
DAY 1 · LANGUAGE MODELS · PART 3 OF 5

Attention and Transformers

Why context matters, self-attention, multi-head attention and the transformer block

Pavlos Protopapas

Harvard University



Outline

- › Motivation for Attention
- › Attention Basics
- › Building Blocks of Transformers, BERT and GPT
 - › Multi-head Attention Block
 - › Positional Encoding
- › Bringing It All Together

Limitations

Language Model Wishlist?

- › Need **long context**
- › We want to capture the **context dependent** semantics of words

FCNN have fixed context.

Word2Vec gives us
one embedding per word.

Limitations

Ambiguity in static embeddings:

Word2Vec gives us
one embedding per word.

The **bank** is open on Fridays.

I went to the **bank** to take a walk by the river.

Limitations

Language Model Wishlist?

- › Need long context
- › We want to capture the **context dependent** semantics of words
- › We want to respect the **sequential** nature of language
- › We need an architecture that can be trained in **parallel** (non-Markovian property)

FCNN have fixed context.

Word2Vec gives us one embedding per word.

FCNN is not able to capture sequential information.

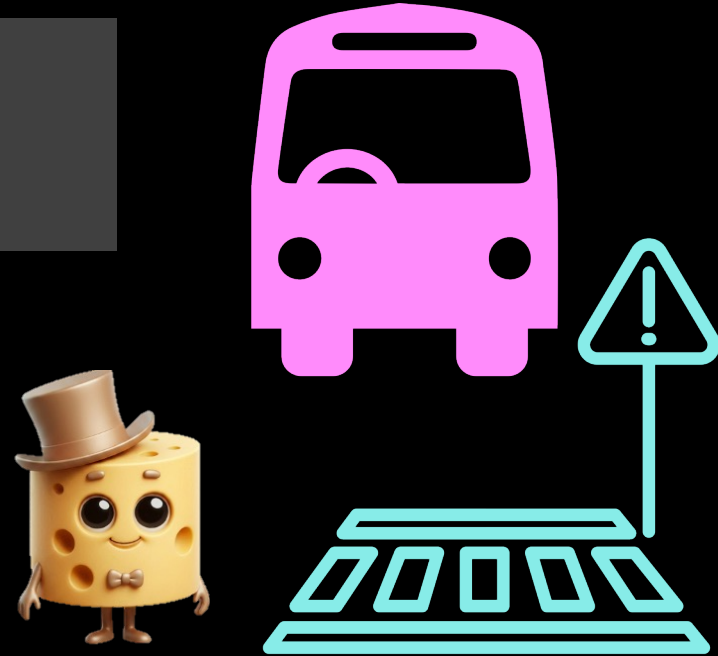
RNNs, GRUs and LSTM can not be trained in parallel as they rely on previous states.

Outline

- › Motivation for Attention
- › **Attention Basics**
- › Building Blocks of Transformers, BERT and GPT
 - › Multi-head Attention Block
 - › Positional Encoding
- › Bringing It All Together

Attention | Example

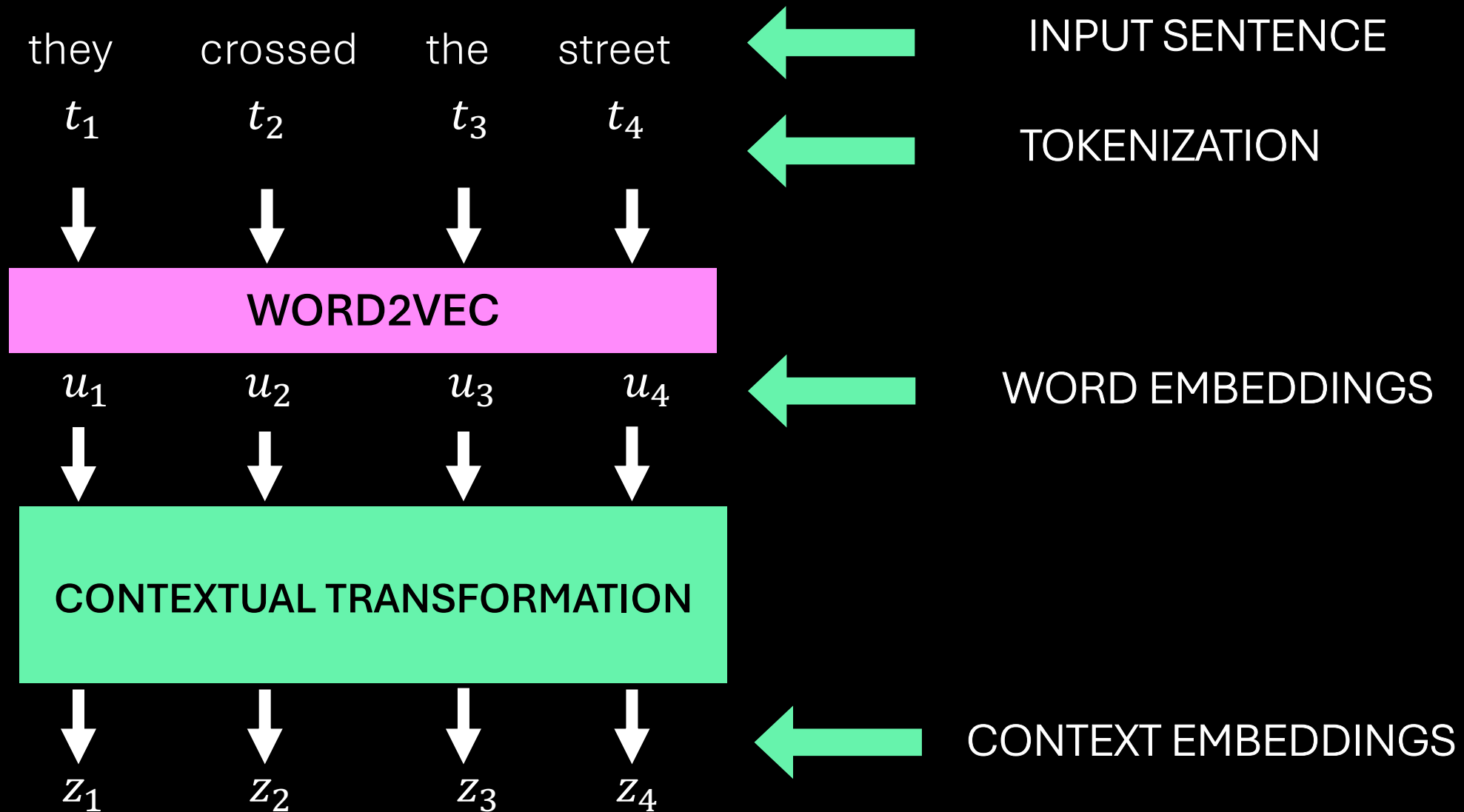
Chester was hit by a bus because **they** crossed the street.



How do we find the referent of the word '**they**' in the sentence?

No anthropomorphic cheeses were harmed in this lecture. This is a work of fiction. Any resemblance to actual people, places, events, or cheeses, living or otherwise, is purely coincidental.

Contextual Embeddings

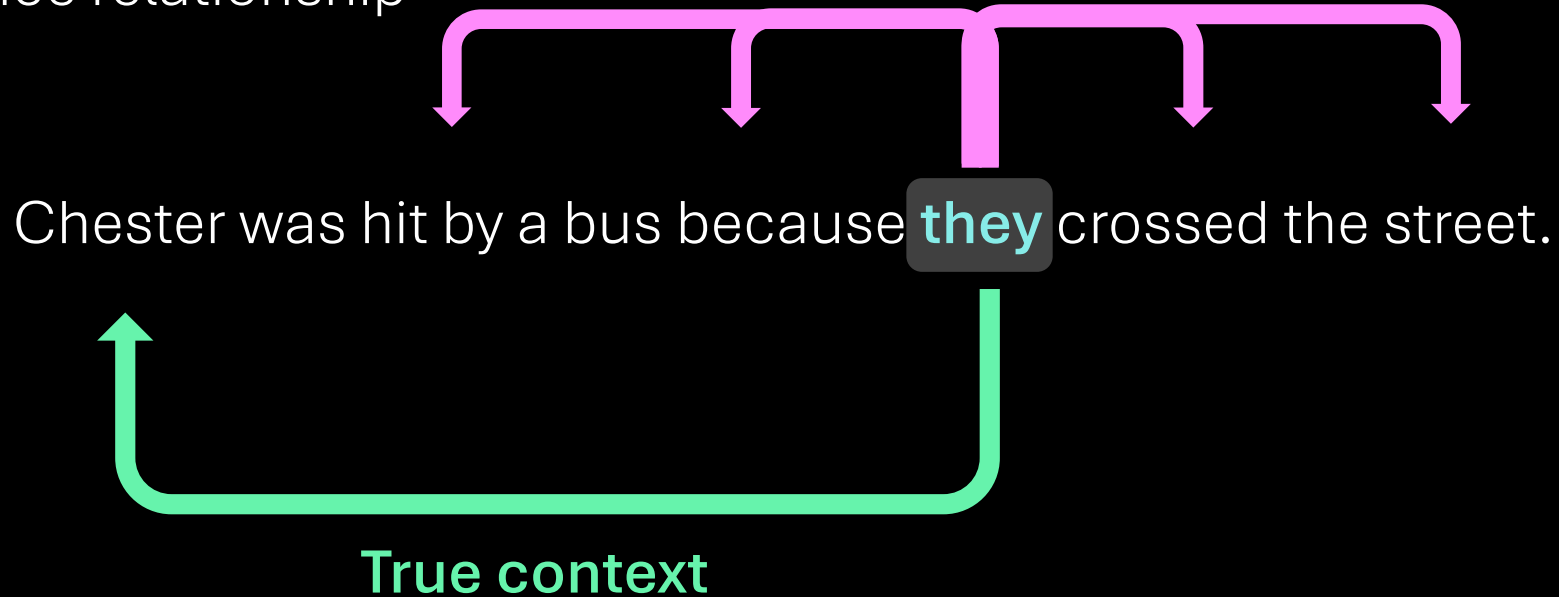


Attention | Example



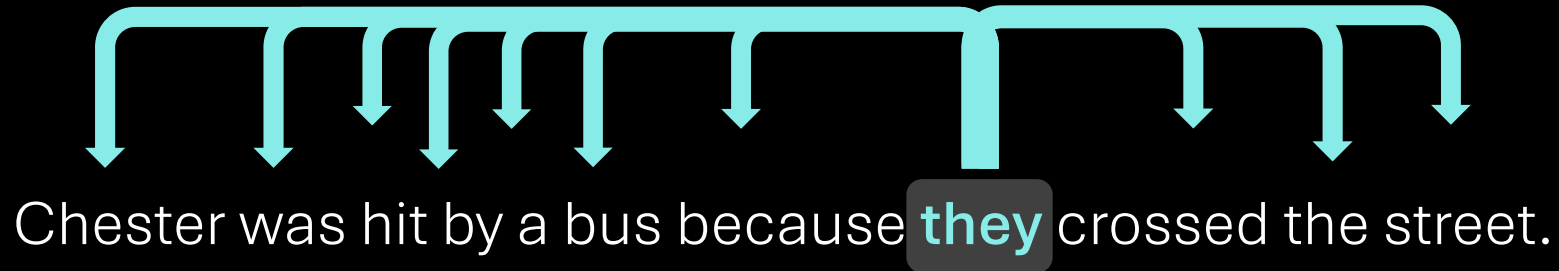
IDEA #1:

Distance relationship



❌ This idea does not work because context can be **unevenly** spread out in a sentence.

Attention | Example



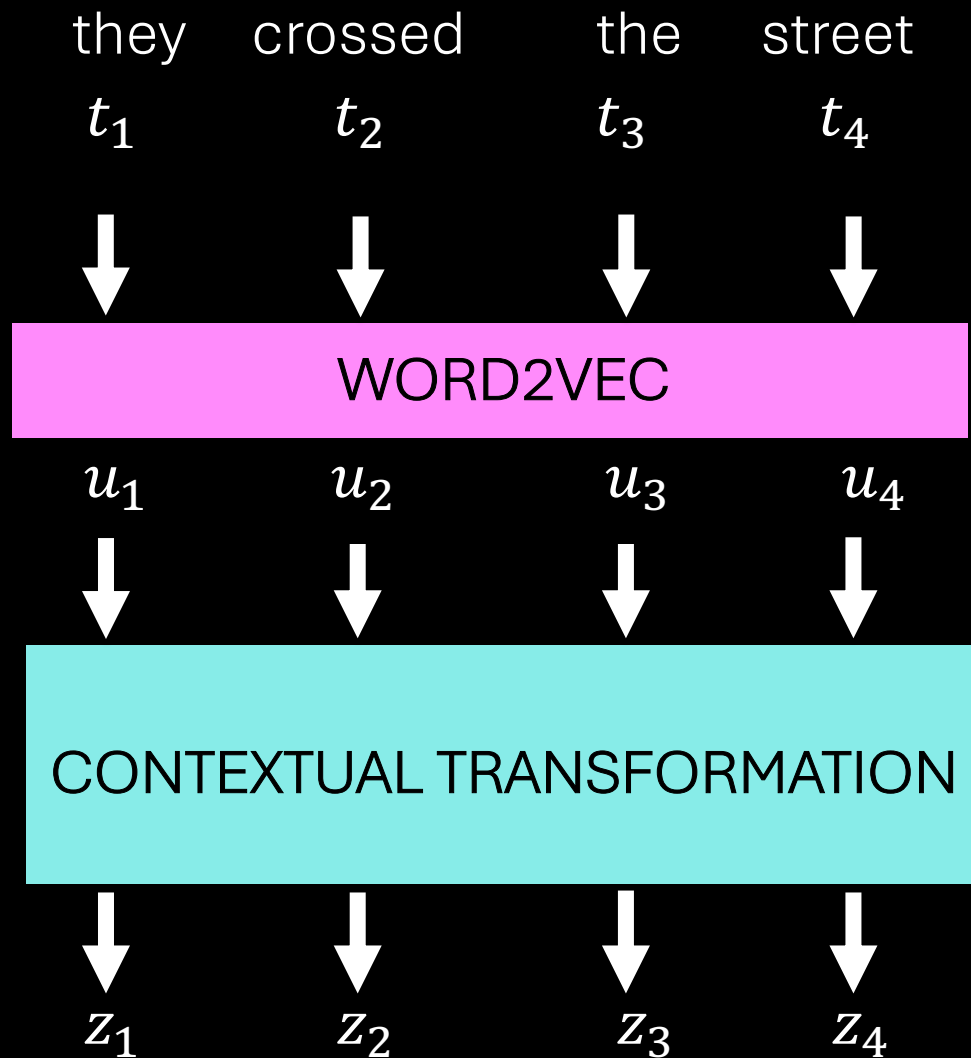
We would want the word ‘**they**’ to be related to every other word in the sentence, irrespective of the distance.

One way to do this would be to make the contextual embedding of the word ‘they’ to be a **linear combination** of the other word embeddings.

Outline

- › Motivation for Attention
- › Attention Basics
- › **Building Blocks of Transformers, BERT and GPT**
 - › Multi-head Attention Block
 - › Positional Encoding
- › Bringing It All Together

Contextual Embeddings



Contextual embeddings, z , can be obtained by applying a linear transformation to the word embeddings, u .

So, we want:

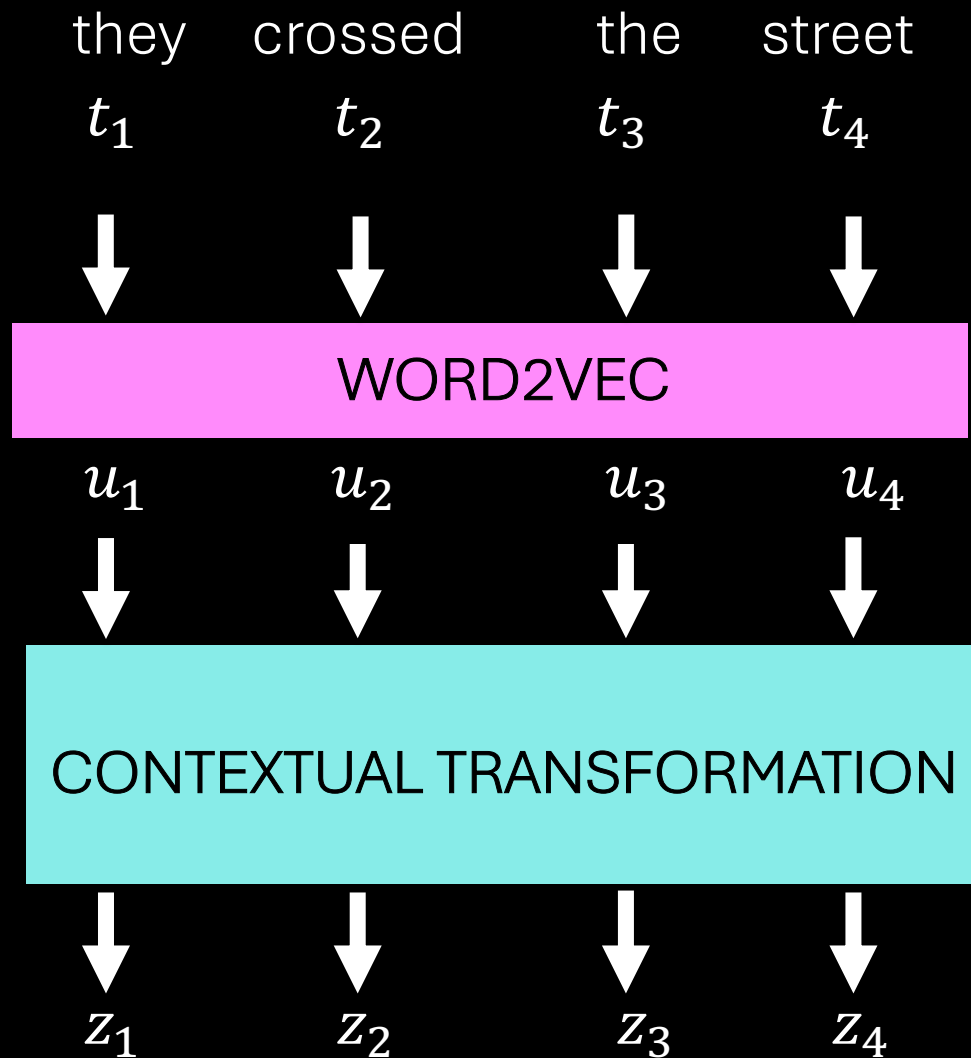
$$z_1 = \alpha_{11}u_1 + \alpha_{12}u_2 + \alpha_{13}u_3 + \alpha_{14}u_4$$

$$z_2 = \alpha_{21}u_1 + \alpha_{22}u_2 + \alpha_{23}u_3 + \alpha_{24}u_4$$

$$z_3 = \alpha_{31}u_1 + \alpha_{32}u_2 + \alpha_{33}u_3 + \alpha_{34}u_4$$

$$z_4 = \alpha_{41}u_1 + \alpha_{42}u_2 + \alpha_{43}u_3 + \alpha_{44}u_4$$

Contextual Embeddings



Contextual embeddings, z , can be obtained by applying a linear transformation to the word embeddings, u .

So, we want:

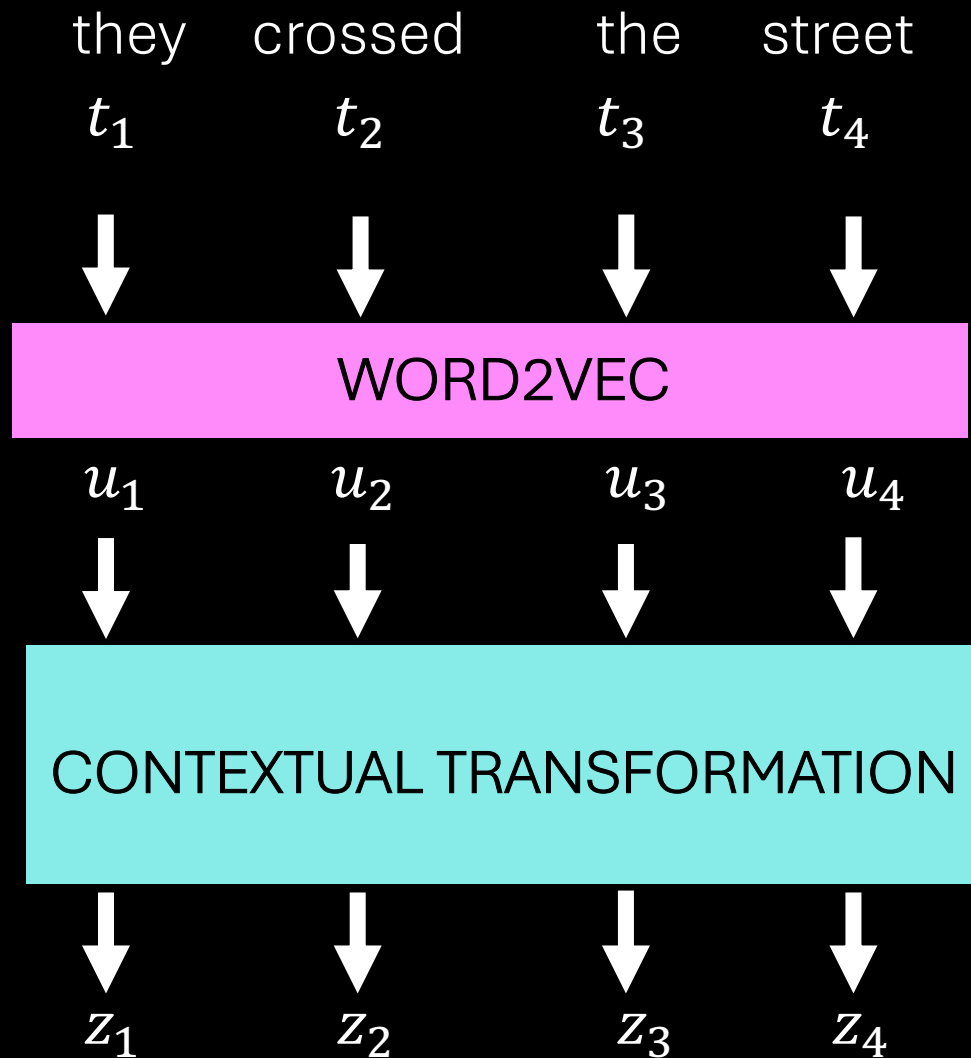
$$z_1 = \alpha_{11}u_1 + \alpha_{12}u_2 + \alpha_{13}u_3 + \alpha_{14}u_4$$

$$z_2 = \alpha_{21}u_1 + \alpha_{22}u_2 + \alpha_{23}u_3 + \alpha_{24}u_4$$

$$z_3 = \alpha_{31}u_1 + \alpha_{32}u_2 + \alpha_{33}u_3 + \alpha_{34}u_4$$

$$z_4 = \alpha_{41}u_1 + \alpha_{42}u_2 + \alpha_{43}u_3 + \alpha_{44}u_4$$

Contextual Embeddings



So, we want

$$z_1 = \alpha_{11}u_1 + \alpha_{12}u_2 + \alpha_{13}u_3 + \alpha_{14}u_4$$

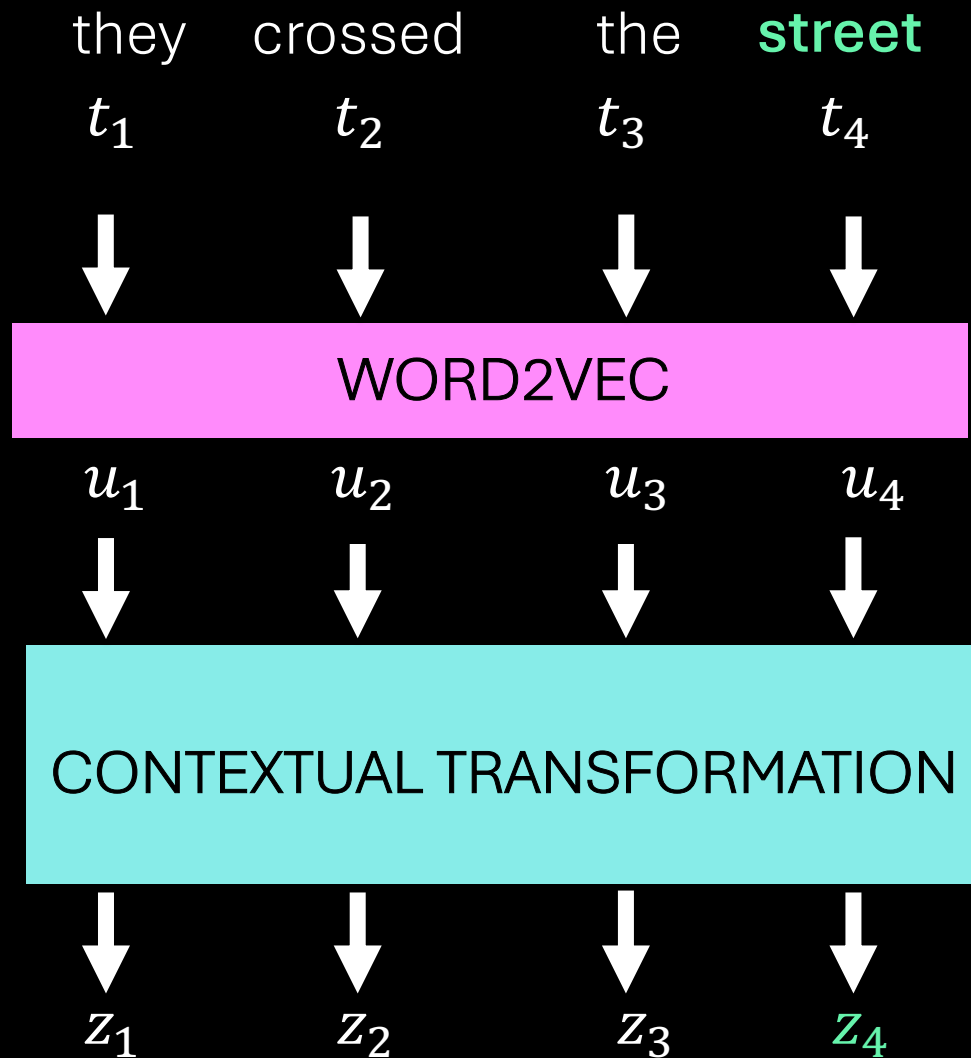
$$z_2 = \alpha_{21}u_1 + \alpha_{22}u_2 + \alpha_{23}u_3 + \alpha_{24}u_4$$

$$z_3 = \alpha_{31}u_1 + \alpha_{32}u_2 + \alpha_{33}u_3 + \alpha_{34}u_4$$

$$z_4 = \alpha_{41}u_1 + \alpha_{42}u_2 + \alpha_{43}u_3 + \alpha_{44}u_4$$

What are these coefficients?

Contextual Embeddings



So, we want

$$z_1 = \alpha_{11}u_1 + \alpha_{12}u_2 + \alpha_{13}u_3 + \alpha_{14}u_4$$

$$z_2 = \alpha_{21}u_1 + \alpha_{22}u_2 + \alpha_{23}u_3 + \alpha_{24}u_4$$

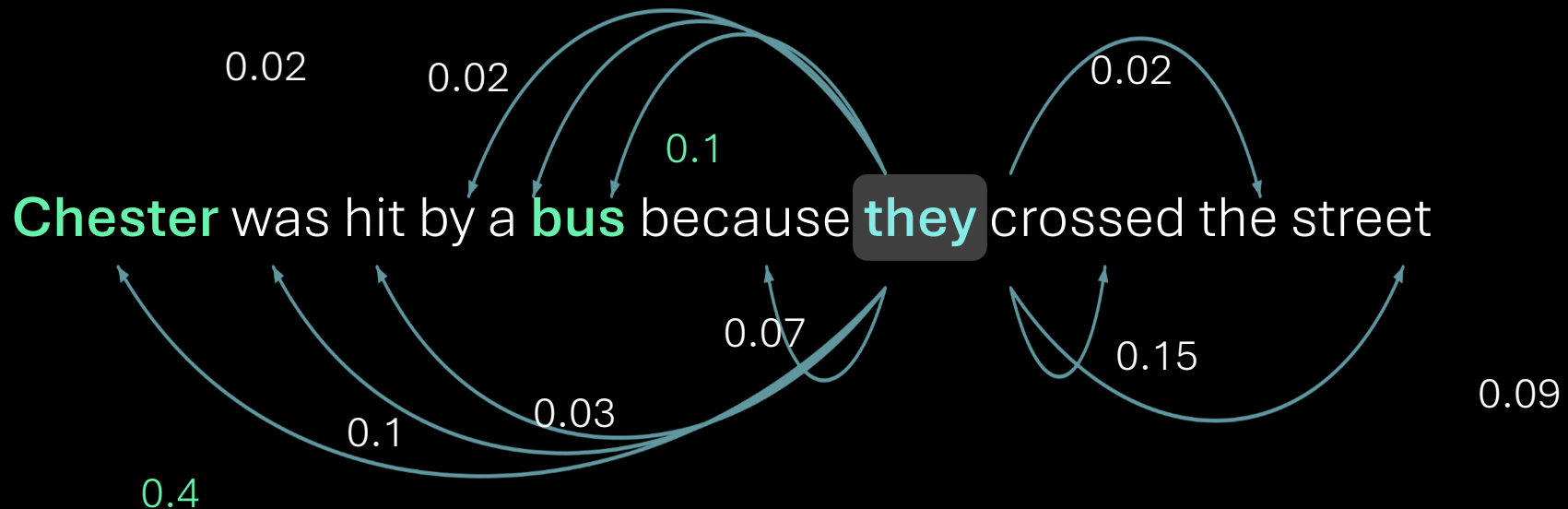
$$z_3 = \alpha_{31}u_1 + \alpha_{32}u_2 + \alpha_{33}u_3 + \alpha_{34}u_4$$

$$z_4 = \alpha_{41}u_1 + \alpha_{42}u_2 + \alpha_{43}u_3 + \alpha_{44}u_4$$

What are these coefficients?

Contextual Embeddings | Coefficients

The coefficients are weights that encapsulate the degree to which new contextual embeddings should incorporate the **influence** of the other tokens.

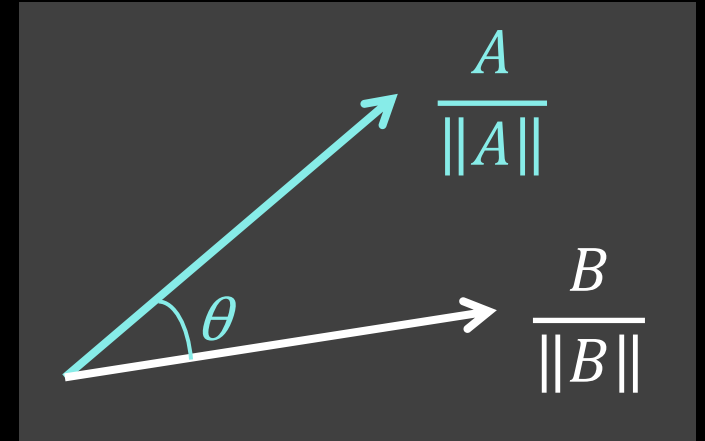


Contextual Embeddings | Coefficients



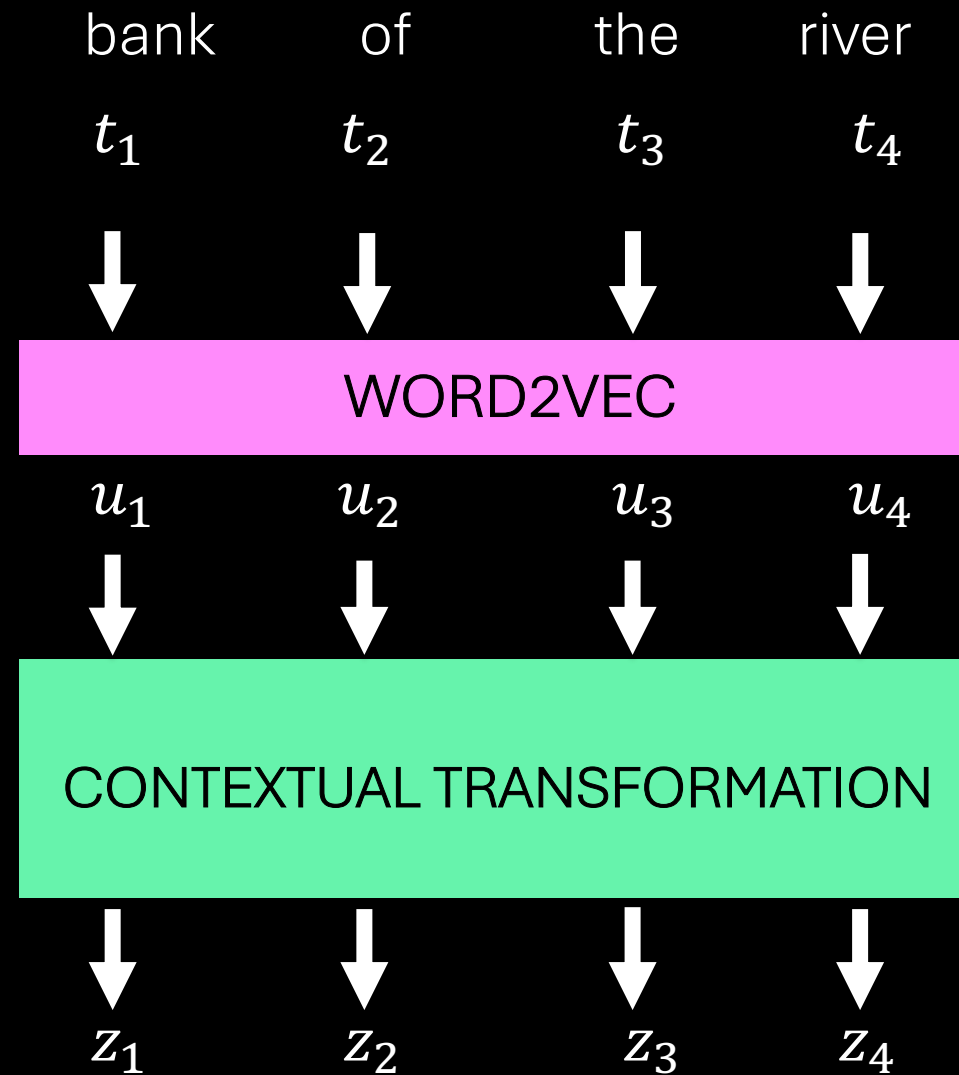
How do we calculate these coefficients?

We can use **cosine similarity** between the word embeddings as the coefficients.

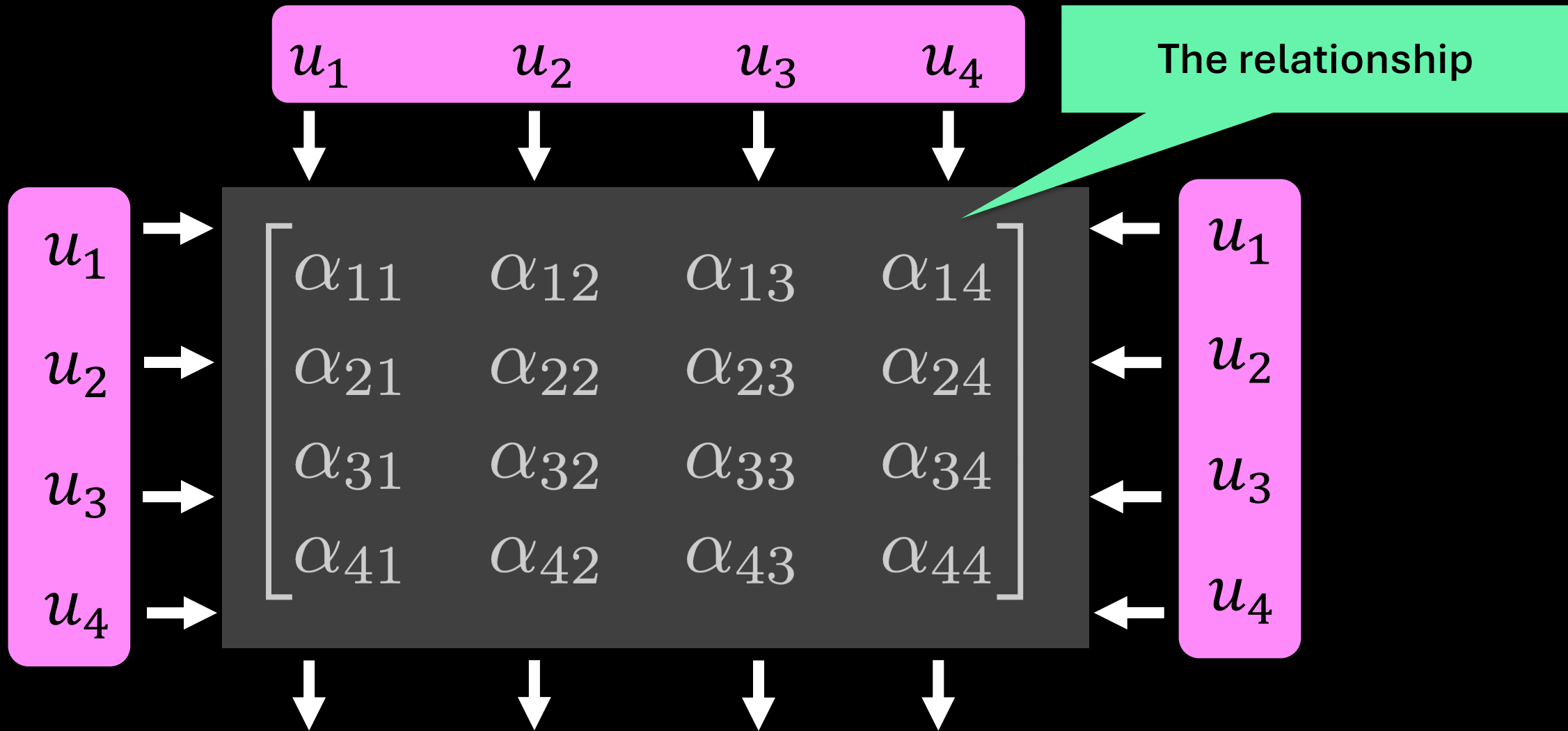


$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

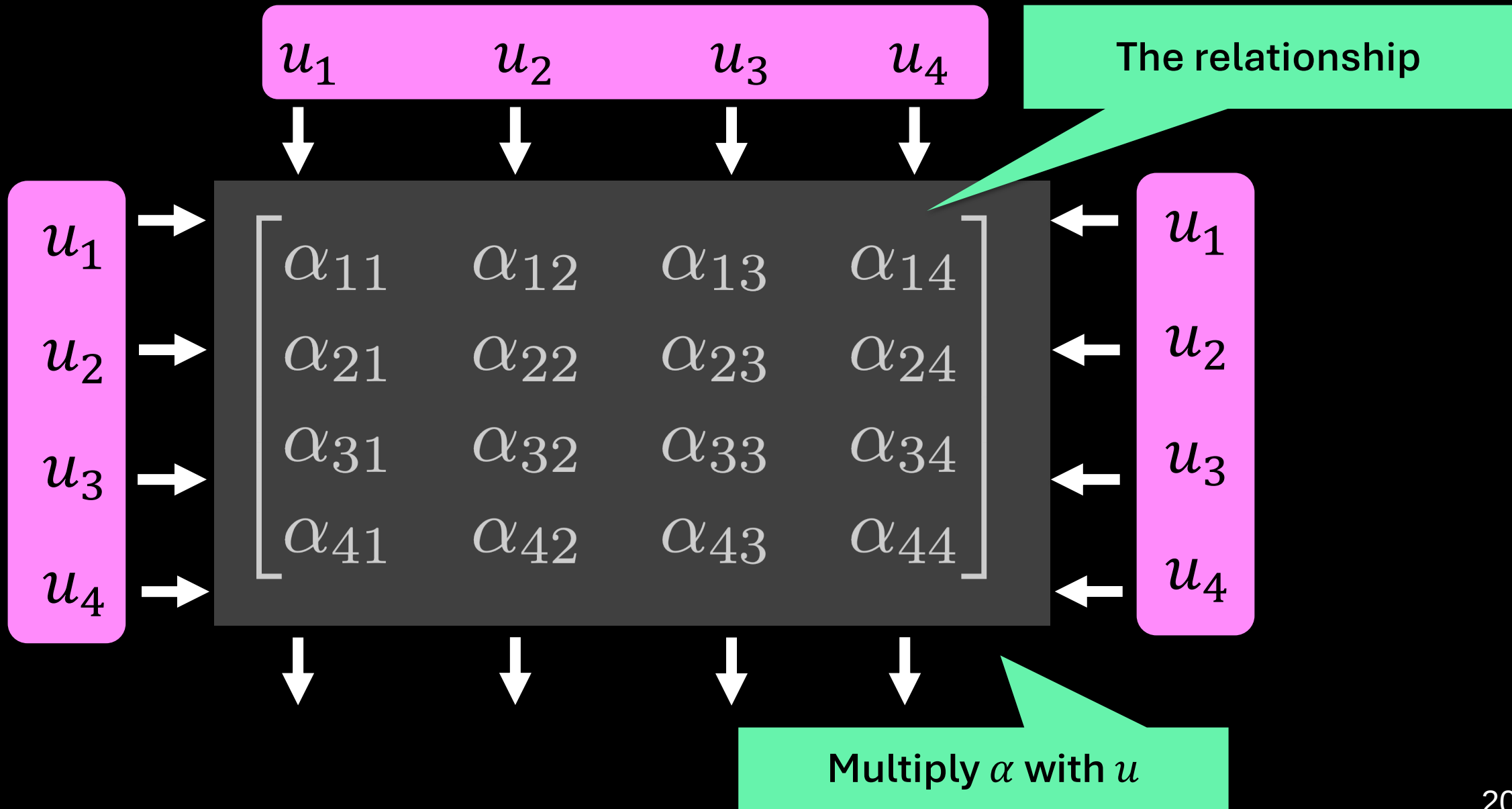
Attention | Example #2



Attention | Behind the Scenes



Attention | Behind the Scenes



Attention | Behind the Scenes

$$u_1 \circ u_1 = s_{11}$$

$$u_1 \circ u_2 = s_{12}$$

$$u_1 \circ u_3 = s_{13}$$

$$u_1 \circ u_4 = s_{14}$$

softmax(.)



WEIGHTS
NORMALIZATION

$$\alpha_{11}$$

$$\alpha_{12}$$

$$\alpha_{13}$$

$$\alpha_{14}$$

Attention | Behind the Scenes

$$u_1 \circ u_1 = s_{11}$$

$$u_1 \circ u_2 = s_{12}$$

$$u_1 \circ u_3 = s_{13}$$

$$u_1 \circ u_4 = s_{14}$$

softmax(.)



WEIGHTS
NORMALIZATION

$$\alpha_{11}$$

$$\alpha_{12}$$

$$\alpha_{13}$$

$$\alpha_{14}$$

“Cosine similarity”

Attention | Behind the Scenes

$$u_1 \circ u_1 = s_{11}$$

$$u_1 \circ u_2 = s_{12}$$

$$u_1 \circ u_3 = s_{13}$$

$$u_1 \circ u_4 = s_{14}$$

softmax(.)



WEIGHTS
NORMALIZATION

$$\alpha_{11}$$

$$\alpha_{12}$$

$$\alpha_{13}$$

$$\alpha_{14}$$

“Cosine similarity”

Attention | Behind the Scenes

Softmax is a function that scales numbers (logits, s_{ij}) into probabilities.

$$\text{softmax}(a_{ij}) = \frac{e^{s_{ij}}}{\sum_{j=1}^k e^{s_{ij}}}$$

$$u_1 \circ u_1 = s_{11}$$

$$\alpha_{11}$$

$$u_1 \circ u_2 = s_{12}$$

$$\alpha_{12}$$

$$u_1 \circ u_3 = s_{13}$$

$$\alpha_{13}$$

$$u_1 \circ u_4 = s_{14}$$

$$\alpha_{14}$$

softmax(.)



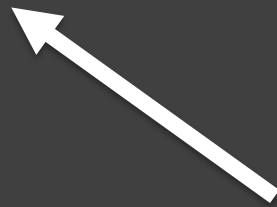
WEIGHTS
NORMALIZATION

$$\sum \alpha_{1i} = 1$$

“Cosine similarity”

Attention | Behind the Scenes

$$z_1 = \alpha_{11}u_1 + \alpha_{12}u_2 + \alpha_{13}u_3 + \alpha_{14}u_4$$



New contextual
word embedding

Attention | Behind the Scenes

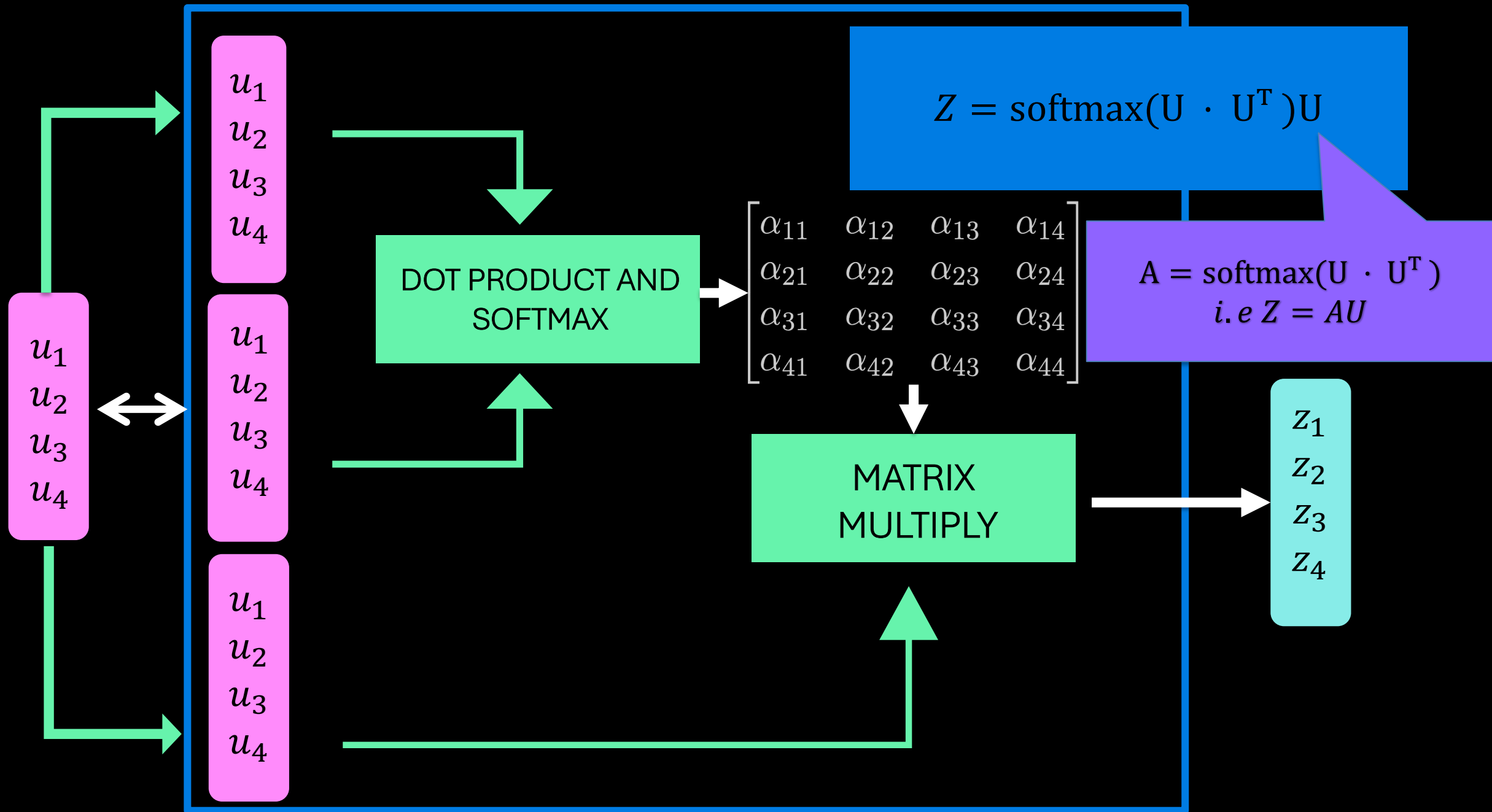
Similarly...

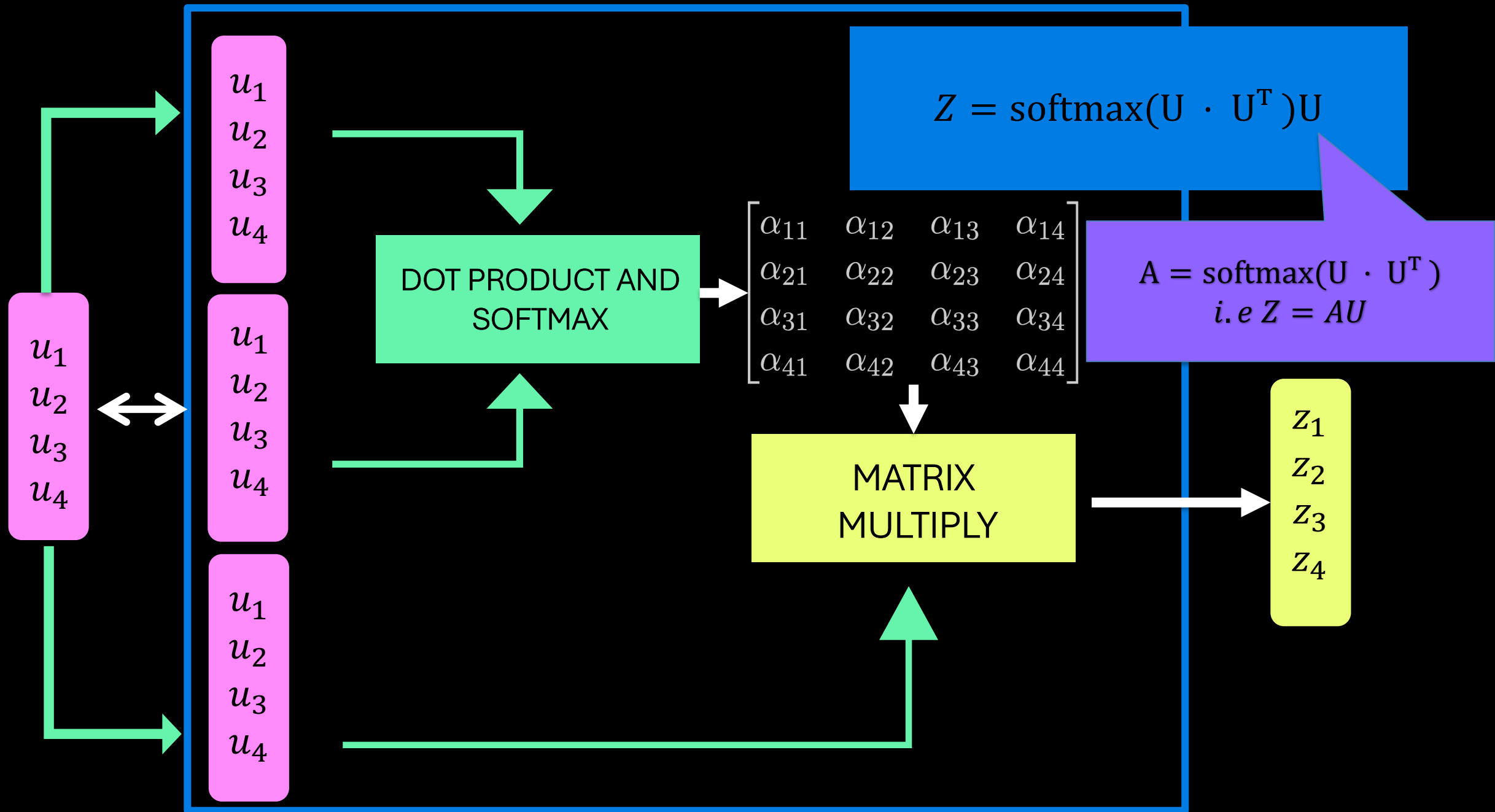
$$z_1 = \alpha_{11}u_1 + \alpha_{12}u_2 + \alpha_{13}u_3 + \alpha_{14}u_4$$

$$z_2 = \alpha_{21}u_1 + \alpha_{22}u_2 + \alpha_{23}u_3 + \alpha_{24}u_4$$

$$z_3 = \alpha_{31}u_1 + \alpha_{32}u_2 + \alpha_{33}u_3 + \alpha_{34}u_4$$

$$z_4 = \alpha_{41}u_1 + \alpha_{42}u_2 + \alpha_{43}u_3 + \alpha_{44}u_4$$



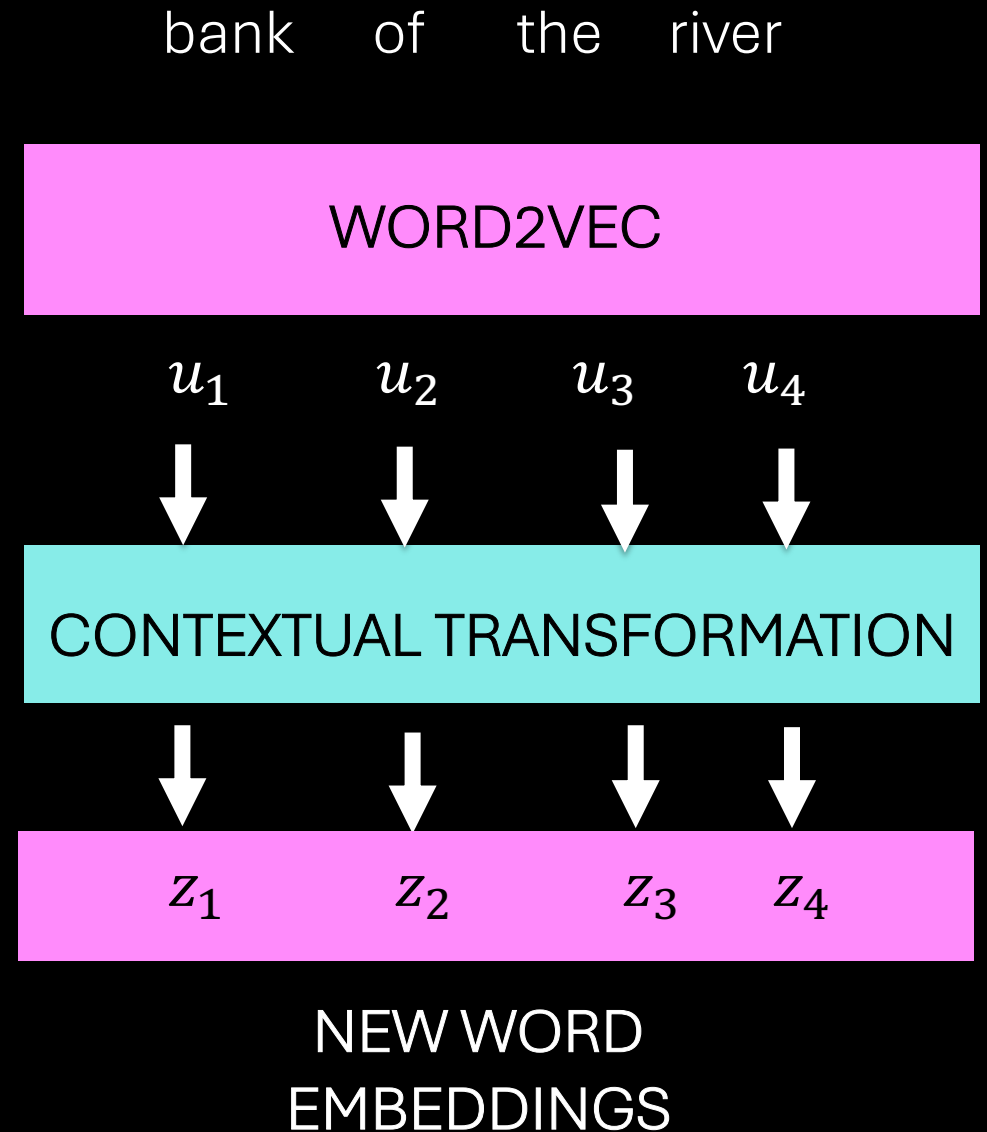


Attention

ATTENTION ISSUES?

This process gives us new embeddings with some context. However, we still have the following issues:

1. No weights are trained in the process.
2. The context captured is shallow — not enough representational power.
3. There is no positional information encoded.

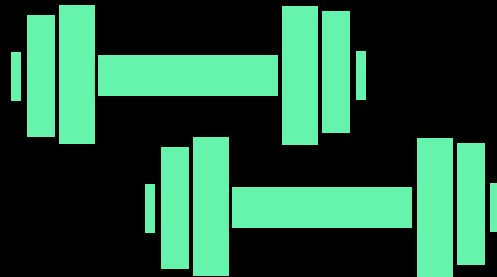


Attention

Can we do even better?

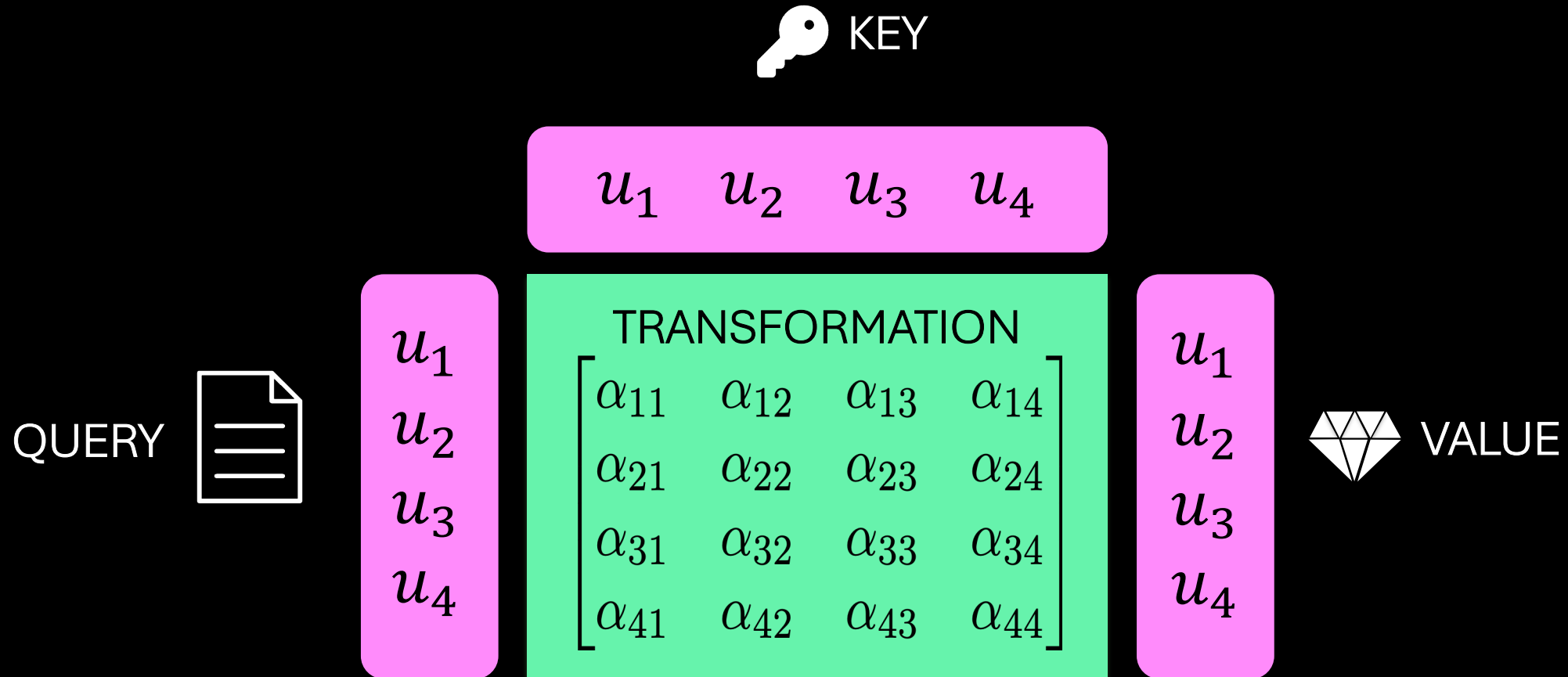


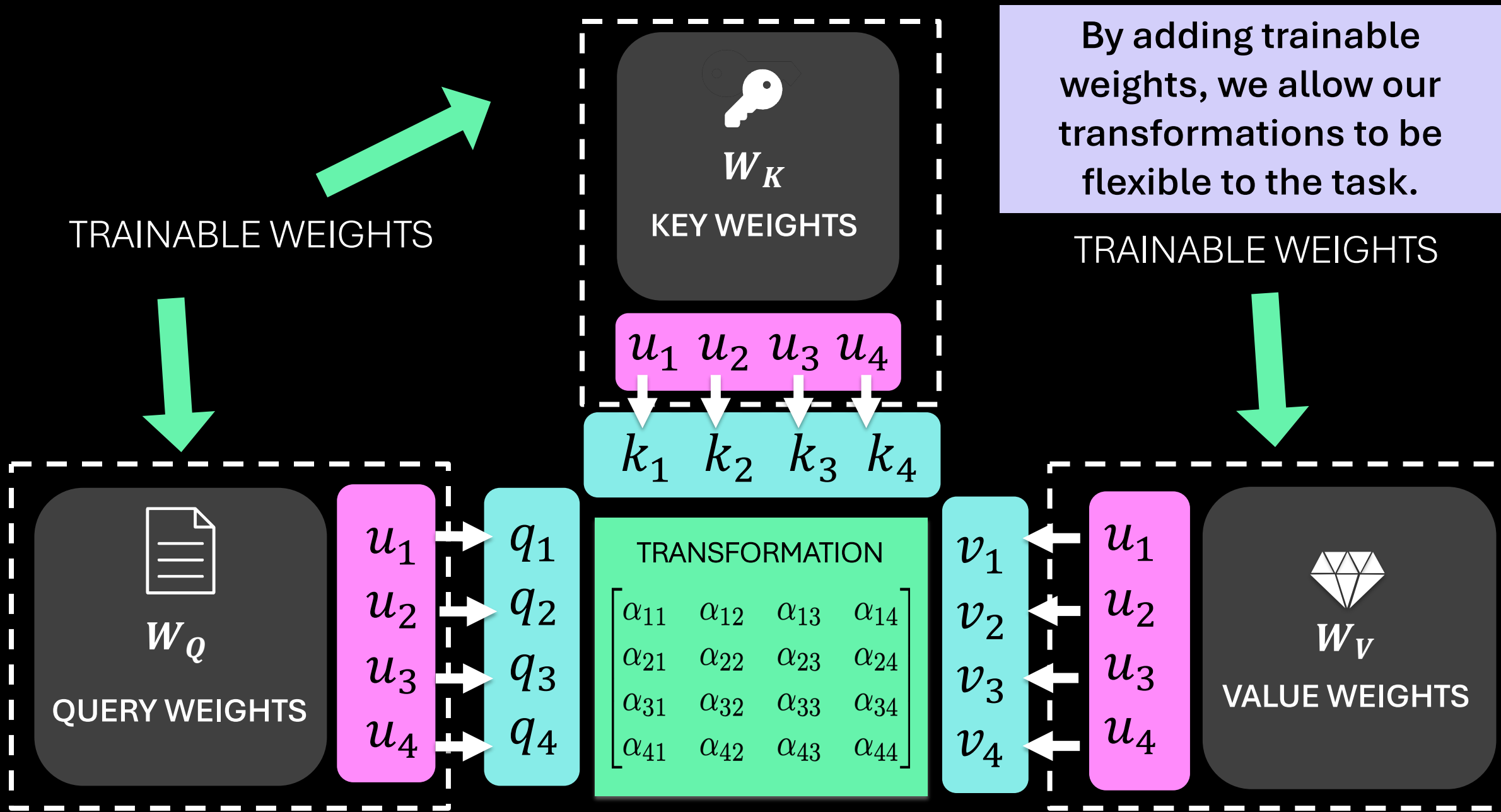
Yes! Let's add some weights and train.



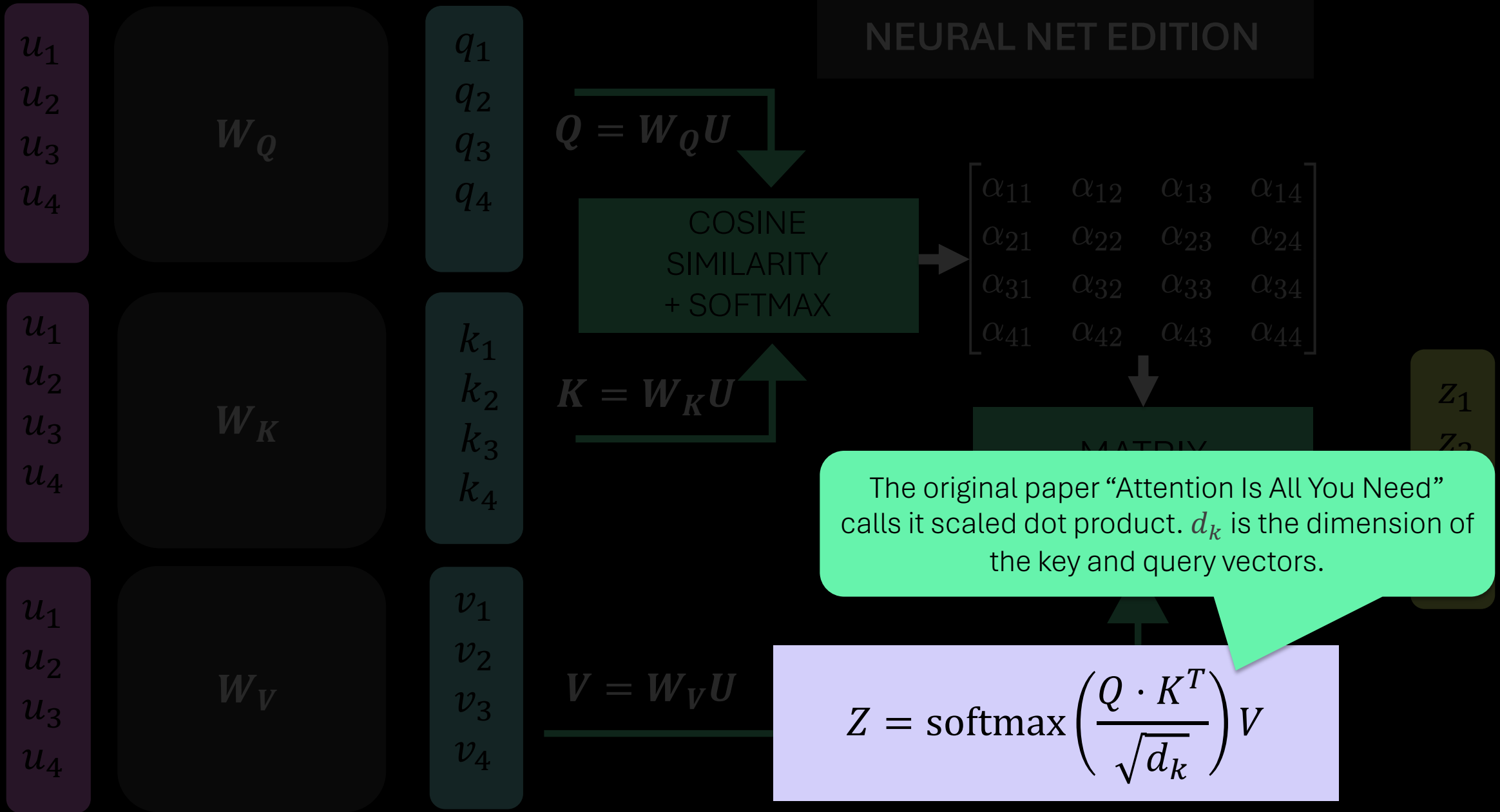
Attention | Database Analogy

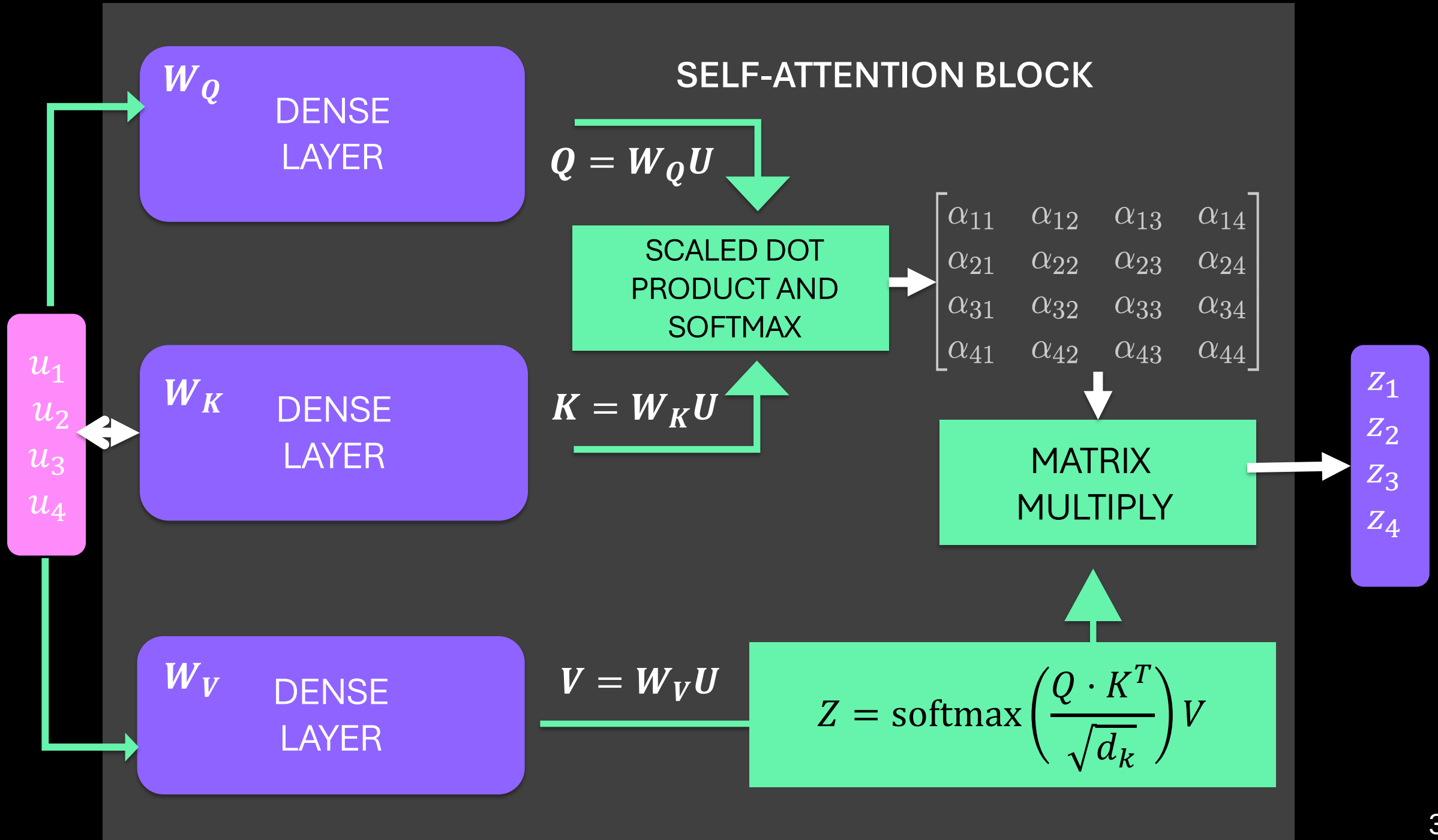
For simplicity, we stick to our database analogy of QUERY, KEY & VALUE.





NEURAL NET EDITION





Attention

ATTENTION ISSUES?

1. ~~No weights are trained in the process.~~
2. Attention leads to **limited contextual mapping**.
3. There is no positional information encoded.



Attention | Do We Have Enough Attention?



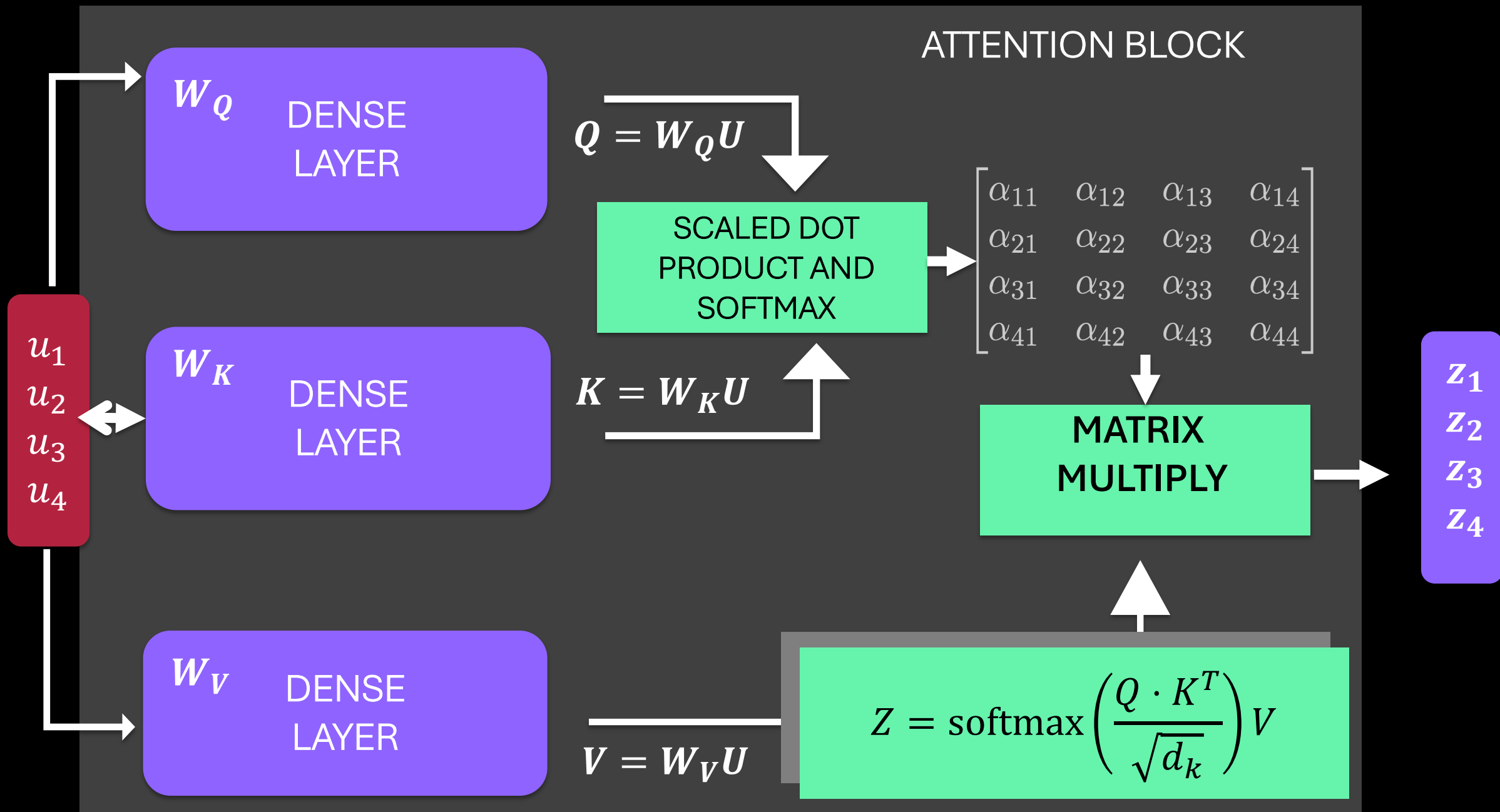
We need to have multiple attention mechanisms to look for different relations.

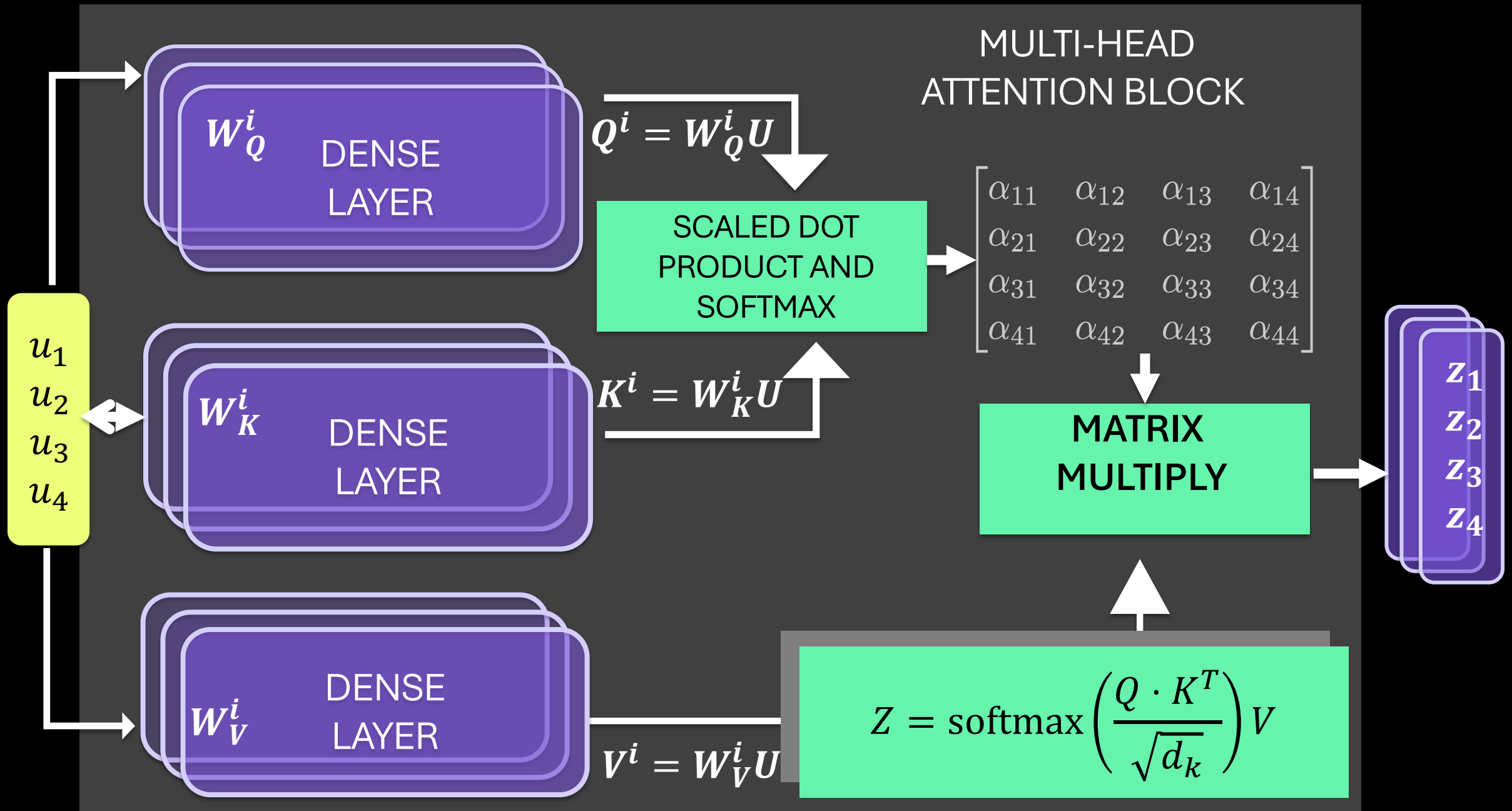
ATTENTION
MECHANISM #1

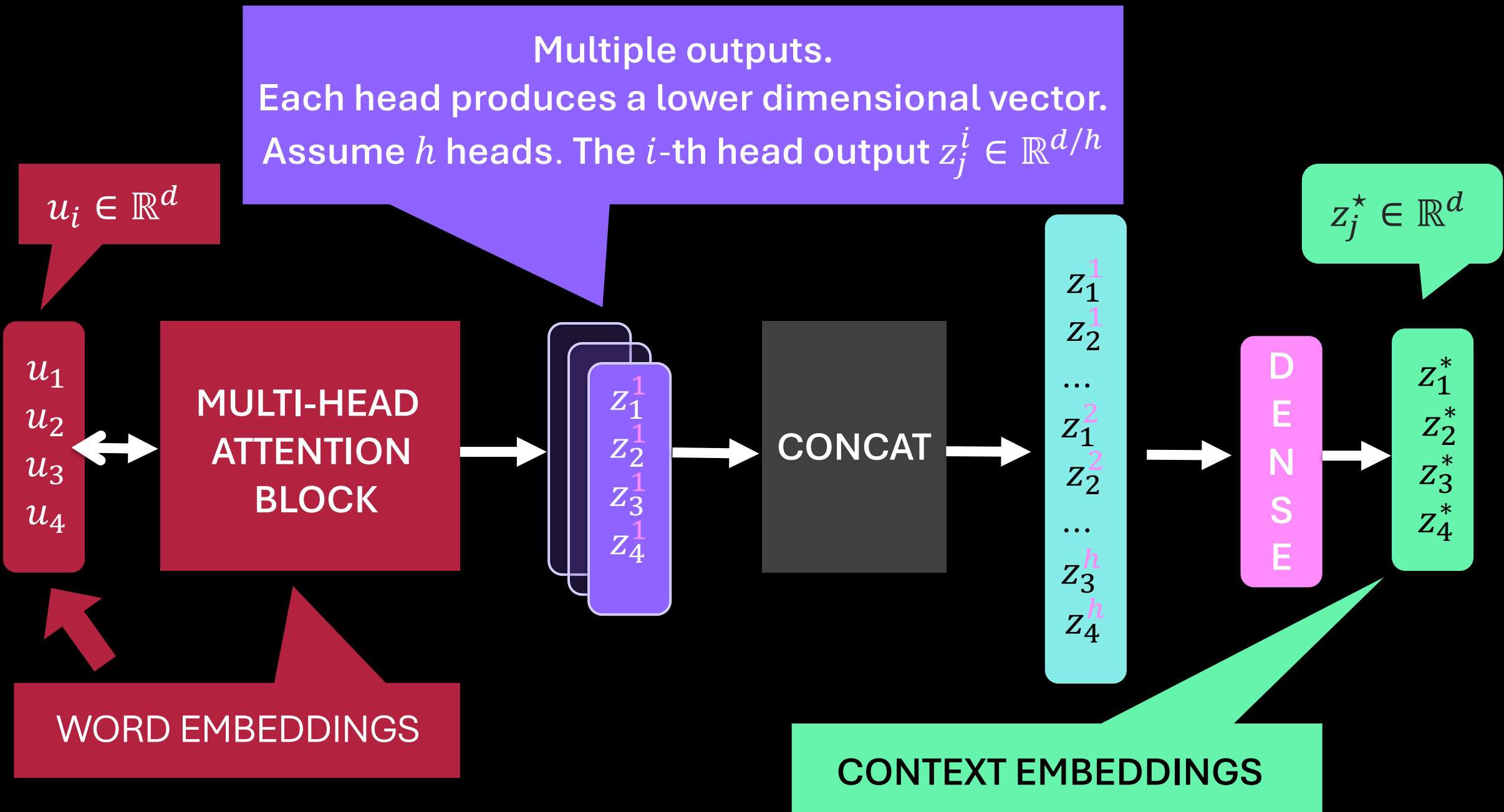
ATTENTION
MECHANISM #2

Chester spoke to Pavlos about attention.

ATTENTION
MECHANISM #3







CONTEXT EMBEDDINGS

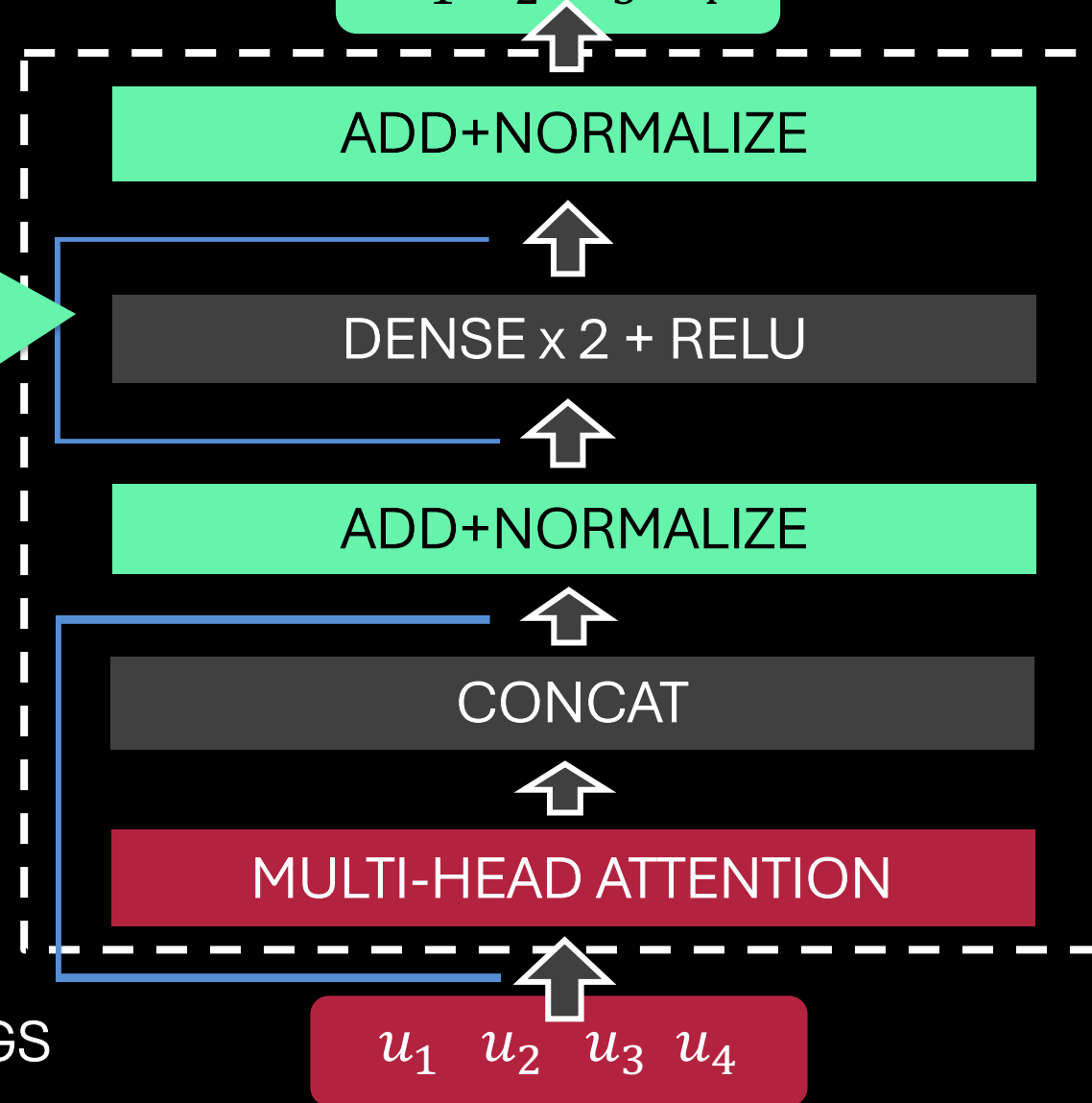
z_1^* z_2^* z_3^* z_4^*

TRANSFORMER LAYER

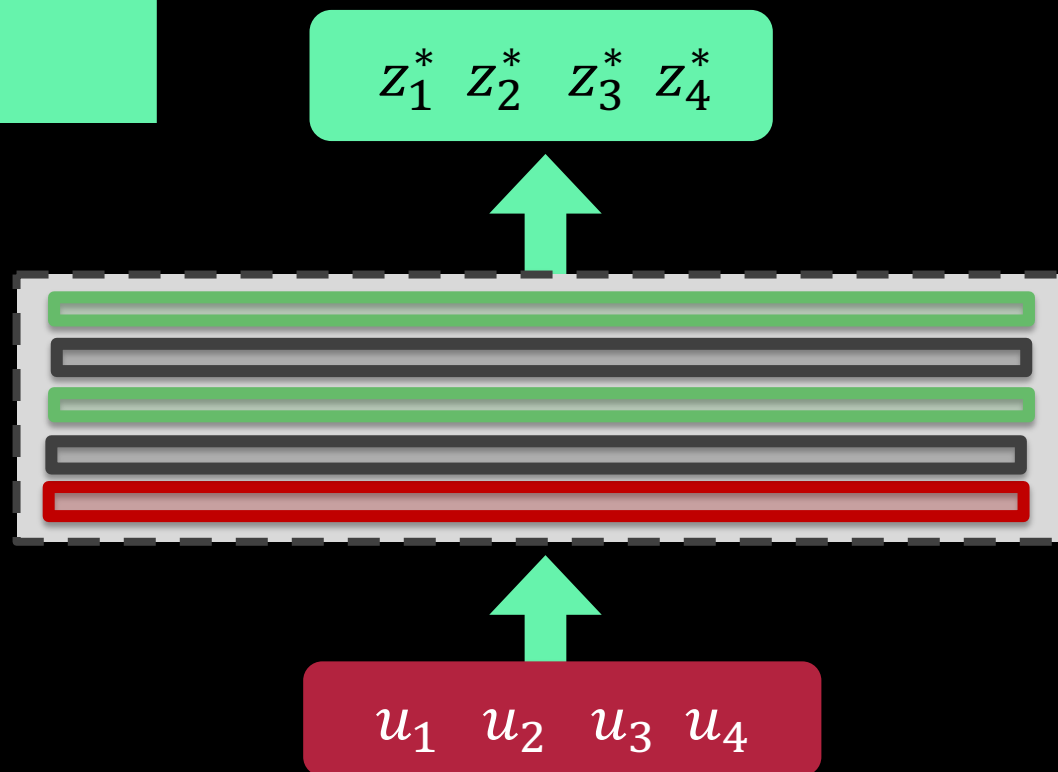
A skip connection passes inputs directly to later layers, helping preserve information and improve training.

WORD EMBEDDINGS

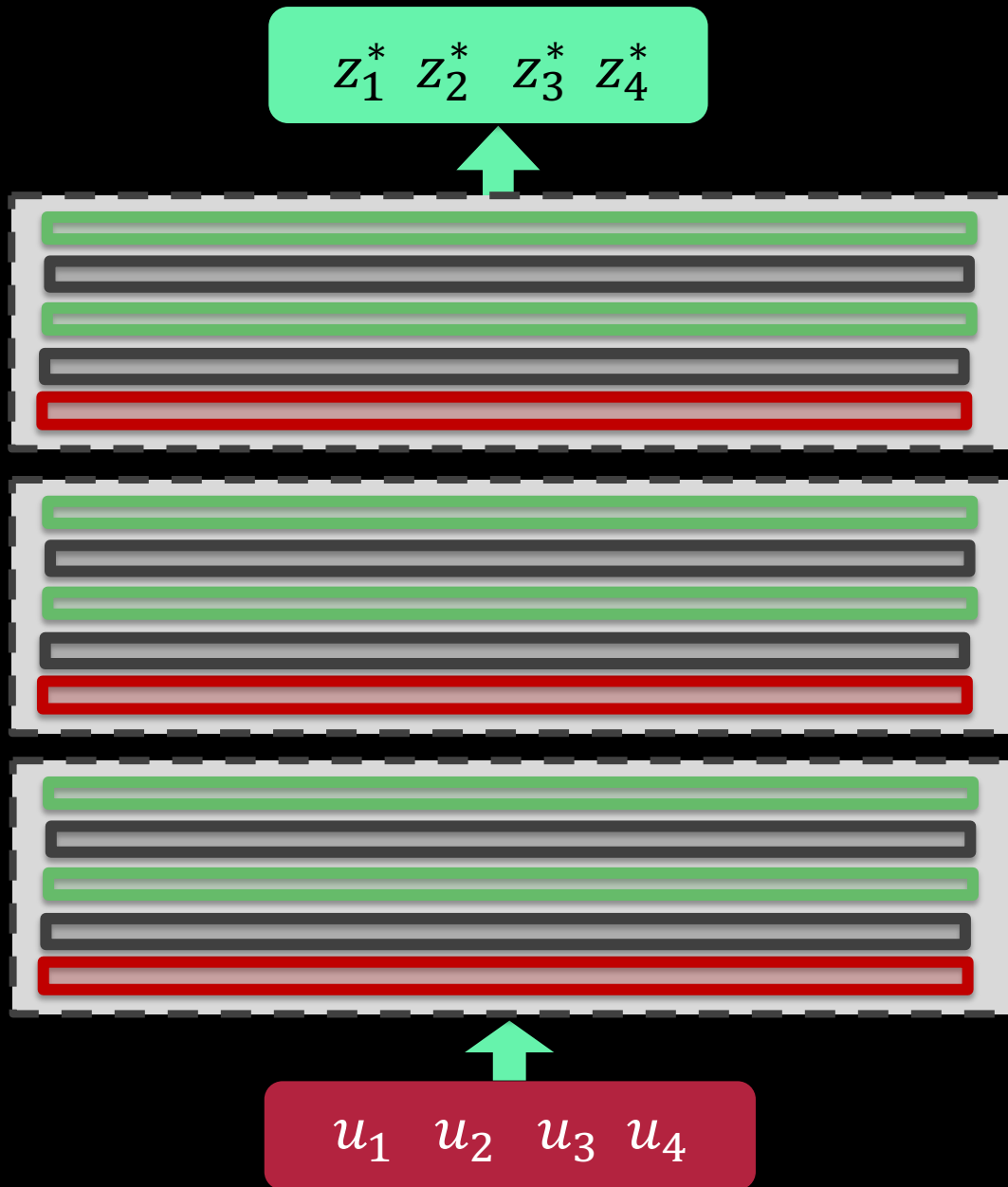
u_1 u_2 u_3 u_4



Why not stack
multiple transformer
layers?



TRANSFORMER
LAYER



Now we can potentially look for richer contextual mappings in a sentence without worrying about vanishing gradients.



Attention

ATTENTION ISSUES?

1. ~~No weights are trained in the process.~~
2. ~~Attention leads to limited contextual mapping.~~
3. There is **no positional information encoded**.



Chester spoke to **Pavlos** about attention.



Attention

ATTENTION ISSUES?

1. ~~No weights are trained in the process.~~
2. ~~Attention leads to limited contextual mapping.~~
3. There is **no positional information encoded**.



Pavlos spoke to **Chester** about attention.



What We Want

LANGUAGE MODEL WISHLIST

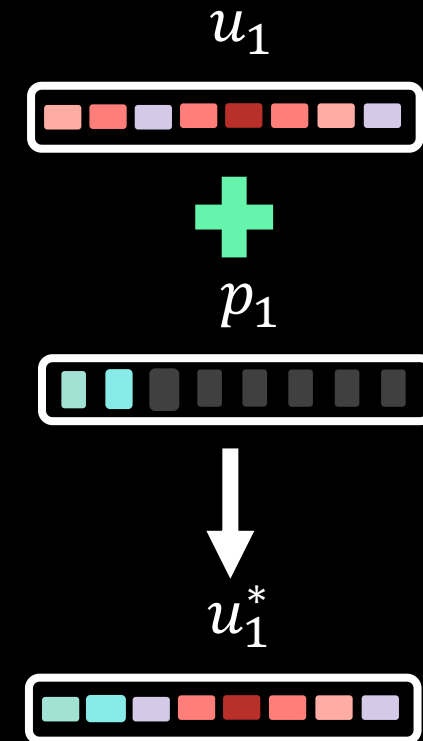
- › **Position** and **order** of words are the essential parts of any language.
- › Recurrent Neural Networks (RNNs) inherently take the **order** of word into account.
- › Multi-head attention blocks do **not** take such an order by design, so there's the need to incorporate the order of the words separately.

**IDEA #255:
Positional Encoding**

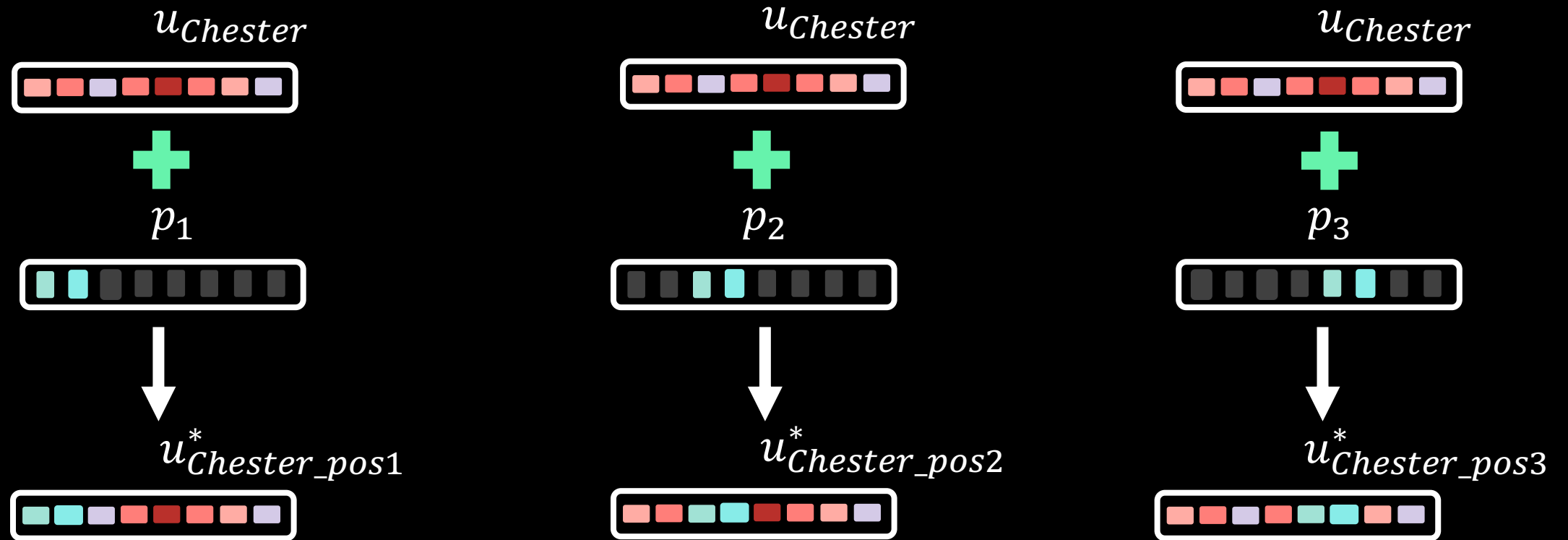


Positional Encoding

- › A very simple way to do this would be to **modify** the input embedding u_i with a **positional vector** p_i that encodes some information about the position of the input embedding.
- › Thus, the same input embedding u_i will have a different value u_i^* depending on the position of the embedding in the sentence.



Positional Encoding

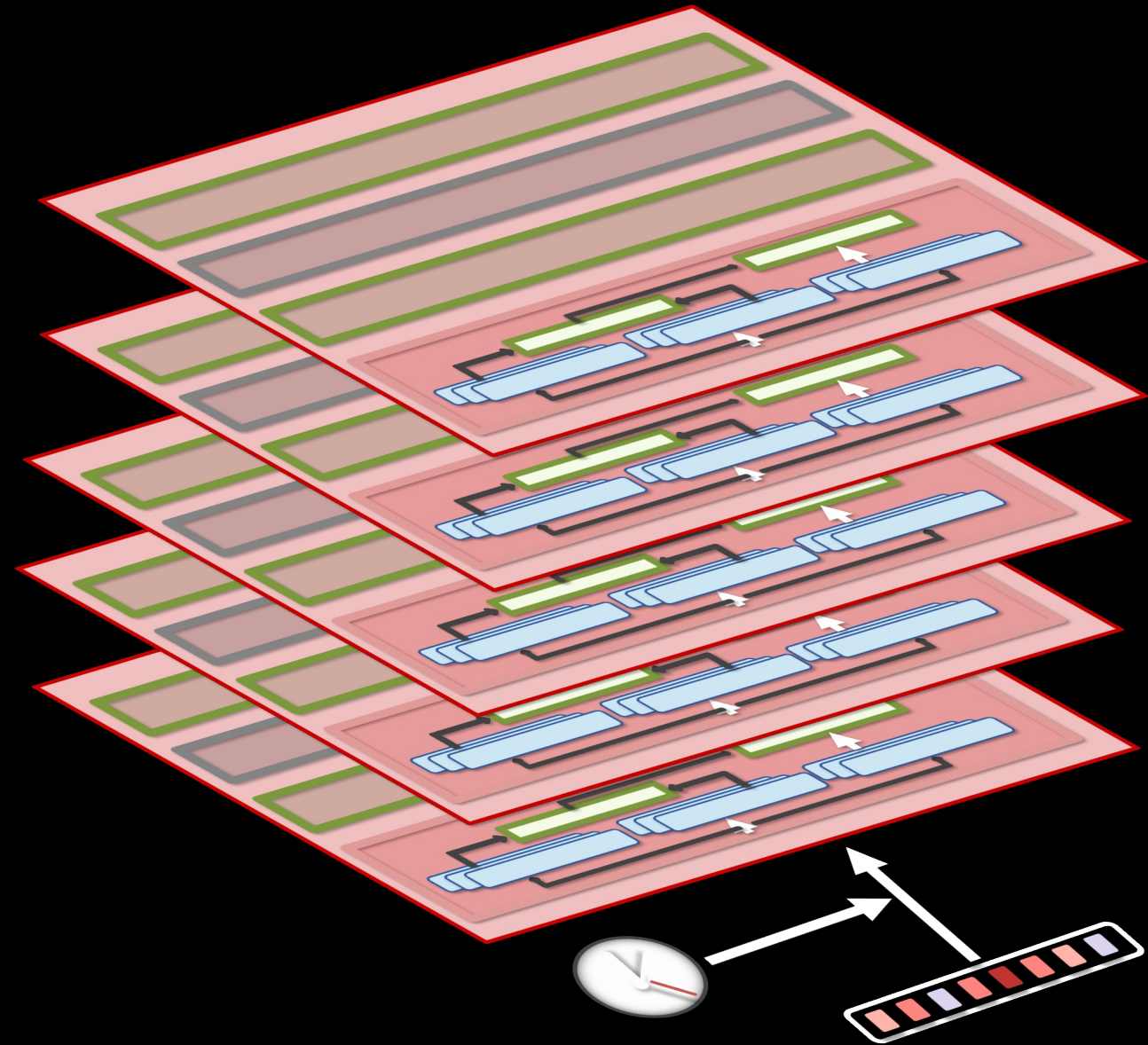


In the above case, $u_{Chester_pos1}^* \neq u_{Chester_pos2}^* \neq u_{Chester_pos3}^*$.

Transformers | Summary

Language Model Wishlist?

- ☑ We want to have strong contextual relations between words - **DONE**
- ☑ We want words to have sequential information - **DONE**
- ☑ We need an architecture that can be trained in parallel (non-Markovian property) - **DONE**



More on this
in Part 4!

