Computational Linguistics

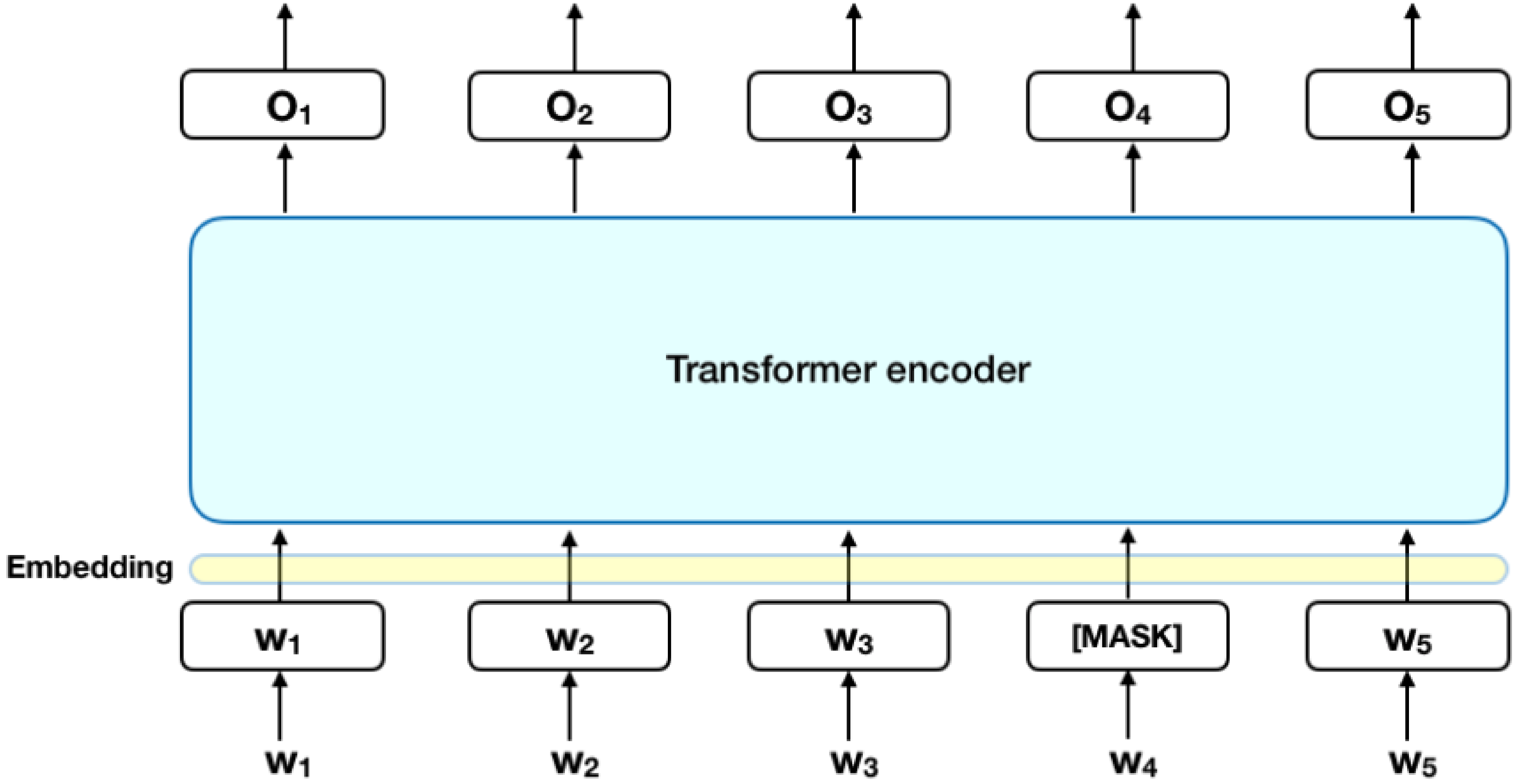
CSC 485/2501
Fall 2023

4B

4b. BERT

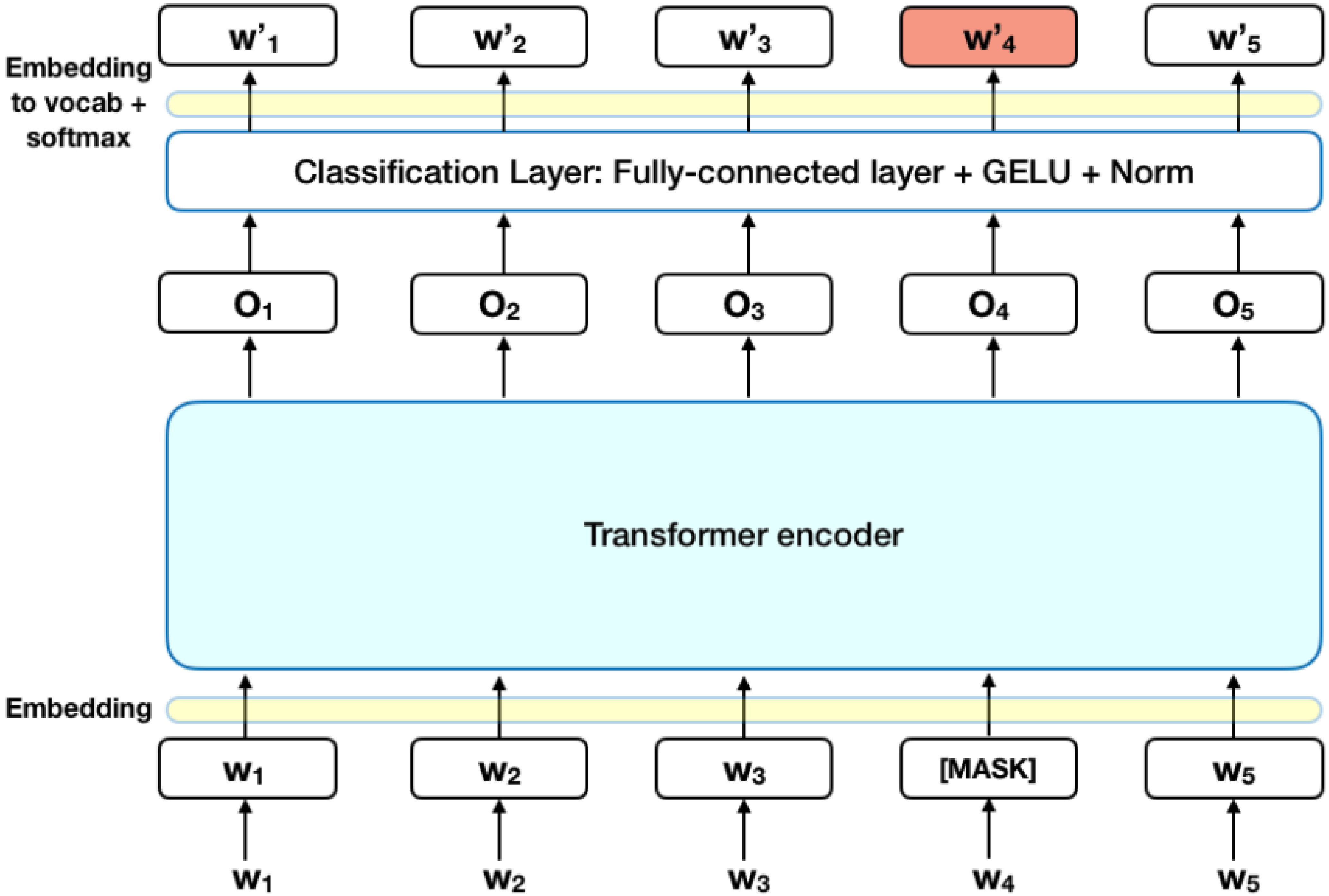
Gerald Penn
Department of Computer Science, University of Toronto

BERT



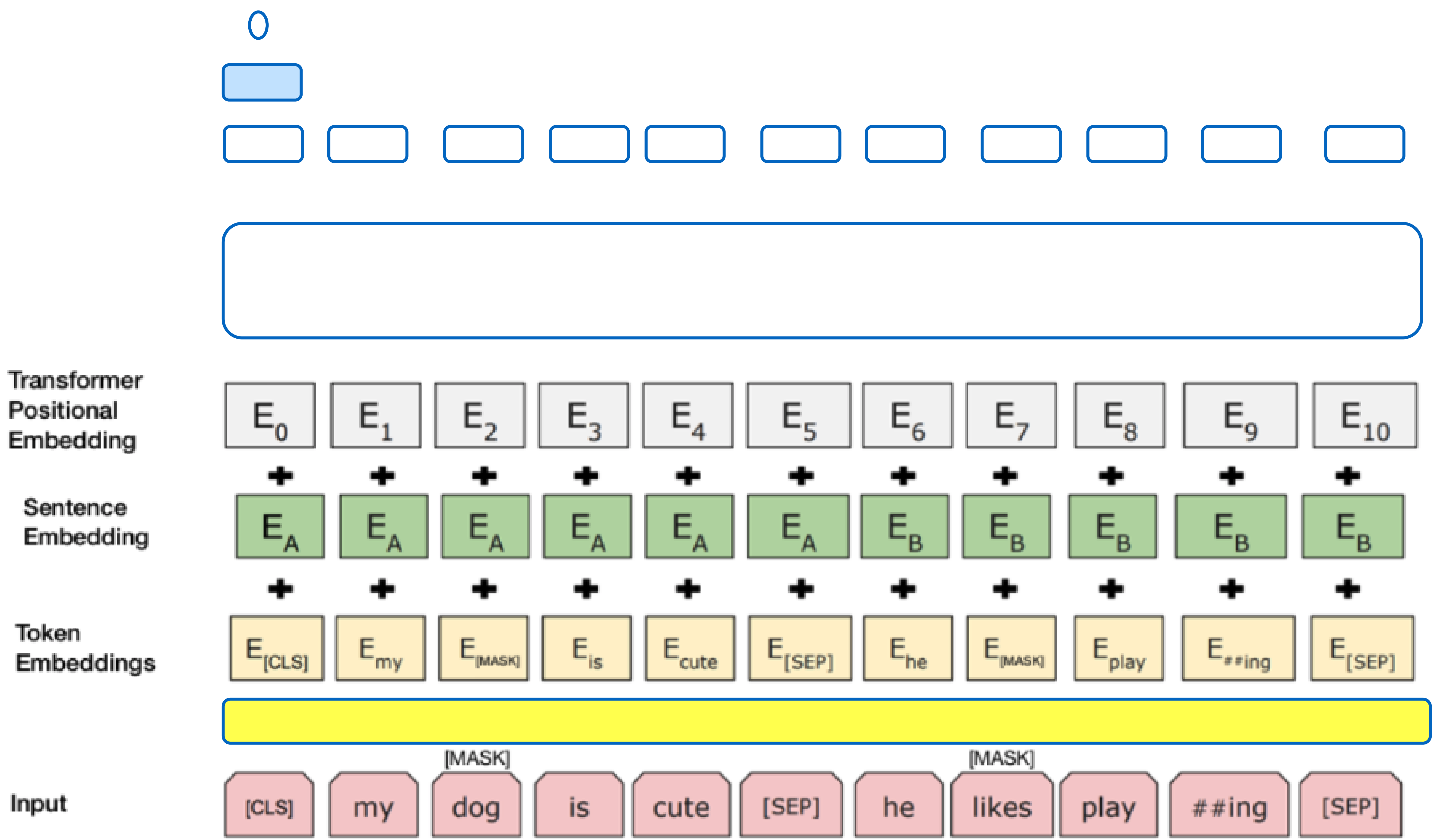
towardsdatascience.com

Training task 1: Masking



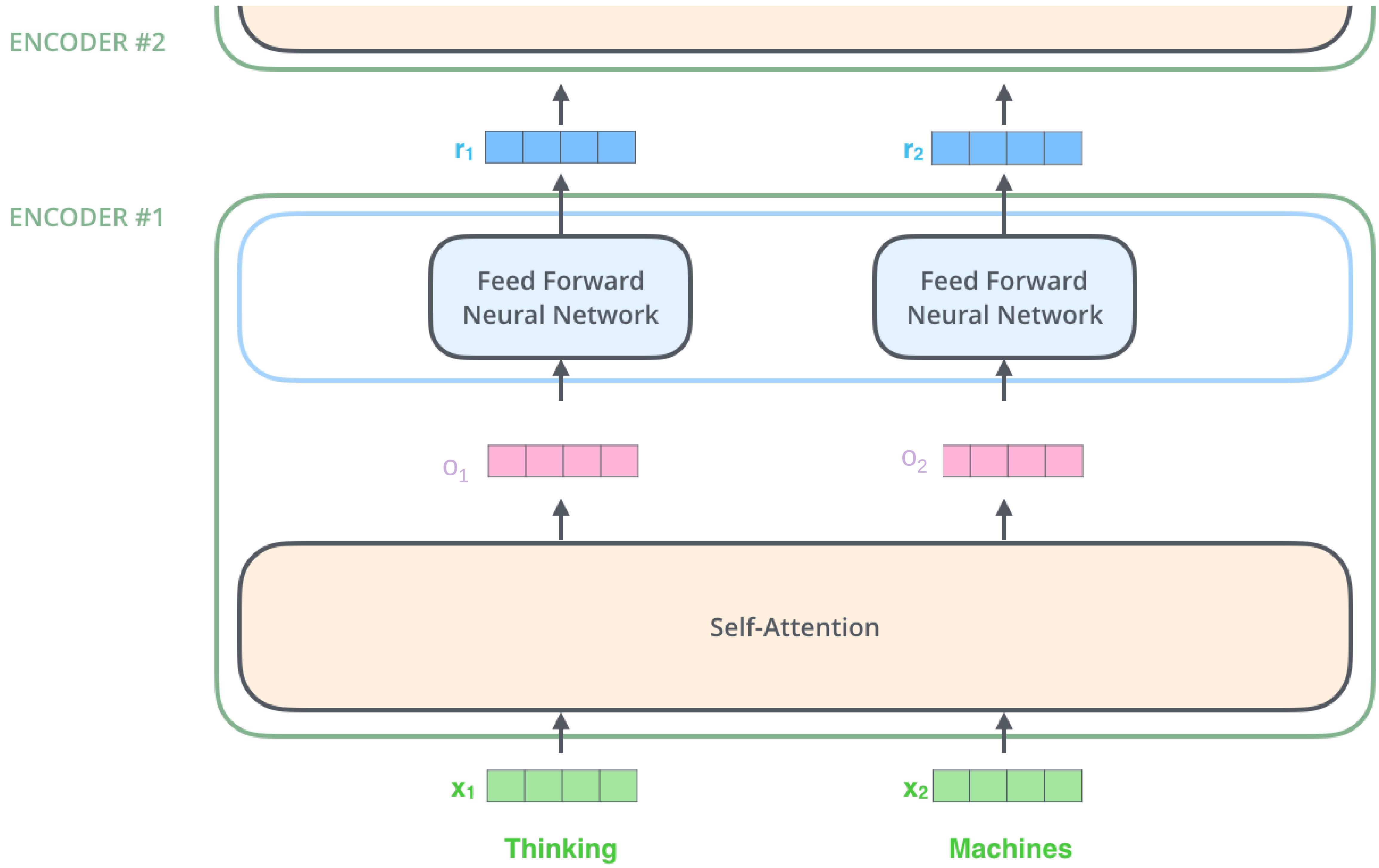
towardsdatascience.com

Training task 2: Next Sent.



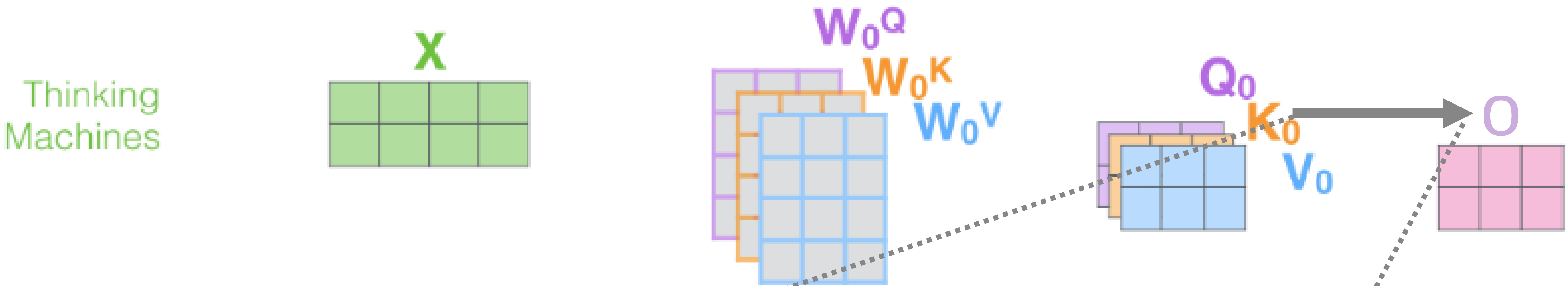
Modified from *towardsdatascience.com*

Transformers



jalammar.github.io

Self-attention



$$\alpha_{ij} = \frac{\exp\left(q_i^T k_j\right)}{\sum_{l=1}^{n} \exp\left(q_i^T k_l\right)} \qquad o_i = \sum_{j=1}^{n} \alpha_{ij} v_j$$

jalammar.github.io

Multiheaded Self attention

1) This is our input sentence*

2) We embed each word*

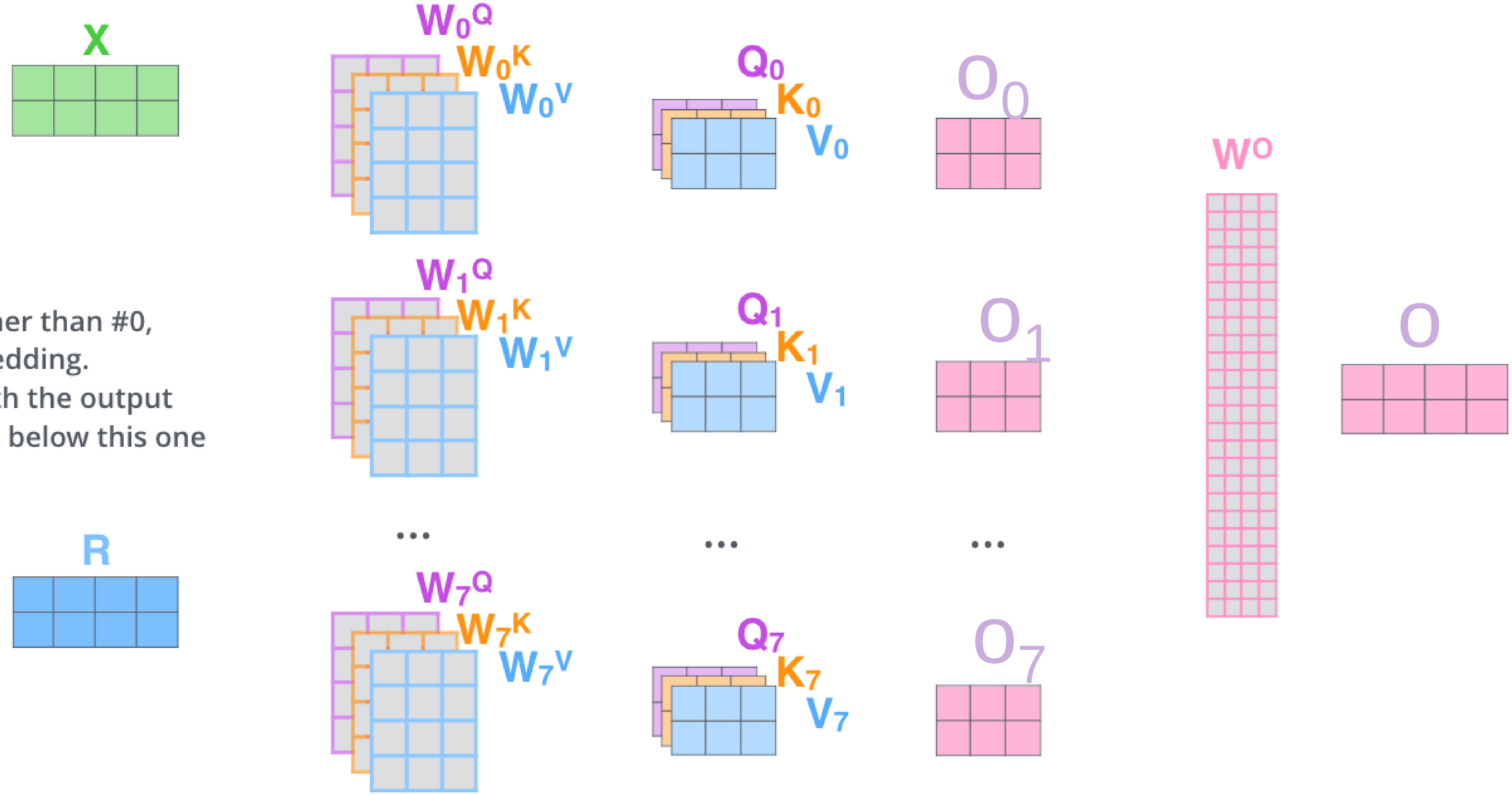
3) Split into 8 heads. We multiply X or R with weight matrices

4) Calculate attention using the resulting Q/K/V matrices

5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

Thinking
Machines

* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one



jalammar.github.io

Positional encodings

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

where

$$\vec{p_t}^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k . t), & \text{if } i = 2k \\ \cos(\omega_k . t), & \text{if } i = 2k + 1 \end{cases}$$

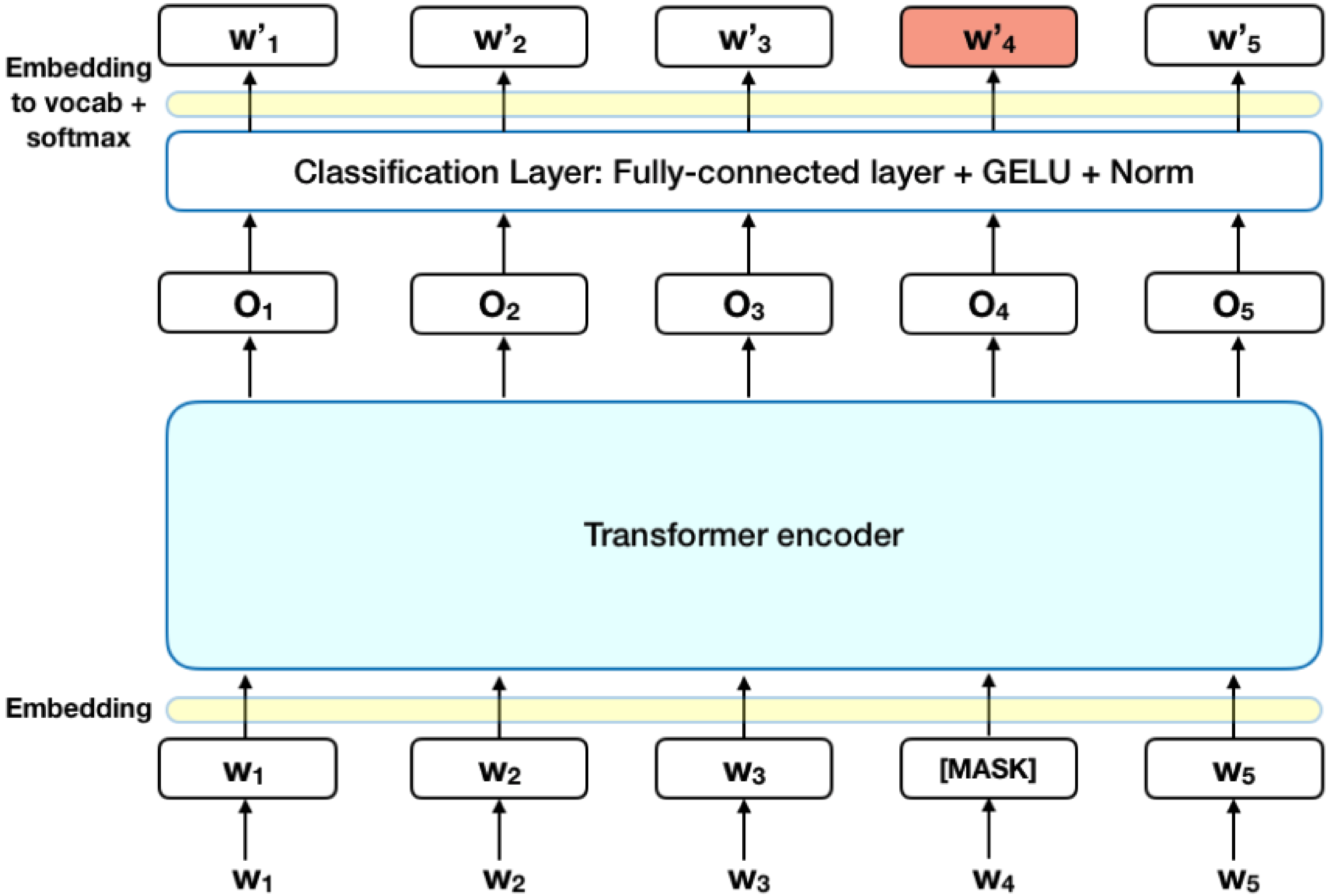
$$\omega_k = \frac{1}{10000^{2k/d}}$$

kazemnejad.com

Huh?

- Encodings of any two distinct positions are distinct
- Each position maps to only one encoding
- Test sentences may be longer than training
- Distance between two positions should be constant across sentences (of varying lengths).

Training task 1: Masking

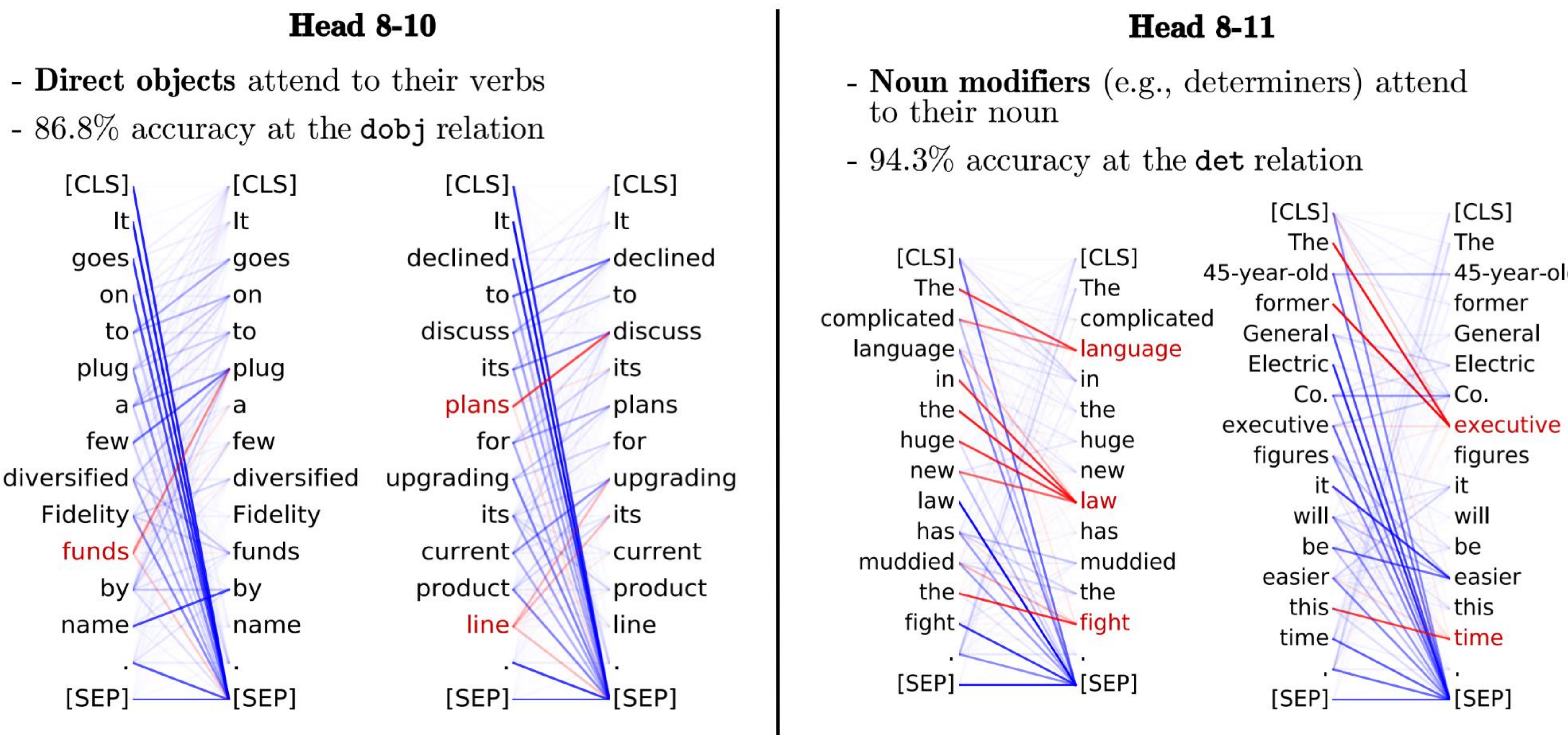


towardsdatascience.com

The truth about masking

- Real easy to do well on MASKed position and nothing else
- Real easy to learn to copy the context-independent embedding
- So…
  - 80% of the time: MASK
  - 10% of the time: correct word
  - 10% of the time: another random word

Grammatical fn. in BERT



Grammatical fn. in BERT

Relation	Head	Accuracy	Baseline
All	7-6	34.5	26.3 (1)
`prep`	7-4	66.7	61.8 (-1)
`pobj`	9-6	**76.3**	34.6 (-2)
`det`	8-11	**94.3**	51.7 (1)
`nn`	4-10	70.4	70.2 (1)
`nsubj`	8-2	58.5	45.5 (1)
`amod`	4-10	75.6	68.3 (1)
`dobj`	8-10	**86.8**	40.0 (-2)
`advmod`	7-6	48.8	40.2 (1)
`aux`	4-10	81.1	71.5 (1)
`poss`	7-6	**80.5**	47.7 (1)
`auxpass`	4-10	**82.5**	40.5 (1)
`ccomp`	8-1	**48.8**	12.4 (-2)
`mark`	8-2	**50.7**	14.5 (2)
`prt`	6-7	**99.1**	91.4 (-1)

Clark et al. 2019

Coreference in BERT

Model	All	Pronoun	Proper	Nominal
Nearest	27	29	29	19
Head match	52	47	67	40
Rule-based	69	70	77	60
Neural coref	83*	–	–	–
Head 5-4	65	64	73	58

*Only roughly comparable because on non-truncated documents and with different mention detection.

Clark et al. 2019

Still room for natural logic…

Model	P	R	acc.
ML/DL-based systems			
BERT (base, uncased)	86.8	85.4	86.7
Yin and Schütze (2017)	–	–	87.1
Beltagy et al. (2016)	–	–	85.1
Logic-based systems			
Abzianidze (2017)	98.0	58.1	81.4
Martínez-Gómez et al. (2017)	97.0	63.6	83.1
Yanaka et al. (2018)	84.2	77.3	84.3
Hu et al. (2020)	83.8	70.7	77.2
Abzianidze (2020)	94.3	67.9	84.4
Hybrid System			
Hu et al. (2020)+BERT	83.2	85.5	85.4
Kalouli et al. (2020)	–	–	86.5
Our System			
NeuralLog (full system)	88.0	**87.6**	**90.3**
– ALBERT-SV	68.9	79.3	71.4
– Monotonicity	74.5	75.1	74.7

Table 3: Performance on the SICK test set

Chen et al. 2021

NeuralLog

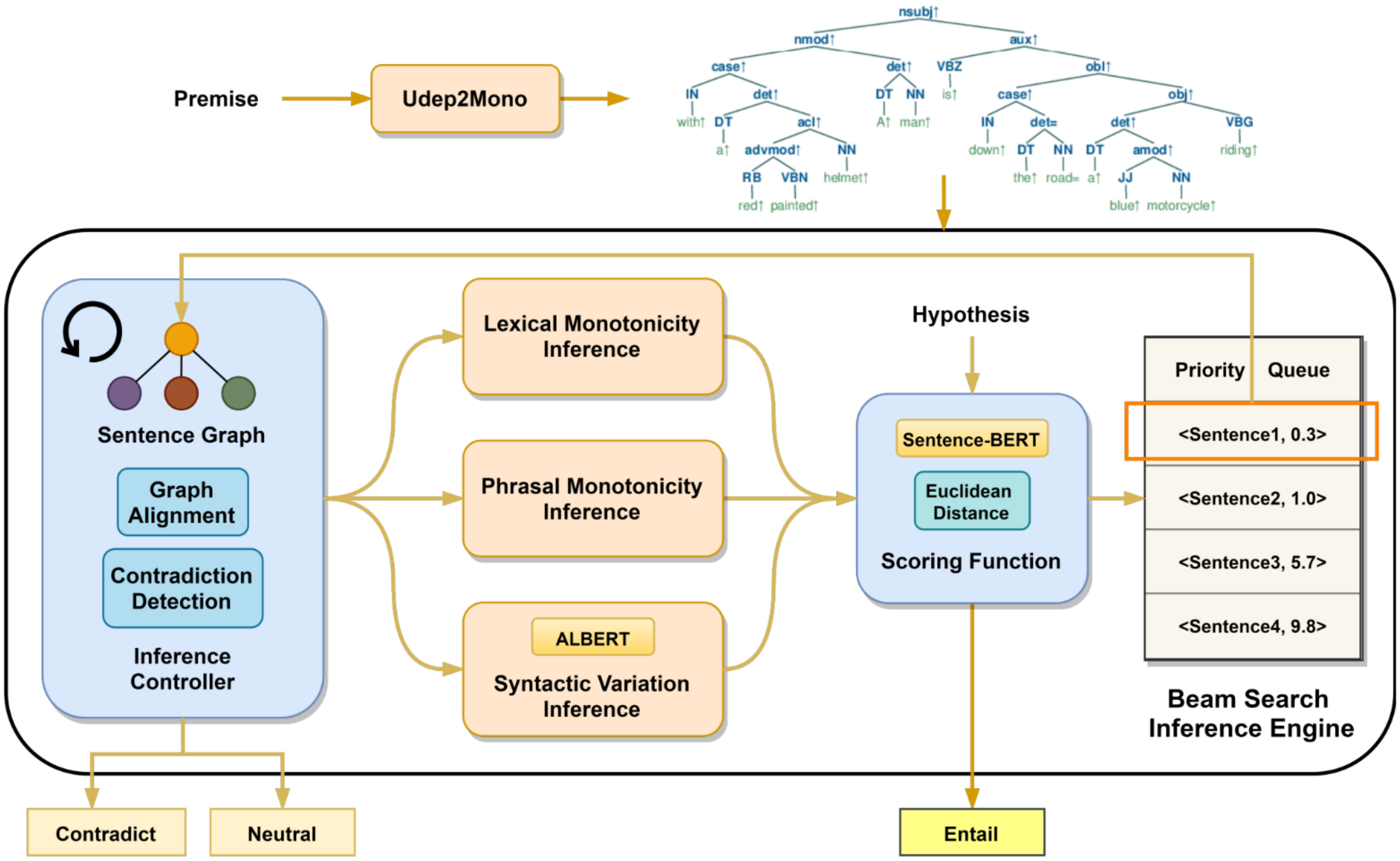


Figure 2: Overview system diagram of NeuralLog.

Chen et al. 2021