Seq2seq and Attention

Lena Voita

What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
    - Idea
    - Self-Attention
    - Masked Self-Attention
    - Multi-Head Attention
    - Model Architecture
- Subword Segmentation: BPE
- Analysis and Interpretability

Idea: Attention is All You Need

	Seq2seq without attention	Seq2seq with attention
processing within encoder	RNN/CNN	RNN/CNN
processing within decoder	RNN/CNN	RNN/CNN
decoder-encoder interaction	static fixed-sized vector	attention

Idea: Attention is All You Need

	Seq2seq without attention	Seq2seq with attention	Transformer
processing within encoder	RNN/CNN	RNN/CNN	attention
processing within decoder	RNN/CNN	RNN/CNN	attention
decoder-encoder interaction	static fixed-sized vector	attention	attention

Idea: Attention is All You Need

The animation is from the Google AI blog post.

Idea: Attention is All You Need

Encoder

Who is doing:

What they are doing:

The animation is from the Google AI blog post.

Idea: Attention is All You Need

Encoder

Who is doing:

- all source tokens

What they are doing:

The animation is from the Google AI blog post.

Idea: Attention is All You Need

Encoder

Who is doing:

- all source tokens

What they are doing:

- look at each other
- update representations

repeat N times

The animation is from the Google AI blog post.

Idea: Attention is All You Need

Decoder

Who is doing:

What they are doing:

The animation is from the Google AI blog post.

Idea: Attention is All You Need

Decoder

Who is doing:

- target token at the current step

What they are doing:

The animation is from the Google AI blog post.

Idea: Attention is All You Need

Decoder

Who is doing:

- target token at the current step

What they are doing:

- looks at previous target tokens
- looks at source representations
- update representation

repeat
N times

The animation is from the Google AI blog post.

Why can this be better than RNNs?

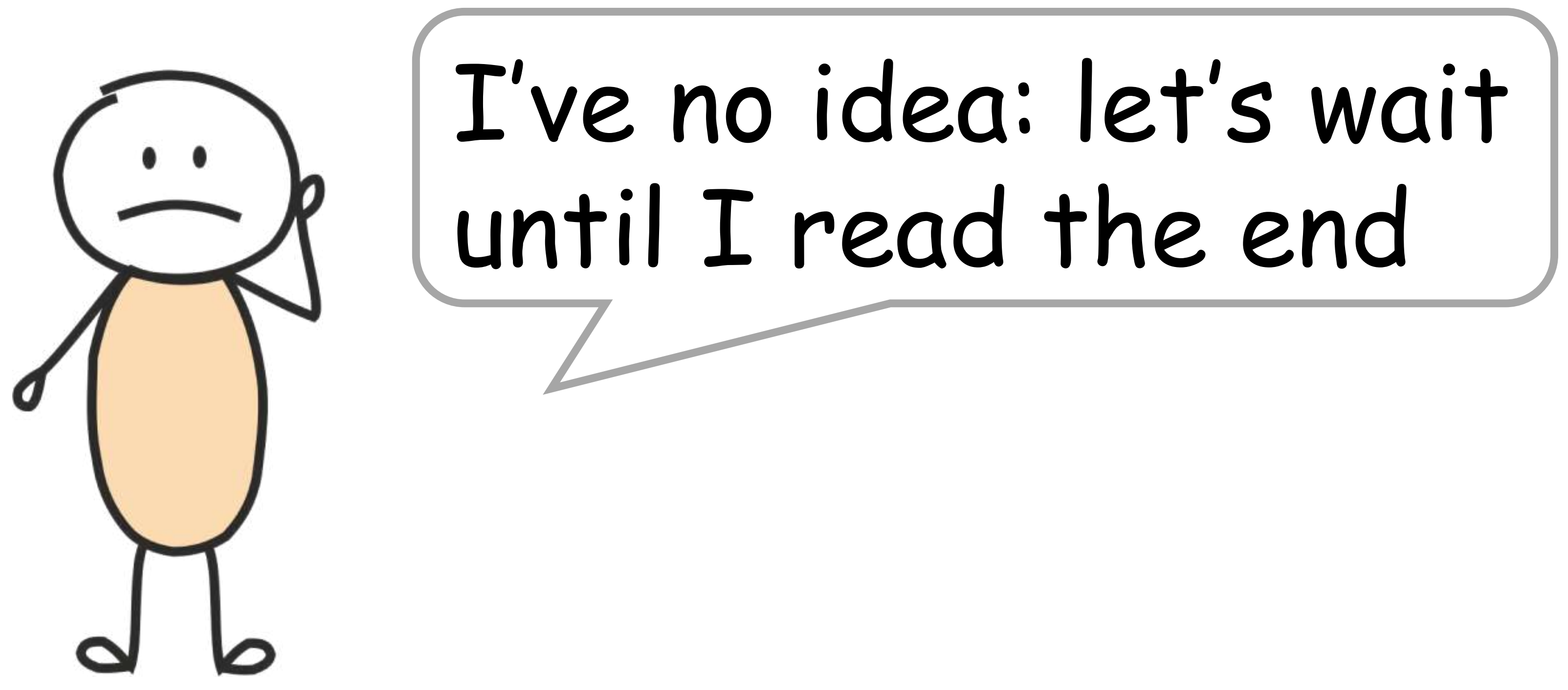
I arrived at the bank after crossing the … ...street? ...river?

What does **bank** mean in this sentence?

Why can this be better than RNNs?

I arrived at the bank after crossing the …   …street?   …river?

What does **bank** mean in this sentence?



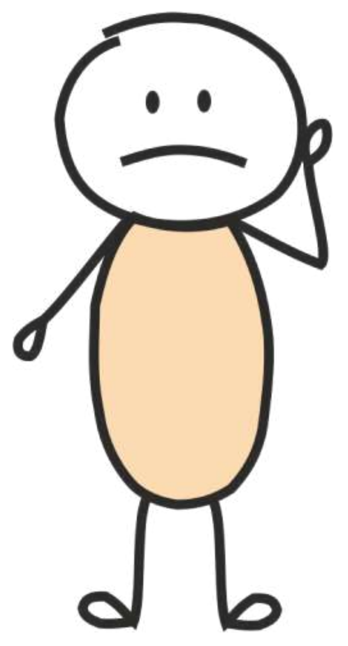
RNNs

O(N) steps to process a sentence with length N

Why can this be better than RNNs?

I arrived at the bank after crossing the ... ...street? ...river?

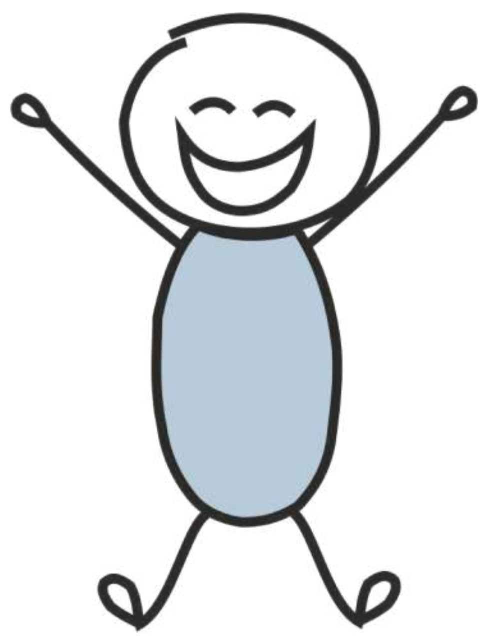
What does **bank** mean in this sentence?



I've no idea: let's wait until I read the end

RNNs

O(N) steps to process a sentence with length N



I don't need to wait - I see all words at once!

Transformer

Constant number of steps to process any sentence

What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
  - Idea
  - Self-Attention
  - Masked Self-Attention
  - Multi-Head Attention
  - Model Architecture
- Subword Segmentation: BPE
-  Analysis and Interpretability

What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
  - Idea
  - Self-Attention
  - Masked Self-Attention
  - Multi-Head Attention
  - Model Architecture
- Subword Segmentation: BPE
-  Analysis and Interpretability

Self-Attention: Why "Self"?

Decoder-encoder attention is looking

- from: one current decoder state
- at: all encoder states

Self-Attention: Why "Self"?

Decoder-encoder attention is looking

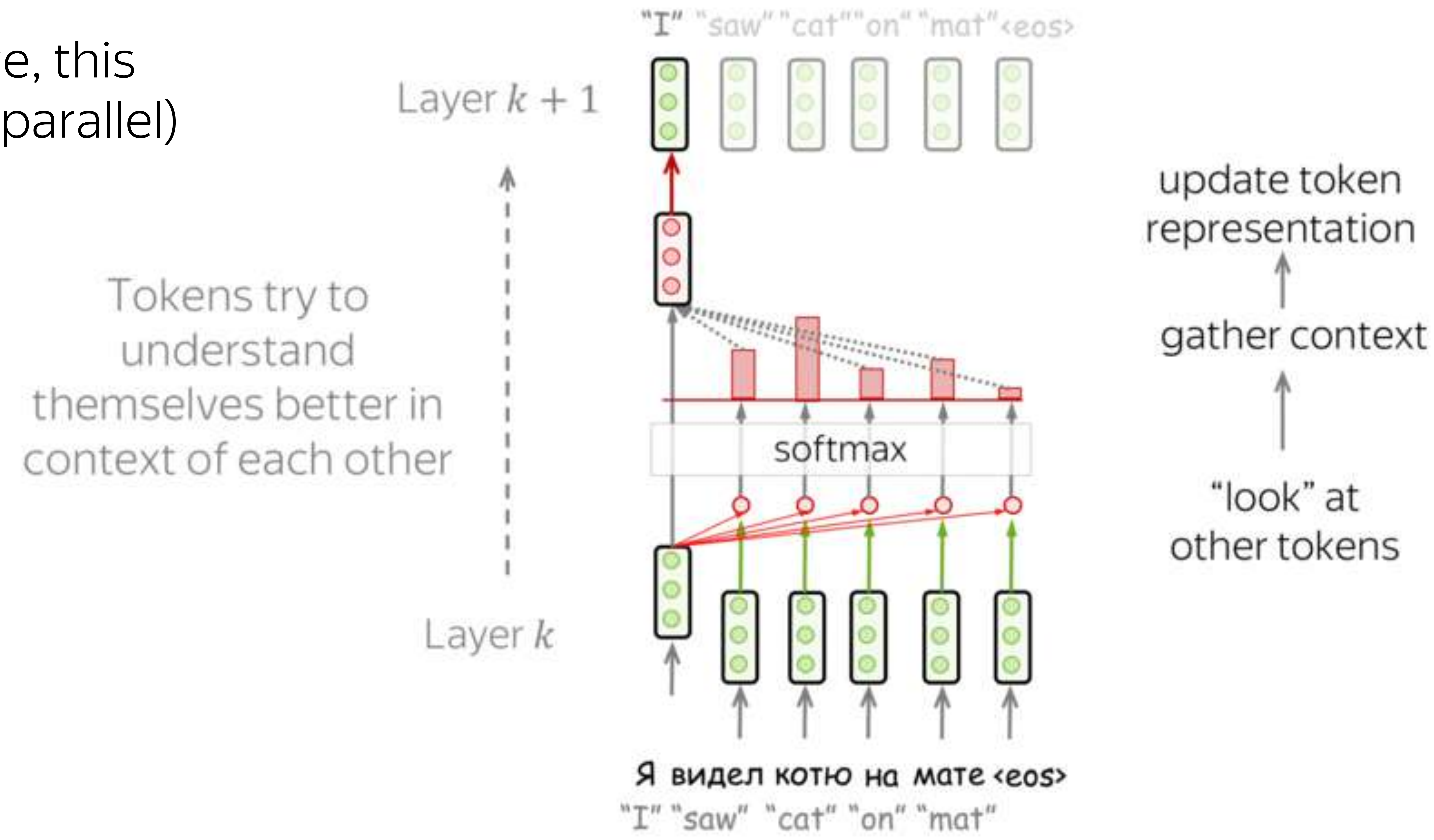
- **from**: one current decoder state
- **at**: all encoder states

Self-attention is looking

- **from**: each state from a set of states
- **at**: all other states in the same set

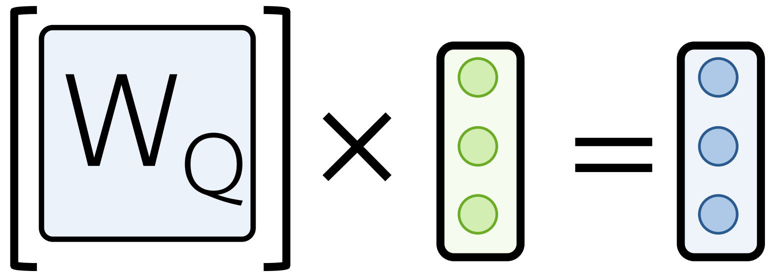
Self-Attention: The “Look at Each Other” Part

(in practice, this happens in parallel)



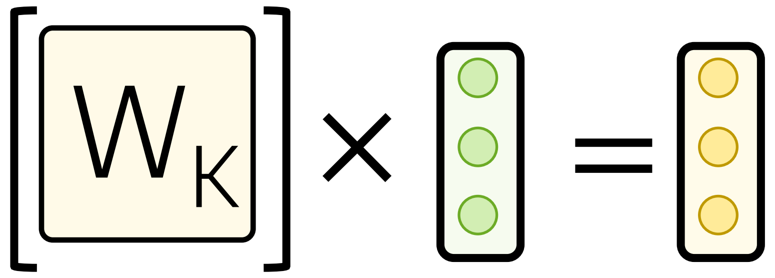
Query, Key, Value

Each vector receives three representations ("roles")



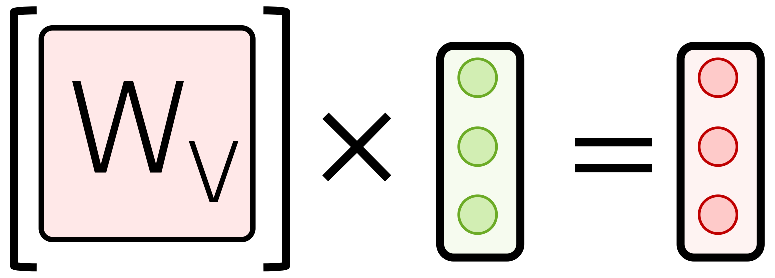
Query: vector **from** which the attention is looking

"Hey there, do you have this information?"



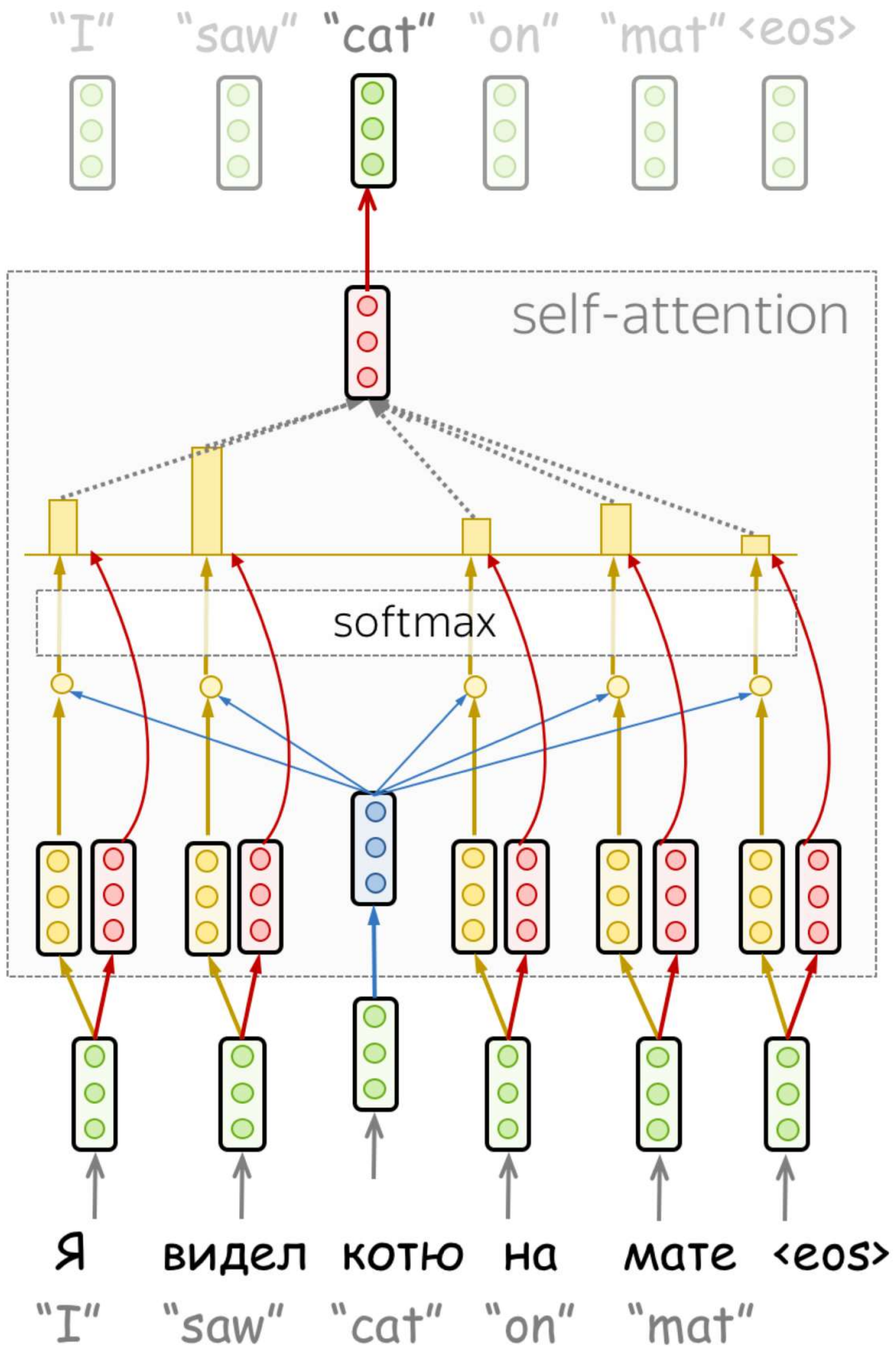
Key: vector **at** which the query looks to compute weights

"Hi, I have this information – give me a large weight!"

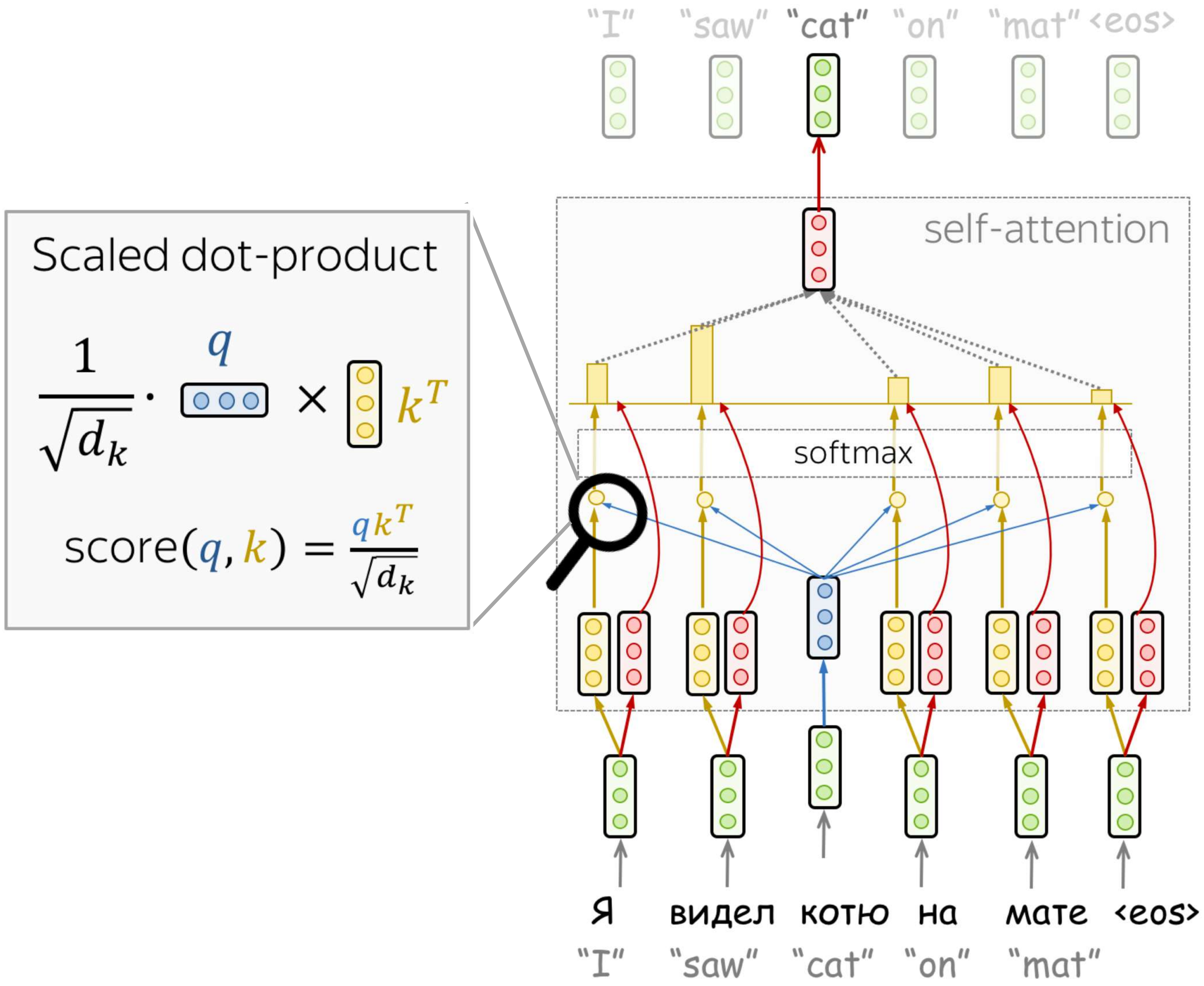


Value: their weighted sum is attention output

"Here's the information I have!"



Query, Key, Value

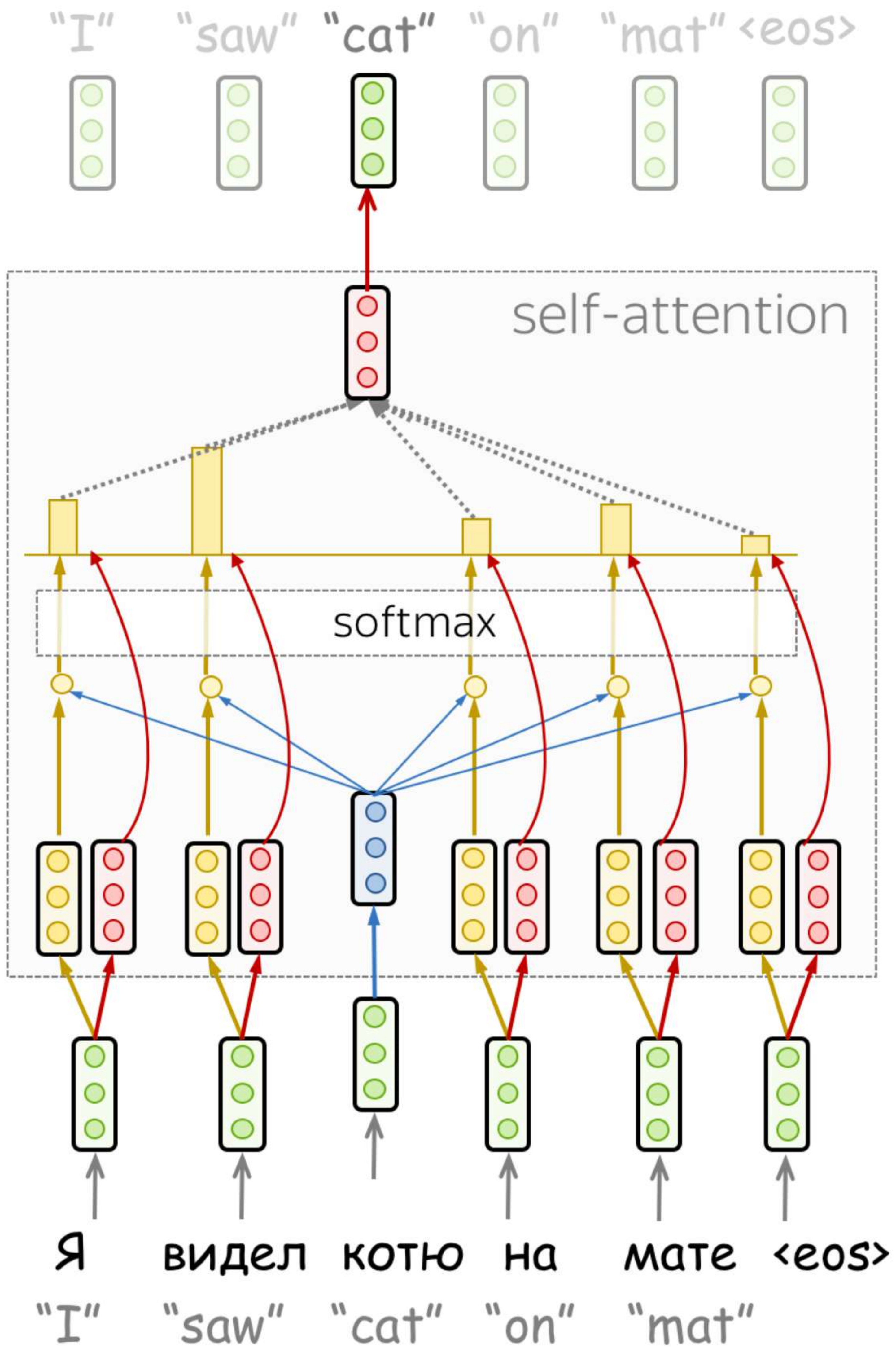


Query, Key, Value

$$Attention(q, k, v) = \overbrace{softmax\left(\frac{qk^T}{\sqrt{d_k}}\right)}^{\text{Attention weights}} v$$

from to

vector dimensionality of K, V



What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
    - Idea
    - Self-Attention
    - Masked Self-Attention
    - Multi-Head Attention
    - Model Architecture
- Subword Segmentation: BPE
- Analysis and Interpretability

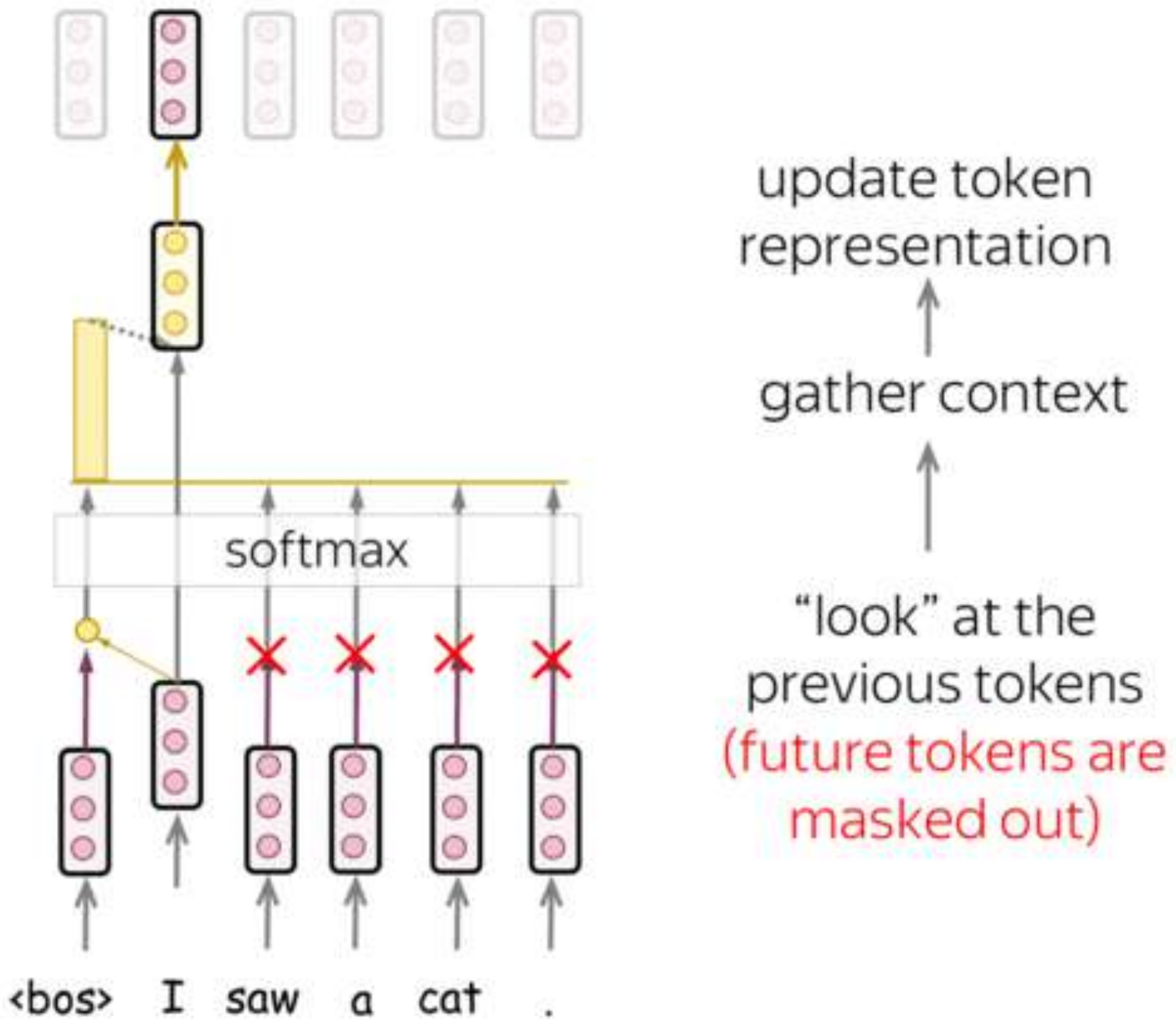
What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
    - Idea
    - Self-Attention
    - Masked Self-Attention
    - Multi-Head Attention
    - Model Architecture
- Subword Segmentation: BPE
- Analysis and Interpretability

Masked Self-Attention: “Don’t Look Ahead”

In the decoder, we forbid looking at future tokens – we don’t know them

Note: in training, decoder processes all target tokens at once – without masks, it would see future



What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
- Subword Segmentation: BPE
-  Analysis and Interpretability

  - Idea
  - Self-Attention
  - Masked Self-Attention
  - Multi-Head Attention
  - Model Architecture

What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
  - Idea
  - Self-Attention
  - Masked Self-Attention
  - Multi-Head Attention
  - Model Architecture
- Subword Segmentation: BPE
- Analysis and Interpretability

Multi-Head Attention

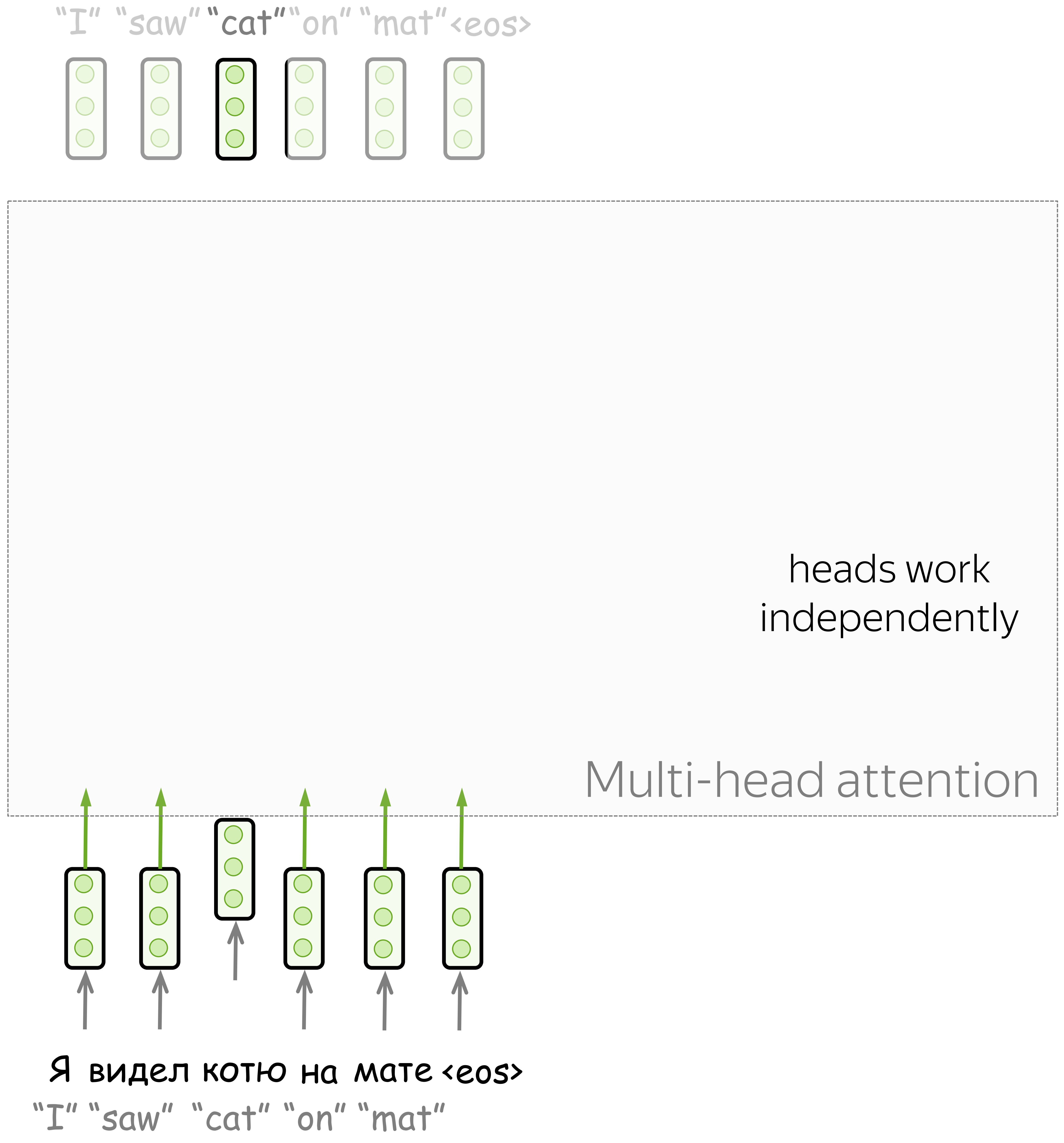
Она руководит **новым** проектом

We need to track many different things at once!

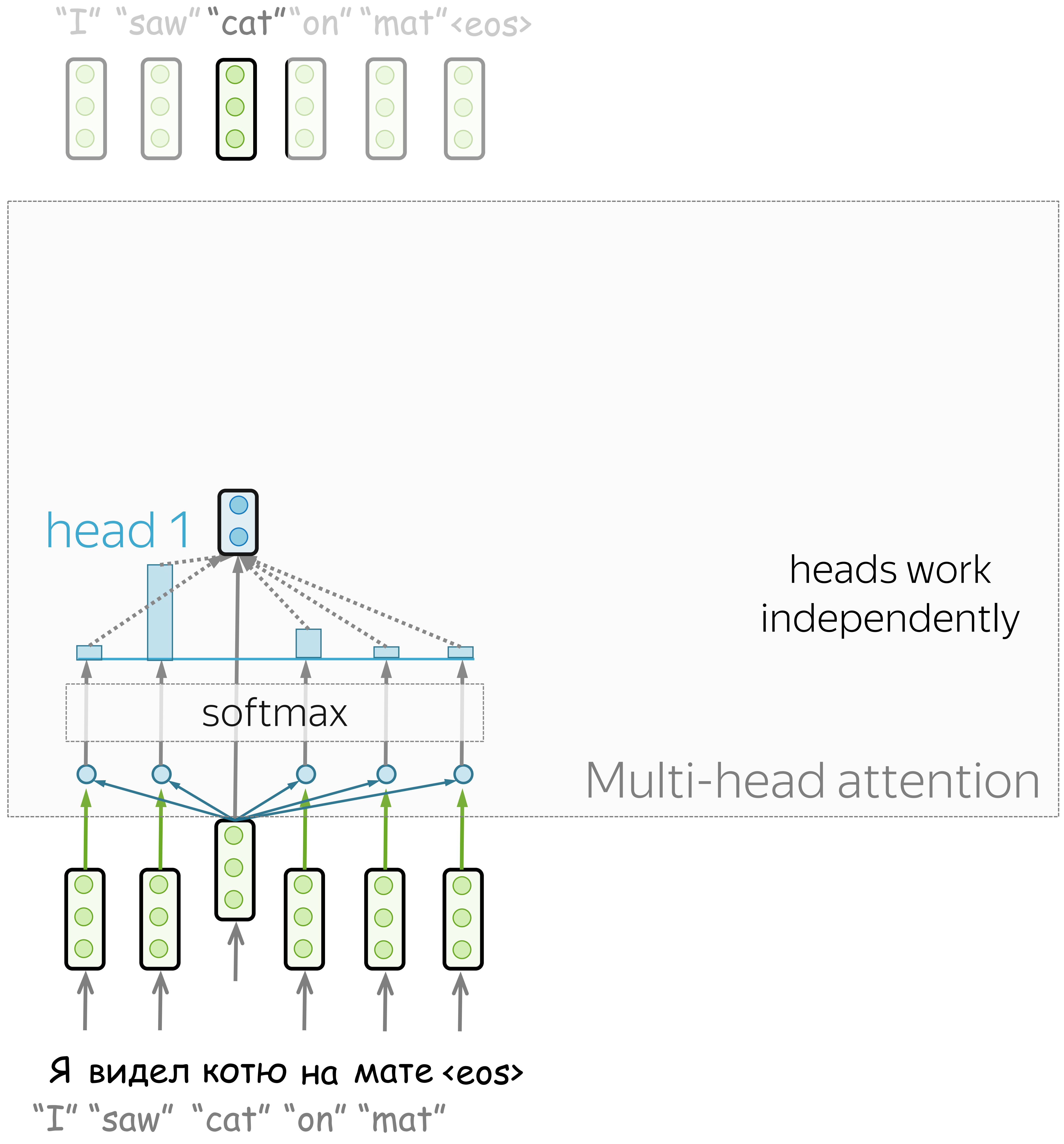
- Gender agreement
- Case government
- Lexical preferences
- ...

The example is from David Talbot

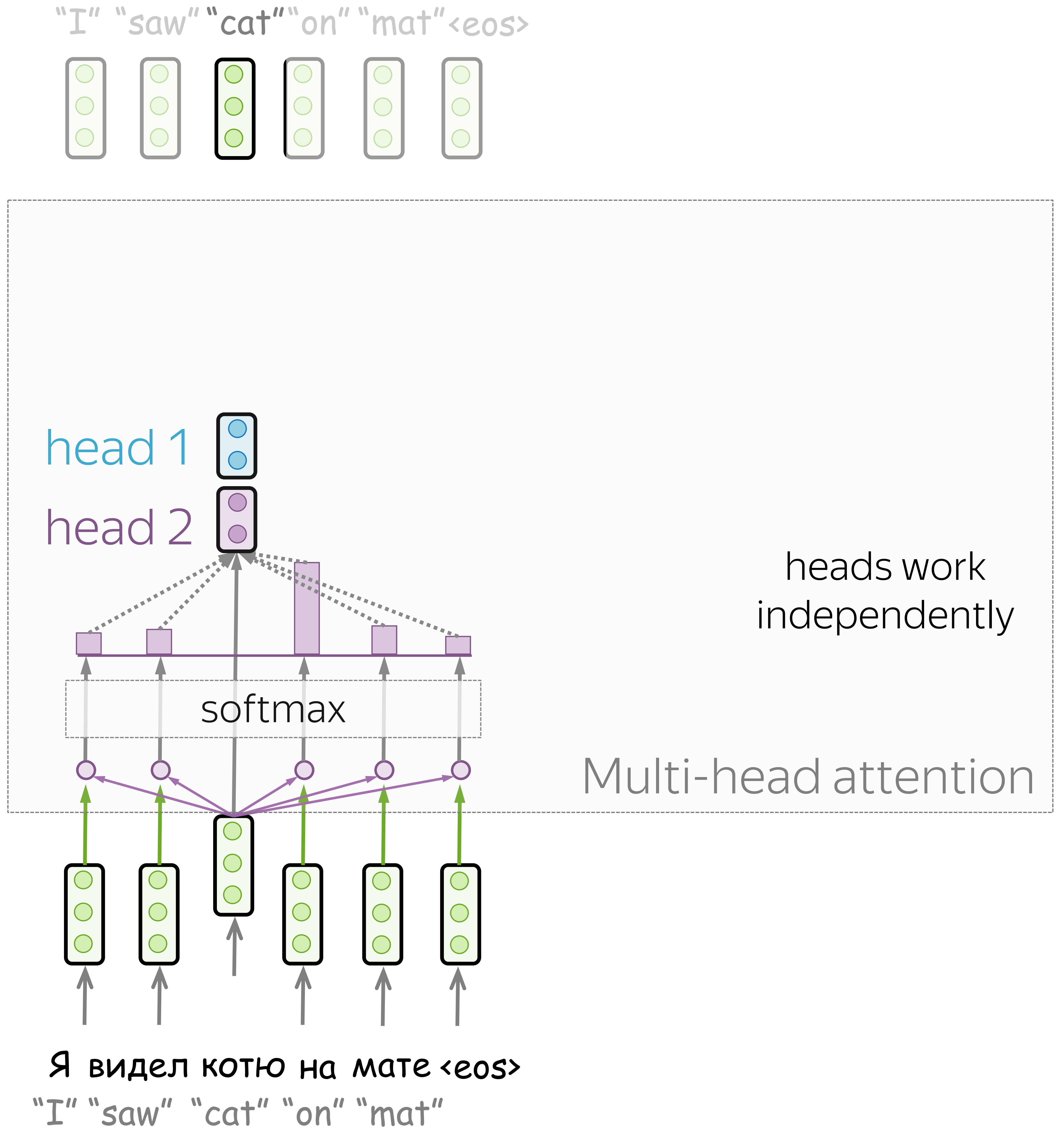
Multi-Head Attention



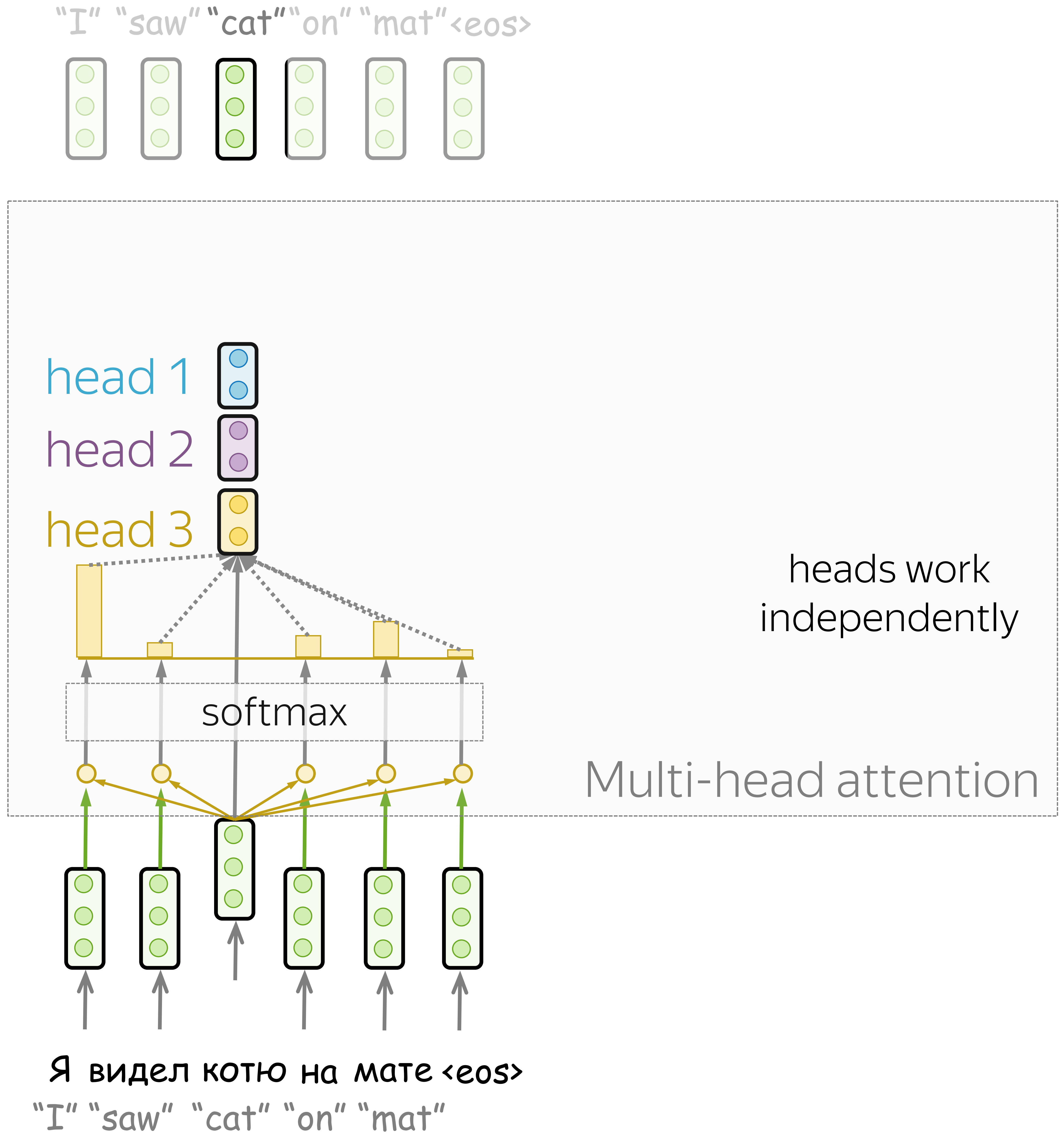
Multi-Head Attention



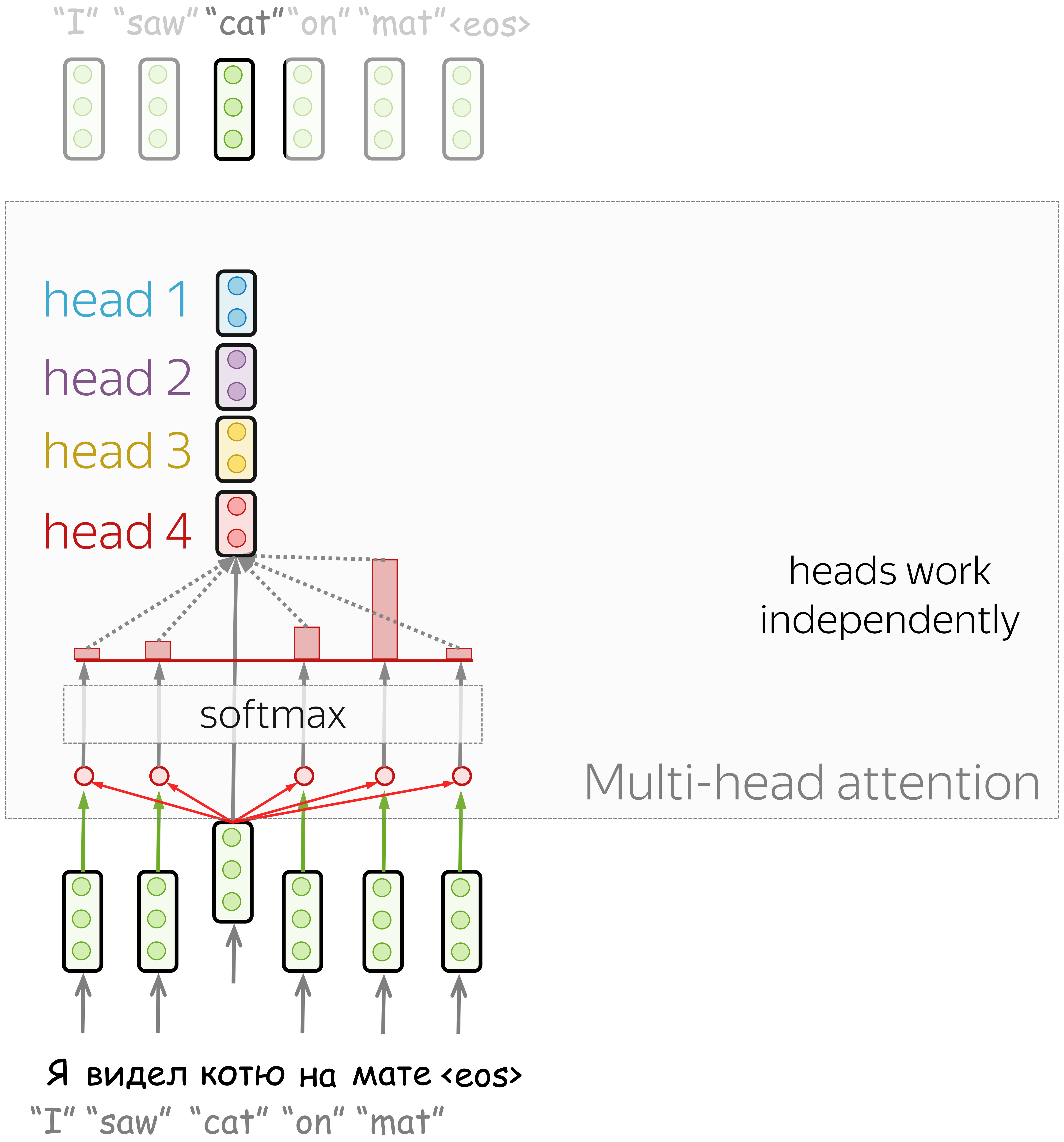
Multi-Head Attention



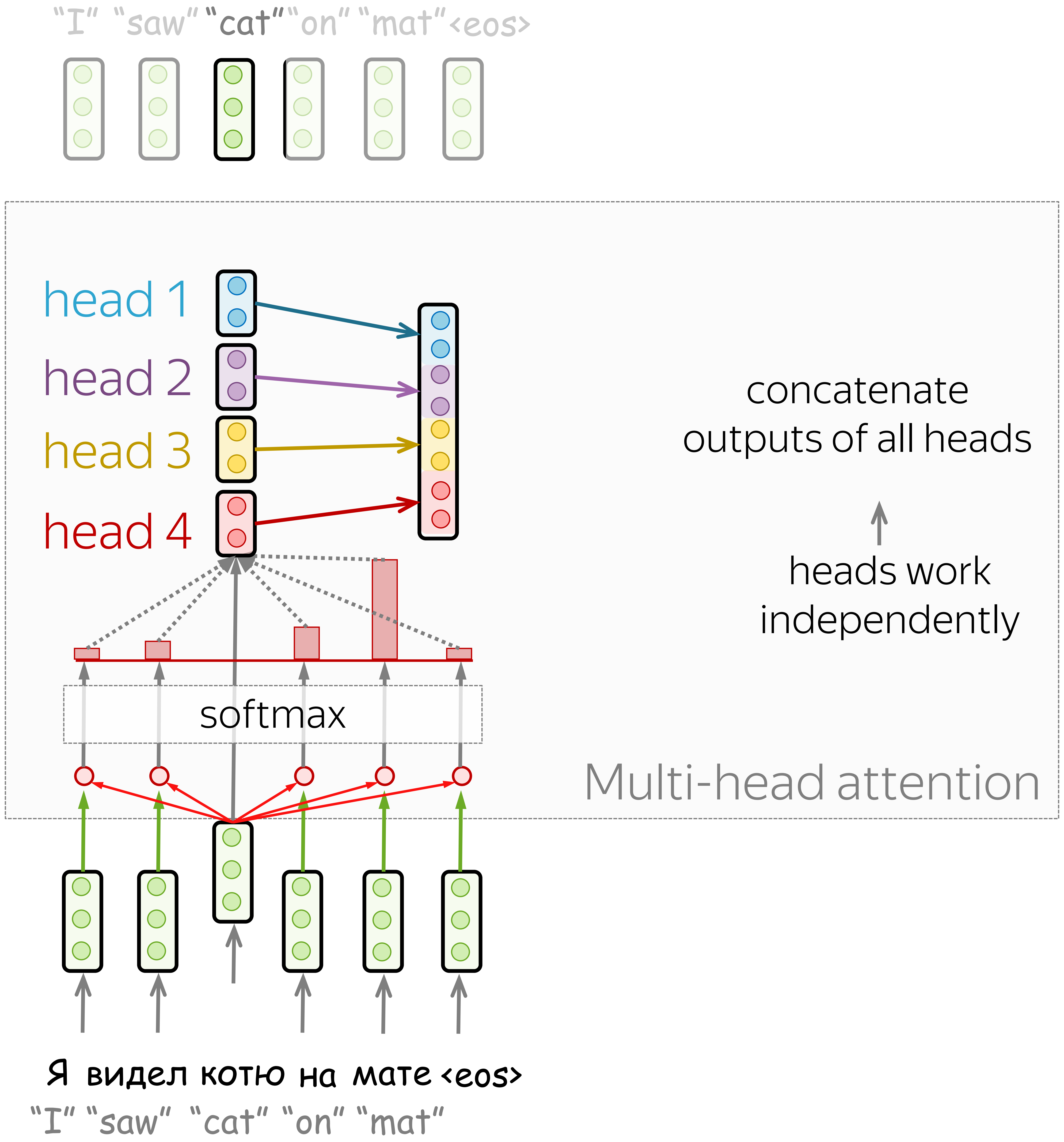
Multi-Head Attention



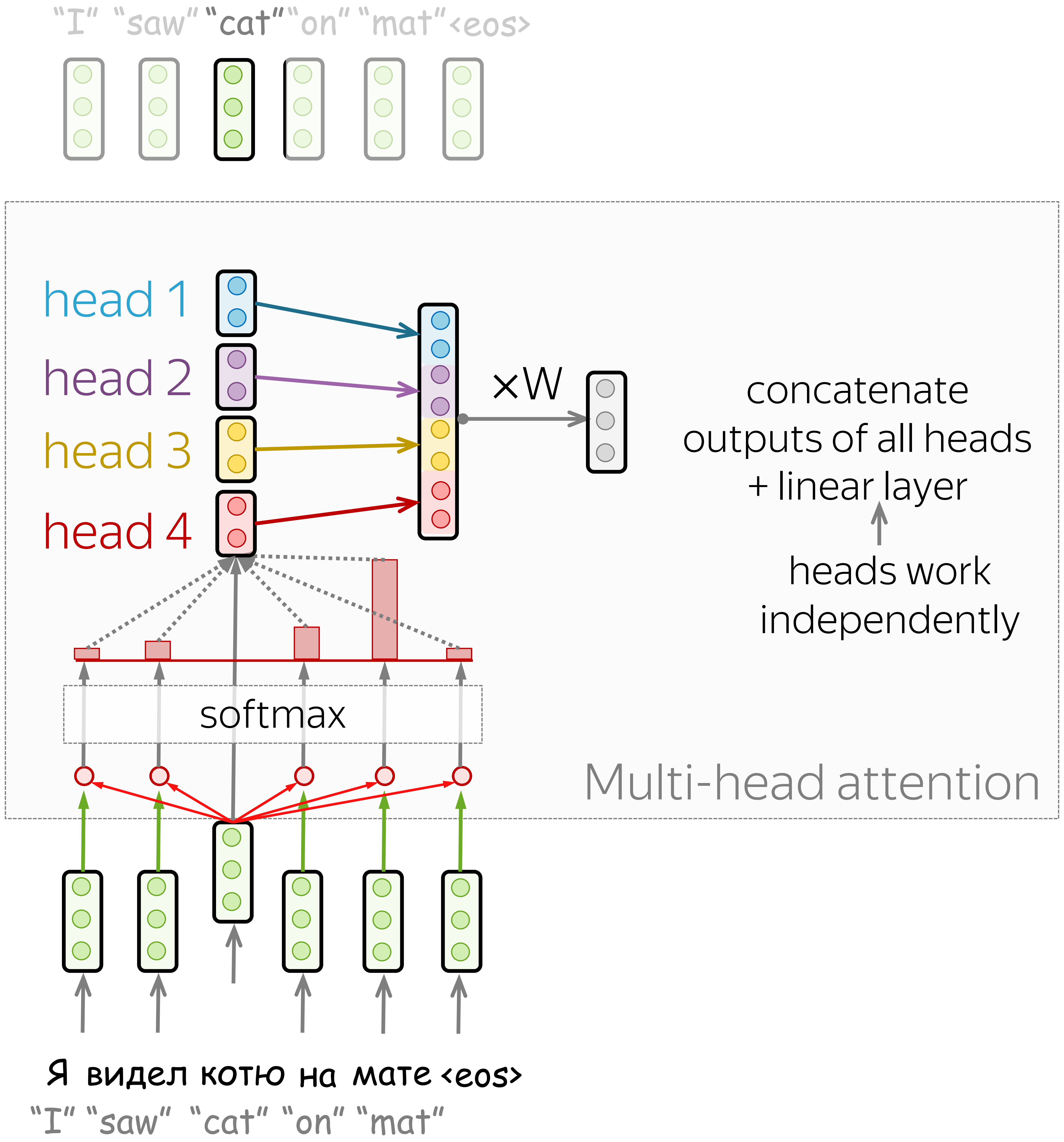
Multi-Head Attention



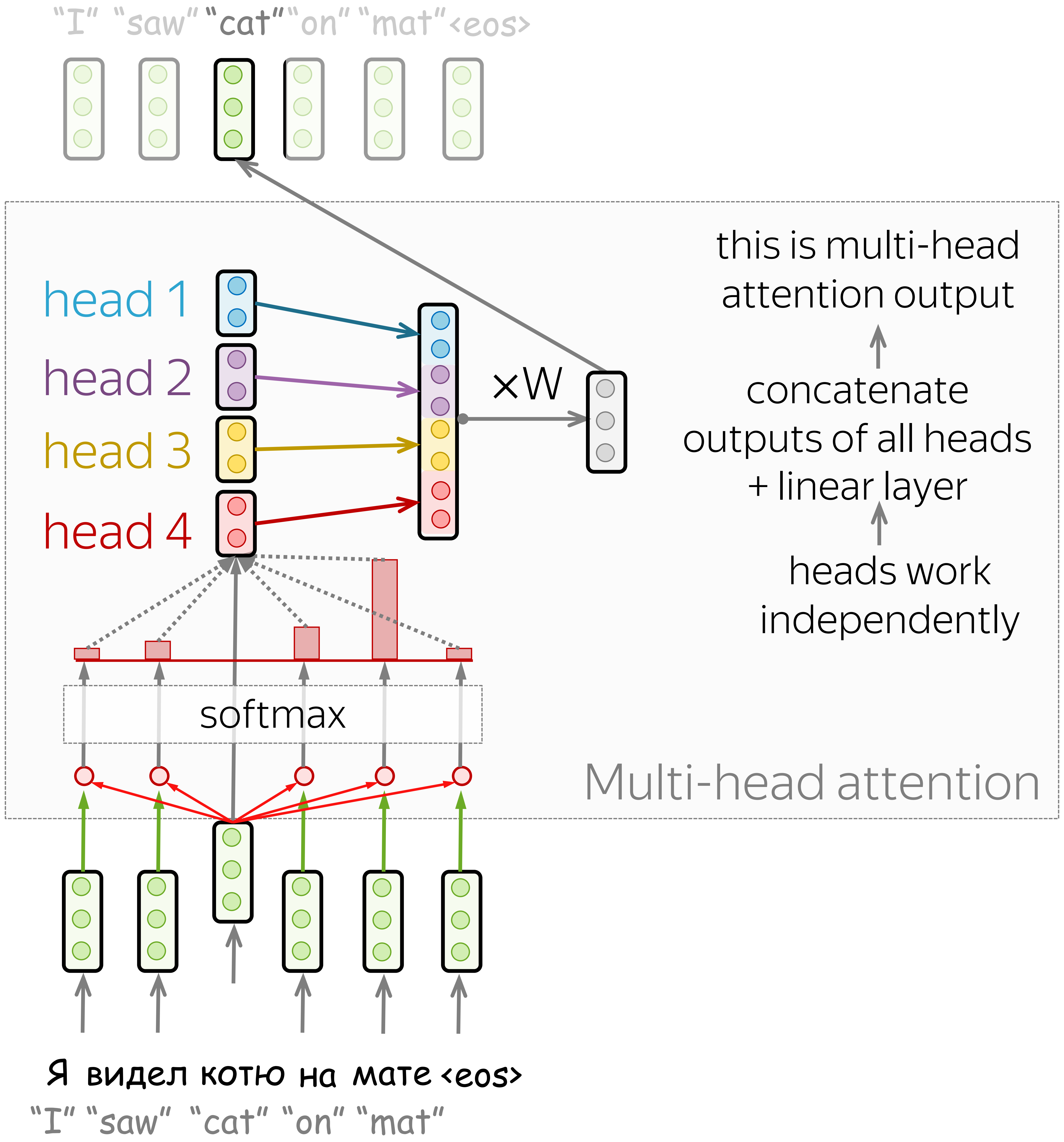
Multi-Head Attention



Multi-Head Attention



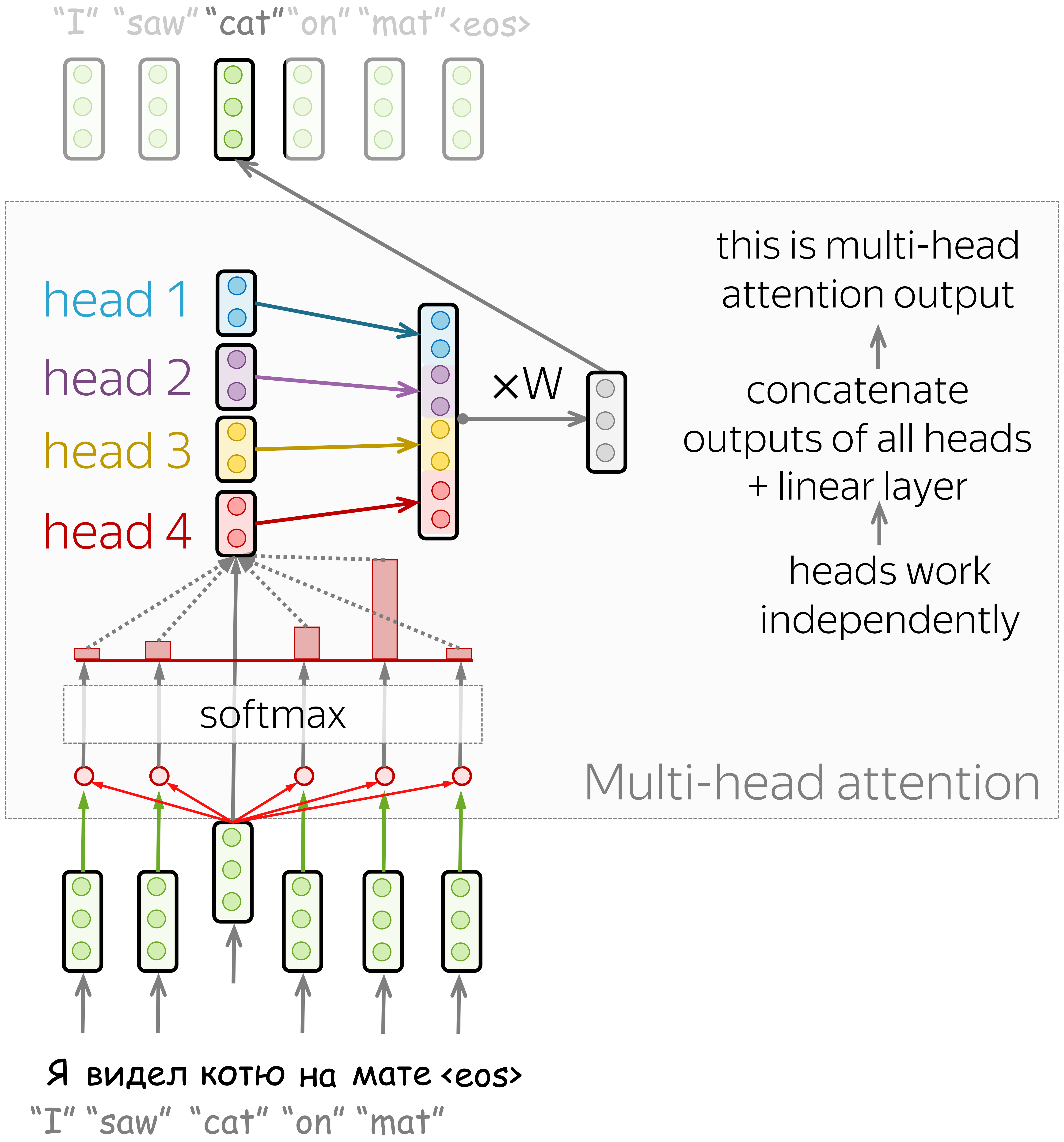
Multi-Head Attention



Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_n)W_o,$$

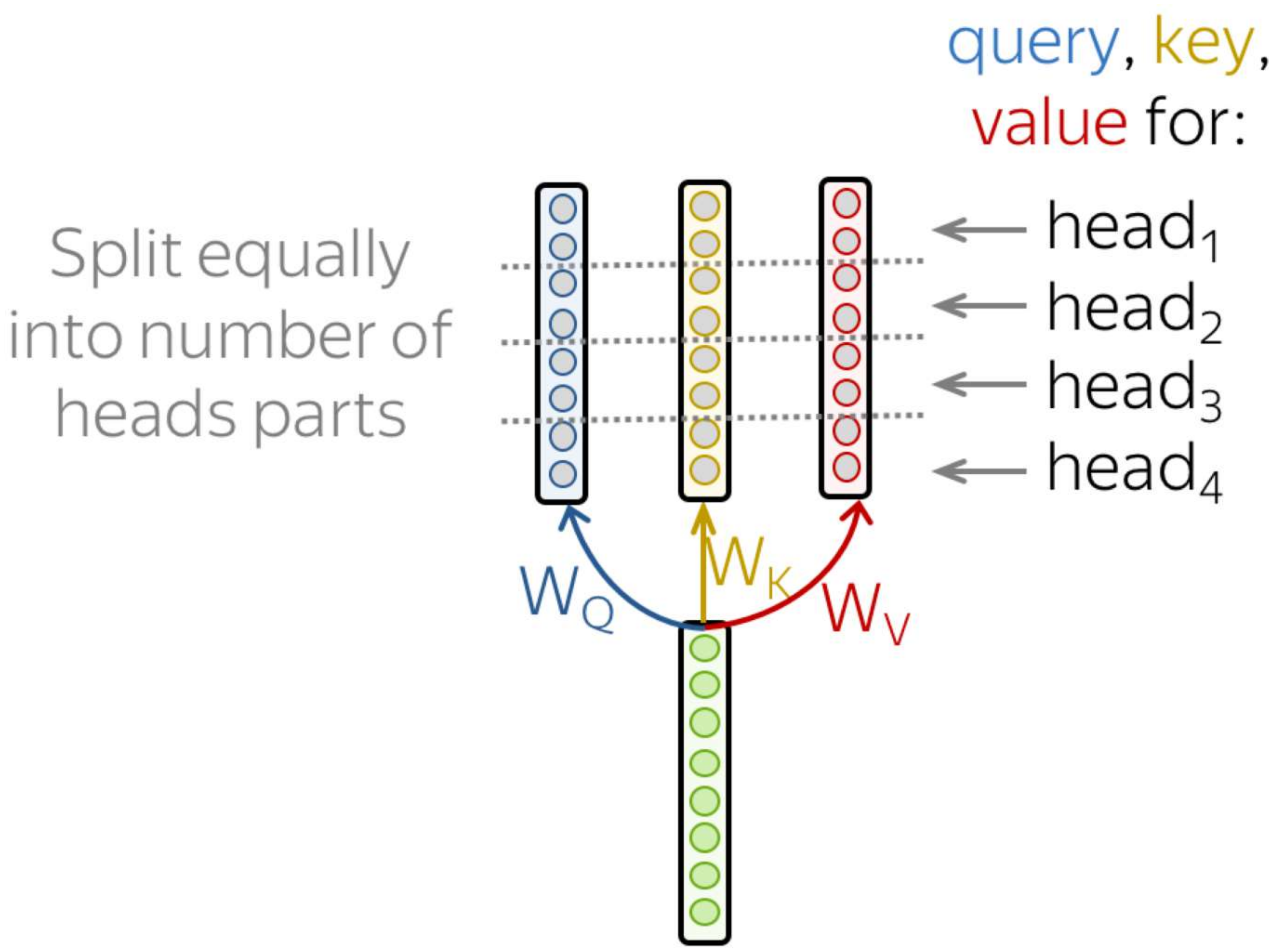
$$\text{head}_i = \text{Attention}(QW_Q^i, KW_K^i, VW_V^i)$$

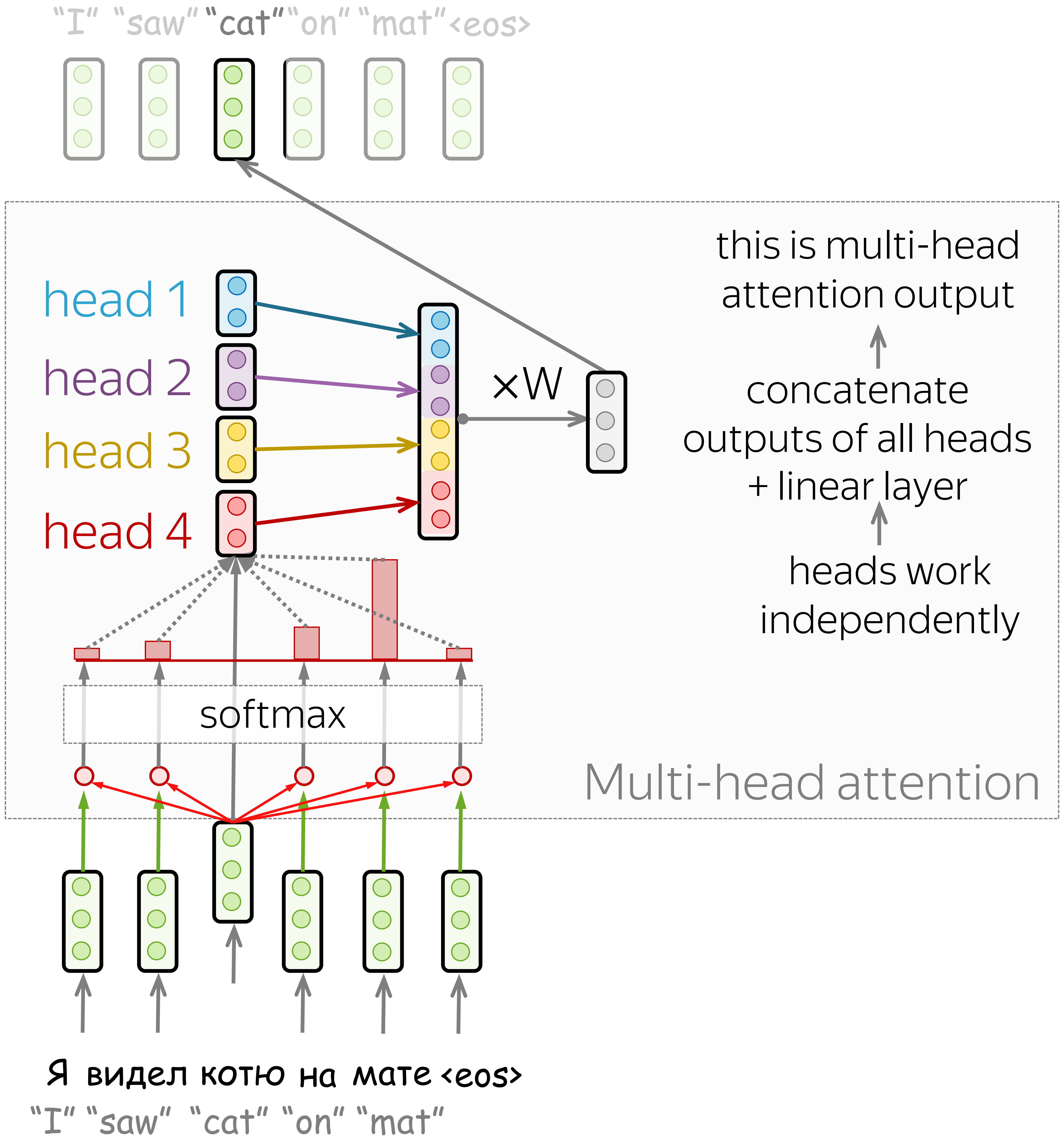


Multi-Head Attention

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_n) W_o,$$

$$\mathrm{head}_i = \mathrm{Attention}(QW_Q^i, KW_K^i, VW_V^i)$$





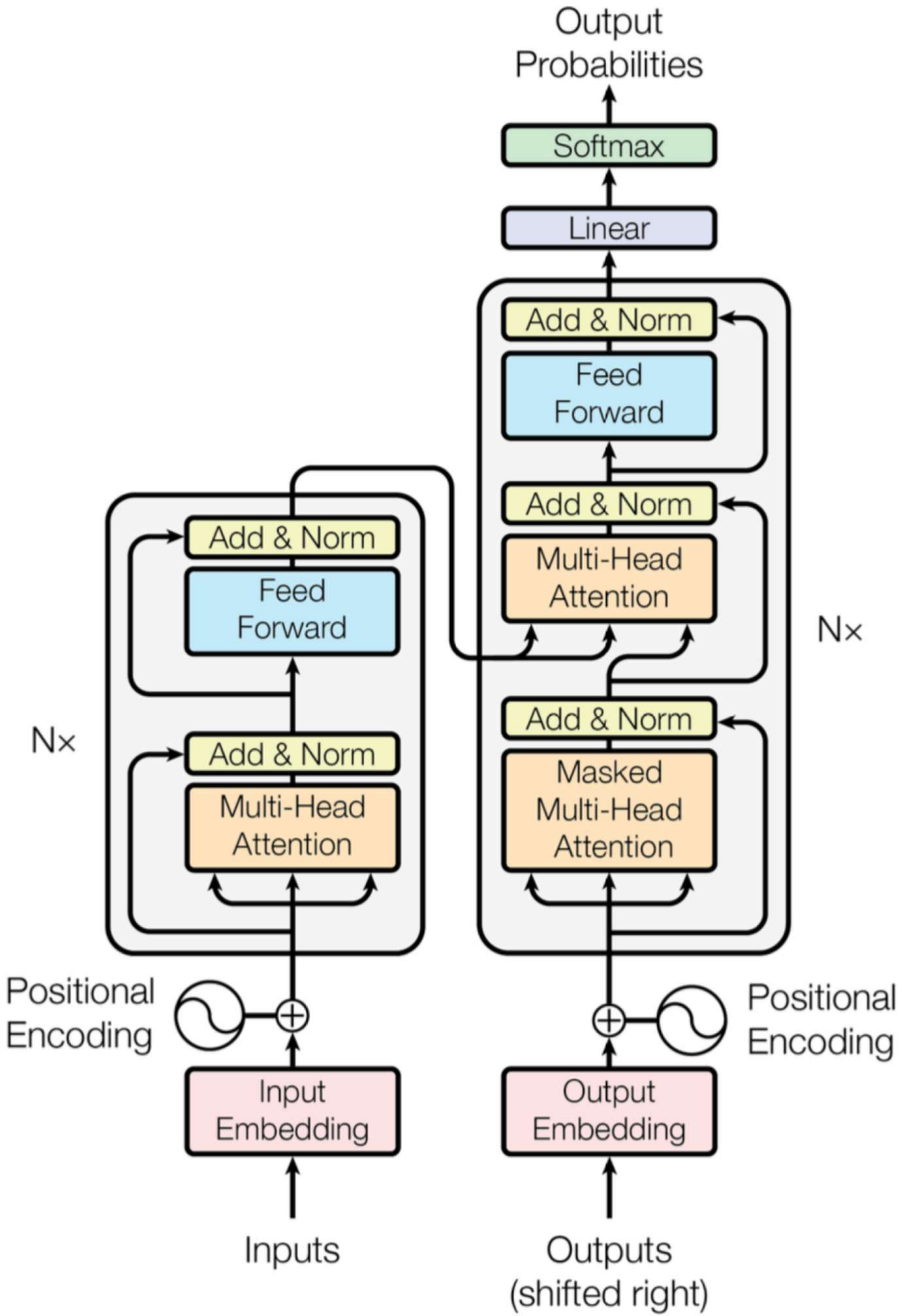
What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
    - Idea
    - Self-Attention
    - Masked Self-Attention
    - Multi-Head Attention
    - Model Architecture
- Subword Segmentation: BPE
- Analysis and Interpretability

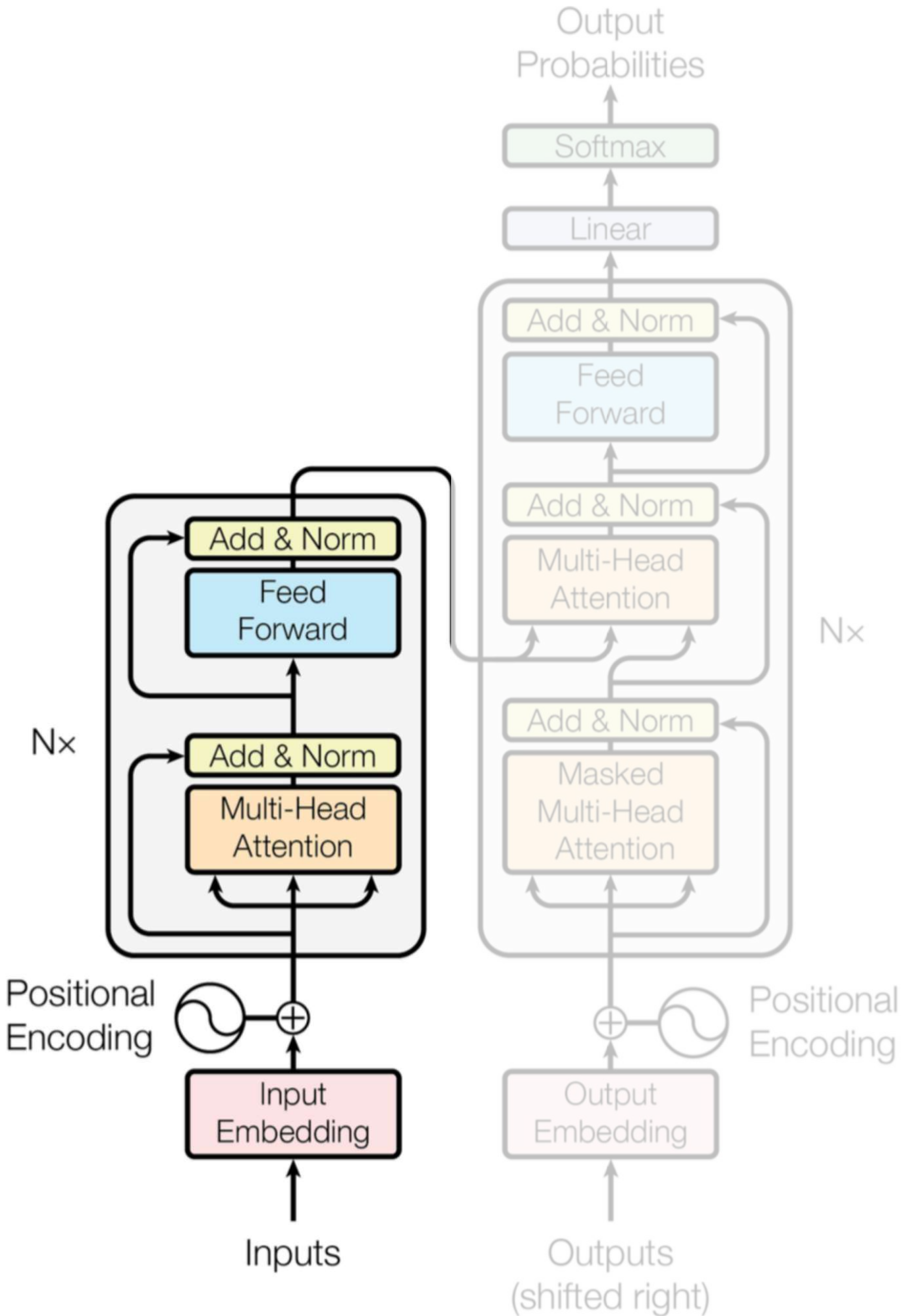
What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
  - Idea
  - Self-Attention
  - Masked Self-Attention
  - Multi-Head Attention
  - Model Architecture
- Subword Segmentation: BPE
-  Analysis and Interpretability

Transformer



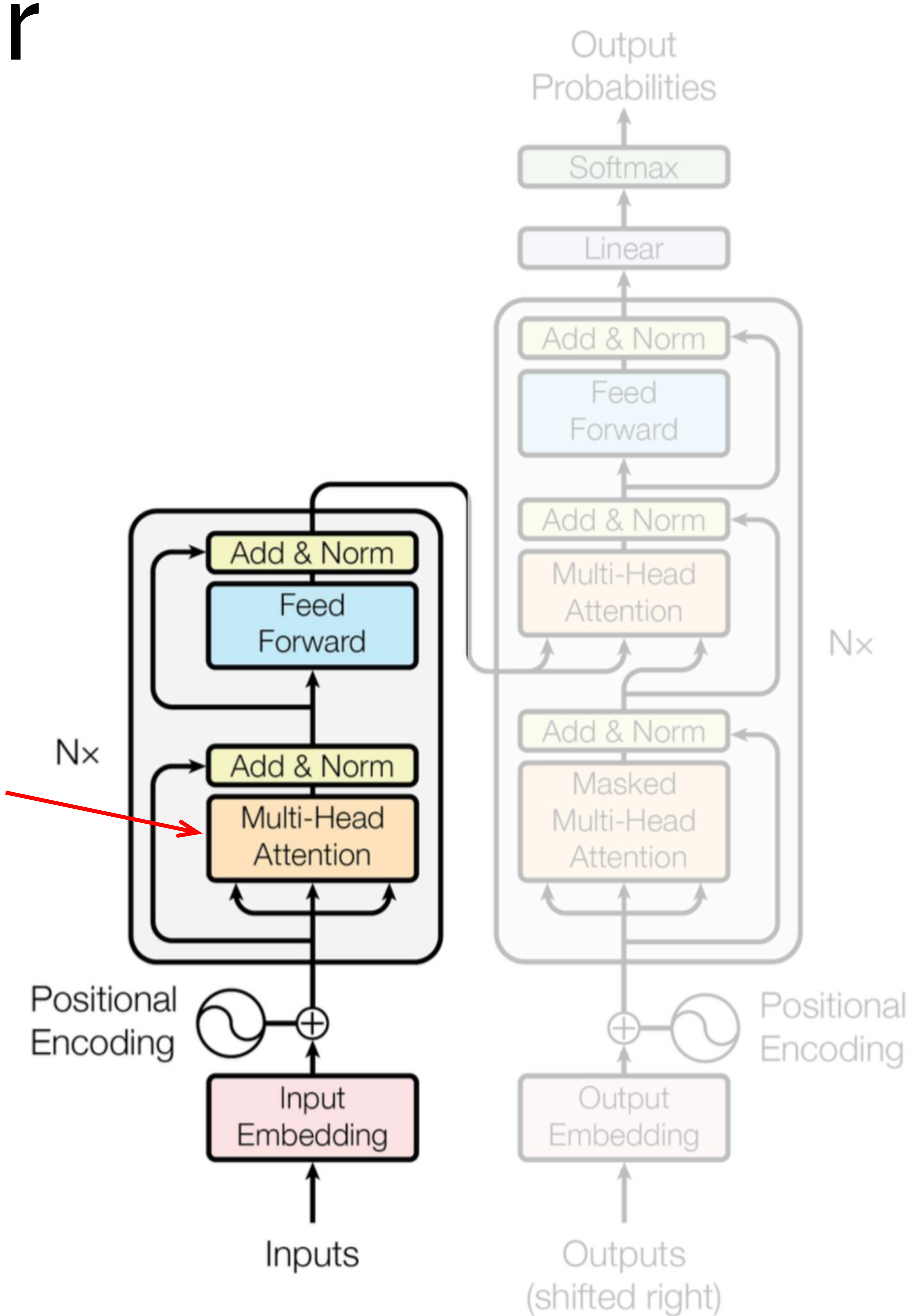
Transformer



Transformer

Encoder self-attention:
tokens look at each other

queries, keys, values
are computed from
encoder states

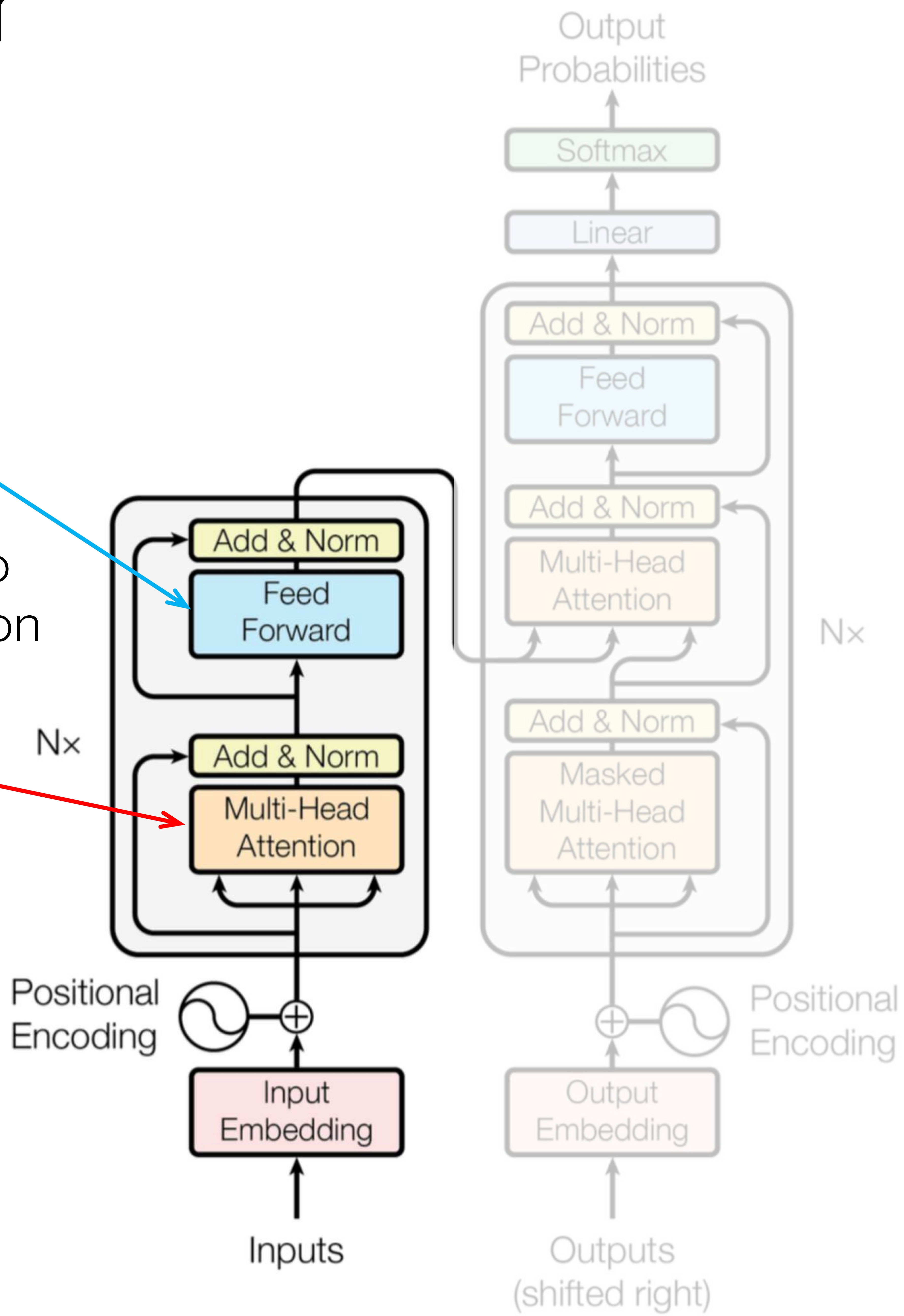


Transformer

Feed-forward network: after taking information from other tokens, take a moment to think and process this information

Encoder self-attention: tokens look at each other

queries, keys, values are computed from encoder states

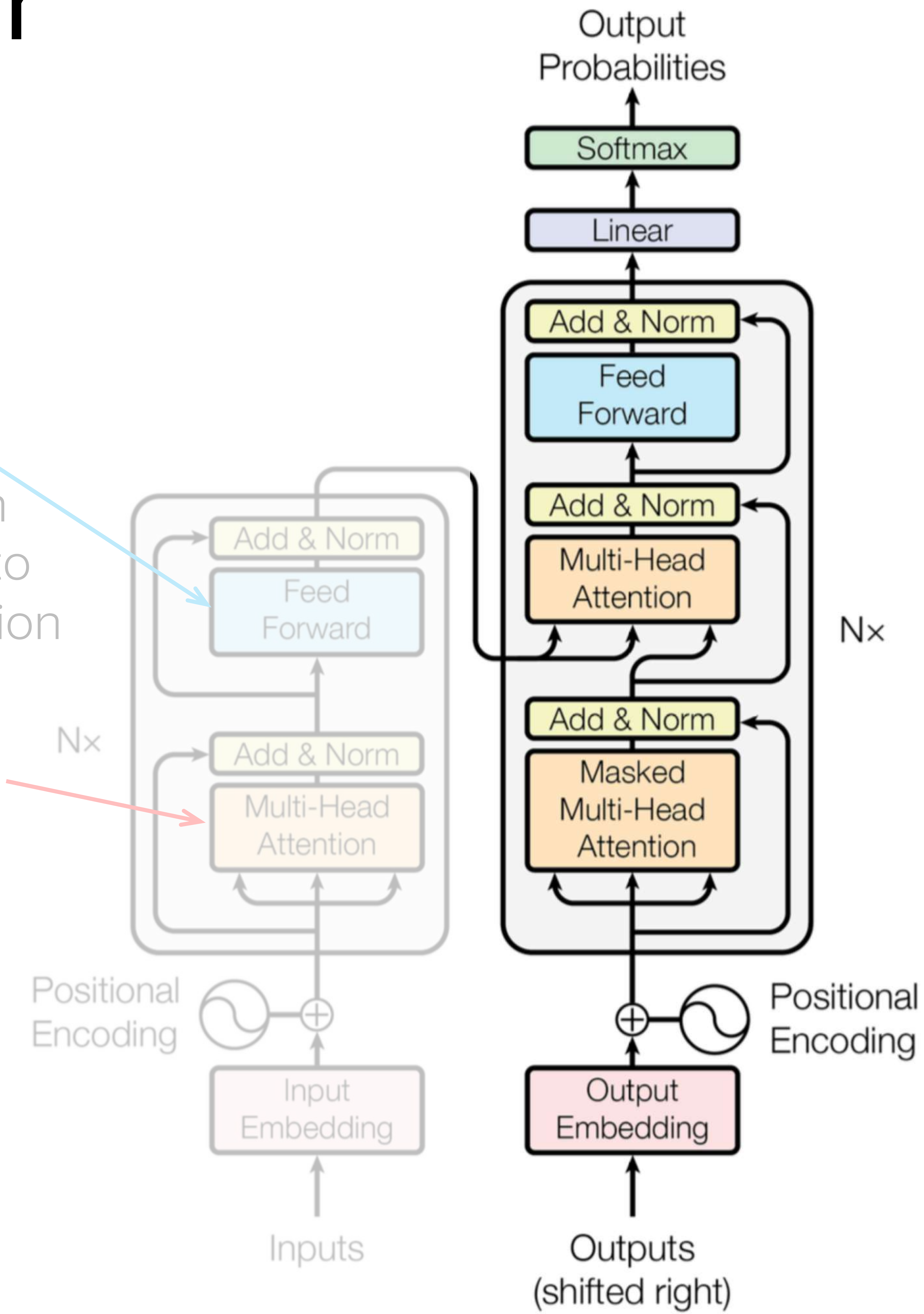


Transformer

Feed-forward network:
after taking information from other tokens, take a moment to think and process this information

Encoder self-attention:
tokens look at each other

queries, keys, values are computed from encoder states

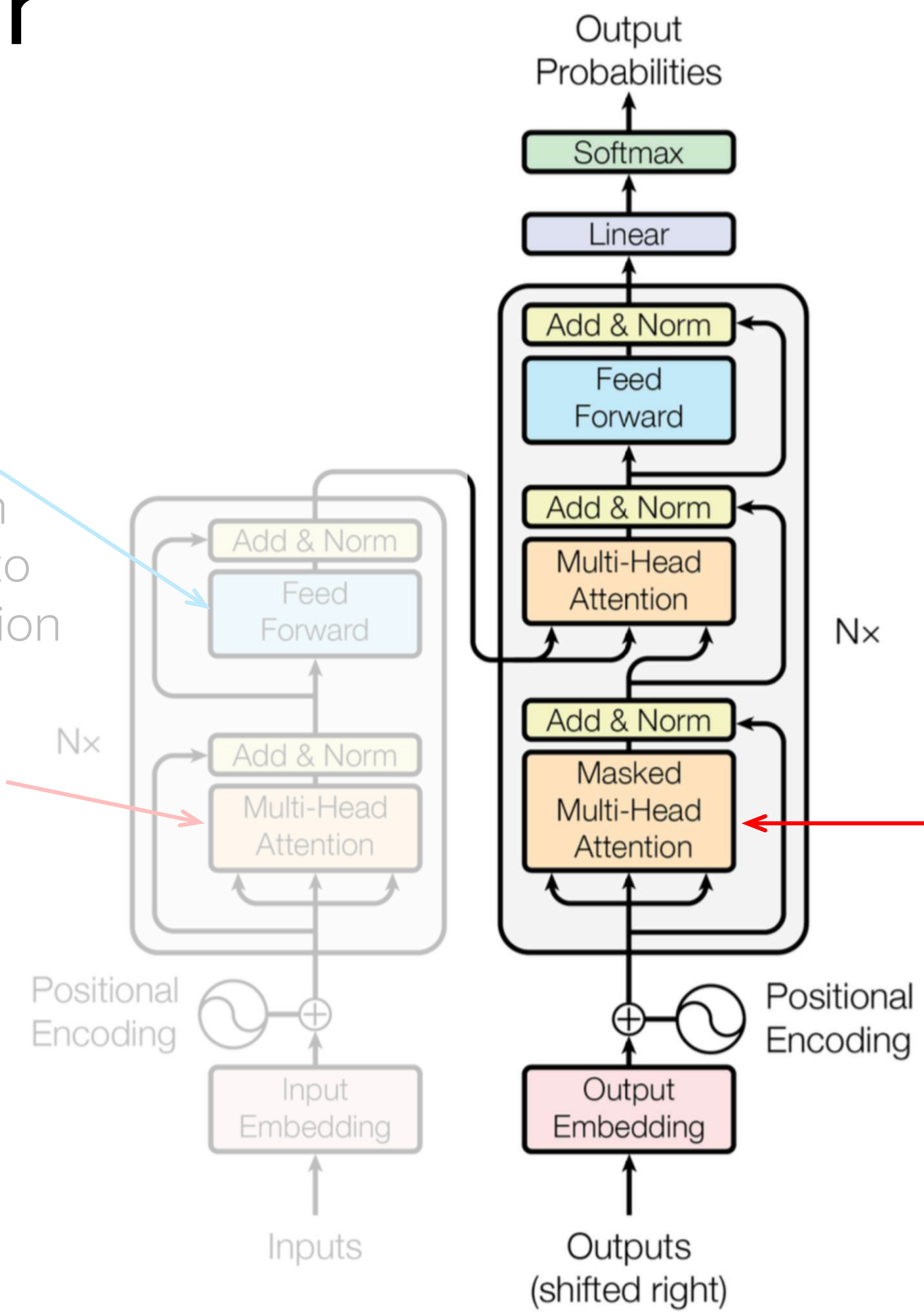


Transformer

Feed-forward network:
after taking information from other tokens, take a moment to think and process this information

Encoder self-attention:
tokens look at each other

queries, keys, values are computed from encoder states



Decoder self-attention (masked):
tokens look at the previous tokens

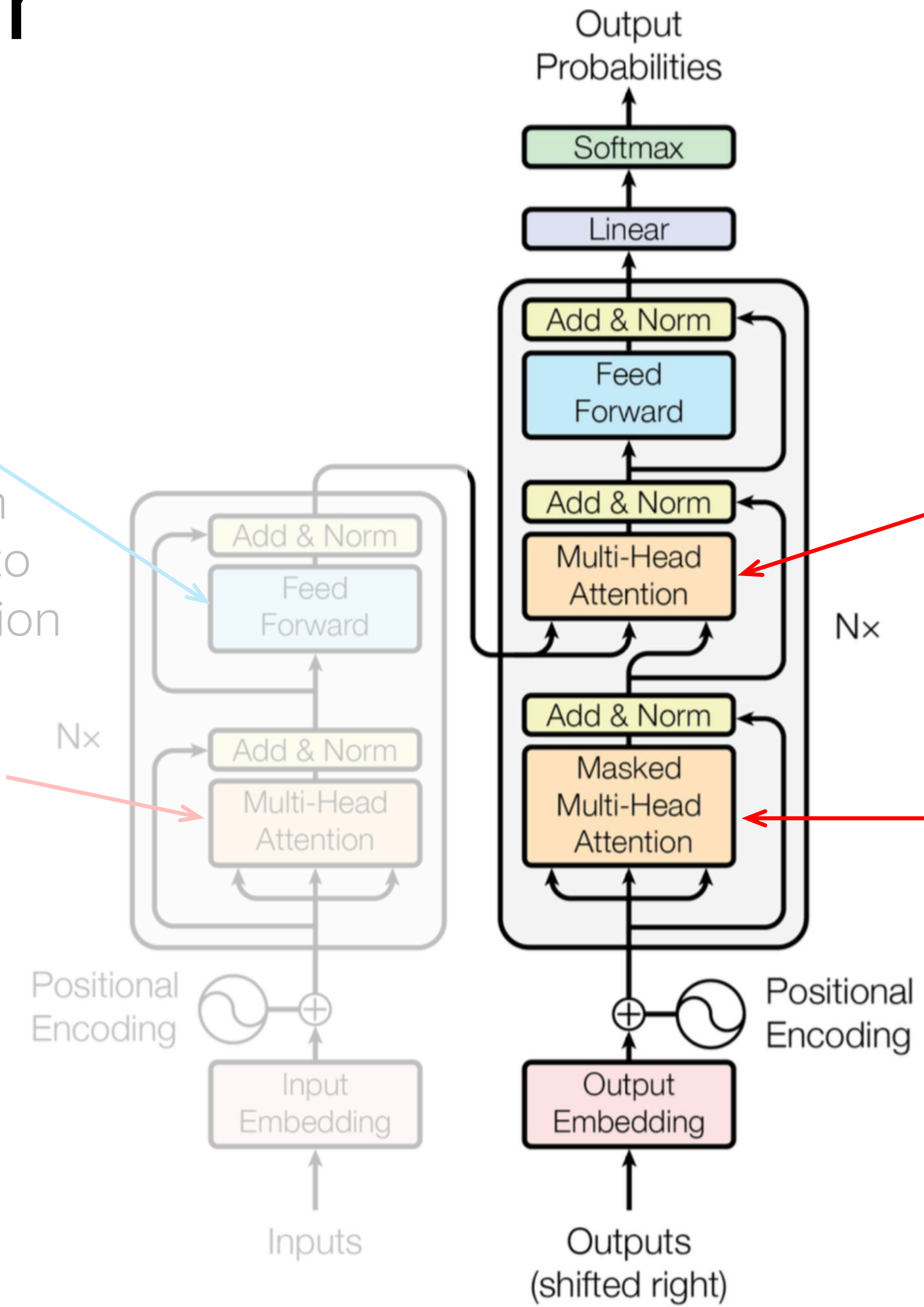
queries, keys, values are computed from decoder states

Transformer

Feed-forward network:
after taking information from other tokens, take a moment to think and process this information

Encoder self-attention:
tokens look at each other

queries, keys, values are computed from encoder states



Decoder-encoder attention:
target token looks at the source

queries – from decoder states; keys and values from encoder states

Decoder self-attention (masked):
tokens look at the previous tokens

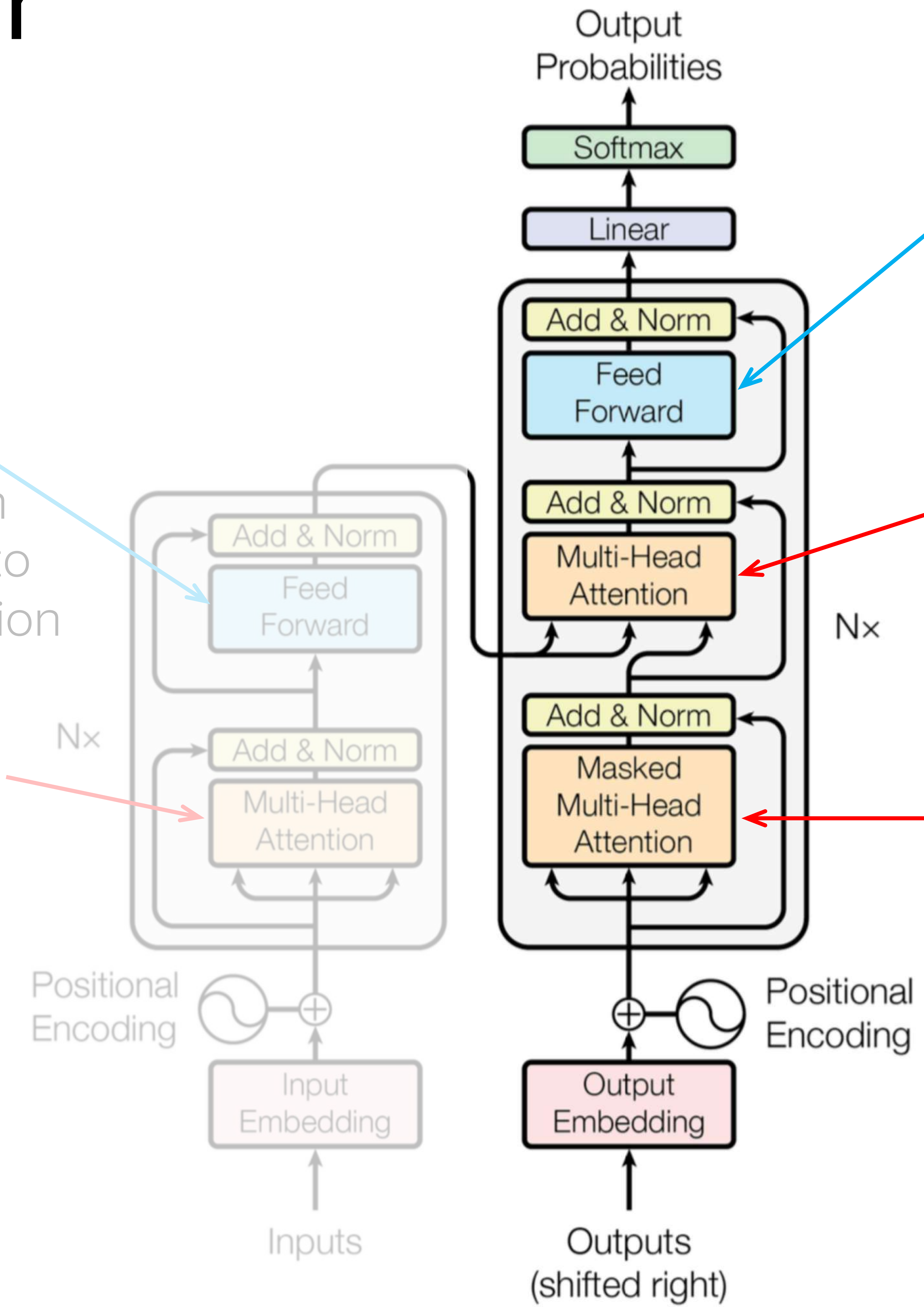
queries, keys, values are computed from decoder states

Transformer

Feed-forward network:
after taking information from other tokens, take a moment to think and process this information

Encoder self-attention:
tokens look at each other

queries, keys, values are computed from encoder states



Feed-forward network:
after taking information from other tokens, take a moment to think and process this information

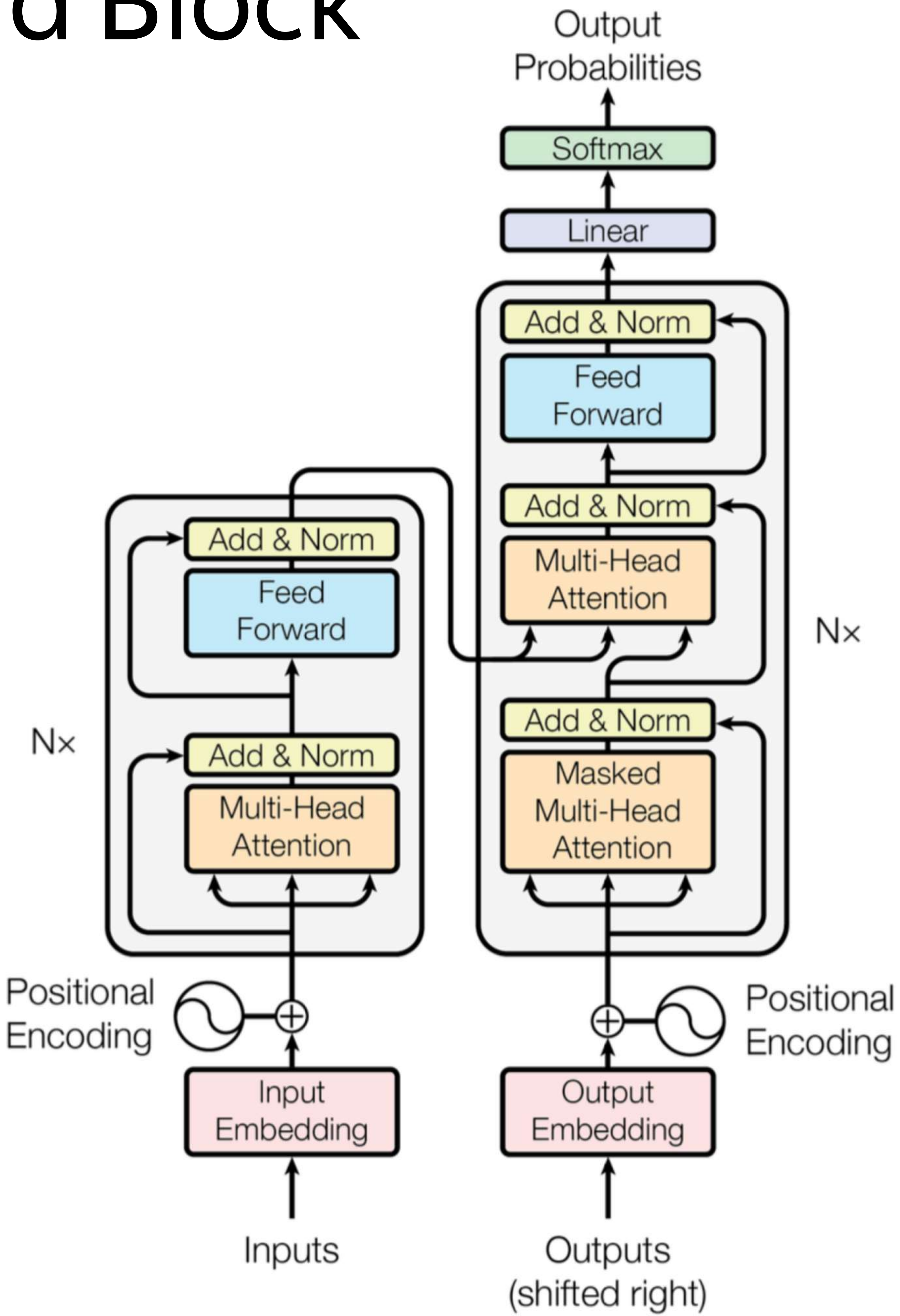
Decoder-encoder attention:
target token looks at the source

queries – from decoder states; keys and values from encoder states

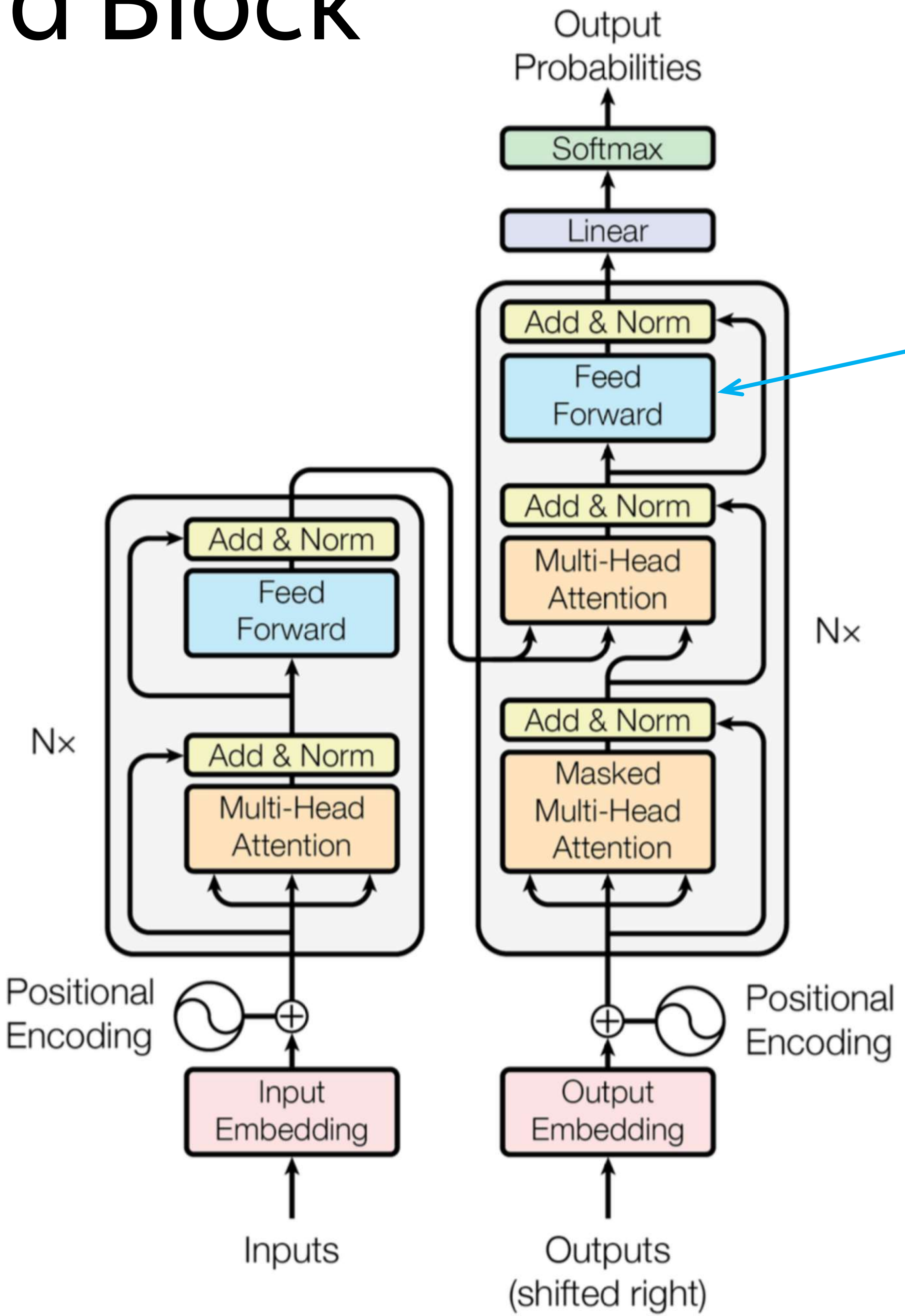
Decoder self-attention (masked):
tokens look at the previous tokens

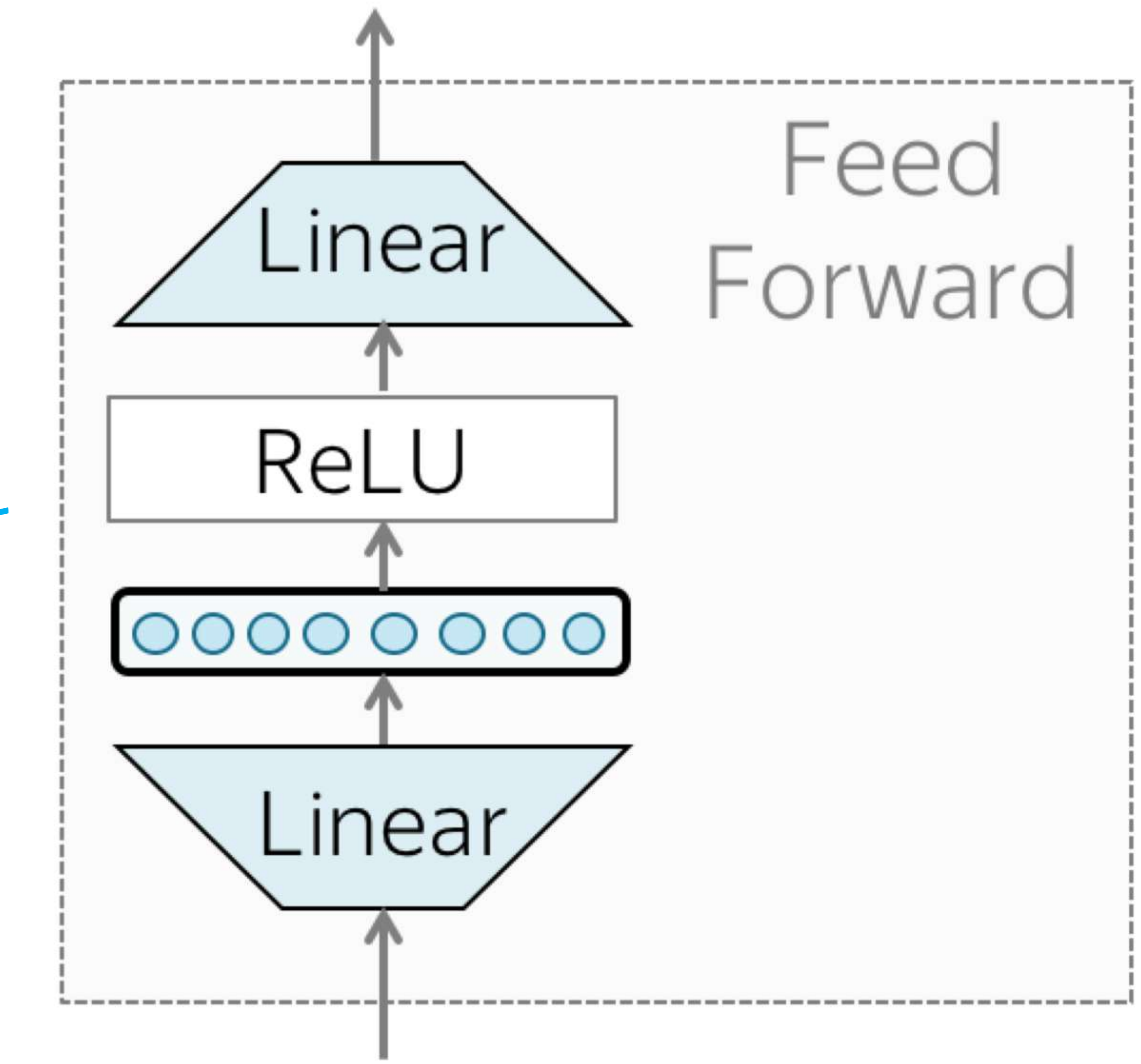
queries, keys, values are computed from decoder states

Feed Forward Block



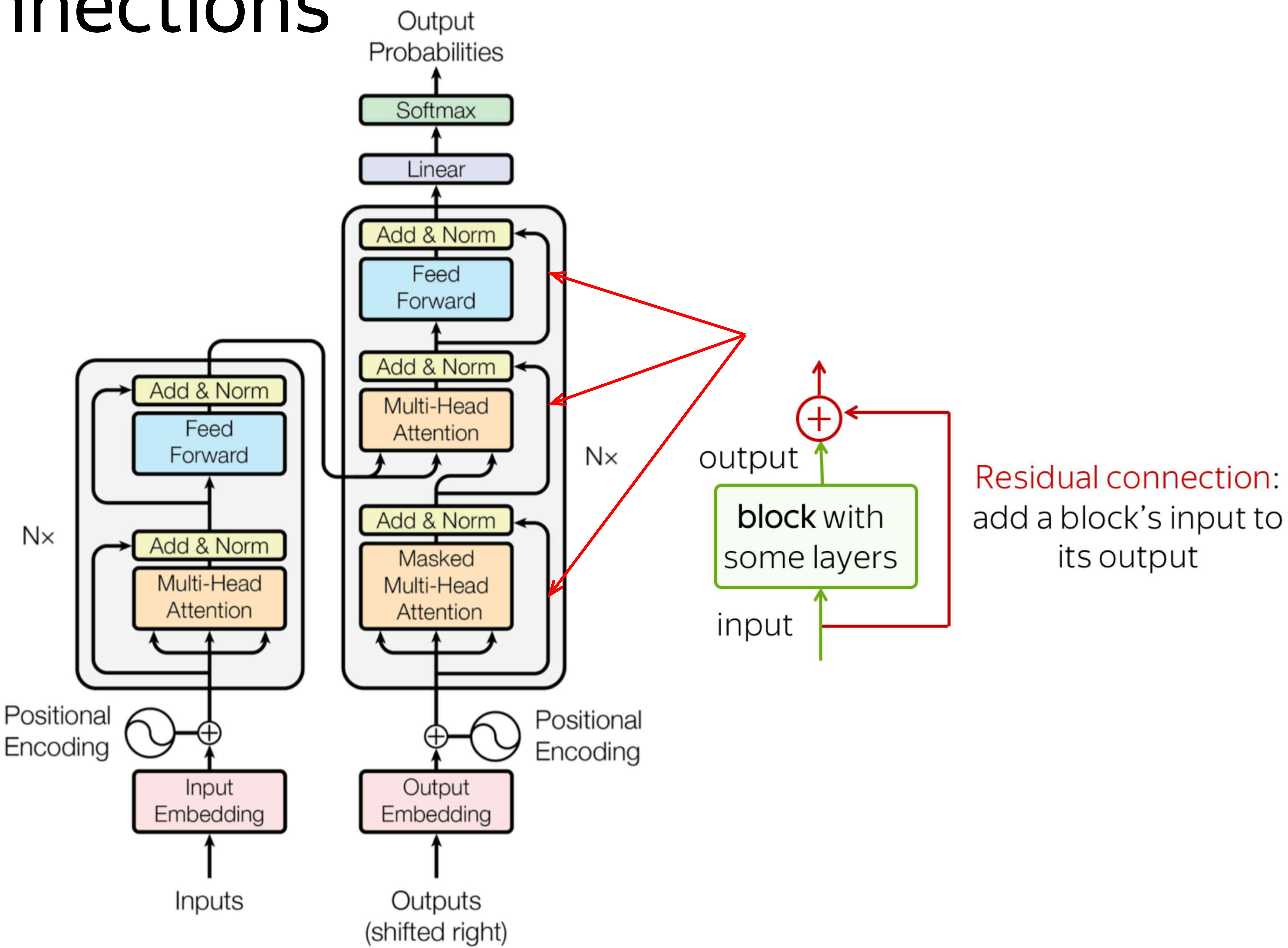
Feed Forward Block



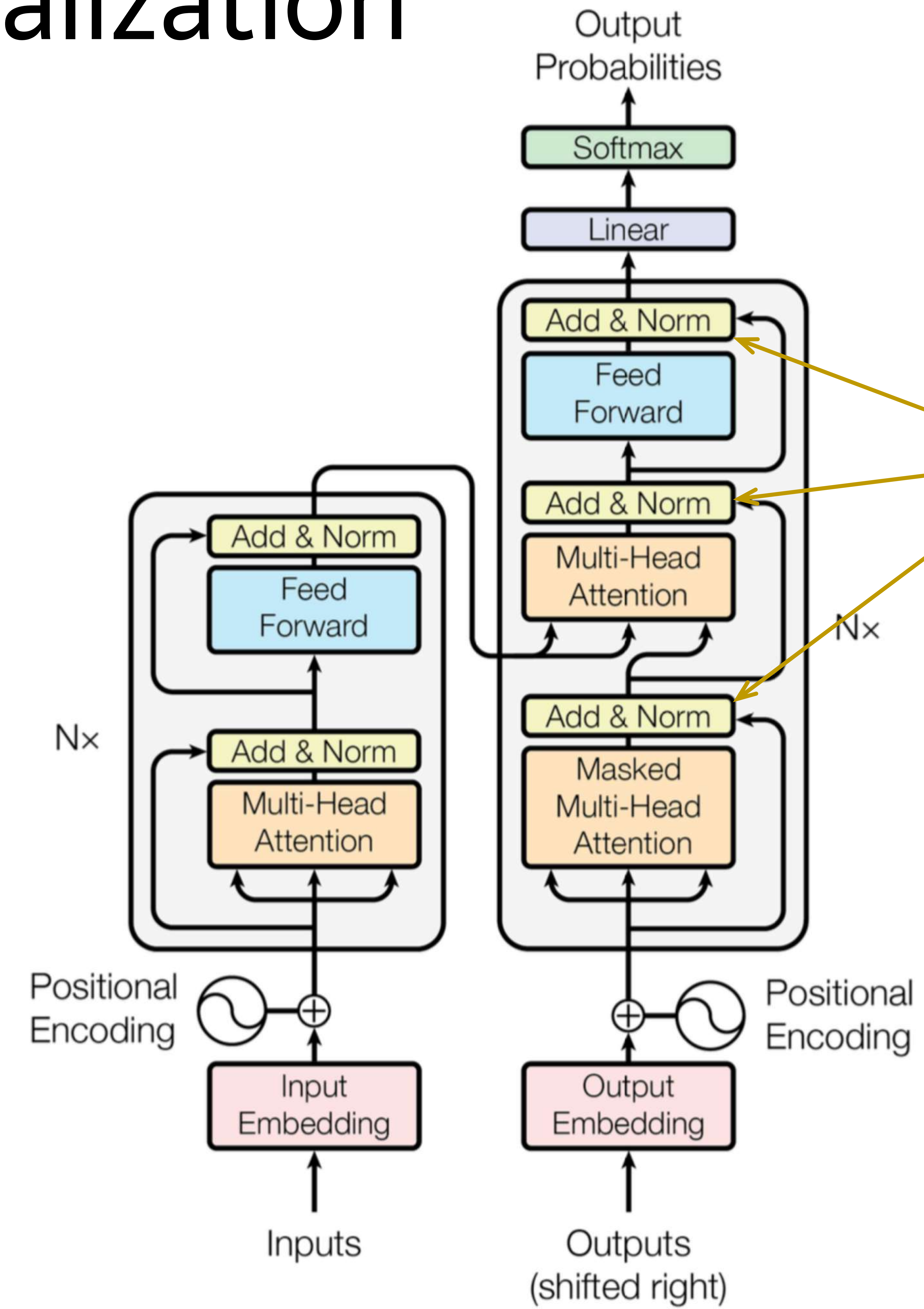


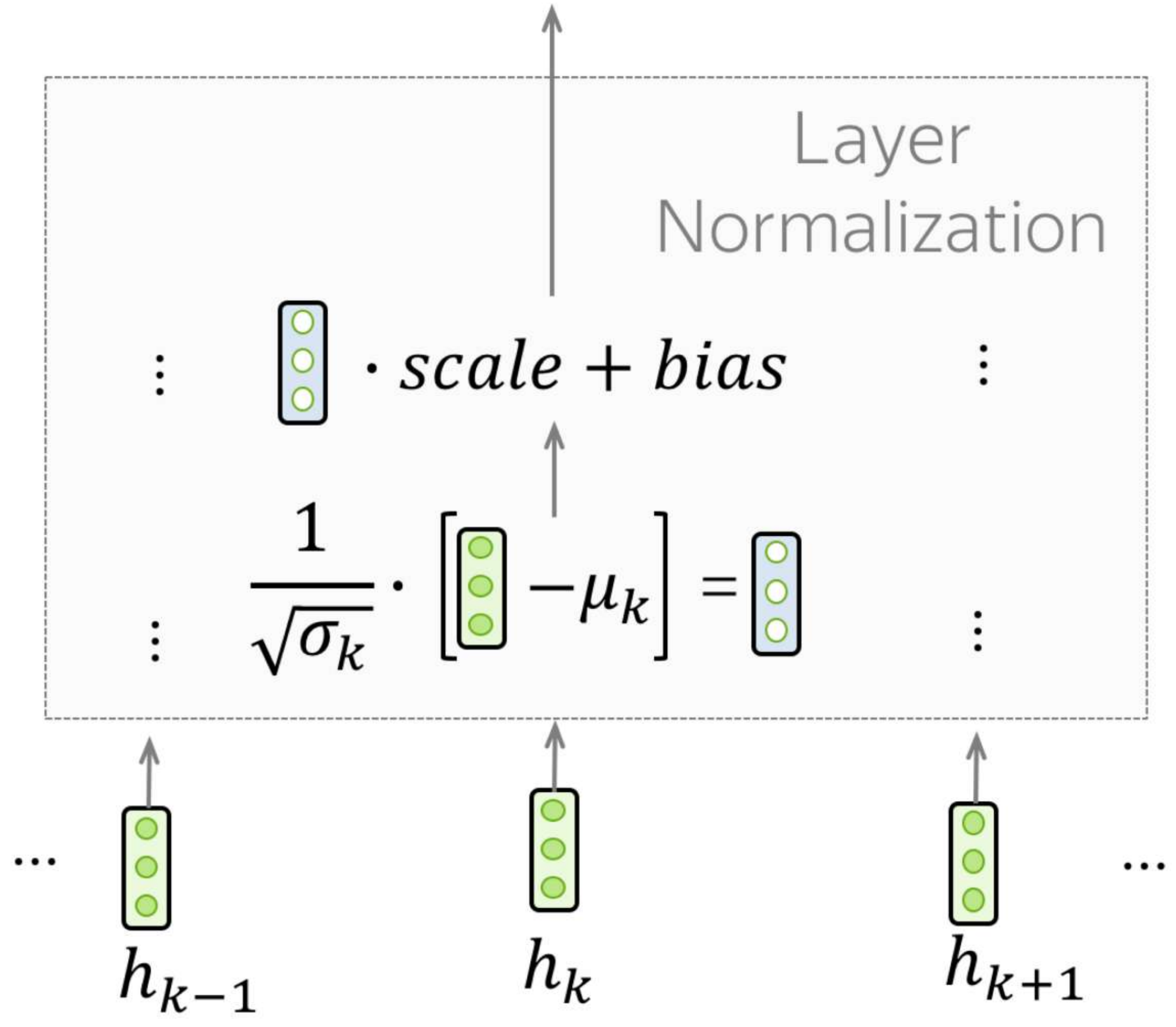
$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Residual Connections

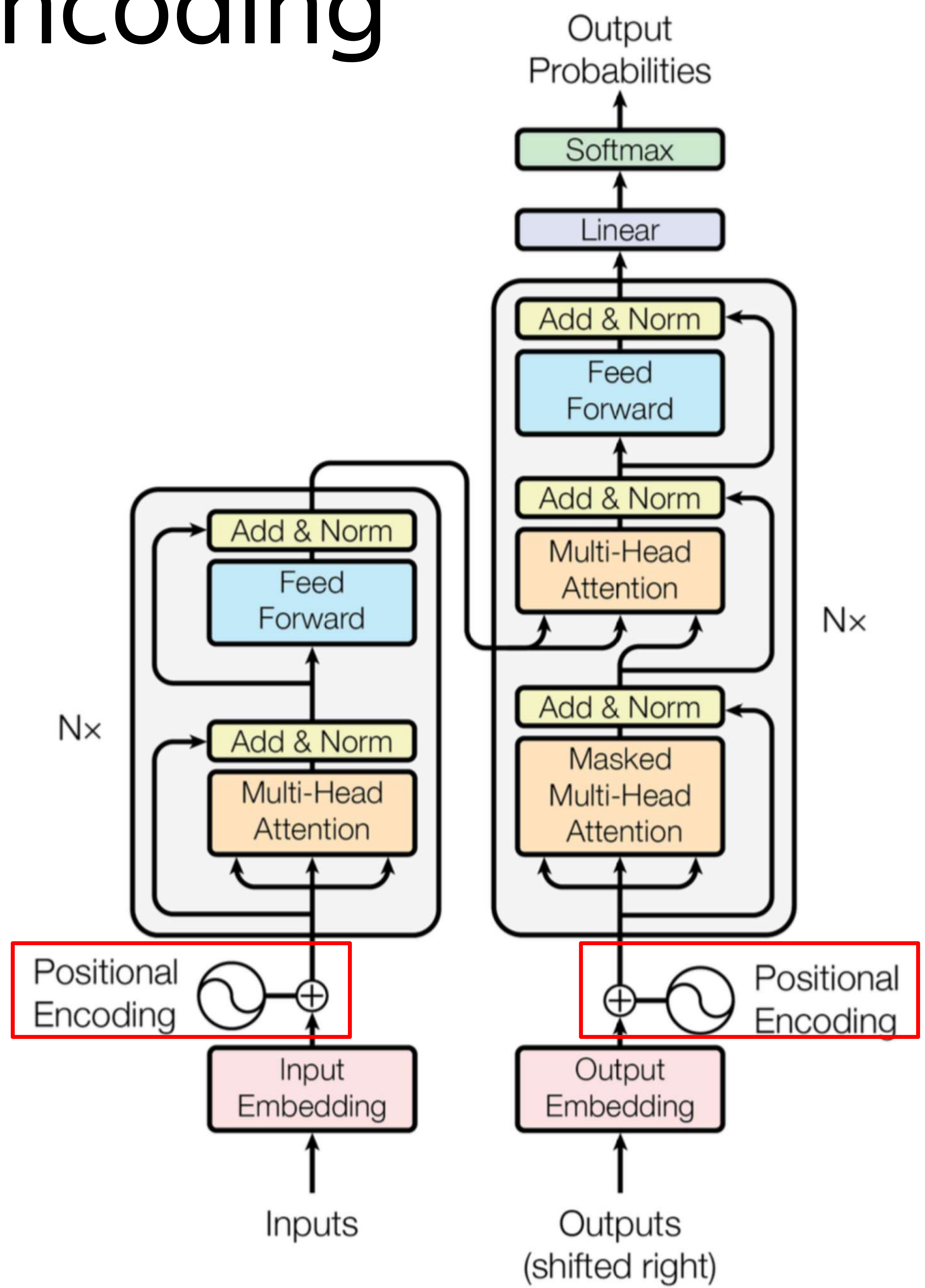


Layer Normalization





Positional Encoding



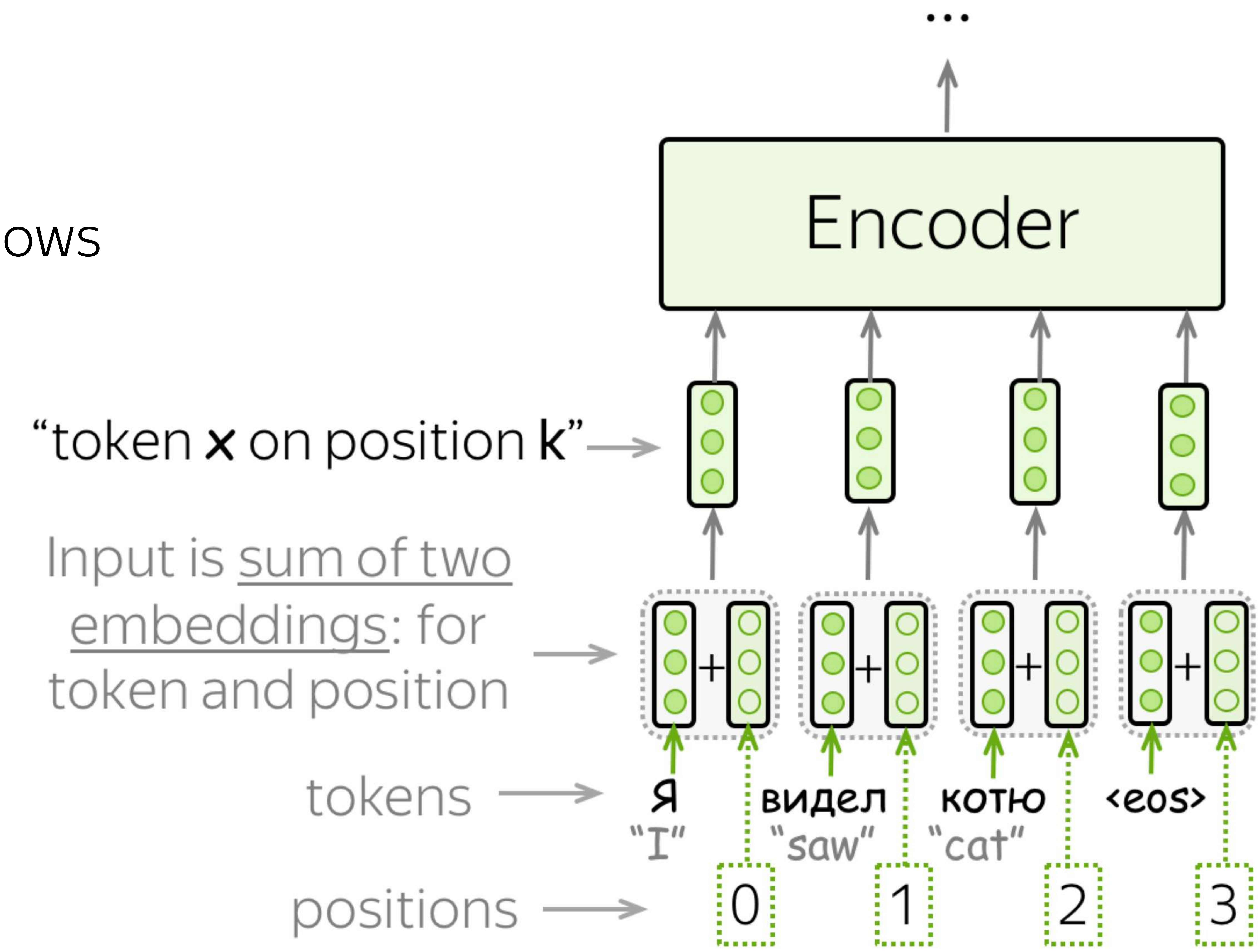
Positional Encoding

Problem:

- without recurrence or convolution, the model knows nothing about position

Solution:

- encode position explicitly and add



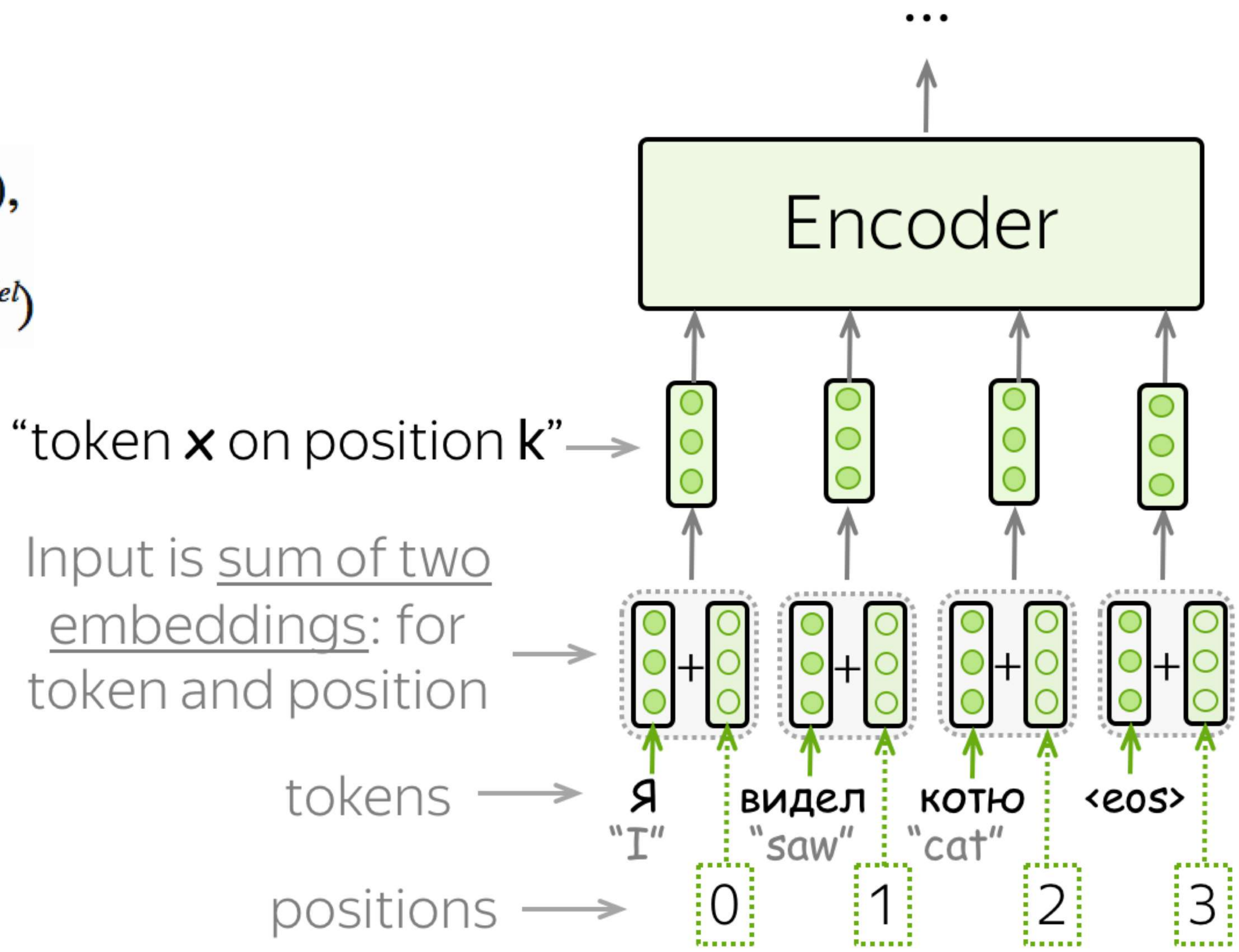
Positional Encoding

Fixed encodings:

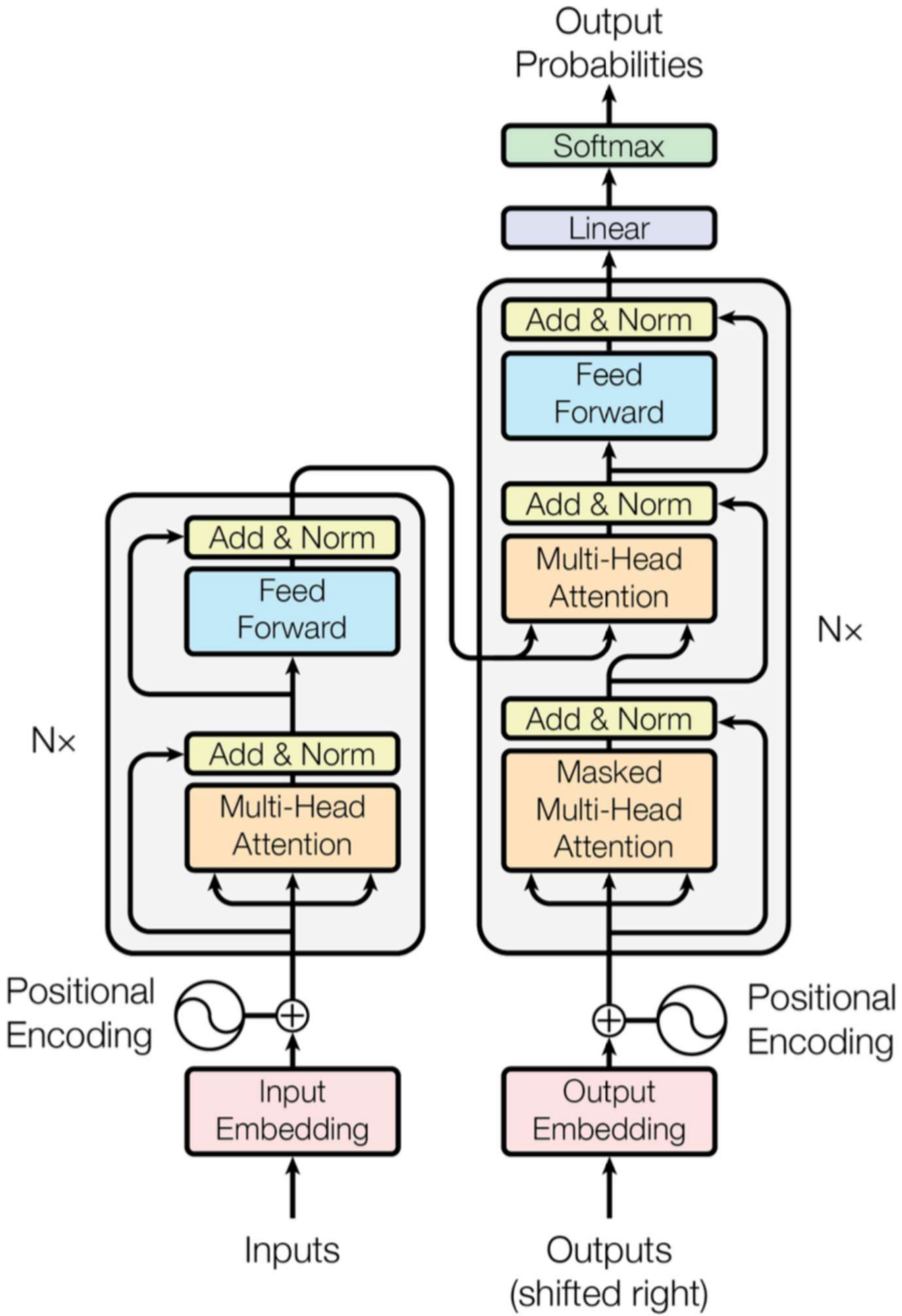
$$PE_{pos,2i} = \sin(pos/10000^{2i/d_{model}}),$$

$$PE_{pos,2i+1} = \cos(pos/10000^{2i/d_{model}})$$

pos – position, i - dimension



Transformer



What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
- Subword Segmentation: BPE
- Analysis and Interpretability

What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
- Subword Segmentation: BPE → NLP Course For You: read here
- Analysis and Interpretability

What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
- Subword Segmentation: BPE
- Analysis and Interpretability

Analysis Methods

Model-specific:

- Looking at model components
- …

Model-agnostic:

Analysis Methods

Model-specific:

- Looking at model components
- …

In the previous lectures:

- Convolutional filters of classifiers
- Neurons in RNN/CNN LMs

Model-agnostic:

Analysis Methods

Model-specific:

- Looking at model components
- ...

In the previous lectures:

- Convolutional filters of classifiers
- Neurons in RNN/CNN LMs

Today:

- Heads in Multi-Head Attention

Model-agnostic:

Analysis Methods

Model-specific:

- Looking at model components
- …

In the previous lectures:

- Convolutional filters of classifiers
- Neurons in RNN/CNN LMs

Today:

- Heads in Multi-Head Attention

Model-agnostic:

In the previous lecture:

- Look at the predictions: contrastive evaluation of specific phenomena

Analysis Methods

Model-specific:

- Looking at model components
- ...

In the previous lectures:

- Convolutional filters of classifiers
- Neurons in RNN/CNN LMs

Today:

- Heads in Multi-Head Attention

Model-agnostic:

In the previous lecture:

- Look at the predictions: contrastive evaluation of specific phenomena

Today:

- Probing: What do representations capture?

Analysis Methods

<u>Model-specific:</u>

- Looking at model components
- …

In the previous lectures:

- Convolutional filters of classifiers
- Neurons in RNN/CNN LMs

Today:

- Heads in Multi-Head Attention

<u>Model-agnostic:</u>

In the previous lecture:

- Look at the predictions: contrastive evaluation of specific phenomena

Today:

- Probing: What do representations capture?

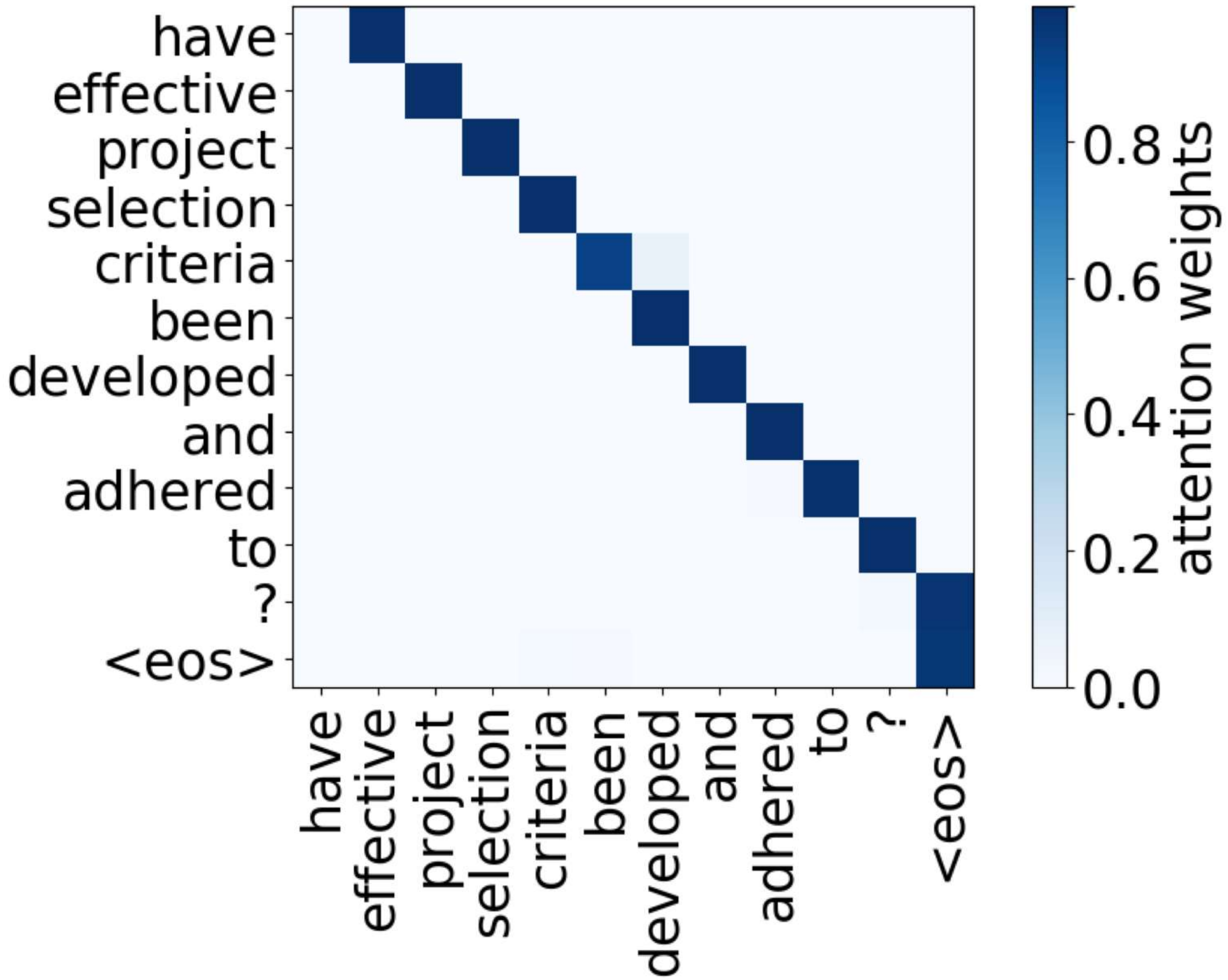
Multi-Head Attention: What are the Heads Doing?

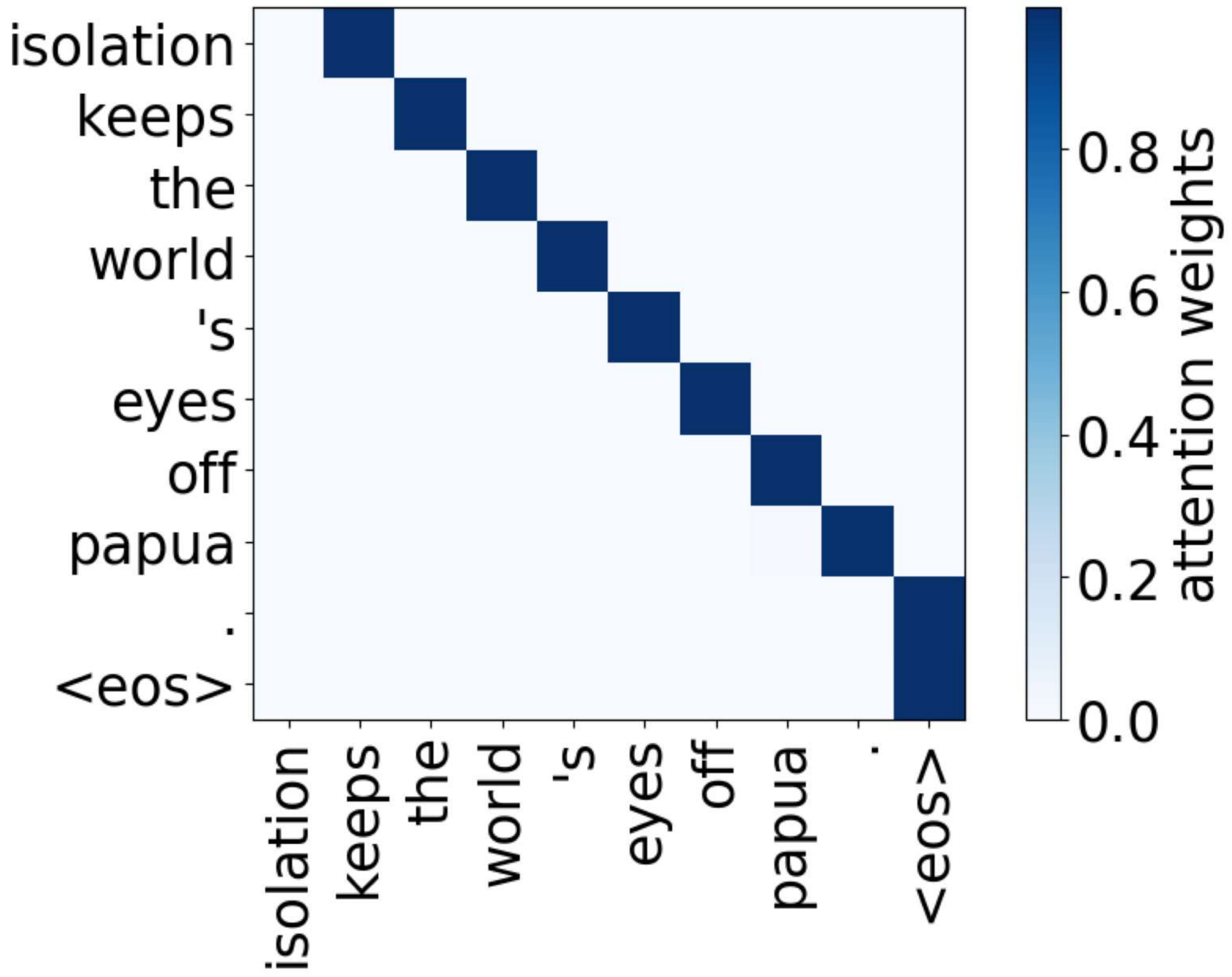
Heads which on average contribute more to generated translations (“important heads”) play interpretable roles:

- Positional
- Syntactic
- Attention to rare tokens

Paper: Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned

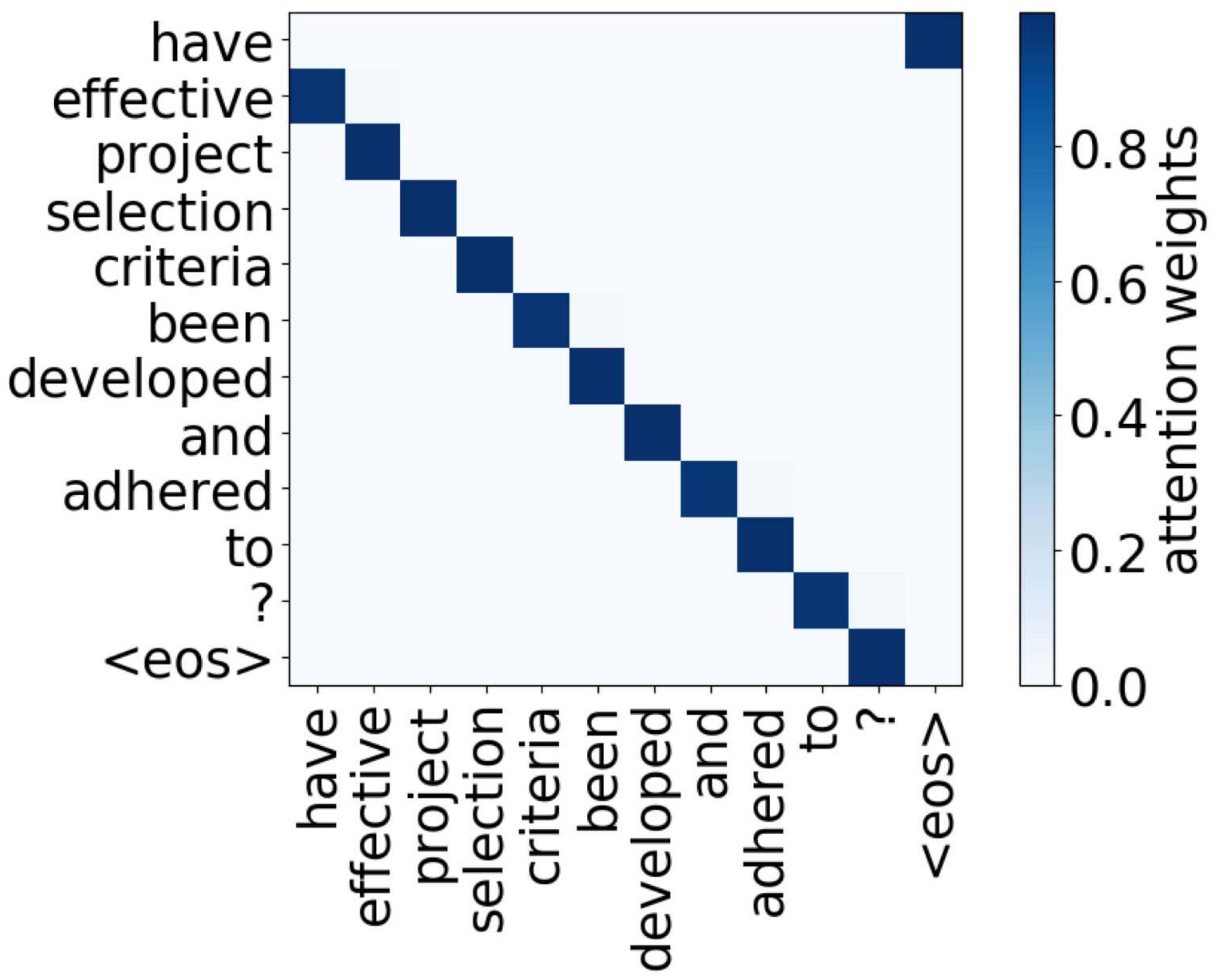
Positional Heads: Attention to Neighbors

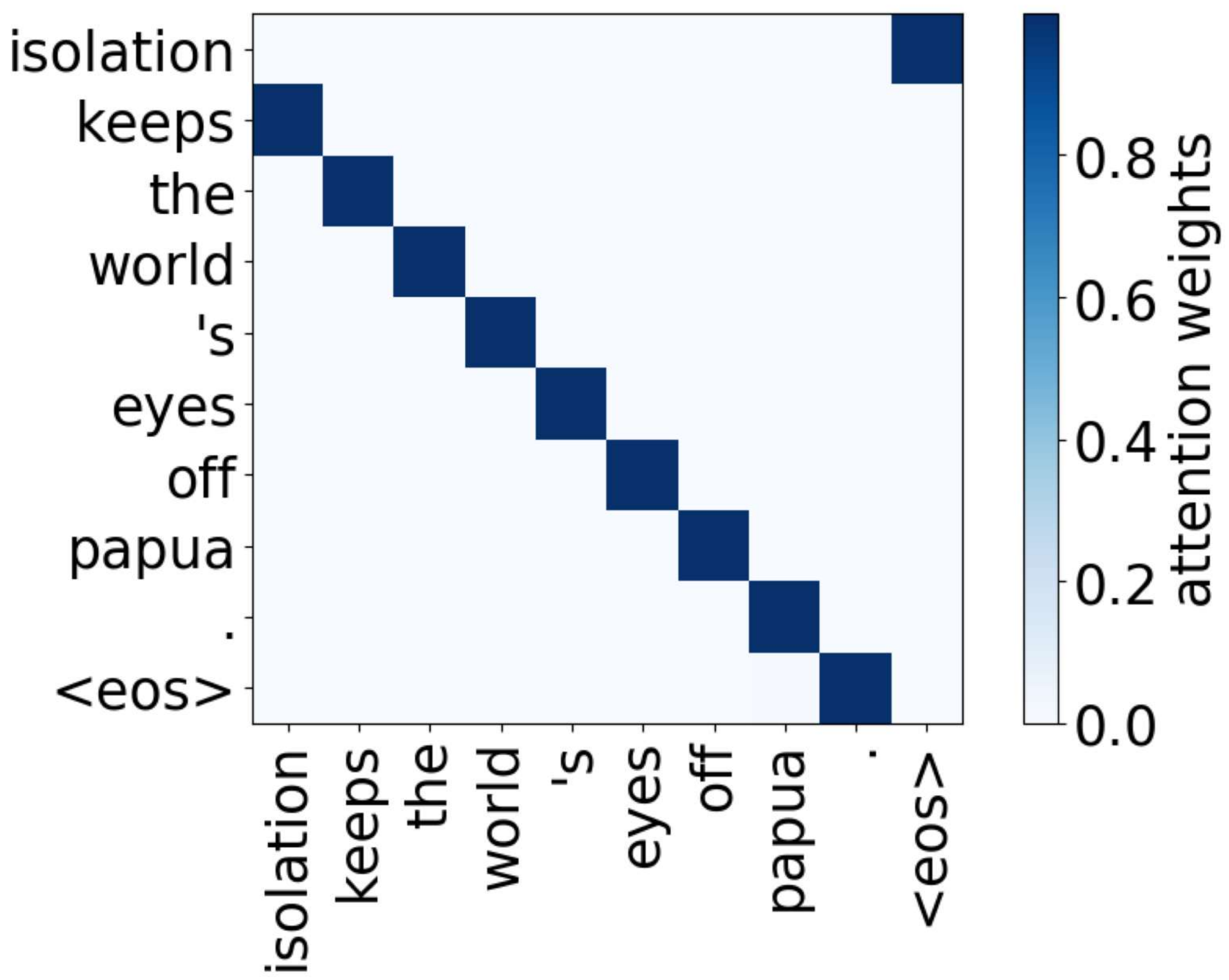




Paper: Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned

Positional Heads: Attention to Neighbors





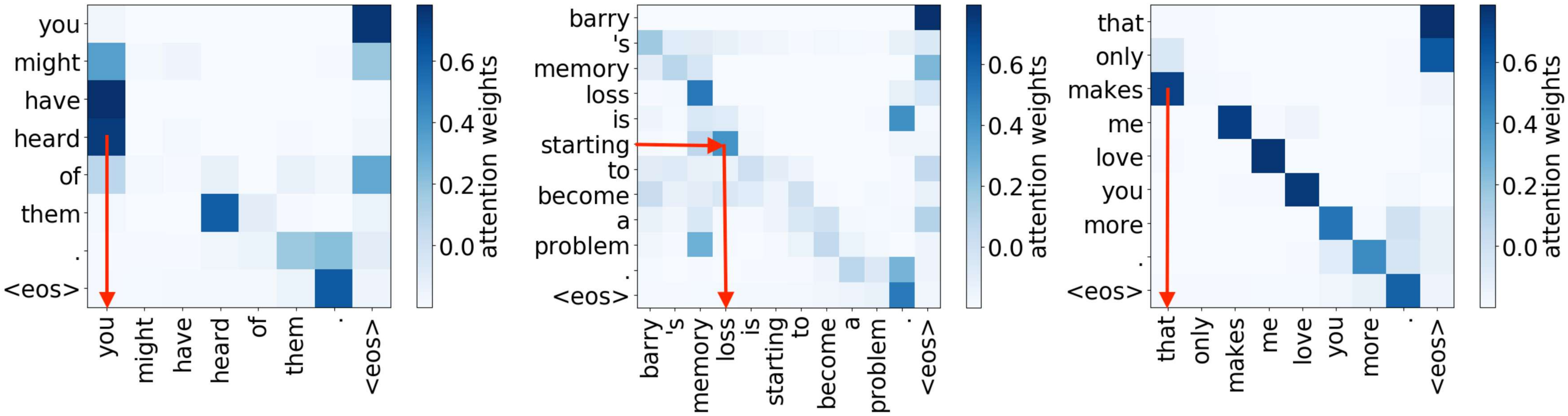
Paper: Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned

Syntactic Heads: Track Dependencies

- verb->subject

Она руководит **новым** проектом

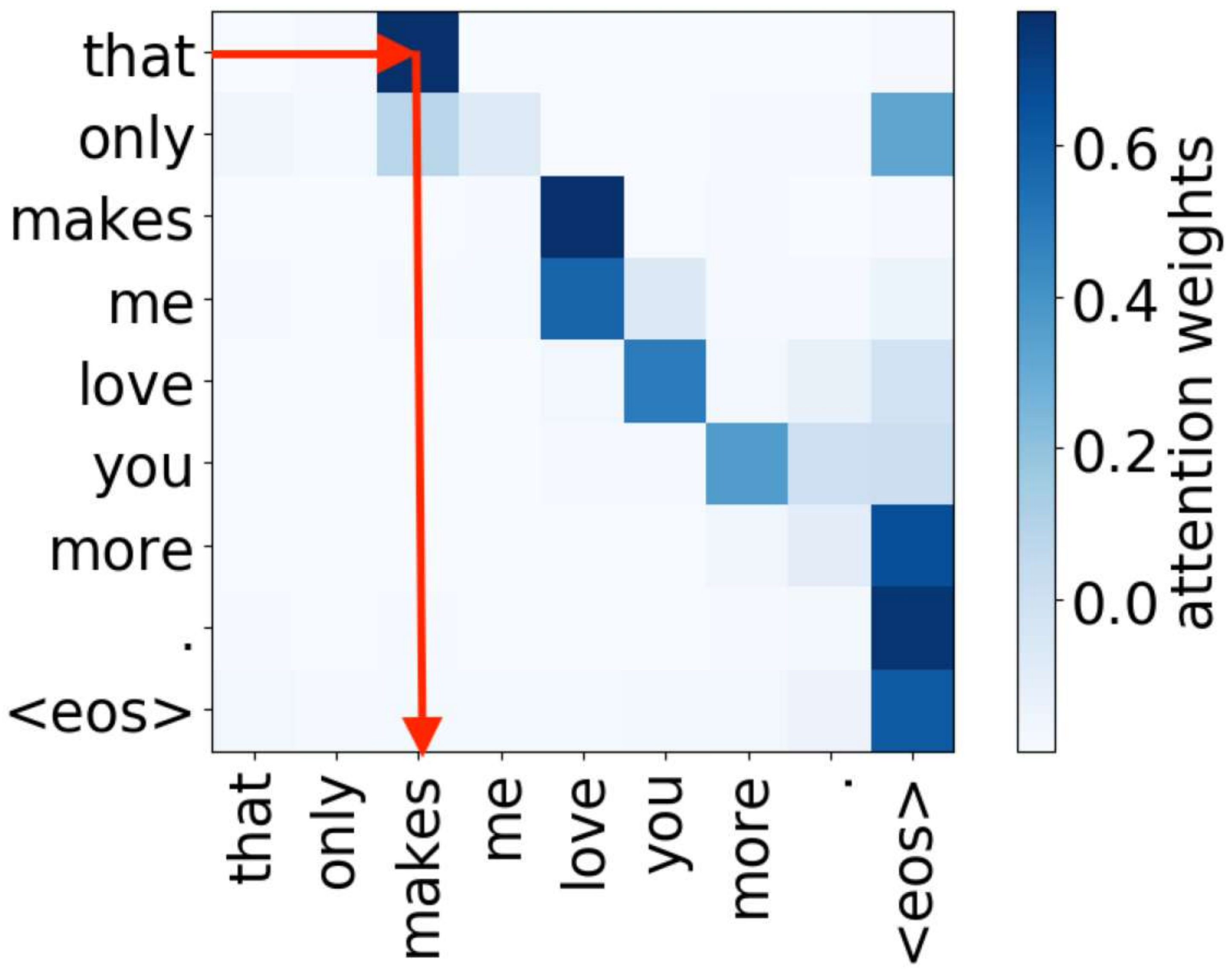
- Gender agreement
- Case government
- Lexical preferences
- ...

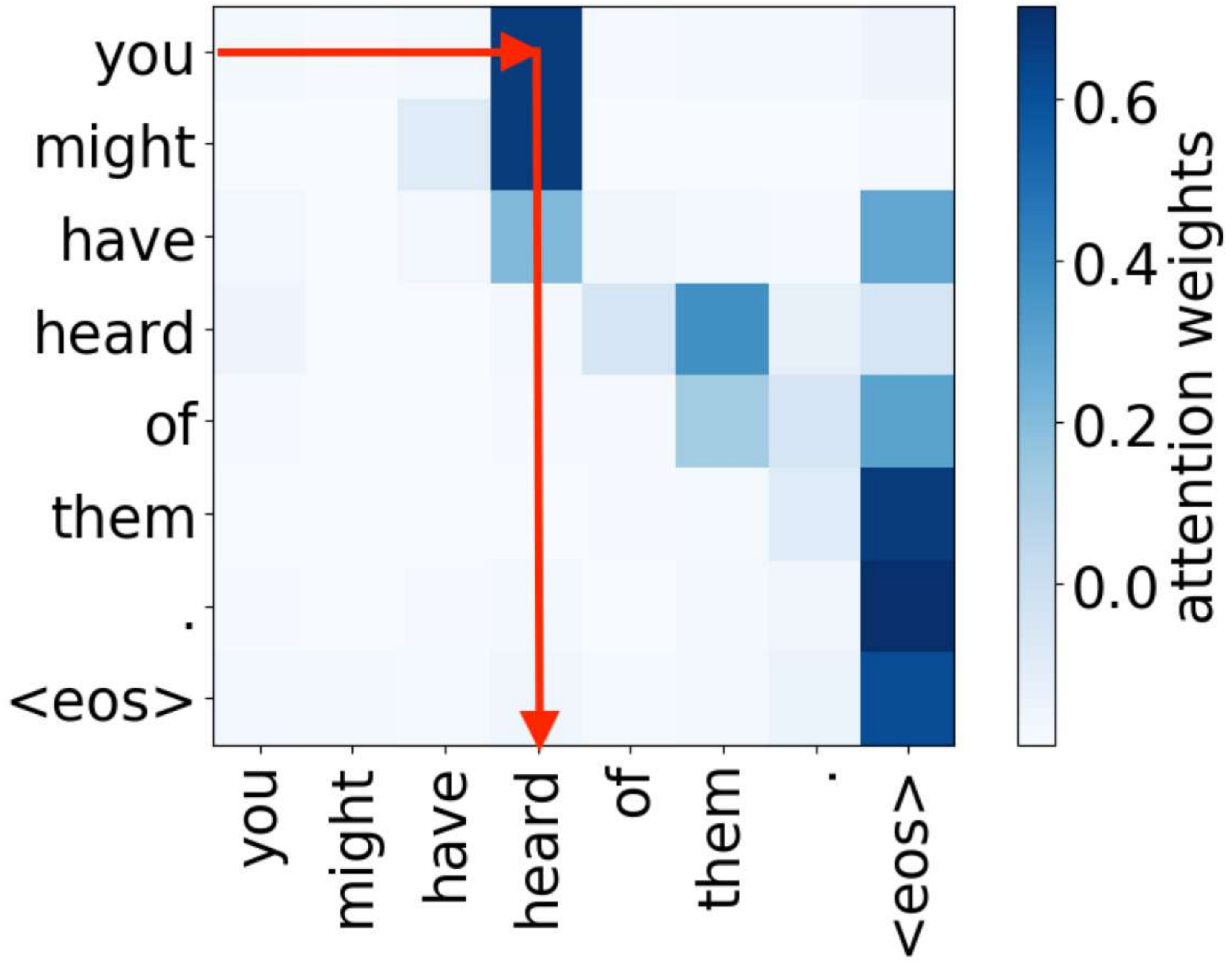


Paper: [Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned]

Syntactic Heads: Track Dependencies

- subject->verb



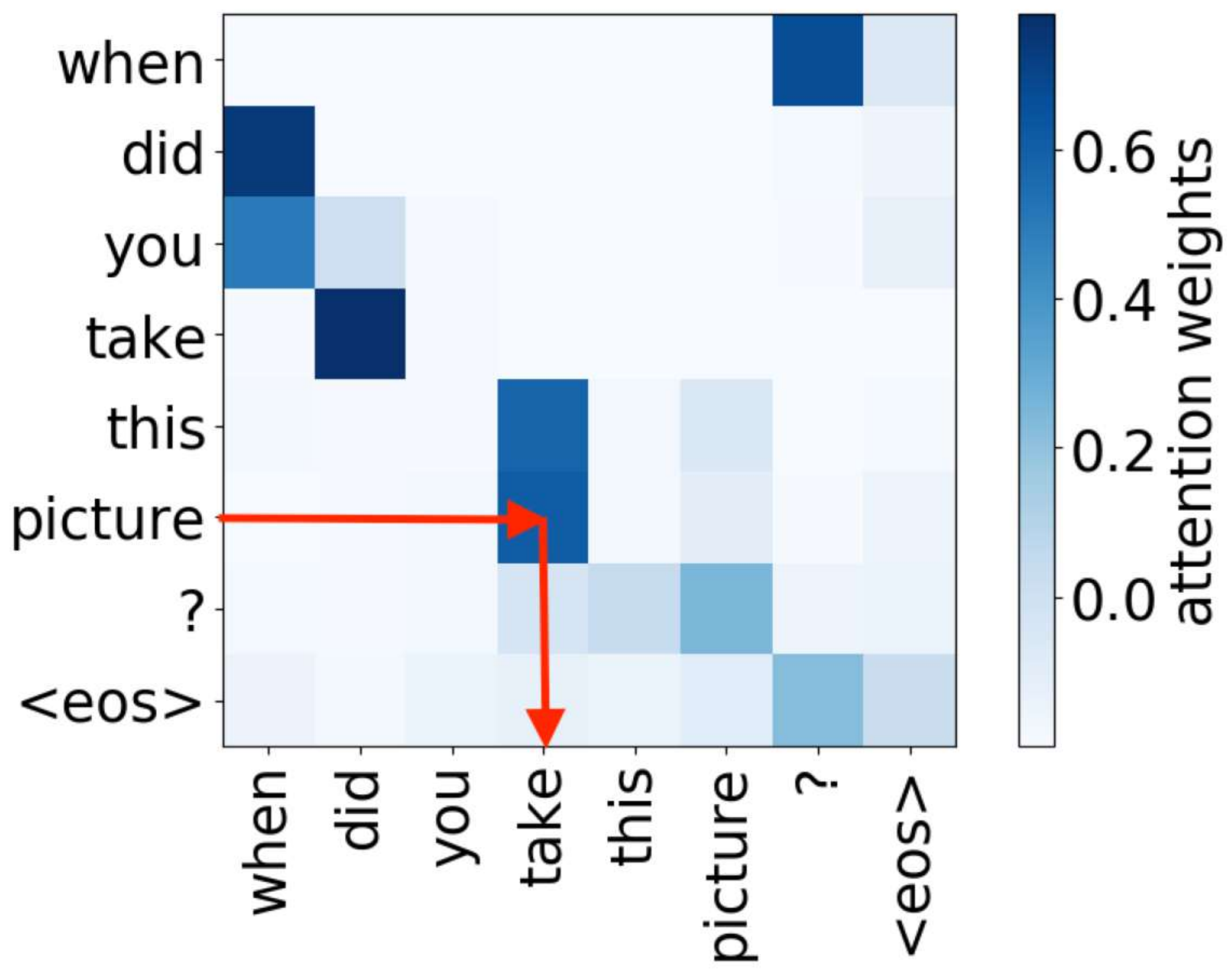


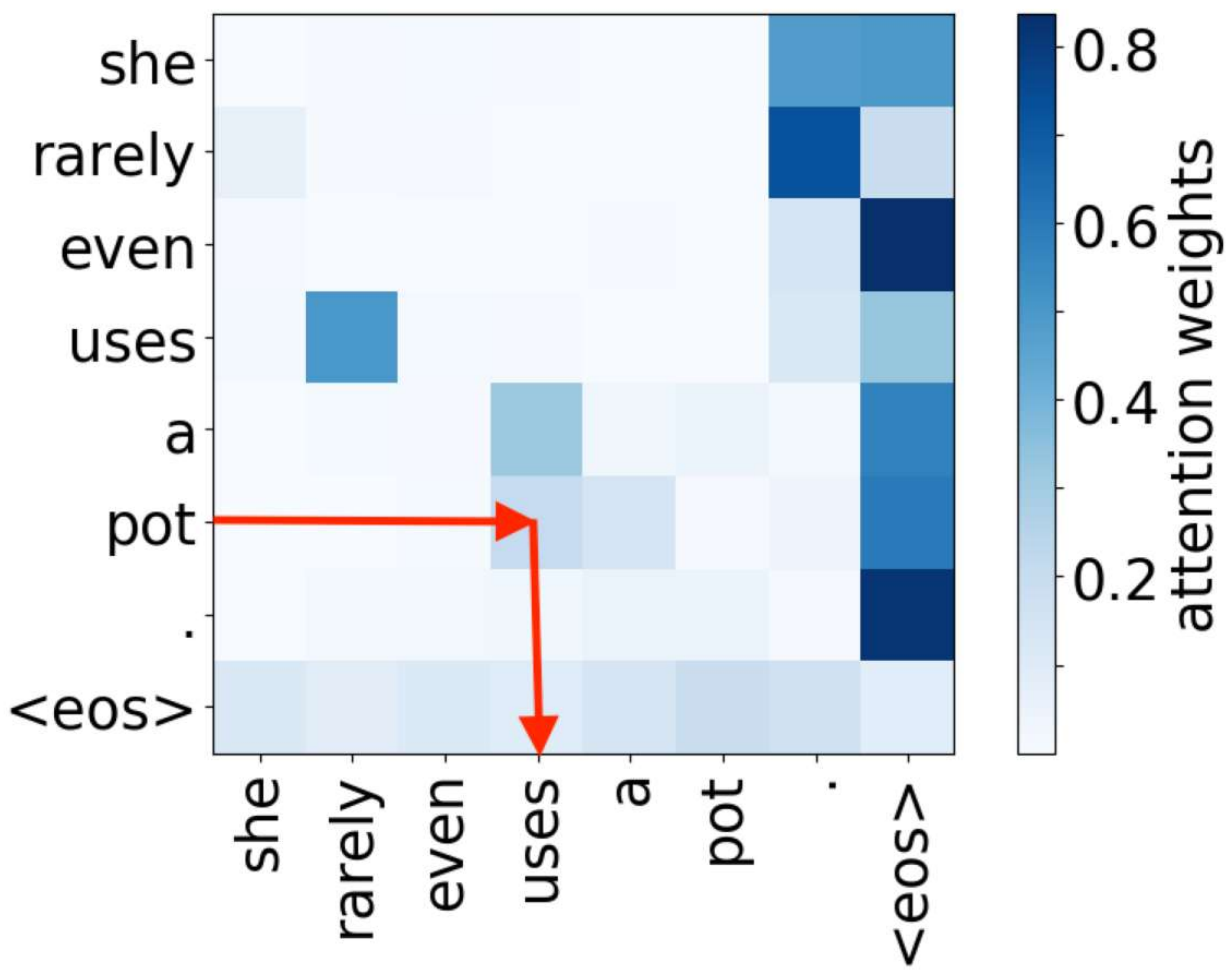
- Gender agreement
- Case government
- Lexical preferences
- ...

Paper: Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned

Syntactic Heads: Track Dependencies

- object->verb

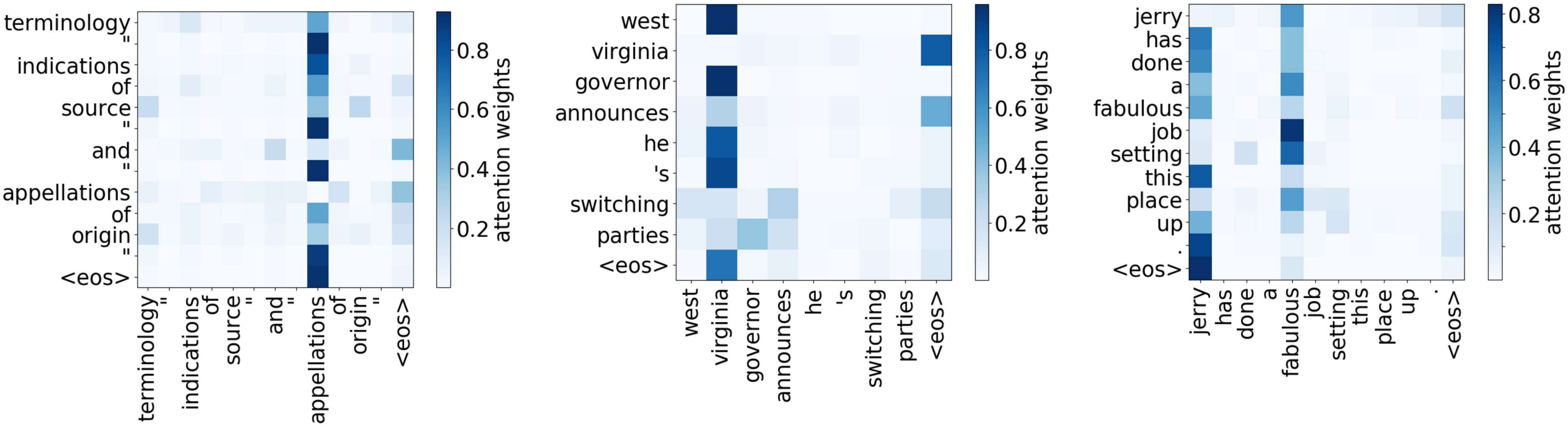




- Gender agreement
- Case government
- Lexical preferences
- ...

Paper: Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned

Attention to Rare Tokens



Paper: Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned

Analysis Methods

<u>Model-specific:</u>

- Looking at model components
- ...

In the previous lectures:

- Convolutional filters of classifiers
- Neurons in RNN/CNN LMs

Today:

- Heads in Multi-Head Attention

<u>Model-agnostic:</u>

In the previous lecture:

- Look at the predictions: contrastive evaluation of specific phenomena

Today:

- Probing: What do representations capture?

Analysis Methods

Model-specific:

- Looking at model components
- ...

In the previous lectures:

- Convolutional filters of classifiers
- Neurons in RNN/CNN LMs

Today:

- Heads in Multi-Head Attention

Model-agnostic:

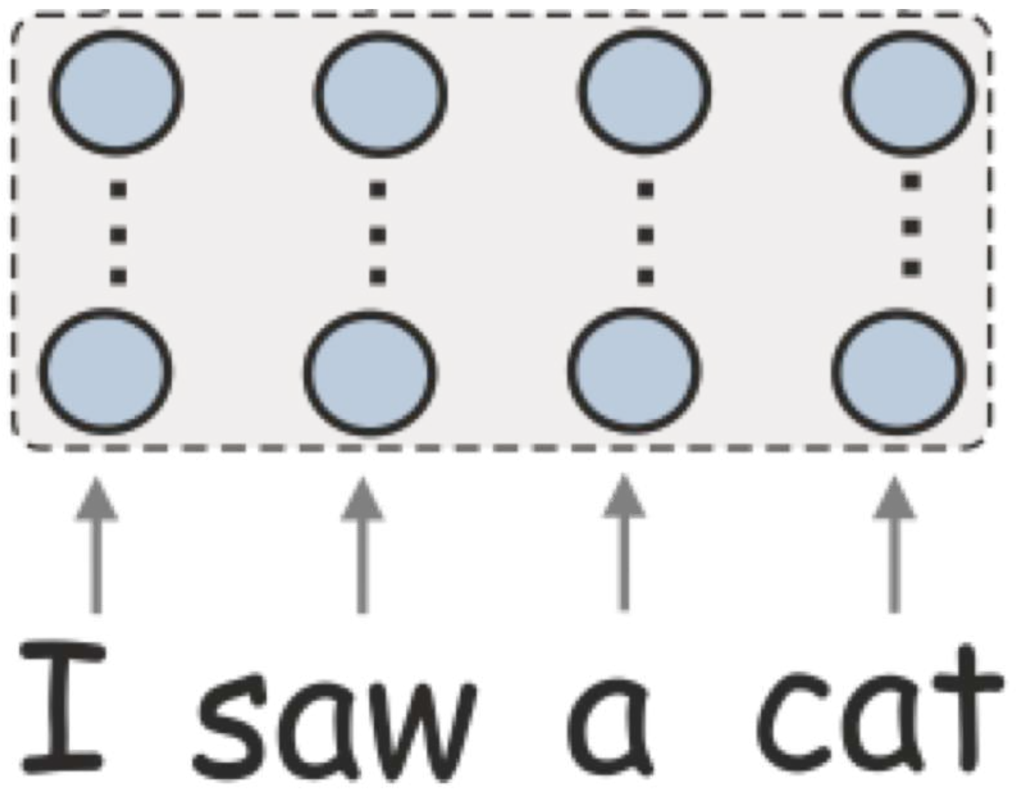
In the previous lecture:

- Look at the predictions: contrastive evaluation of specific phenomena

Today:

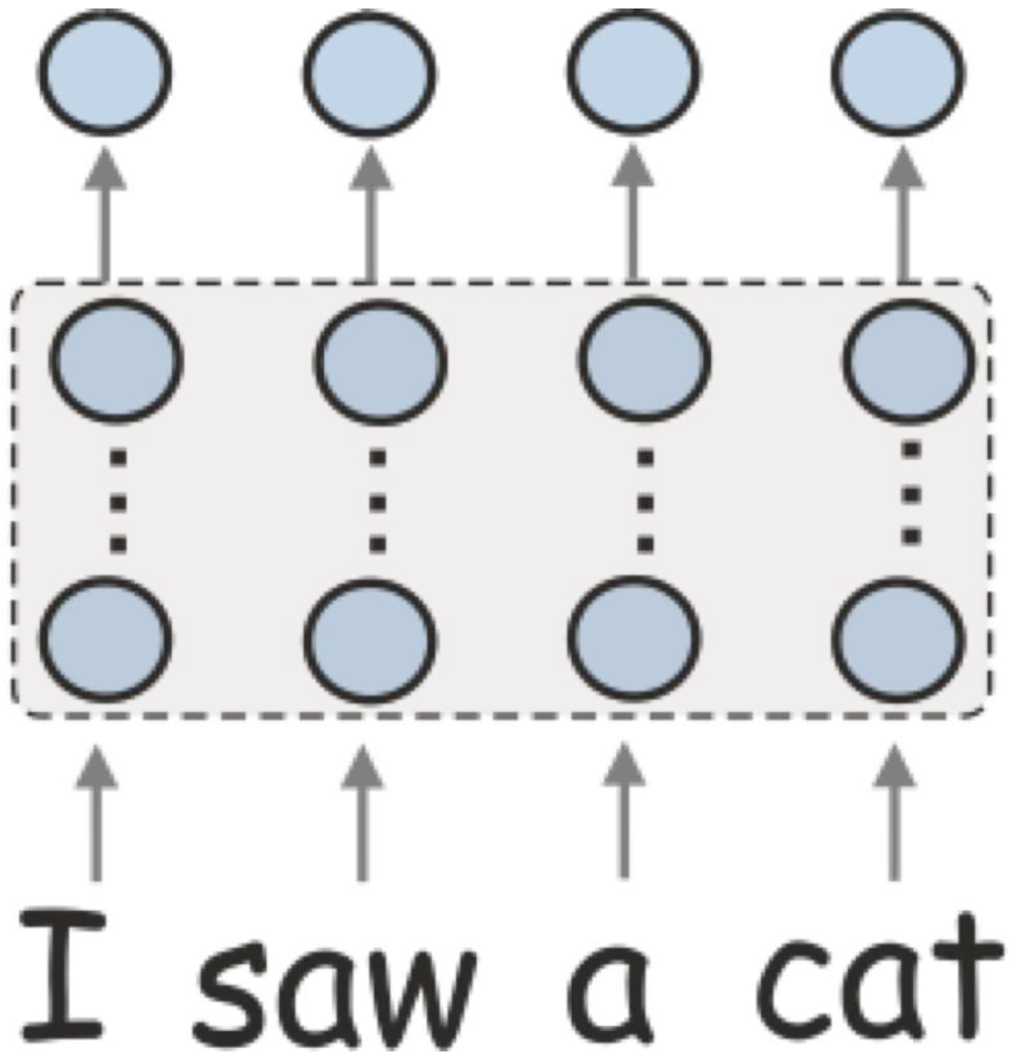
- Probing: What do representations capture?

How to understand if a model captures a linguistic property?



Pretrained model

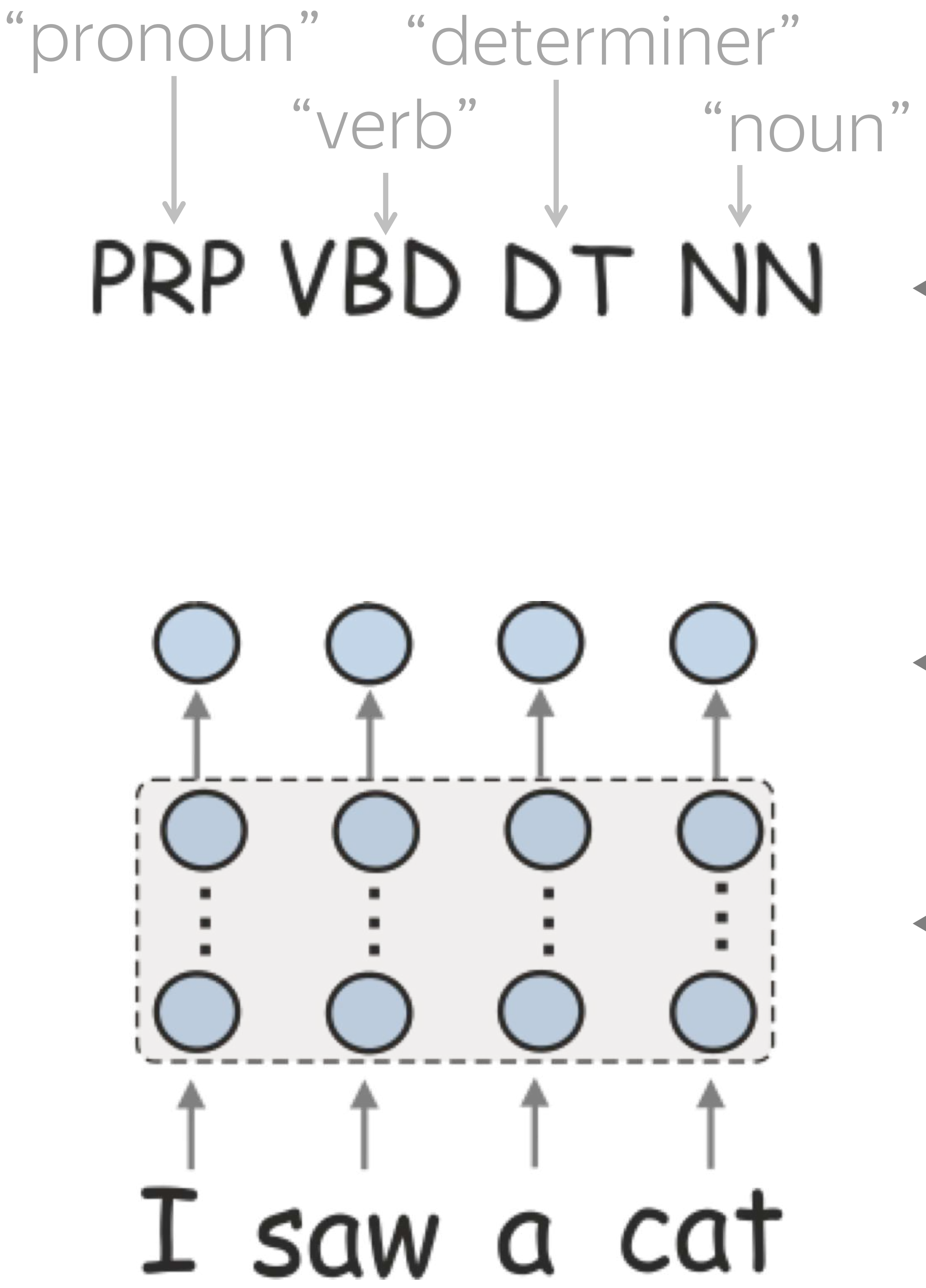
How to understand if a model captures a linguistic property?



Representations: $x_{1:n}$

Pretrained model

How to understand if a model captures a linguistic property?

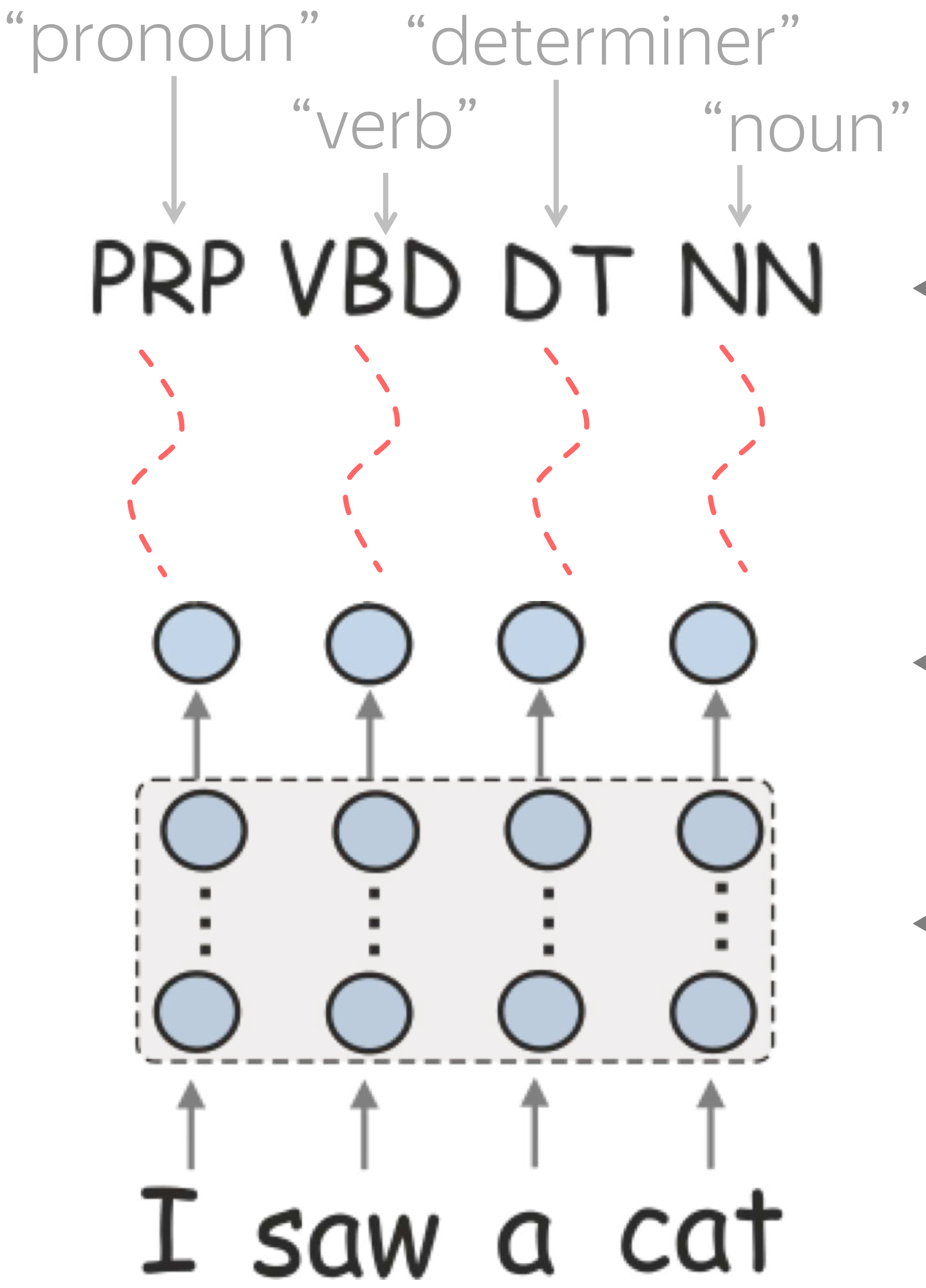


Labels: $y_{1:n}$

Representations: $x_{1:n}$

Pretrained model

How to understand if a model captures a linguistic property?



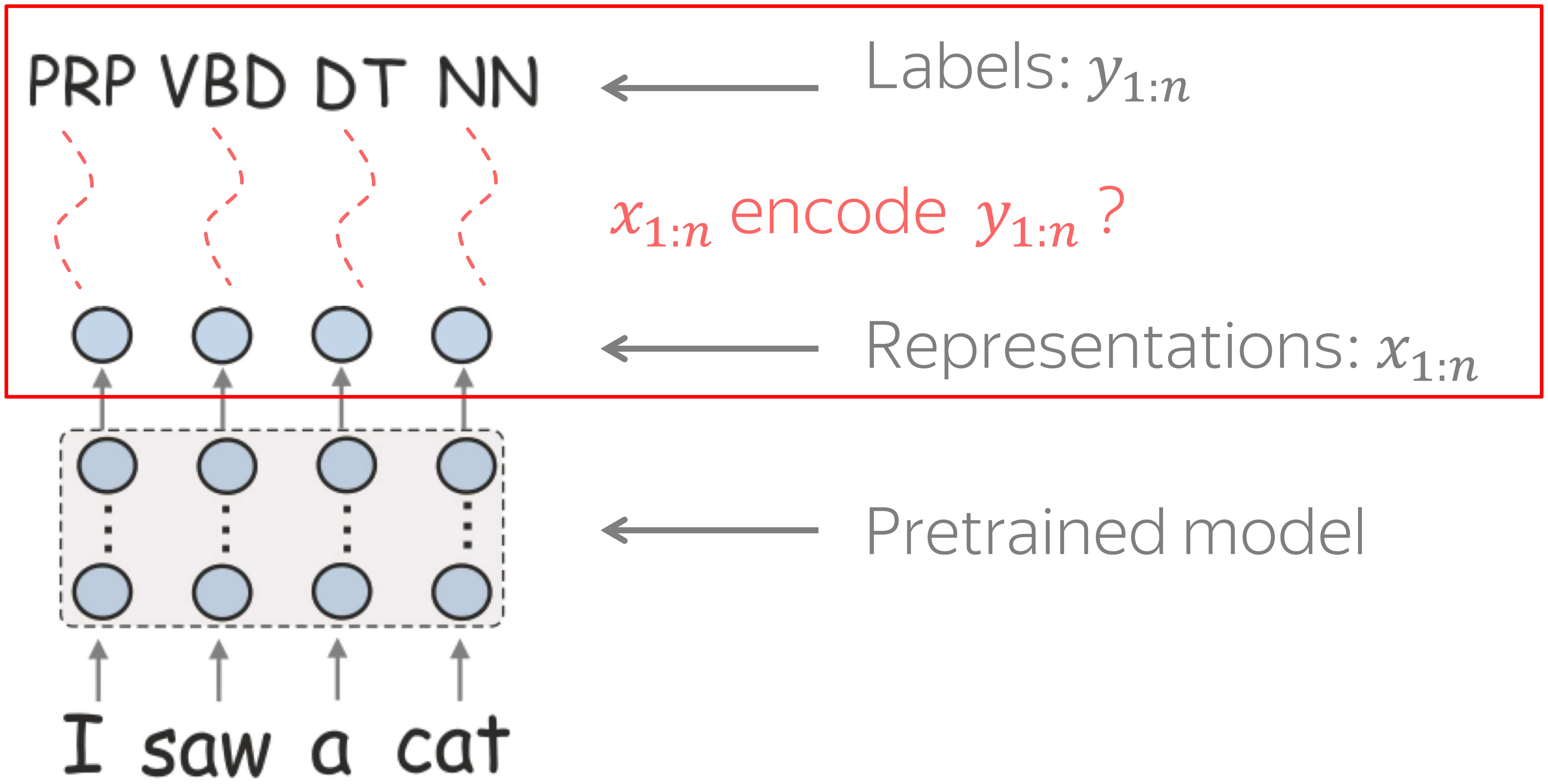
Labels: $y_{1:n}$

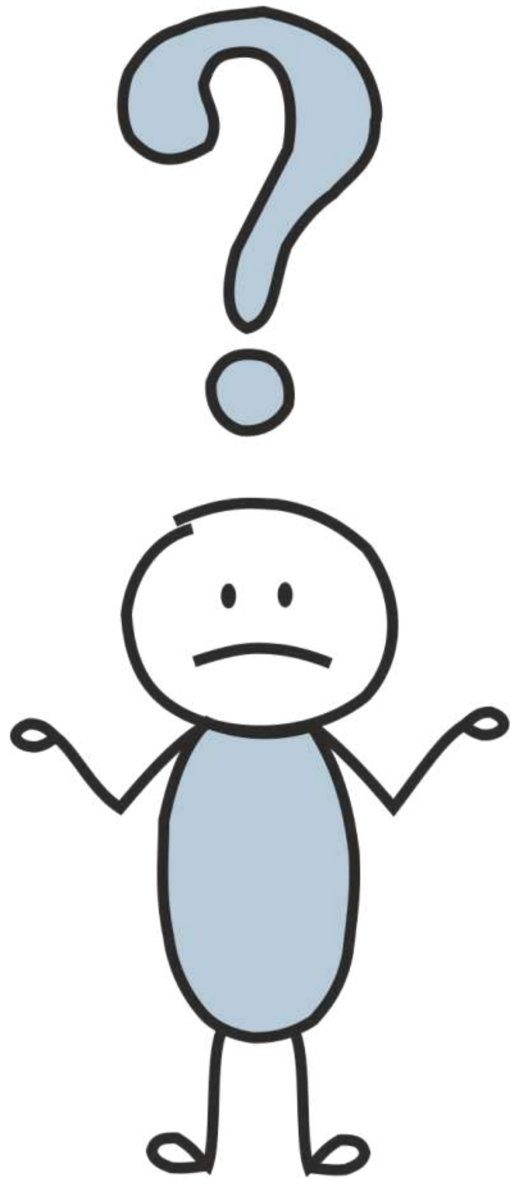
$x_{1:n}$ encode $y_{1:n}$?

Representations: $x_{1:n}$

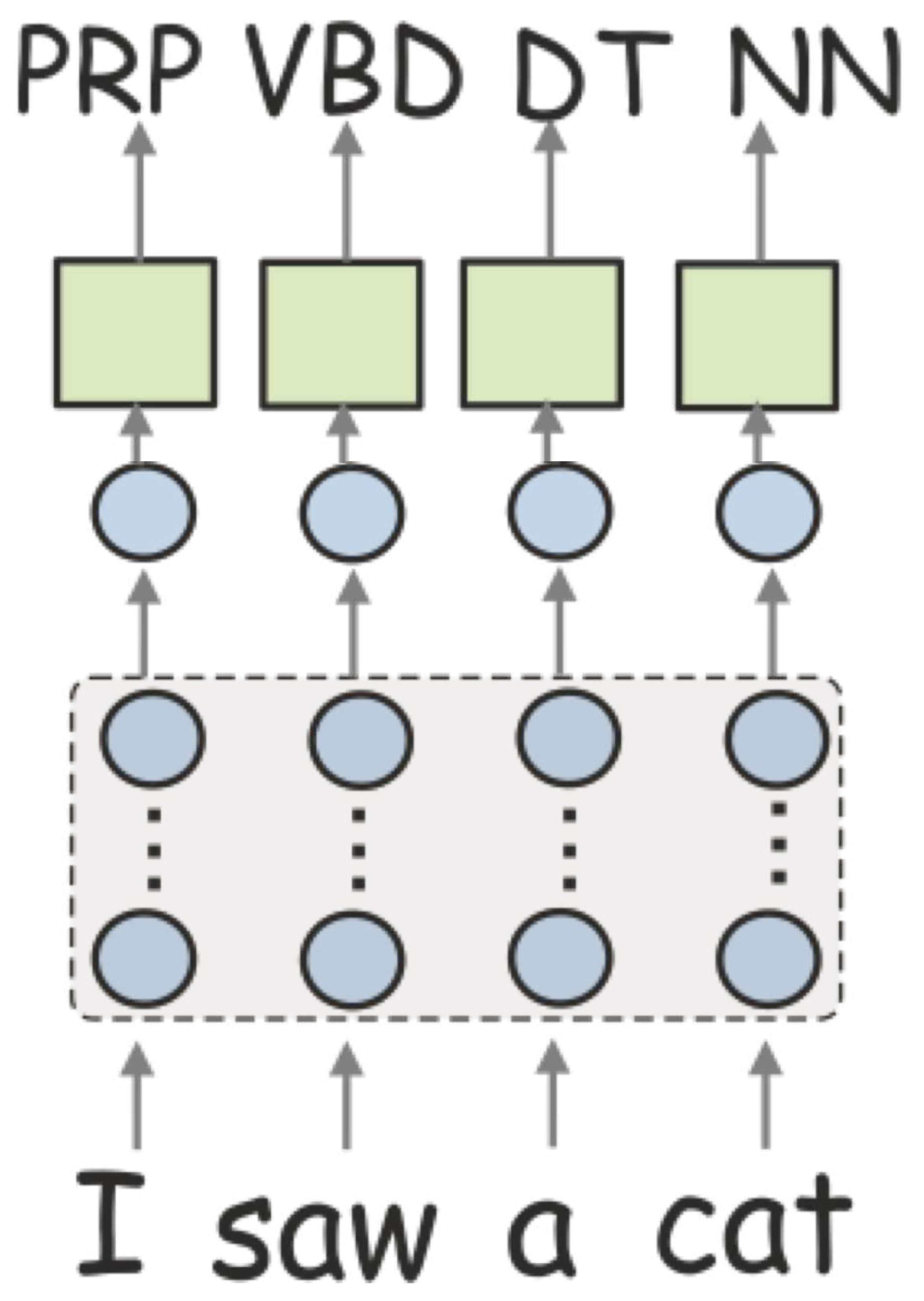
Pretrained model

How to understand if a model captures a linguistic property?





Standard Probing: train a classifier, use its accuracy



Labels: $y_{1:n}$

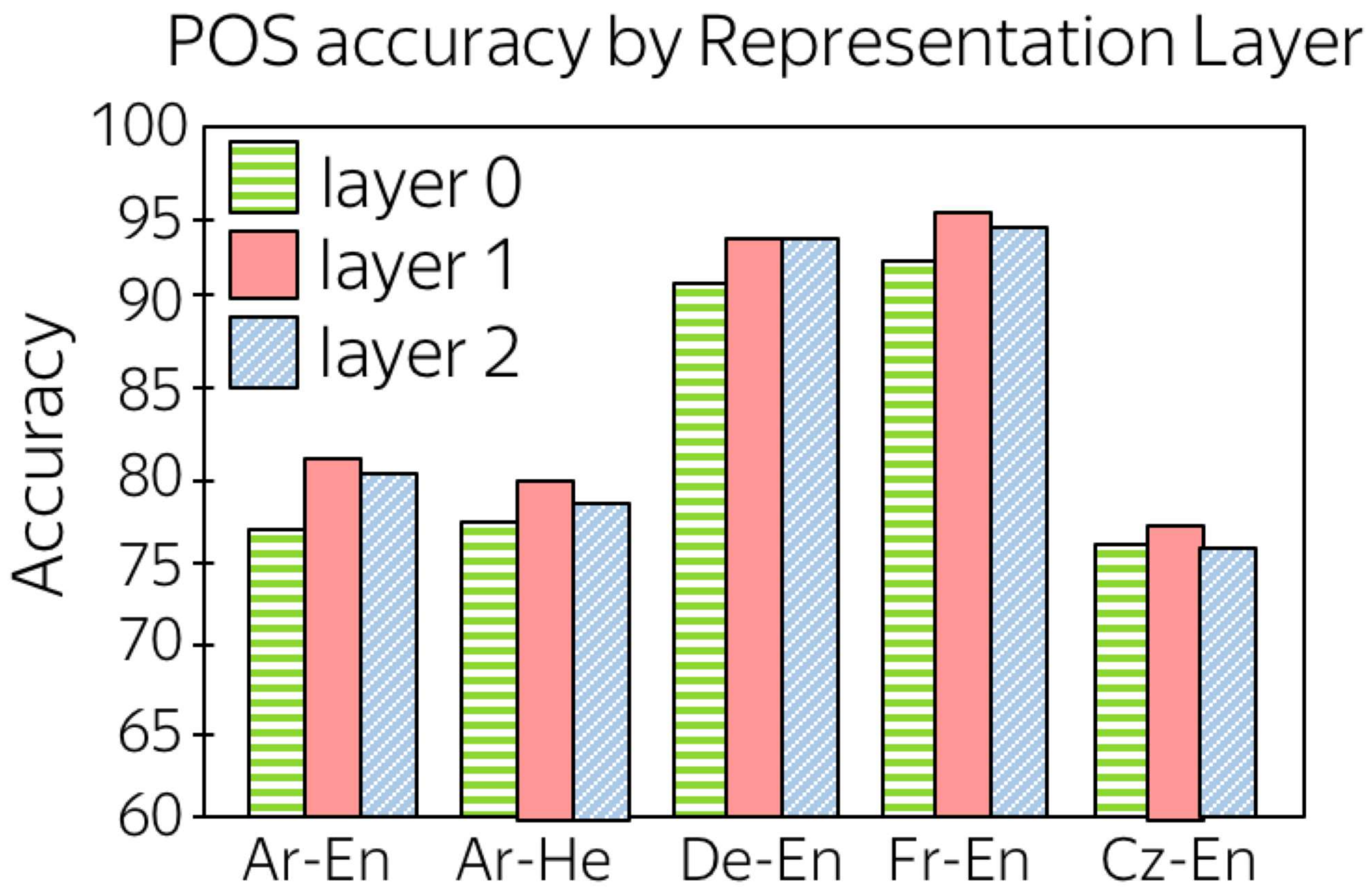
Probing classifier

Representations: $x_{1:n}$

Pretrained model

What Do NMT Models Learn About Morphology?

- Take NMT models for different language pairs
- Look at the encoder layers



Paper: What do Neural Machine Translation Models Learn about Morphology?

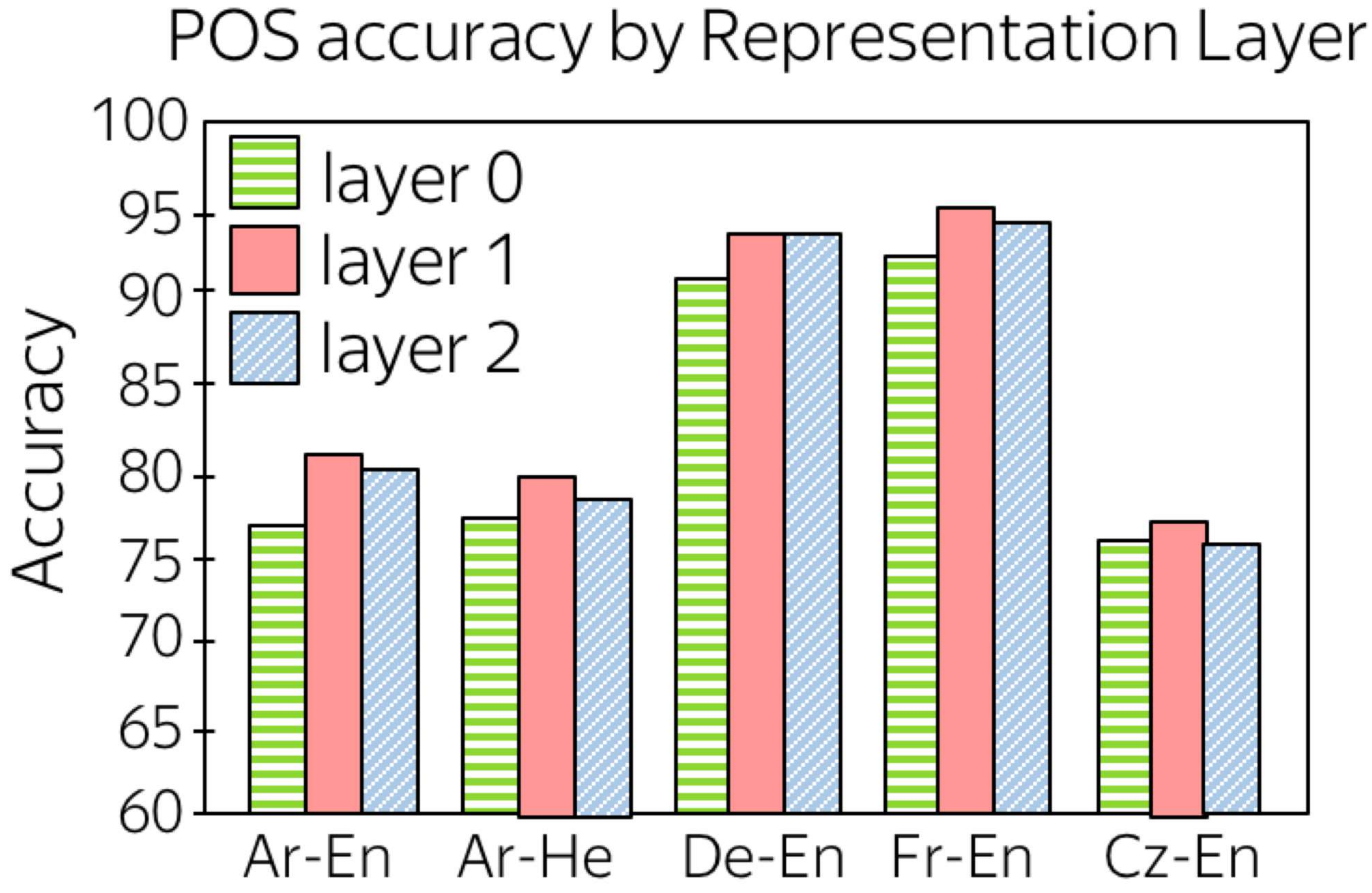
What Do NMT Models Learn About Morphology?

- Take NMT models for different language pairs

- Look at the encoder layers

<u>Results:</u>

  - Encoding helps: layer 0 (word embeddings) is the worst



Paper: What do Neural Machine Translation Models Learn about Morphology?

What Do NMT Models Learn About Morphology?

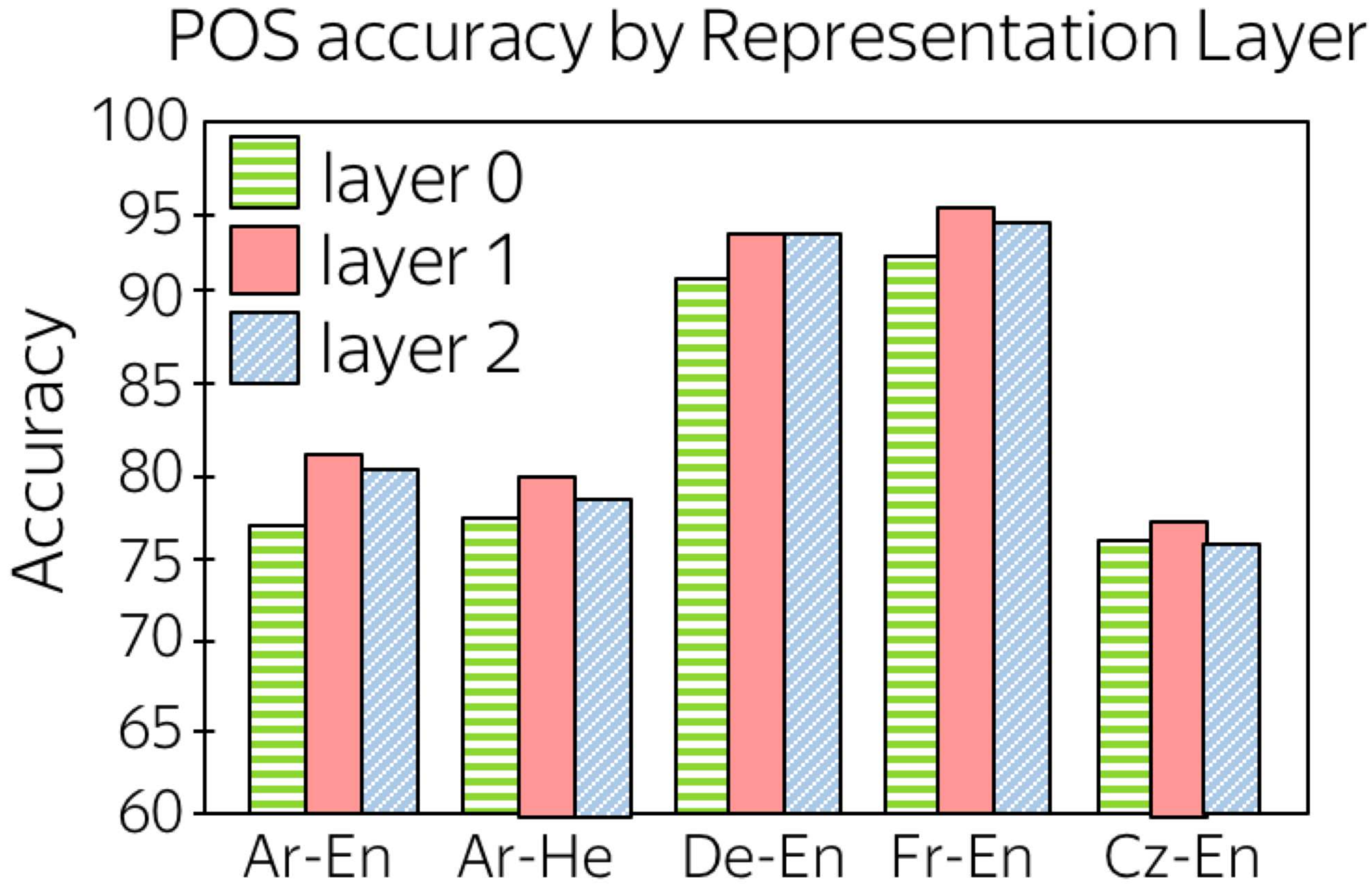
- Take NMT models for different language pairs

- Look at the encoder layers

Results:

- Encoding helps: layer 0 (word embeddings) is the worst

- Layer 1 is better than layer 2

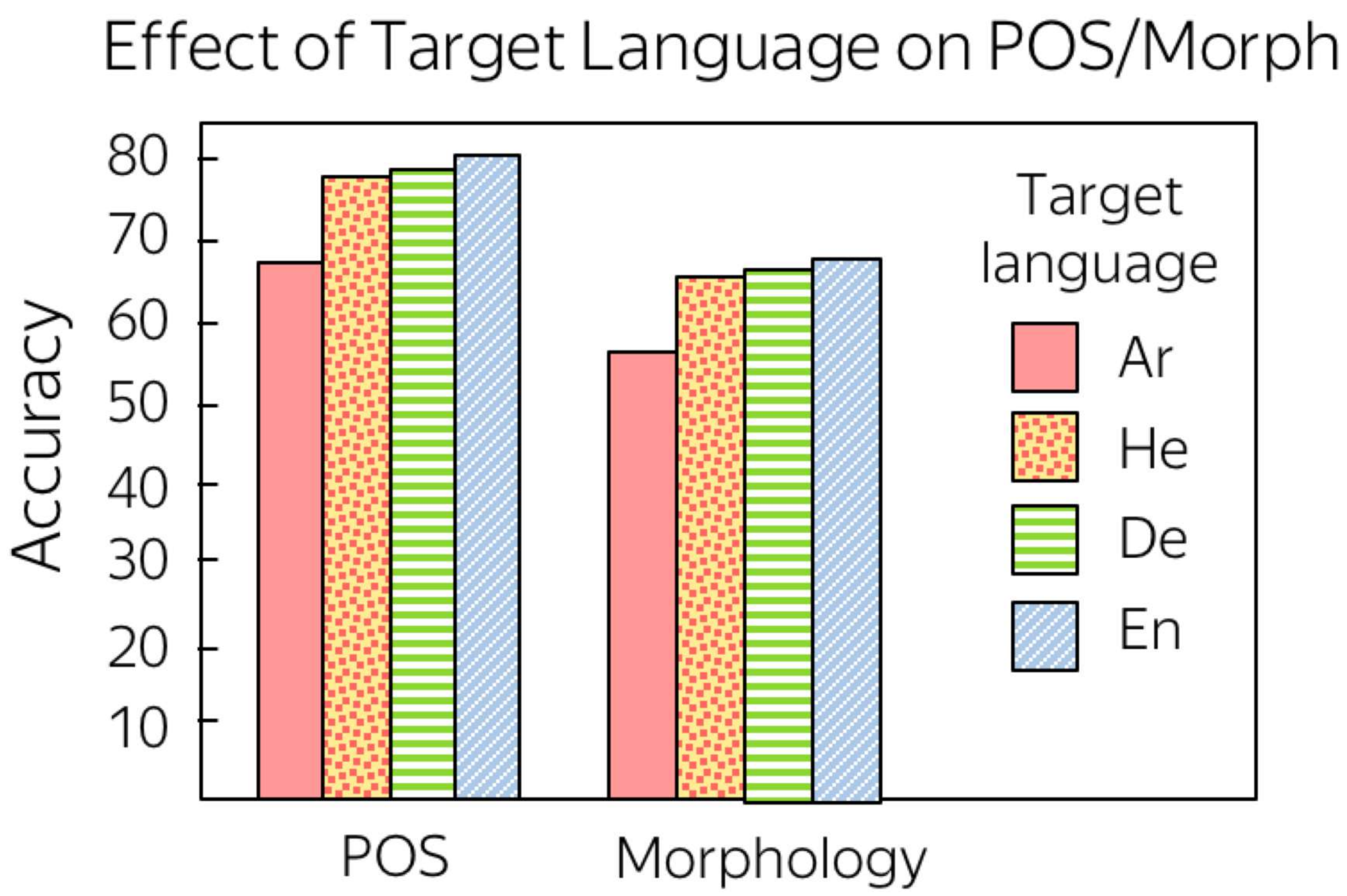


Paper: What do Neural Machine Translation Models Learn about Morphology?

What Do NMT Models Learn About Morphology?

- Take NMT models with the same source language and different target languages

- Look at the encoder



Paper: <u>What do Neural Machine Translation Models Learn about Morphology?</u>

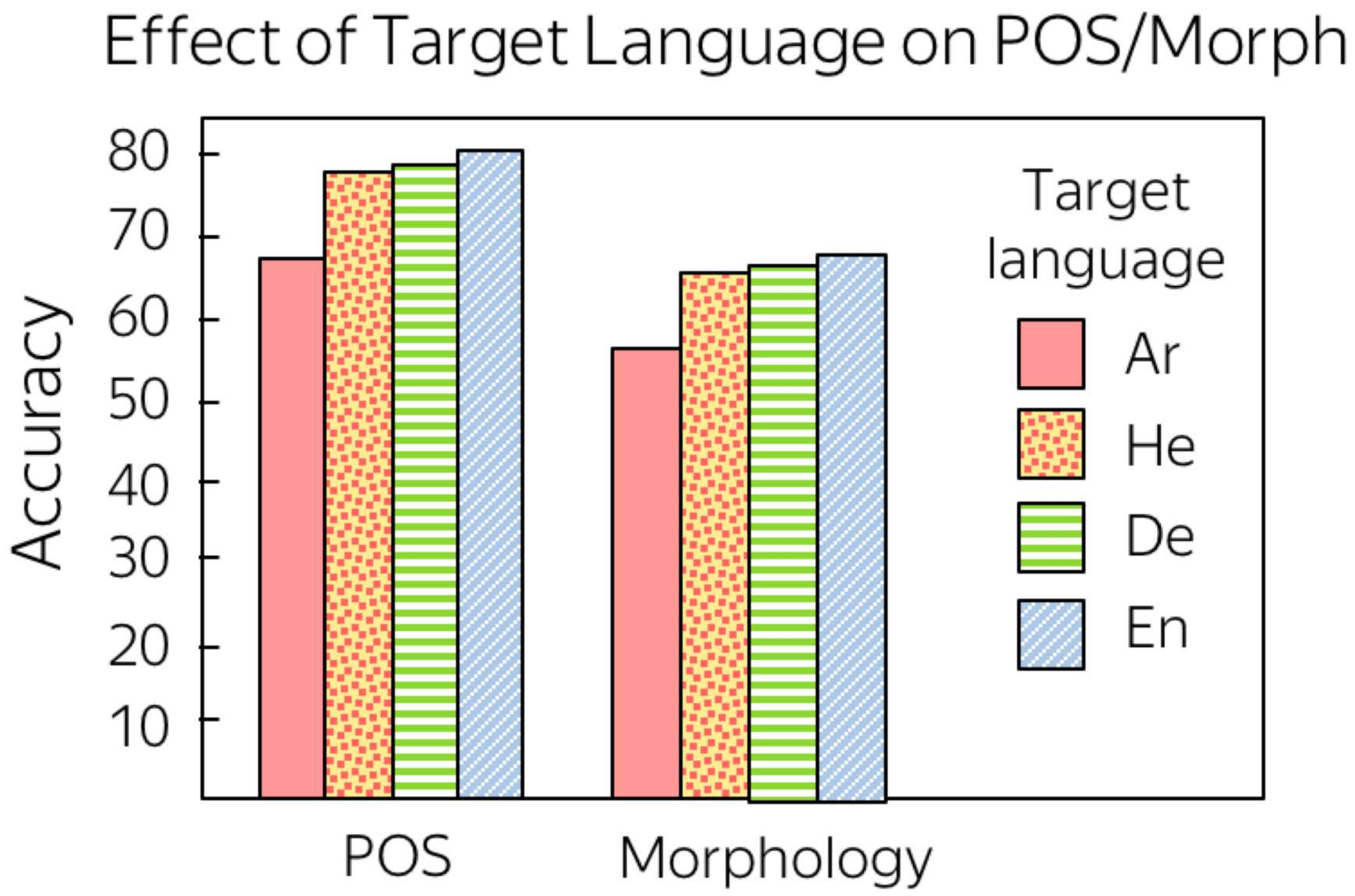
What Do NMT Models Learn About Morphology?

- Take NMT models with the same source language and different target languages

- Look at the encoder

Results:

- Surprising: weaker target morphology leads to stronger encoder



Paper: What do Neural Machine Translation Models Learn about Morphology?

What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
- Subword Segmentation: BPE
- Analysis and Interpretability

What is going to happen:

- Seq2seq Basics
- Attention
- Transformer
- Subword Segmentation: BPE
- Analysis and Interpretability

Learn more in the NLP Course For You → This is up to You!

Thank you!

Lena Voita

PhD student, Uni Edinburgh & Uni Amsterdam

Facebook PhD Fellow in NLP

lena-voita@hotmail.com

https://lena-voita.github.io

@lena_voita

lena-voita