Advanced Generative AI: Models and Architecture

Attention Mechanism and Transformers

Quick Recap

Lorem ipsum dolor sit amet, adipiscing elit, sed do incididunt ut labore magna aliqua.?

- In a variational autoencoder, how does the concept of a **latent space** contribute to its ability to generate new data points?
- The roles of the **generator** and **discriminator** in a generative adversarial network. How does their adversarial relationship during training lead to the generation of realistic images?

Engage and Think

- Imagine creating an AI system tasked with summarizing a long, complex scientific article. This article contains important points, technical terms, and detailed explanations. The system aims to produce a concise, clear summary. It must capture the article's essence without omitting any significant details.

- How can you design your AI to identify and highlight the most important parts of a complex scientific article while considering the overall context, akin to using a spotlight?

Learning Objectives

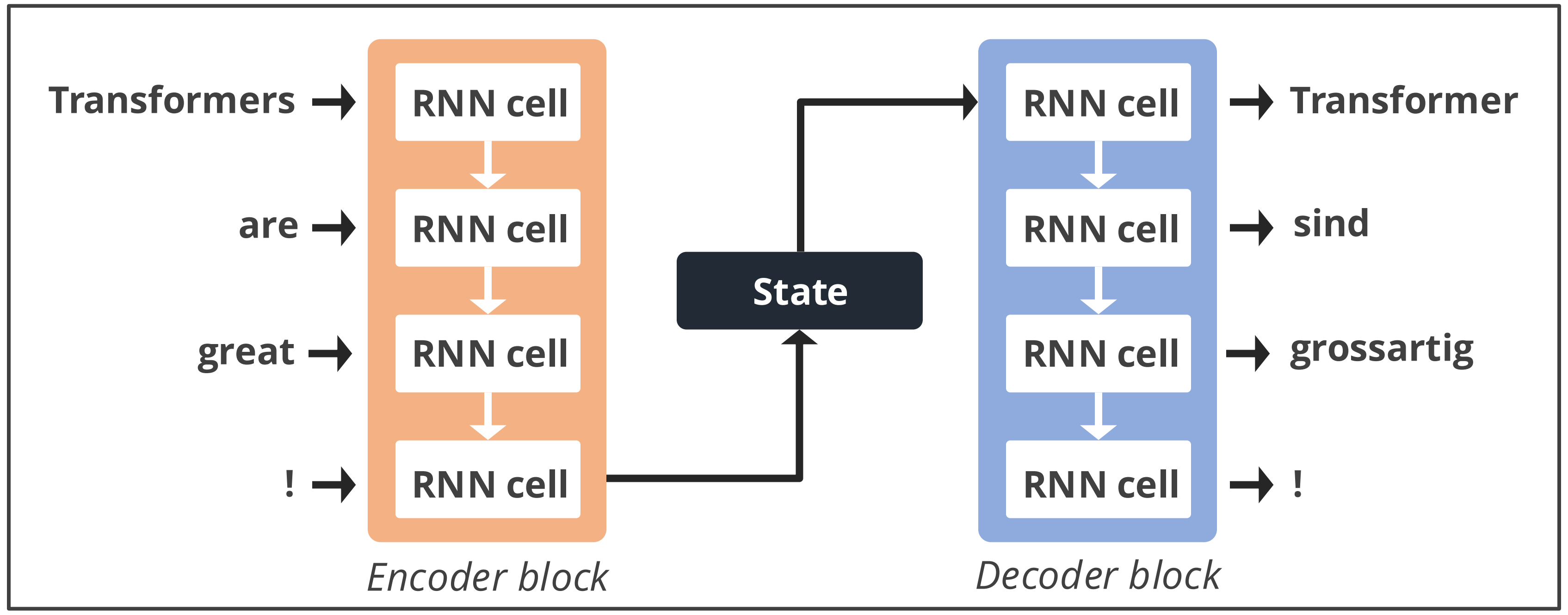
By the end of this lesson, you will be able to:

- Analyze the foundational concepts of attention mechanisms in Transformer models to emphasize their impact on language processing capabilities
- Determine how multi-head attention mechanisms enhance the Transformer architecture in processing complex language models
- Identify the practical applications of Transformer-based models across various fields, including healthcare, finance, and recommendation systems

Introduction to Attention Mechanism

Traditional Architecture

Below is a traditional encoder and decoder architecture:



Limitation of Traditional Architecture

The limitations of the traditional encoder and decoder architecture include:

Information bottleneck

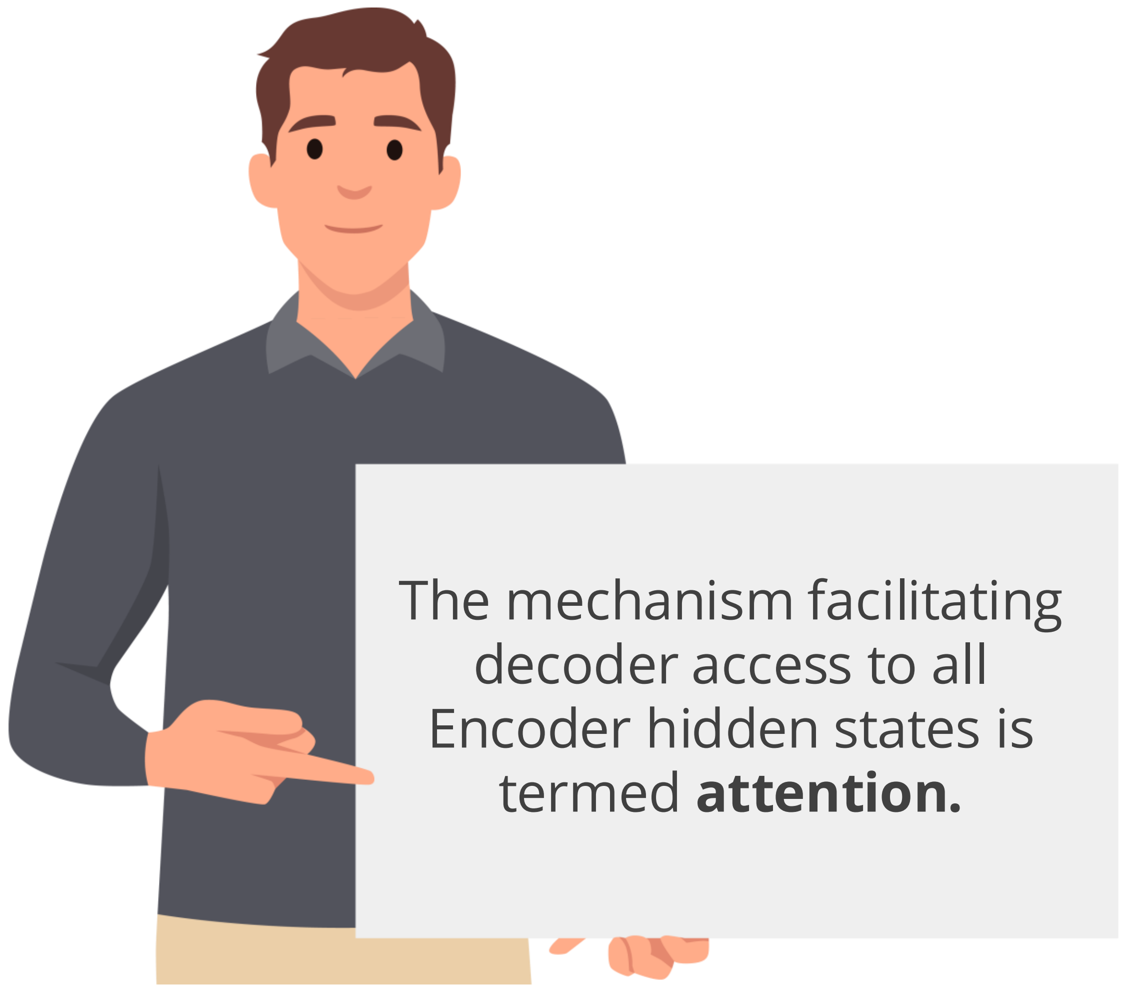
The final hidden state of the encoder must encapsulate the entire input sequence, posing challenges for long sequences as early information might be lost.

Challenges with long sequences

Traditional architectures struggle with long sequences as they compress all information into a single, fixed representation, making it difficult to maintain the integrity of the entire sequence.

Attention Mechanism

To address this bottleneck, a solution involves granting the decoder access to all the encoder's hidden states.



The mechanism facilitating decoder access to all Encoder hidden states is termed **attention.**

Attention Mechanism

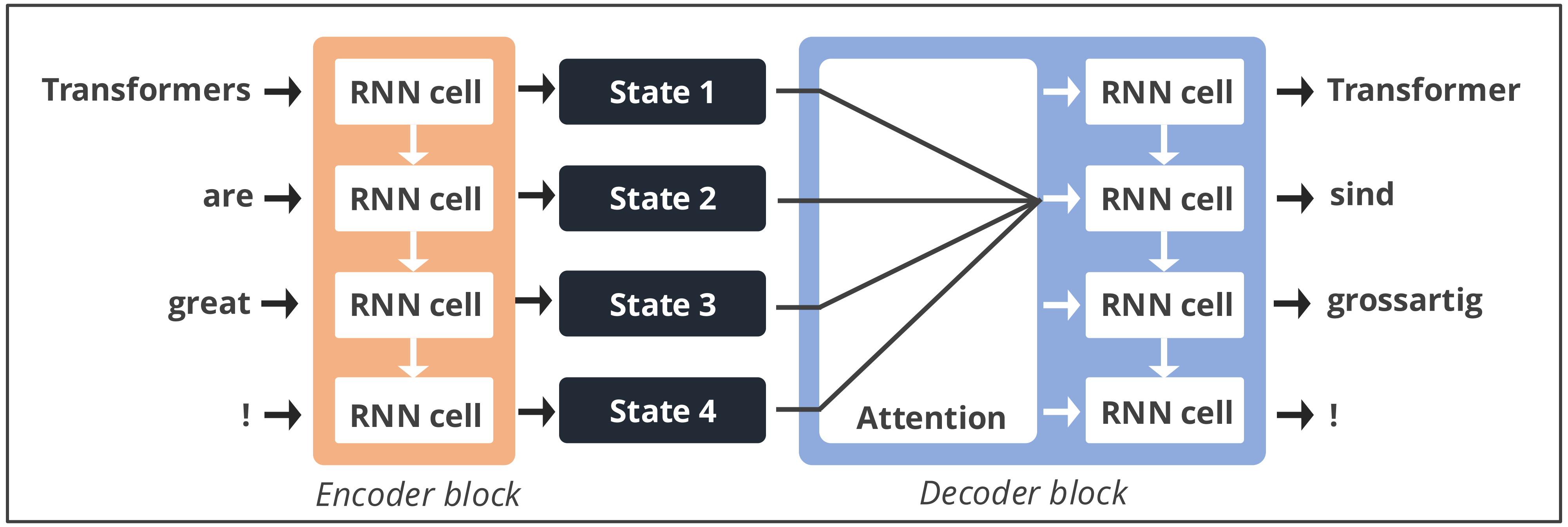
In machine learning, attention is a technique that helps models focus on the most relevant input data by assigning varying importance to different elements. This improves tasks like language translation and image recognition by prioritizing critical information.

The concept of attention in machine learning was inspired by human cognitive processes, where we naturally focus on certain aspects of our environment while processing information

Attention Based Architecture

Bahdanau et al. (2014) introduced the attention mechanism in their paper **Neural Machine Translation by Jointly Learning to Align and Translate**.

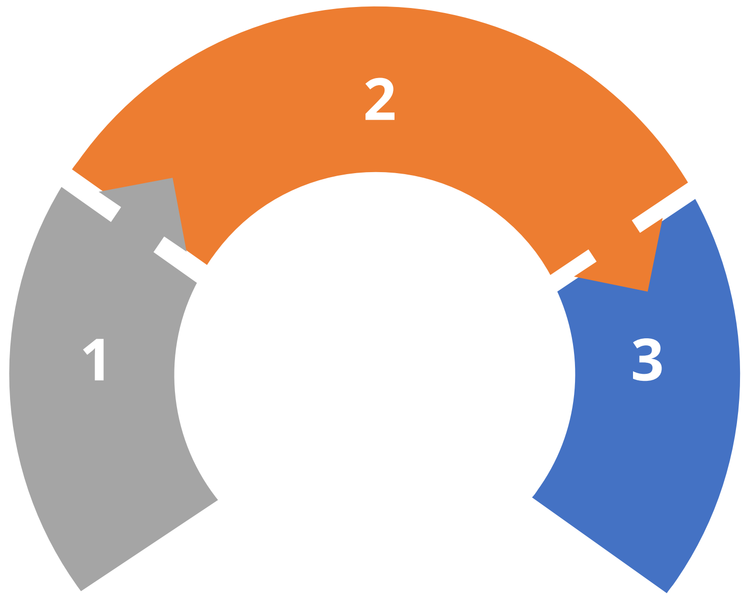
Below is the encoder and decoder architecture with attention:



Working of Attention Mechanism

Below are the working stages of an attention mechanism:

Assigning Weights
Models assign weights to input elements based on their relevance to a specific context or query.



Focusing on Relevant Information
Attention mechanisms enable models to selectively focus on relevant parts of the input data when generating output.

Contextual Computation
This weighted sum or context vector captures crucial information for the model’s decision-making.

Types of Attention Mechanism

Various attention mechanisms are tailored for different applications, and three such types are the following:

$$e_{ij} = v_a^T \tanh(W_a[h_i; s_{j-1}])$$

Additive attention

- The model creates a weighted sum of input elements by applying learned weights or parameters.
- This method calculates attention by learning the importance of each element in the input sequence.

Types of Attention Mechanism

Various attention mechanisms are tailored for different applications, and three such types are the following:

$$\alpha_{ij} = \frac{\exp(h_i^T W_a s_{j-1})}{\sum_k \exp(h_k^T W_a s_{j-1})}$$

Multiplicative attention

- The model generates attention weights through element-wise multiplication between the input elements and a learned parameter vector.
- This approach allows the model to capture complex interactions between elements.

Types of Attention Mechanism

Various attention mechanisms are tailored for different applications, and three such types are the following:

Self Attention

- Model attends to different positions of the input sequence by comparing each position with all other positions.
- It involves comparing each element with every other element in the sequence, including itself, to reweigh their importance based on contextual relevance.
- Self-attention is fundamental in generative AI and is commonly used in Transformers.

Quick Check

What is the primary limitation of traditional encoder-decoder architectures in processing sequence data?

A. They are optimized for handling long sequences efficiently.

B. They require additional layers to process sequences of any length.

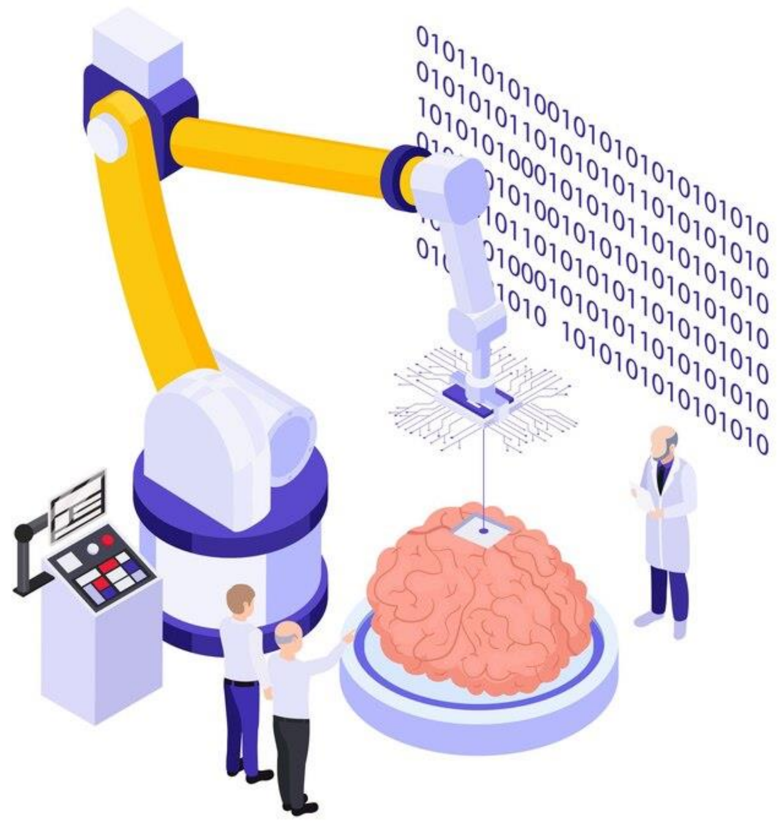
C. They can lose early sequence information due to the information bottleneck.

D. They are unable to process sequences in a fixed representation.

Self-Attention Mechanism

Understanding Self-Attention

In an embedding matrix, each token will share some context with the other, but to capture the relationship, you can assign a score to each embedding based on the token position in the sentence or context.



This is achieved by adding the positional encoding directly and by the mechanism of self-attention.

Understanding Self-Attention

Self-attention, a key component in Natural Language Processing (NLP), enables the network to focus on particular words or phrases in a sentence for improving context understanding.

Example:

You are reading a mystery novel.

While reading a novel, you simultaneously focus on the current page and recall earlier events, characters, and clues.

This context helps you understand the story better and predict future events.

Understanding Self-Attention

Consider an example below where the term *Apple* has different meanings:

Sentence 1: Apple was juicy.

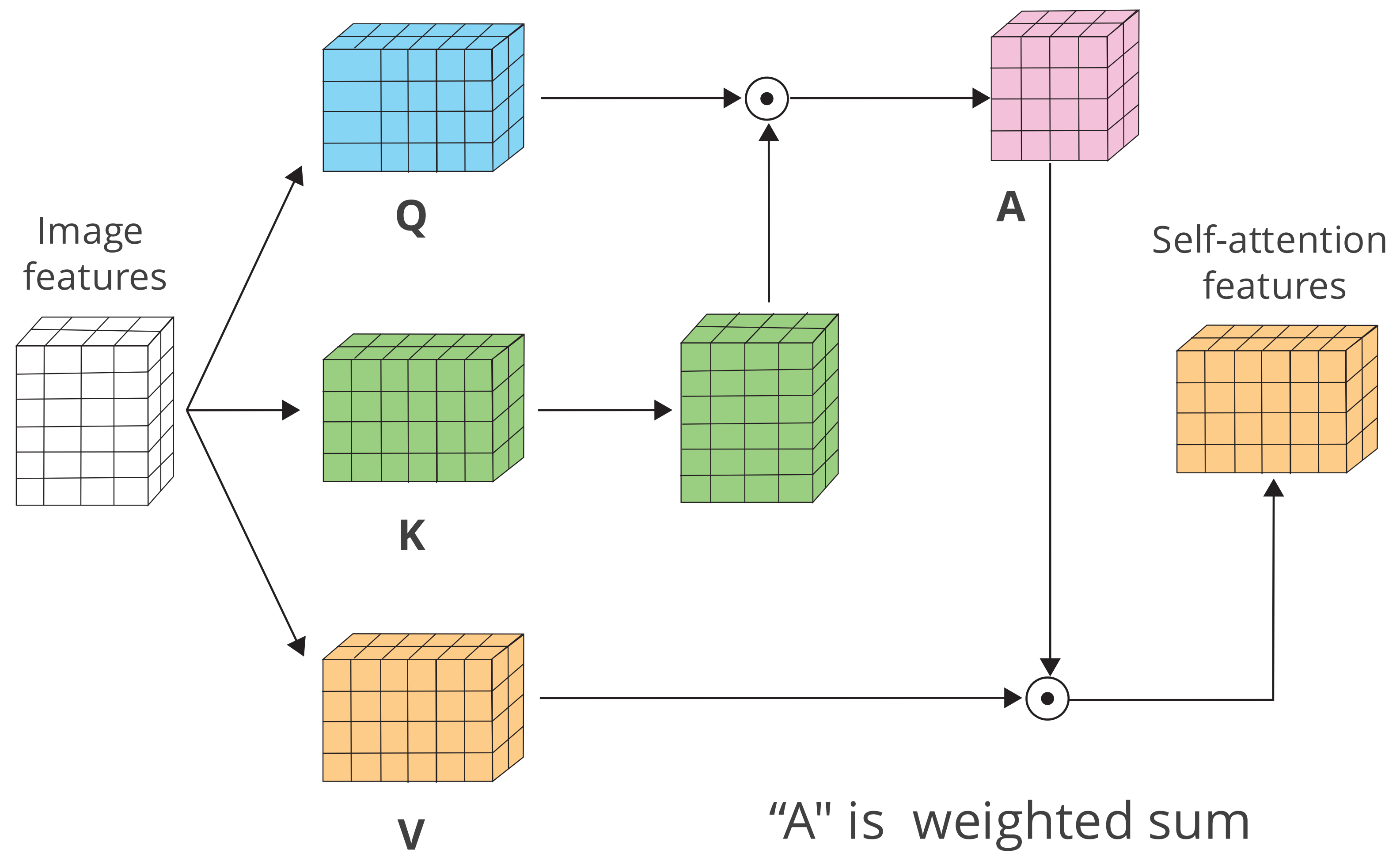
Sentence 2: Apple stock crashed today.

In order to differentiate them, you need to capture their contextual embedding. So, to get these new embeddings, you need to pass the old embedding through the self-attention mechanism.

In these two sentences, the context of *Apple* differs from the surrounding words and the aim is to capture that instead of using an averaged embedding.

Self-Attention

In the self-attention mechanism, the input image feature is defined as Query (Q), Key (K), and Value (V).



"A" is weighted sum

The attention mechanism is to calculate the attention weight between the query and key and then used to strengthen the value.

Self-Attention

For each word (let's focus on the word *Apple* in the first sentence), calculate how much attention it should pay to every other word in the sentence, including itself. This is done using three components: Query (Q), Key (K), and Value (V).

- Query (Q): Represents the word we are focusing on (e.g., *Apple*)
- Key (K): Represents each word in the sentence (including *Apple* itself)
- Value (V): Also represents each word in the sentence but will be weighted based on the attention scores

Self-Attention Formula

Vectors Q, K, and V are derived by multiplying the embedding of the word different weight matrices. The formula for scaled dot-product is:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where,

- Q is the query matrix
- K is the key matrix
- V is the value matrix
- dk is the dimension of the key vectors

Revolutionizing Sequence Modeling with Attention

Attention is all you need introduces the transformer model, a groundbreaking architecture that eliminates recurrence and convolution to achieve faster, scalable, and more accurate sequence modeling.

The research was conducted by Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Łukasz Kaiser, and Illia Polosukhin and was presented at NeurIPS 2017.

The transformer replaces traditional sequence models with a fully attention-based approach, leading to breakthroughs in machine translation, natural language processing (NLP), and AI development.

Revolutionizing Sequence Modeling with Attention

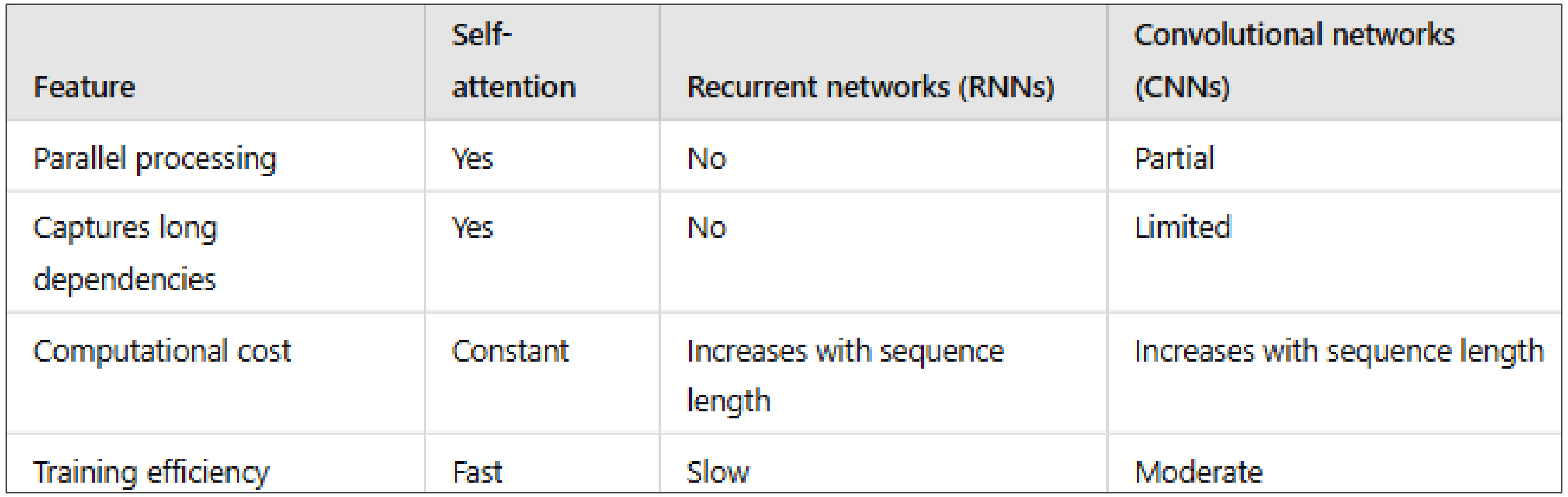
A self-attention mechanism was required when CNNs and RNNs couldn’t perform well. Some limitations of CNNs and RNNs are listed below:

Recurrent neural networks (RNNs), including LSTMs and GRUs, suffer from sequential processing, making them difficult to parallelize. This limitation results in slower training times and an inability to capture long-range dependencies effectively.

Convolutional neural networks (CNNs) improve parallelization but require deep layers to capture dependencies. As the sequence length increases, CNNs become computationally expensive and inefficient.

Revolutionizing Sequence Modeling with Attention

- Unlike RNNs and CNNs, the self-attention mechanism processes all tokens at once, enabling efficient parallel computation.
- Self-attention enables shorter paths between dependencies, making it easier for models to understand and relate information across long sequences.
- This efficiency is what makes transformers the foundation of modern AI architectures like BERT, GPT, and T5.



Feature	Self-attention	Recurrent networks (RNNs)	Convolutional networks (CNNs)
Parallel processing	Yes	No	Partial
Captures long dependencies	Yes	No	Limited
Computational cost	Constant	Increases with sequence length	Increases with sequence length
Training efficiency	Fast	Slow	Moderate

Revolutionizing Sequence Modeling with Attention

The transformer model also achieved new milestones by setting new records in machine translation performance on the WMT 2014 benchmark:

English-to-German Translation: Achieved a BLEU score of 28.4, surpassing the best prior model (26.36)

English-to-French translation: Achieved a BLEU score of 41.0, outperforming the previous best model (40.4)

Attention is not just a mechanism; it is the foundation of modern AI and the future of deep learning.

Mechanics Behind Self-Attention

In the self-attention mechanism, attention scores are crucial, determining the importance assigned to each word in a sentence during the encoding process.

Calculation of Attention Scores

1. For every word in a sentence, the model computes an attention score about every other word.
2. This computation involves taking the dot product of the word’s vector representation with the vector representations of all other words.

Mechanics Behind Self-Attention

In the self-attention mechanism, attention scores are crucial, determining the importance assigned to each word in a sentence during the encoding process.

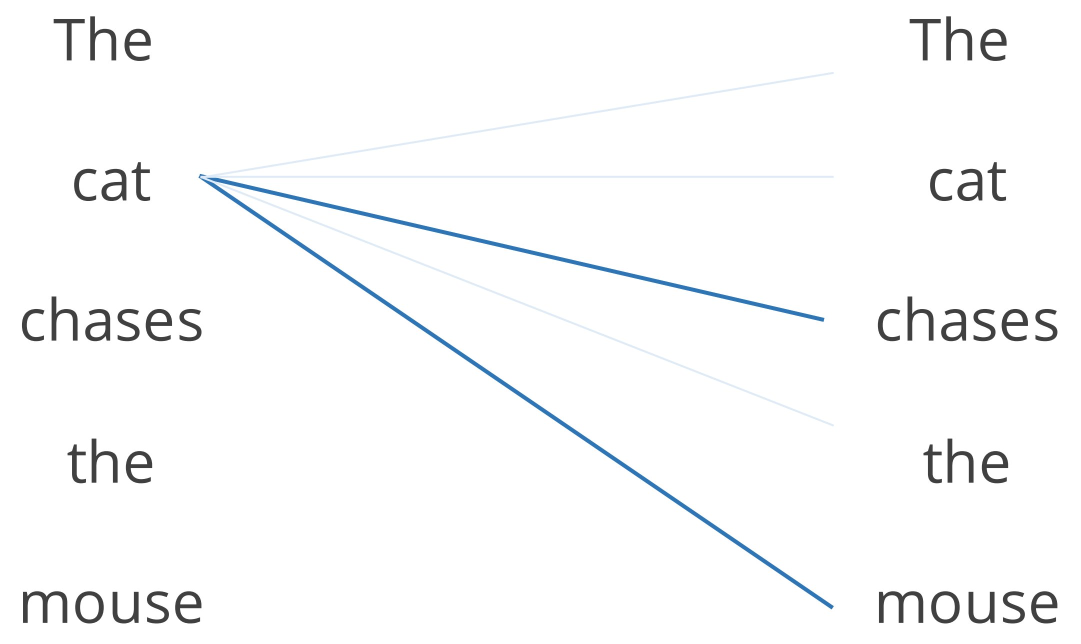
Calculation of Attention Scores

3 After the dot product, a softmax function is applied.

4 A softmax function ensures that all the attention scores are normalized and sum up to 1, representing probabilities.

Mechanics Behind Self-Attention

The attention scores are based on the relative positions and semantic similarities of words, with those closer in position or more similar semantically receiving higher scores.



Example: The cat chases the mouse.

- The word "cat" has semantic connections to both "chases" and "mouse."
- As a result, the pairs "cat-chases" and "cat-mouse" achieve high attention scores.
- This is because "cat" performs the action "chases," and "mouse" is the action's object.
- The model identifies these relationships and assigns higher attention scores accordingly.

Mechanics Behind Self-Attention

The process of context-aware embeddings are as follows:

Constructing contextual embeddings:

- After computing the attention scores, form a contextual representation of the sentence.
- Achieve this by creating a weighted sum of the vector representations (embeddings) of all words in the sentence.
- Use the previously computed attention scores as weights for this summation.

Mechanics Behind Self-Attention

The process of context-aware embeddings are as follows:

The Significance of weights:

- The weights, also known as attention scores, are essential.
- They determine the contribution of each word to the representation of another word.
- A higher attention score indicates a significant impact on the context of the other word.

Mechanics Behind Self-Attention

The process of context-aware embeddings are as follows:

Outcome:

- The process results in a new set of word representations.
- These new representations, known as context-aware embeddings, differ from the original word embeddings.
- Context-aware embeddings capture the semantics of each word within the context of the given sentence.
- They represent not just the word, but also its meaning and role in the sentence.

Quick Check

Question: What is the primary purpose of the self-attention mechanism in the context of NLP models?

A. To reduce the computational complexity of the model.

B. To assign higher attention scores to words that are semantically similar or closer in position.

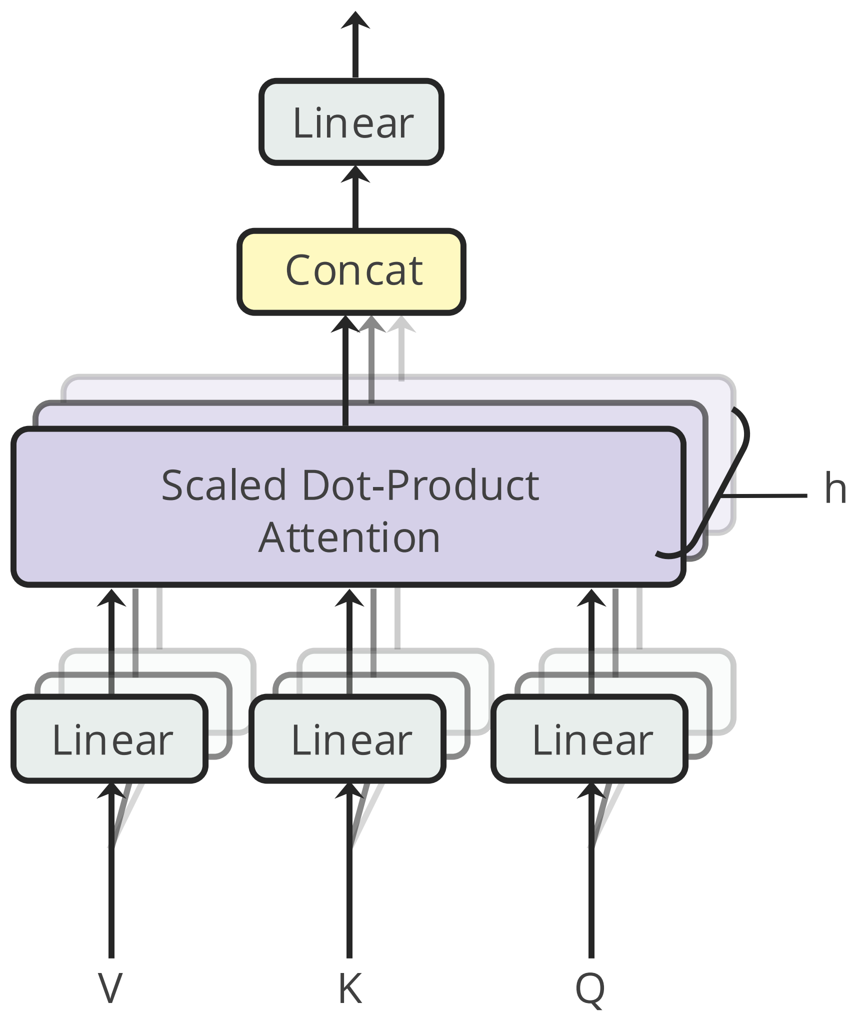
C. To increase the size of the model's vocabulary.

D. To translate sentences from one language to another.

Multi-Head Attention Mechanism

Multi-Head Attention

It is an extension of the self-attention mechanism that enables a model to focus on various parts of the input sequence at different times.

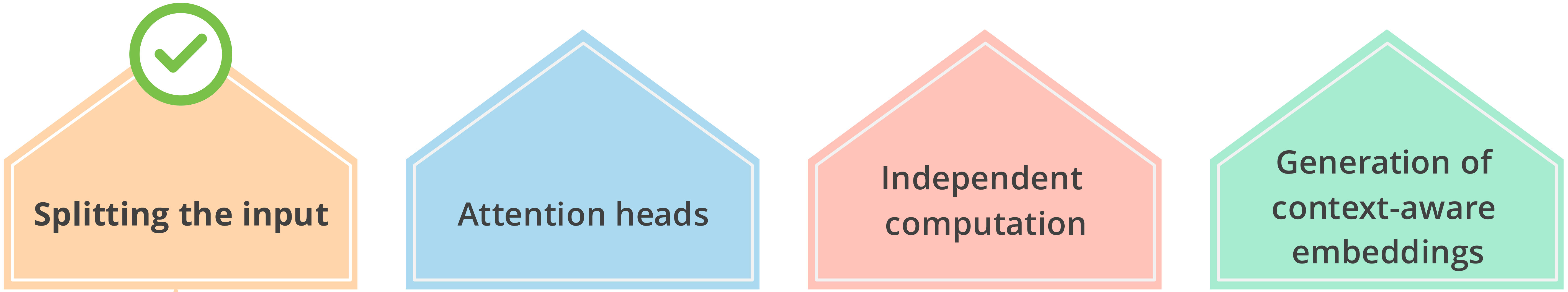


- Multi-head attention employs independent linear layers for keys (K), queries (Q), and values (V).
- It combines each attention head's output and processes it through a non-linear function.
- It enables the model to focus on different data positions.
- It treats inputs as sets of elements rather than as sequences.

h = Parallel attention layers or heads

Mechanics Behind Multi-Head Attention

Let’s understand the steps by considering a simple sentence: "The cat sat on the mat.”



- Assume you use a two-head attention mechanism.
- First, transform the input embeddings of each word in the sentence using two distinct linear transformations.
- This process yields two sets of transformed inputs.
- Each set provides a unique perspective on the sentence.

Mechanics Behind Multi-Head Attention

Splitting the input

✓ Attention heads

Independent computation

Generation of context-aware embeddings

- Each transformed input undergoes processing by a separate self-attention mechanism known as an **attention head.**
- The first attention head focuses on syntactic features, such as word order.
- The second attention head concentrates on semantic features, such as word meaning.

Mechanics Behind Multi-Head Attention

Splitting the input

Attention heads

✓ **Independent computation**

Generation of context-aware embeddings

- Every attention head operates independently.
- Each attention head calculates attention scores by comparing each word in its input with every other word.
- For instance, when processing **cat**:
    a. The first attention head might assign high scores to **sat** and **mat**, due to their syntactic relation to a **cat**.
    b. The second attention head could give a high score to **the**, commonly found before nouns.

Mechanics Behind Multi-Head Attention

- Splitting the input
- Attention heads
- Independent computation
- Generation of context-aware embeddings ✓

- Attention heads generate new embeddings for words using attention scores.
- For a word like **cat**, its new embedding is a weighted sum of all sentence word embeddings.
- These weights come from the attention scores.
- This method ensures that each word's embedding captures its meaning and sentence context.

Quick Check

Which component of Transformer models enhances their ability to understand the context by allowing various perspectives on the data?

A. Multi-head attention

B. Dividing attention heads

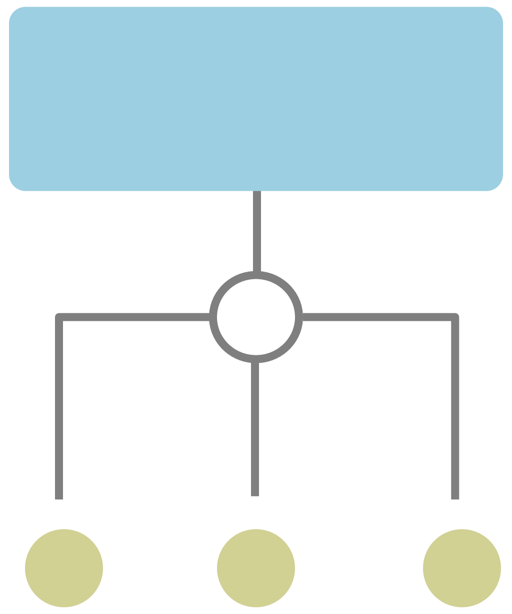
C. Attention mechanism

D. Transformer architecture

Introduction to Transformers

What Is Transformer?

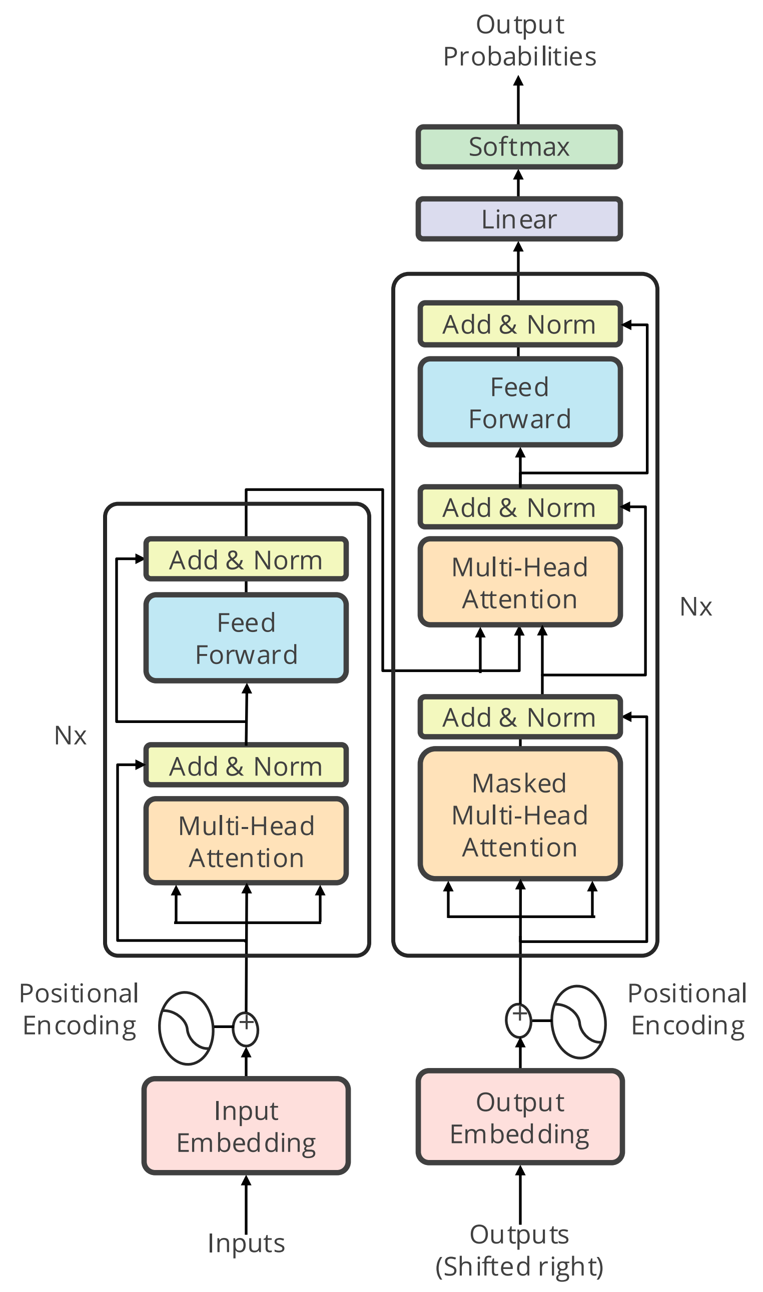
It is a type of sequence-to-sequence model that uses self-attention to process input sequences and generate output sequences.



A new architecture unlike traditional models relies entirely on attention mechanisms, (self-attention) discarding the need for recurrence and convolutions like traditional architecture does.

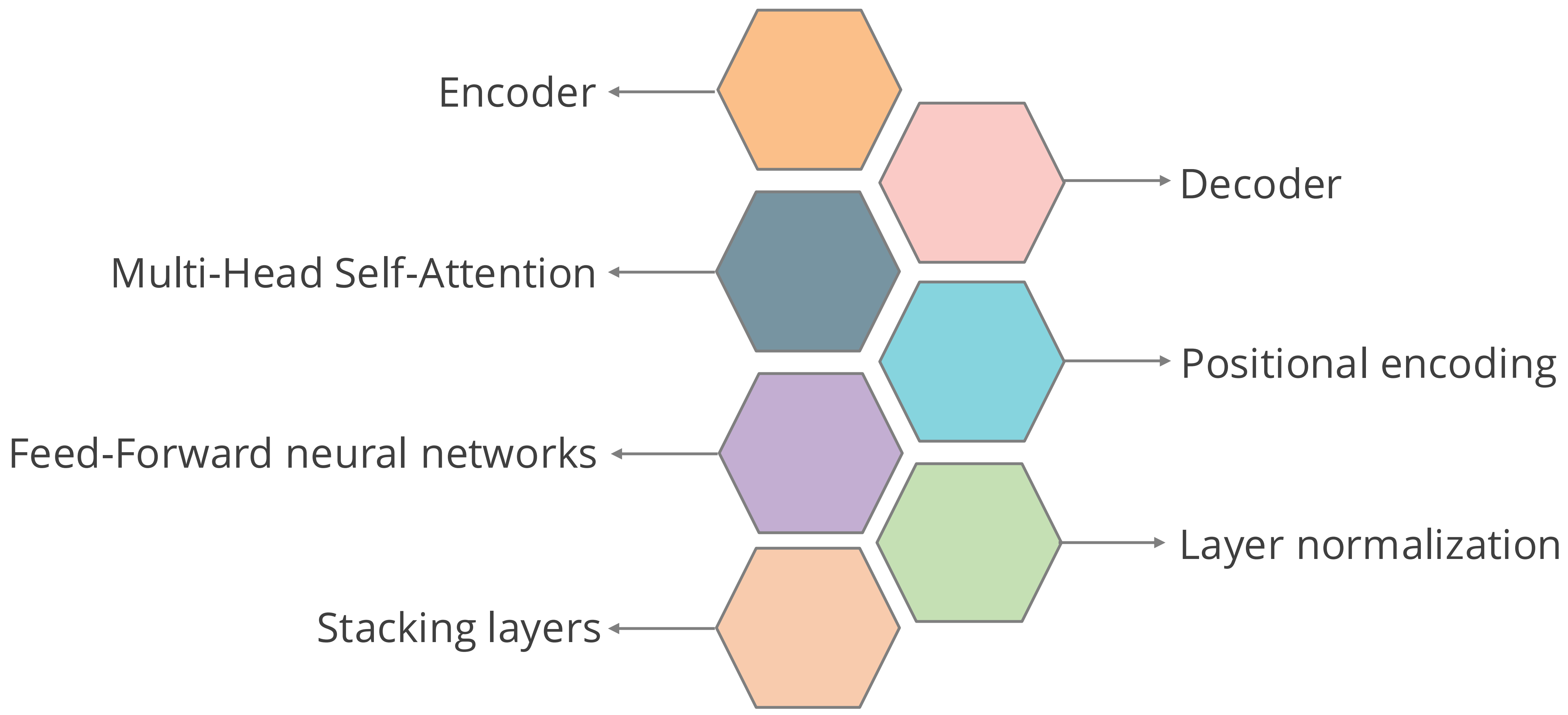
What Is Transformer?

The Transformer architecture has been widely used in NLP tasks such as machine translation, text generation, and language modeling.

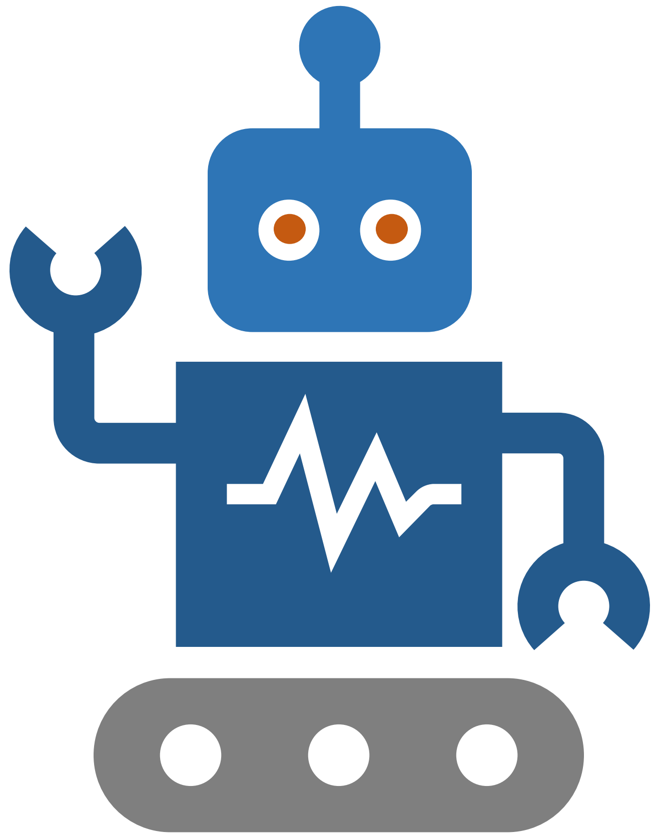


Components of Transformer

Following are the components of Transformer:



Practical Applications of Transformers



- **Medical Record Summarization**
  Simplify the process of summarizing lengthy medical records, helping healthcare professionals make more informed decisions in less time.

- **Sentiment Analysis in Trading**
  Traders use transformers for sentiment analysis of financial news and social media to gain insights into market trends and make data-driven decisions.

- **Recommendation Systems**
  Transformers power recommendation engines, offering personalized product suggestions based on user behavior and preferences.

Quick Check

Which component is NOT part of the Transformer model?

A. Recurrent Neural Layers

B. Encoder

C. Positional Encoding

D. Layer Normalization

Quick Check

What is a key advantage of using Transformers in sentiment analysis for financial trading?

A. They reduce the computational costs associated with trading algorithms.

B. They can predict future market prices with absolute certainty.

C. They utilize self-attention to understand context and sentiment from financial news.

D. They eliminate the need for manual data entry in trading systems.

Industry Use Case: Transformers

Problem Scenario

Context:

In the healthcare sector, managing vast quantities of patient data is an ongoing challenge. Physicians and healthcare professionals often need to sift through extensive medical records to extract critical information. This process is time-consuming, and there is a risk of important details being overlooked, leading to delayed diagnosis and treatment planning.

Problem Scenario

Challenges:

- **Time-consuming data review:** Healthcare providers spend a substantial amount of time reviewing lengthy medical records, affecting their efficiency and the time available for patient care.
- **Risk of missing key information:** The human review process can inadvertently miss important information, which can have critical implications for patient care.
- **Accuracy and consistency:** Ensuring that medical records are consistently reviewed is a significant challenge, as the quality of summaries can vary among healthcare professionals.

Solution with Transformers

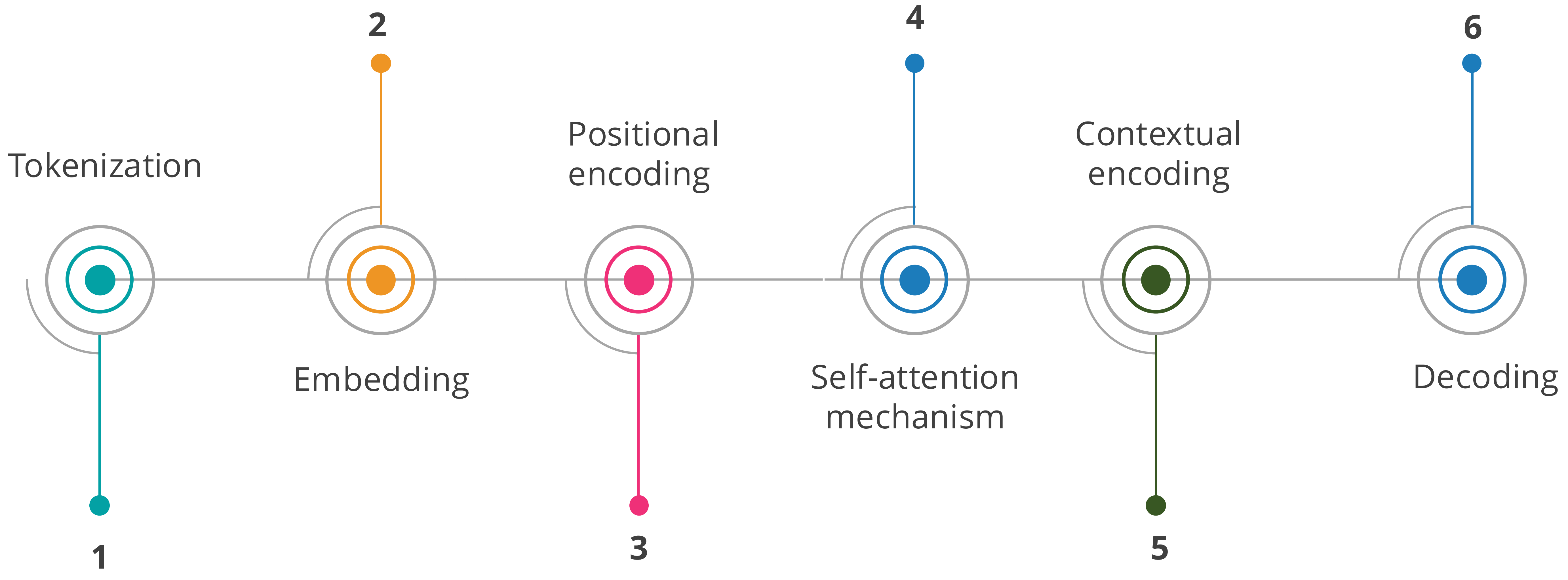
Solution:

To address these challenges, the healthcare industry has turned to Transformers, such as GPT (Generative Pre-trained Transformers) and BERT (Bidirectional Encoder Representations from Transformers) models. These models are specifically designed for natural language understanding and can analyze and summarize medical records with remarkable efficiency.

Text Generation with Transformer

Steps of Text Generation

Below are sequential stages utilized by Transformers for generating text:



Steps of Text Generation

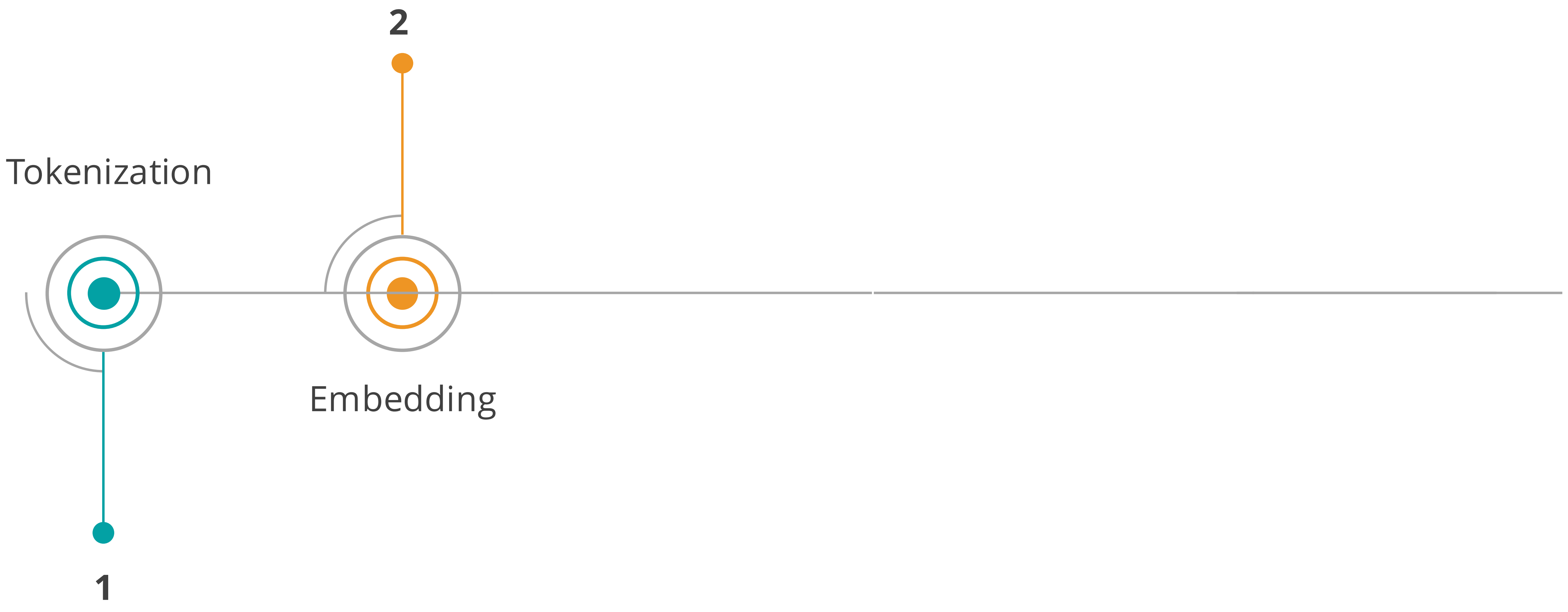
Let's take look at each step-in detail:

Tokenization

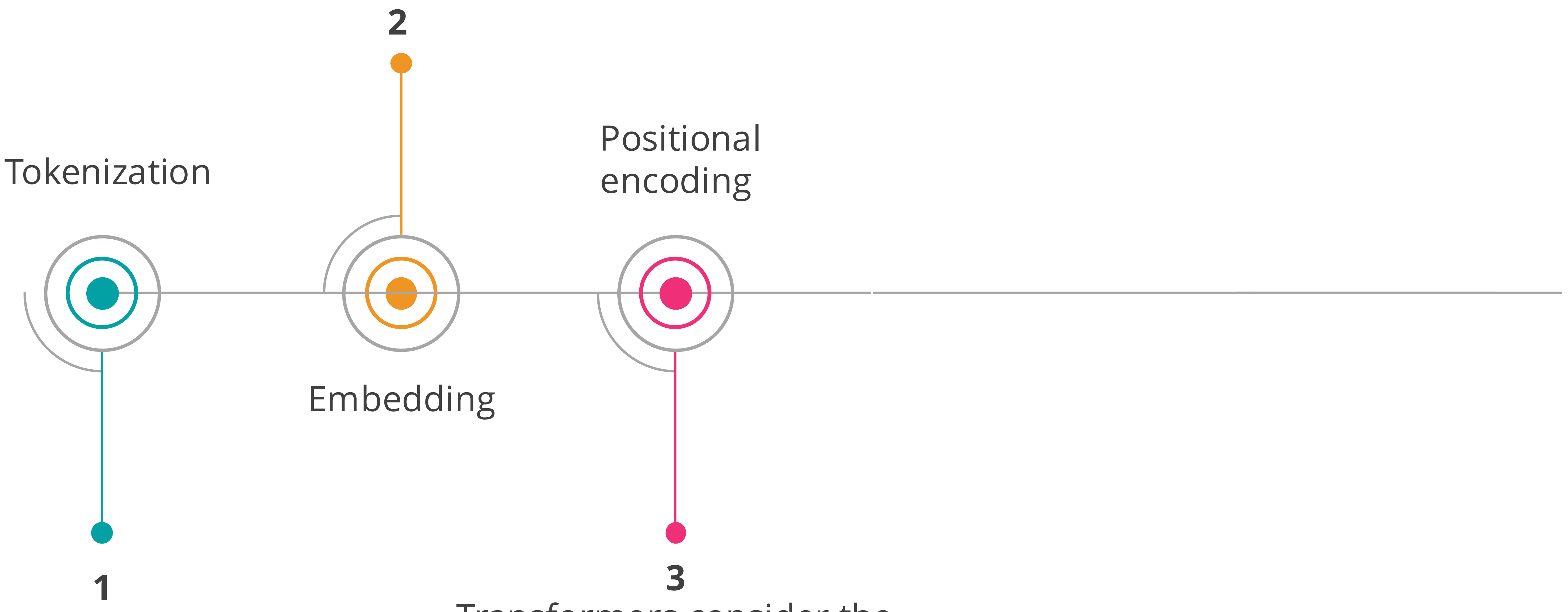
1

The input text is divided into smaller units called tokens, which can be words or subwords.

Steps of Text Generation

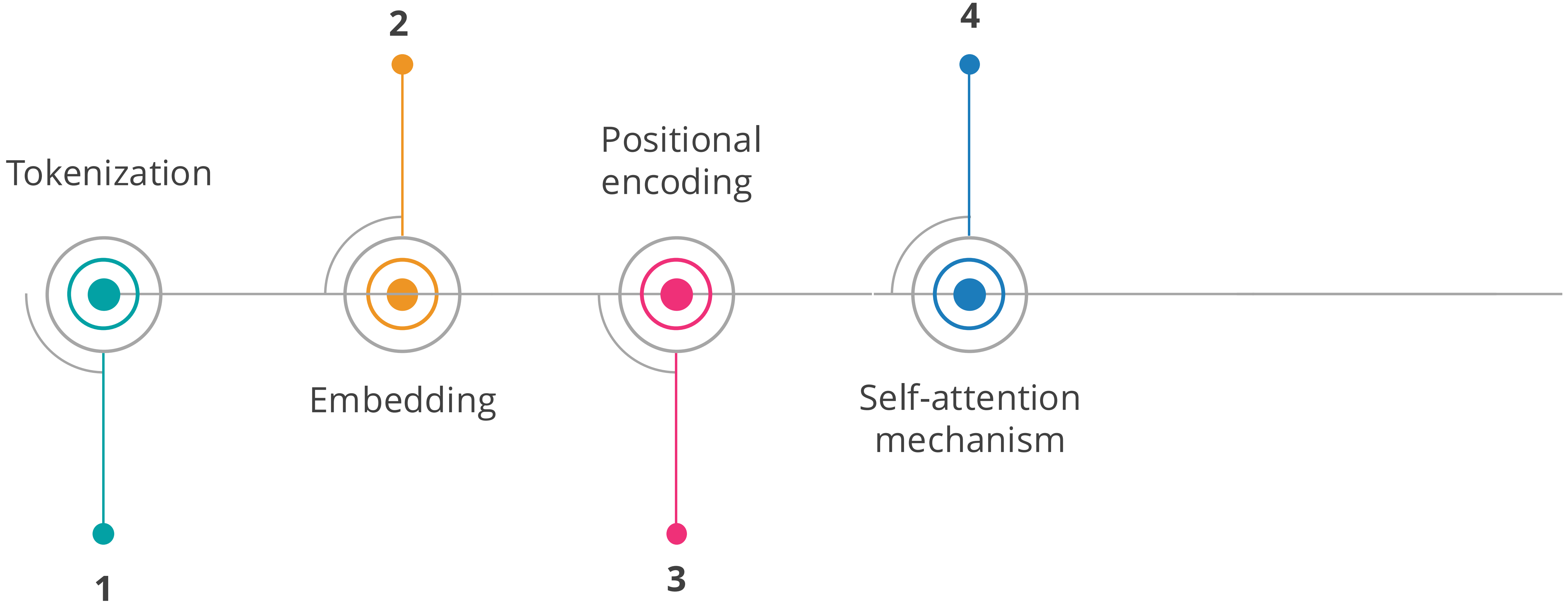

Each token is converted into a numerical vector representation, capturing its meaning.
2
Tokenization
Embedding
1

Steps of Text Generation



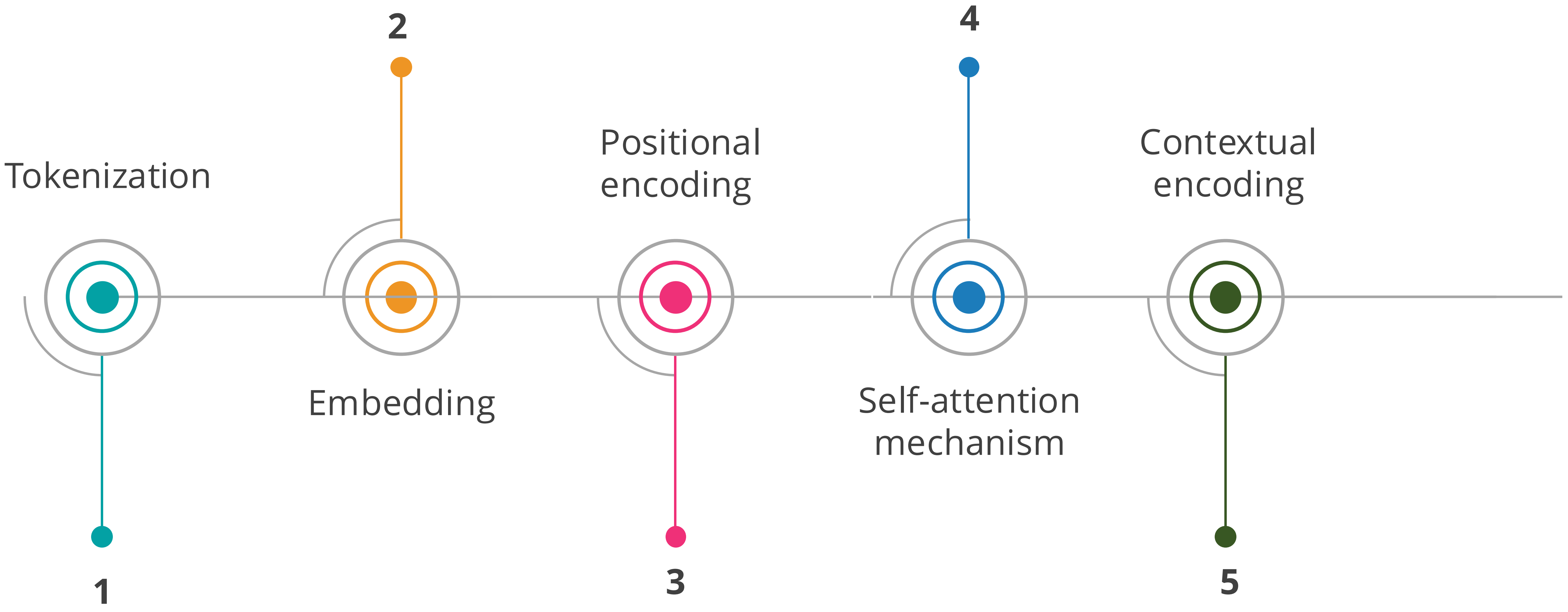
Transformers consider the order of tokens through positional encodings, ensuring context is maintained.

Steps of Text Generation


This step involves the multi-head self-attention mechanism to weigh the importance of each token concerning others in the sequence.
1
Tokenization
2
Embedding
3
Positional encoding
4
Self-attention mechanism

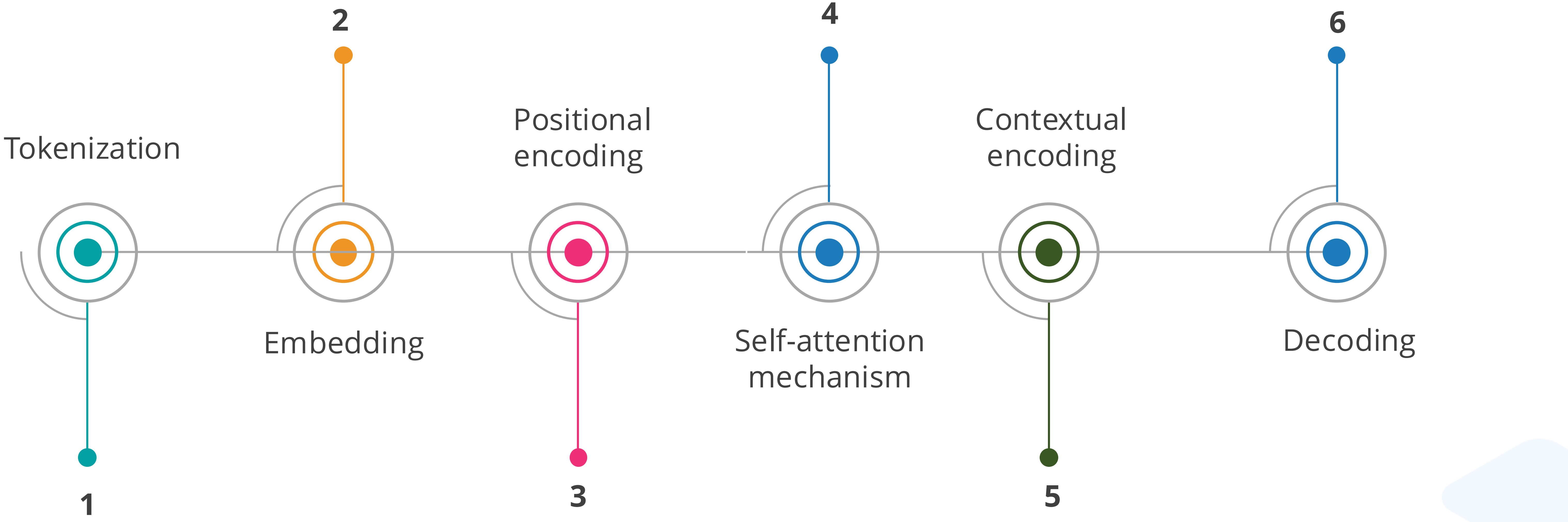
The model learns which words to focus on for each generated token.

Steps of Text Generation



The model processes the sequence, updating the token representations in each layer.

Steps of Text Generation


The model generates one token at a time, using the previous tokens and the context learned in previous layers to predict the next token.
1
Tokenization
2
Embedding
3
Positional encoding
4
Self-attention mechanism
5
Contextual encoding
6
Decoding

This process continues iteratively until the desired length of text is achieved.

Quick Check

Which step in the text generation process using transformers assigns meaning to each word by converting it into a numerical vector?

A. Tokenization

B. Embedding

C. Positional Encoding

D. Self-Attention Mechanism

Image Generation with Transformer

Evolution in Image Generation

Convolutional Neural Networks (CNNs) traditionally dominated image generation, but recently, Transformers have made remarkable advancements in this field.

Transformers operate on a grid-like structure that represents the pixels of the image.

Transformers employ a self-attention mechanism to analyze relationships between pixels and generate coherent images.

Image Generation Process

The image generation process in a Transformer-based model involves the following key steps:

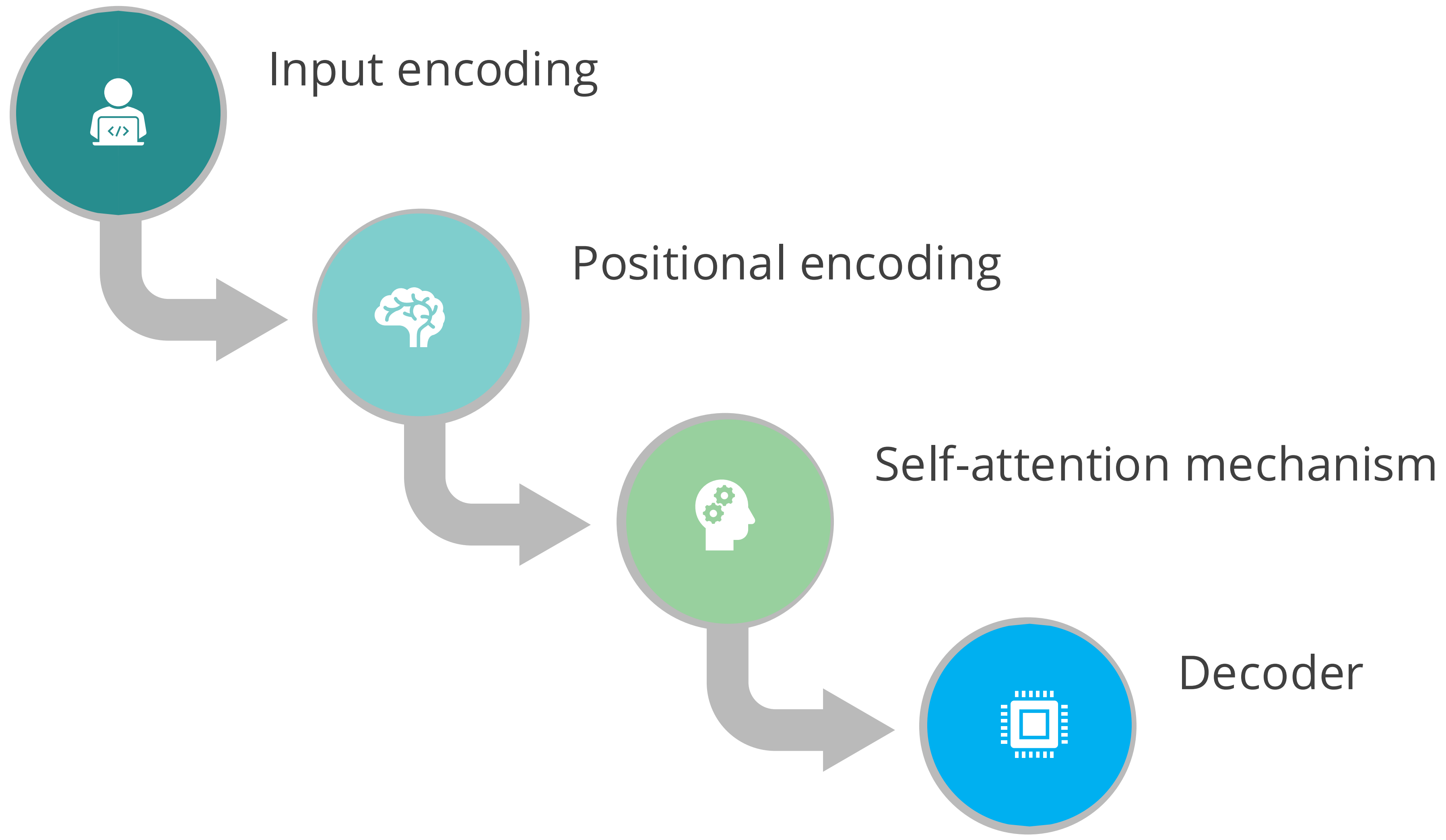
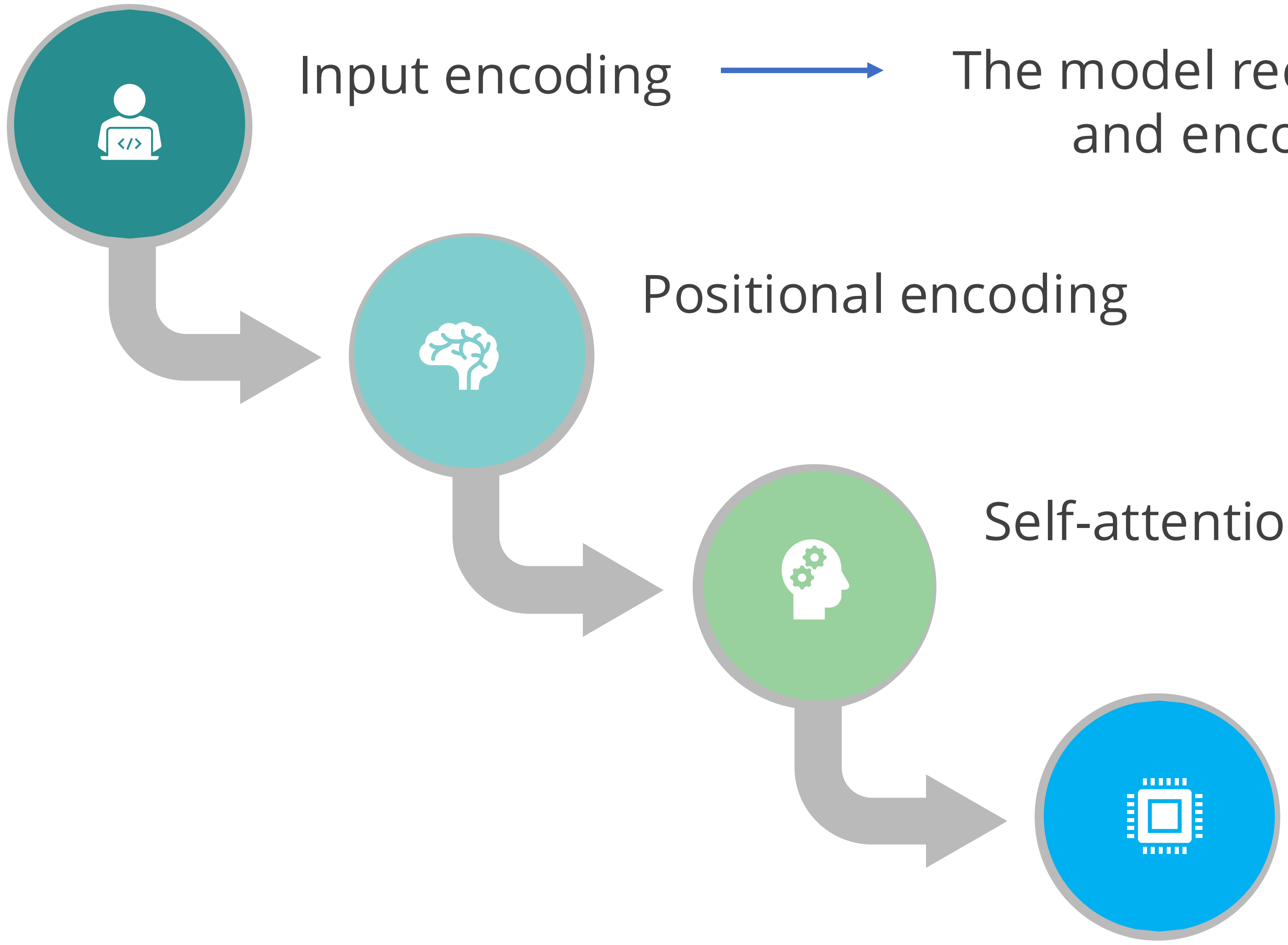


Image Generation Process



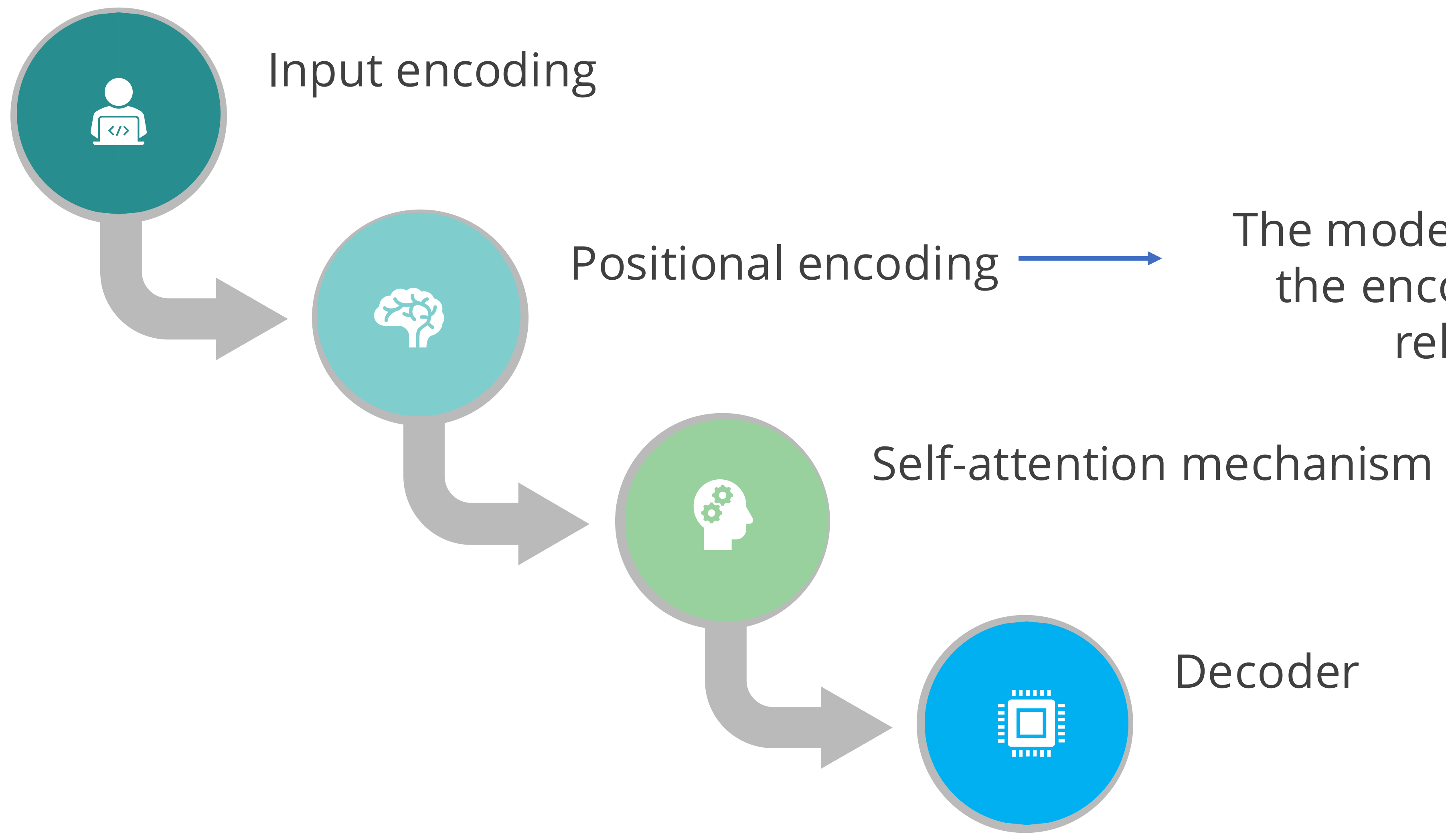
Input encoding → The model receives a prompt or an initial image and encodes it into numerical vectors.

Positional encoding

Self-attention mechanism

Decoder

Image Generation Process



The model introduces positional information to the encoded data to understand the spatial relationships between elements.

Image Generation Process

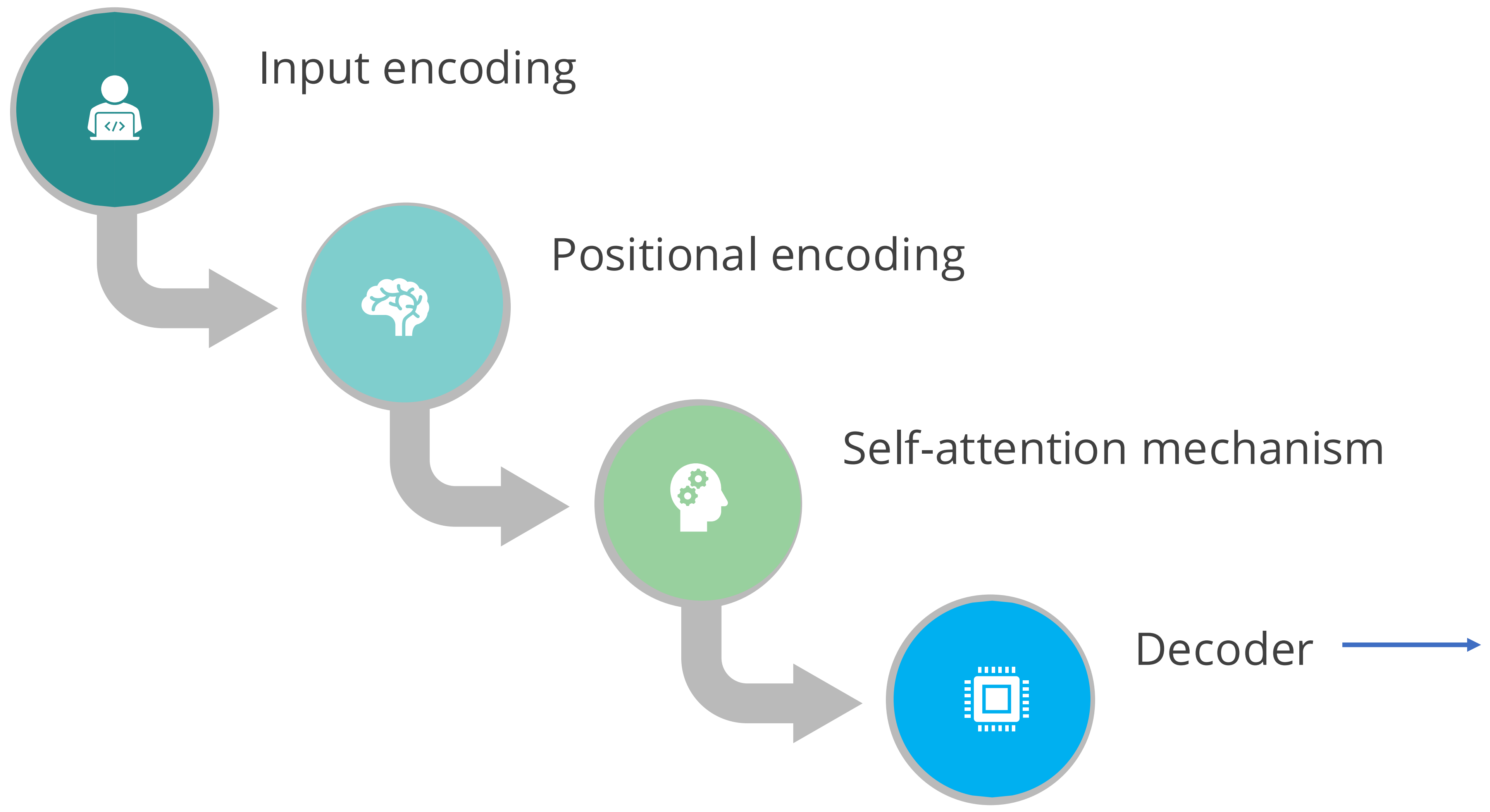
Input encoding

Positional encoding

Self-attention mechanism → The model employs self-attention to focus on different parts of the growing image.

Decoder

Image Generation Process



The model, like text generation, employs an iterative decoding method. It predicts one portion of the image at a time and blends it with the present image.

This procedure is repeated until the full image is created.

Demo: Transformer Applications

Duration: 10 minutes

Overview:
In the realm of natural language processing, understanding and generating human-like text are crucial challenges. This demo addresses two core issues which can be easily solved by the transformer:

Text classification: Determining the sentiment of a piece of text (positive or negative).
Text generation: Generating contextually relevant and coherent text based on given input.

Note

Please download the solution document from the Reference Material Section and follow the Jupyter Notebook for step-by-step execution.

Quick Check

How do Transformers differ from Convolutional Neural Networks (CNNs) in the context of image generation?

A. Transformers use grid-like pixel structures, while CNNs do not.

B. Transformers replace the need for data augmentation, unlike CNNs.

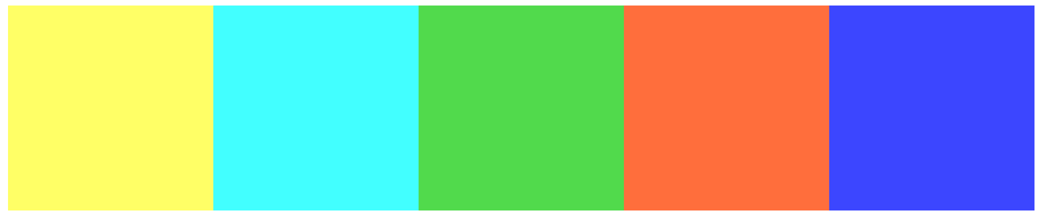
C. Transformers employ self-attention to analyze pixels, whereas CNNs rely on filter-based convolutions.

D. Transformers focus on reducing image overfitting, a problem commonly associated with CNNs.

Transformer Based Model Example

DALL-E

DALL-E is an artificial intelligence program developed by OpenAI.



- DALL-E is a famous image generation model that uses a 12-billion parameter GPT-3 architecture for creating images from textual descriptions.
- It demonstrates how Transformers can create diverse and contextually relevant images.

GPT

GPT is a series of language processing AI models developed by OpenAI

- In the context of GPT (Generative Pre-trained Transformers) models, the Transformer architecture is combined with a large language model (LLM) to create a powerful and versatile AI system
- GPT models are pre-trained on large datasets of unlabeled text and can generate novel human-like content
- GPT models use a decoder-only architecture, which means they can generate text without the need for an encoder

LLaMa

LLaMA, or Language Model for Many Applications, is a Transformer-based language model developed by Meta AI.

- The LLaMA (Large Language Model Meta AI) model uses Transformer architectures for NLP tasks.
- It uses public data for pre-training and can be fine-tuned for specific use cases.
- It can be used for tasks such as text generation, translation, summarization, rewriting content, and classification and categorization

BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers, developed by researchers at Google.

BERT

(Bidirectional Encoder Representations from Transformers)

- BERT is a state-of-the-art model for NLP that uses the Transformer architecture.
- It enables models to process text in a bidirectional manner, from start to finish and from finish to start.
- This capability called bidirectionality, allows the model to understand the context and ambiguity in language more effectively.

Quick Check

Which generative AI model is specifically designed for creating images from textual descriptions?

A. DALL-E

B. GPT models

C. LLMA

D. BERT

Guided Practice

Overview **Duration: 20 minutes**

This activity aims to enhance practical skills in advanced AI technologies through various challenges. It covers the implementation of attention mechanisms in Generative AI models, troubleshooting Large Language Models, utilizing multi-head attention in transformers, and employing transformer models for generating text and images.

Additionally, it involves analyzing and selecting appropriate models like Dall-e, GPT, LLMA, and BERT for specific tasks. This exercise serves as an opportunity to deepen understanding and proficiency in these cutting-edge AI domains.

Key Takeaways

- Attention allows models to handle long sentence effectively
- Self-attention empowers models to attend to different input segments dynamically
- Transformer is a sequence-to-sequence model that uses self-attention to process input sequences and generate output sequences
- GPT models use a decoder-only architecture, which means they can generate text without the need for an encoder

Q&A