Large Language Models

Artificial Intelligence

- What is Artificial Intelligence (AI) ?

Artificial intelligence is the science and engineering of making intelligent machines, especially intelligent computer programs.

It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself tomethods that are biologically observable.

📖 J. McCarthy (2007). "What is artificial intelligence?" Retrieved from https://oreil.ly/C7sja and https://oreil.ly/n9X8O.

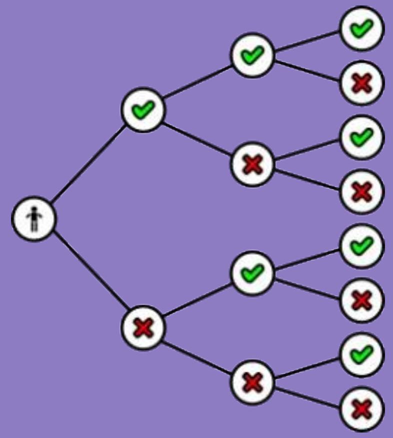
Artificial Intelligence: Learning Techniques

Artificial Intelligence

Methods capable of imitating human behavior

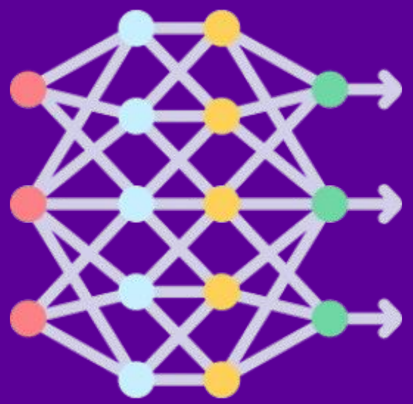
Machine Learning

Methods capable of automaticalmly learning from data



Deep Learning

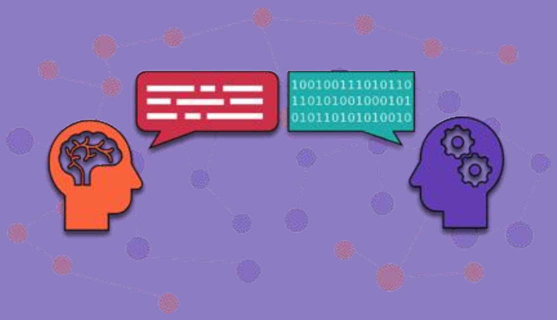
Use of deep neural networks



Artificial Intelligence: Application Domain

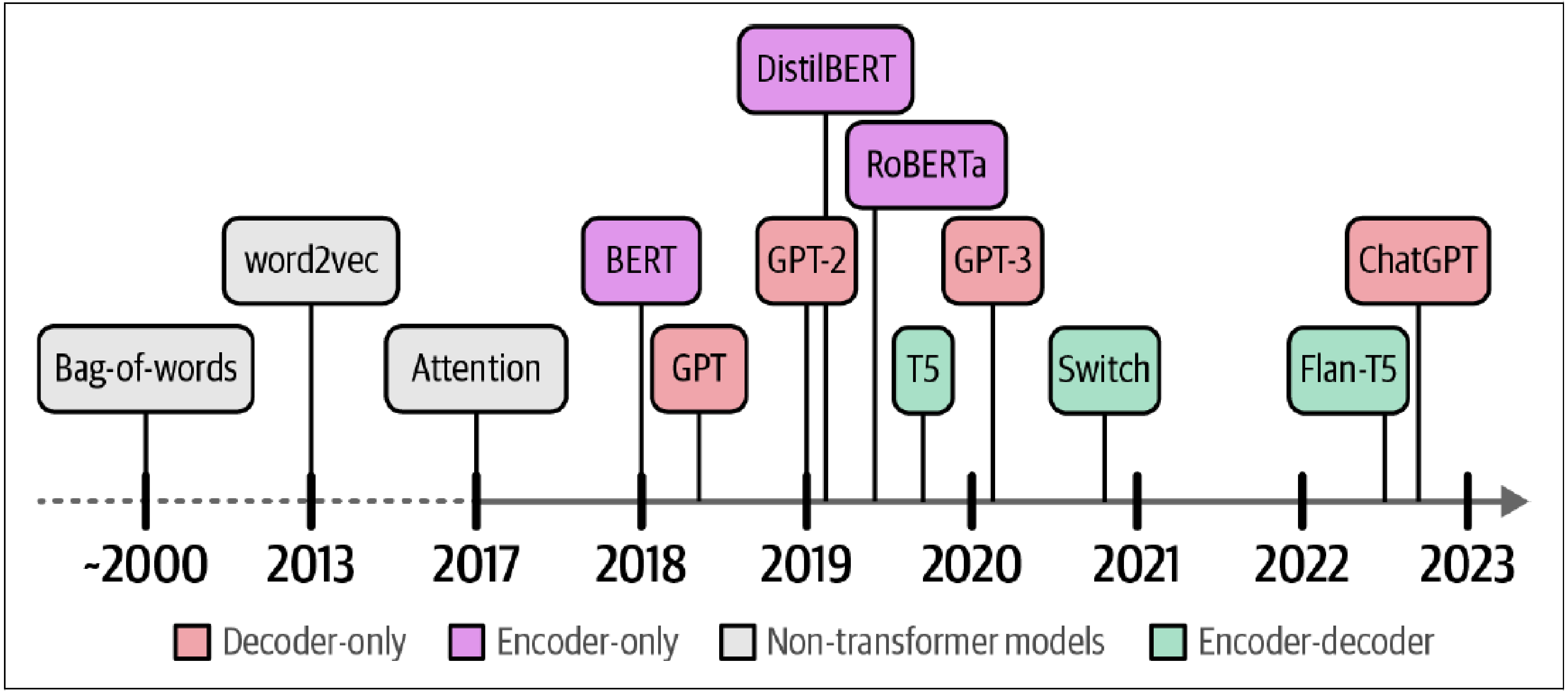
Artificial Intelligence

Natural Language Processing



Computer Vision

A Recent History of Language AI



📖 Alammar, J., & Grootendorst, M.(2024). Hands-On Large Language Models. O'Reilly Media.

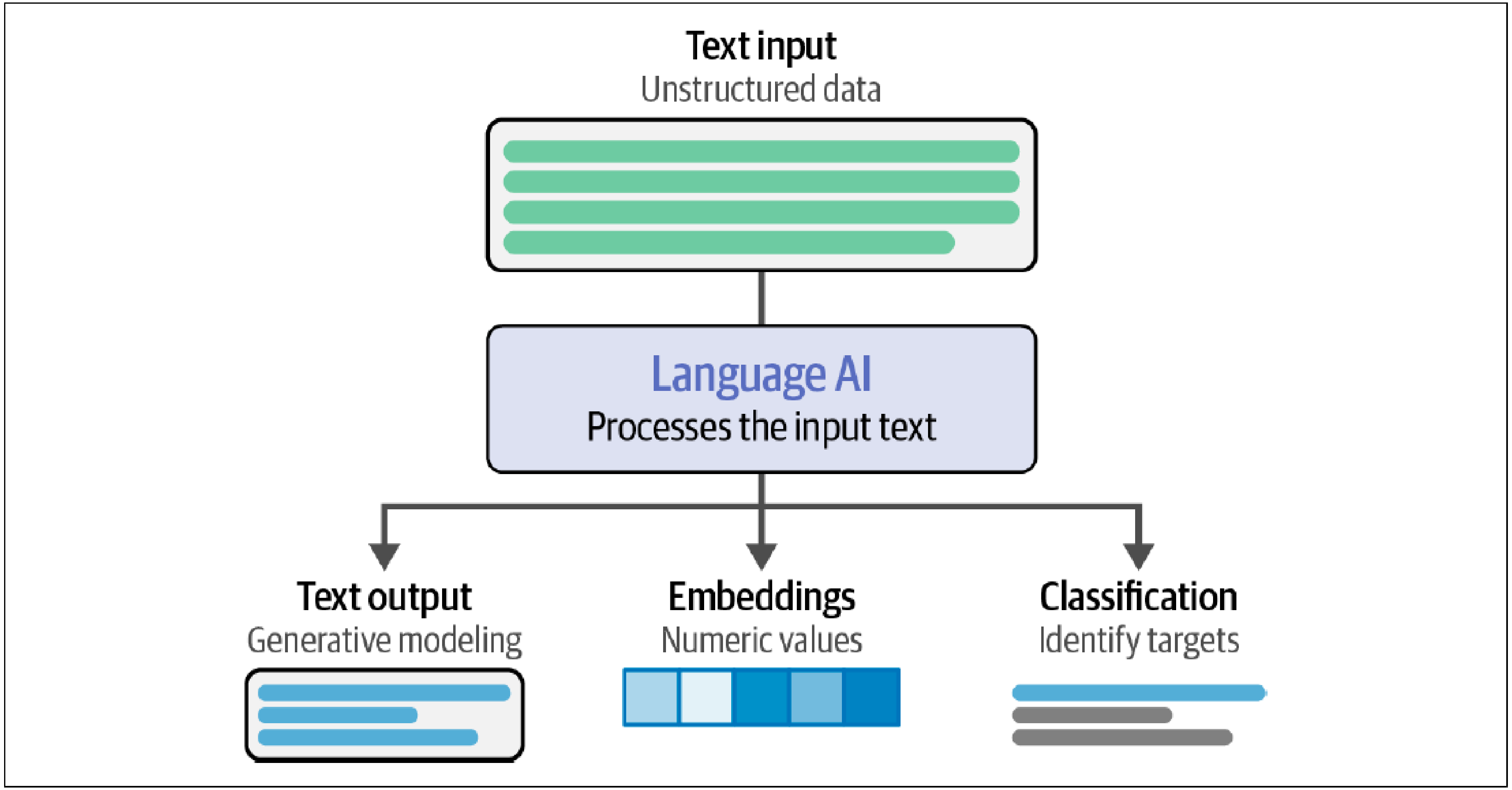
Artificial Intelligence: Natural Language Processing

- What is Natural Language Processing (NLP) ?

Language AI refers to a sunfield of AI that focuses on developing technologies capable of understanding, processing, and generating human language. The term Language AI can often be used interchangeable with **Natural Langage Processing (NLP).**

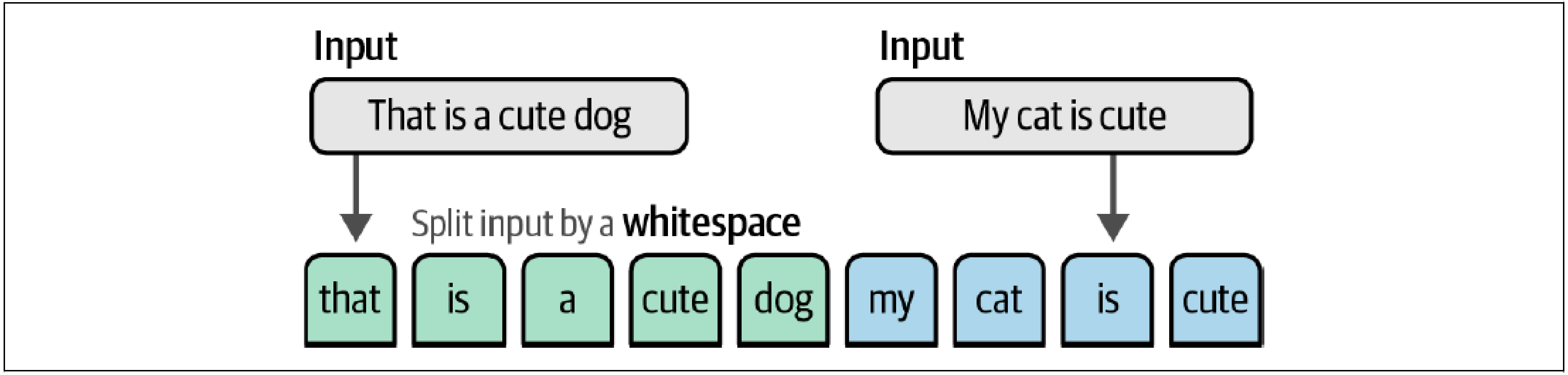
📖 Alammar, J., & Grootendorst, M.(2024). Hands-On Large Language Models. O'Reilly Media.

A Recent History of Language AI



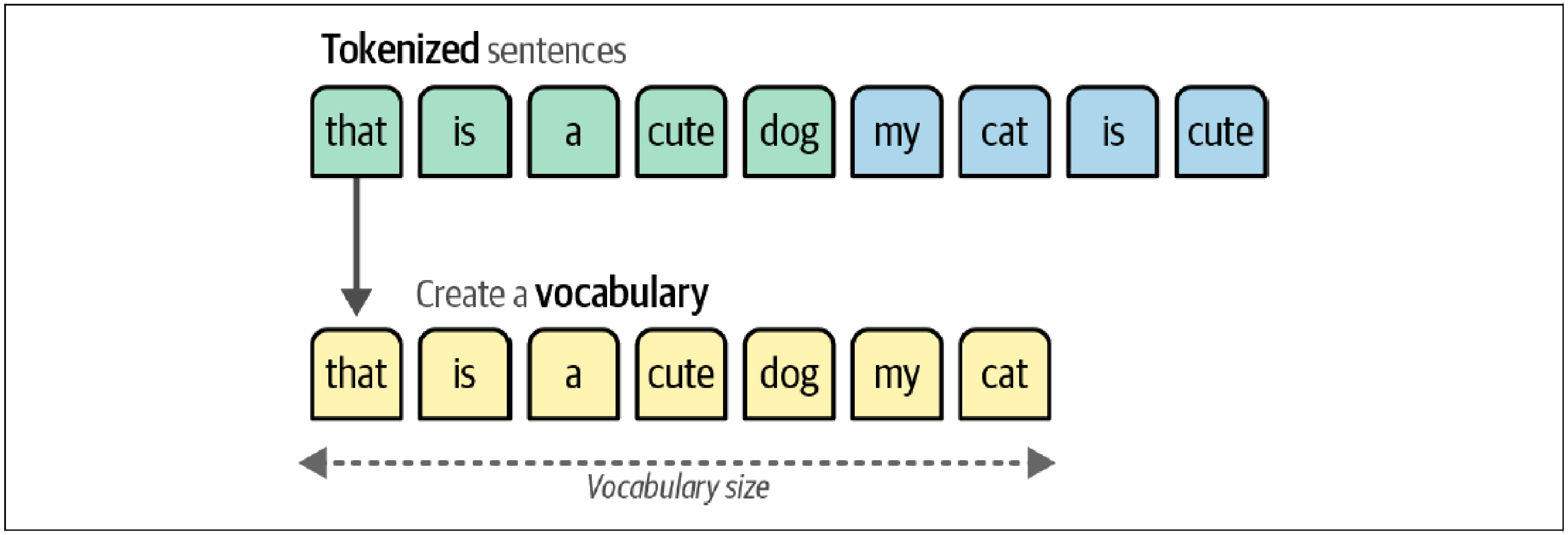
📖 Alammar, J., & Grootendorst, M.(2024). Hands-On Large Language Models. O'Reilly Media.

Representing Language as a Bag-of-Words



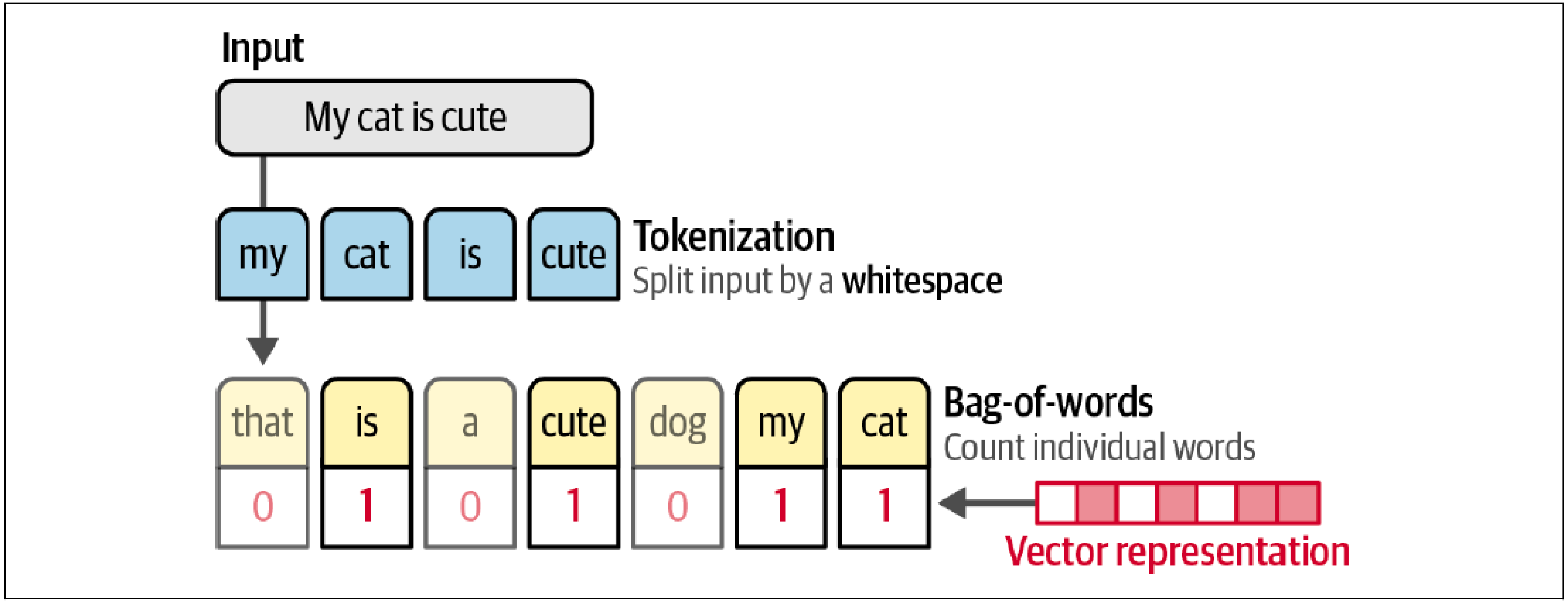
📖 Alammar, J., & Grootendorst, M.(2024). Hands-On Large Language Models. O'Reilly Media.

Representing Language as a Bag-of-Words



📖 Alammar, J., & Grootendorst, M.(2024). Hands-On Large Language Models. O'Reilly Media.

Representing Language as a Bag-of-Words



📖 Alammar, J., & Grootendorst, M.(2024). Hands-On Large Language Models. O'Reilly Media.

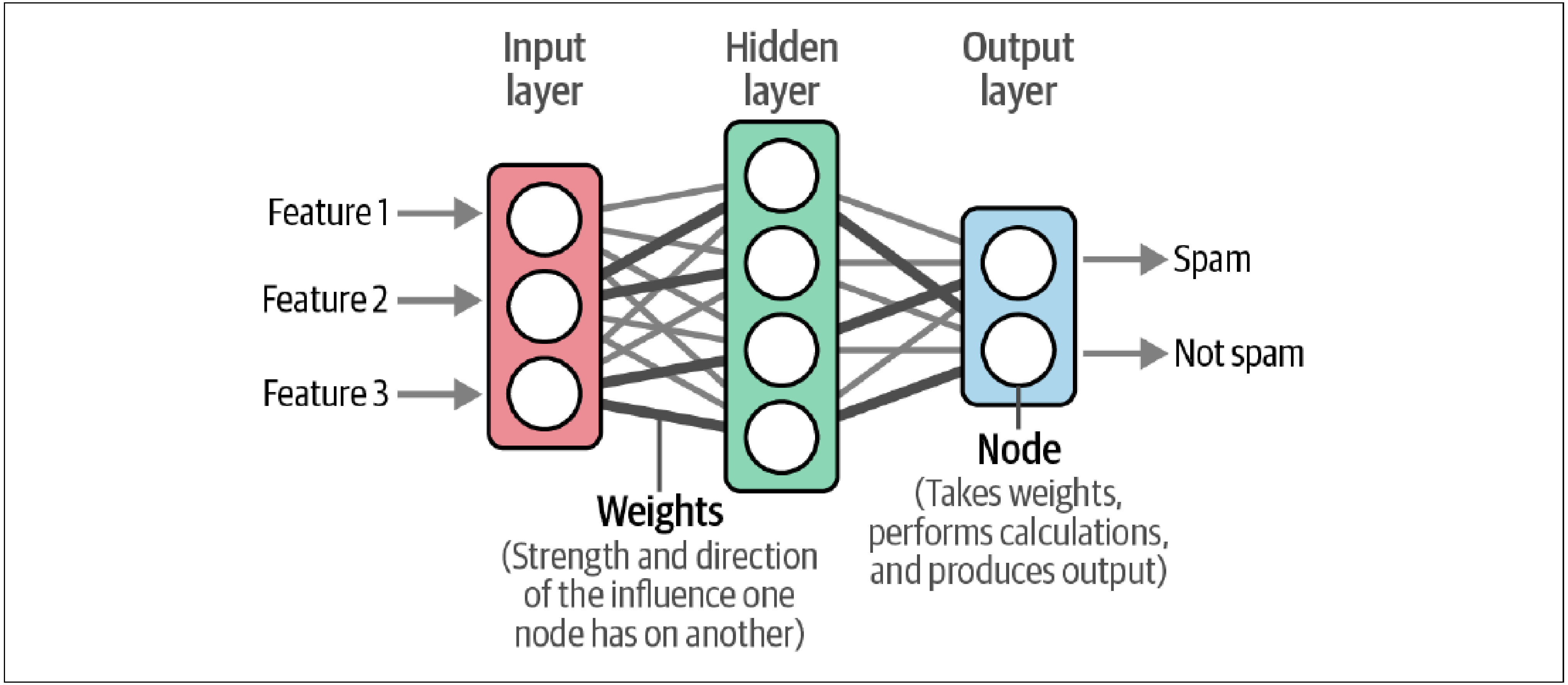
Natural Langage Processing

Better Representations with Dense Vector Embeddings

Better Representation with Dense Vector Embeddings

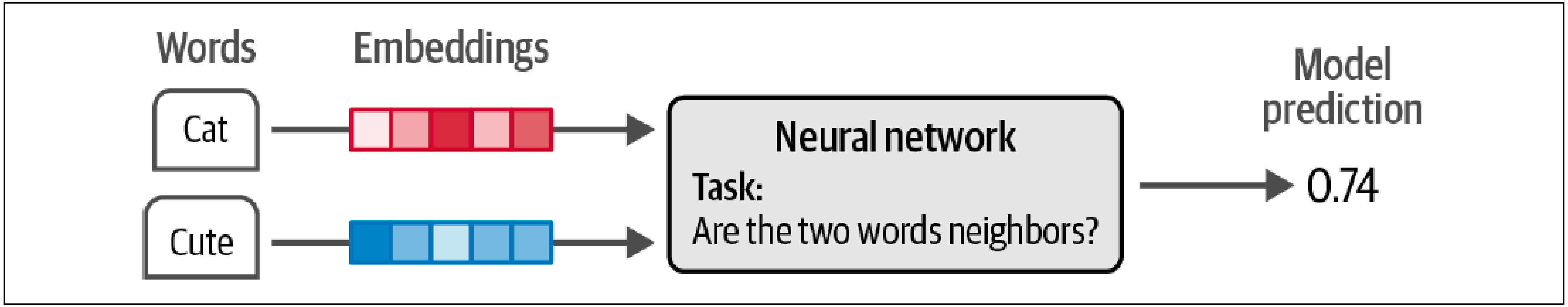
- Bag-of-words ignores the semantic nature, or meaning, of text.
- Word2vec was one of the first succesful attempts at capturing the meaning of text in embeddings.
- **Embeddings** are vector representations of data that attempt to capture its meaning.

Better Representation with Dense Vector Embeddings

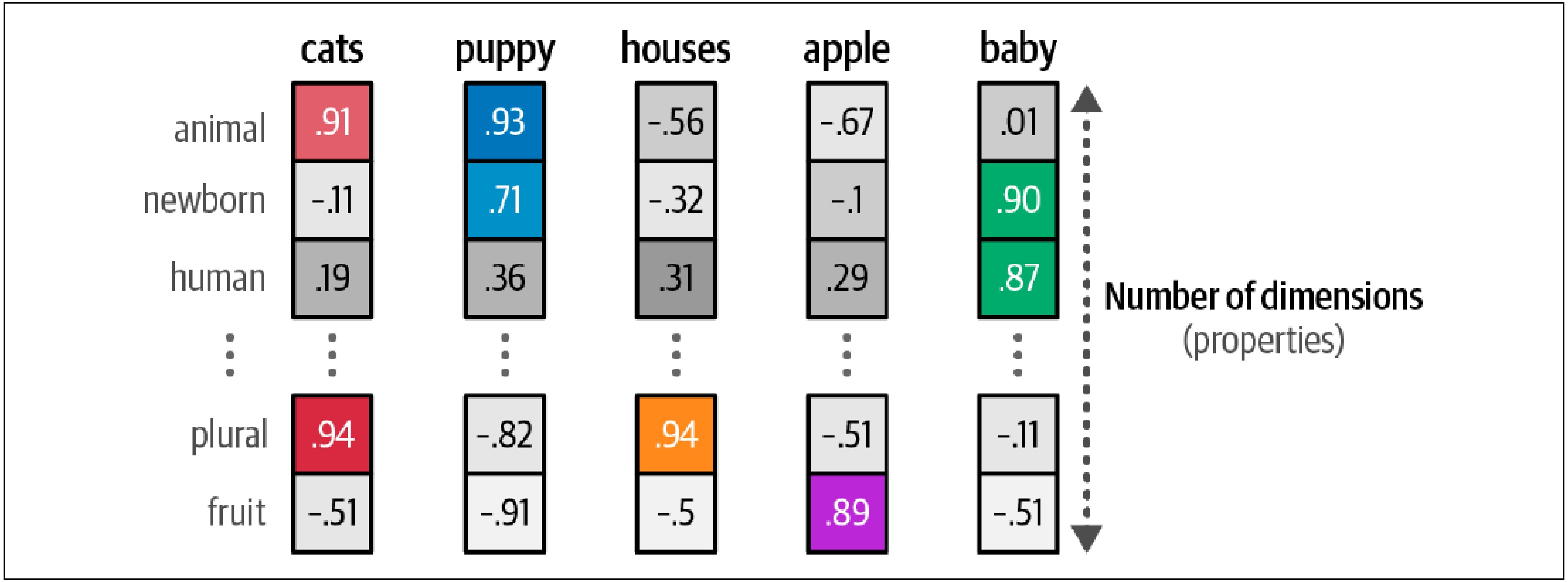


📖 Alammar, J., & Grootendorst, M.(2024). Hands-On Large Language Models. O'Reilly Media.

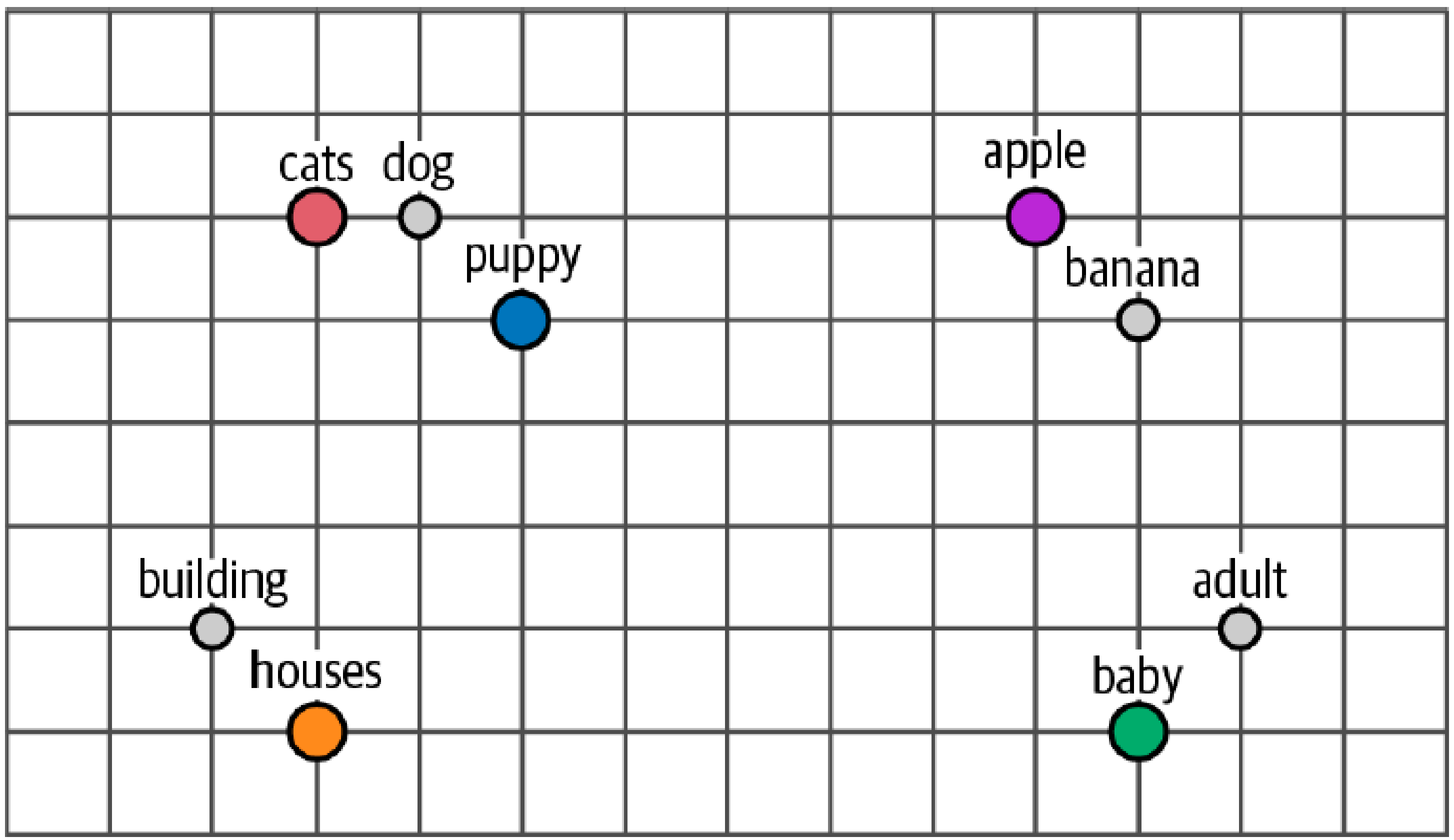
Better Representation with Dense Vector Embeddings



Better Representation with Dense Vector Embeddings



Better Representation with Dense Vector Embeddings



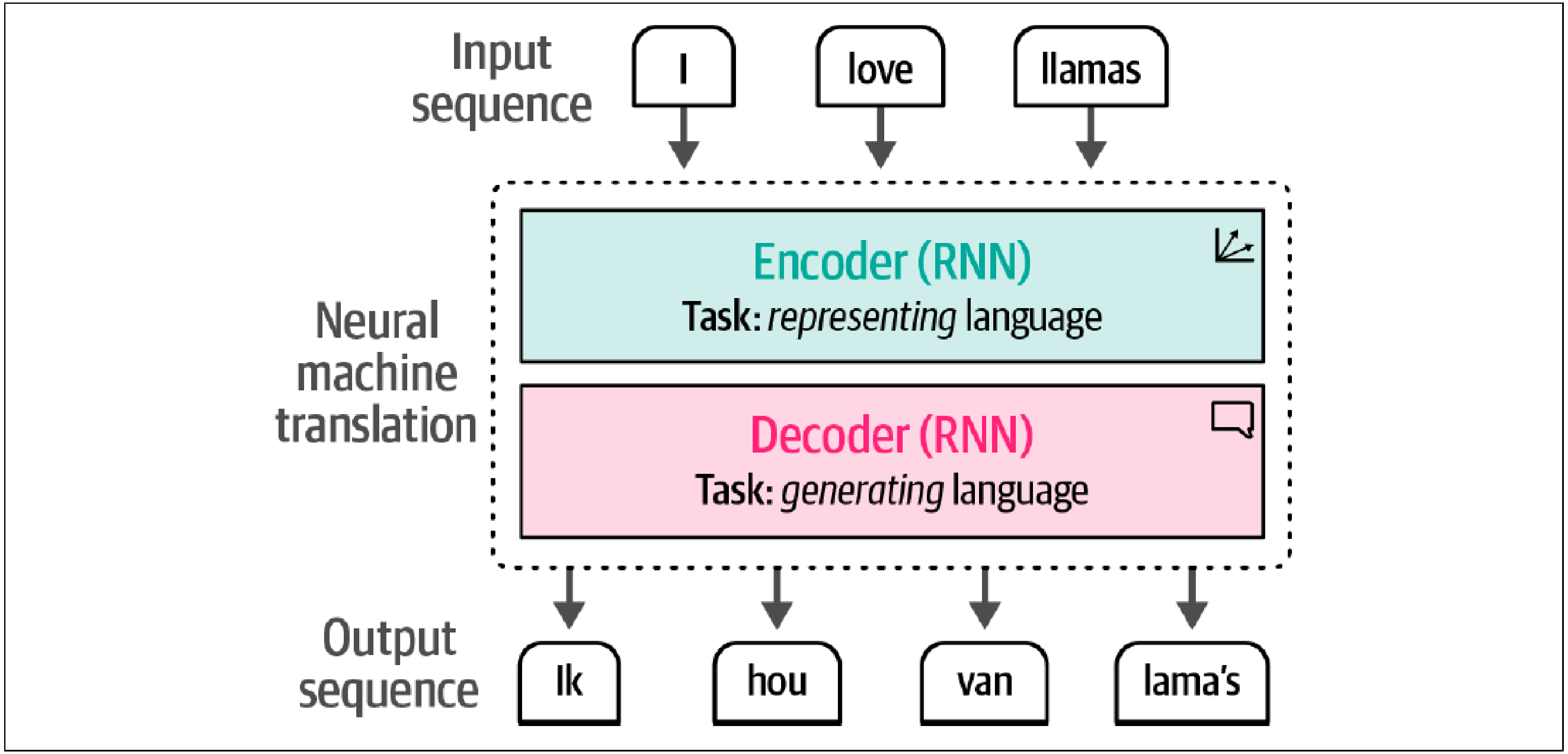
Natural Langage Processing

Encoding and Decoding Context

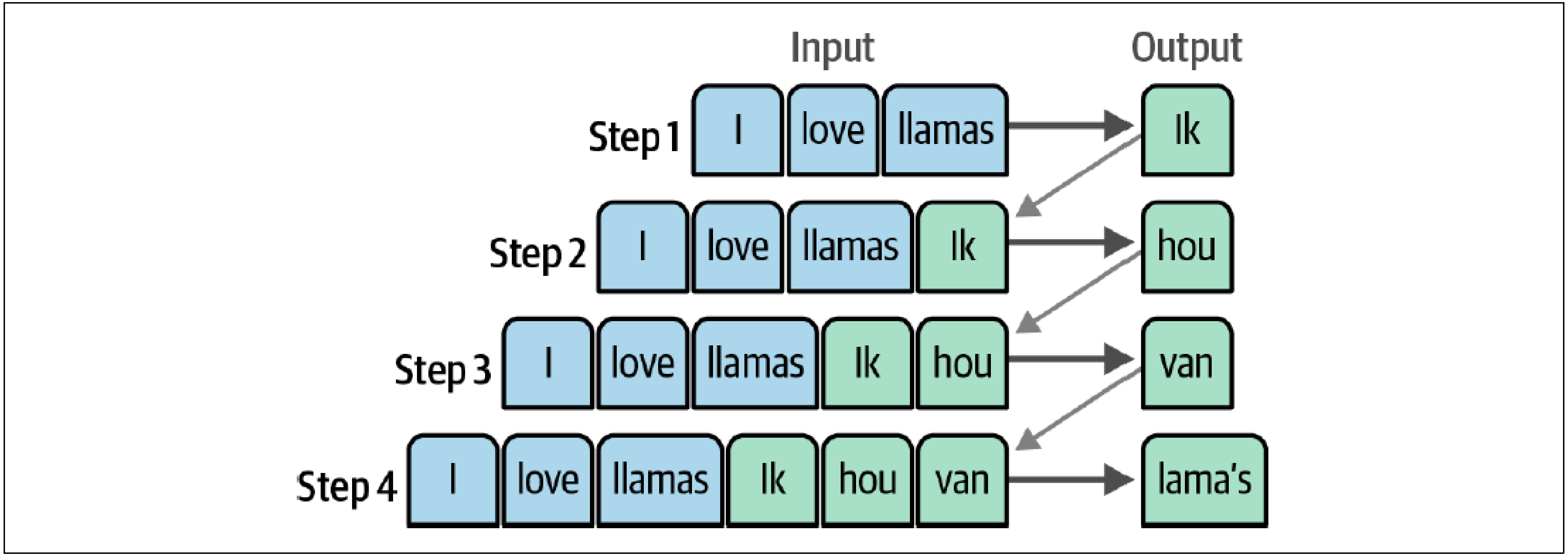
Encoding and Decoding Context with Attention

- Word2vec creates static, downloadable representations of words.
- For instance, the word '**bank**' can refer to both a **financial bank** as well as the **bank of a river**.
- Its meaning, and therefore its embeddings, should change **depending on the context**.

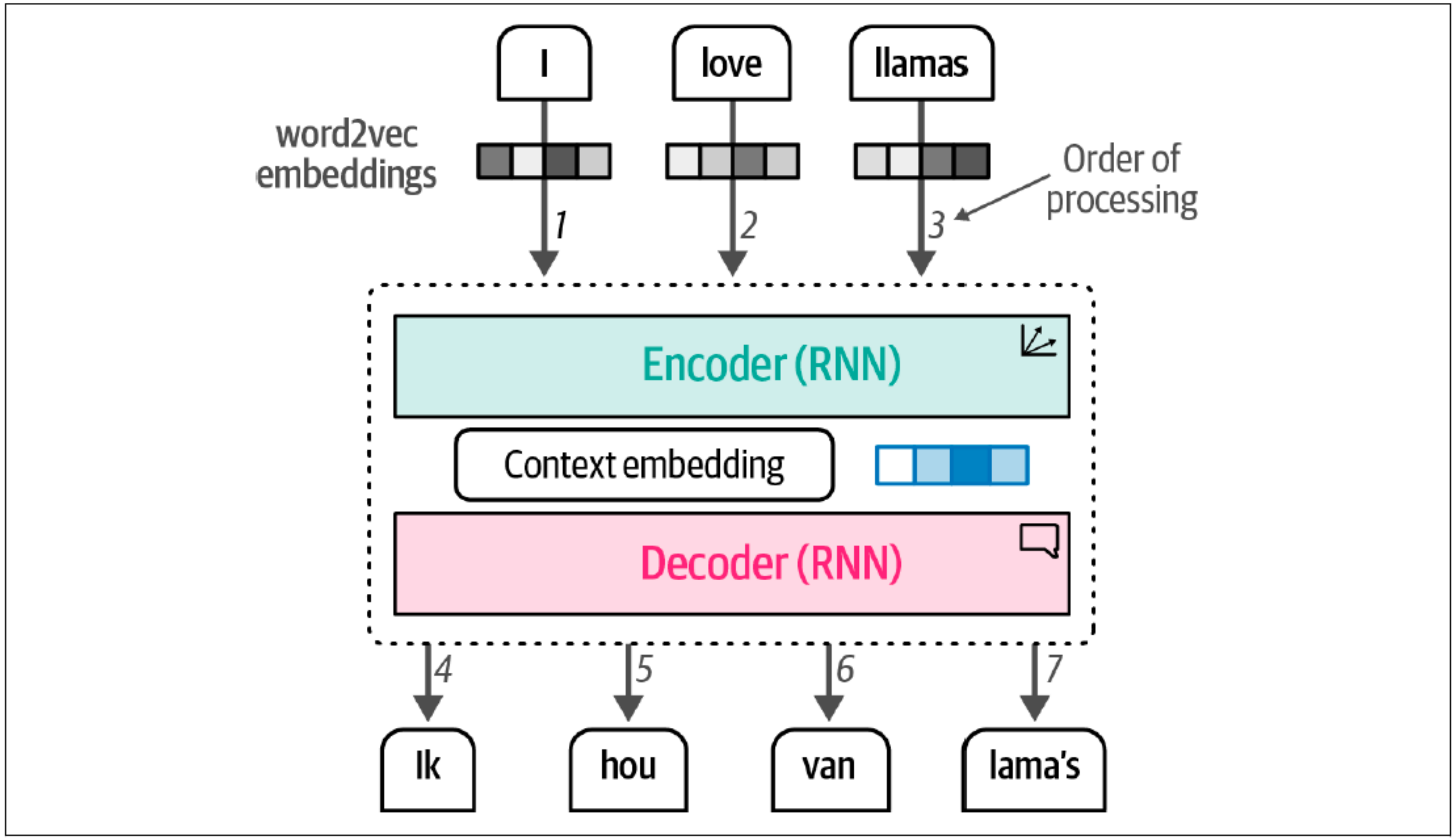
Encoding and Decoding Context



Encoding and Decoding Context



Encoding and Decoding Context



Encoding and Decoding Context

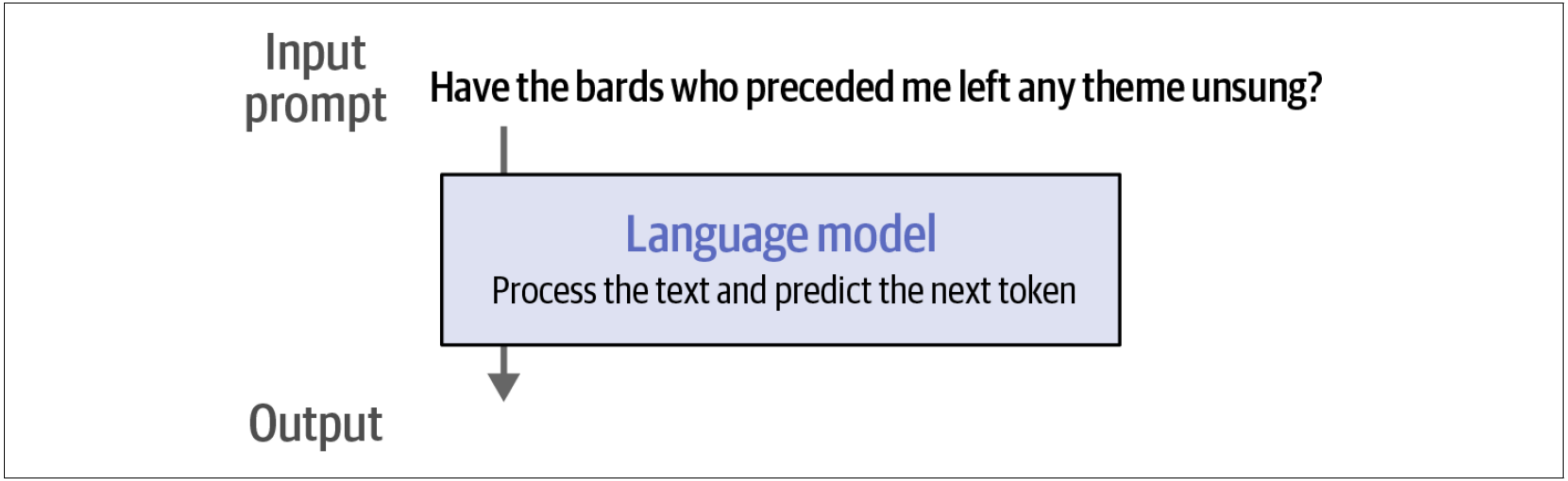
- This context embedding, however, makes it difficult to deal with longer sentences since it is merely a single embedding representing the entire input.

- In 2014, a solution called **attention** was introduced that highly improved upon the original architecture

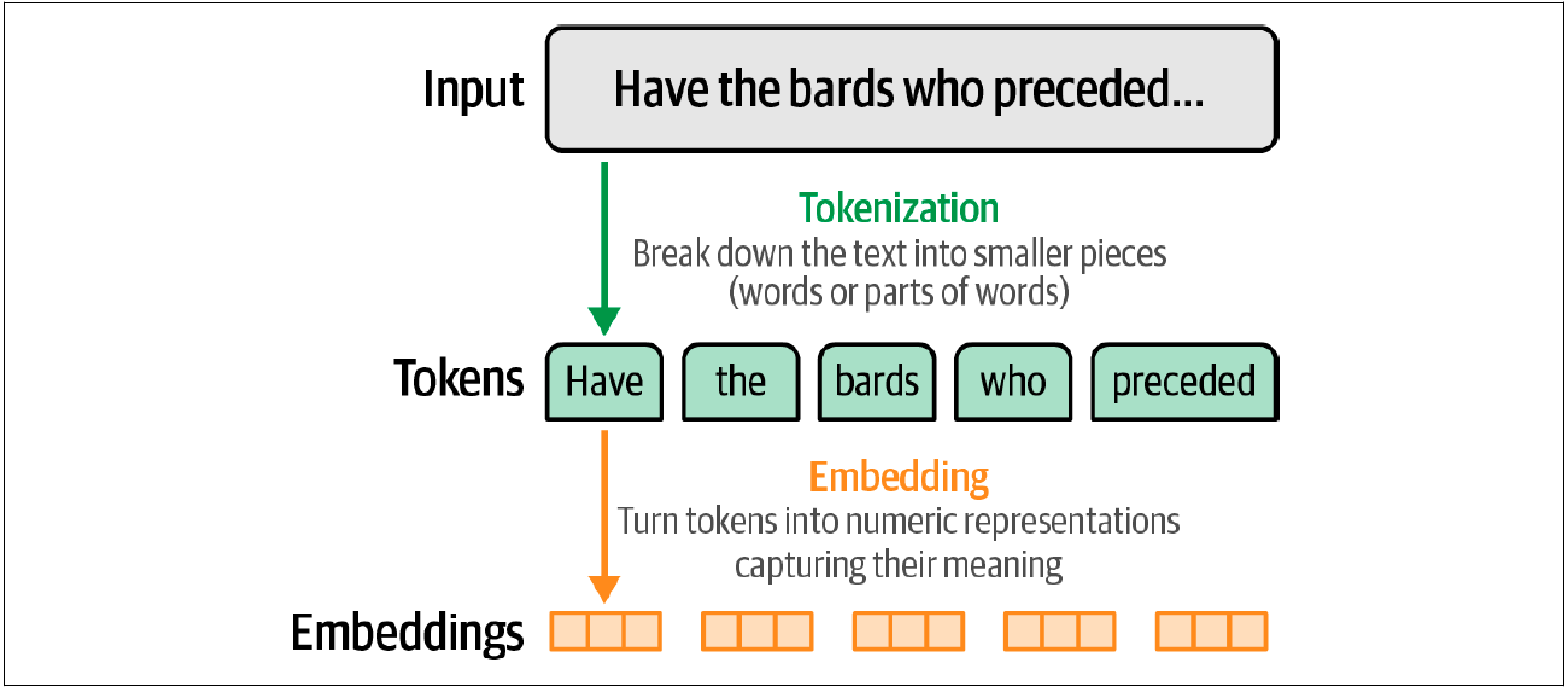
- **Attention mechanisms** play a crucial role in transformers.

Large Language Models: Tokens and Embeddings

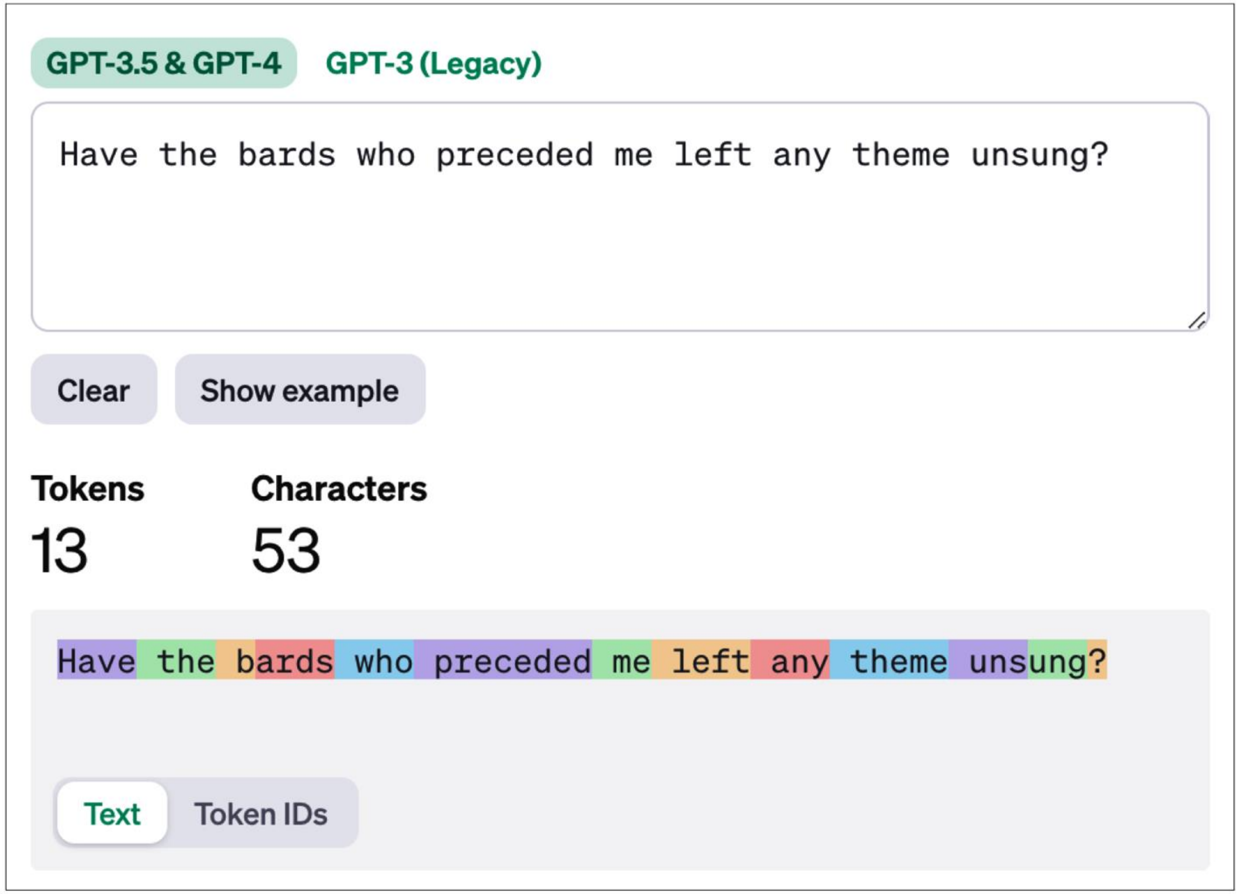
Tokens and Embeddings



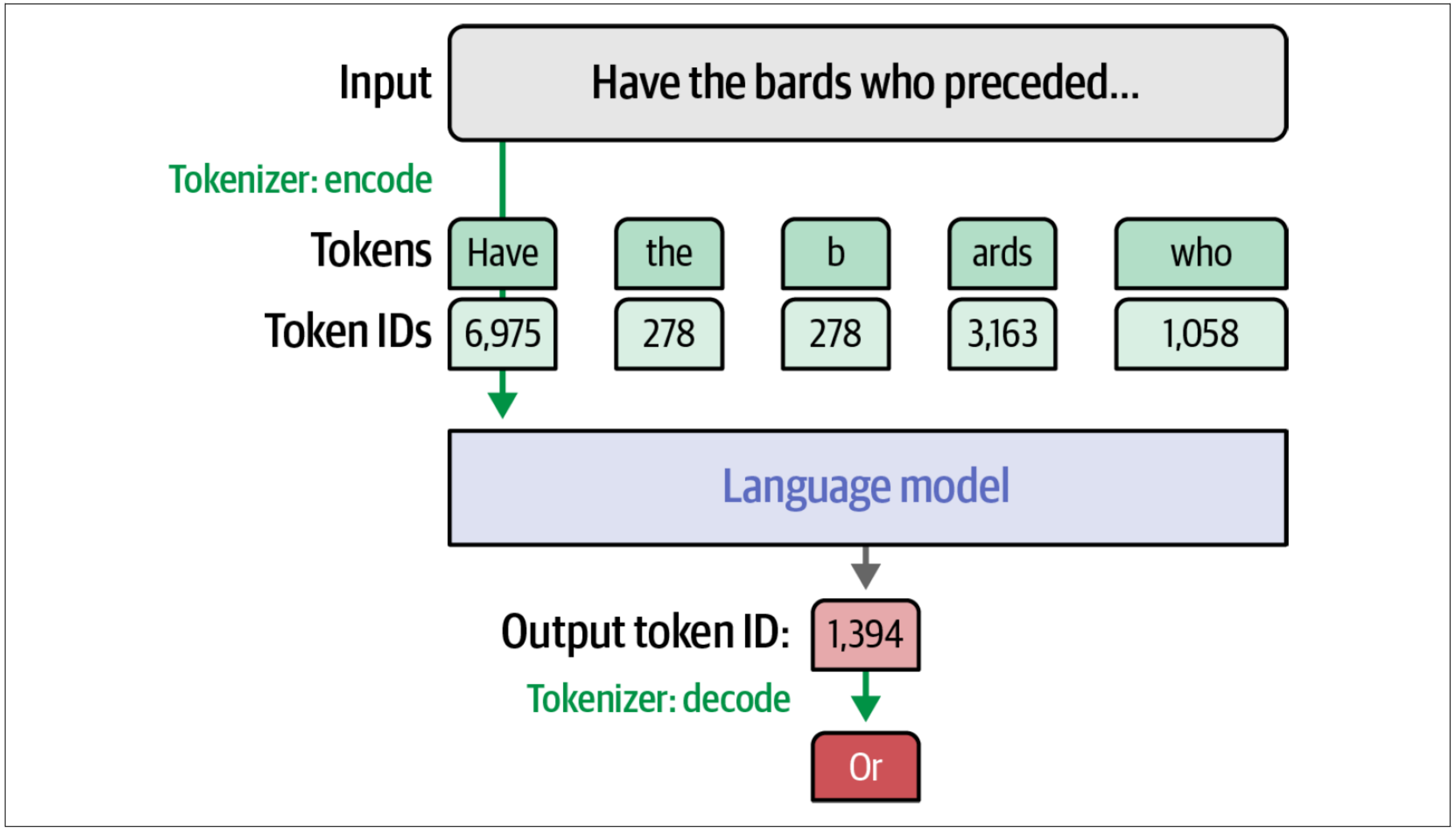
Tokens and Embeddings



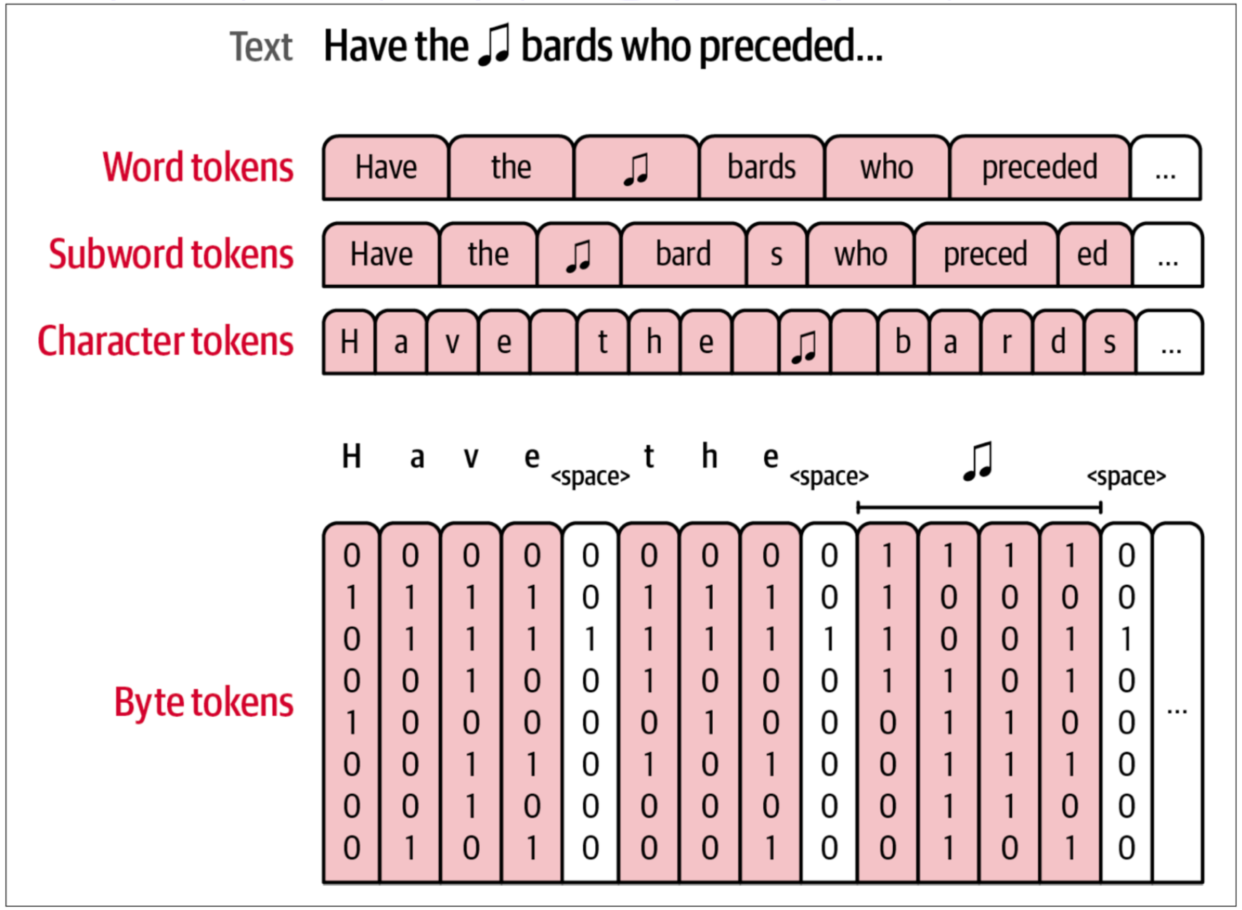
Tokens and Embeddings



Tokens and Embeddings



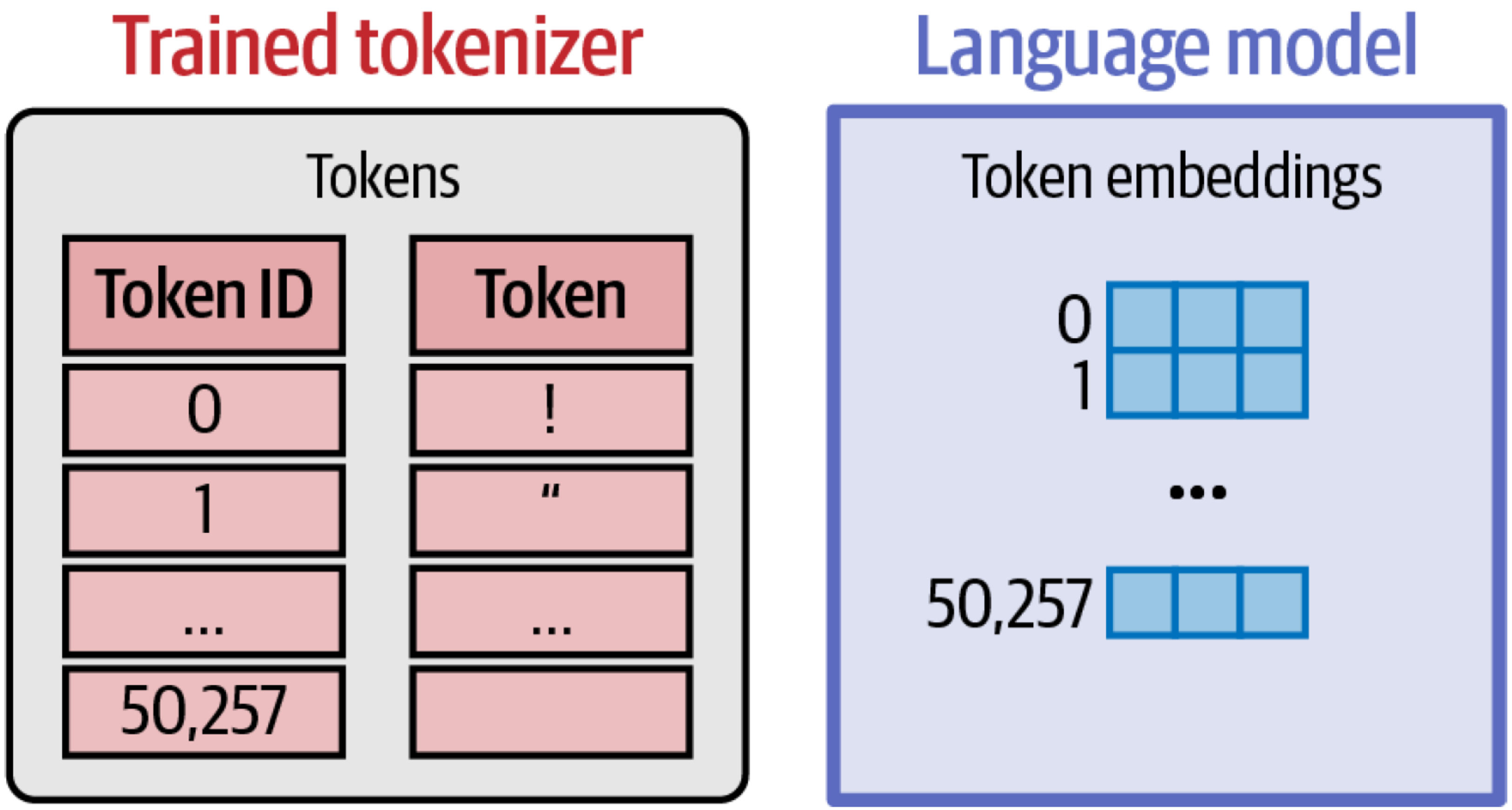
Tokens and Embeddings



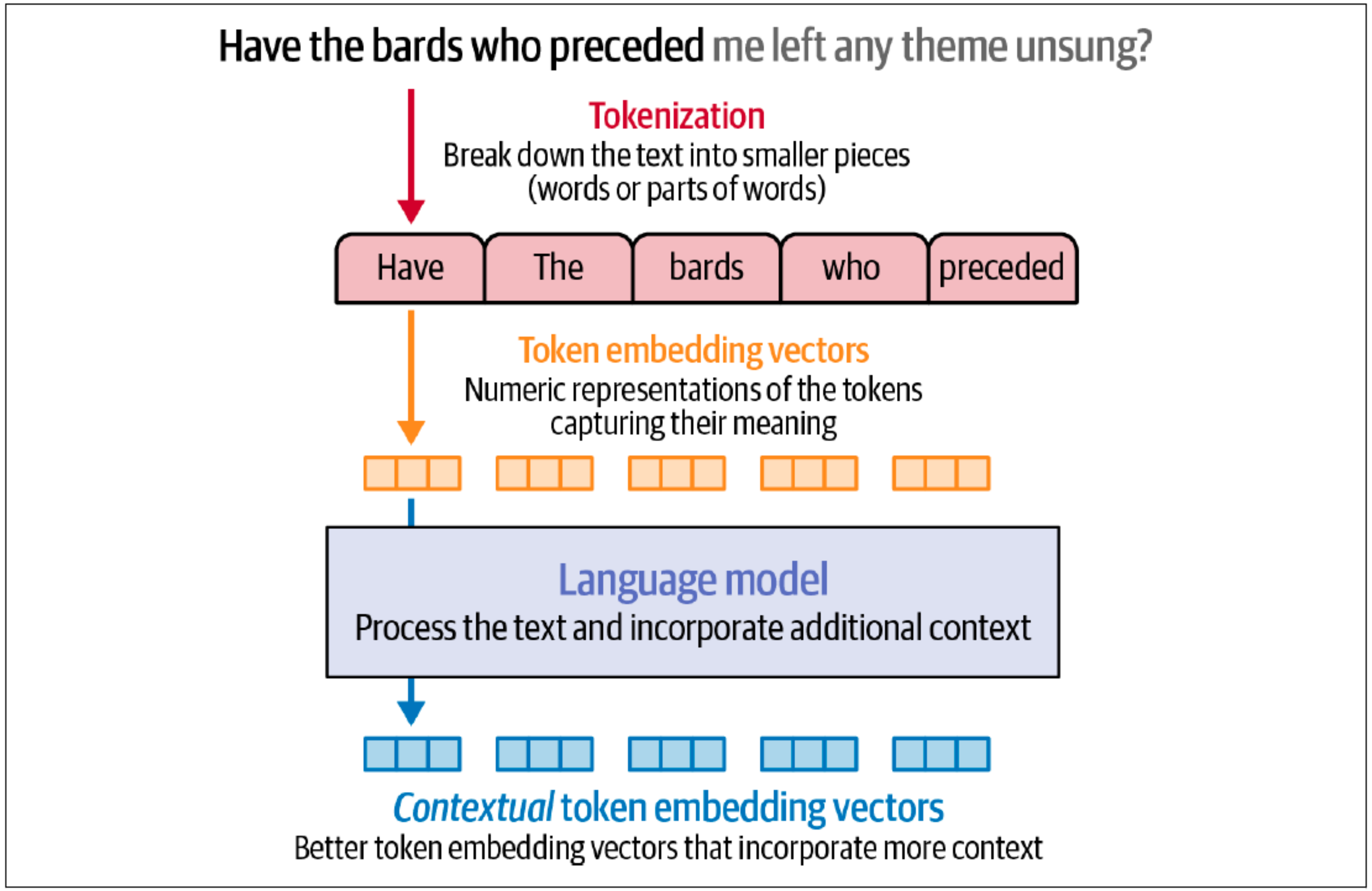
Tokens and Embeddings

BERT base model (uncased)	[CLS] english and capital ##ization [UNK] [UNK] show _ token ##s false none eli ##f = = > = else : two tab ##s : " " three tab ##s : " " 12 . 0 * 50 = 600 [SEP]
BERT base model (cased)	[CLS] English and CA ##PI ##TA ##L ##I ##Z ##AT ##ION [UNK] [UNK] show _ token ##s F ##als ##e None el ##if = = > = else : two ta ##bs : " " Three ta ##bs : " " 12 . 0 * 50 = 600 [SEP]
GPT-2	English and CAP ITAL IZ ATION � � � � � � show _ t ok ens False None el if == >= else : two tabs :" " Three tabs : " " 12 . 0 * 50 = 600
FLAN-T5	English and CA PI TAL IZ ATION <unk> <unk> show _ to ken s Fal s e None e l if = = > = else : two tab s : " " Three tab s : " " 12. 0 * 50 = 600 </s>
GPT-4	English and CAPITAL IZATION � � � � � � show _tokens False None elif == >= else : two tabs :" " Three tabs : " " 12 . 0 * 50 = 600
StarCoder	English and CAPITAL IZATION � � � � � show _ tokens False None elif == >= else : two tabs :" " Three tabs : " " 1 2 . 0 * 5 0 = 6 0 0

Tokens and Embeddings

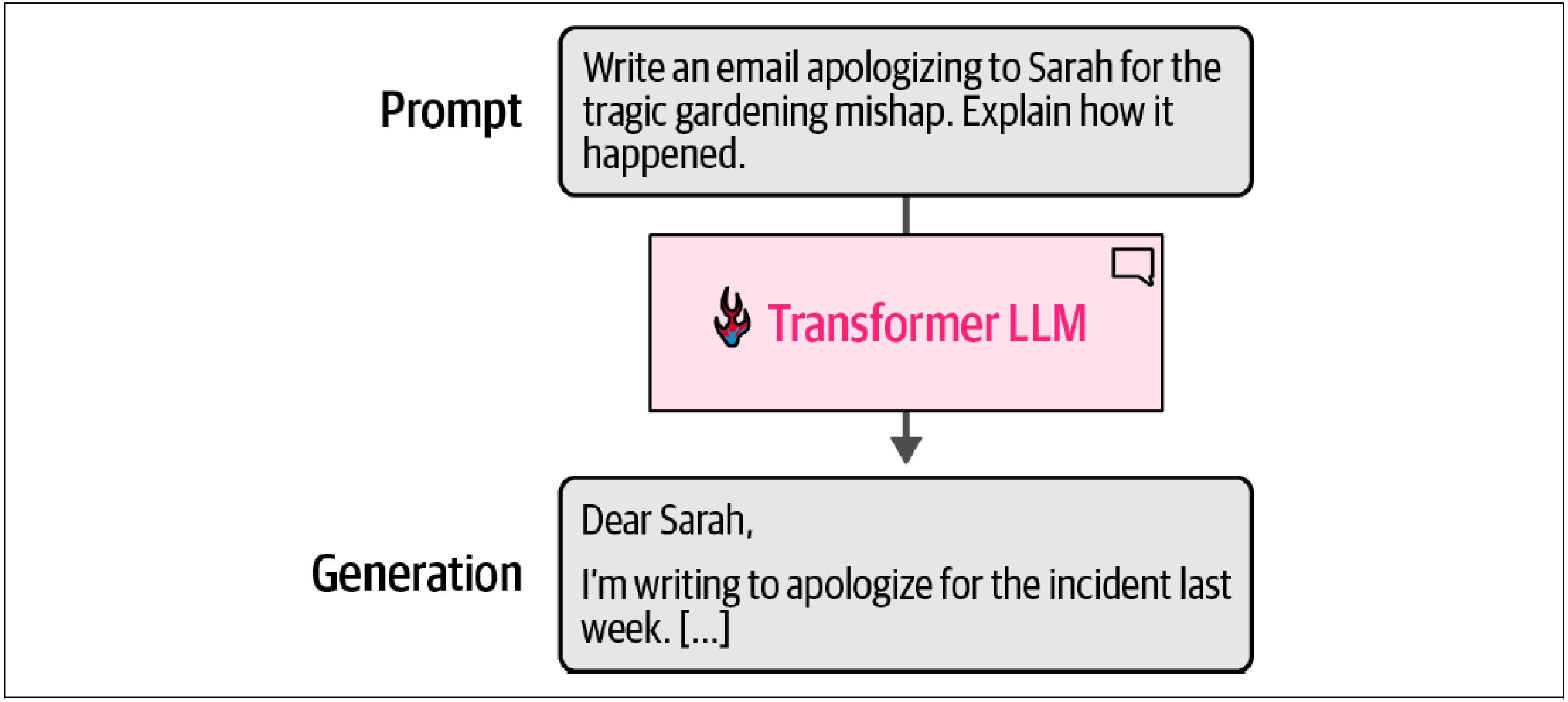


Tokens and Embeddings

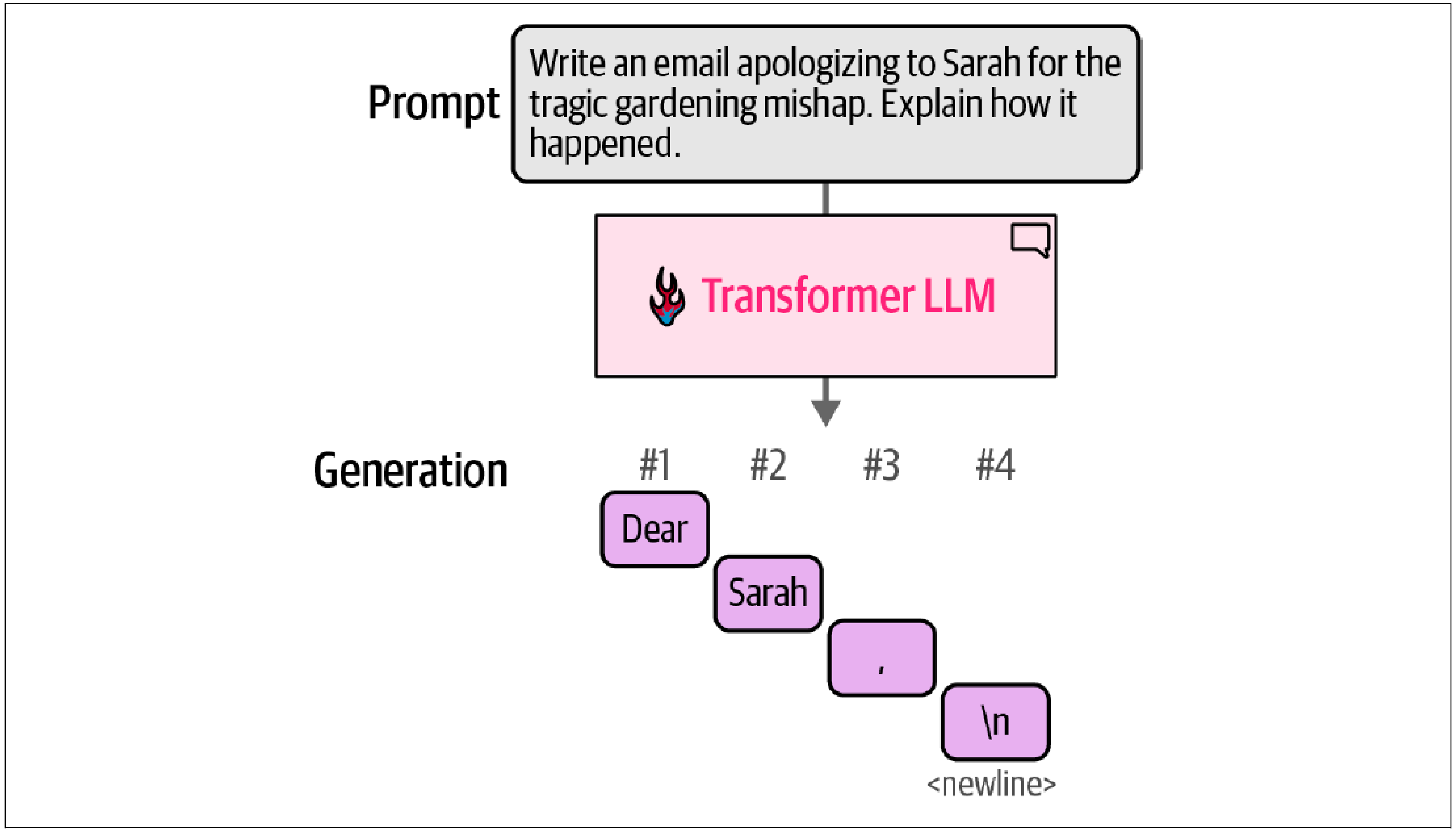


Large Language Models: An Overview of Transformer Models

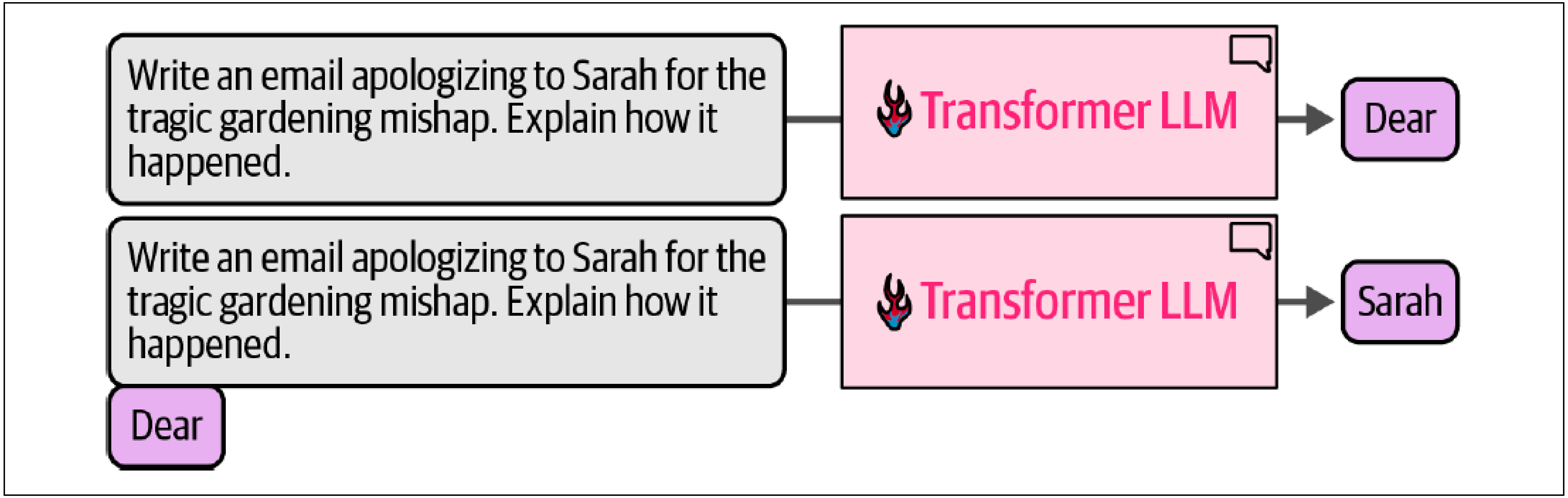
An Overview of Transformer Models



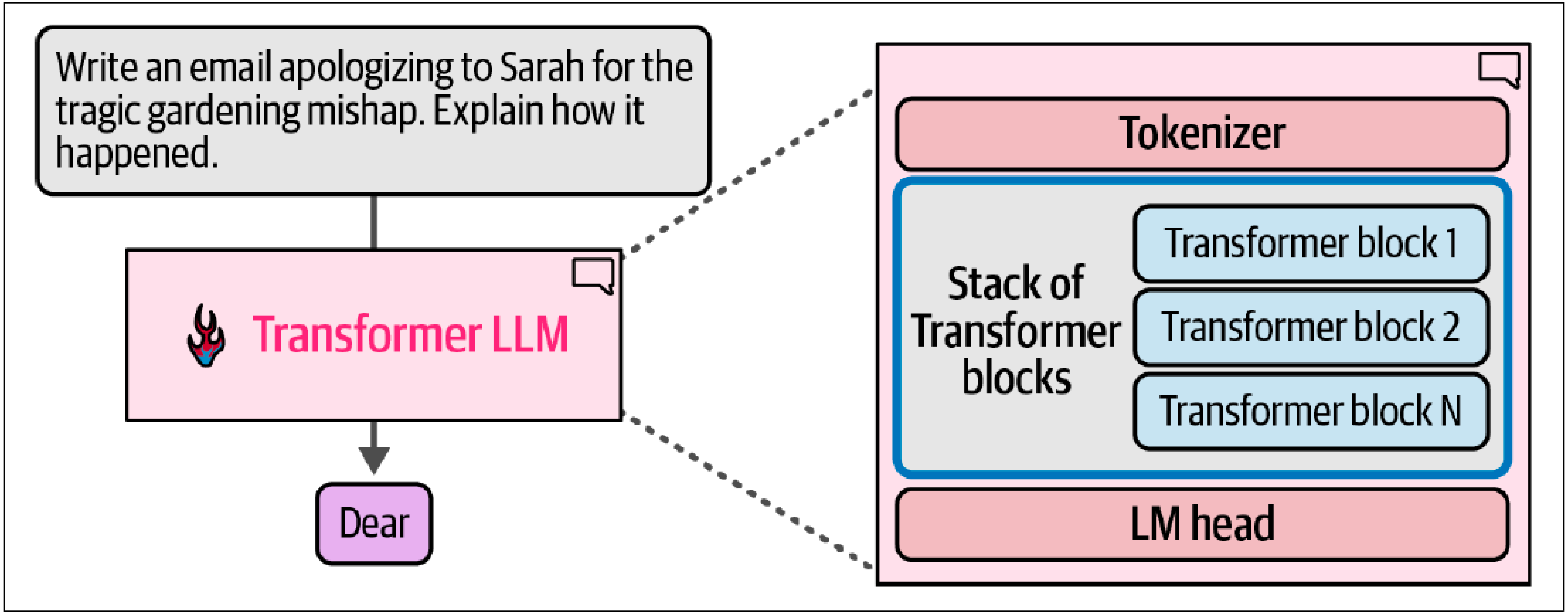
An Overview of Transformer Models



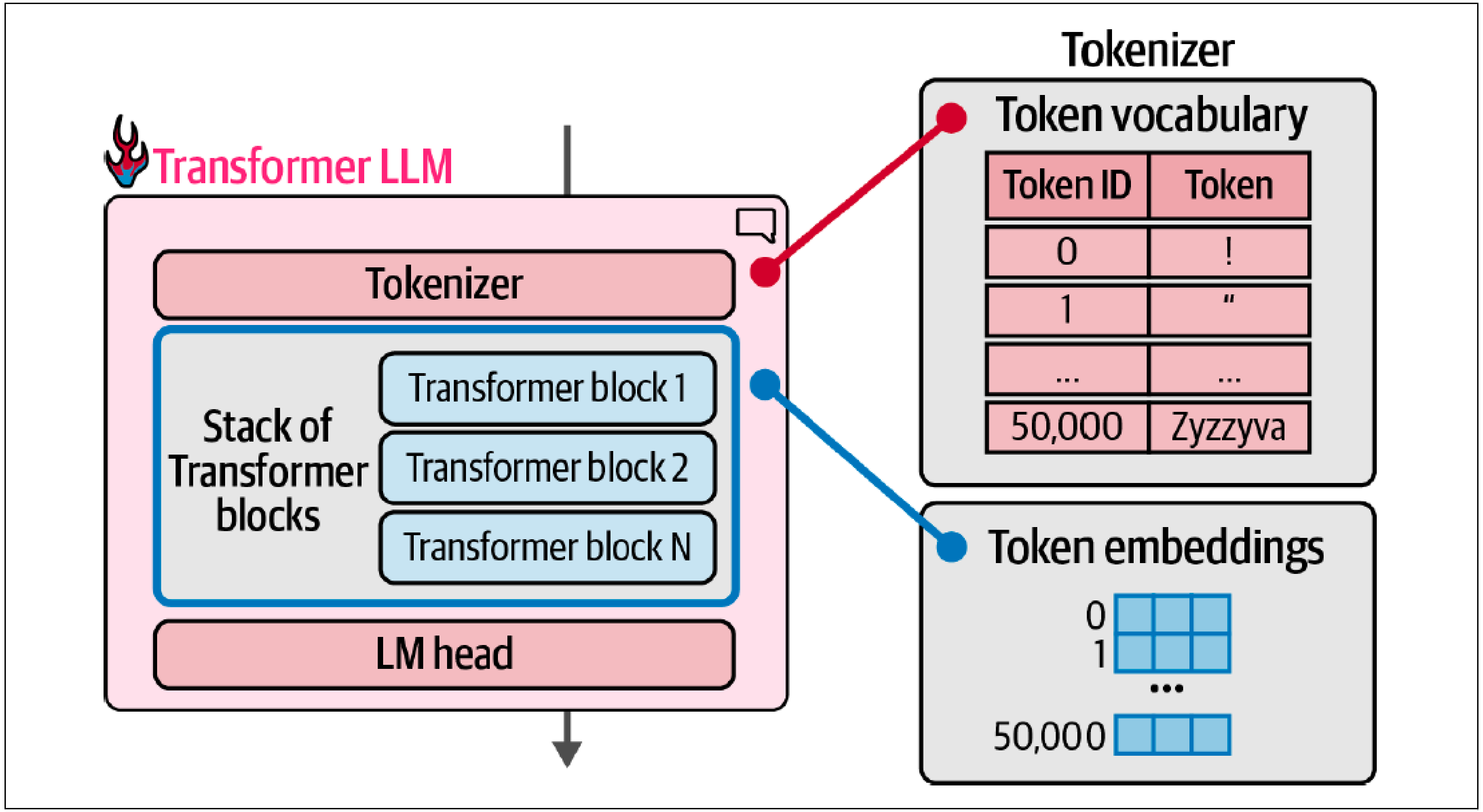
An Overview of Transformer Models



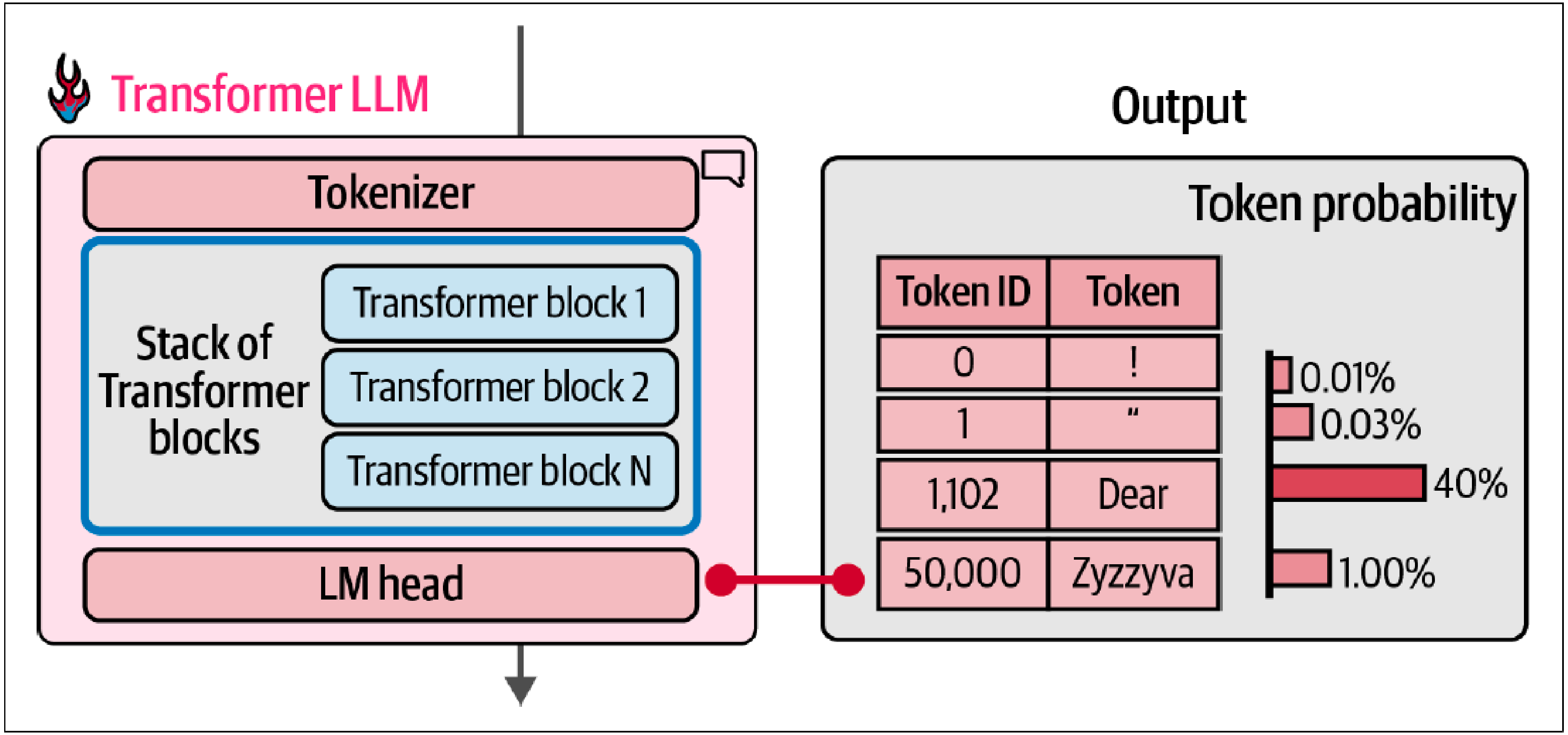
The Components of the Forward Pass



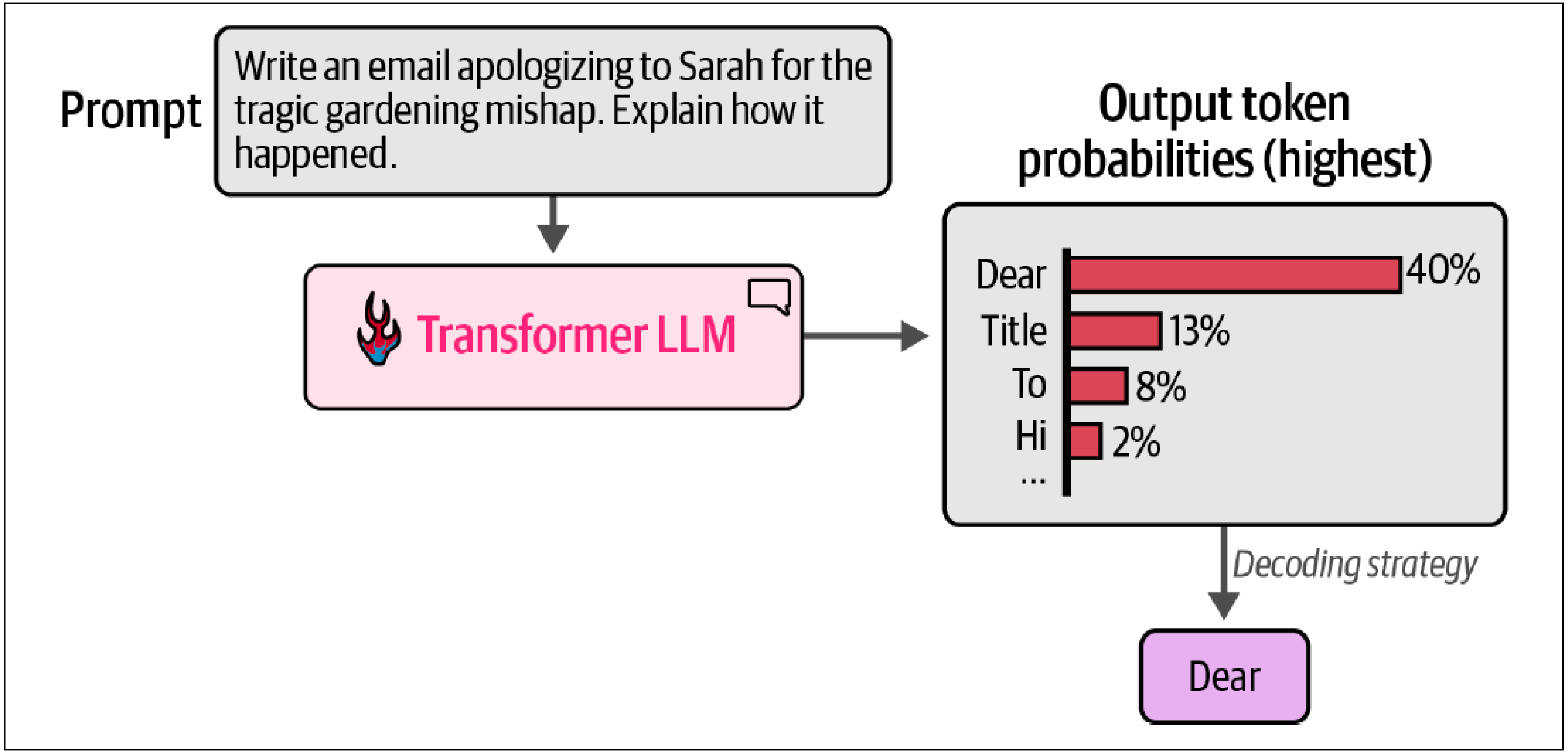
The Components of the Forward Pass



The Components of the Forward Pass



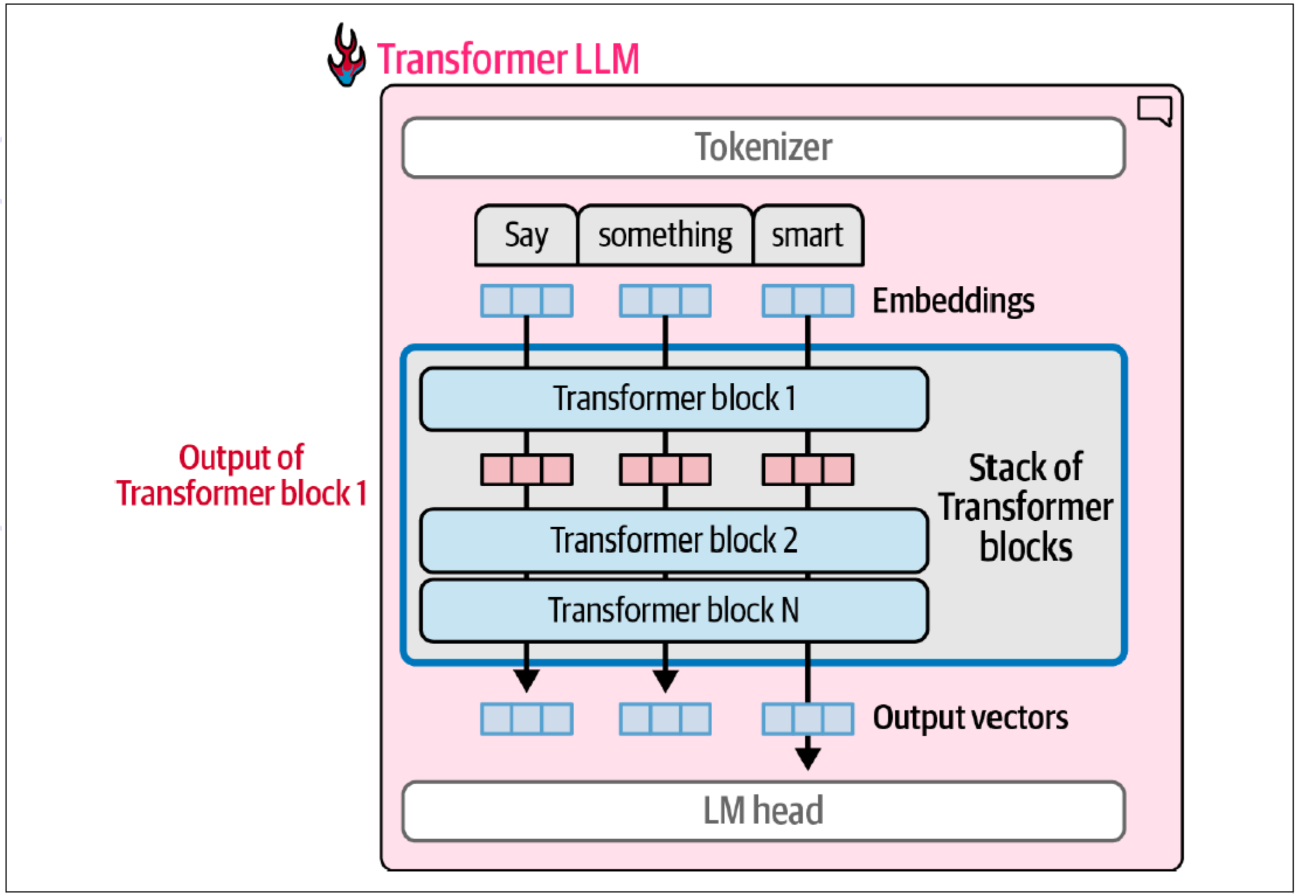
The Components of the Forward Pass



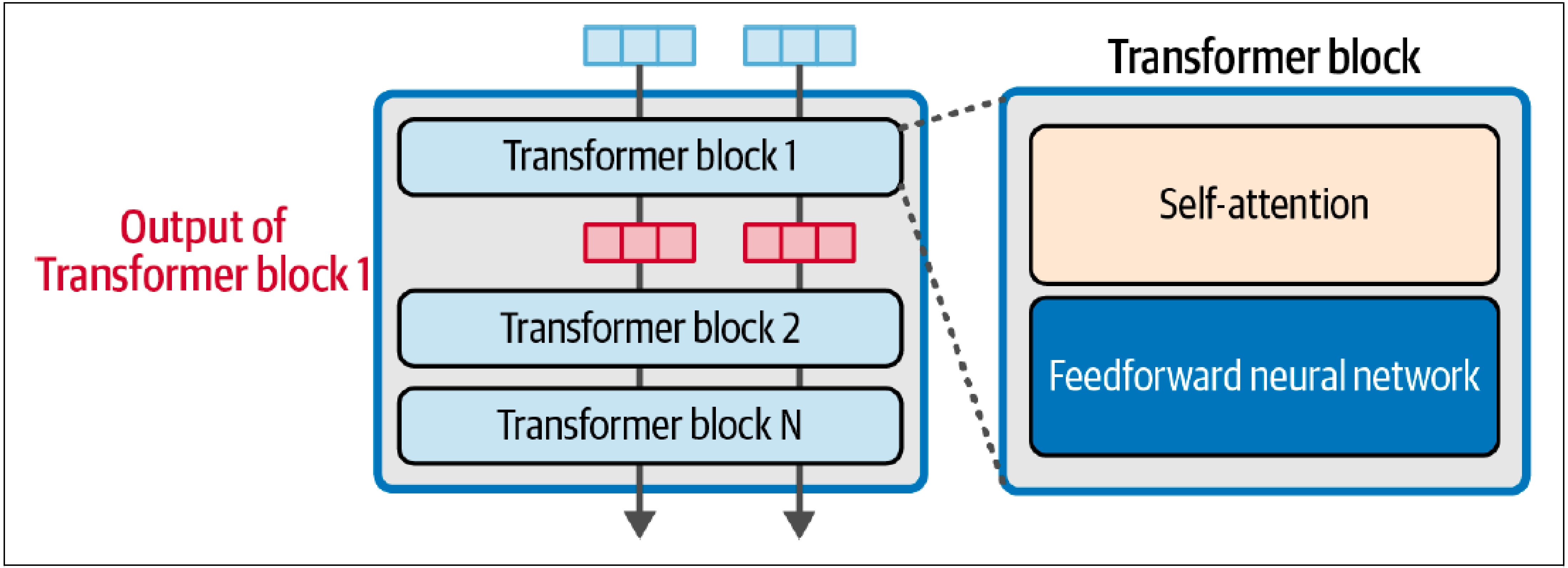
Large Language Models: An Overview of Transformer Models

Inside the Transformer Block

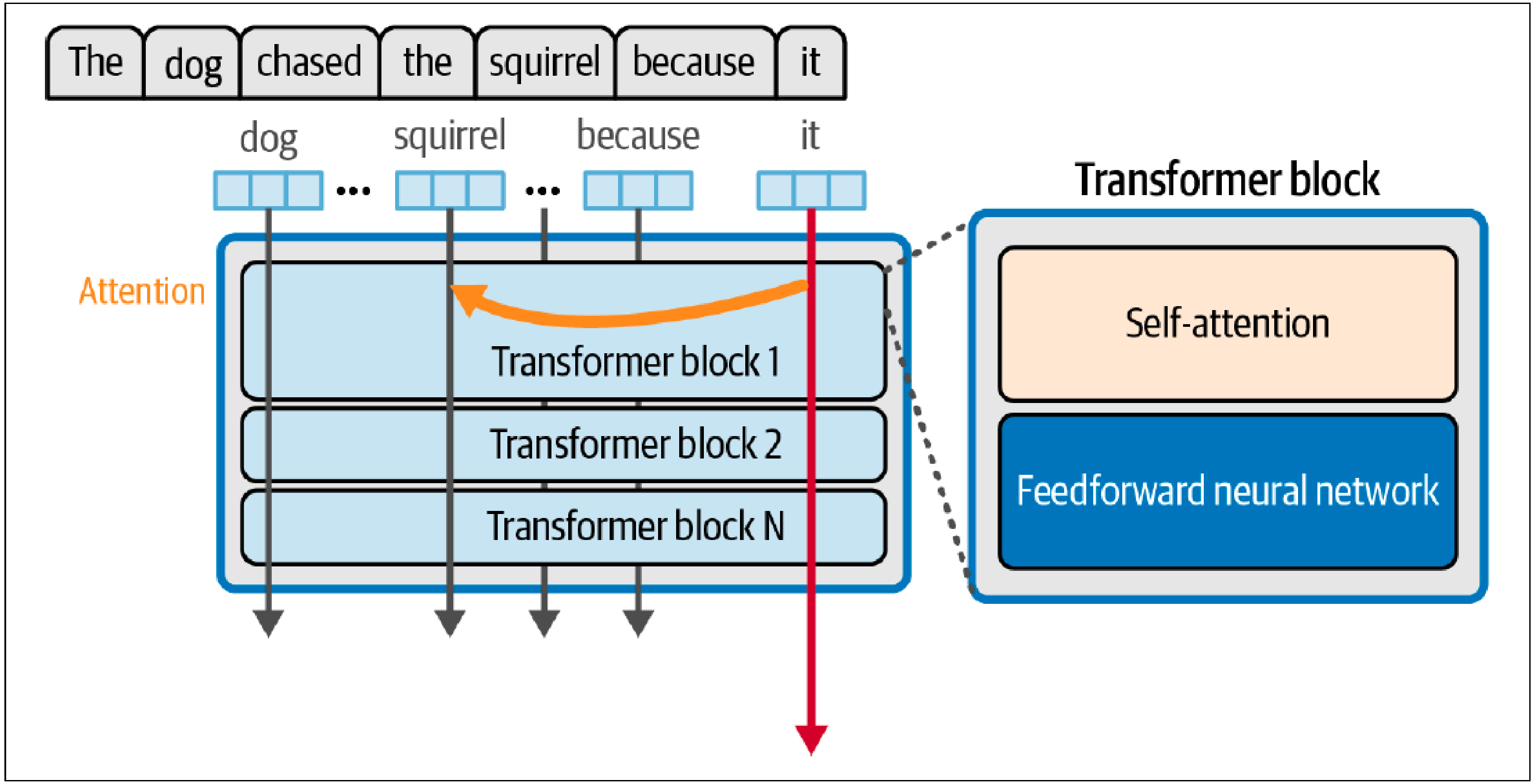
Inside the Transformer Block



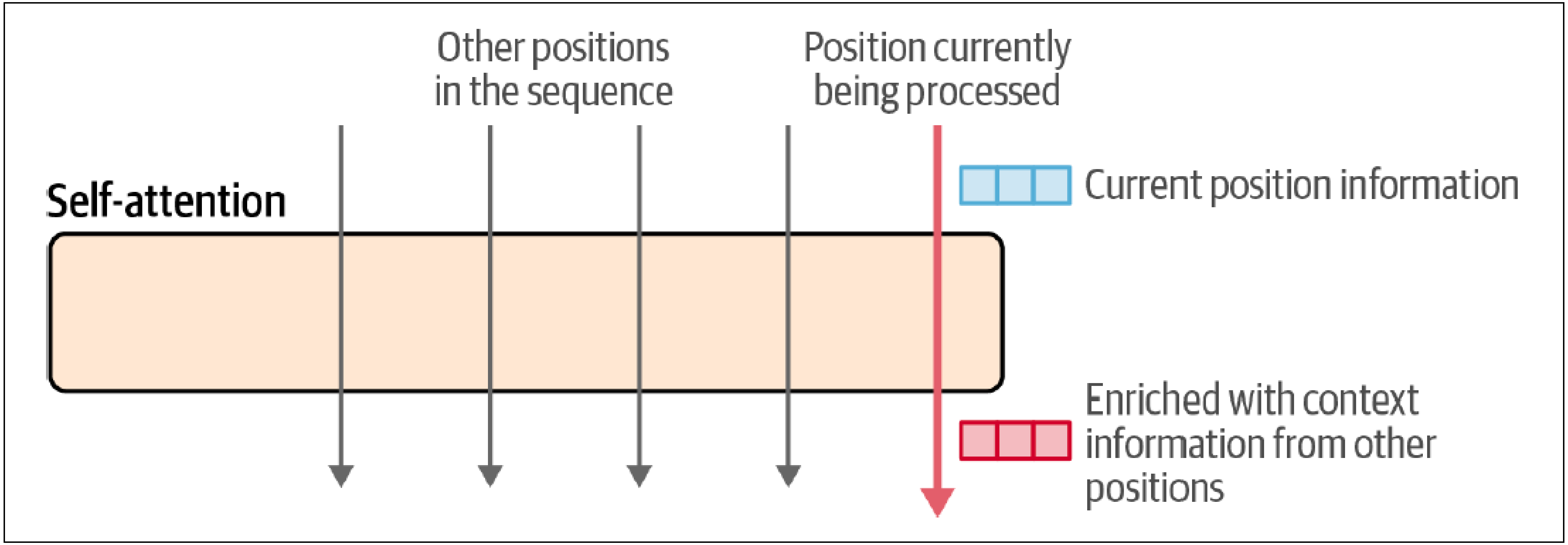
Inside the Transformer Block



Attention is all you need



Attention is all you need



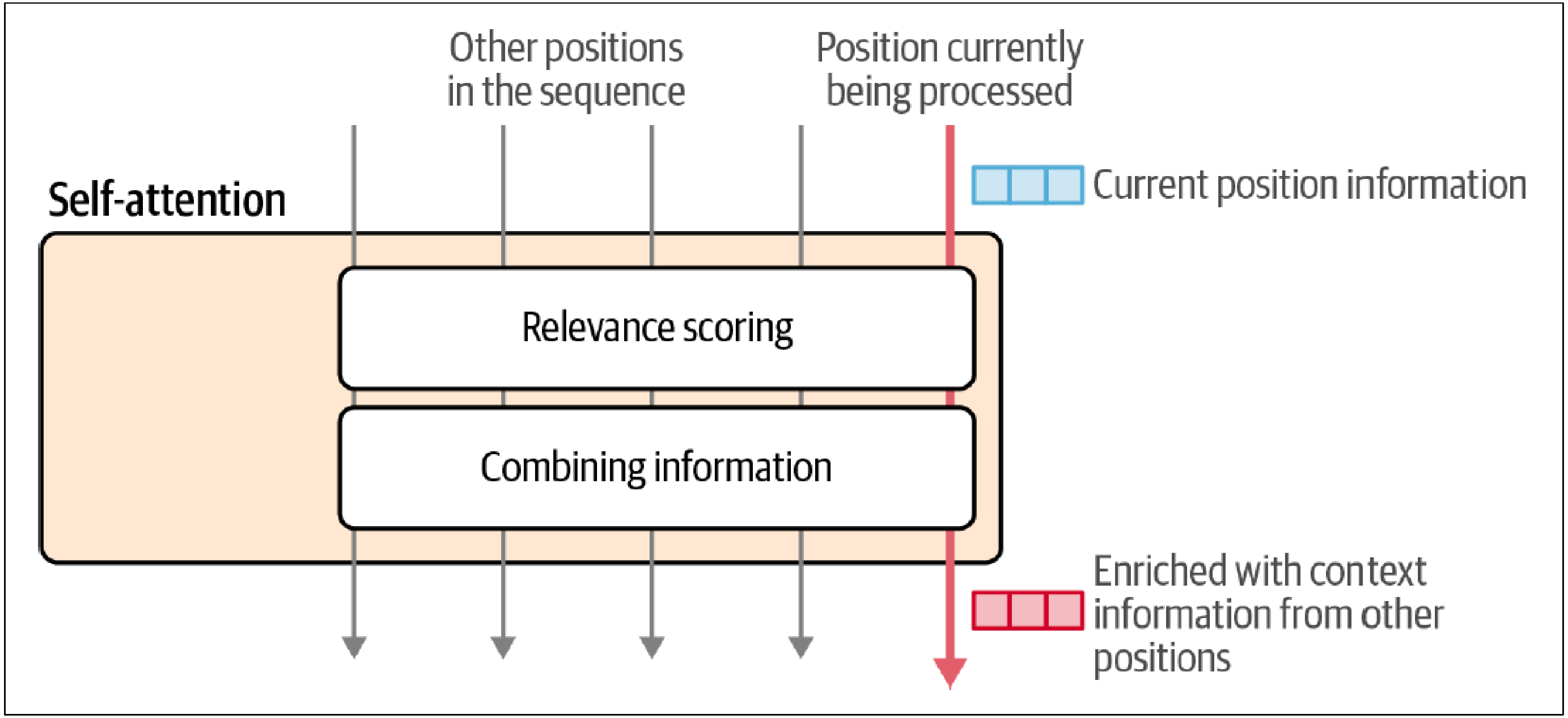
Attention is all you need

- Two main steps are involved in the attention mechanism:

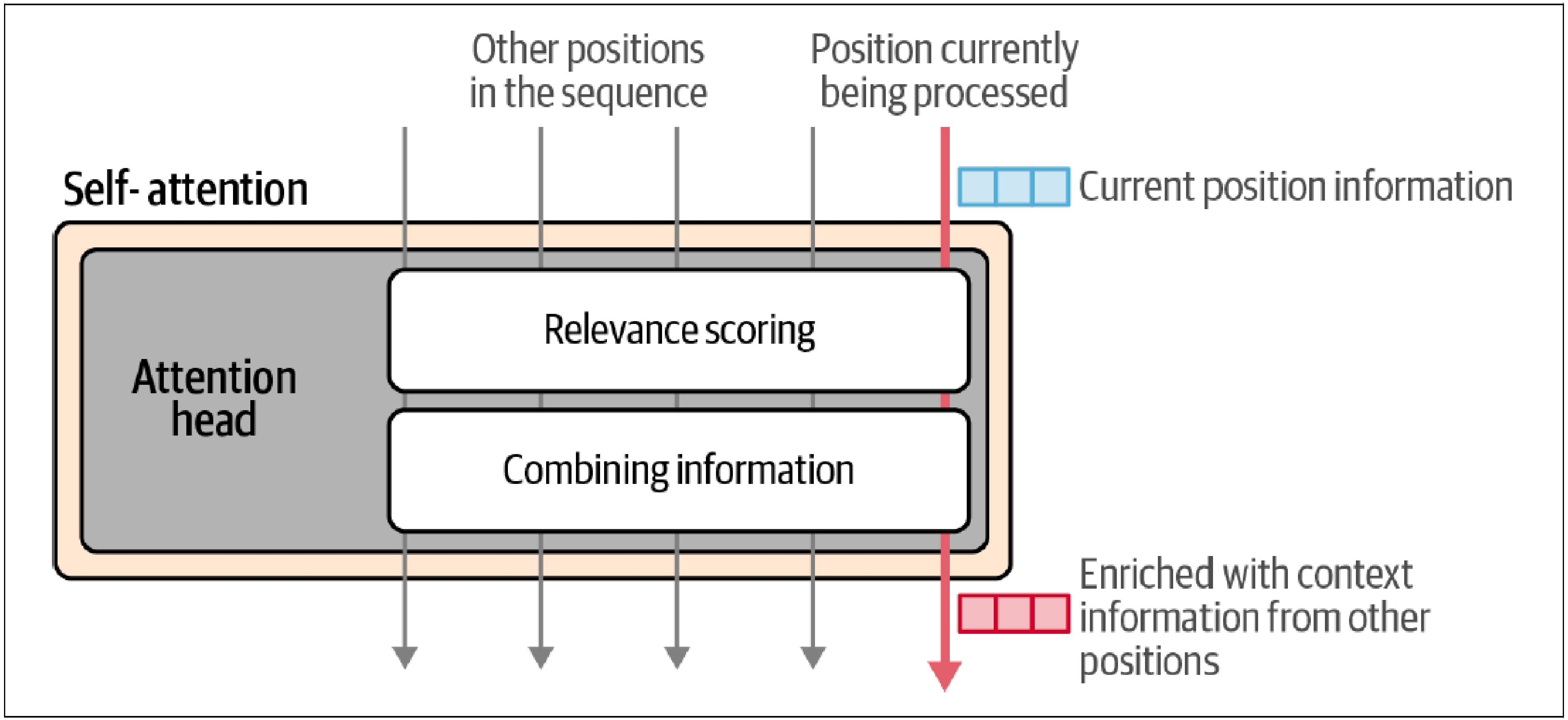
  - A way to score how relevant each of the previous input tokens are to the current token being processed.

  - Using those scores, we combine the information from the various positions into a single output vector.

Attention is all you need



Attention is all you need



How attention is calculated

- The attention layer (of a generative LLM) is processing attention for a single position.

- The inputs to the layer are:
  - The vector representation of the current position or token
  - The vector representations of the previous tokens

How attention is calculated

- The goal is to produce a new representation of the current position that incorporates relevant information from the previous tokens:

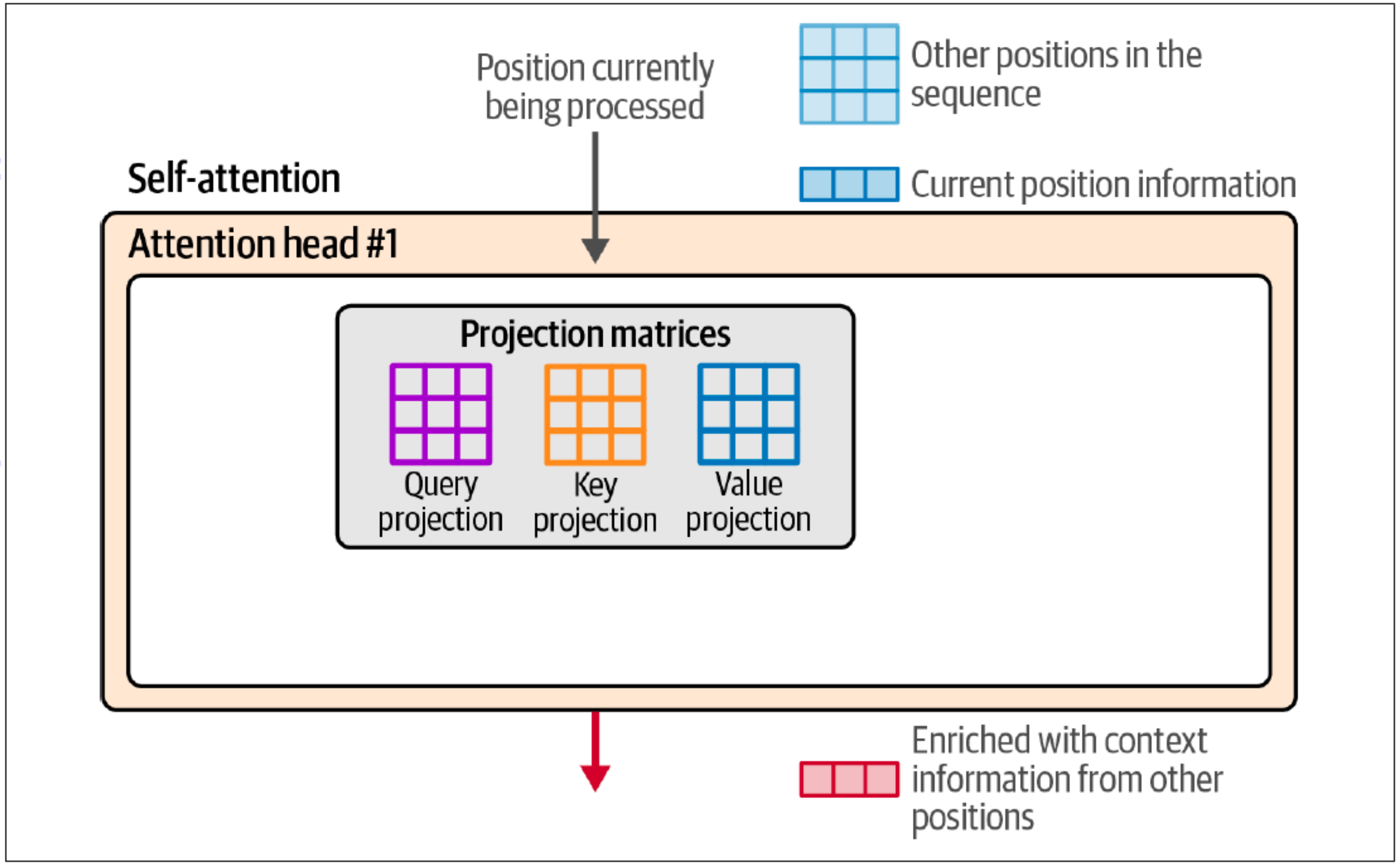
  - For example, if we're processing the last position in the sentence 'Sarah fed the cat because it', we want 'it' to represent the cat, so attention bakes in 'cat information' from the cat token.

How attention is calculated

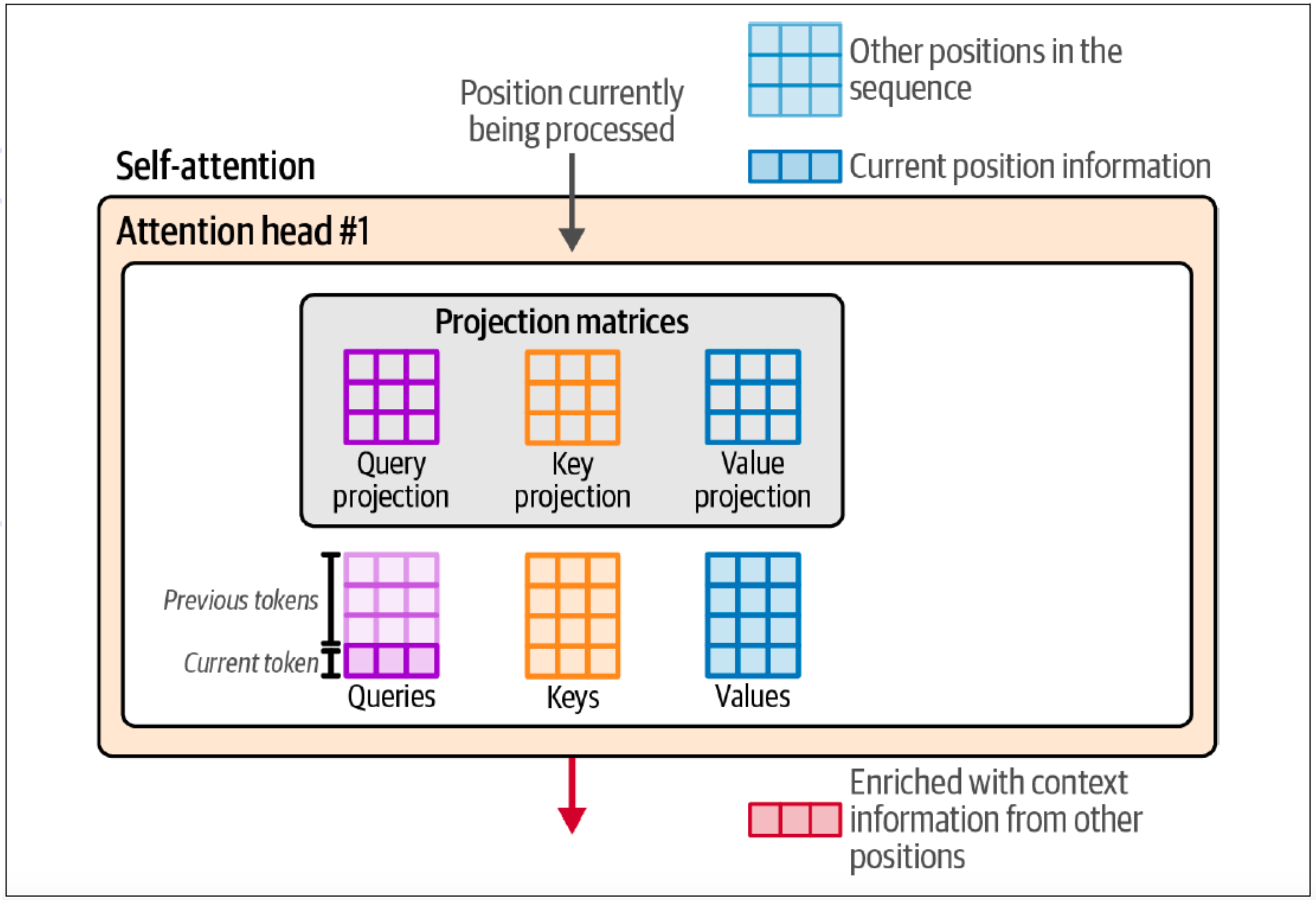
- The training process produces three projection matrices that produce the components that interact in this calculation:

  - A query projection matrix
  - A key projection matrix
  - A value projection matrix

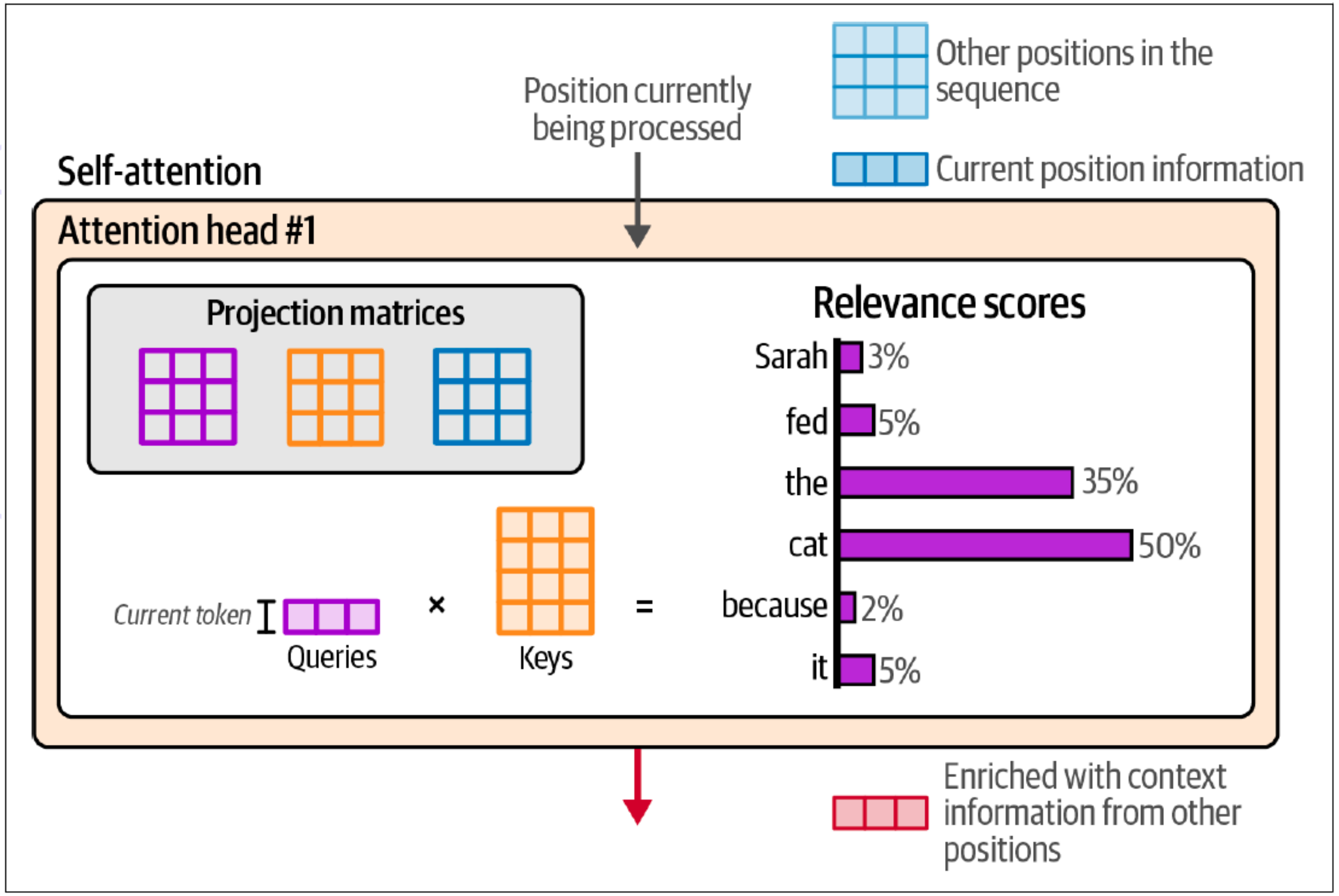
How attention is calculated



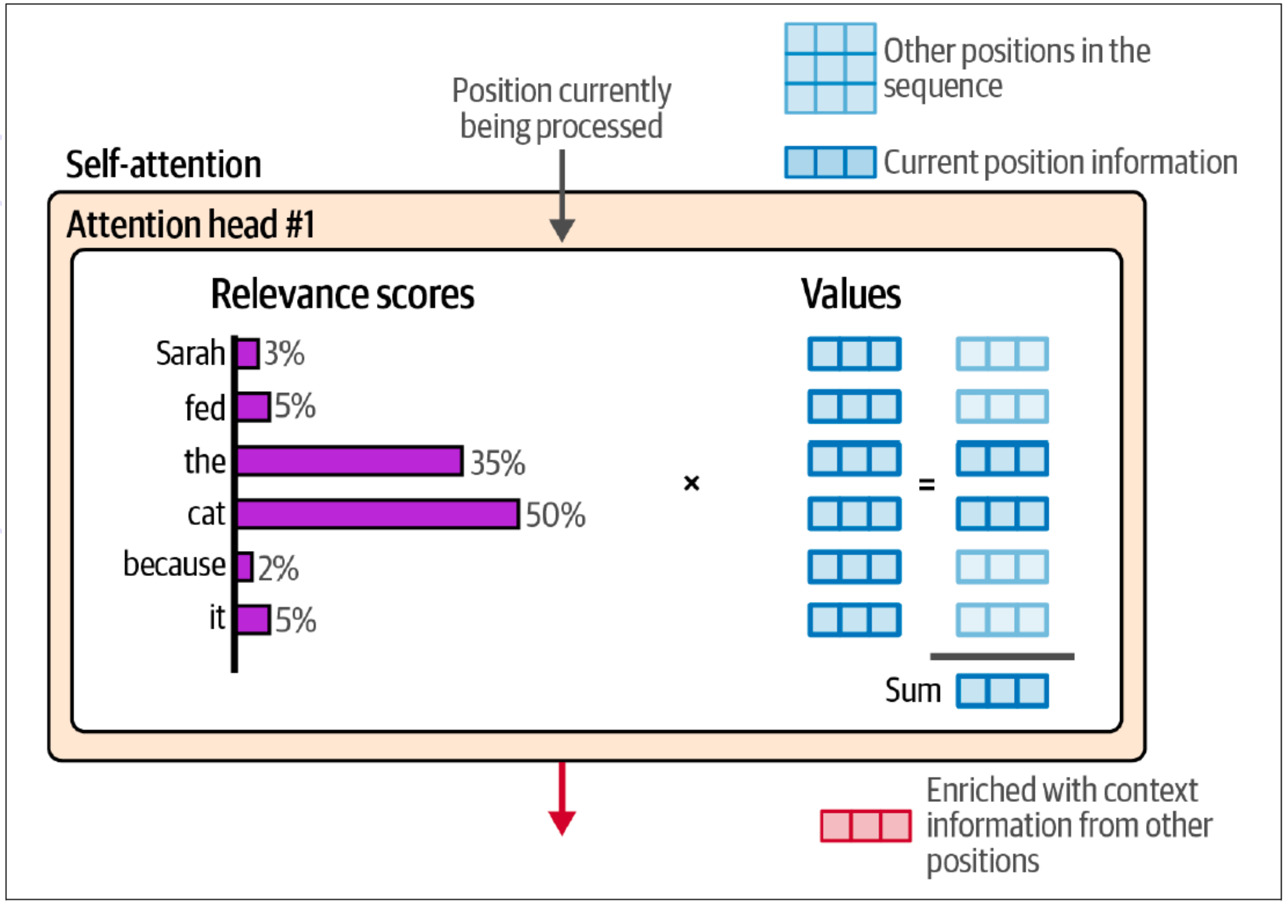
How attention is calculated



Self-attention: Revelance scoring



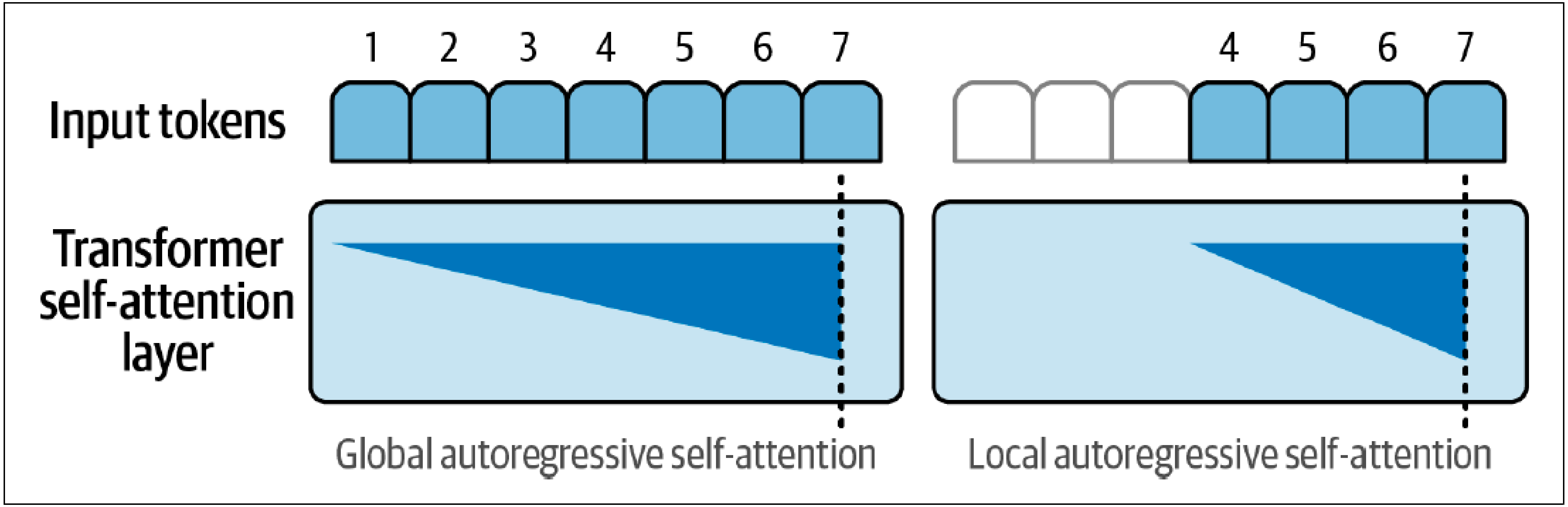
Self-attention: Revelance scoring



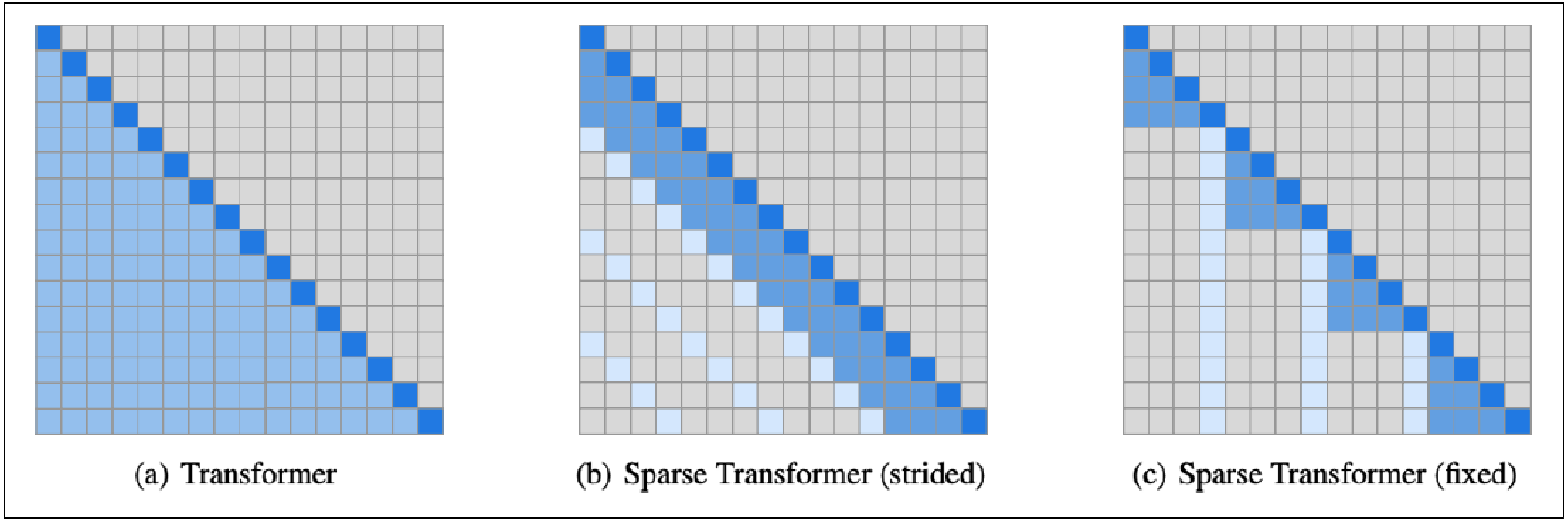
Large Language Models: Recent Improvements to the Transformer Architecture

Inside the Transformer Block

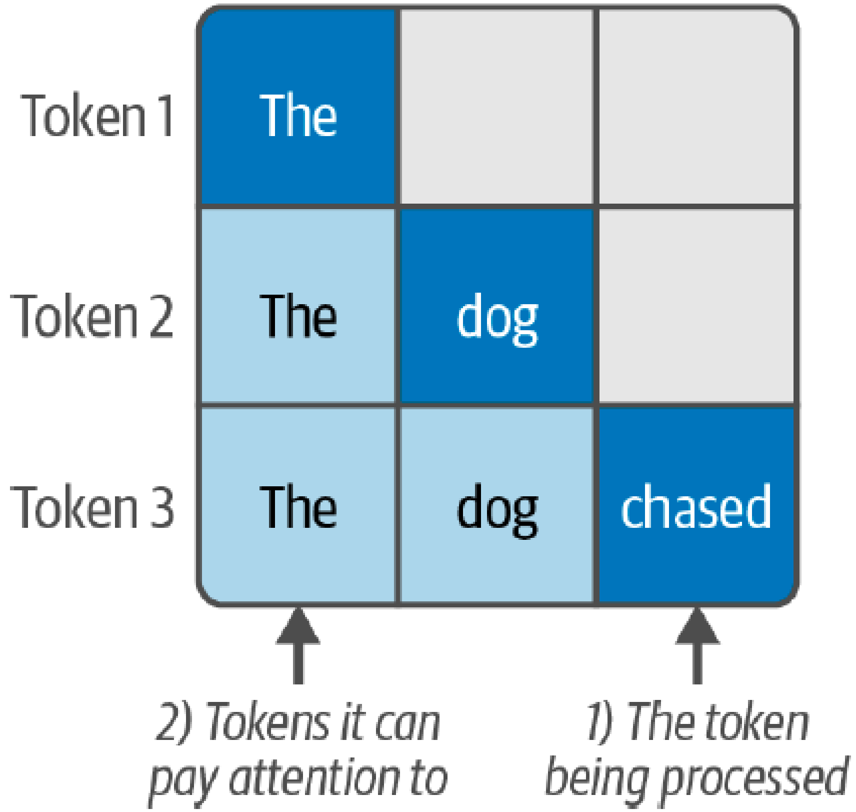
Local/Sparse attention



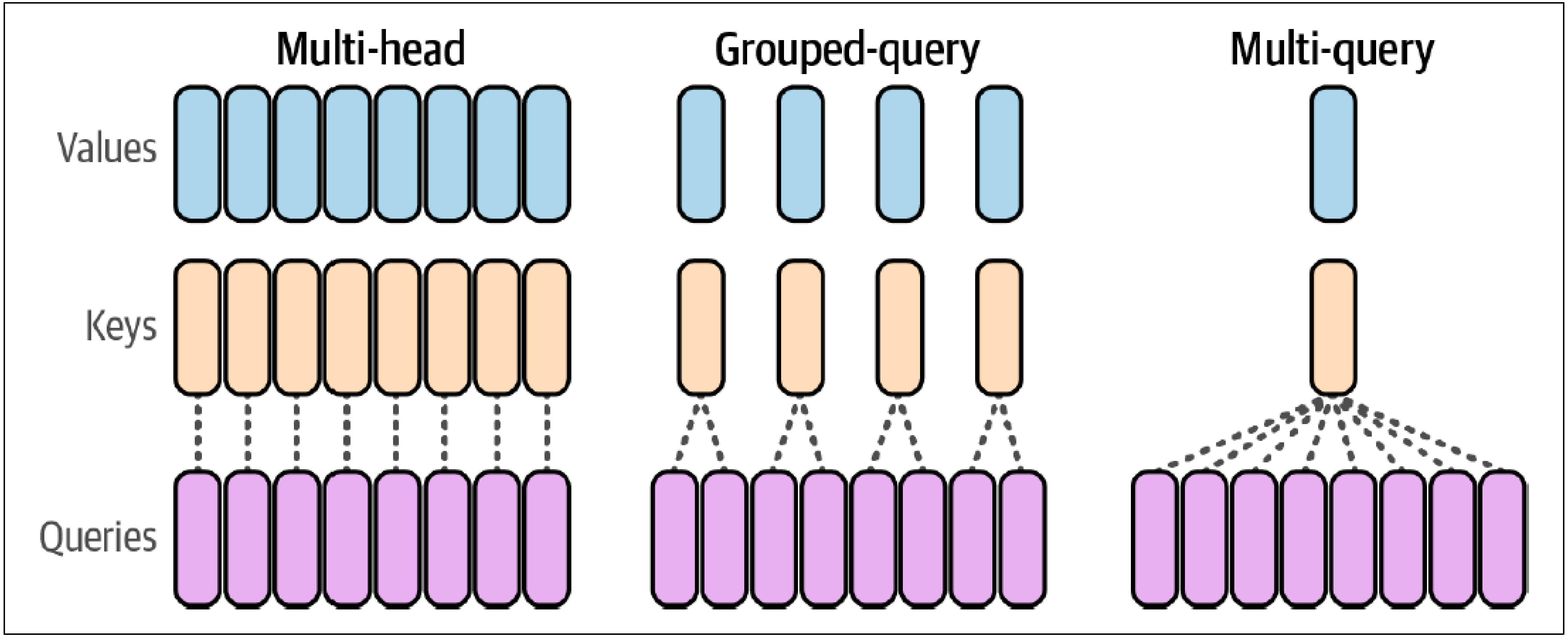
Local/Sparse attention



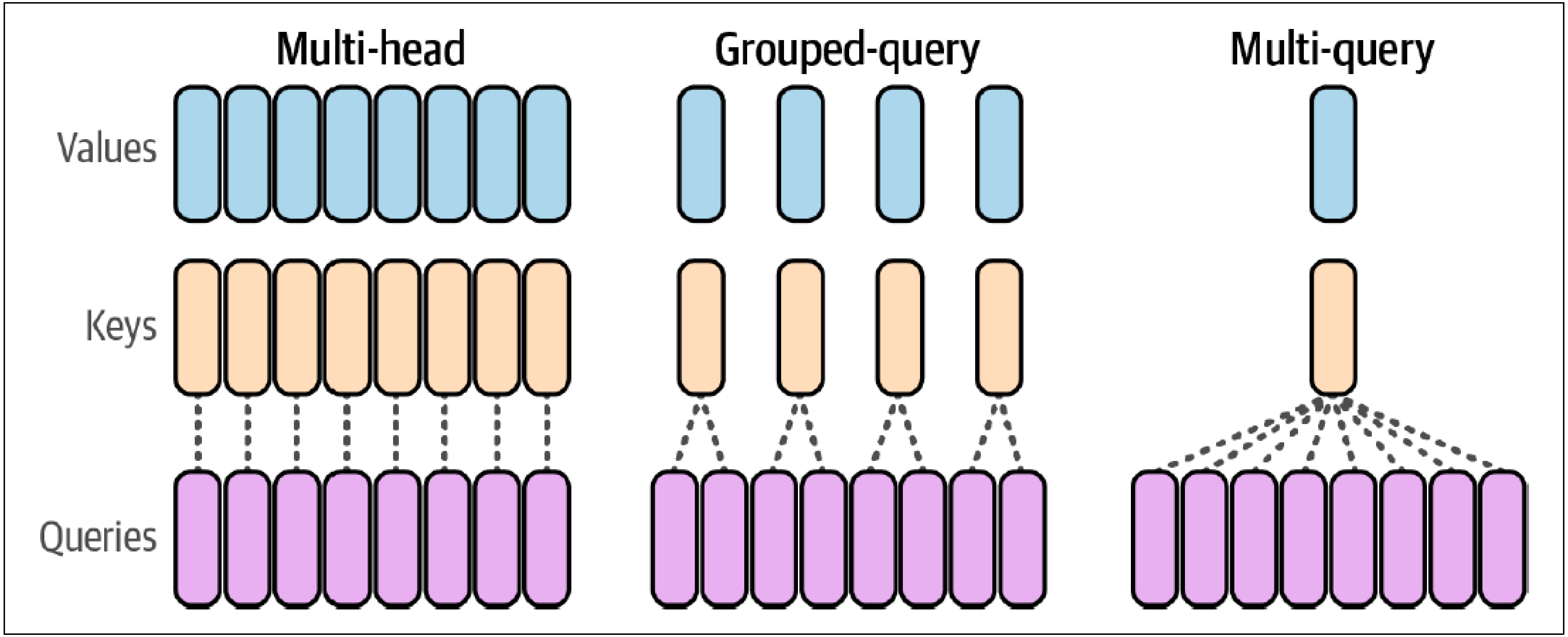
Local/Sparse attention



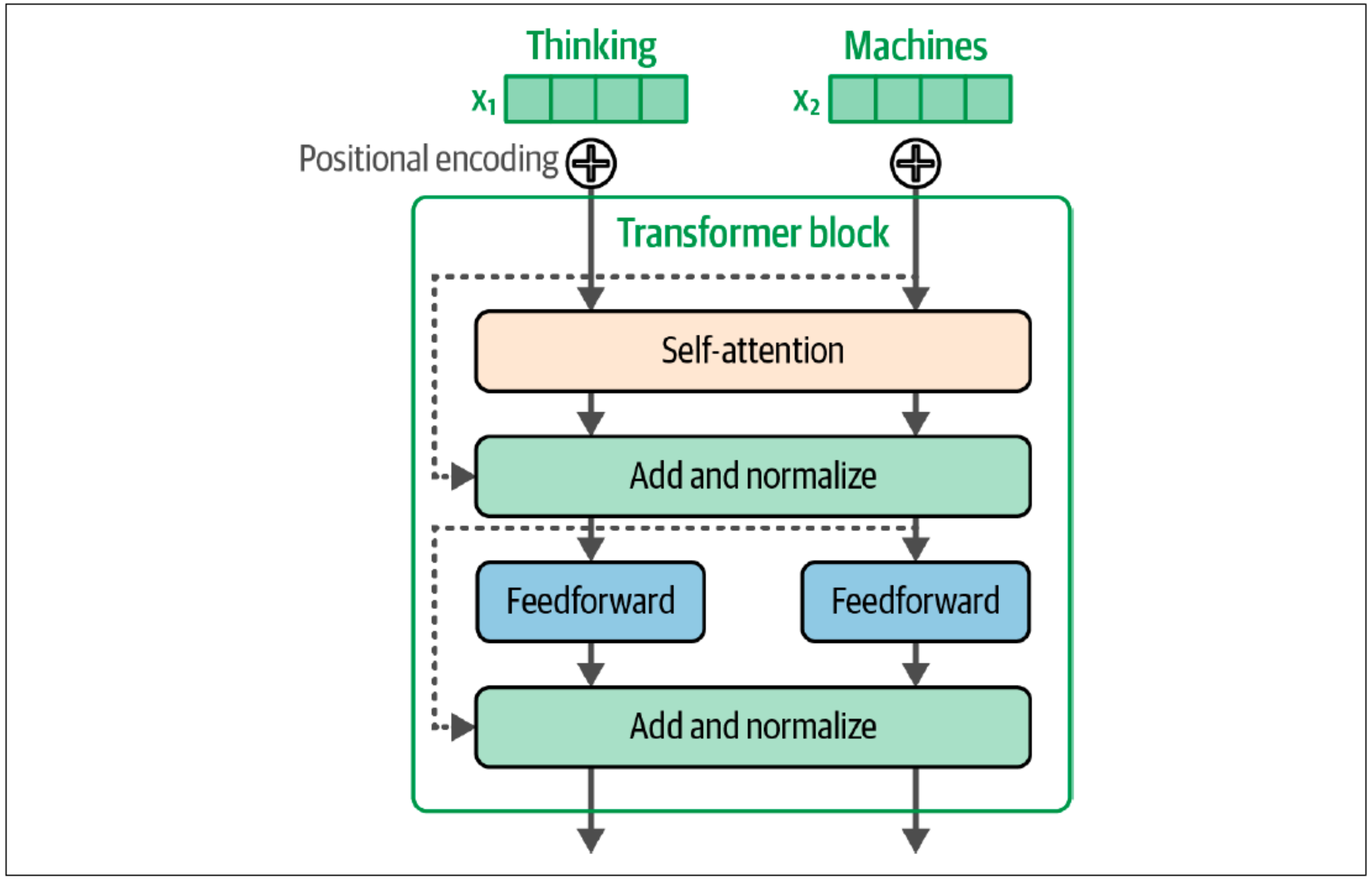
Multi-query and grouped-query attention



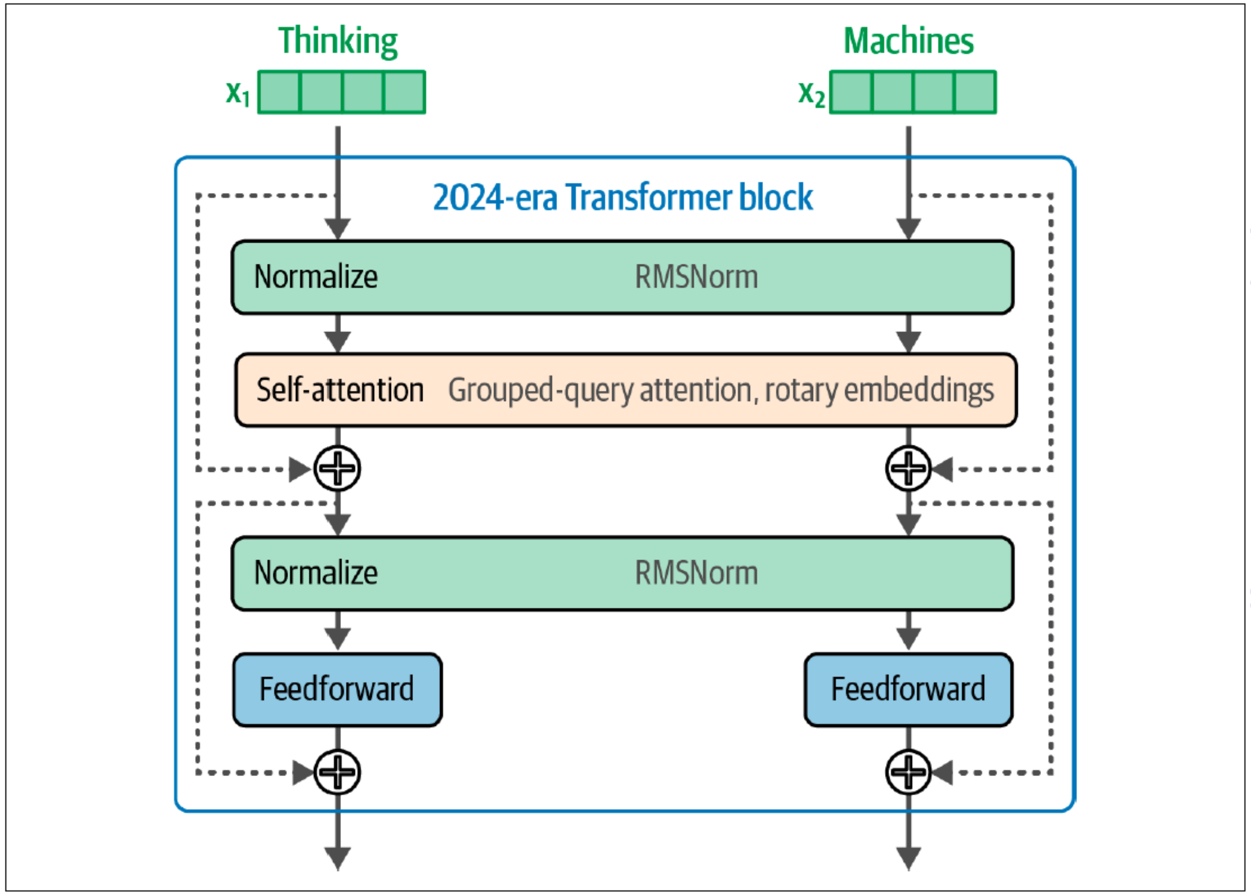
Multi-query and grouped-query attention



The Transformer Block



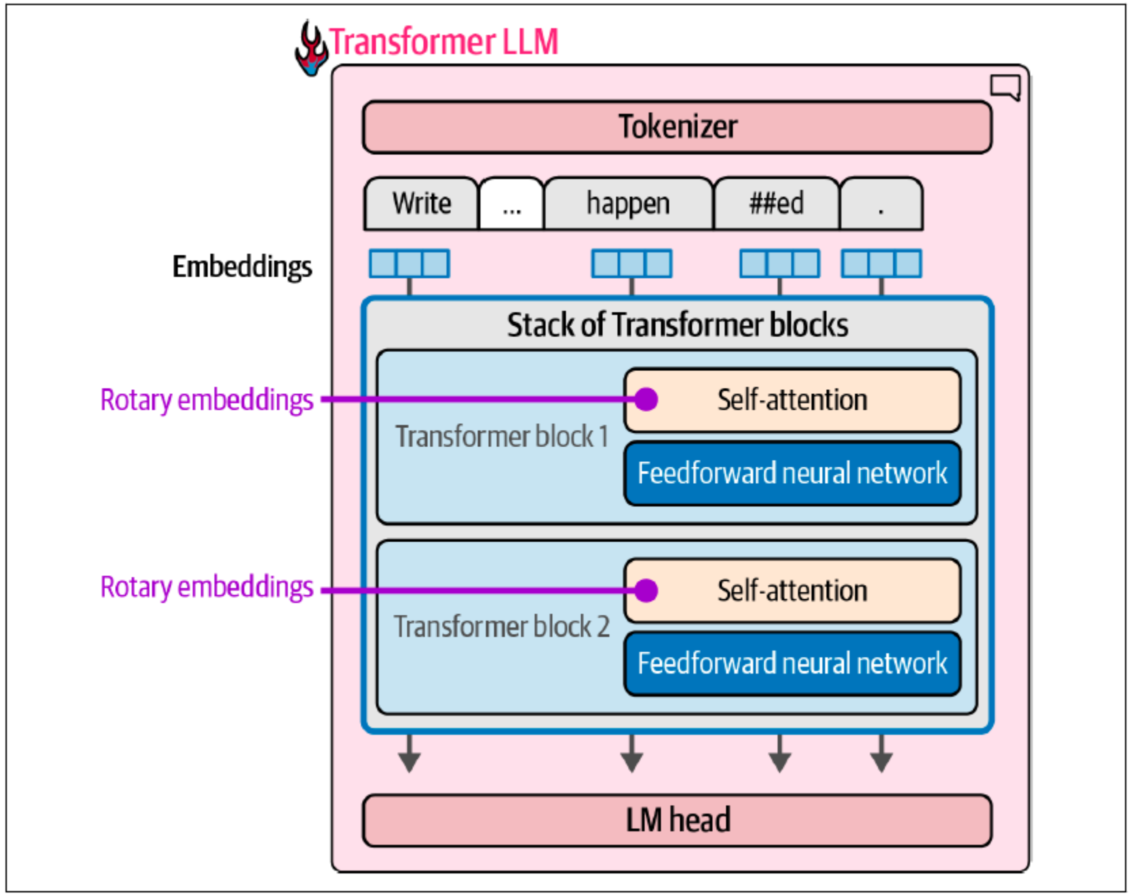
The Transformer Block



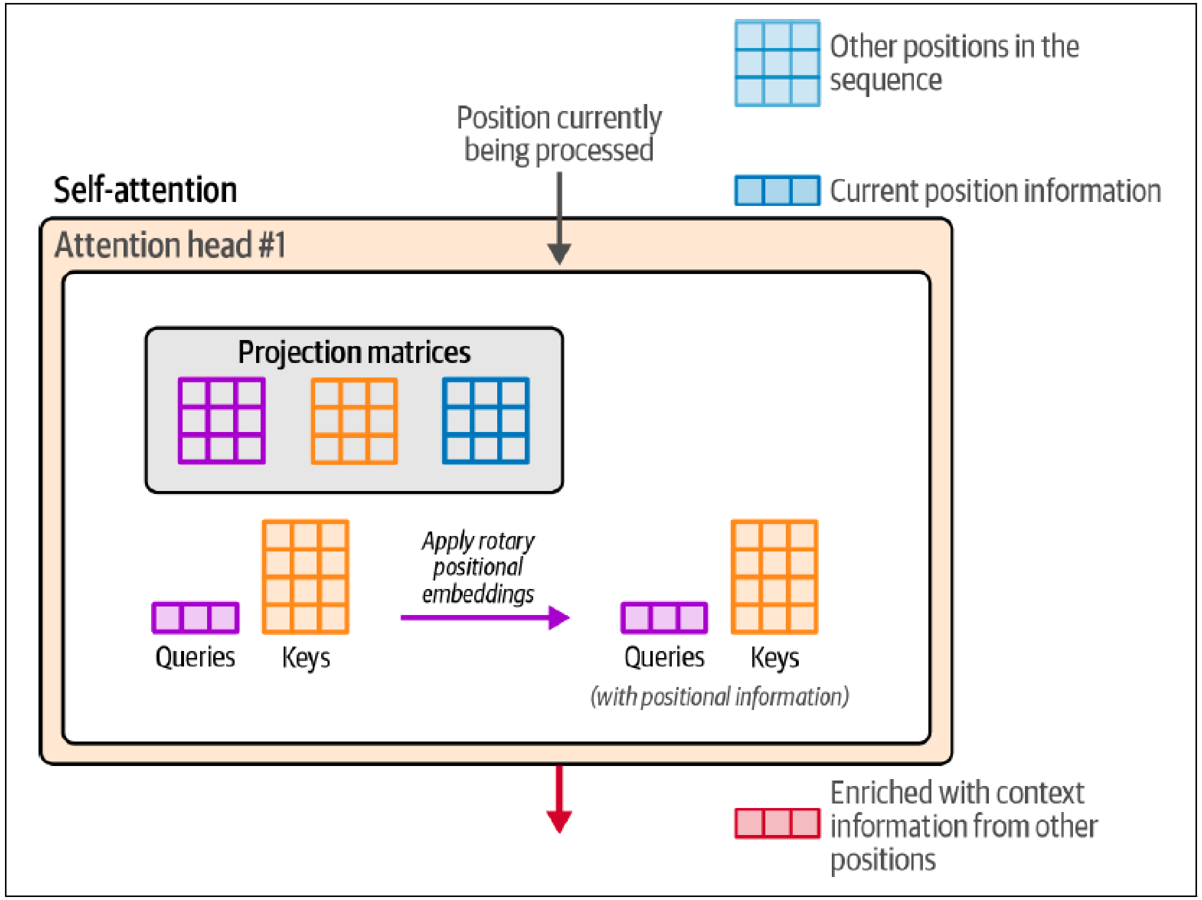
Positional Embeddings (RoPE)

- **Positional embeddings** enable the model to keep track of the order of tokens/words in a sequence/sentence, wich is an indispensable source information in language.

Positional Embeddings (RoPE)

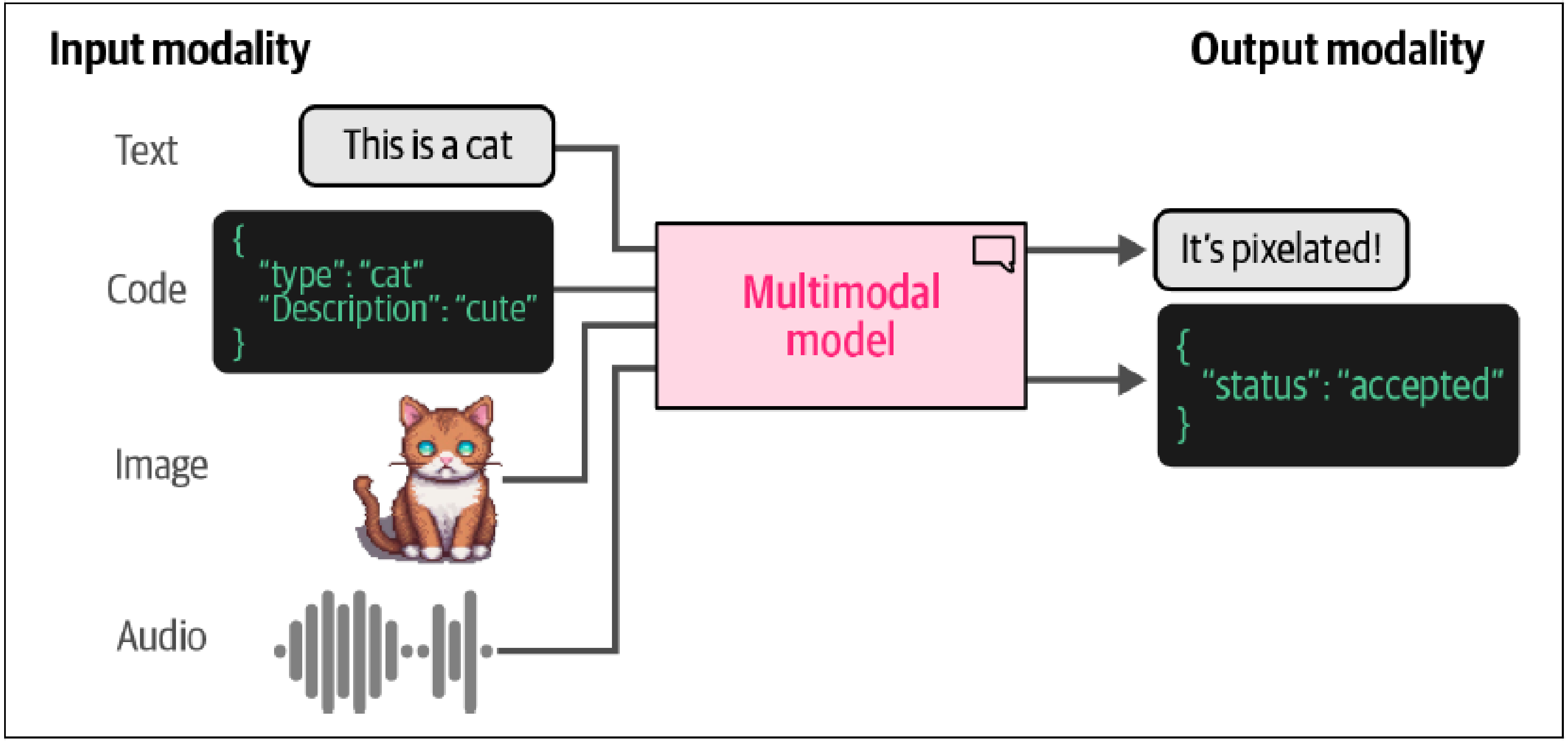


Positional Embeddings (RoPE)

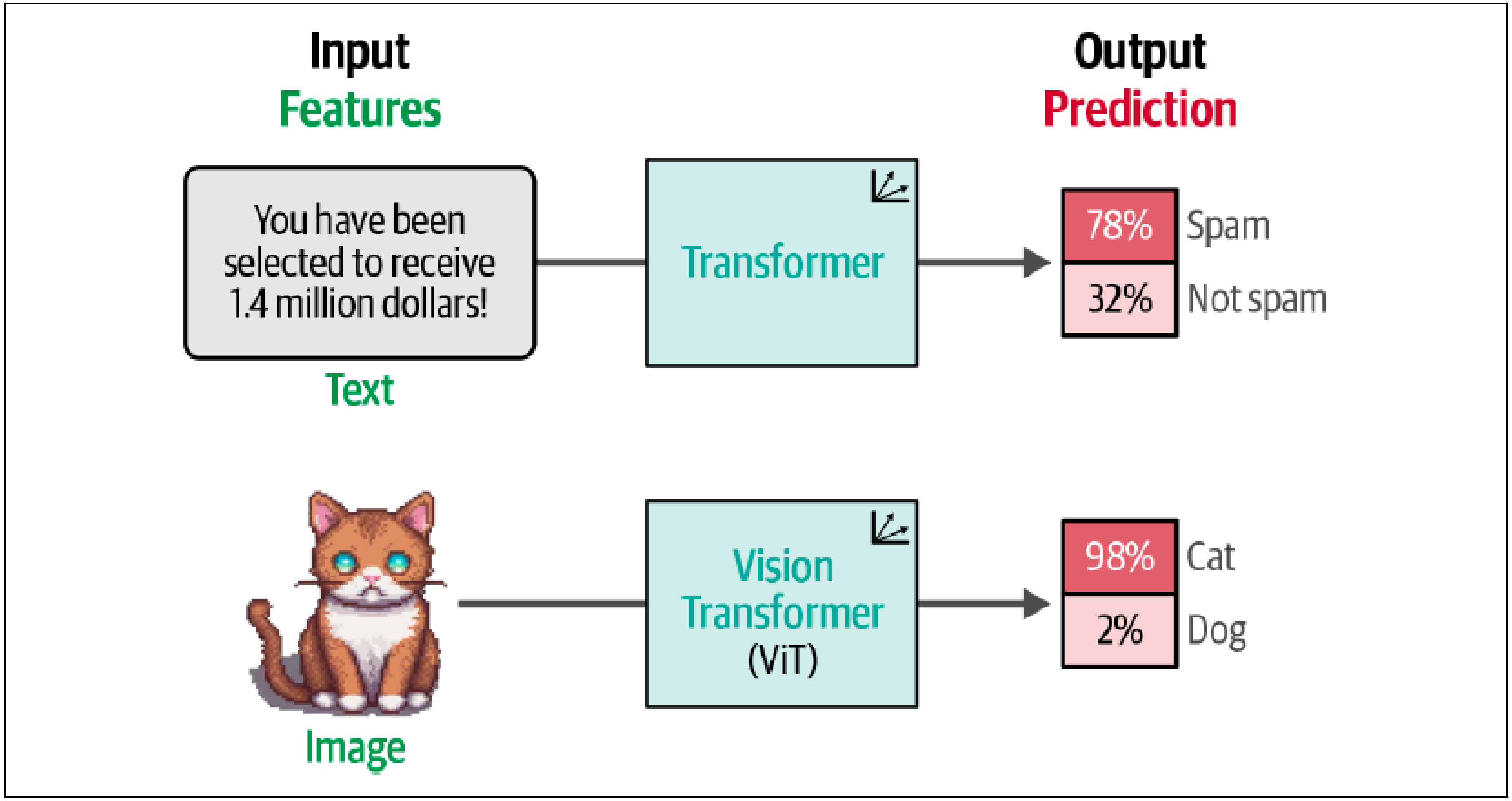


Multimodal Large Language Models

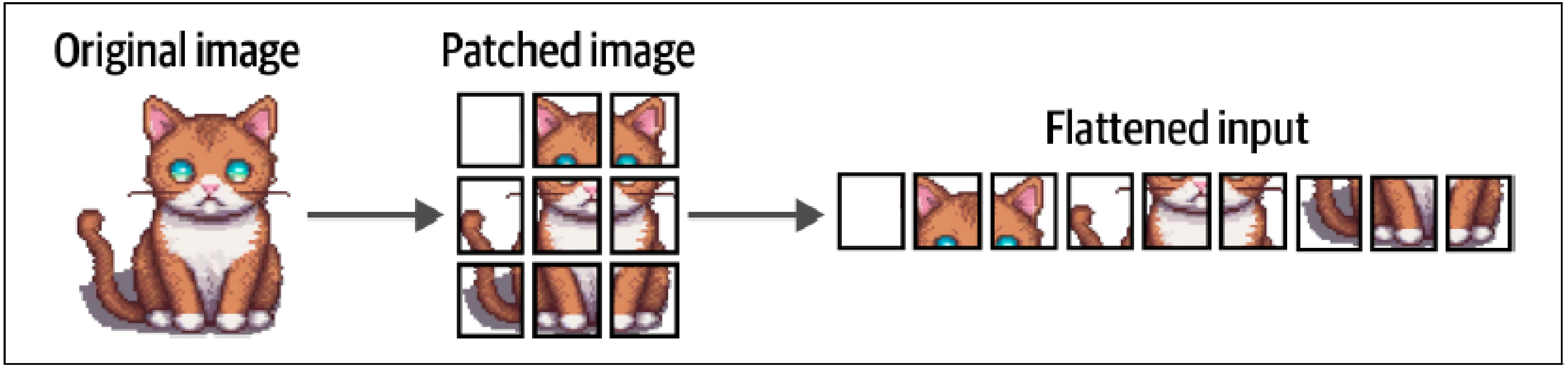
Multimodal Large Language Models



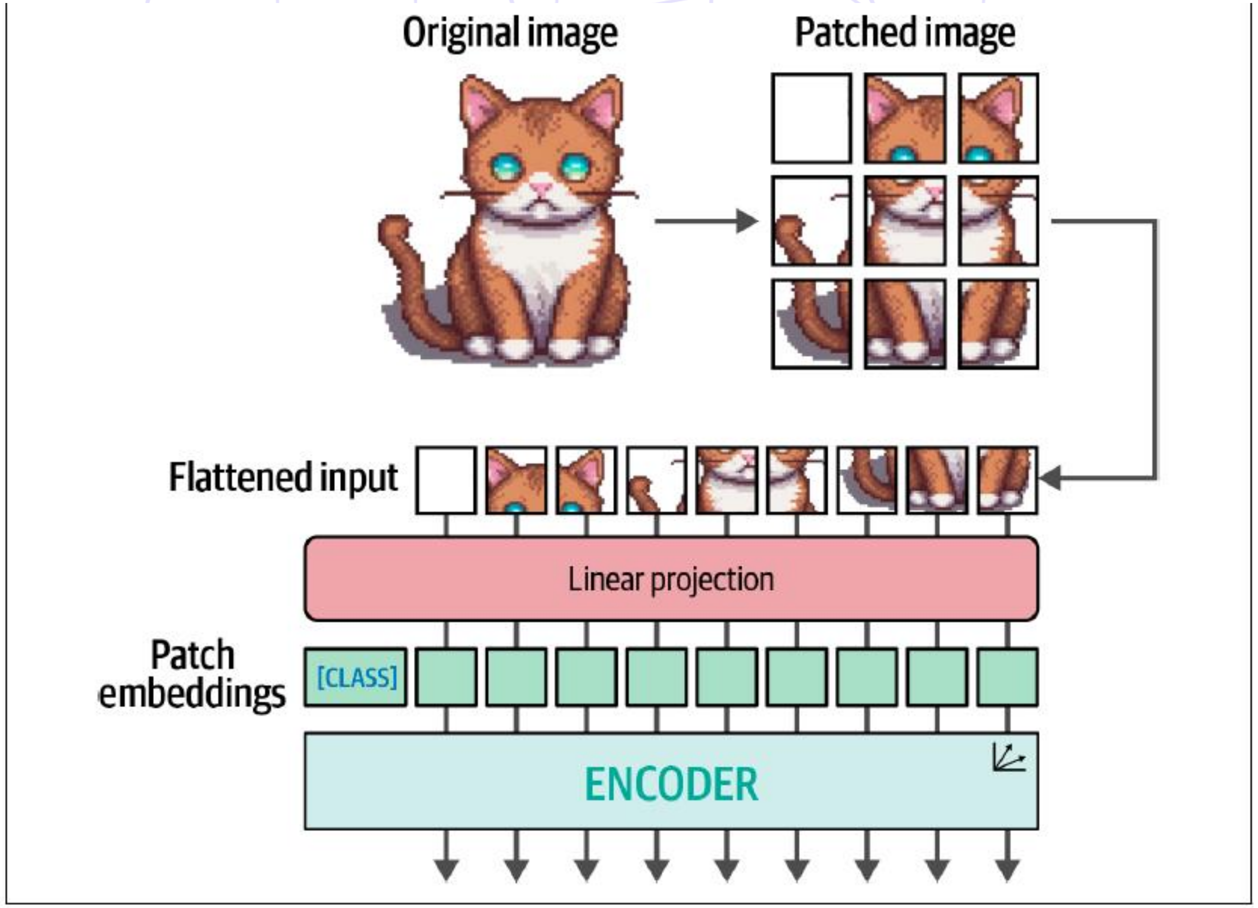
Multimodal Large Language Models



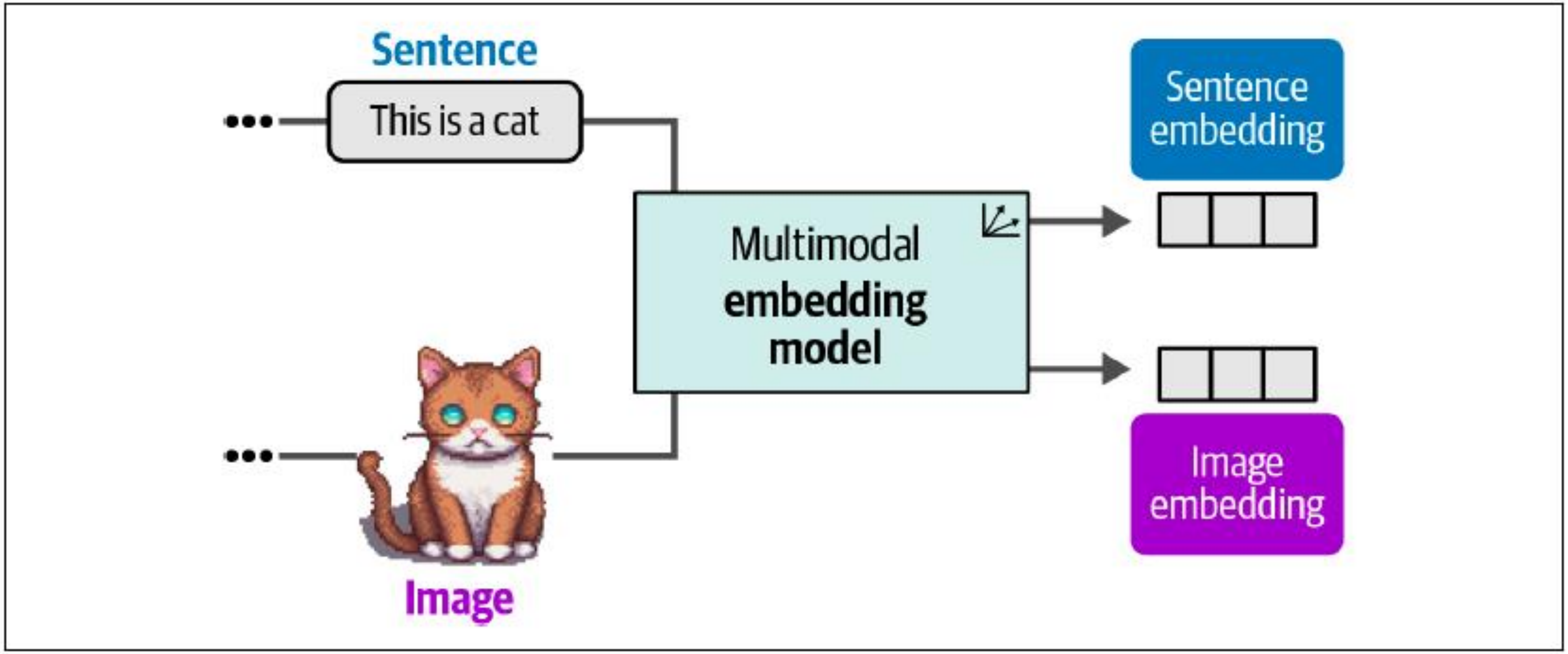
Multimodal Large Language Models



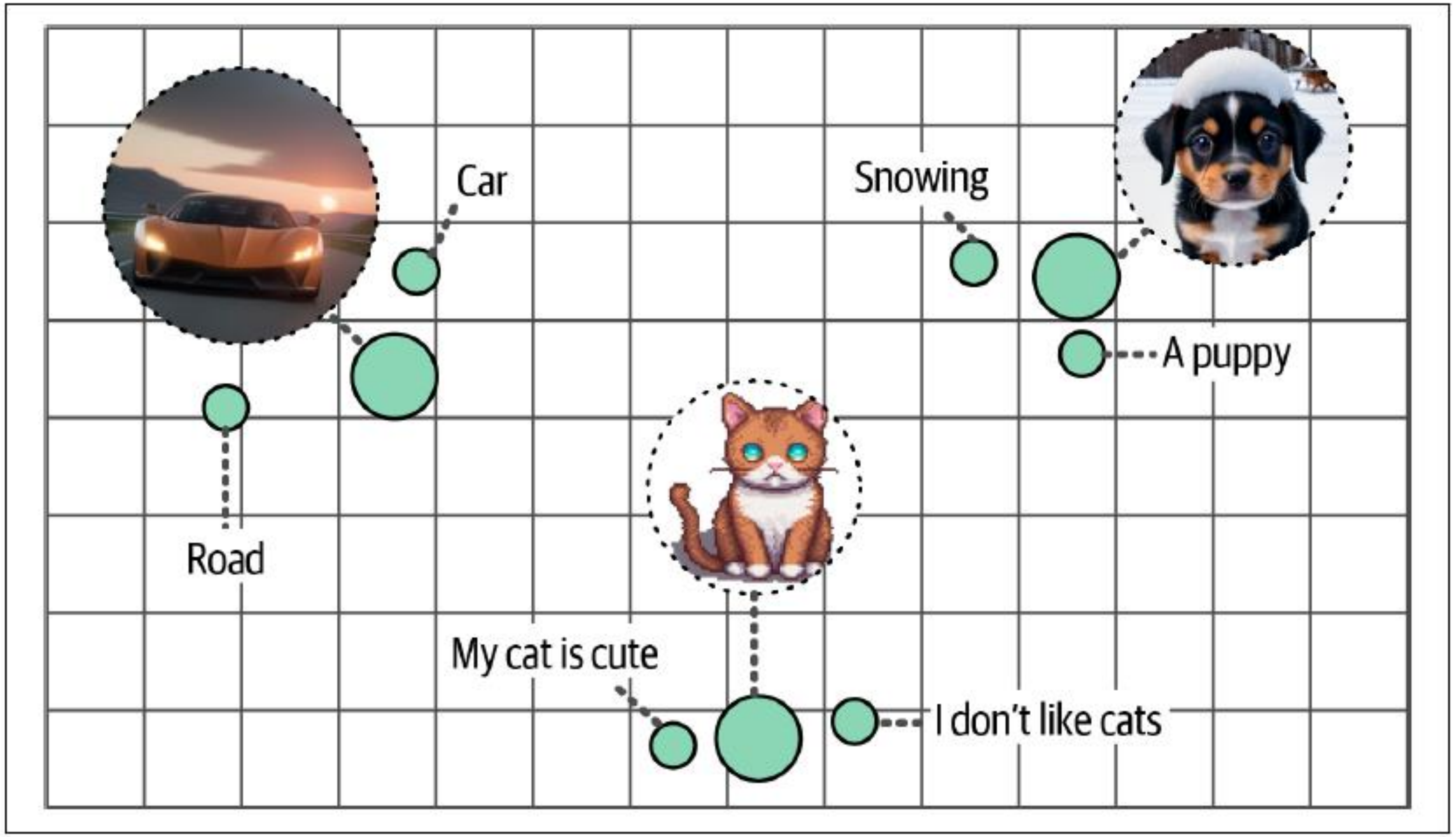
Multimodal Large Language Models



Multimodal Large Language Models



Multimodal Large Language Models



Multimodal Large Language Models

Contrastive Language-Image Pre-Training: Connecting Text and Images

CLIP: Connecting Text and Images

- **Zero-shot classification**:

We can compare the embedding of an image with that of the description of its possible classes to find wich class is most similar.

- **Clustering**

Cluster both images and a collection of keywords to find which keywords belong to which sets of images.

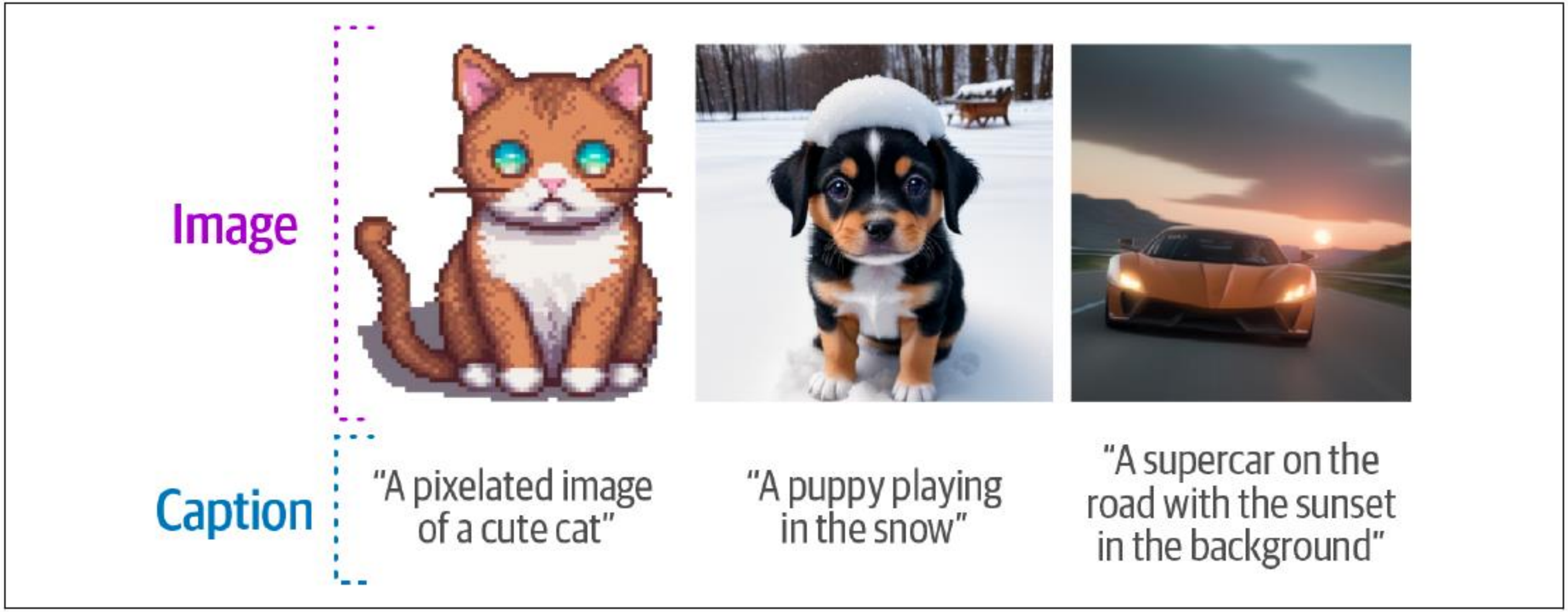
- **Search**

Across billions of texts or images, we can quickly find what relates to an input text or image.

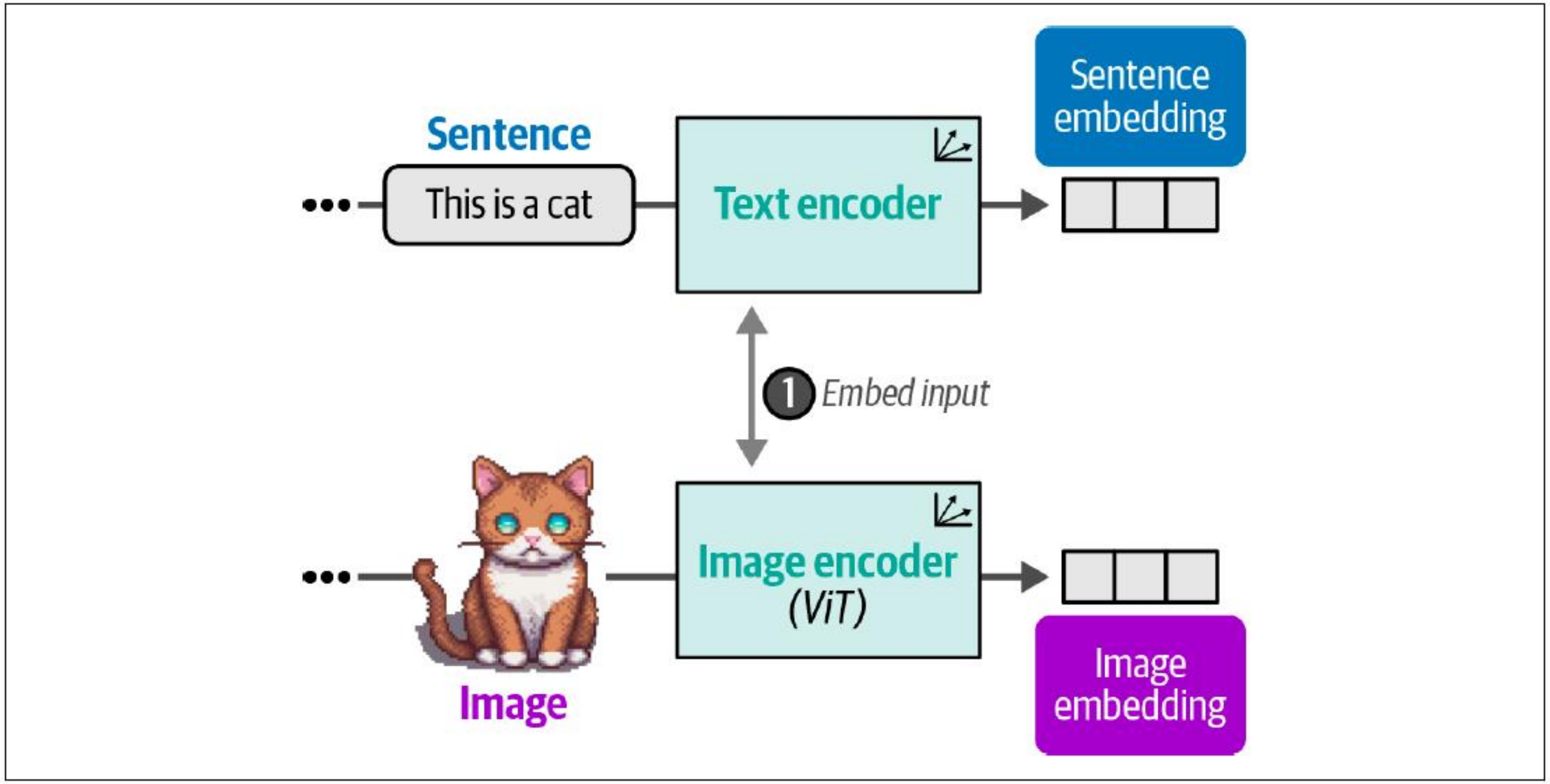
- **Generation**

Use multimodal embeddings to drive the generation of images

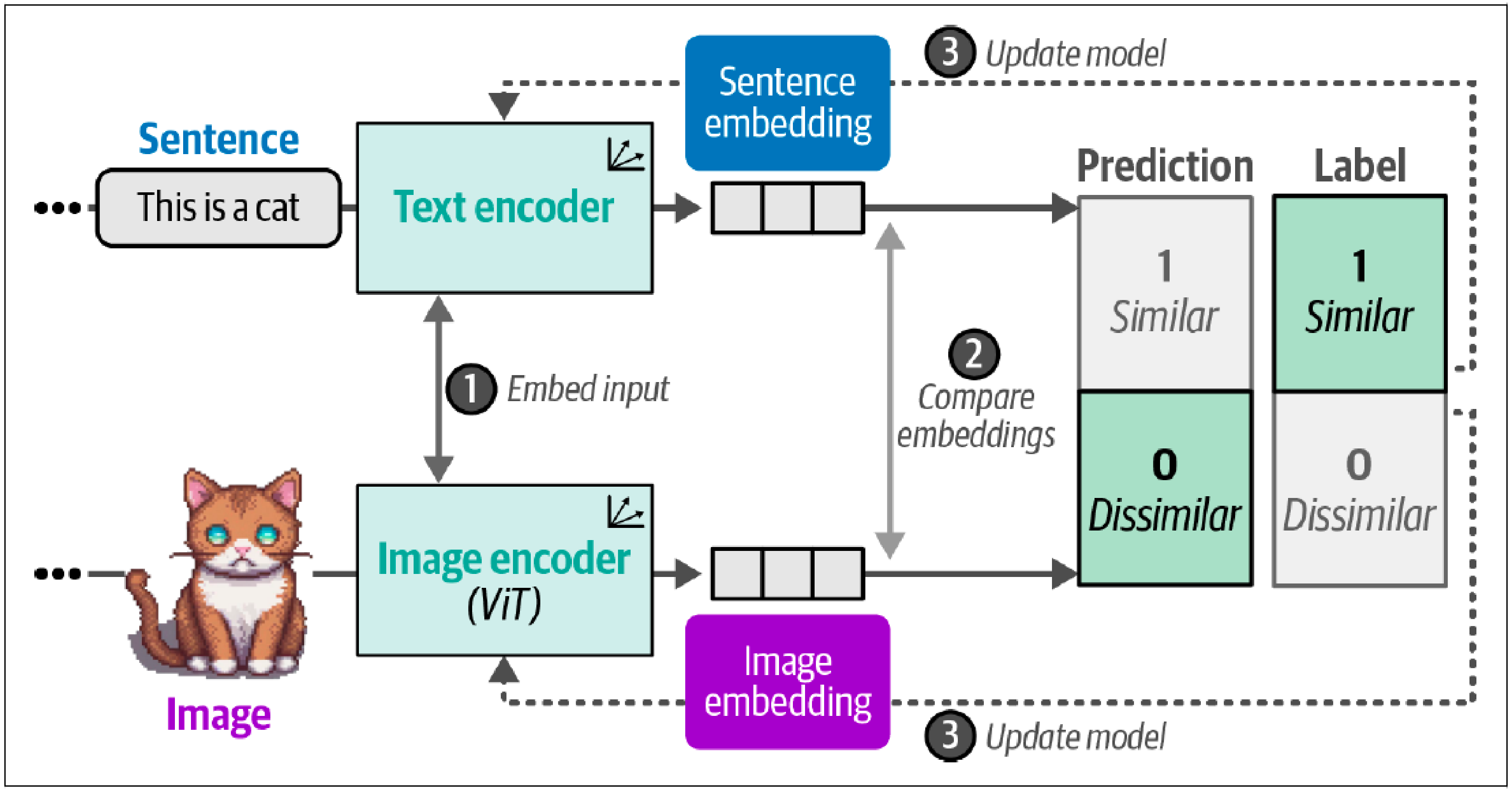
How Can CLIP Generate embeddings to drive the generation of images



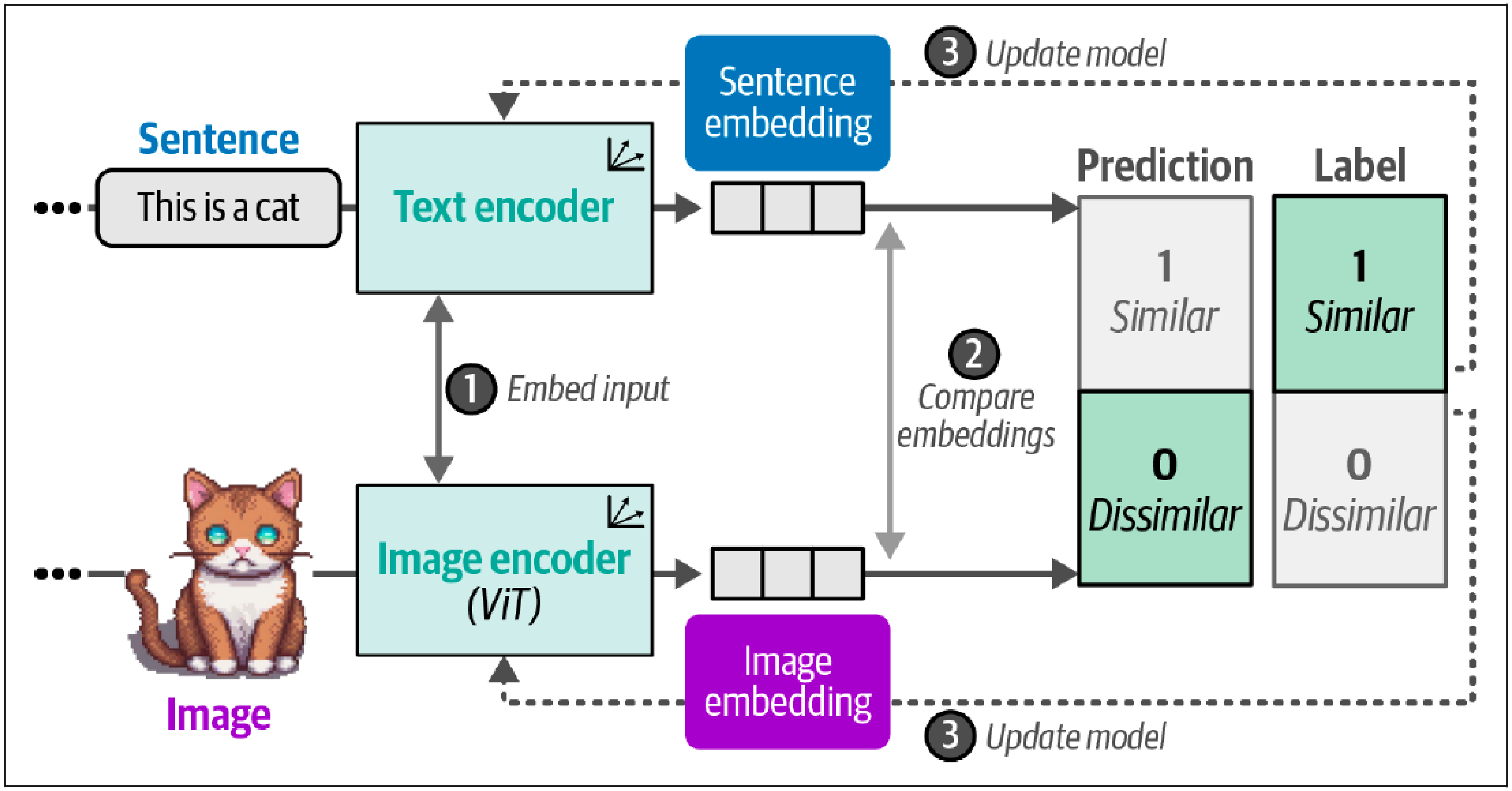
How Can CLIP Generate embeddings to drive the generation of images



How Can CLIP Generate embeddings to drive the generation of images



How Can CLIP Generate embeddings to drive the generation of images



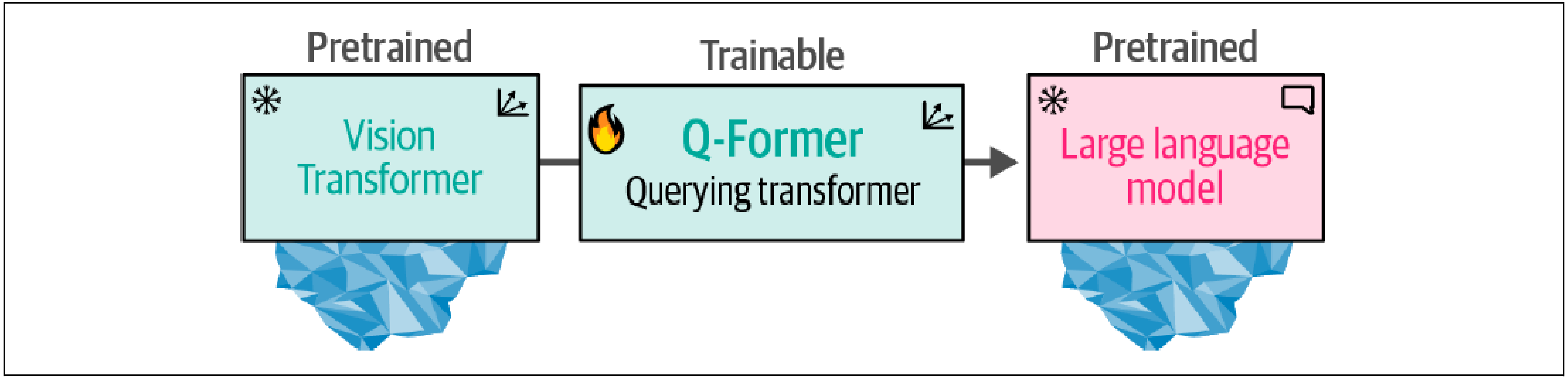
Multimodal Large Language Models

Bootstrapping Langugage-Image Pre-training for Unified Vision-Langugage Understanding and Generation 2: Briding the Modality Gap

BLIP-2: Briding the Modality Gap

- Instead of building the architecture from scratch, BLIP-2 bridges the vision-language gar by building a bridge, named the Querying Transformer (Q-Former), that connects a pretrained image encoder and a pretrained LLM.

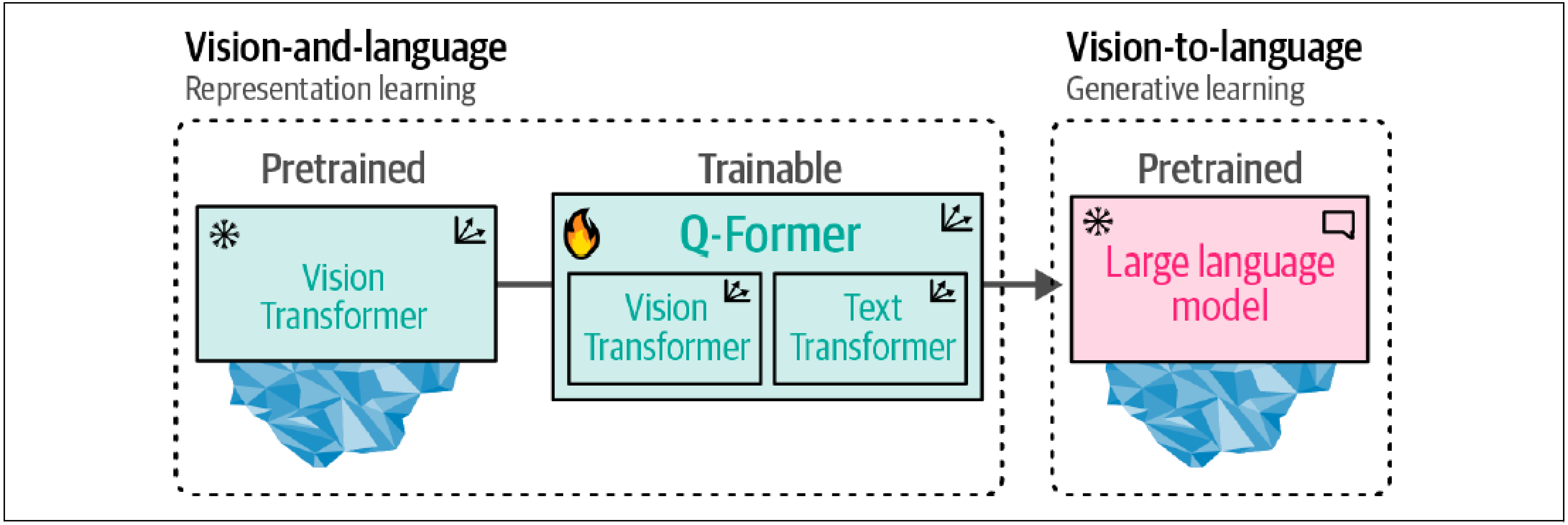
BLIP-2: Briding the Modality Gap



BLIP-2: Briding the Modality Gap

- To connect the two pretrained models, the Q-Former mimics their architectures. It has two modules that share their attention layers:
    - An Image Transformer to interact with the frozen Vision Transformer for feature extraction.
    - A Text transformer that can interact with the LLM.

BLIP-2: Briding the Modality Gap



BLIP-2: Briding the Modality Gap

- With these inputs, the Q-Former is then trained on three tasks:

Image-text contrastive learning

This task attempts to align pairs of image and text embeddings such that they maximize their mutual information.

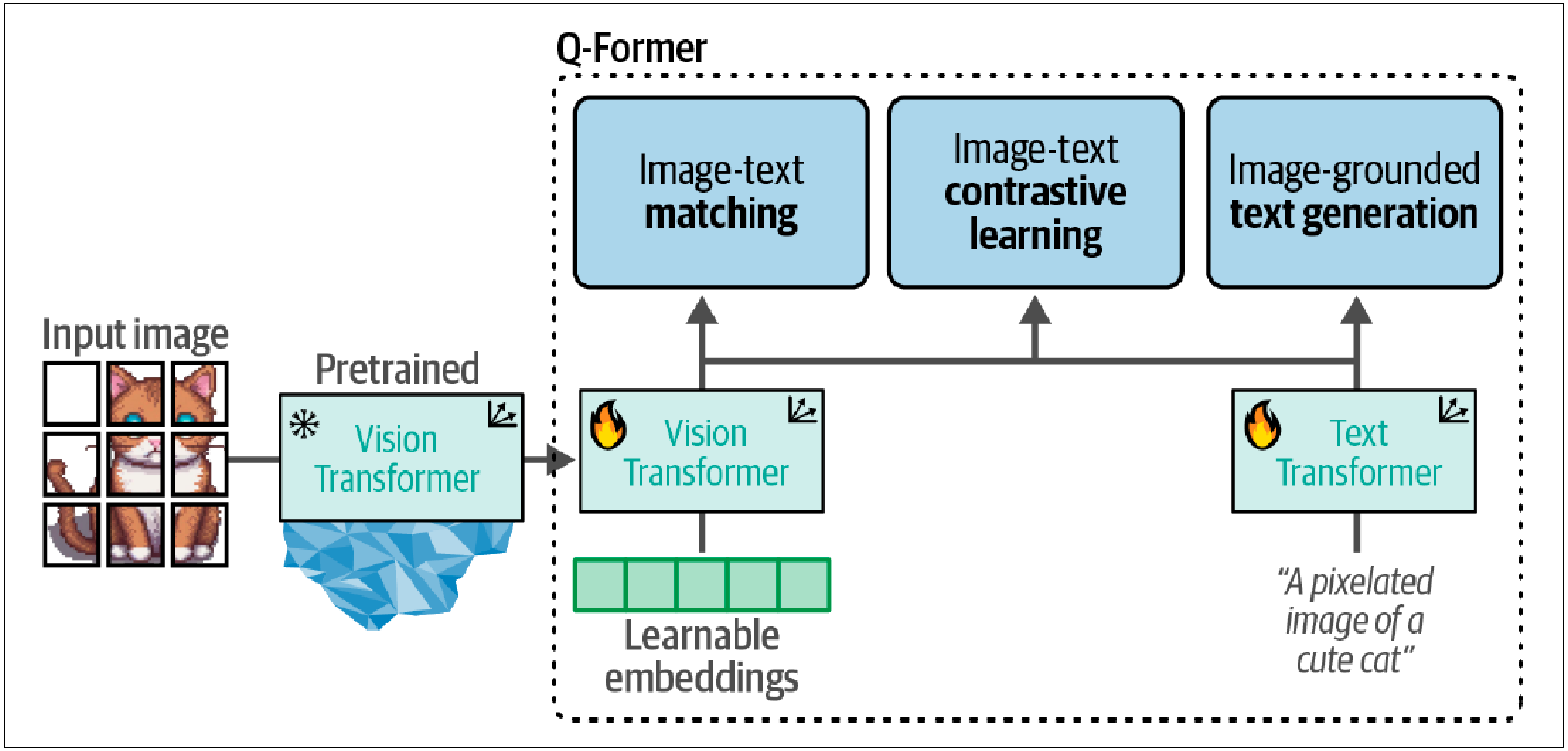
Image-text matching

A classification task to predict whether an image and text pair is positive (matched) or negative (unmatched).

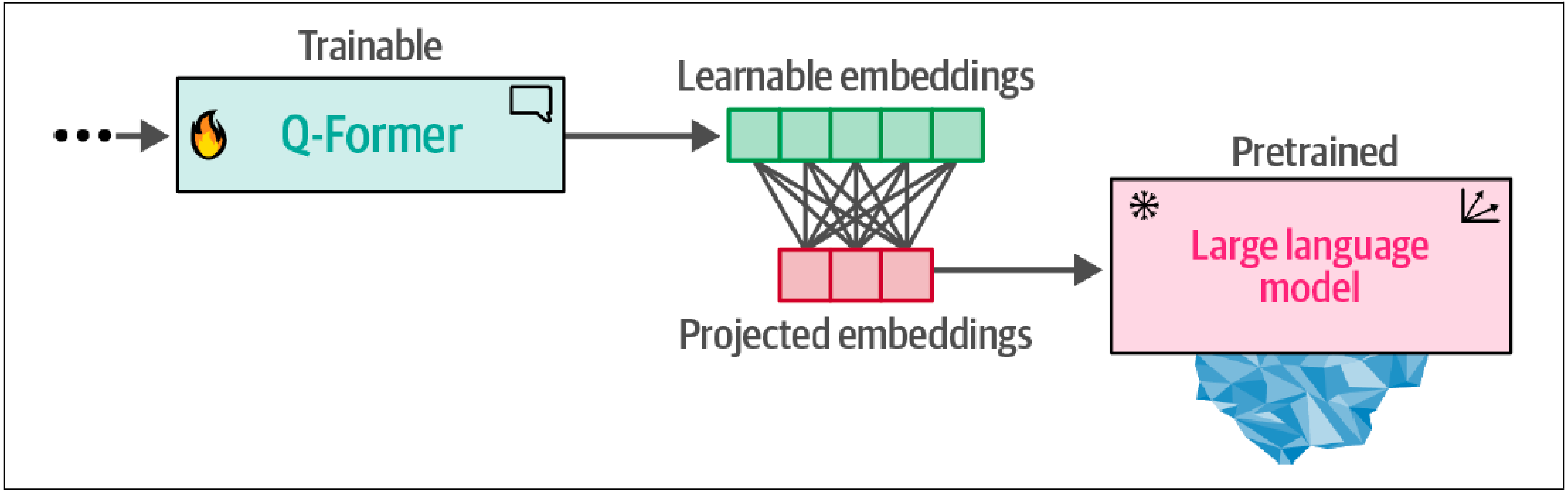
Image-grounded text generation

Trains the model to generate text based on information extracted from the input image.

BLIP-2: Briding the Modality Gap



BLIP-2: Briding the Modality Gap



BLIP-2: Briding the Modality Gap