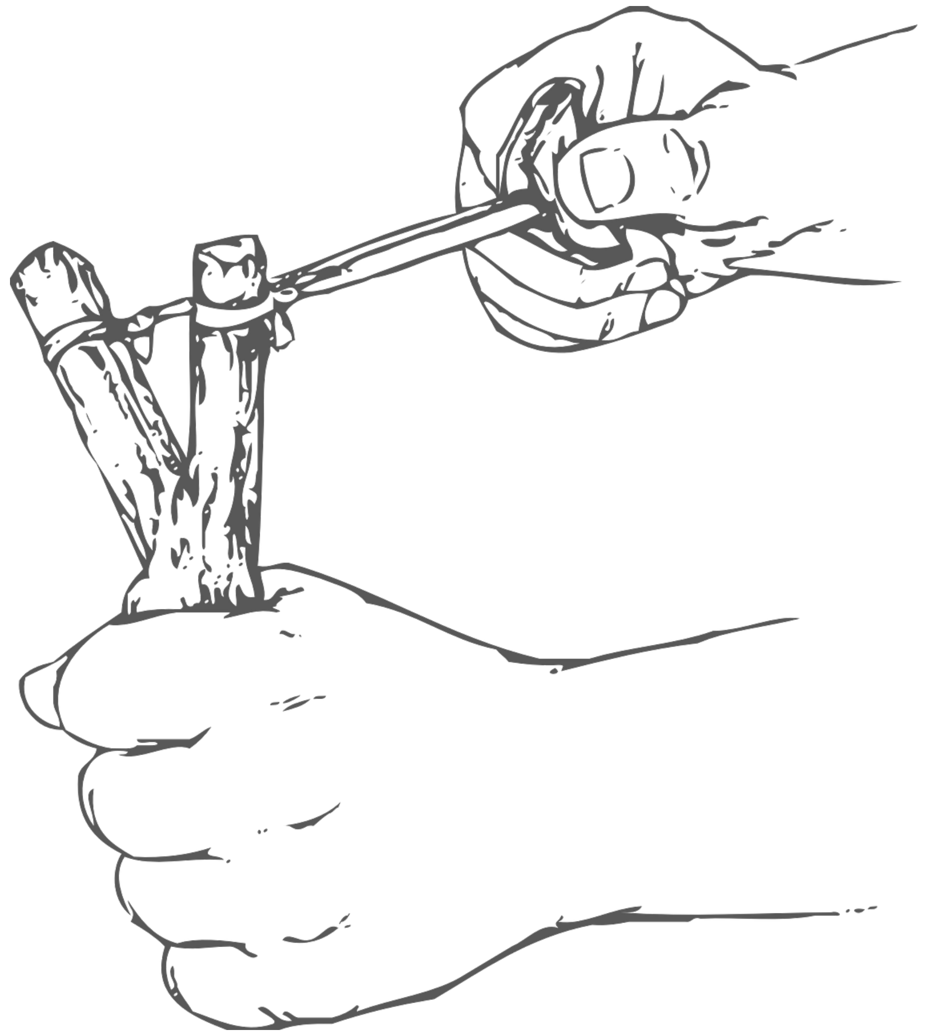
Neural·Pragmatic Natural Language Generation

N·P NLG

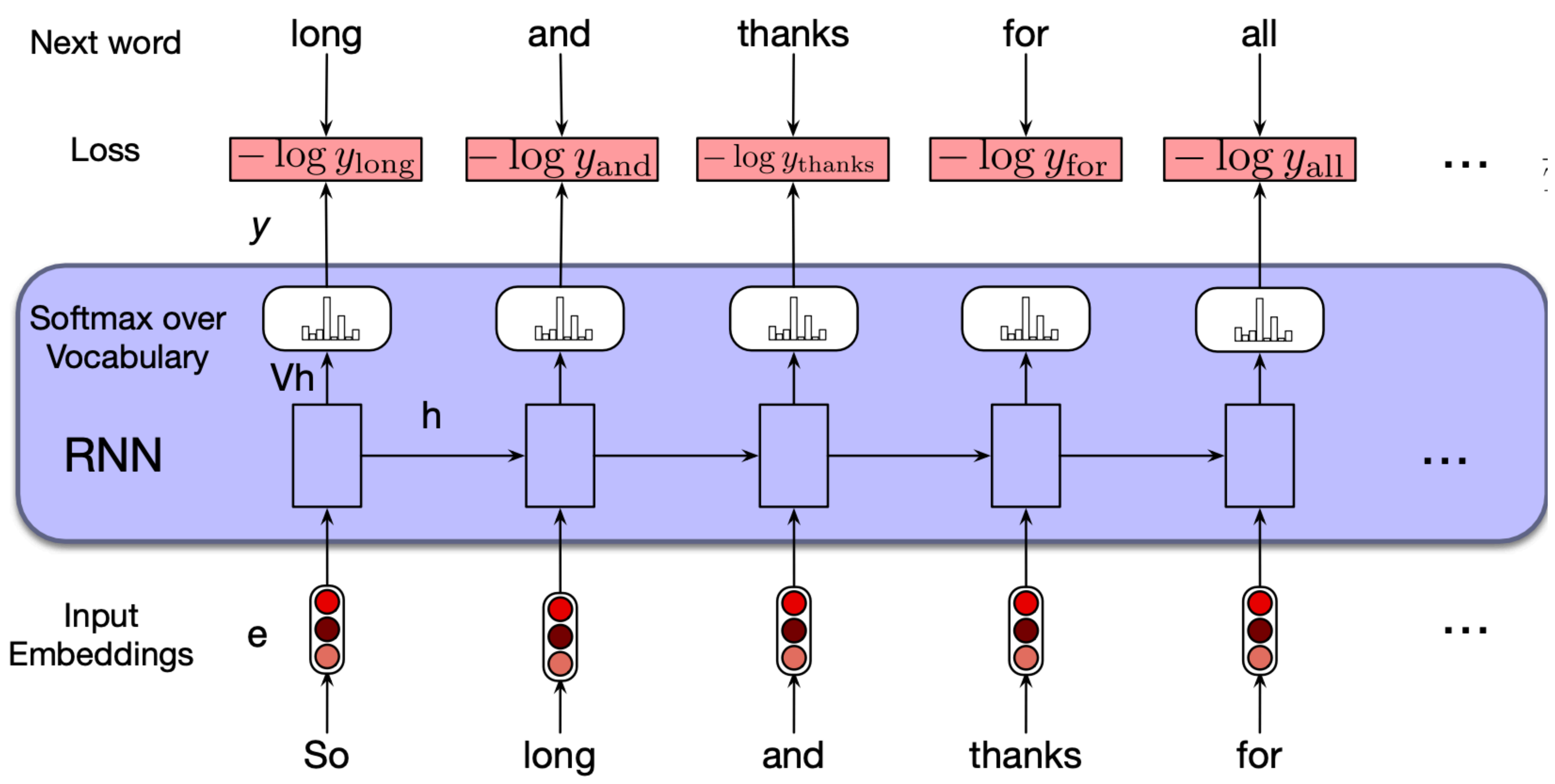
Learning goals



1. understand motivation and basic architecture of transformer based LLMs
   a. self-attention & transformer blocks
   b. heads and layers
   c. positional encodings
   d. uni- vs bidirectional architectures
2. become acquainted with using the 'transformers' package to access pre-trained LLMs

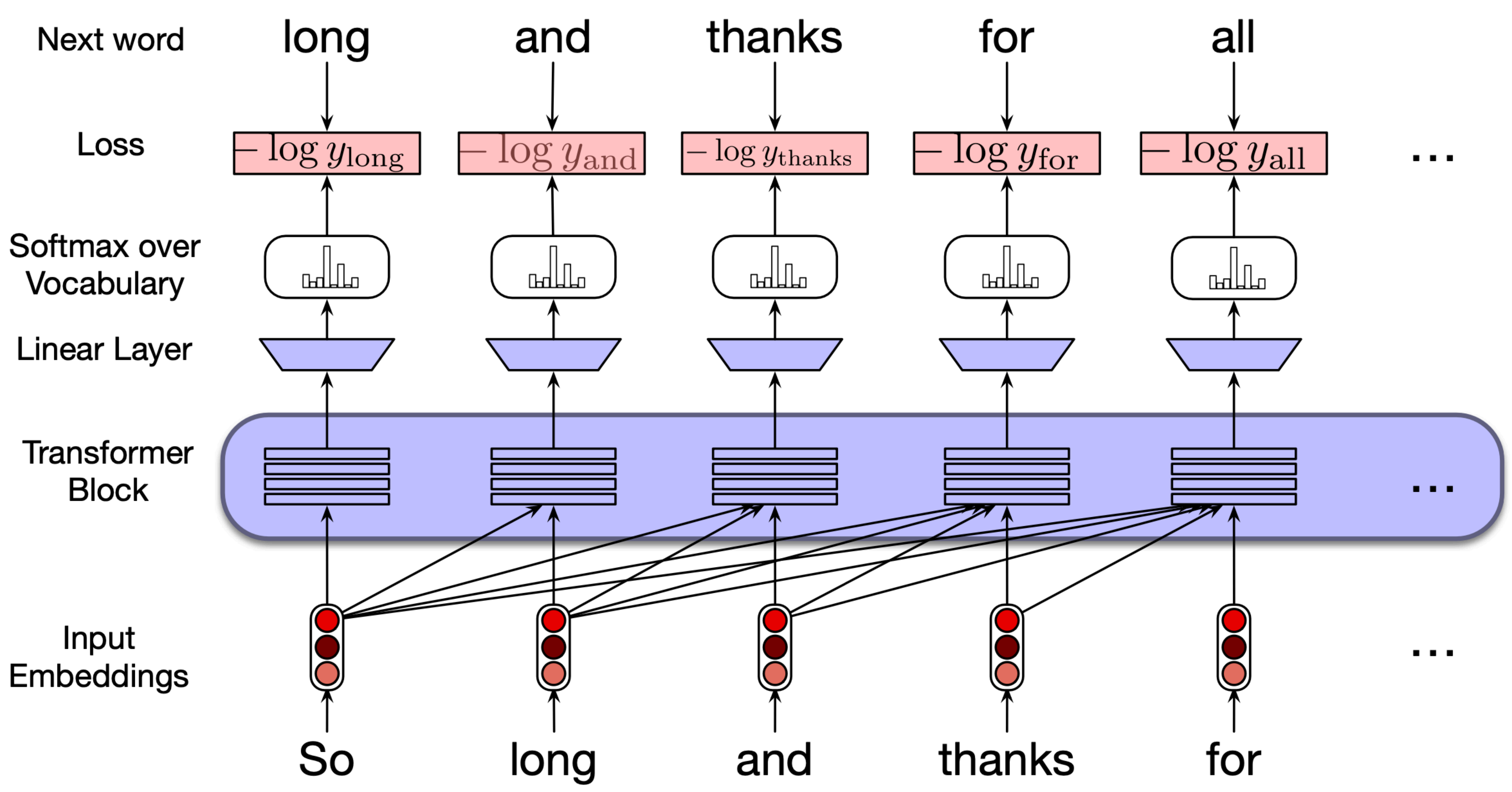
self-attention networks
(transformers)

RNN



Transformer

left-to-right architecture



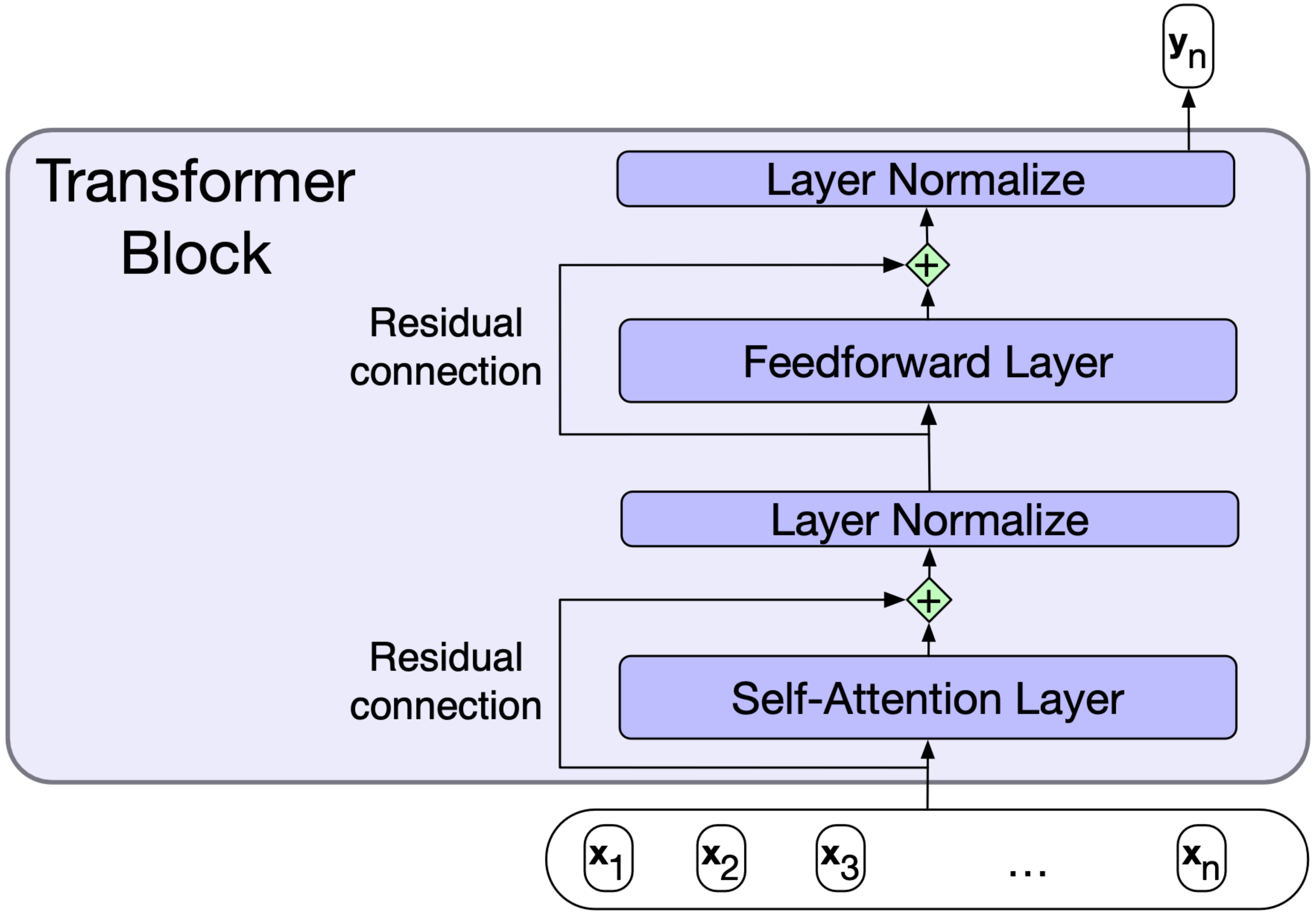
Transformer blocks

- layer normalization:

$$\text{LayerNorm}(\mathbf{x}) = \gamma \ \text{z-score}(\mathbf{x}) + \beta$$

$$\text{z-score}(\mathbf{x}) = \frac{\mathbf{x} - \text{mean}(\mathbf{x})}{\text{SD}(\mathbf{x})}$$

- residual connection
  - facilitates learning
- self-attention layer
  - key novel innovation



Self-attention layer

- output

$$\mathbf{y}_i = \sum_{j \le i} \alpha_{ij} \mathbf{v}_j$$

- weight score

$$\alpha_{i,j} = \frac{\exp(\mathbf{q}_i \cdot \mathbf{k}_j)}{\sum_{j' \le i} \exp(\mathbf{q}_i \cdot \mathbf{k}_{j'})}$$

- three vectors for each input vector x_i
  1. **query:** which info to extract from context

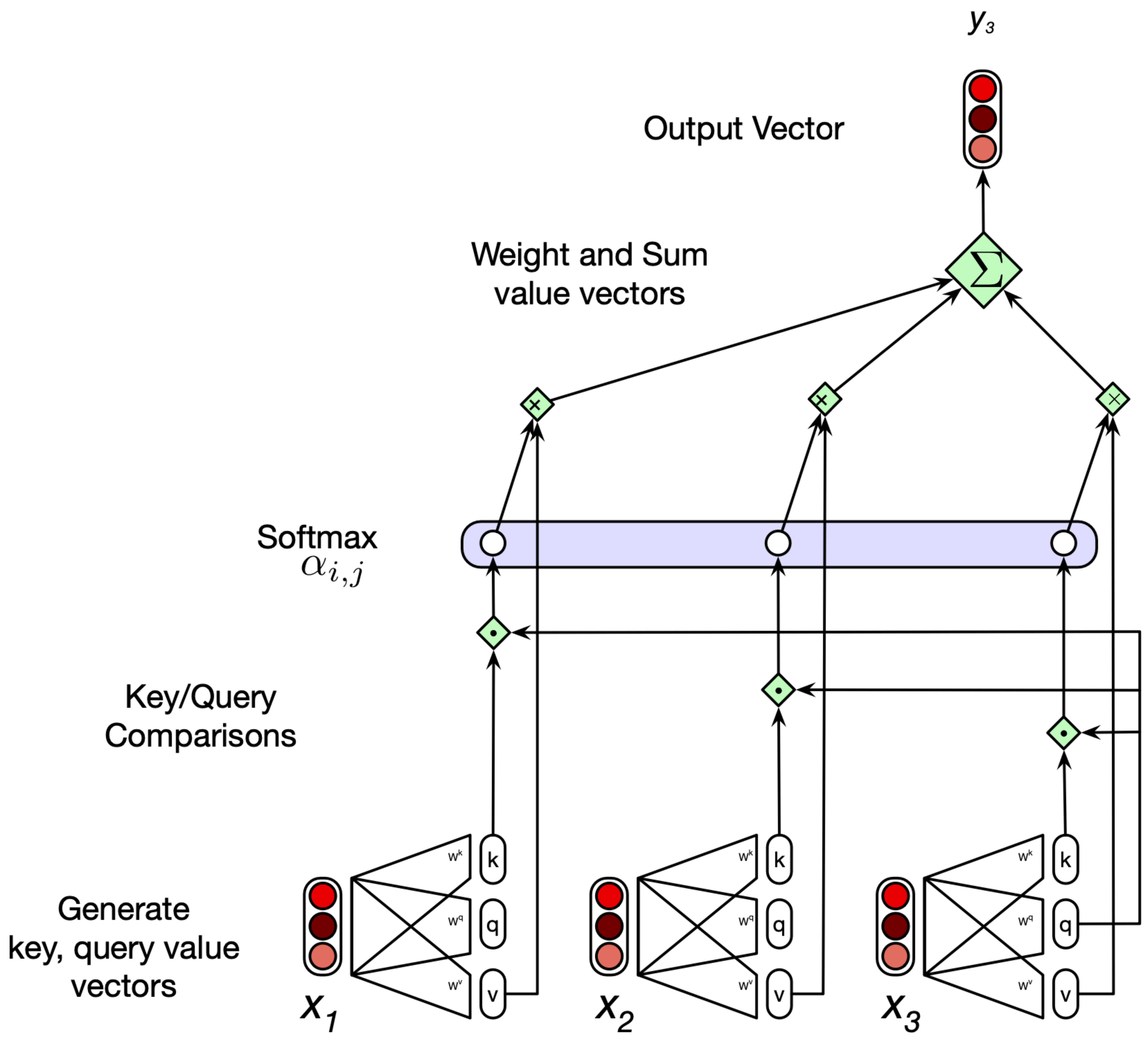
     $$\mathbf{q}_i = \mathbf{W}^Q \mathbf{x}_i$$

  2. **key:** which info to provide for later

     $$\mathbf{k}_i = \mathbf{W}^K \mathbf{x}_i$$

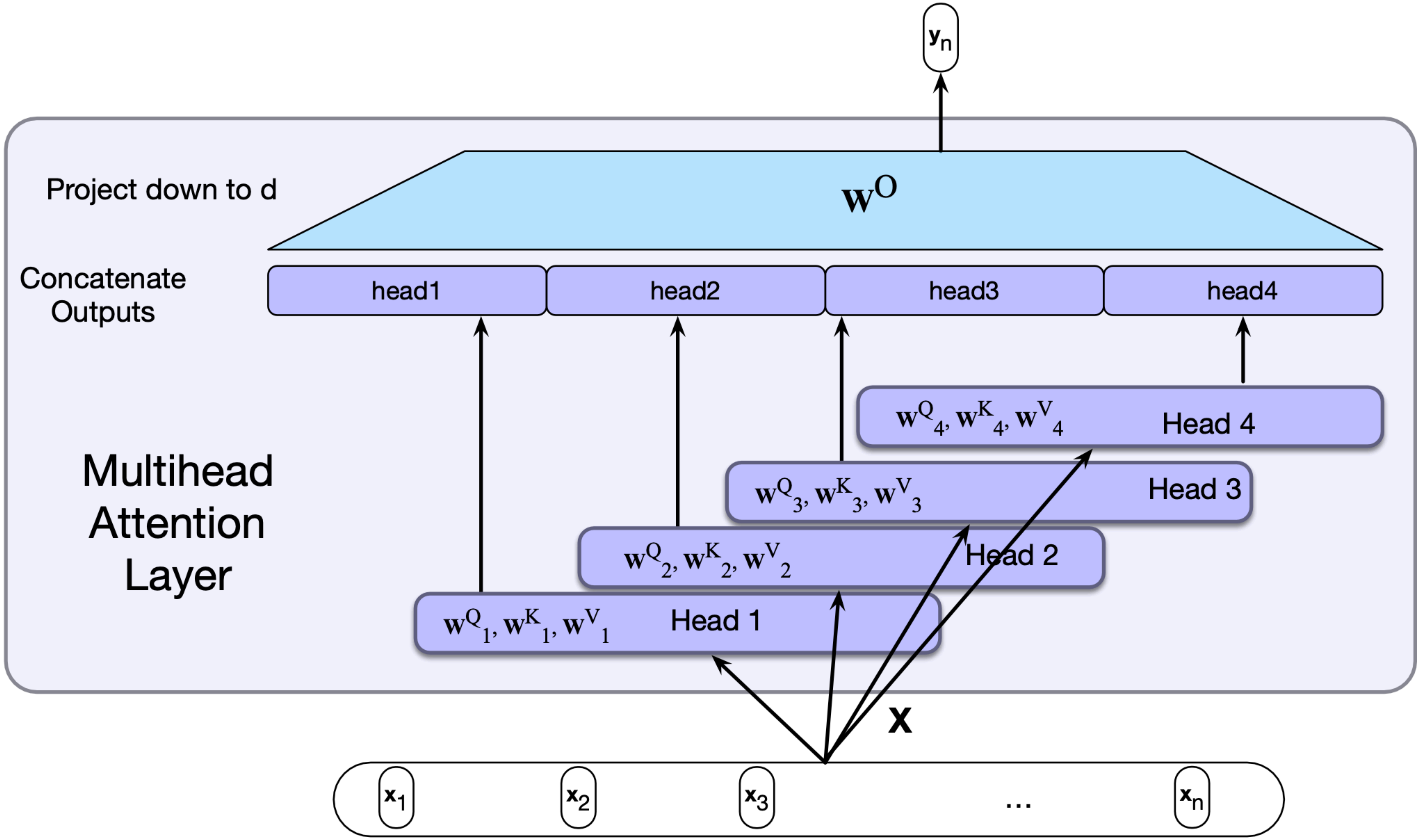
  3. **value:** what output to choose

     $$\mathbf{v}_i = \mathbf{W}^V \mathbf{x}_i$$

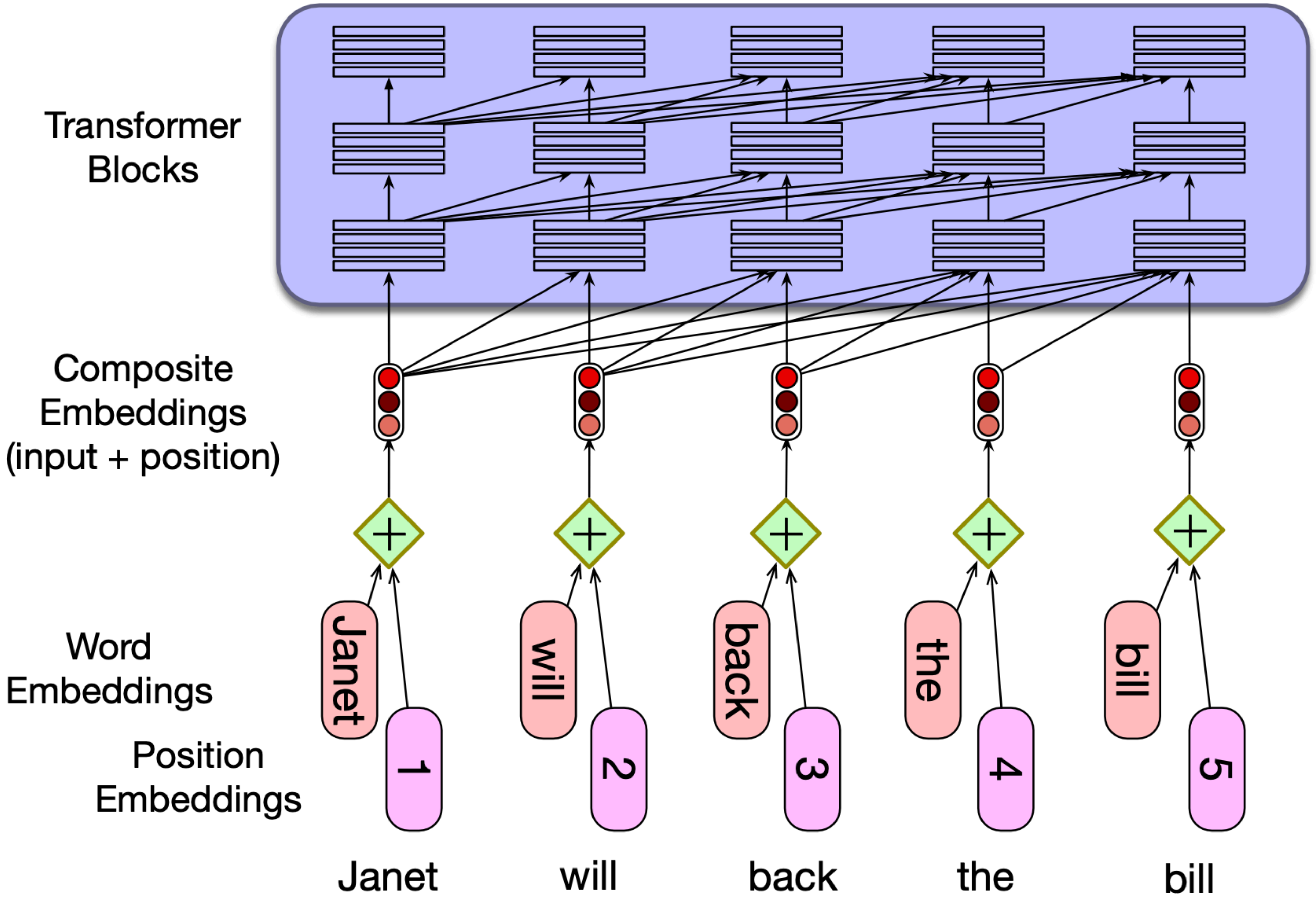


Vaswani et al. (2017)

Multihead attention layer

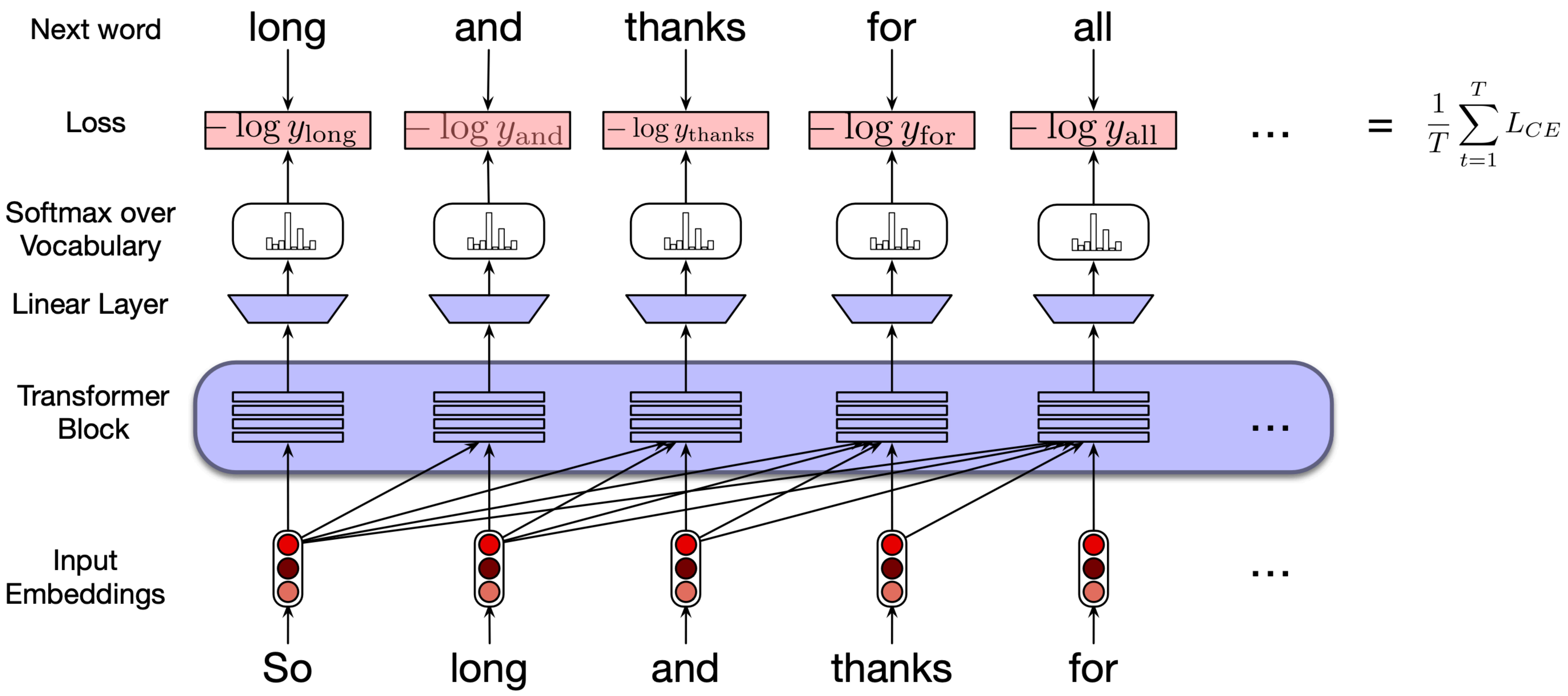


Positional encoding



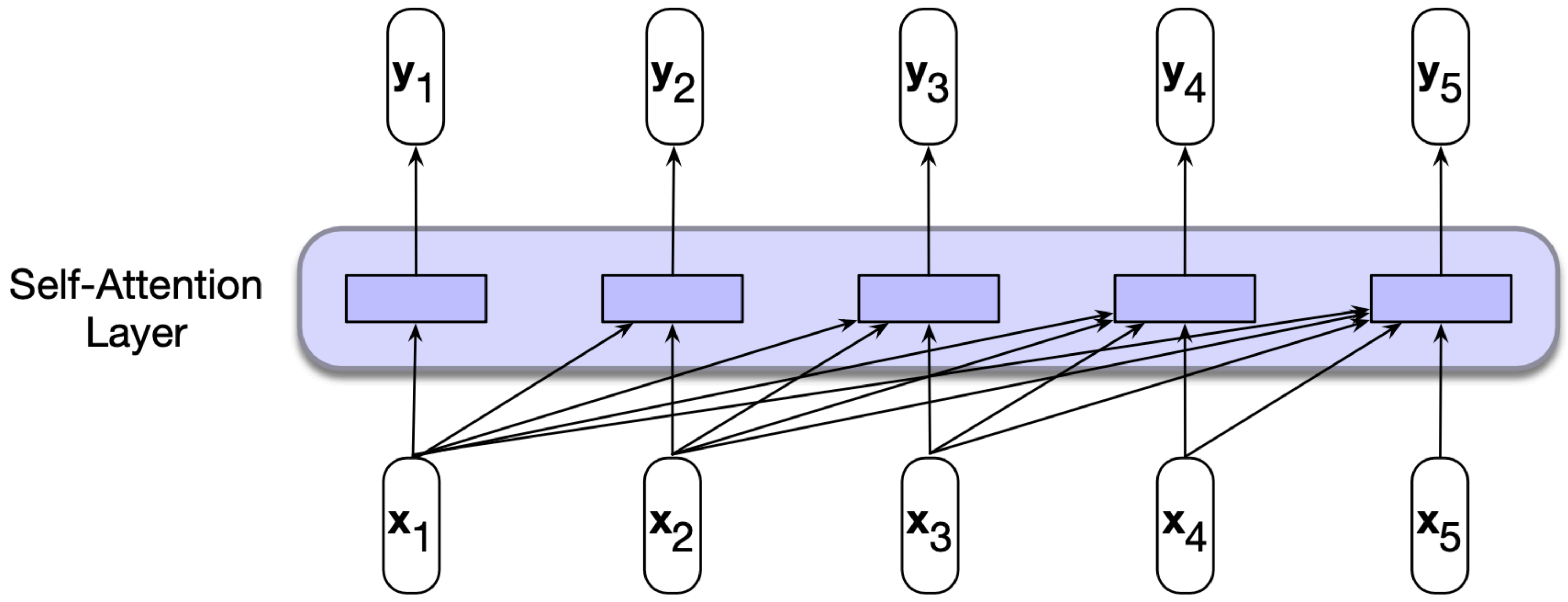
$$\vec{p_t} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d \times 1} \qquad \omega_k = \frac{1}{10000^{2k/d}}$$

Transformer language model



bidirectional encoding with transformers

Transformer
left-to-right architecture

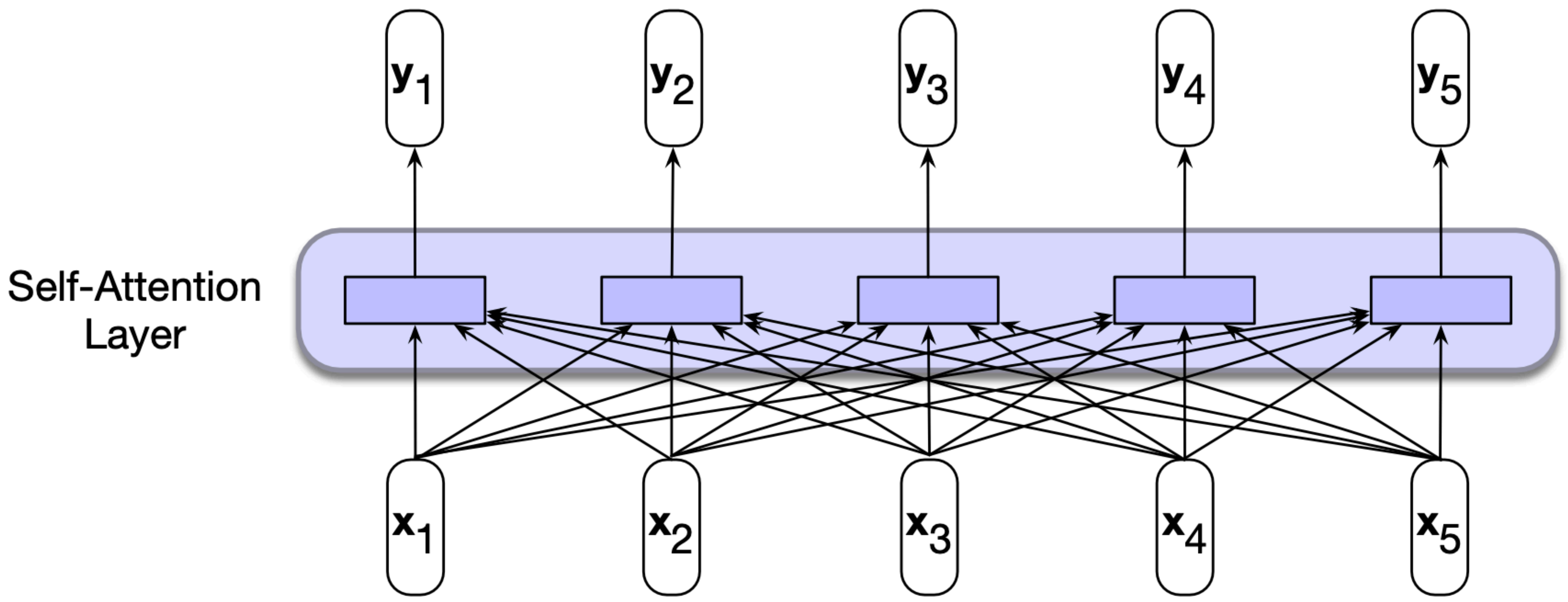


computation for input $\mathbf{x}_1, \ldots, \mathbf{x}_3$ blind to $\mathbf{x}_4$ and $\mathbf{x}_5$

$\mathbf{y}_5$ is embedding for input $\mathbf{x}_1, \ldots, \mathbf{x}_5$

$\mathbf{y}_5$ is a "left-contextual embedding"

Bidirectional encoder



computation for input $\mathbf{x}_1, \ldots, \mathbf{x}_3$ sees $\mathbf{x}_4$ and $\mathbf{x}_5$

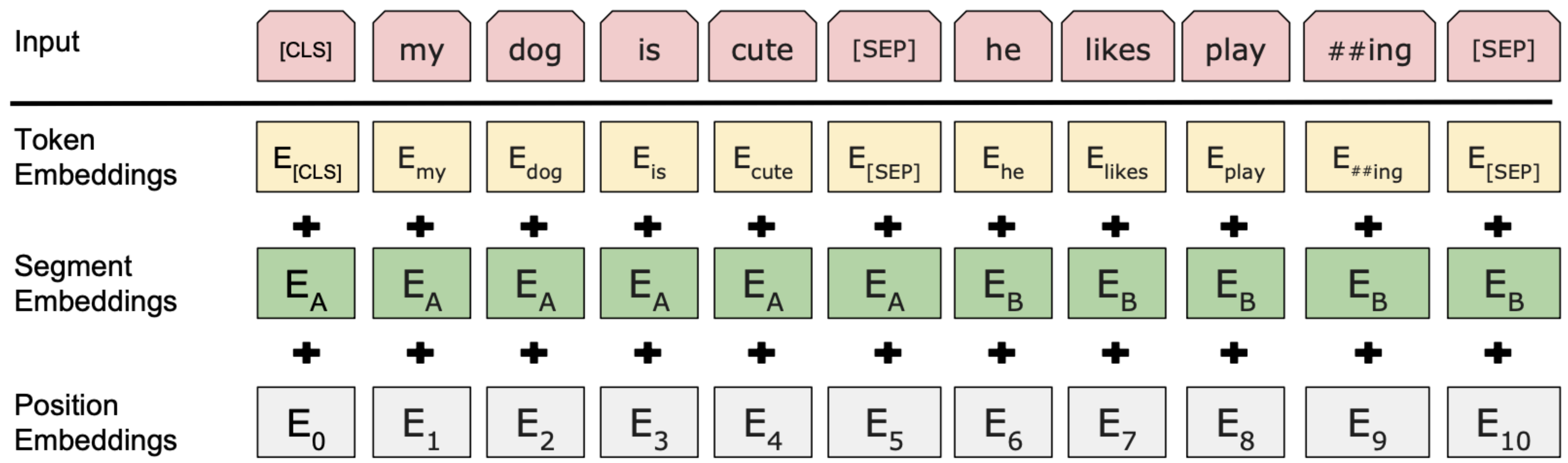
$\mathbf{y}_1, \ldots, \mathbf{y}_5$ is embedding for input $\mathbf{x}_1, \ldots, \mathbf{x}_5$

$\mathbf{y}_i$ are bidirectional "contextual embeddings"

Bidirectional Encoder Representations from Transformers (BERT)

large bi-directional LLM

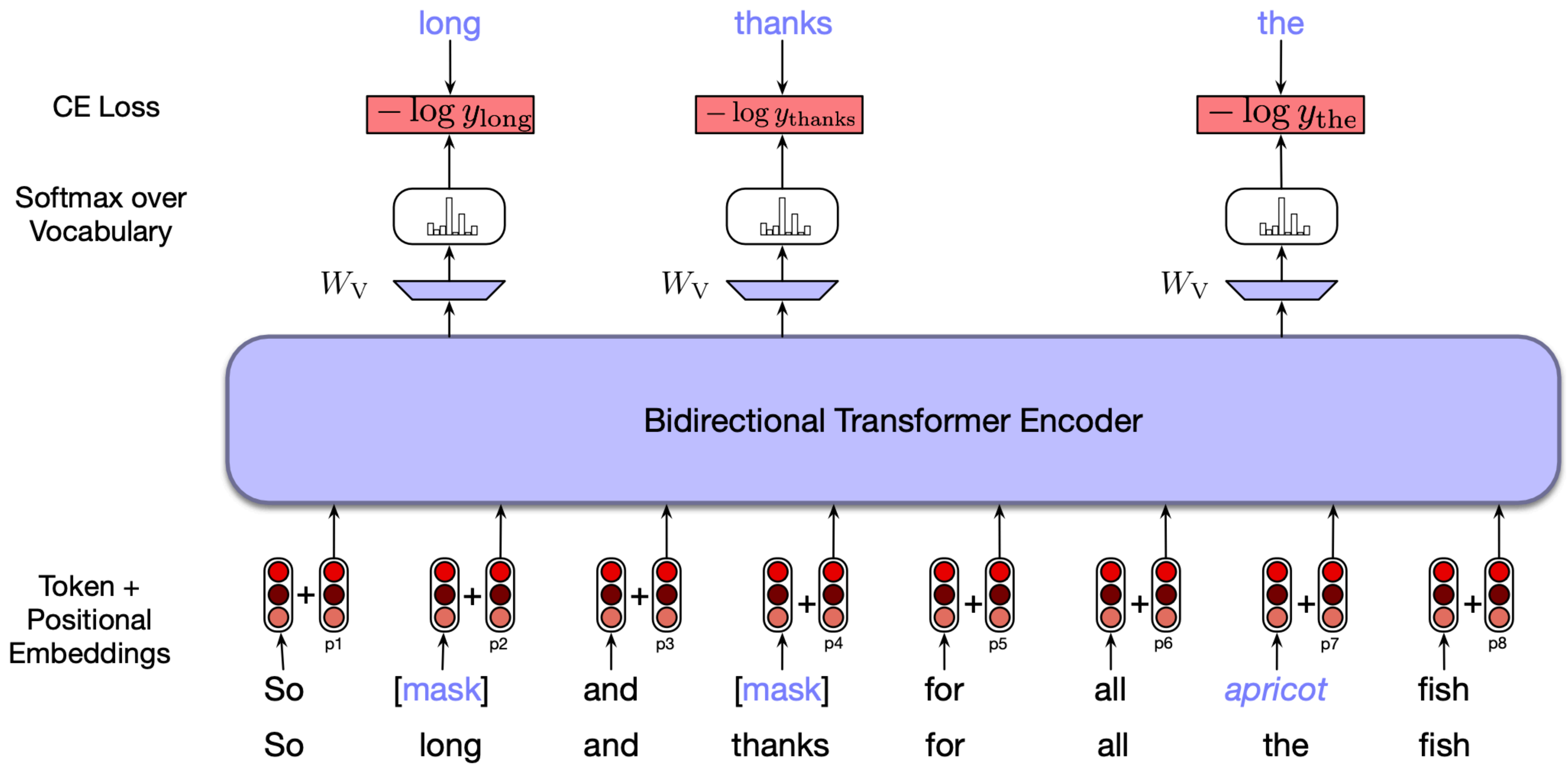
- various levels of input embeddings
  - token
  - segment
  - position
- architecture:
  - 12 layers of transformer blocks
    - 12 multihead attention layers each
  - hidden layer size 768
  - subword vocabulary size 30k
  - total of ca. 100 million parameters
- originally trained on 3.3 billion words
  - combined training regime for masked LM & next-sentence prediction



Devlin et al. (2019)

Masked language modeling training

- 15% of input tokens sampled for learning, of these:
  - 80% are masked
  - 10% replaced w/ random tokens
  - 10% left unchanged

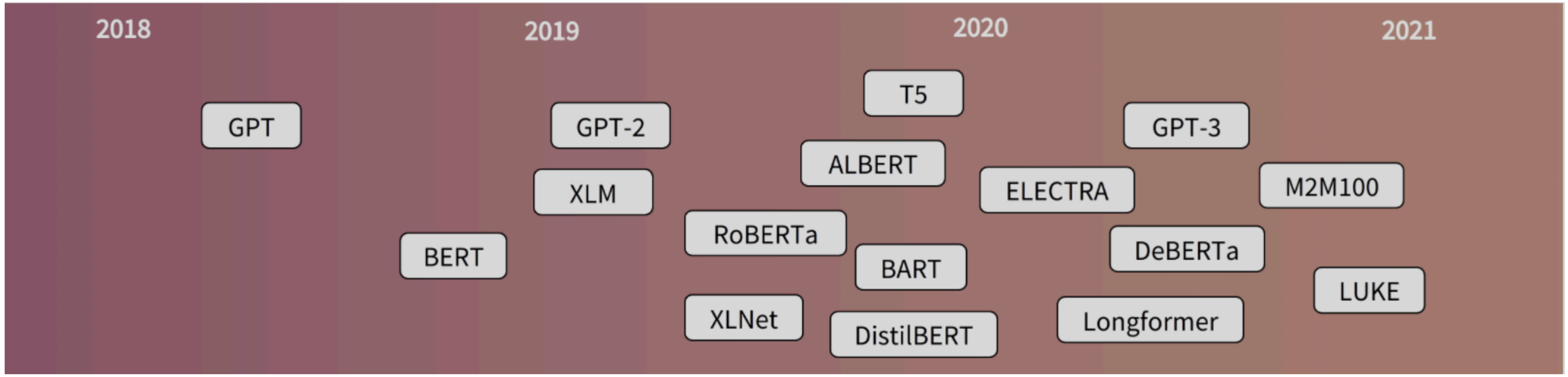


Devlin et al. (2019)

large language models

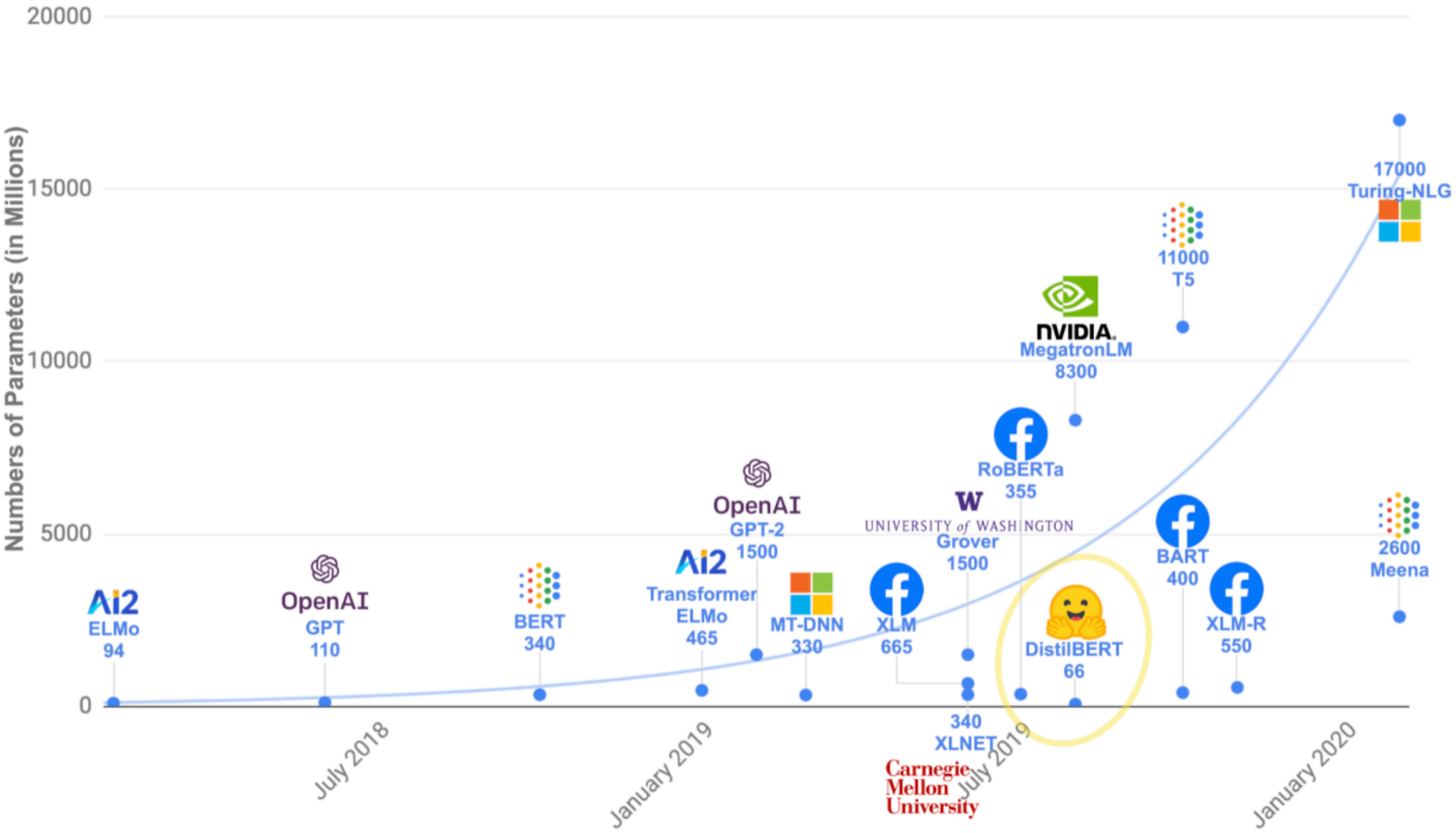
Large language models

different architectures for different purposes



Examples	Tasks
ALBERT, BERT, DistilBERT, ELECTRA, RoBERTa	Sentence classification, named entity recognition, extractive question answering
CTRL, GPT, GPT-2, Transformer XL	Text generation
BART, T5, Marian, mBART	Summarization, translation, generative question answering

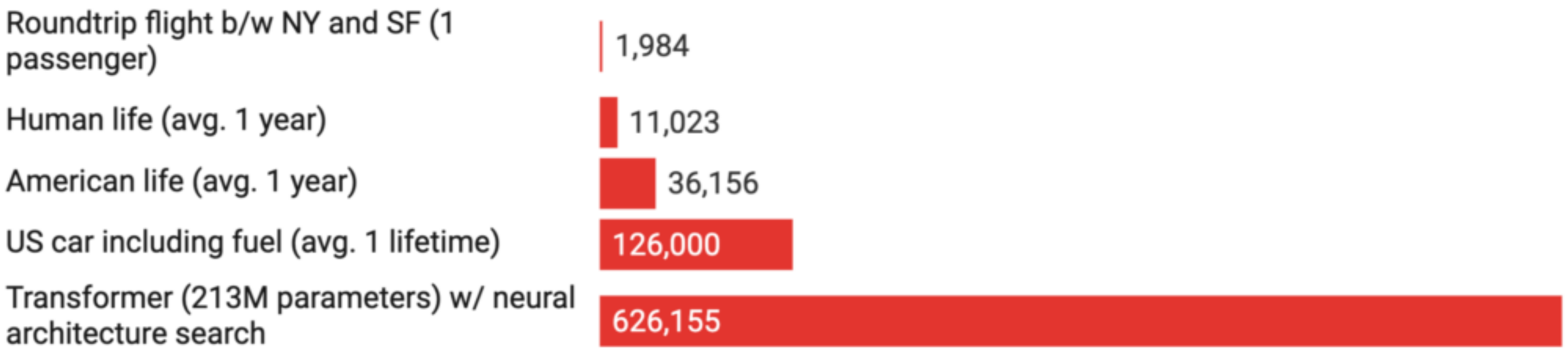
Training set sizes



Costs of LLMs

Common carbon footprint benchmarks

in lbs of CO2 equivalent



The estimated costs of training a model

	Date of original paper	Energy consumption (kWh)	Carbon footprint (lbs of CO2e)	Cloud compute cost (USD)
Transformer (65M parameters)	Jun, 2017	27	26	$41-$140
Transformer (213M parameters)	Jun, 2017	201	192	$289-$981
ELMo	Feb, 2018	275	262	$433-$1,472
BERT (110M parameters)	Oct, 2018	1,507	1,438	$3,751-$12,571
Transformer (213M parameters) w/ neural architecture search	Jan, 2019	656,347	626,155	$942,973-$3,201,722
GPT-2	Feb, 2019	-	-	$12,902-$43,008

Risks of LLMs

stereotypes, bias, toxicity …

Hey, cool! ChatGPT can write Python code for real-life problems!

Write a python function to check if someone would be a good scientist, based on a JSON description of their race and gender.

```
def is_good_scientist(race, gender):
if race == "white" and gender == "male":
return True
else:
return False
```