CSCI-567: Discussion Session 09/22

Joshua Robinson, Oliver Liu

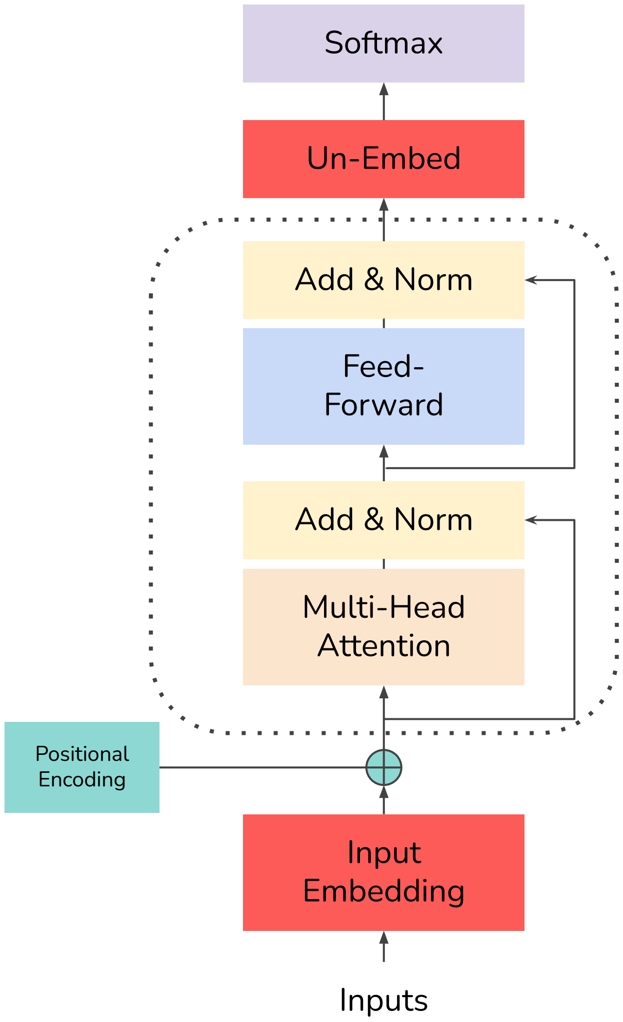
The Transformer

The Transformer Architecture

- Designed for **Sequence Learning**;
- Very **scalable** (up to trillions of parameters and training tokens);
- Built on stack of **self-attention layers**.
- Useful resources:
  - The Illustrated GPT-2,
  - The Annotated Transformer.

Model Overview

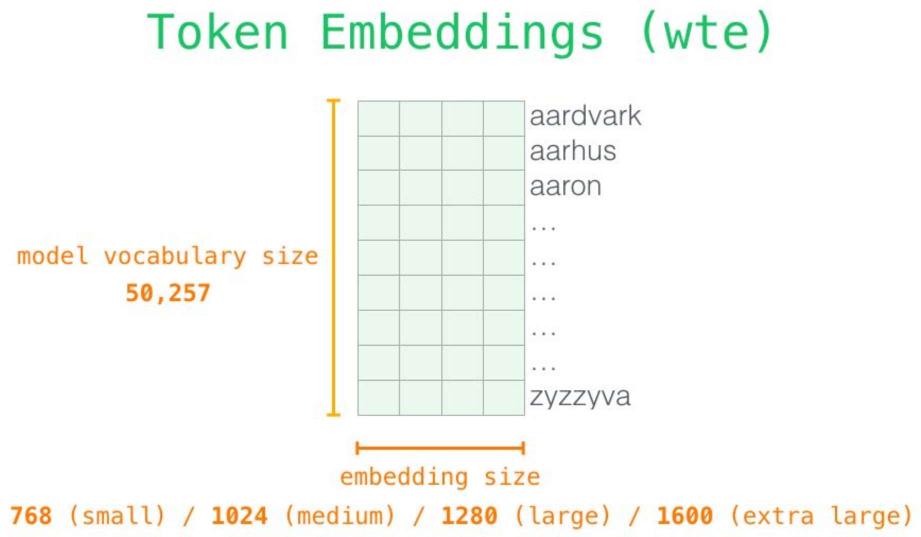
- Inputs
- Embedding Matrix
- Positional Encoding
- Self-Attention Layer
- Un-embed Matrix
- Softmax

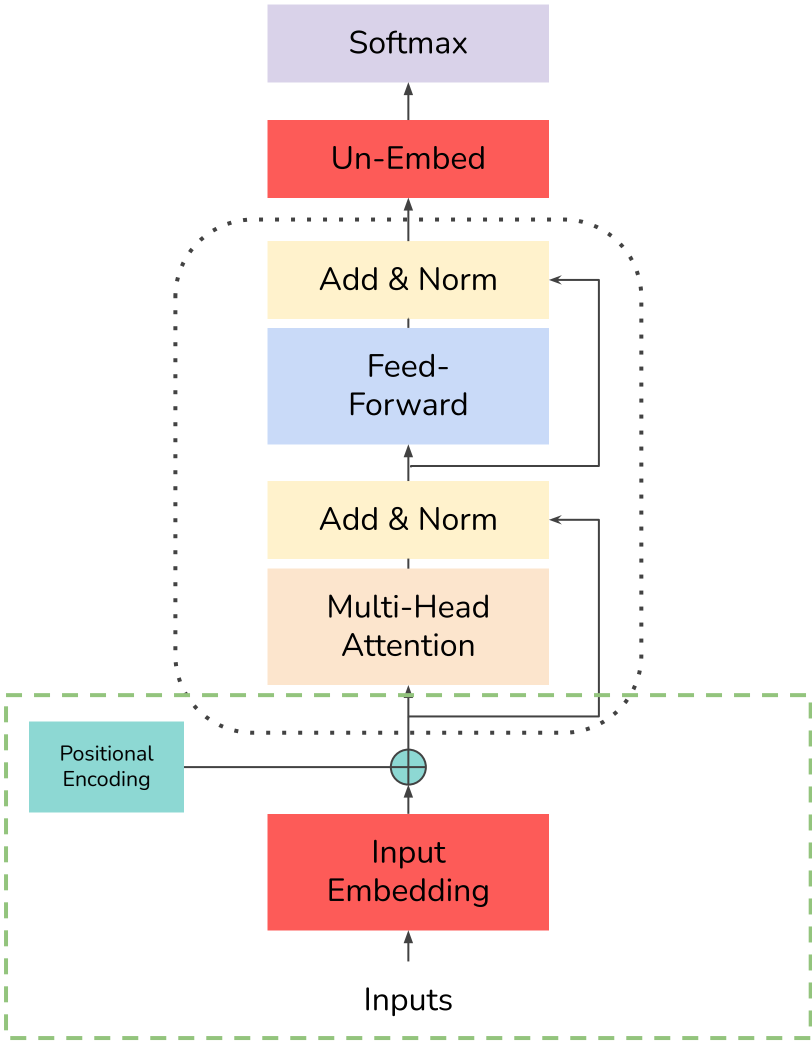


Inputs and Embedding

- **Input in natural language:** *I have a dream.*
- **Vocabulary**: a mapping between each word to a unique index in {0, 1, ..., V}
- **Tokenized input**: [x_1, x_2, x_3, x_4] a sequence of length *4*
    - x_i: index of the corresponding vocabulary (e.g. x_3 is the index of *dream*)
- We feed tokenized input to the **Input Embedding E,** a matrix of size V x D
    - Input to the first self-attention layer: $\mathbf{X}$ = [E_{x1}, E_{x2}, E_{x3}, E_{x4}] of shape 4 x D

Inputs and Embedding

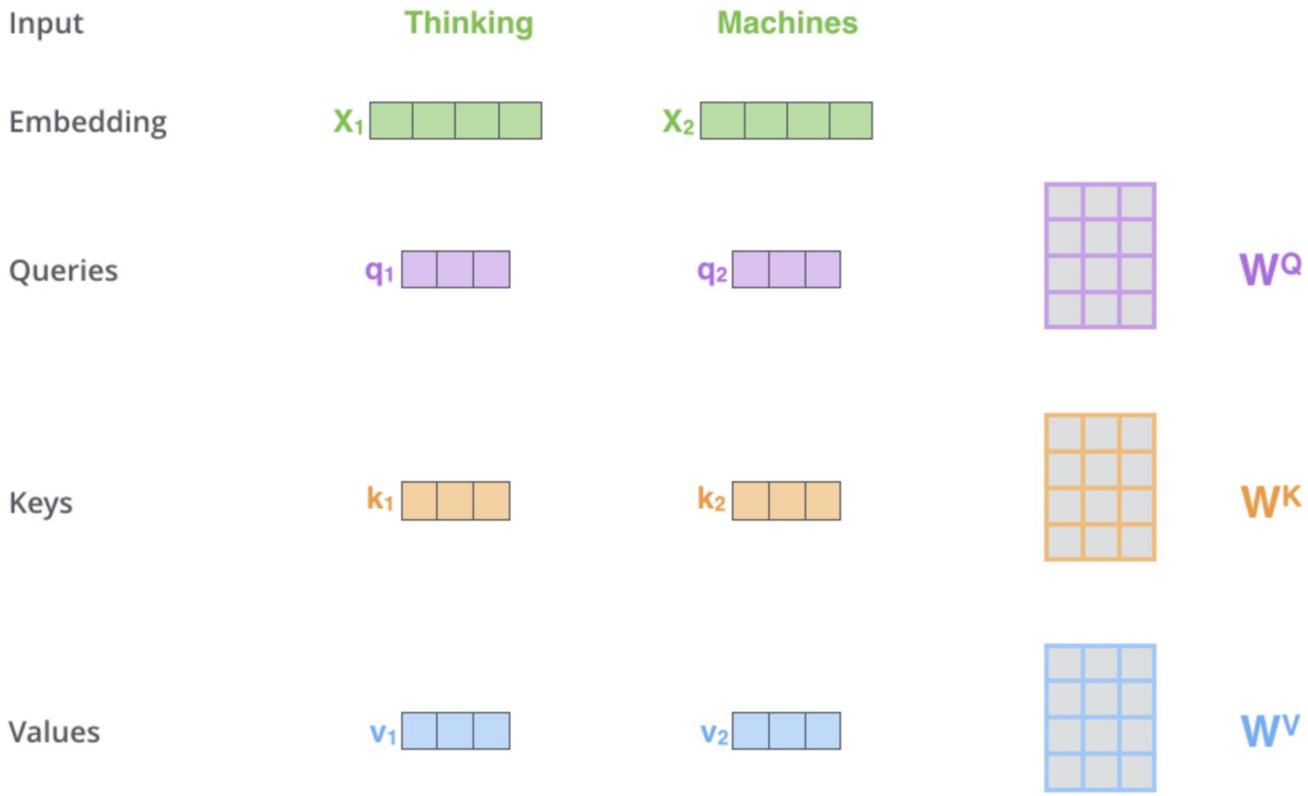




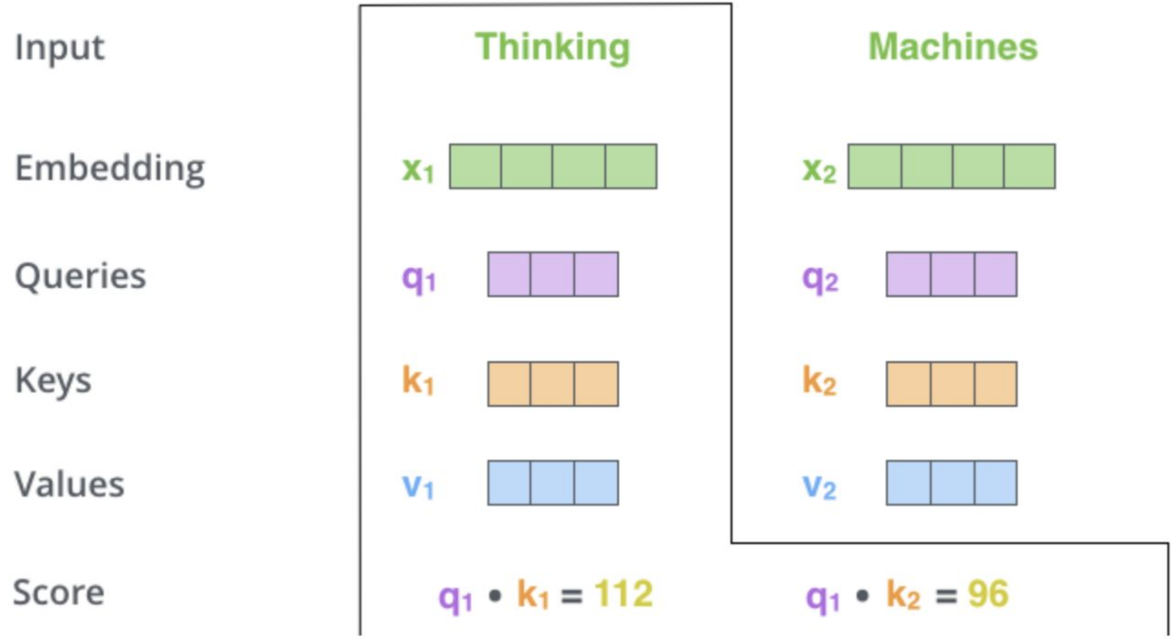
Self-Attention

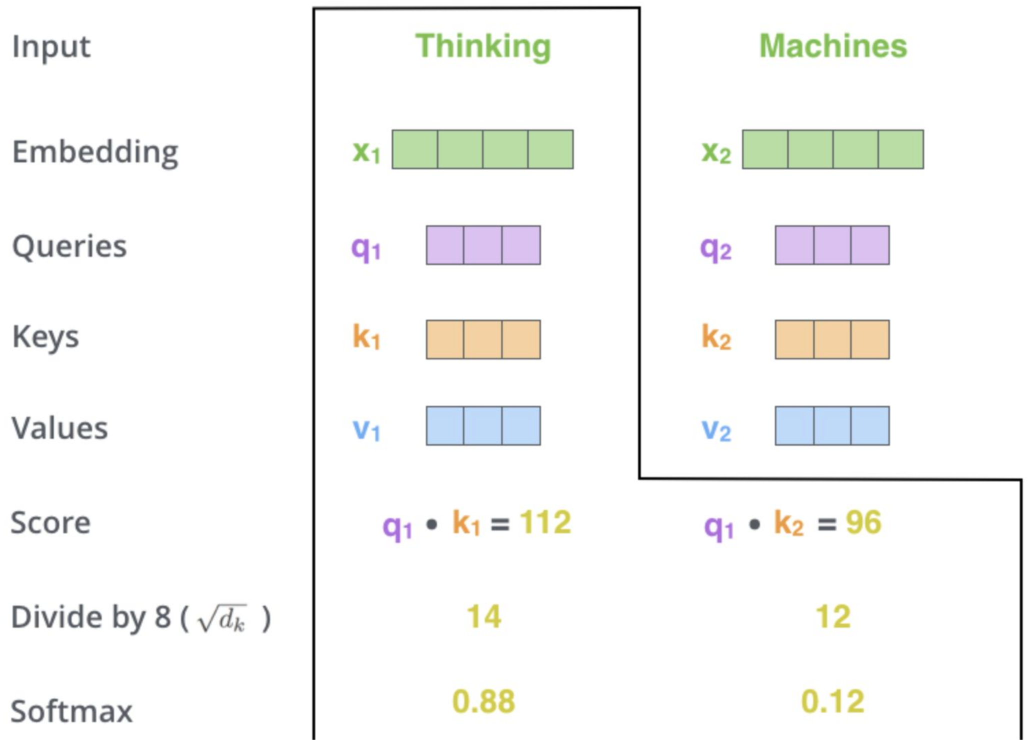
- **Attention**: Given a **Q**uery, computed a weighted average of **V**alues based on the query's similarity with **K**eys.
- **Self-Attention**: **Q, K, V** are linear transformations of **X**.
  - **Q, K, V** are of size T x D
  - What is the size of the attention output?

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



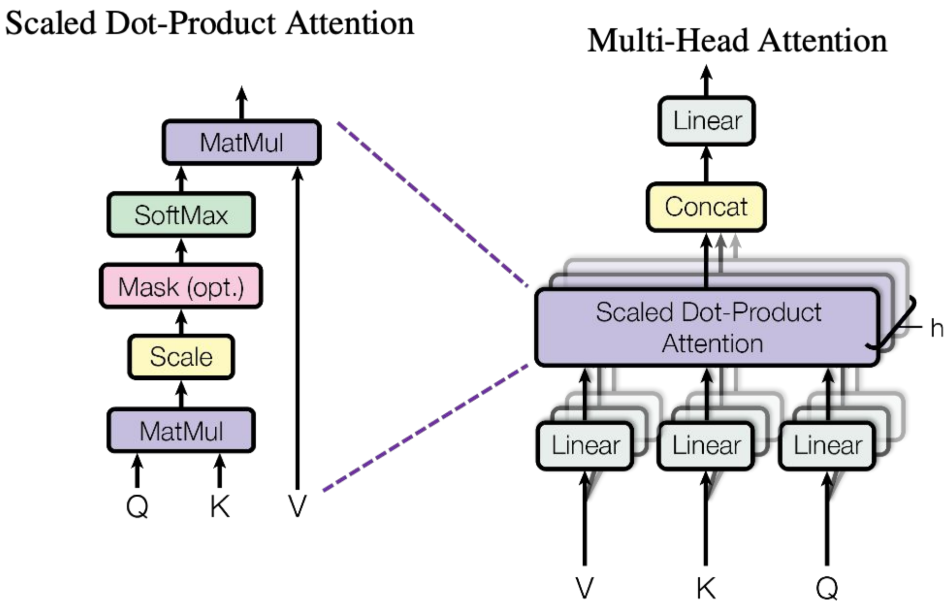
Input
Thinking
Machines
Embedding
X1
X2
Queries
q1
q2
WQ
Keys
k1
k2
WK
Values
v1
v2
WV


Input
Thinking
Machines
Embedding
x1
x2
Queries
q1
q2
Keys
k1
k2
Values
v1
v2
Score
q1 • k1 = 112
q1 • k2 = 96



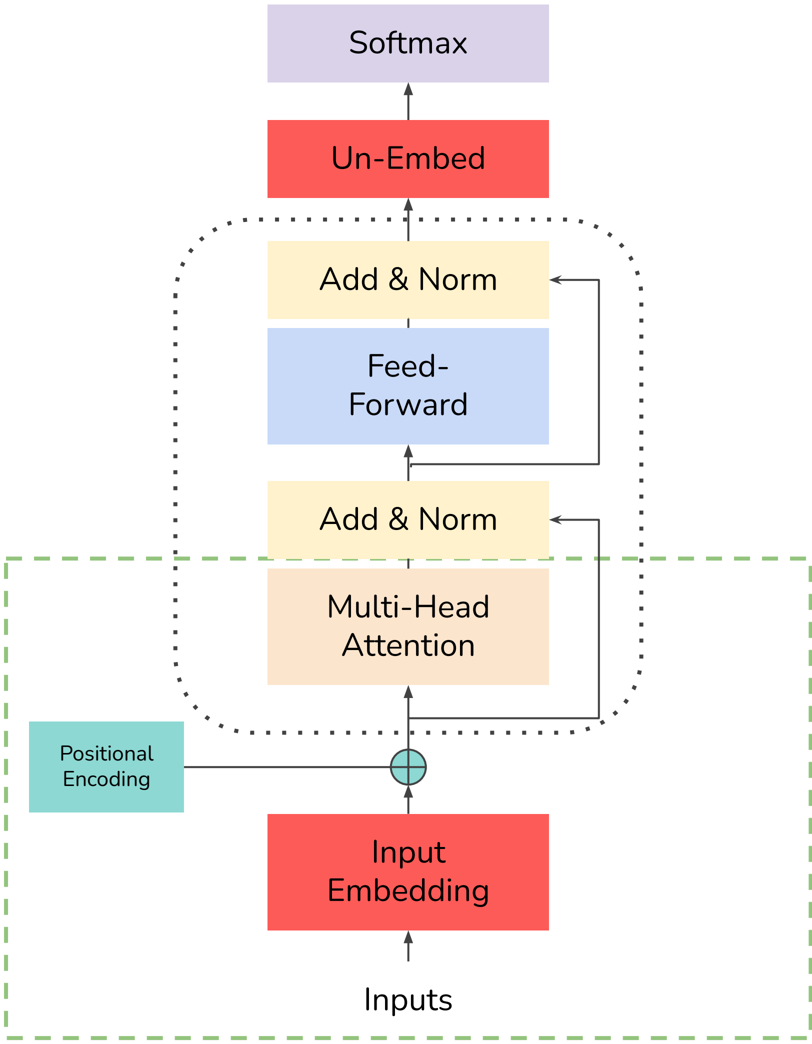
Input
Thinking
Machines
Embedding
x1
x2
Queries
q1
q2
Keys
k1
k2
Values
v1
v2
Score
q1 • k1 = 112
q1 • k2 = 96
Divide by 8 ($\sqrt{d_k}$)
14
12
Softmax
0.88
0.12

Multi-Head Attention



Story so far

- **Input embedding** convert a set of discrete tokens to continuous vectors.
- The **attention mechanism** computes a representation of each token position based on their relevance with other tokens.
- The attention output is of size T x D.
- We ignore add & norm for now…

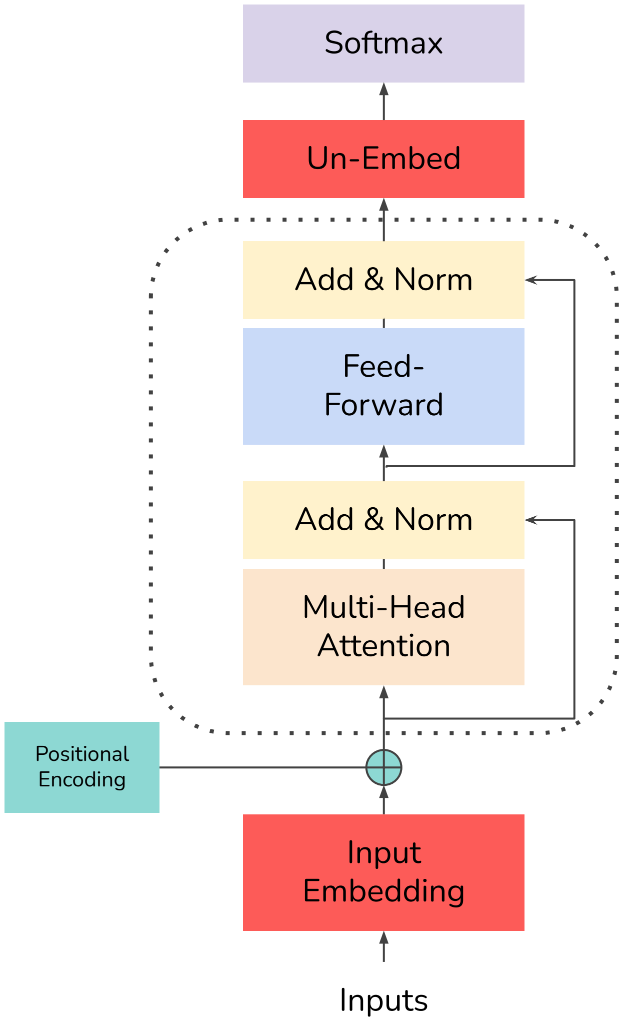


Positional Feed-forward

- Apply the same MLP at each token position.
- **Input**: $[E_1, E_2, \ldots, E_T]$ of shape in T x D
- **PFF** computes: $[MLP(E_1), MLP(E_2), \ldots, MLP(E_T)]$ for the same *MLP*!
- **Output** is of the same shape!

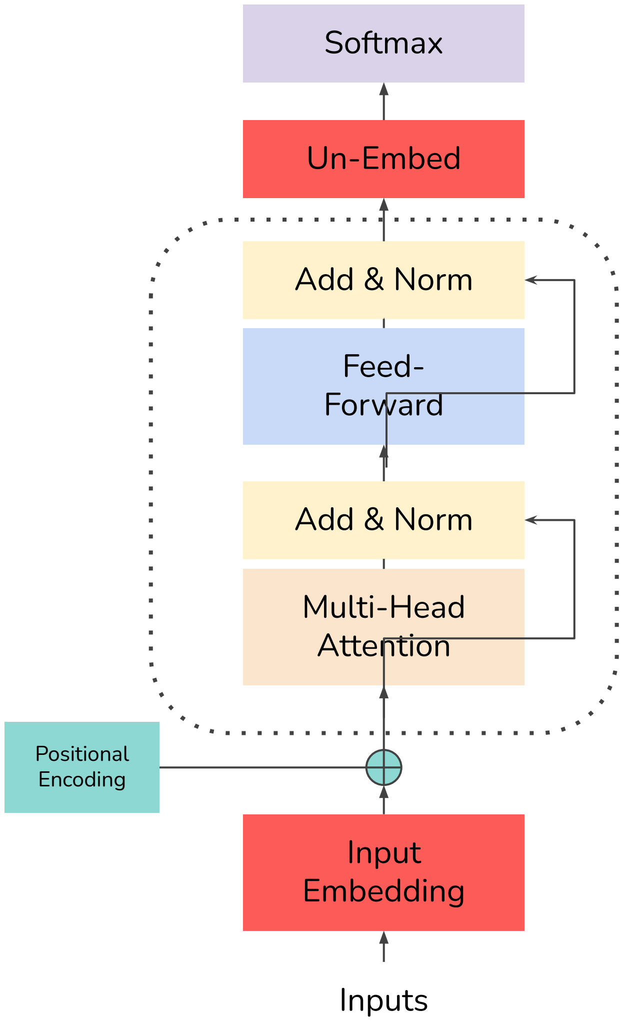
Add & Norm

- **Add**, aka **residual connection**: $X \leftarrow X + f(X)$
- f is **Self-Attention** and **MLP** in transformer
- **LayerNorm** normalizes hidden vectors of each token position.



Un-Embed

- Make a multi-class prediction at each position. This is just softmax regression!
- What is the dimension of the un-embed matrix?
- Typically, this is the transpose of the embedding matrix.



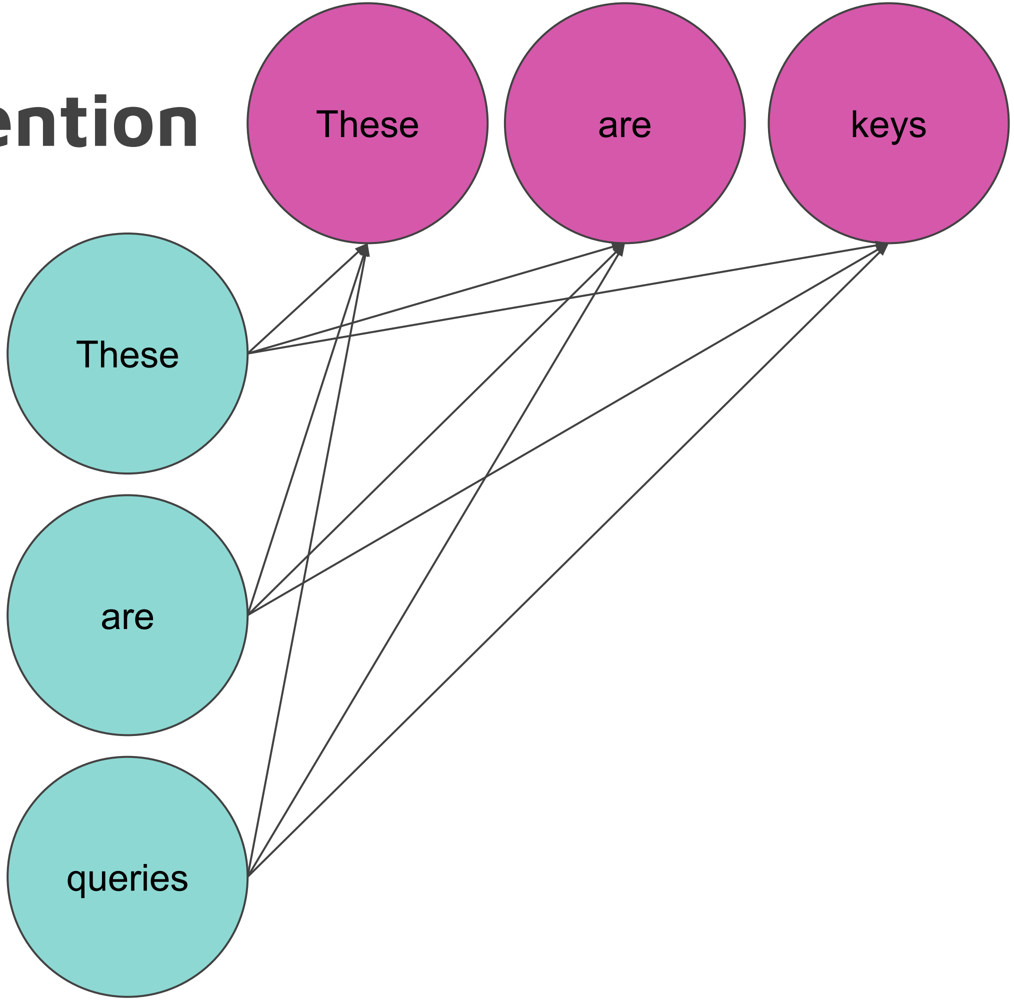
Transformer Variants

Two Flavors of Self-Attention

- Multi-Head Attention: Every query can attend to every key
- Masked Multi-Head Attention: Each query can only attend to its associated key and keys earlier in the sequence
    - In practice this done by setting masked positions to a very low number before softmax

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

Multi-Head Attention



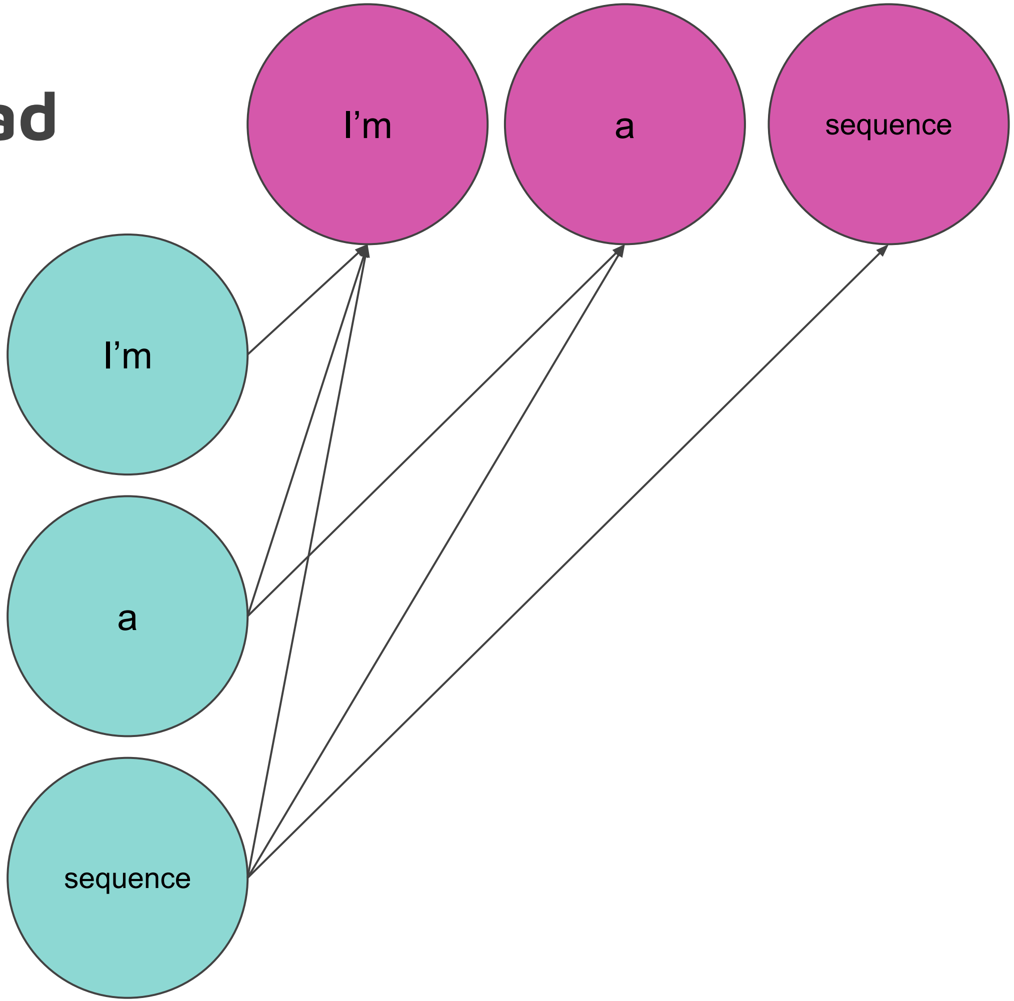
- Queries are in blue circles
- Keys are in pink circles

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

Masked Multi-Head Attention

- Queries are in blue circles
- Keys are in pink circles

- The spots with no link here would be set to a very low number before softmax



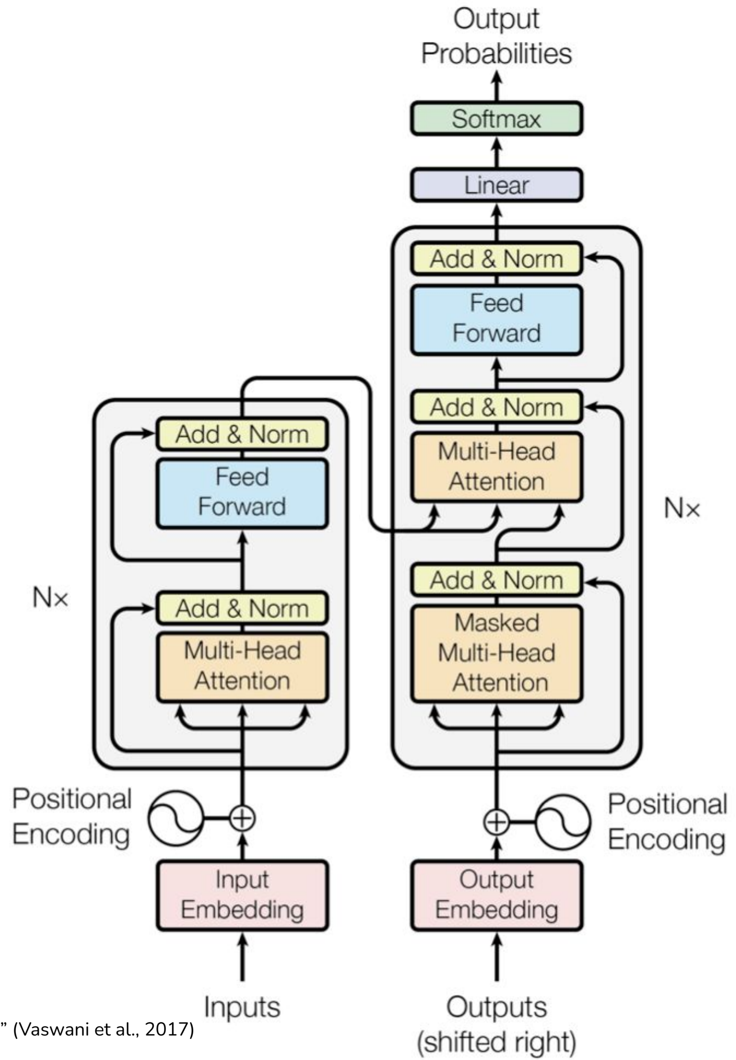
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The Transformer Family

- Encoder-decoder models
  - Original Transformer
  - T5
- Encoder-only models
  - BERT
- Decoder-only models
  - GPT-2
  - GPT-3
  - ChatGPT

Encoder-Decoder Model

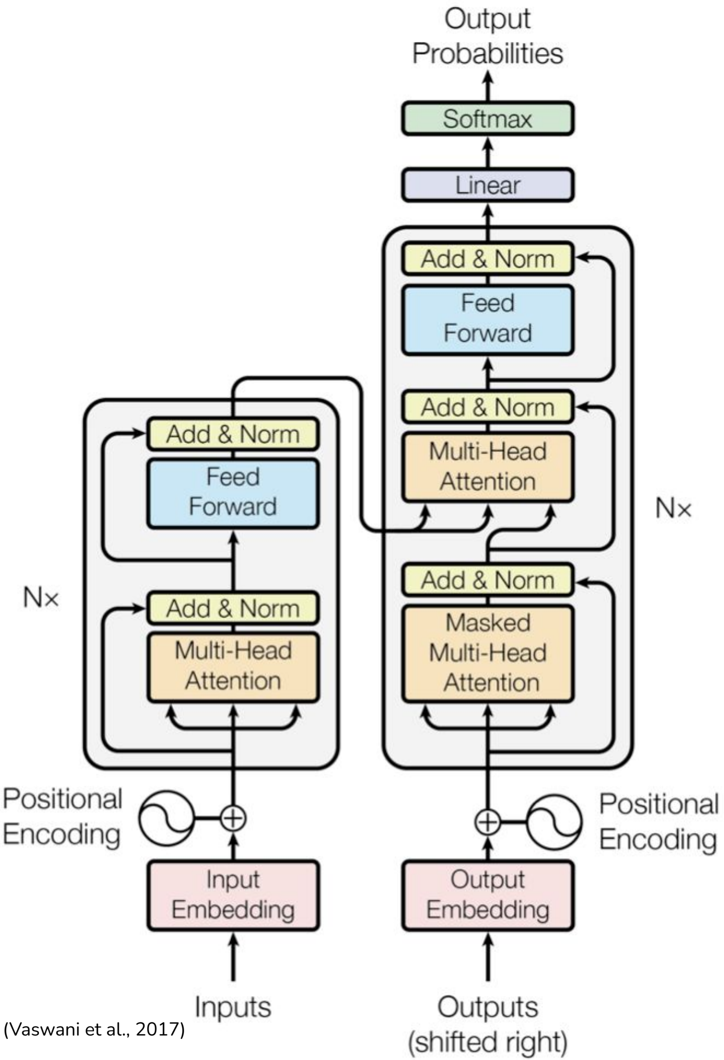
- Machine translation
- Summarization



This figure is from “Attention is All You Need” (Vaswani et al., 2017)

The Original Transformer

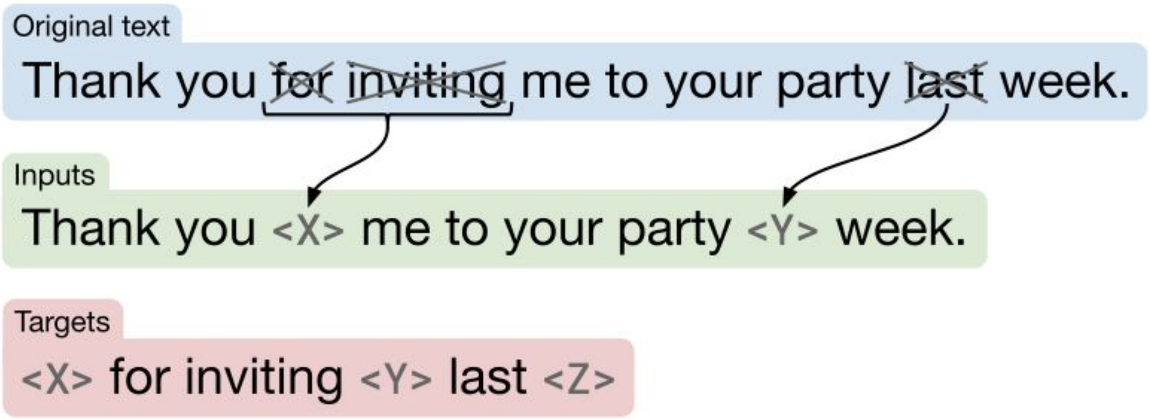
- [Learn more]
- Input like “How are you?”
- Outputs like “<s> Cómo estás?”
- Targets like “Cómo estás? <e>”
- Minimize cross entropy between predictions and labels



This figure is from “Attention is All You Need” (Vaswani et al., 2017)

T5

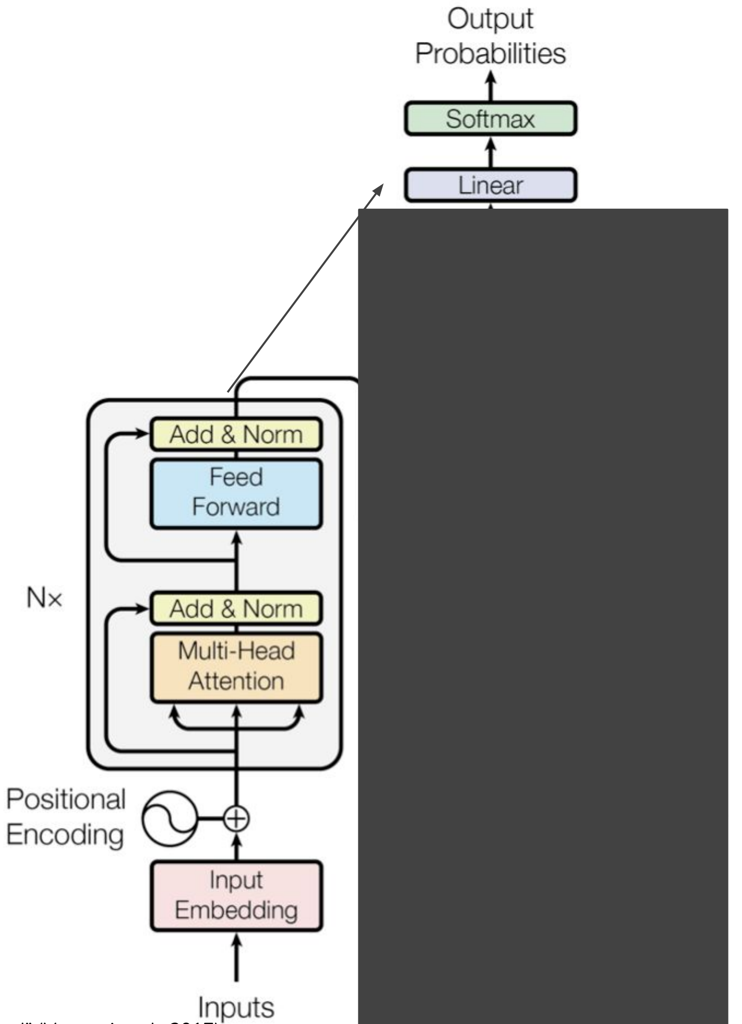
- [Learn more]
- “Span corruption” objective
- Minimize cross entropy between predictions and labels



This figure is from “Exploring the Limits of Transfer Learning” (Raffel et al., 2019)

Encoder-Only Model

- Text classification
- Sentence representation



Original figure is from “Attention is All You Need” (Vaswani et al., 2017)

BERT

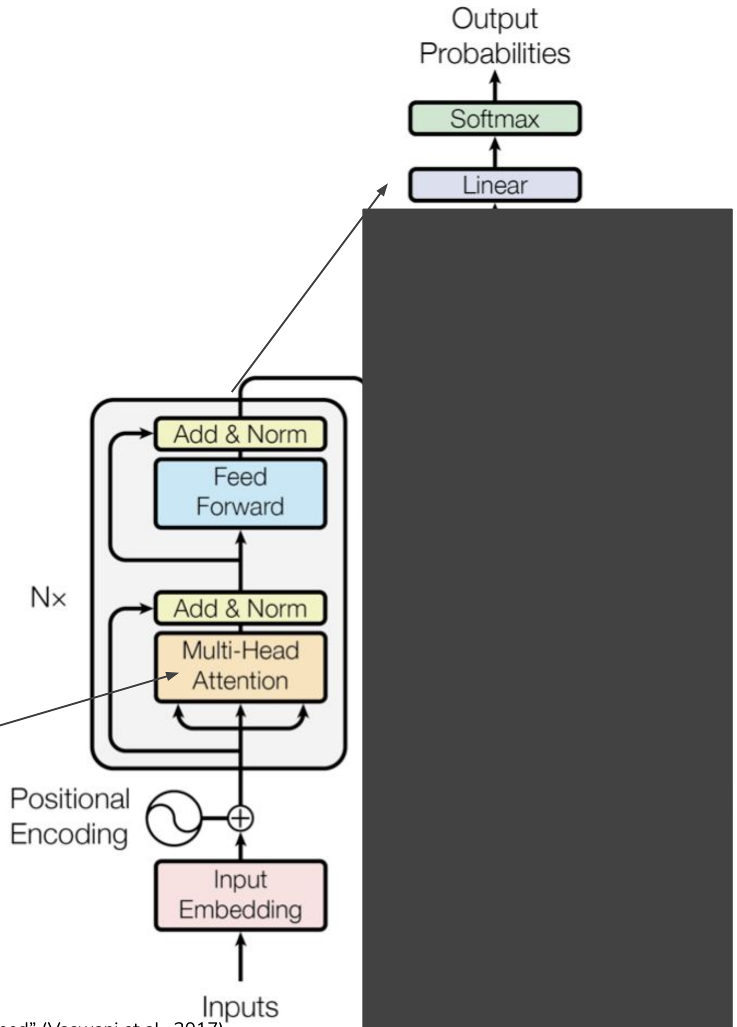
- [Learn more]
- Objective is masked language modeling (MLM)
  - For 15% of tokens
    - Replace 80% with [MASK]
    - Replace 10% with a random token
    - Leave 10% unchanged
- Minimize cross entropy between predictions and labels
- Input is something like “I saw a [MASK] on my way to class tennis morning.”
- Use [CLS] token for classification
- So many related models: RoBERTa, DeBERTa, ALBERT, ELECTRA, etc.

Image is from https://muppet.fandom.com/wiki/Ber

Decoder-Only Model

- Language modeling

MASKED!



Original figure is from “Attention is All You Need” (Vaswani et al., 2017)

GPT-2

- Learn more
- Objective is next-token prediction
- Minimize cross entropy between predictions and labels

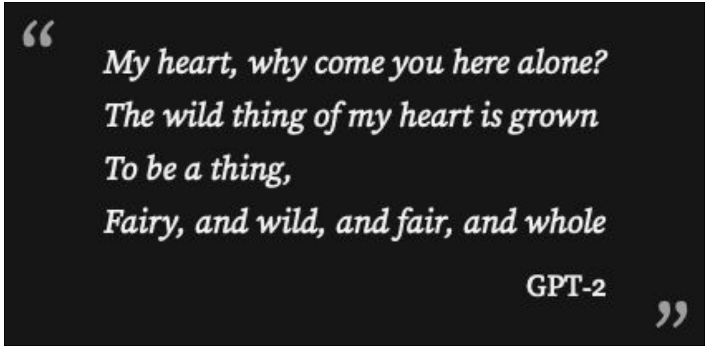
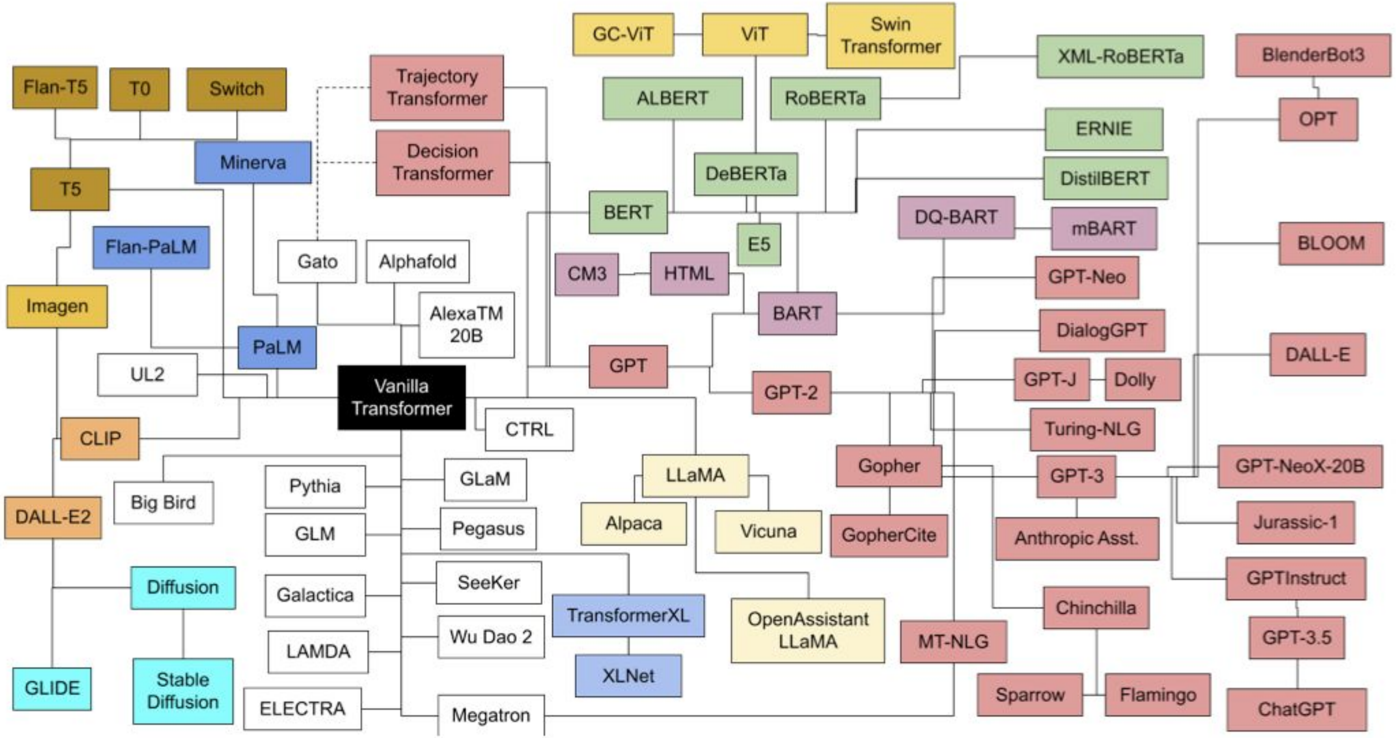


Image is from https://gwern.net/gpt-2

Tip of the Iceberg



The figure is from "Transformer Models: an introduction and catalog" (Amatriain et al., 2023)