wbs Warwick Business School

Data Science & Generative AI

Dr Michael Mortenson

Associate Professor (Reader)
michael.mortenson@wbs.ac.uk

Session 8: Transformers – From T to G via P

Transformers from T to G

Generative

Pretrained

Transformers

Self-Attention by Metaphor

Ram Gopal 78

7 miles away

Professor and part-time dancer.

Hobbies: Chicago Drill, Travelling, MMA.

Looking for: Love in all the wrong places.

Moris Strub 41

5 miles away

Academic specialising in FinTech and maths.

Hobbies: I like maths.

Looking for: Someone who likes maths.

MJ Mortenson 44

0 miles away

AI guy. Has an awesome beard.

Hobbies: Deadlifting, AI, teaching, coffee.

Looking for: Just browsing.

Ephie Wang 21

2 miles away

Academic and actress.

Hobbies: Watching football.

Looking for: A guy in finance. 6’5”, trust fund, blue eyes.

Self-Attention by Metaphor

Ram Gopal 78

7 miles away

Professor and part-time dancer.

Hobbies: Chicago Drill, Travelling, MMA.

Looking for: Love in all the wrong places.

Moris Strub 41

5 miles away

Academic specialising in FinTech and maths.

Hobbies: I like maths.

Looking for: Someone who likes maths.

MJ Mortenson 44

0 miles away

AI guy. Has an awesome beard.

Hobbies: Deadlifting, AI, teaching, coffee.

Looking for: Just browsing.

Ephie Wang 21

2 miles away

Academic and actress.

Hobbies: Watching football.

Looking for: A guy in finance. 6'5", trust fund, blue eyes.

Self-Attention by Metaphor

Ram Gopal 78

7 miles away

Professor and part-time dancer.

Hobbies: Chicago Drill, Travelling, MMA.

Looking for: Love in all the wrong places.

Moris Strub 41

5 miles away

Academic specialising in FinTech and maths.

Hobbies: I like maths

Looking for: Someone who likes maths.

MJ Mortenson 44

0 miles away

AI guy. Has an awesome beard.

Hobbies: Deadlifting, AI, teaching, coffee.

Looking for: Just browsing.

Ephie Wang 21

2 miles away

Academic and actress.

Hobbies: Watching football.

Looking for: A guy in finance. 6'5", trust fund, blue eyes.

Self-Attention by Metaphor

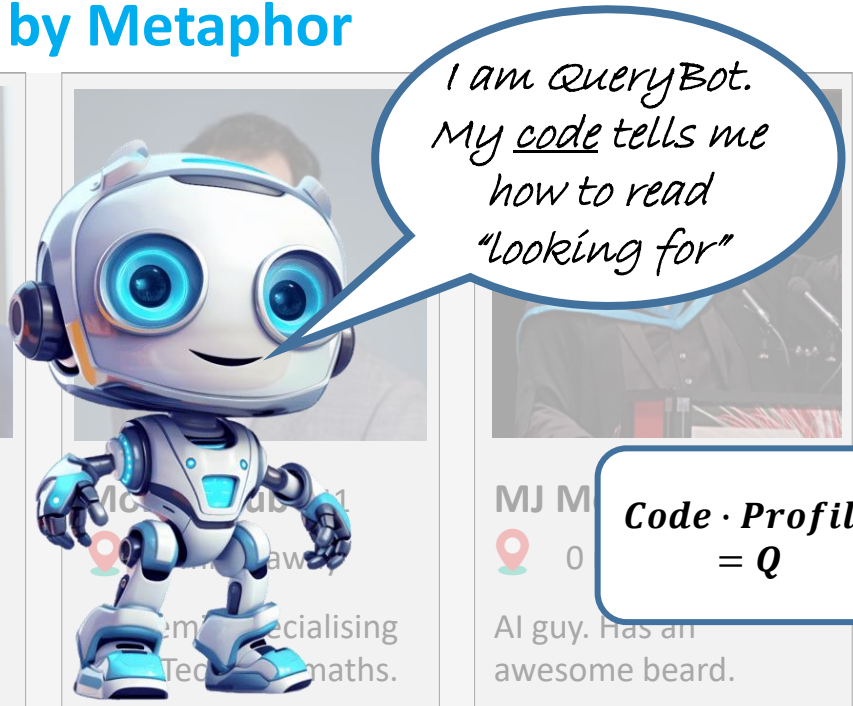
Ram Gopal 78

7 miles away

Professor and part-time dancer.

Hobbies: Chicago Drill, Travelling, MMA.

Looking for: Love in all the wrong places.



Hobbies: I like maths.

Looking for: Someone who likes maths.

AI guy. Has an awesome beard.

Hobbies: Deadlifting, AI, teaching, coffee.

Looking for: Just browsing.

$$Code \cdot Profile = Q$$

Ephie Wang 21

2 miles away

Academic and actress.

Hobbies: Watching football.

Looking for: A guy in finance. 6'5", trust fund, blue eyes.

Self-Attention by Metaphor

Ram Gopal 78

7 miles away

Professor and part-time dancer.

Hobbies: Chicago Drill, Travelling, MMA.

Looking for: Love in all the wrong places.

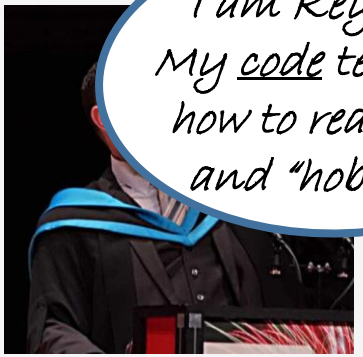
Moris Strub 41

5 miles away

Academic specialising in FinTech and maths.

Hobbies: I like maths

Looking for: Someone who likes maths.



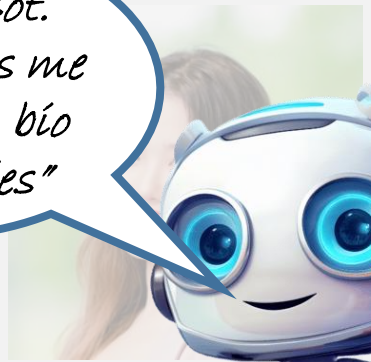
MJ Morter

0 miles

AI guy. Has an awesome beard.

Hobbies: Deadlifting, AI, teaching, coffee.

Looking for: Just browsing.



e War

2 mi aw

demic and ess.

Hobbies: Wa football.

Looking for: A guy in finance. 6'5", trust fund, blue eyes.

I am KeyBot.
My code tells me how to read bio and "hobbies"

$$Code \cdot Profile = K$$

Self-Attention by Metaphor

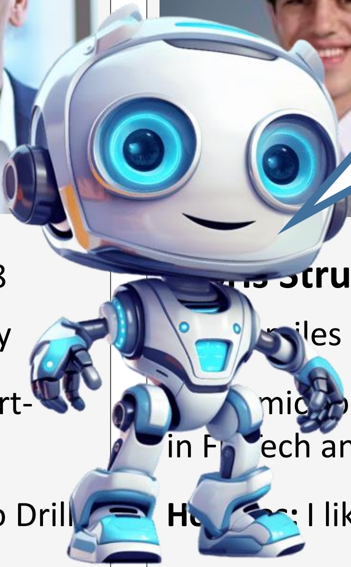
Ram Gopal 78

7 miles away

Professor and part-time dancer.

Hobbies: Chicago Drill, Travelling, MMA.

Looking for: Love in all the wrong places.



I am ScoreBot.
I compare **Q** and **K** to see how similar they are

...is Strub 41

...iles away

...mic ...ecialising in F... ...ech and maths.

H...: I like maths.

Looking for: Someone who likes maths.

$$AttentionScore = Q \cdot K$$

A... awesome beard.

Hobbies: Deadlifting, AI, teaching, coffee.

Looking for: Just browsing.

Ephie Wang 21

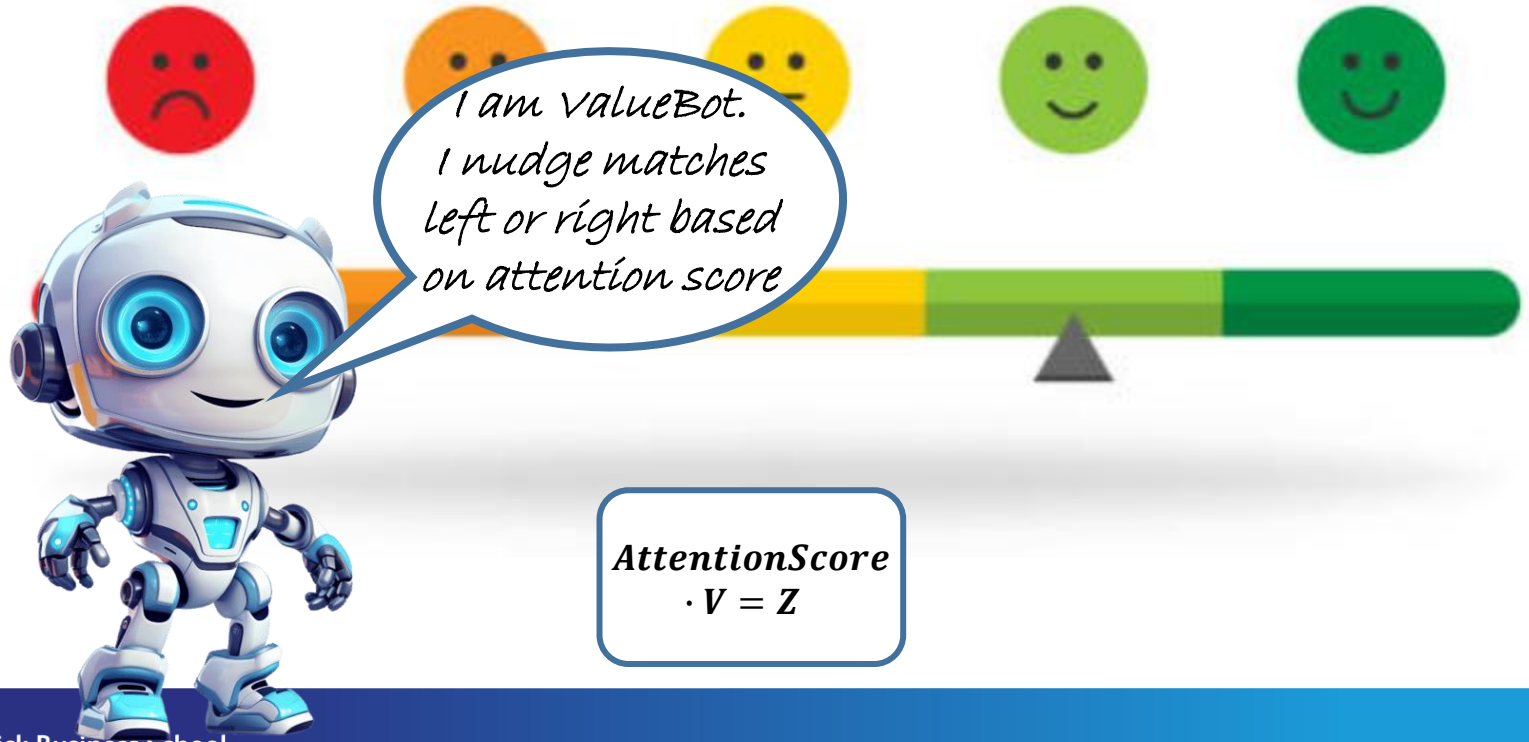
2 miles away

Academic and actress.

Hobbies: Watching football.

Looking for: A guy in finance. 6’5”, trust fund, blue eyes.

Self-Attention by Metaphor



$$AttentionScore \cdot V = Z$$

Self-Attention Layers (by Matrices)

- Like a convolution layers, we scan through the input* each time multiplying the word vector (semantic embedding and positional encoding - x_i) by the query weights (W_Q). This produces an output we call the *query projection* ($Q = XW_Q$) for the given input.
- For every other word (including the word itself) we also multiply those word vectors by the key weights (W_K), creating a set of *key projections* ($K = XW_K$).
- Rather than comparing filter with input, like we would for convolutions, we compare (via dot product similarity) the *query projection* (Q) with every other *key projection* (K). If they are similar, they produce a large number and represent an activation.

*Not literally true (its matrix multiplication so all done at the same time), but methodologically what effectively happens

Self-Attention Layers (by Matrices)

- The calculations on the previous slide gives us an *attention score*, this is effectively a matrix of activations for every word (x_i) we have compared the query (Q) with. To these we apply the softmax algorithm. This exaggerates differences making big activations proportionally bigger, and reducing small activations towards 0.
- We also compute V, the *value projection* for every value in X ($V = XW_v$). By multiplying this with the *attention score*, we get the appropriate nudge to apply to the specific input word vector we have been evaluating (the output of our layer for that input).
- We would now move the scan on to the next input (x_{i+1}) and apply the same process again.*

*Not literally true (its matrix multiplication so all done at the same time), but methodologically what effectively happens

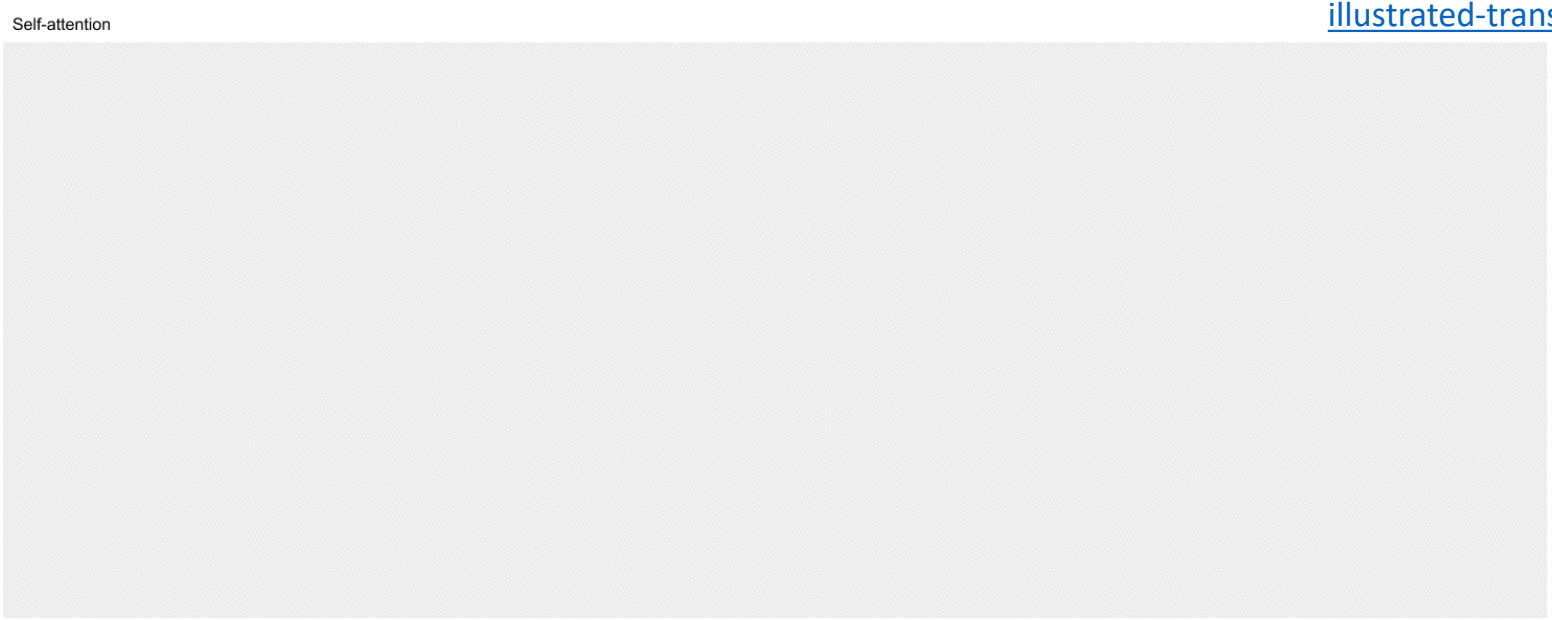
Self-Attention Layers (by Matrices)

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

  where:
  - $Softmax$ changes the output to values that sum to 1. This also exaggerates differences so bigger activations look bigger.
  - QK^T is the query projection (Q) multiplied by the transpose of the key projection (K). The transpose just makes the matrix algebra work (as matrix algebra multiplies rows by columns).
  - $\sqrt{d_k}$ is the square root of the dimensionality of K - the *key projection* (XW_K). This just scales values so that the variance ≈ 1, and means we avoid different variances depending on the size of the input (X).

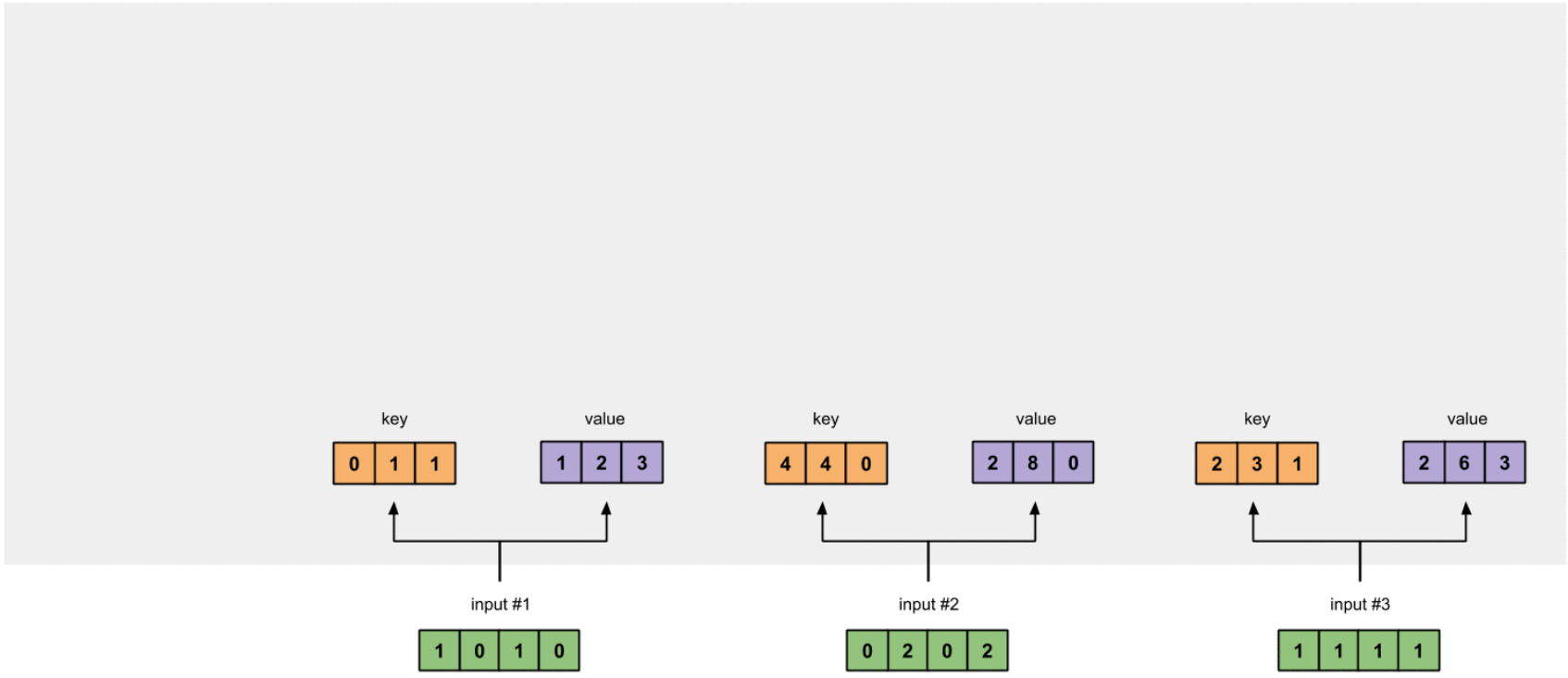
Self-Attention Layers (by Matrices)

Image Credits: https://jalammar.github.io/illustrated-transformer/



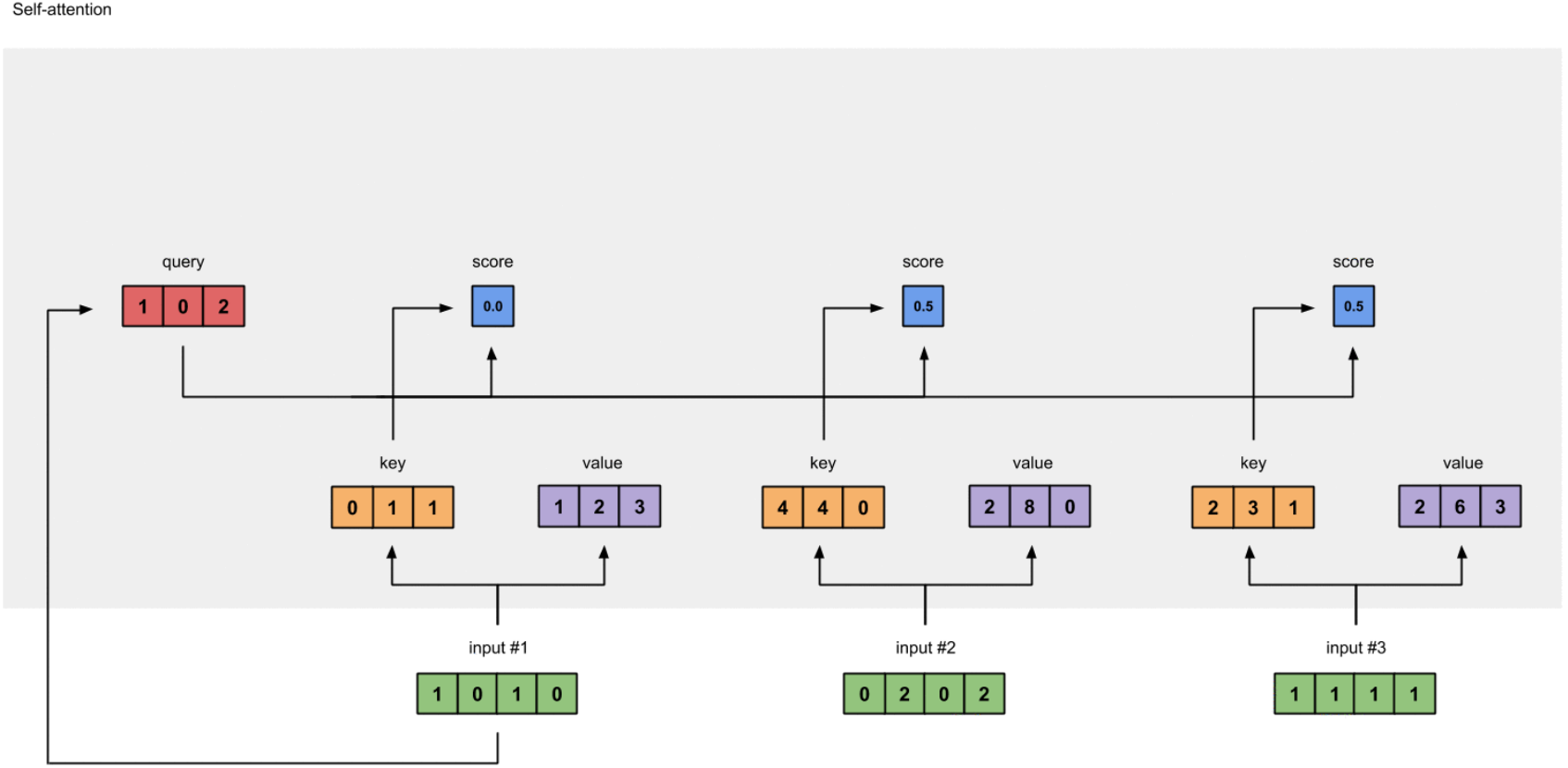
Self-Attention Layers (by Matrices)

Image Credits: https://jalammar.github.io/illustrated-transformer/



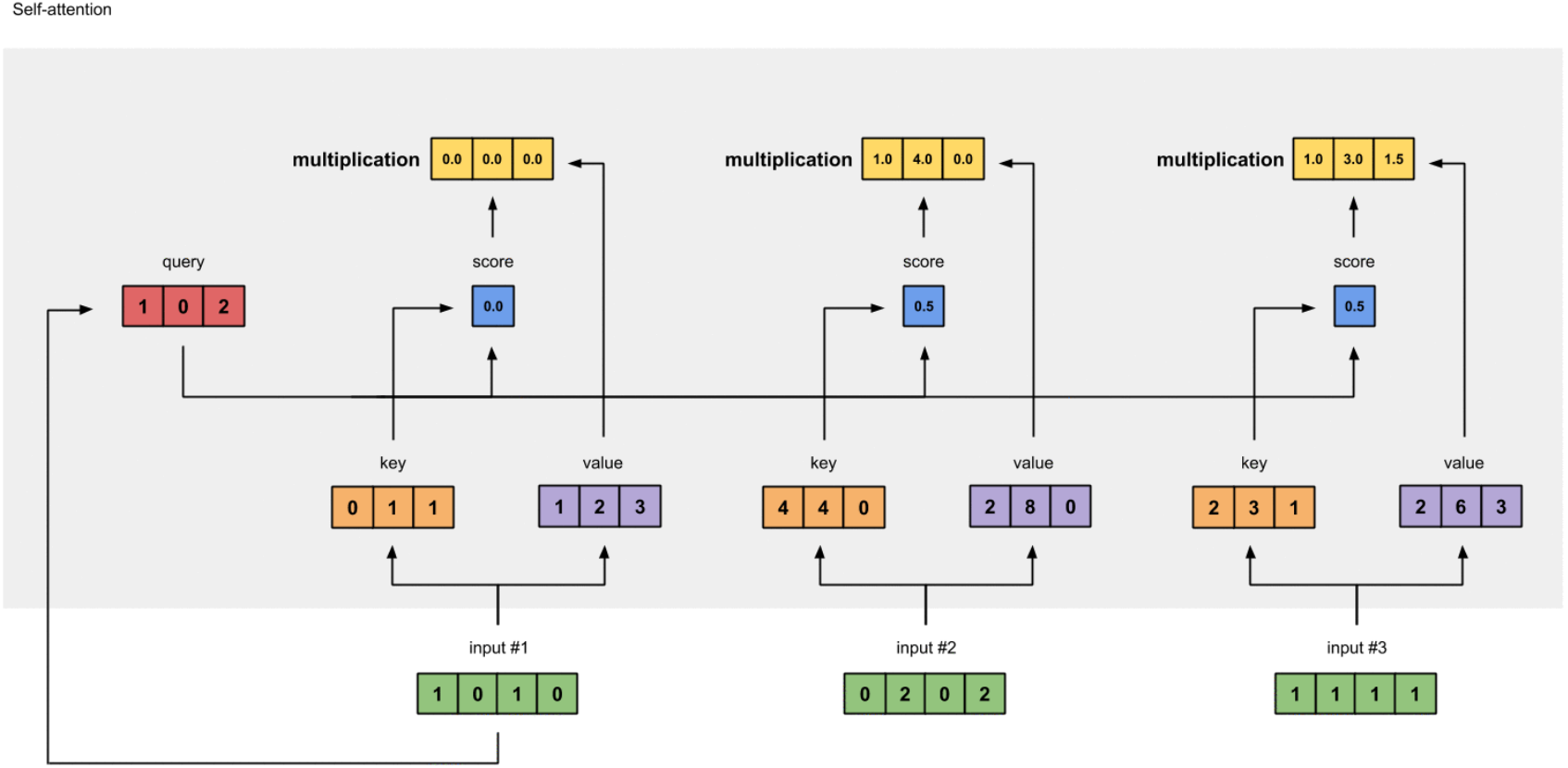
Self-Attention Layers (by Matrices)

Image Credits: https://jalammar.github.io/illustrated-transformer/



Self-Attention Layers (by Matrices)

Image Credits: https://jalammar.github.io/illustrated-transformer/



Multi-headed Self Attention

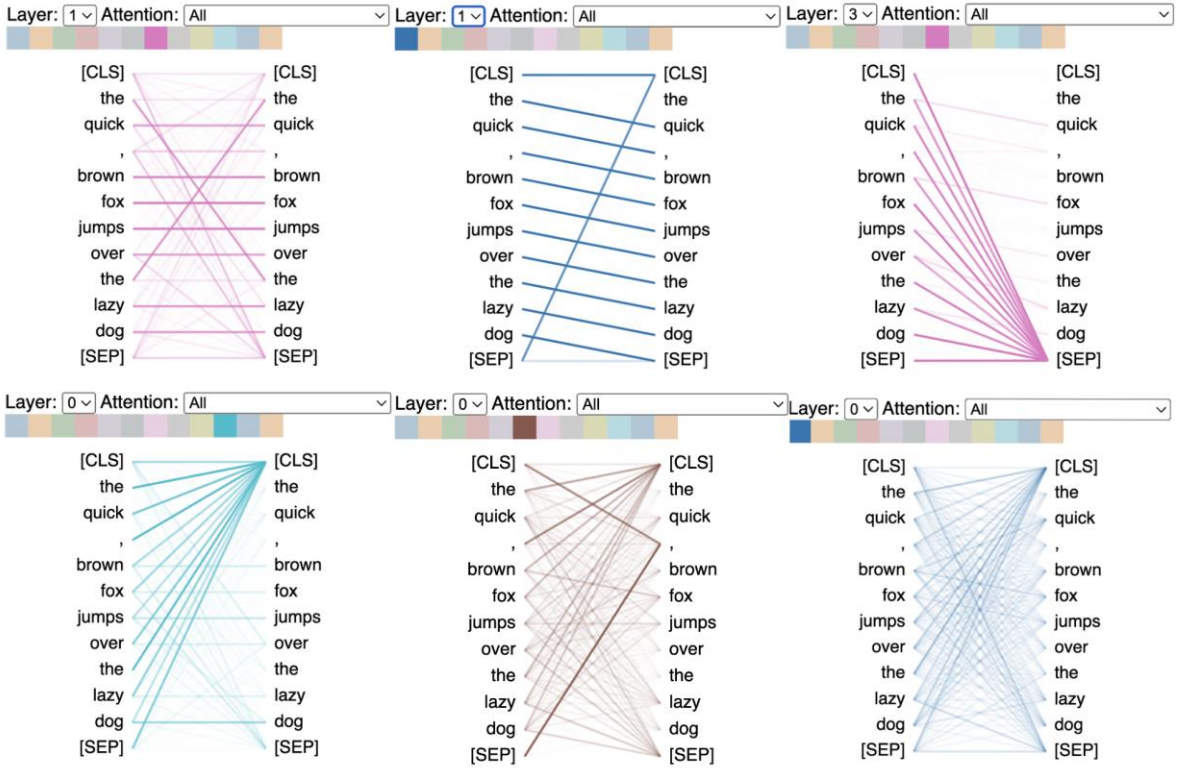


Image Credit: https://www.comet.com/site/blog/explainable-ai-for-transformers/

Pre-training

570Gb

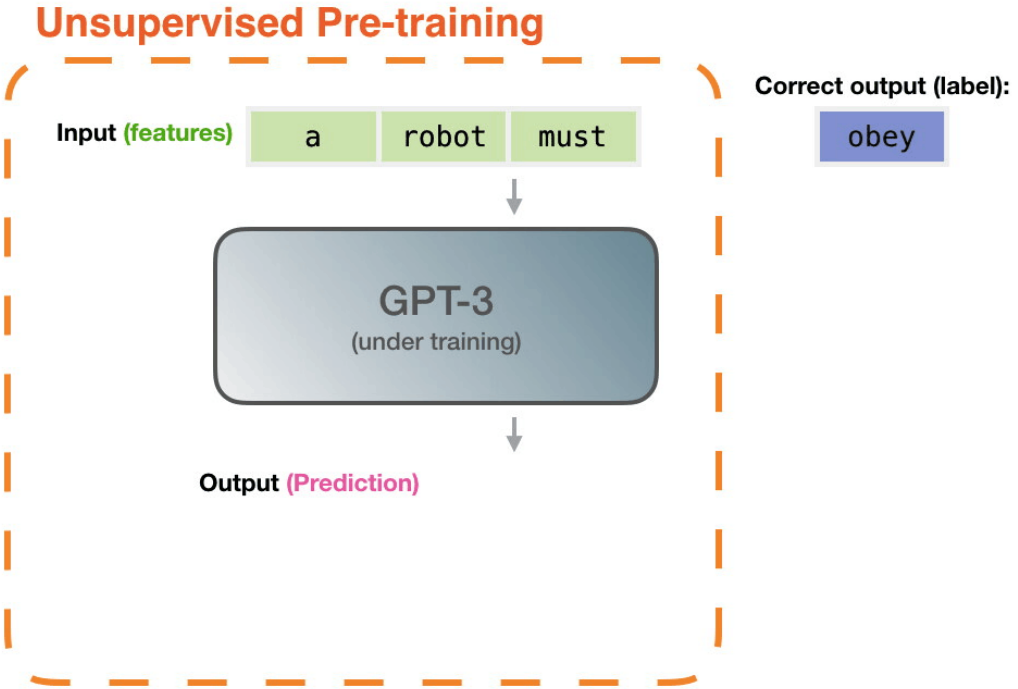


Image Credit: https://jalammar.github.io/illustrated-transformer/

Reinforcement Learning with Human Feedback (RLHF)

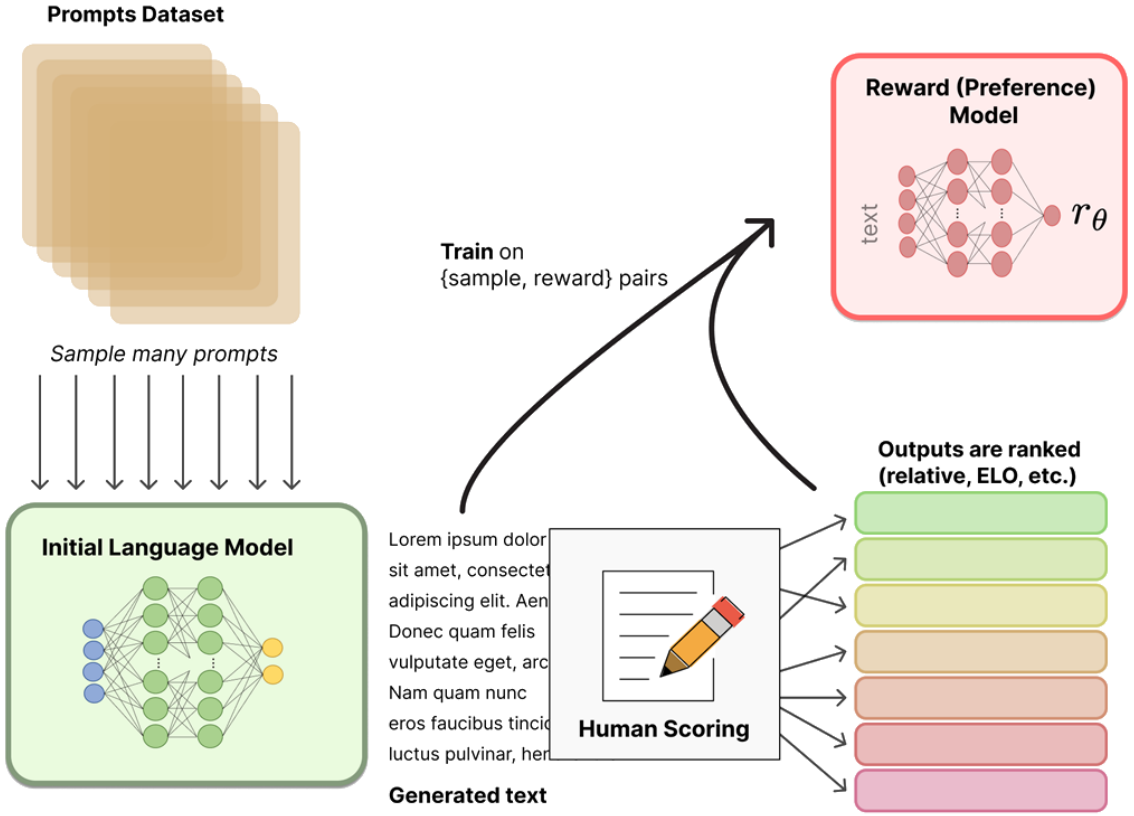


Image Credits:
https://jalammar.github.io/illustrated-transformer/

Output of the last decoder block's linear layer. A score for every word in the vocab.

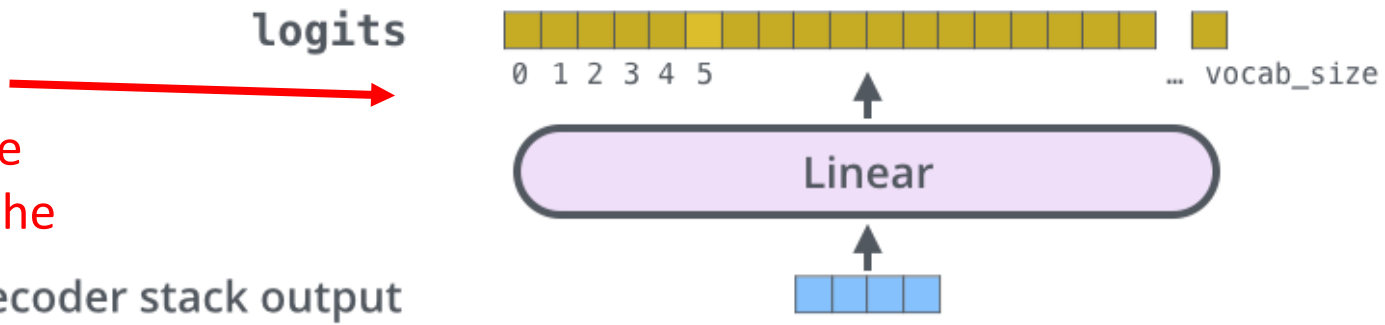


Image Credits:
https://jalammar.github.io/illustrated-transformer/

Convert to class probabilities.

Output of the last decoder block's linear layer. A score for every word in the vocab.

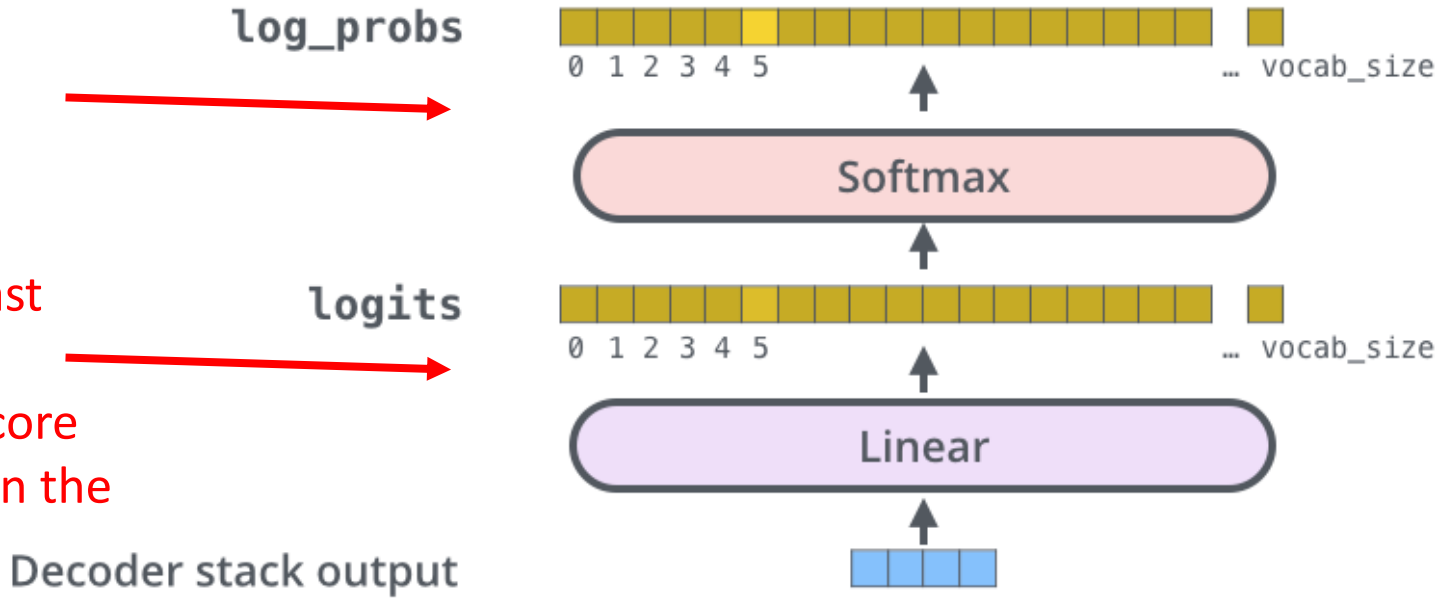
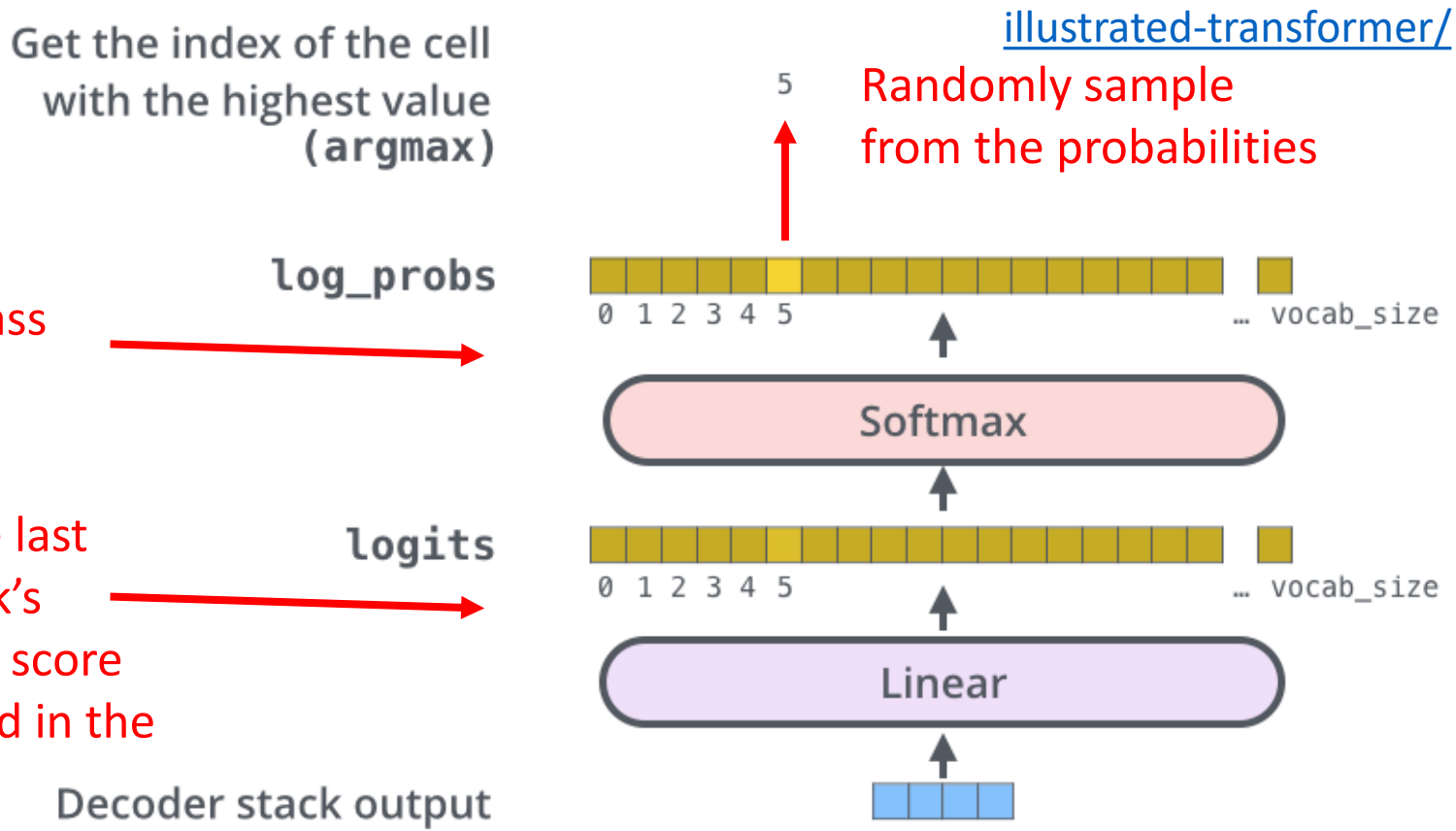


Image Credits:
https://jalammar.github.io/illustrated-transformer/



Which word in our vocabulary is associated with this index?

Get the index of the cell with the highest value (argmax)

log_probs

Convert to class probabilities.

Output of the last decoder block's linear layer. A score for every word in the vocab.

logits

Decoder stack output

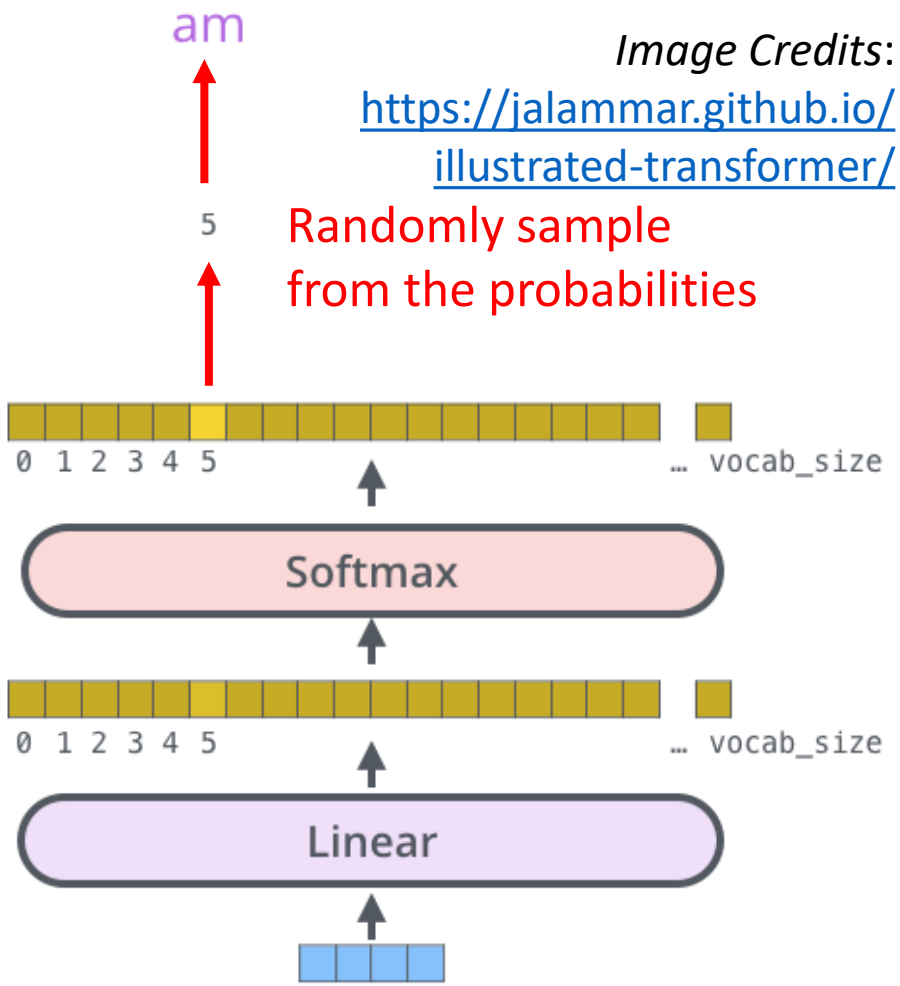


Image Credits: https://jalammar.github.io/illustrated-transformer/