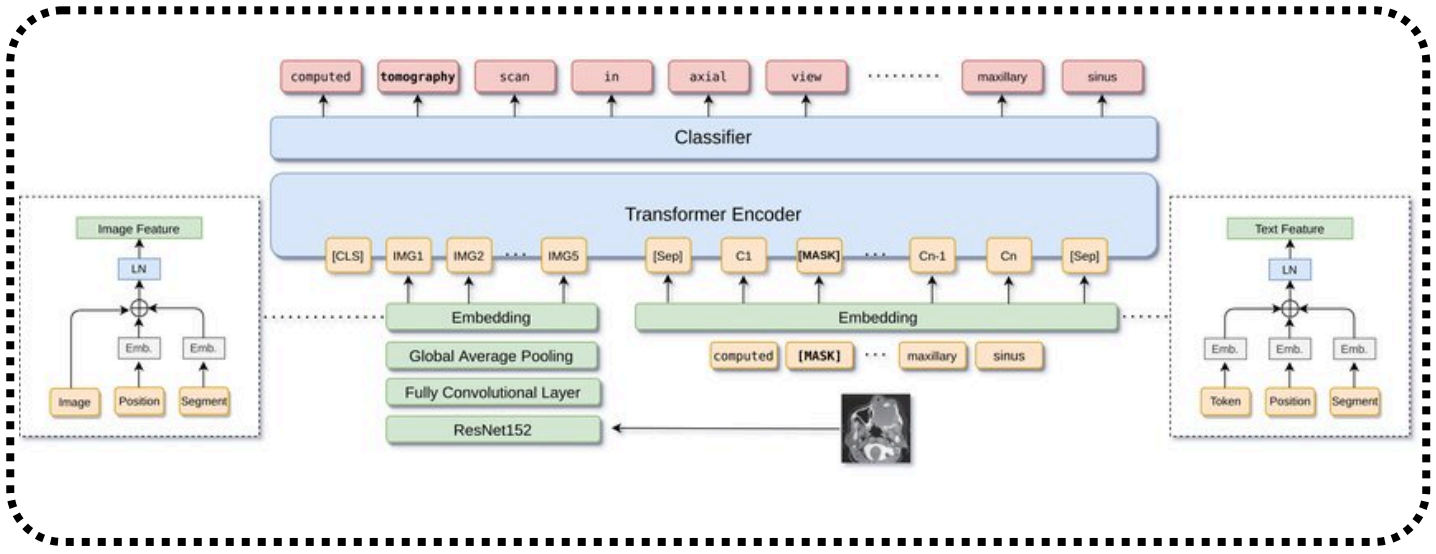
Figure 1 - Left: visualization of reasoning in an embedding space of concepts (task of summarization). Right: fundamental architecture of an LARGE CONCEPT MODEL (LCM). *: concept encoder and decoder are frozen.











LLMs: Large Language Models

LLMs, in essence, are built on transformers that consist of stacked encoder and/or decoder blocks. Here, the typical implementation includes the use of the following:

- Multi-Head Attention Layers: Different attention layers allow the model to simultaneously focus on various parts of the input, with each layer computing the Q, K, V matrices.

- Feed-Forward Neural Networks: When these networks are fed with the output of attentions, they implement two linear transformations with a non-linear activation in between, typically ReLU or GELU.

- Residual Connections and Layer Normalization: Make the training stable by allowing gradients to flow across the deep network and by normalising the network activations.

- Positional Encoding: It infuses position information using sinusoidal or learned positional embeddings as the transformer processes tokens in parallel.

- Multi-Phase Training: Pre-training preceding fine-tuning on curated datasets, followed by alignment, with RLHF being one of the approaches.

LCMs: Large Concept Models

LCMs build upon transformer architectures with specialized components for conceptual understanding, which usually include:

- Enhanced Cross-Attention Mechanisms: Connecting textual tokens to conceptual representations, and connecting the words to the underlying concepts.

- Knowledge Graph Integration: Integration of structured knowledge directly in the architecture or indirectly through pre-training objectives.

- Hierarchical Encoding Layers: These levels capture concepts at various levels of abstraction, from concrete instances to abstract categories.

- Multi-Hop Reasoning Modules: Allow following chains of conceptual relationships for multiple steps.

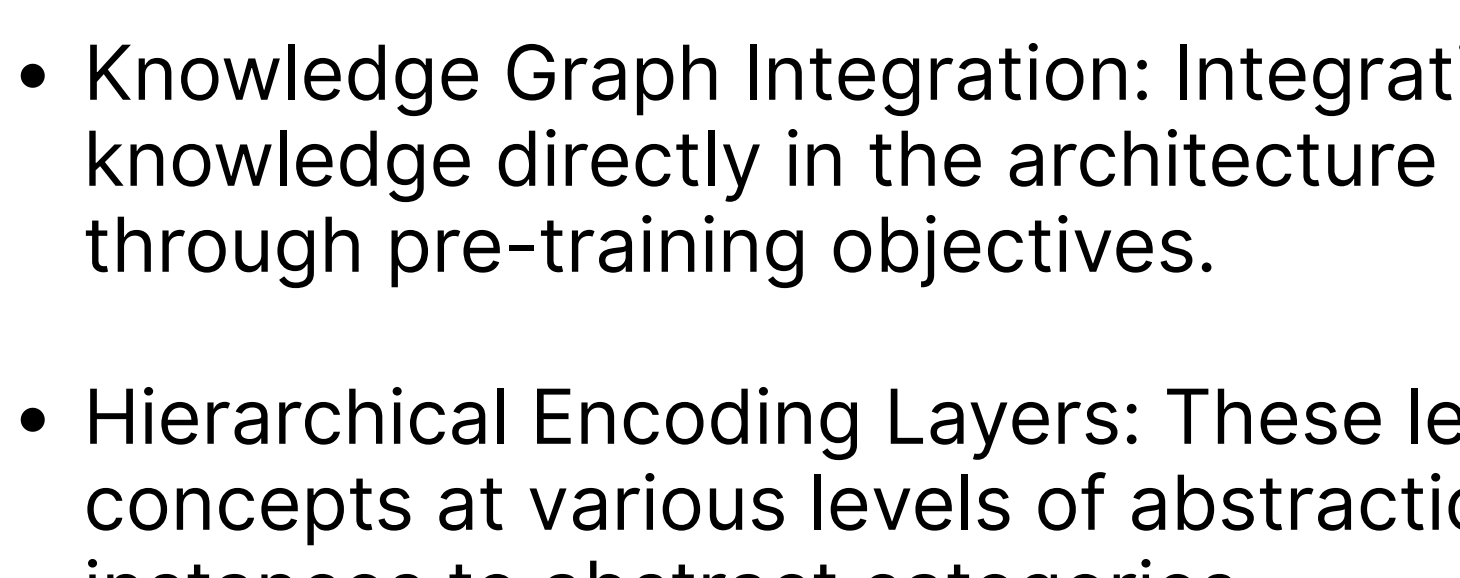
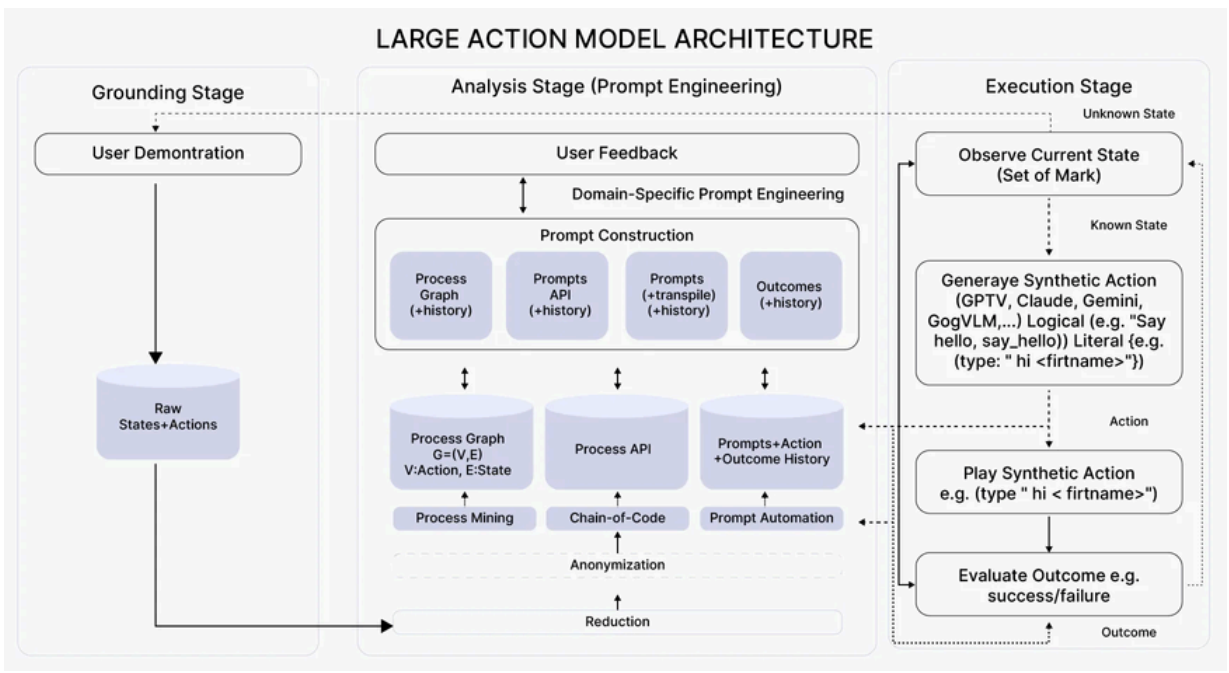


Figure 1 - Left: visualization of reasoning in an embedding space of concepts (task of summarization). Right: fundamental architecture of an LARGE CONCEPT MODEL (LCM). *: concept encoder and decoder are frozen.

LAMs: Large Action Models

LAMs combine language understanding with action execution through a multi-component design:

- Language Understanding Core: Transformer-based LLM for processing instructions and generating reasoning steps.
- Planning Module: Hierarchical planning system that decomposes high-level goals into actionable steps, often using techniques like Monte Carlo Tree Search or hierarchical reinforcement learning.
- Tool Use Interface: API layer for external tool interaction, including discovery mechanisms, parameter binding, execution monitoring, and result parsing.
- Memory Systems: Both short-term working memory and longer-term episodic memory are used to maintain context across actions.



MoEs: Mixture of Experts

MoE implements conditional computation so that different inputs activate different specialized sub-networks:

- Gating Network: The input is sent to the appropriate expert sub-networks, deciding which memories within the model should process each token or sequence.
- Expert Networks: Multi-way, specialized neural sub-networks (the experts), usually feedforward networks embedded in transform blocks.
- Sparse Activation: Only a small fraction of the parameters are activated for each input. This is implemented via top-k routing, where only the top-k scored experts are allowed to process each token.

VLMs: Vision Language Models

VLMs typically implement dual-stream architectures for visual and linguistic streams:

- Visual Encoder: It is generally a Vision Transformer(ViT) or a convolutional neural network (CNN) that subdivides an image into patches and embeds them.
- Language Encoder-Decoder: It is usually a transformer-based language model that takes in text as input and outputs.
- Cross-Modal Fusion Mechanism: This mechanism connects the visual and linguistic streams through the following:
    - Early Fusion: Project visual features into the language embedding space
    - Late Fusion: Process separately, then connect with attention at deeper layers.
    - Interleaved Fusion: There shall be multiple points of interaction across the whole network.
    - Join Embedding Space: A unified representation where visual concepts and textual concepts would be mapped to comparable vectors.

SLMs: Small Language Models

- The SLMs develop specialized techniques optimized for computation efficiency:
-
- Efficient Attention Mechanisms: Alternative systems to the standard self-attention, which scales quadratically and include:
    - Linear attention: Reduces complexity to O(n) by kernel approximations.
    - Local attention: Attend only within local windows, rather than the full sequence.
- State Space Models: Another approach to sequence modeling with linear complexity.
- Parameter Efficient Transformers: Techniques to reduce parameters number include:
    - Low-Rank Factorization: Decomposing weight matrices into the product of smaller matrices.
    - Parameter Sharing: Reuse of weights across layers.
    - Depth-wise Separable Convolutions: Replace dense layers with more efficient ones.

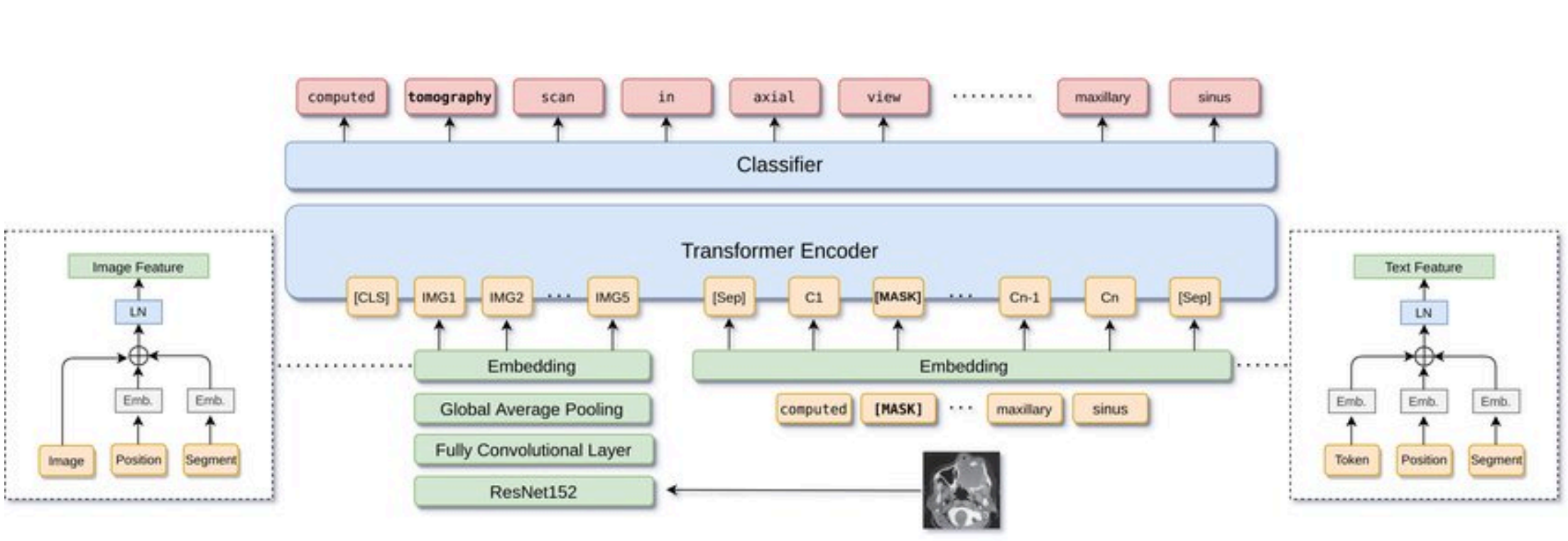
MLMs: Masked Language Models

An MLM implements a bidirectional architecture for holistic contextual understanding:

- Encoder-only Transformer: Unlike decoder-based models that process the text strictly left to right, MLMs, through the encoder blocks, attend to the entire context bidirectionally.

- Masked Self-Attention Mechanism: Each token can attend to all other tokens within the sequence through scaled dot-product attention without any causal mask being applied.

- Token, Position, and Segment Embeddings: These embeddings combine to form input representations that include content and structure information.



SAMs: Segment Anything Models

The architecture of SAM is multi-component for image segmentation:

- Image encoder: It is a vision transformer backbone that encodes the input image to produce a dense feature representation. SAM uses the VIT-H variant, which contains 32 transformer blocks with 16 attention heads per block.

- Prompt Encoder: Processes various sorts of user inputs, like:
  - Point Prompts: Spatial coordinates with background indicators.
  - Box Prompts: Two-point coordinates
  - Text Prompts: Processed through a text encoder
  - Mask Prompts: Encoded as dense spatial features

- Mask Decoder: A transformer decoder combining image and prompt embeddings to produce mask predictions, consisting of cross-attention layers, self-attention layers, and an MLP projection head.