GDAŃSK UNIVERSITY OF TECHNOLOGY

WYDZIAŁ ELEKTRONIKI, TELEKOMUNIKACJI I INFORMATYKI

Introduction to Transformers

BeLight

Magdalena Mazur-Milecka

15 maja 2025

Welcome

Bio of the presenter

Magdalena Mazur-Milecka, PhD, Eng

Assistant Professor at Gdańsk University of Technology

Research areas:

- Computer vision in biomedical applications
- eXplainable AI

Department of Biomedical Engineering

Agenda

1. **What are Transformers?**
   - Definition and Concept
   - What are Transformers?
   - Historical Context
2. Background Knowledge
3. Core Components
4. Examples

Definition and Concept

Transformer - proposed in 2017 by Google for text analysis: Natural Language Processing (translation, understanding, generation) - "Attention is Aall You Need."

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

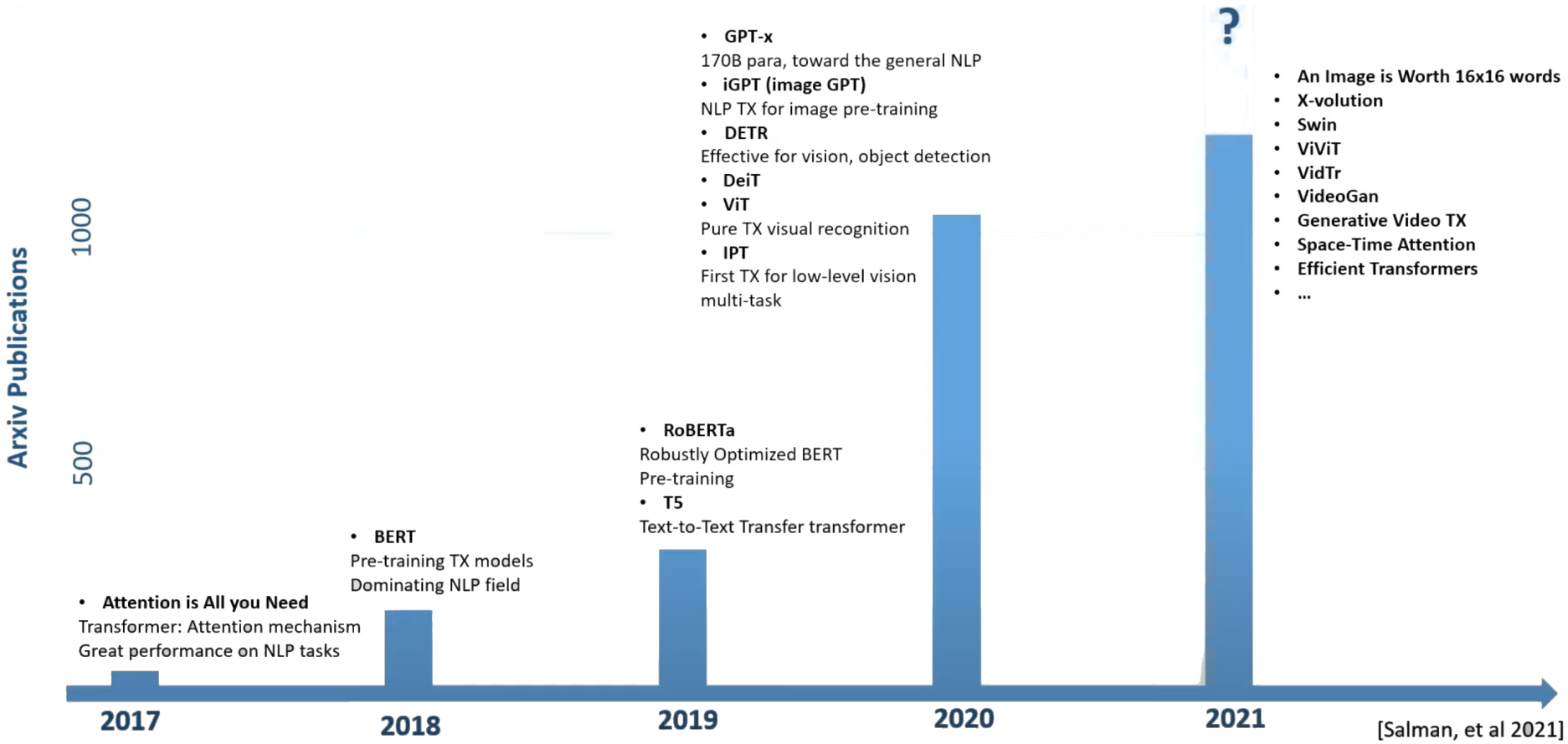
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

Historical Context



Historical Context

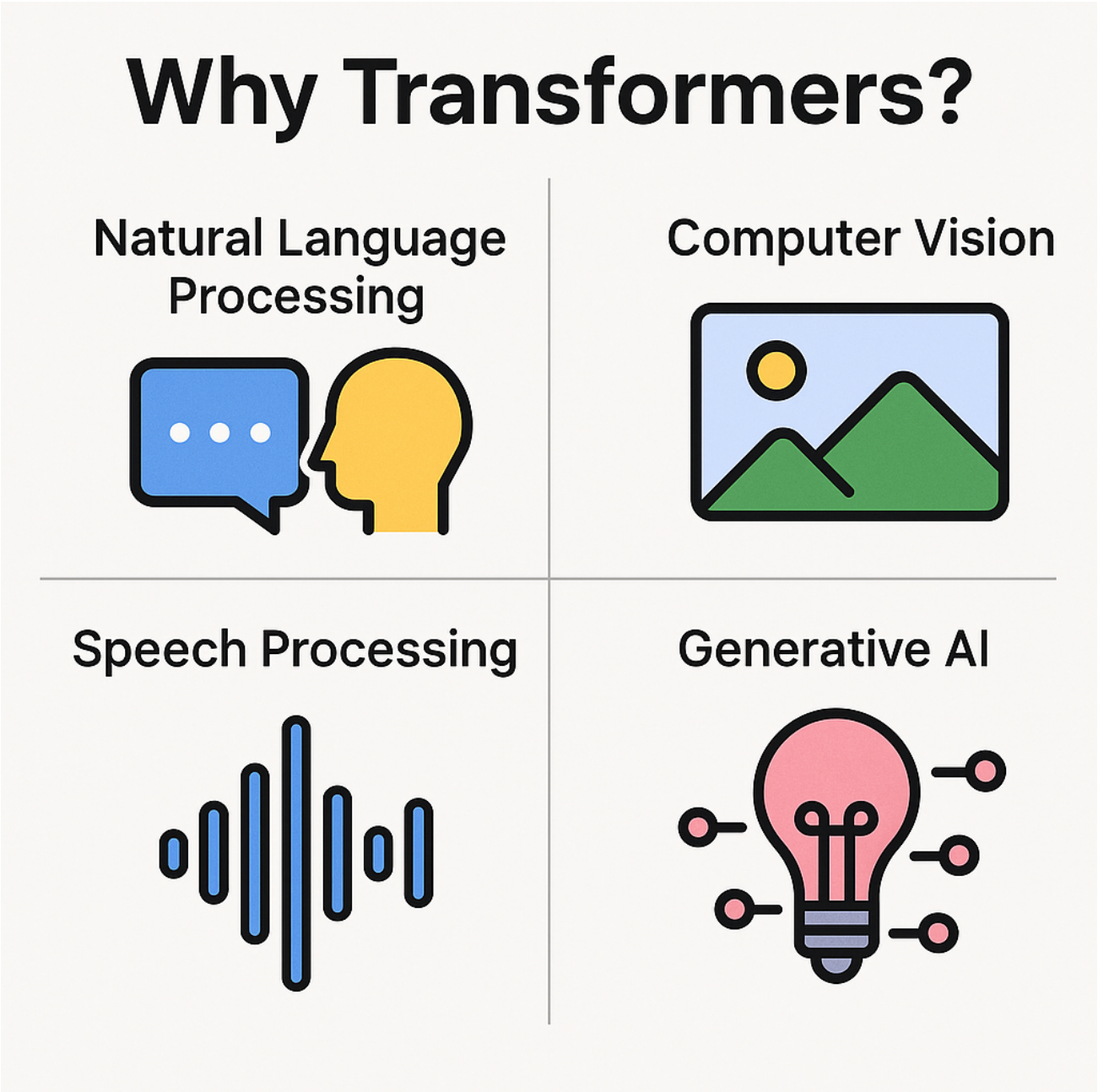
Revolution in NLP

The Transformer - "ImageNet Moment" for Natural Language Processing

Revolutions in Computer Vision

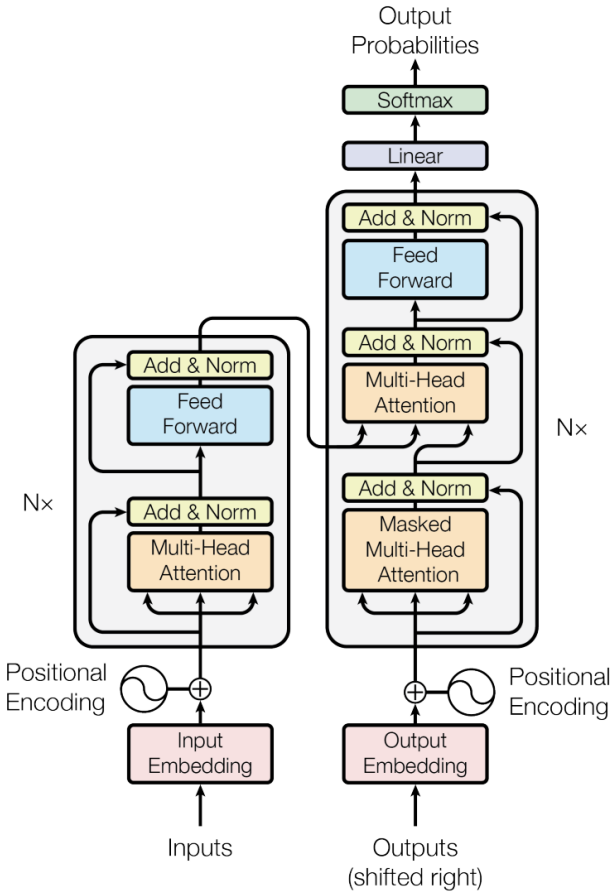
	Deep Learning	**Deep Learning 2.0**
Idea	Convolution (CNN)	Attention
Year	2012	2017 / 2020
Architecture	AlexNet	Transformers / ViT
Replacement for ...	Machine Learning, Standard CV Algorithms	CNN, RNN

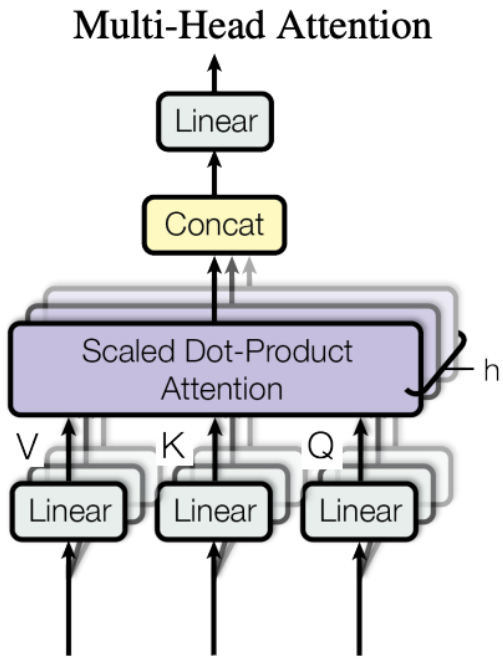
Why do we need them?



Definition and Concept

Transformer - basic idea:





[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

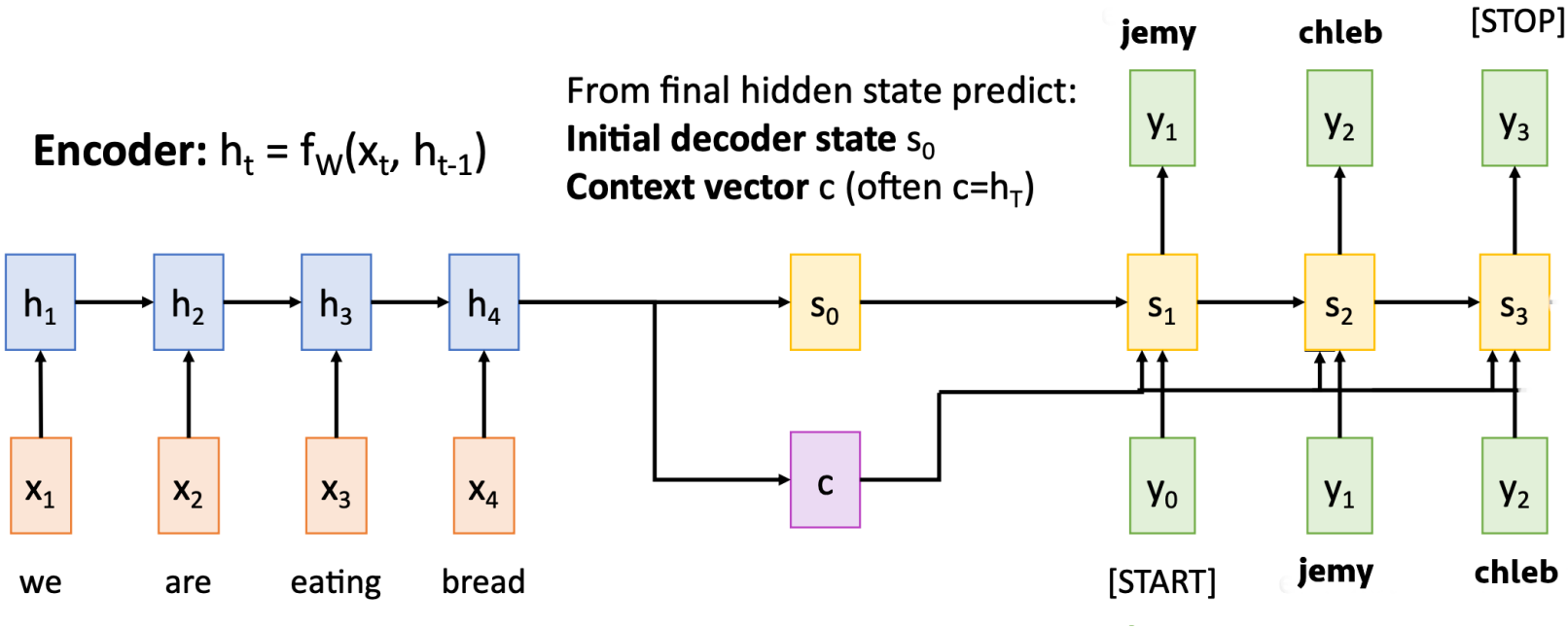
Agenda

2. Background Knowledge

- **Transformers - Origin and Evolution**
  - Introduction to the transformer model in NLP
- **Core Components of Transformers**
  - Attention mechanism
  - Self-attention
  - Multi-head Attention
  - Positional encoding
  - Masked Attention
  - Transformer block
  - Encoder-decoder architecture

Transformer Origin

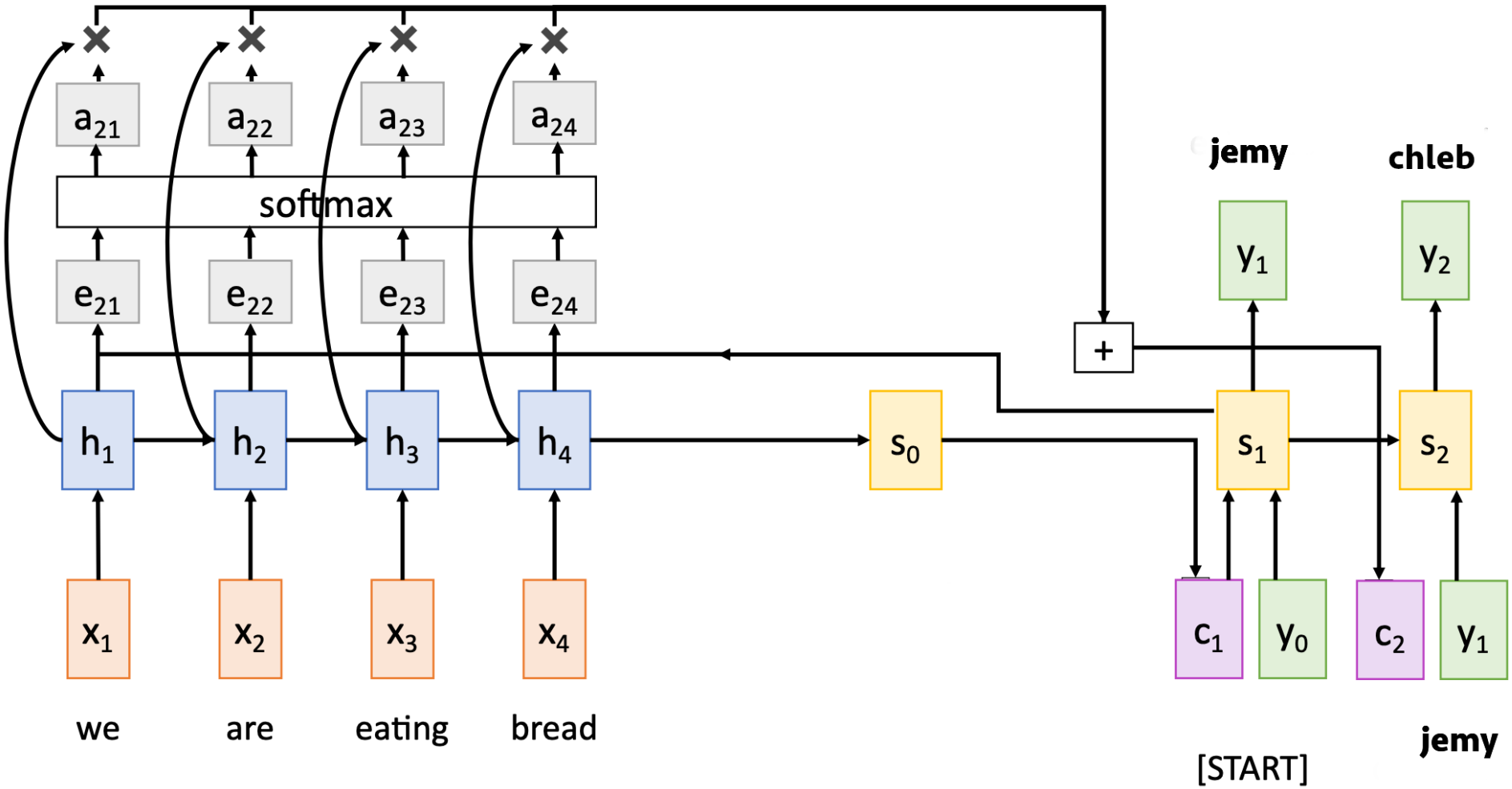
Standard approach to Seq2Seq analysis with RNN (encoder-decoder architecture)



Deep Learning for Computer Vision, Stanford lectures https://cs231n.stanford.edu

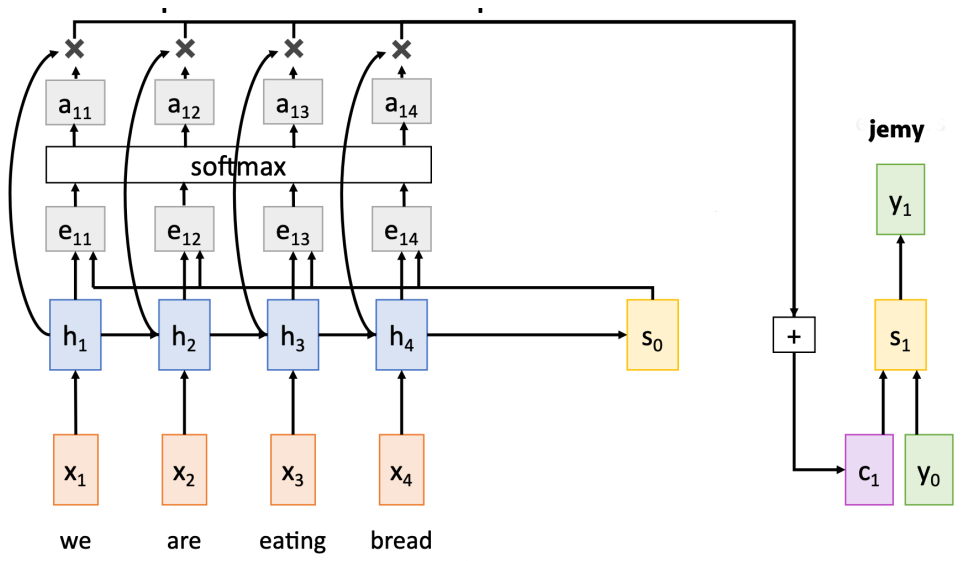
Transformer Origin

Introducing attention to Seq2Seq analysis with RNN



Deep Learning for Computer Vision, Stanford lectures
https://cs231n.stanford.edu

Transformer Origin



- s_0 - initial state of the decoder created based on the last hidden state of the encoder,
- $e_{t,i} = f(s_{t-1}, h_i)$ - alignment score, f is MLP (Multilayer Perceptron),
- softmax - normalization to attention weights [0,1],
- calculating the context vector ($c_t = \sum a_{t,i} h_i$) for each decoder input separately as a linear combination of hidden states (weighted by attention),
- e.g., for c_1 "jemy" the most important are x_1, x_2 and x_3 "we are eating", so a_{11}=0.3 and a_{12}= 0.15 and a_{13}=0.5, and a_{14}= 0.05

Attention

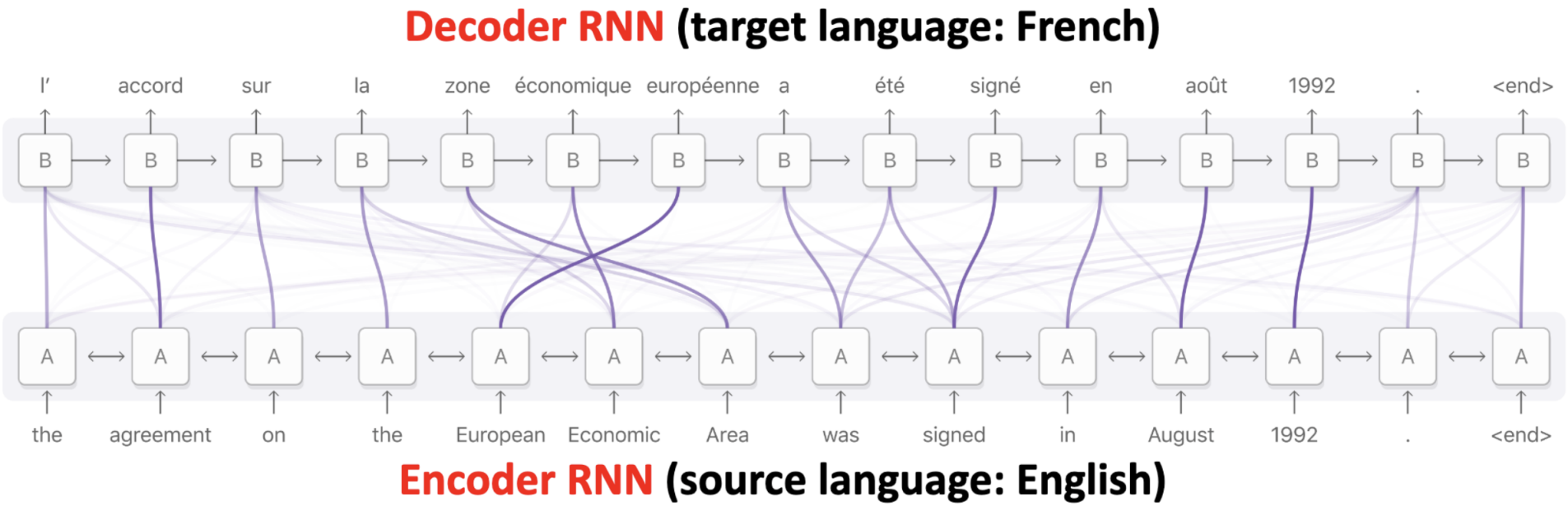


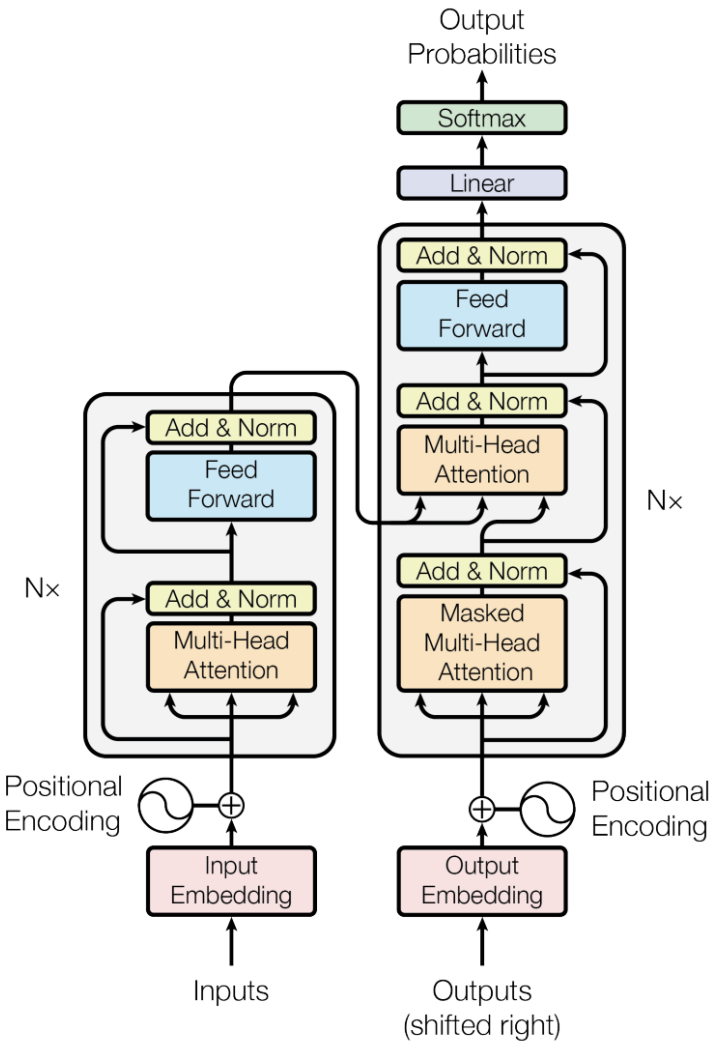
Figure is from https://distill.pub/2016/augmented-rnns/

Attention

What changed with the introduction of Attention in Seq2Seq analysis:

- At each decoding stage (creating word translation) a different *c* (context) vector is considered,
- This vector is calculated based on all the hidden states of the encoder, not just the last one,
- The weights *a* are different for different output words and reflect which input words x should be given "attention",
- More calculations - *a* calculated for each input vector,
- Understanding text/image in line with the natural process of unevenly distributed attention,
- The Attention mechanism itself does not treat the input as a sequence.

Attention is all you need



- Eliminating RNN - replaced by the Attention layer,
- Seq2Seq model with encoder-decoder architecture,
- Based on Attention and Self-Attention mechanisms,

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

Attention Layer

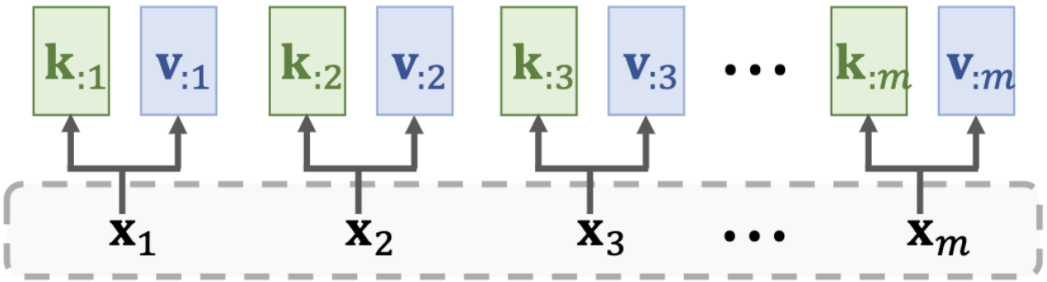
The encoder input vector is transformed into:

- Key: $k_i = W_K x_i$ - identifier of proposed (interesting) places; needed to calculate Attention,
- Value: $v_i = W_V x_i$ - the actual information the model uses after the attention scores are computed; needed to calculate the output.

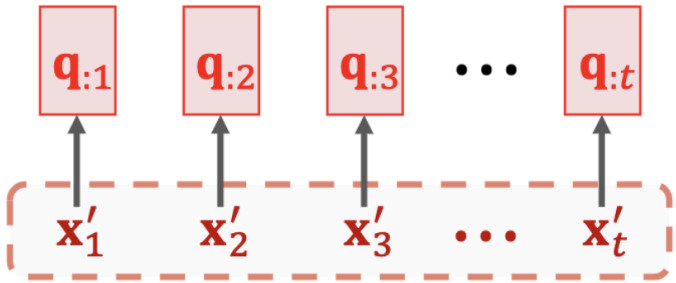
The decoder input vector is transformed into:

- Query: $q_j = W_Q x'_j$ - questions, current token that is being processed

Matrices W_K, W_V, W_Q are learned parameters during training of the encoder and decoder.



encoder



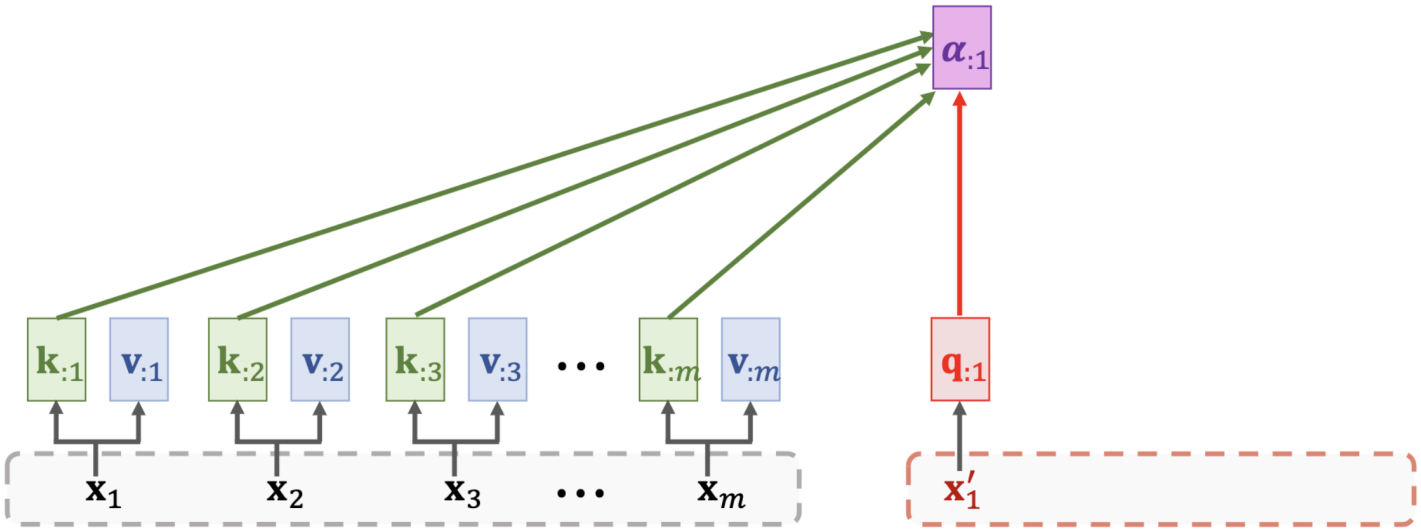
decoder

Attention Layer

We calculate attention using X in the form of K :

$$\alpha_{:1} = Softmax(K^T q_{:1})$$
$$\alpha_{:j} = Softmax(K^T q_{:j})$$

Dot product $K^T Q$ - comparison of vectors
$\alpha_{:1}$ - m-dimensional vector (Attention)

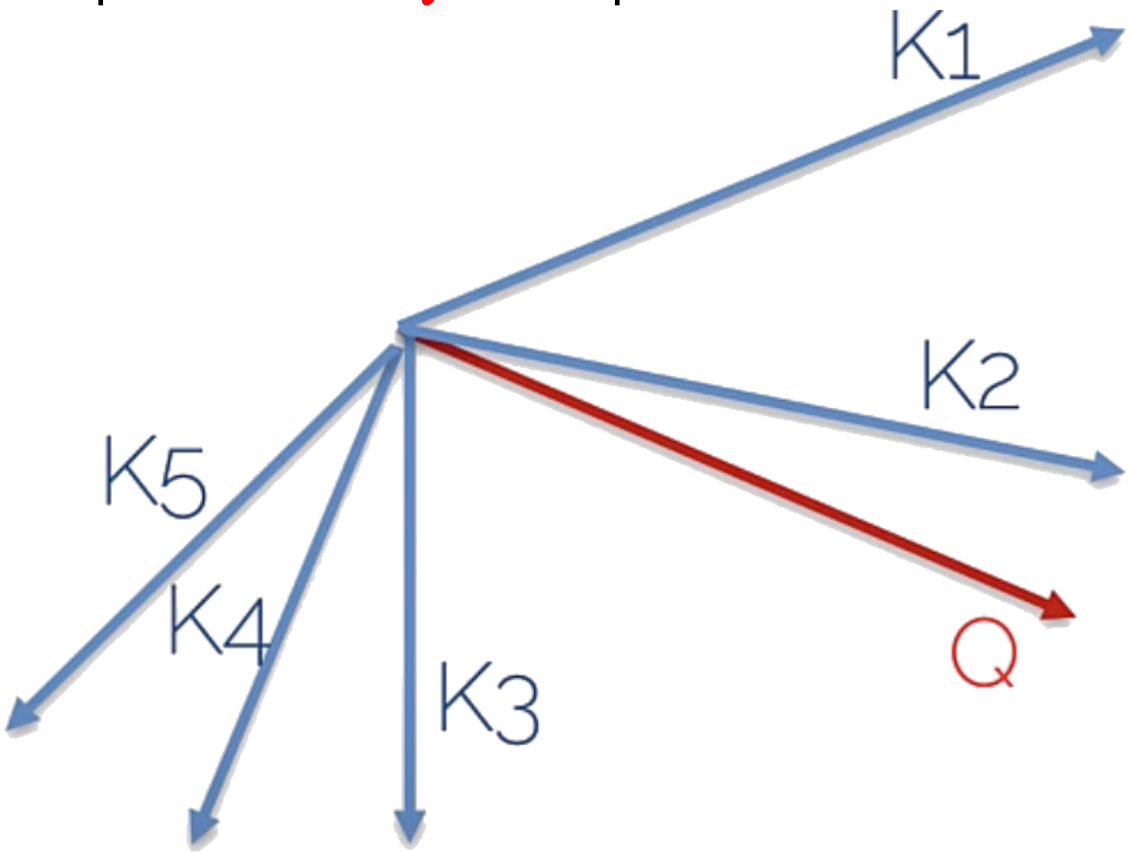


Attention Layer

We calculate attention (weighting coefficients for vector V) using X in the form of K :

$$\alpha_{:1} = Softmax(K^T q_{:1})$$
$$\alpha_{:j} = Softmax(K^T q_{:j})$$

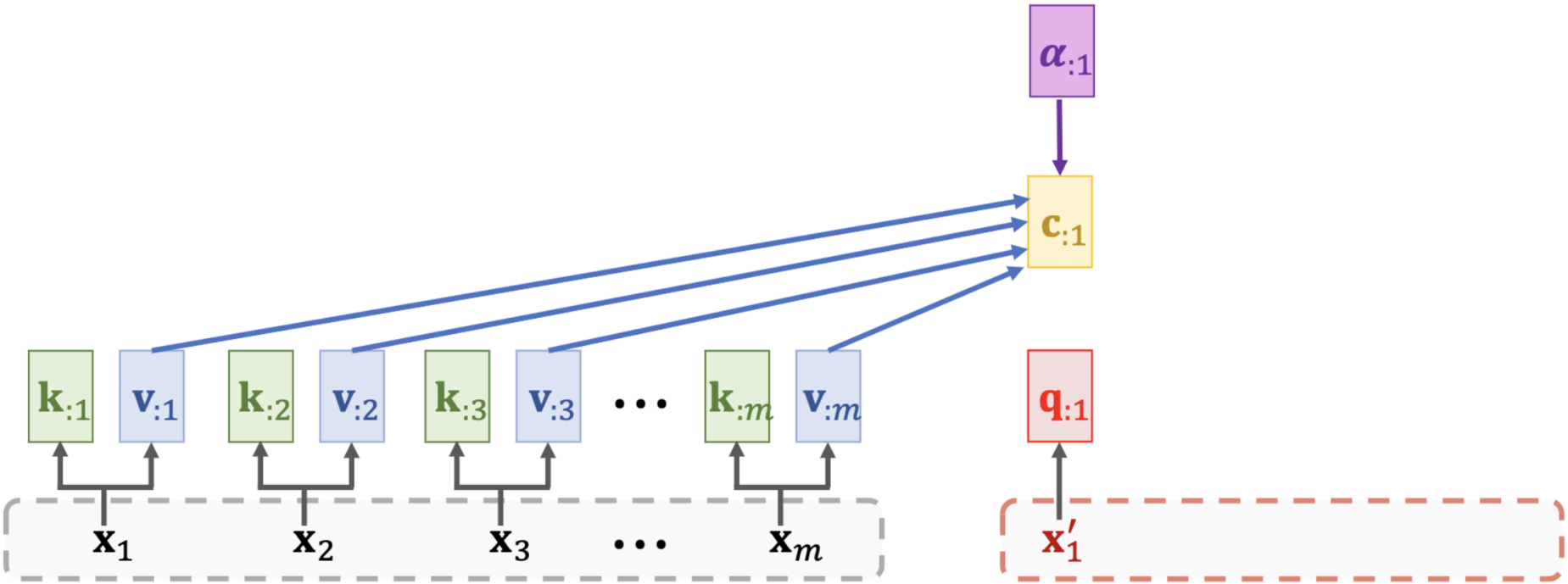
Dot product $K^T Q$ - comparison of vectors



Attention Layer

Calculated context vector:

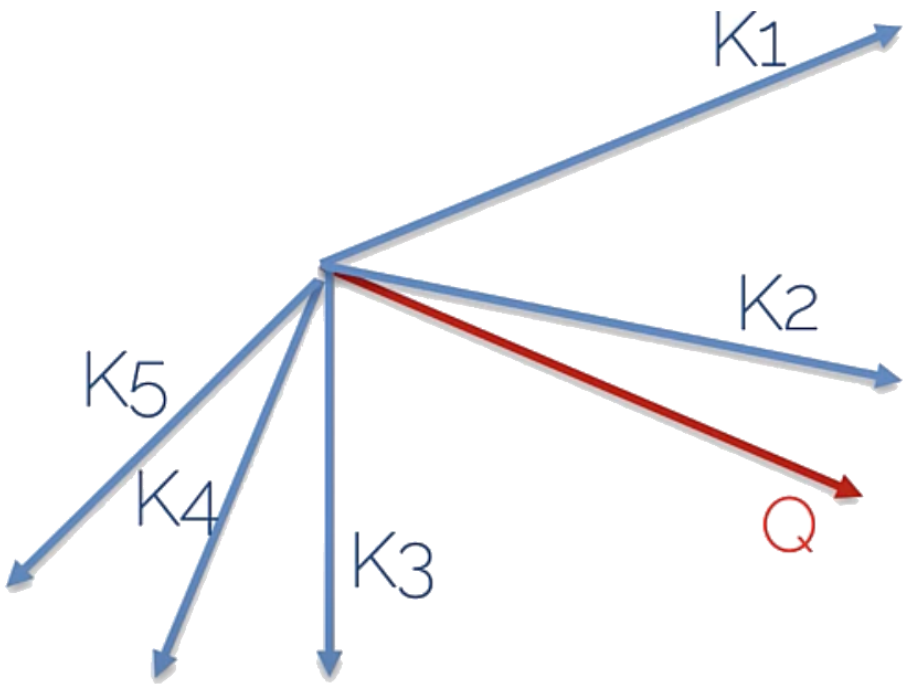
$c_{:1} = \alpha_{11} v_{:1} + \ldots + \alpha_{m1} v_{:m}$



Attention Layer

Calculated context vector:

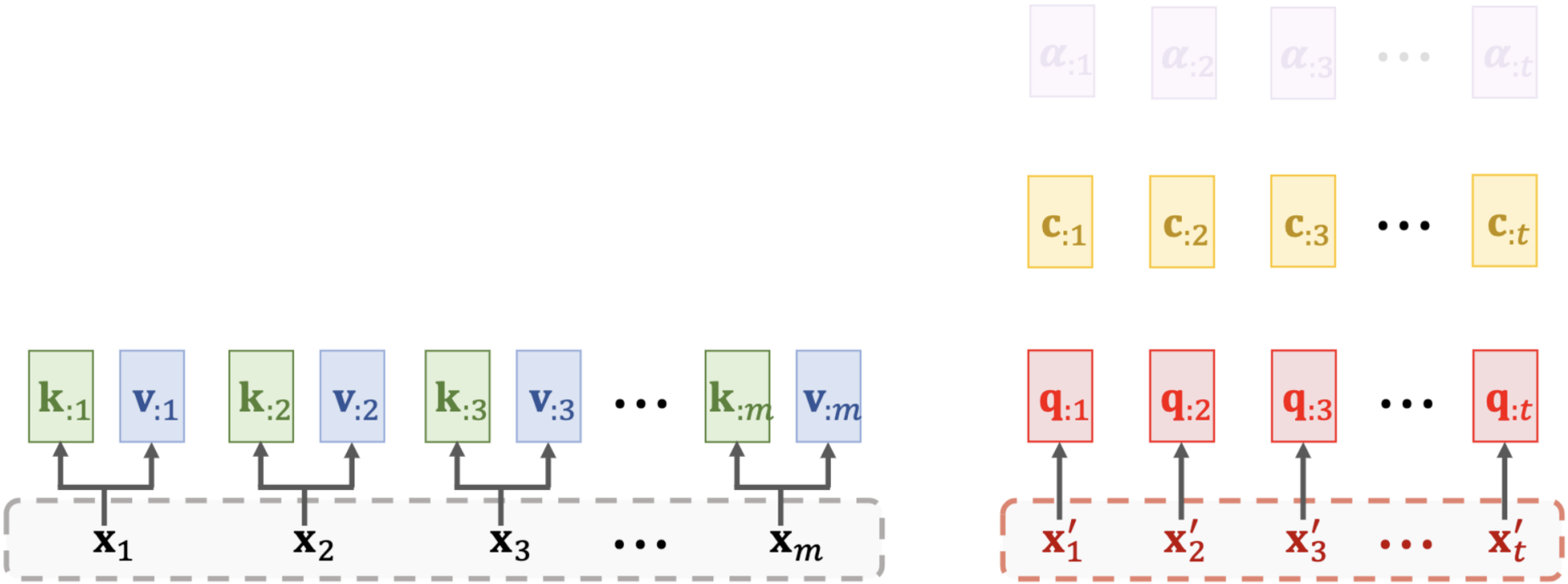
$c_{:1} = \alpha_{11} v_{:1} + \ldots + \alpha_{m1} v_{:m}$



Values
V1
V2
V3
V4
V5

Attention Layer

The length of vector c is the same as the length of the decoder input vector.
To calculate $c_{:3}$ all "Key", all "Value" and one "Query": $q_{:3}$ are considered

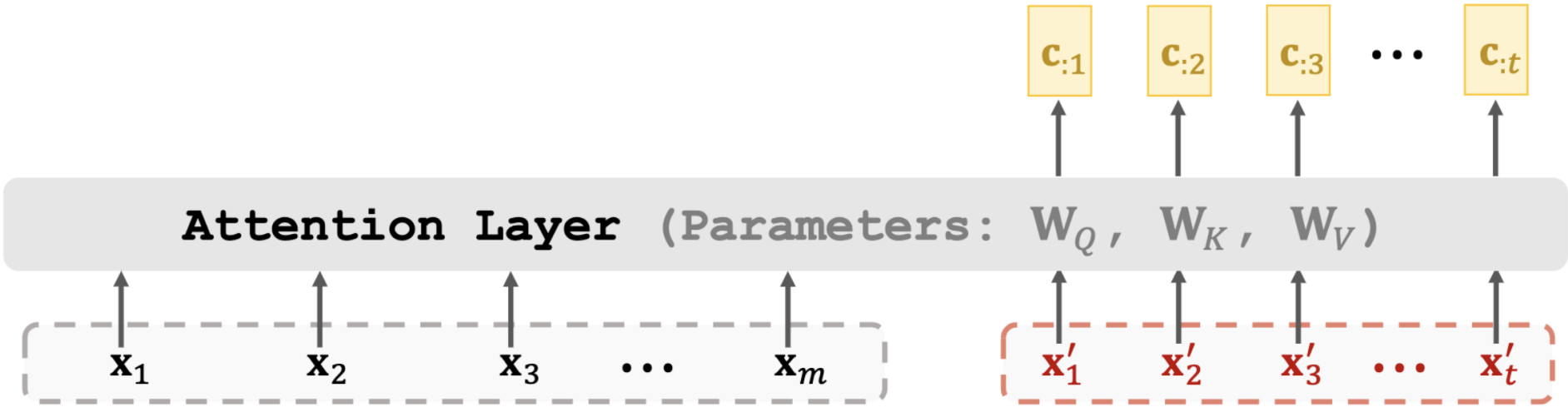


Attention

Attention Layer

$$C = Att(X, X') \qquad Att(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

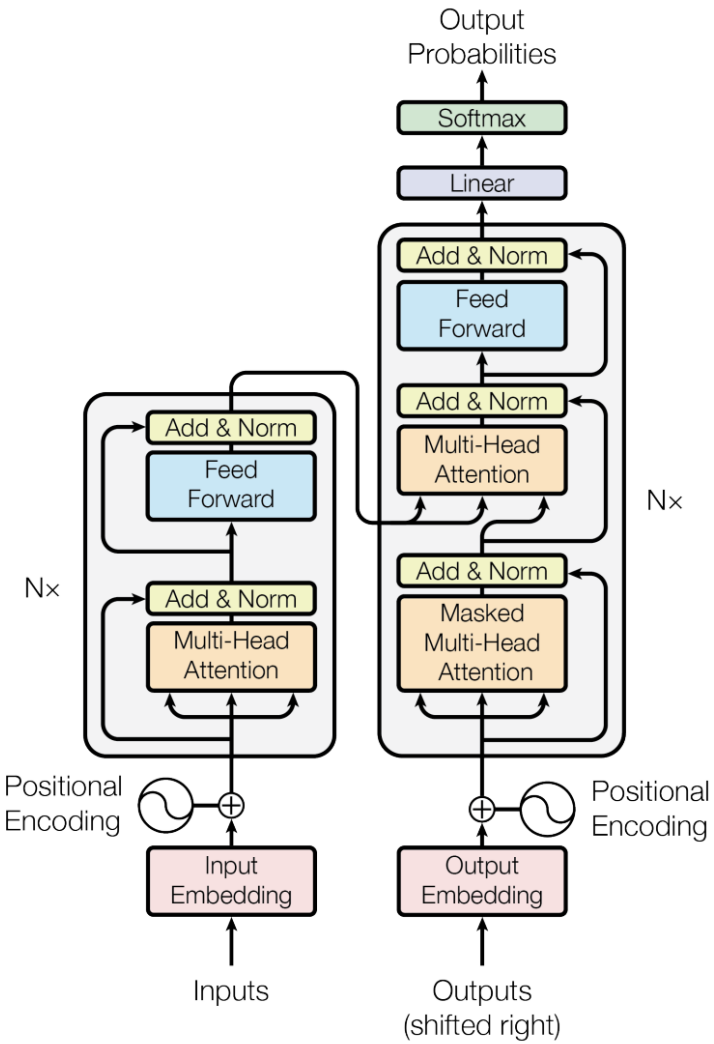
Encoder input $X = [x_1 ... x_m]$ - phrase to be translated
Decoder input $X' = [x'_1 ... x'_t]$ - phrase in the target language



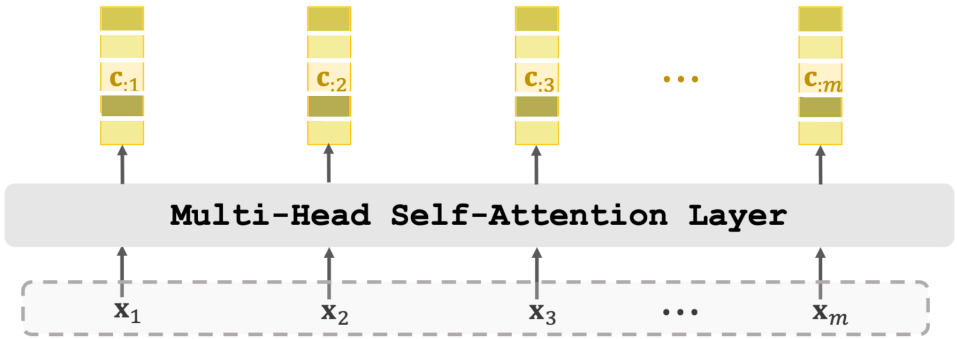
Demo

TransKQV.ipynb

Multi-head Attention

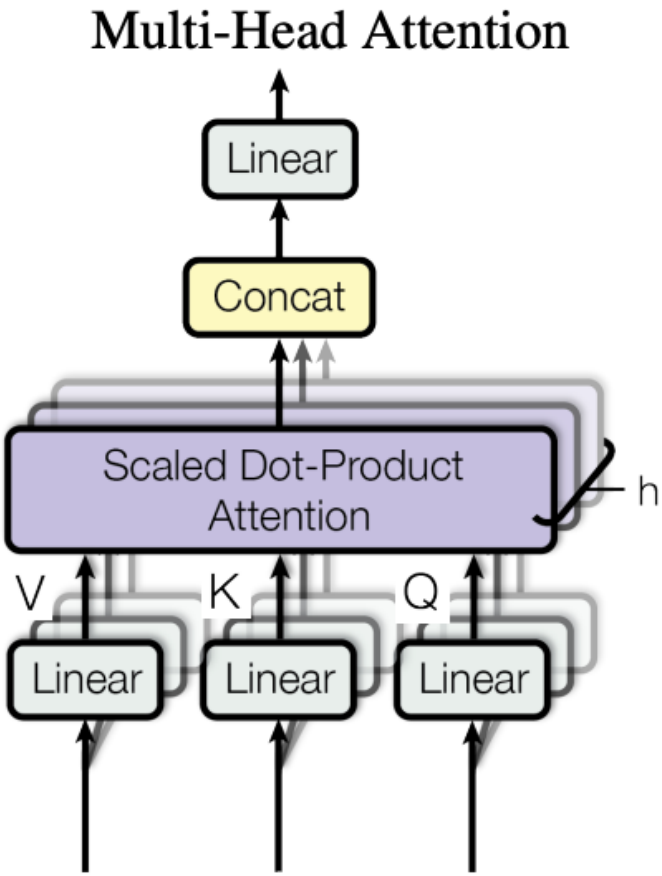


- Using the self-attention layer multiple times (on the same vector X), the layers do not share parameters,



A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

Multi-head Attention

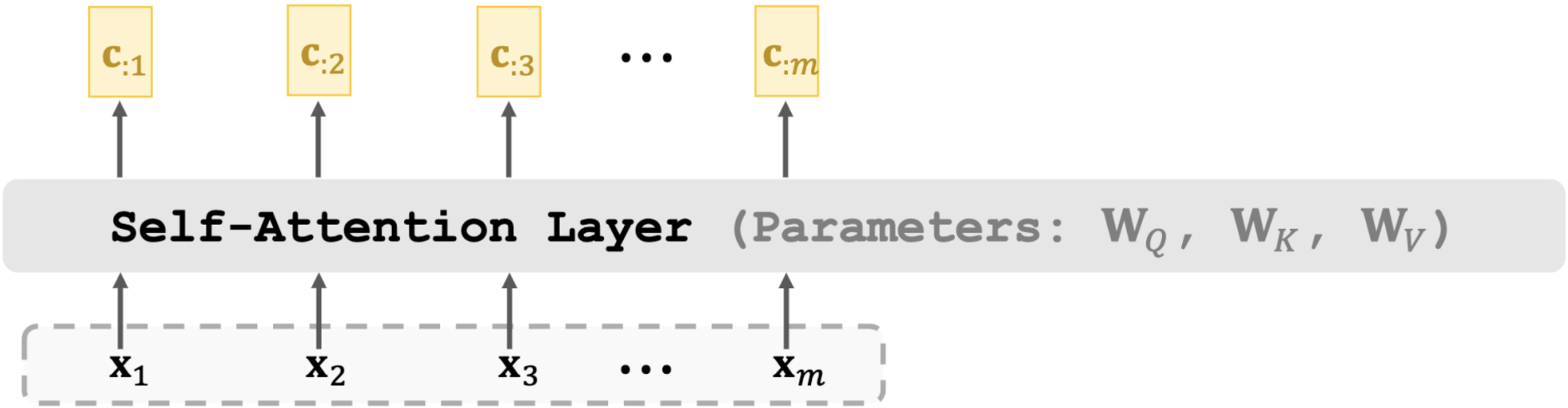


- **h** parameter - number of heads (8)
- different heads try to perform different task
- the output vectors are concatenated

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

Self-Attention

- Decoupling the attention mechanism from Seq2Seg model,
- Self-Attention layer takes a single input vector X,
- Calculates Attention within X,
- $C = Att(X, X)$

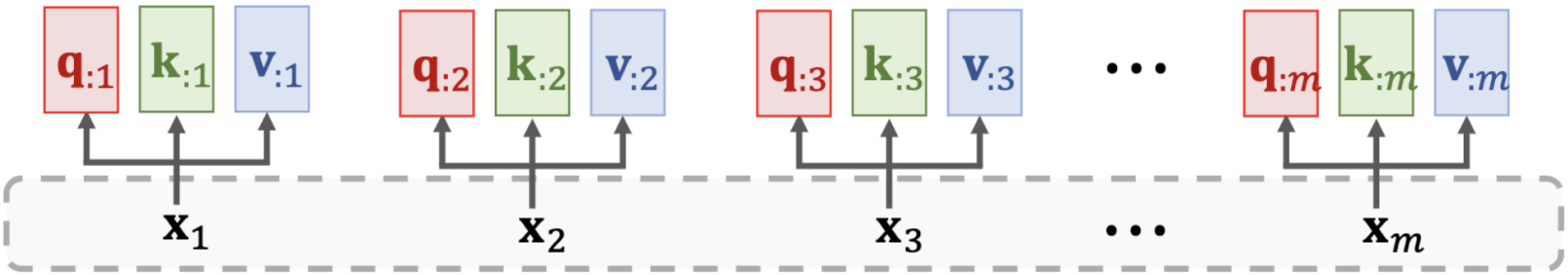


Self-Attention

The input vector is transformed into:

- Query: $q_j = W_Q x_i$
- Key: $k_i = W_K x_i$
- Value: $v_i = W_V x_i$

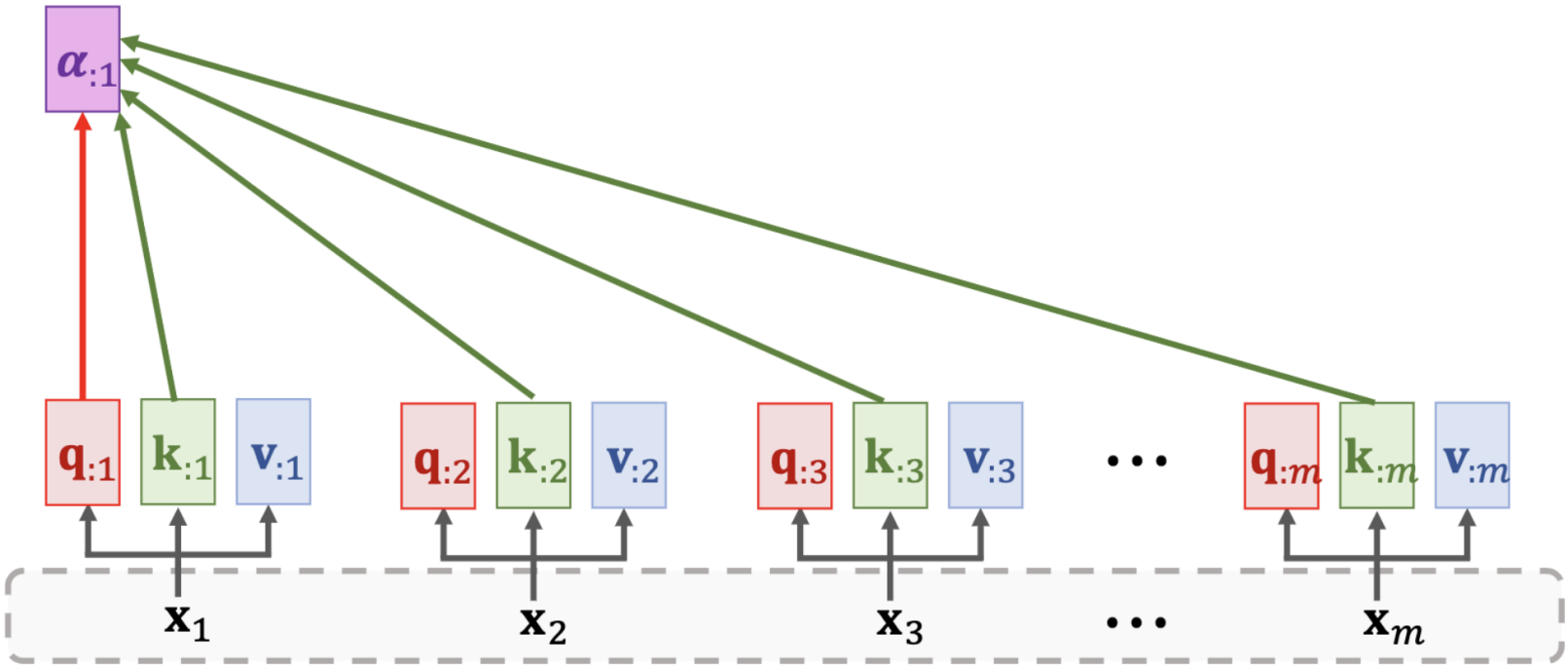
Matrices W_K, W_V, W_Q are learned parameters during training.



Self-Attention

We calculate attention using X in the form of K and a $q_{:1}$:

$$\alpha_{:1} = Softmax(K^T q_{:1})$$
$$\alpha_{:i} = Softmax(K^T q_{:i})$$

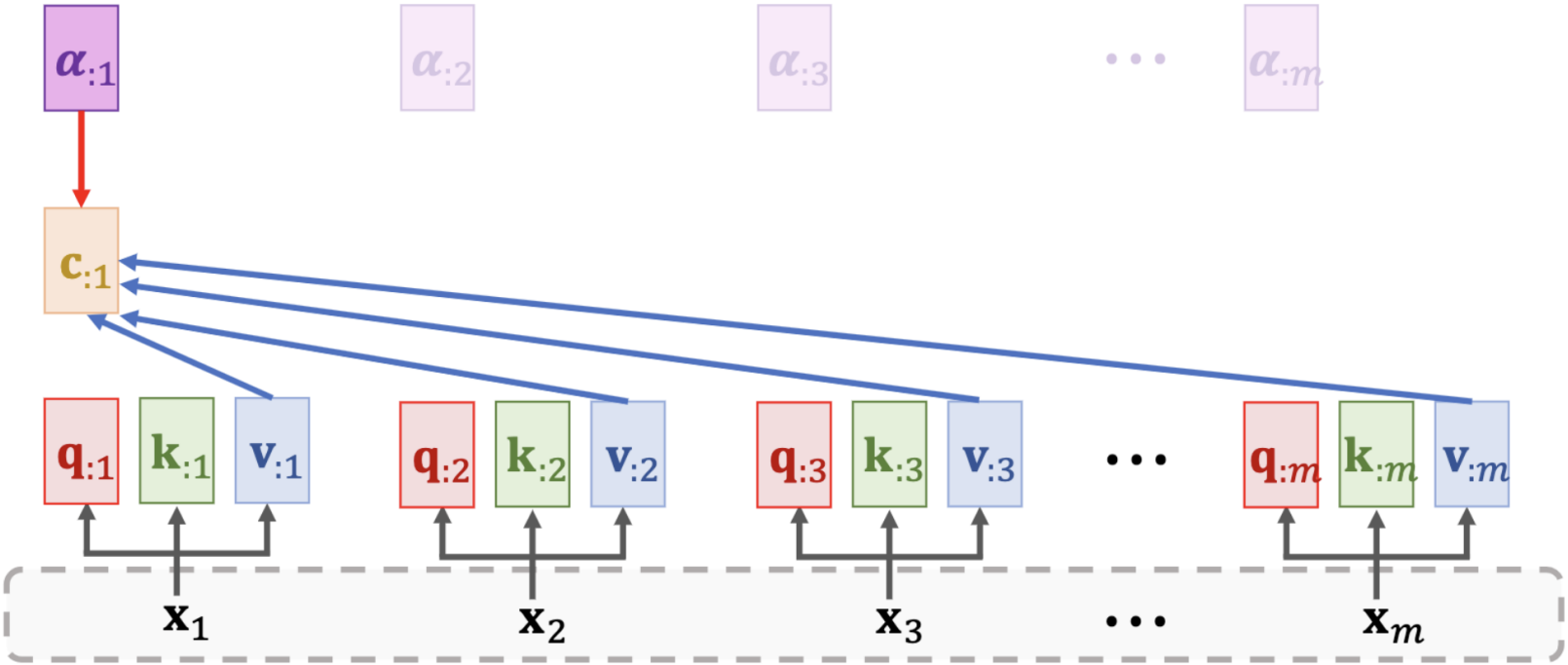


Self-Attention

Calculated context vector:

$c_{:1} = \alpha_{11} v_{:1} + ... + \alpha_{m1} v_{:m}$

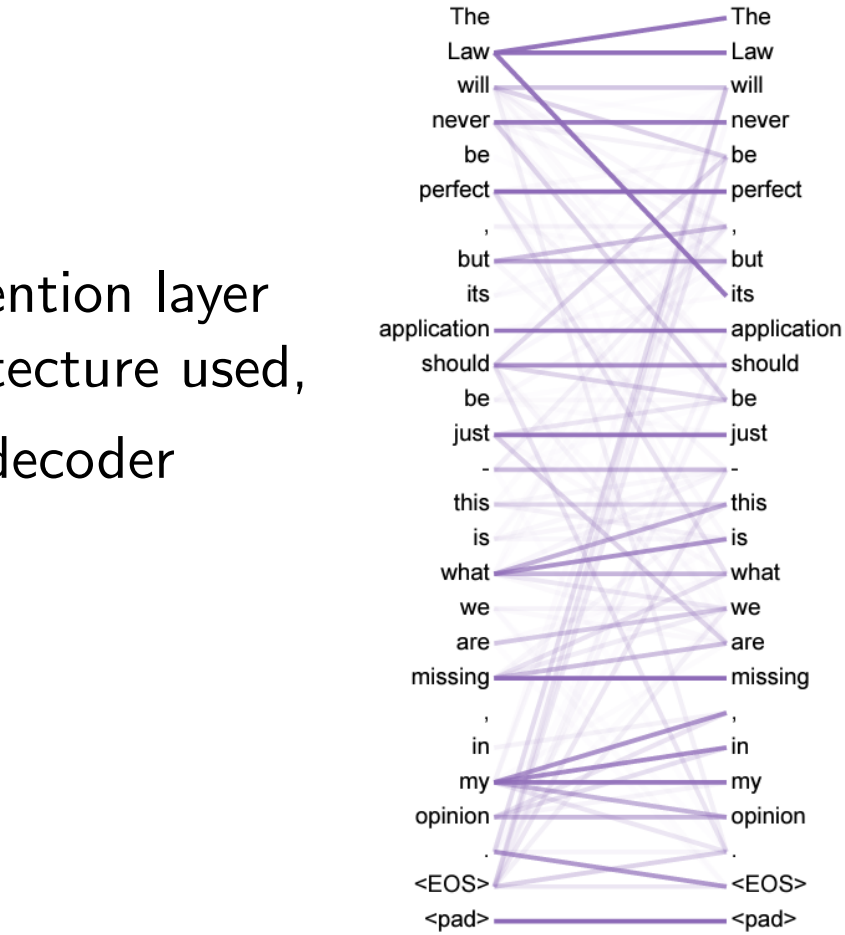
$c_{:i} = \alpha_{1i} v_{:1} + ... + \alpha_{mi} v_{:m}$



Self-Attention

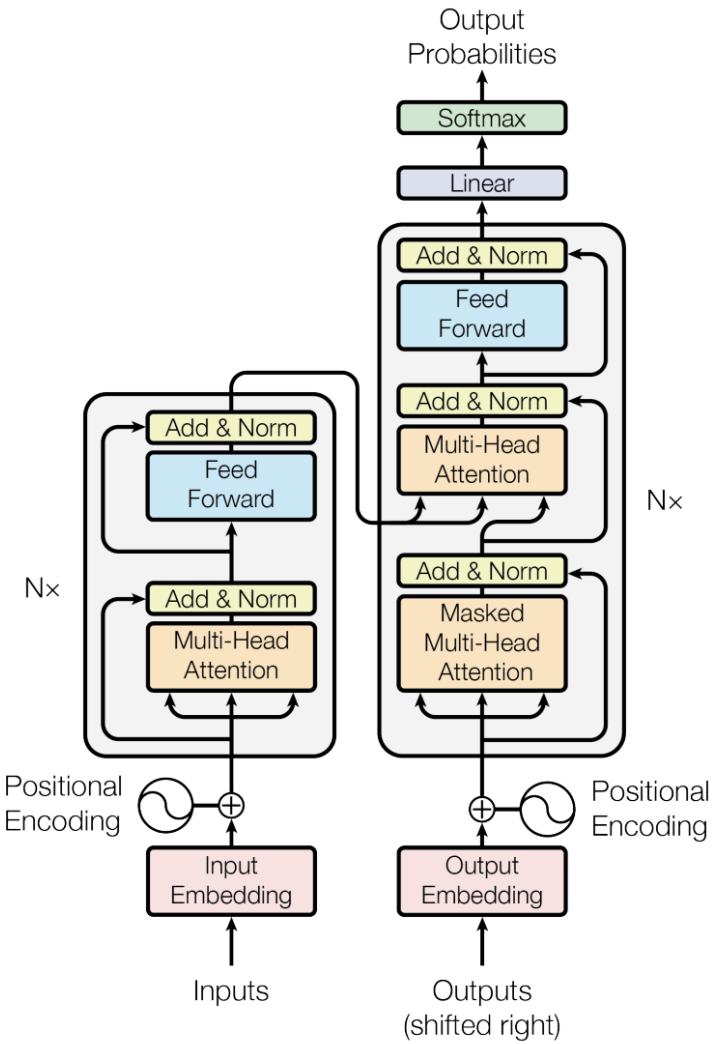
What have changed the introduction of Self-Attention compared to Attention?:

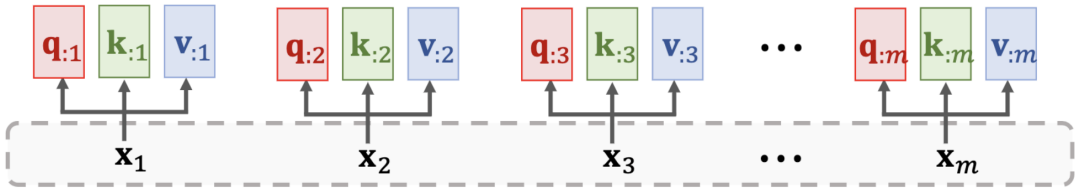
- Ability to use the Self-Attention layer independently of the architecture used,
- No restriction to encoder-decoder (Seq2Seq) networks,
- Easier to combine multiple Self-Attention layers,



A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

Positional Encoding

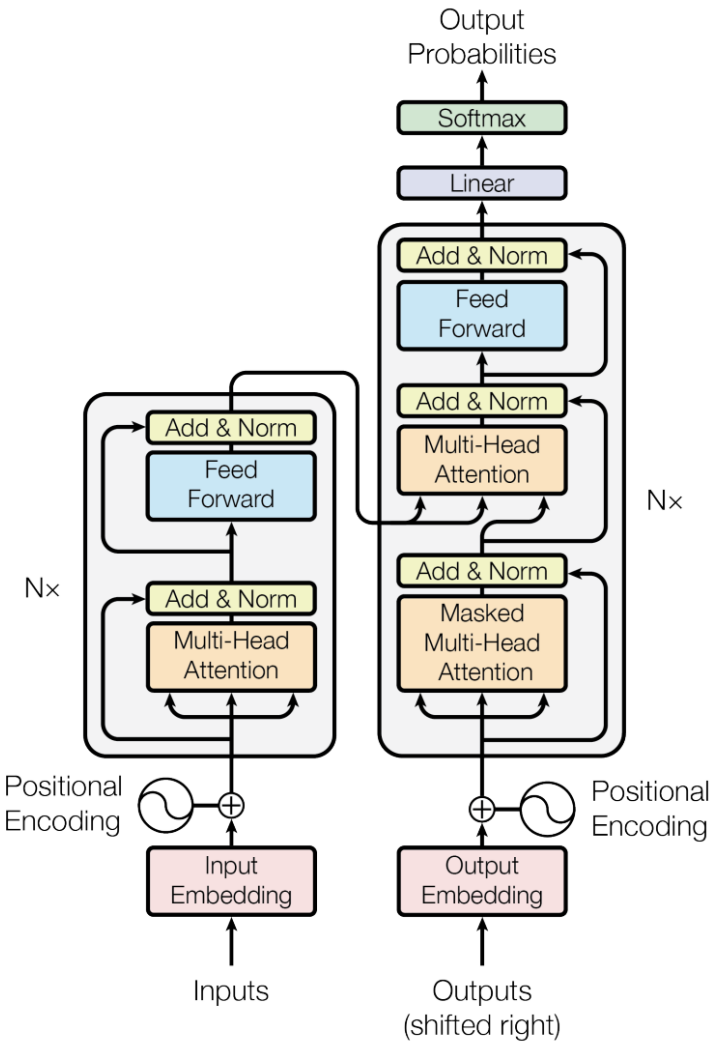


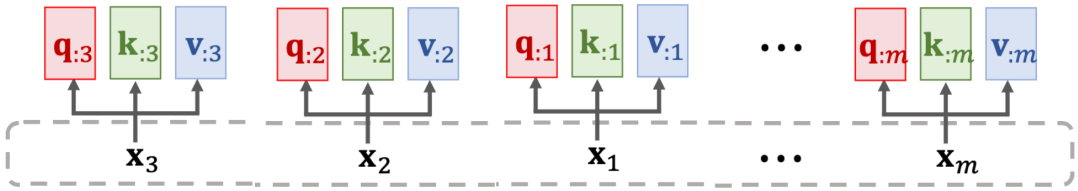


- The order of vector elements doesn't matter - the output c will match the input,
- Hence, positional information is introduced, called **positional encoding**, assigning each input vector element a specific position.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

Positional Encoding

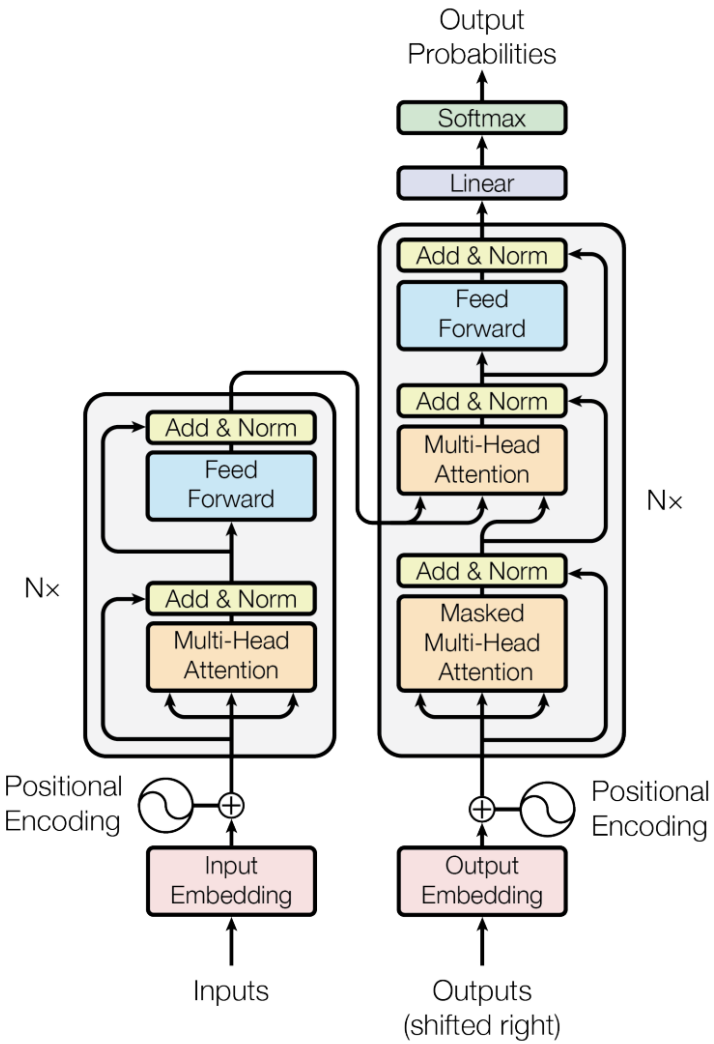


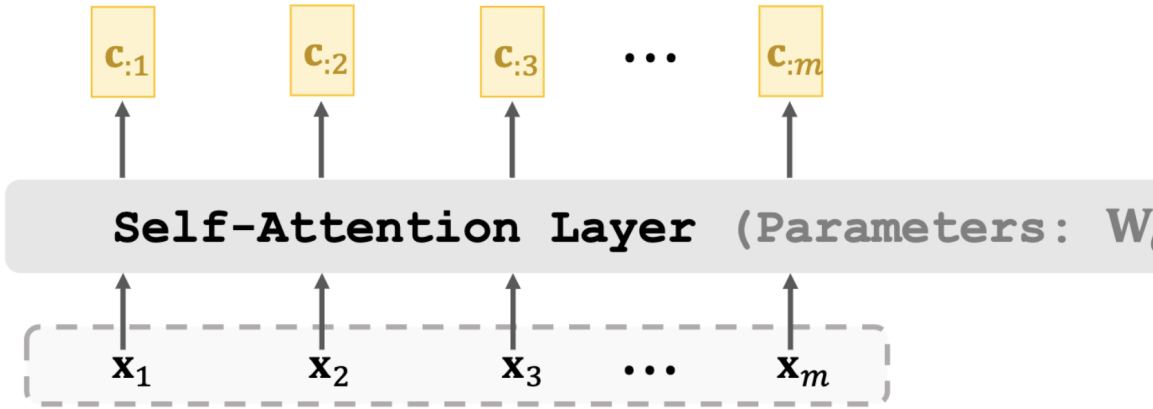


- The order of vector elements doesn't matter - the output c will match the input,
- Hence, positional information is introduced, called **positional encoding**, assigning each input vector element a specific position.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

Positional Encoding

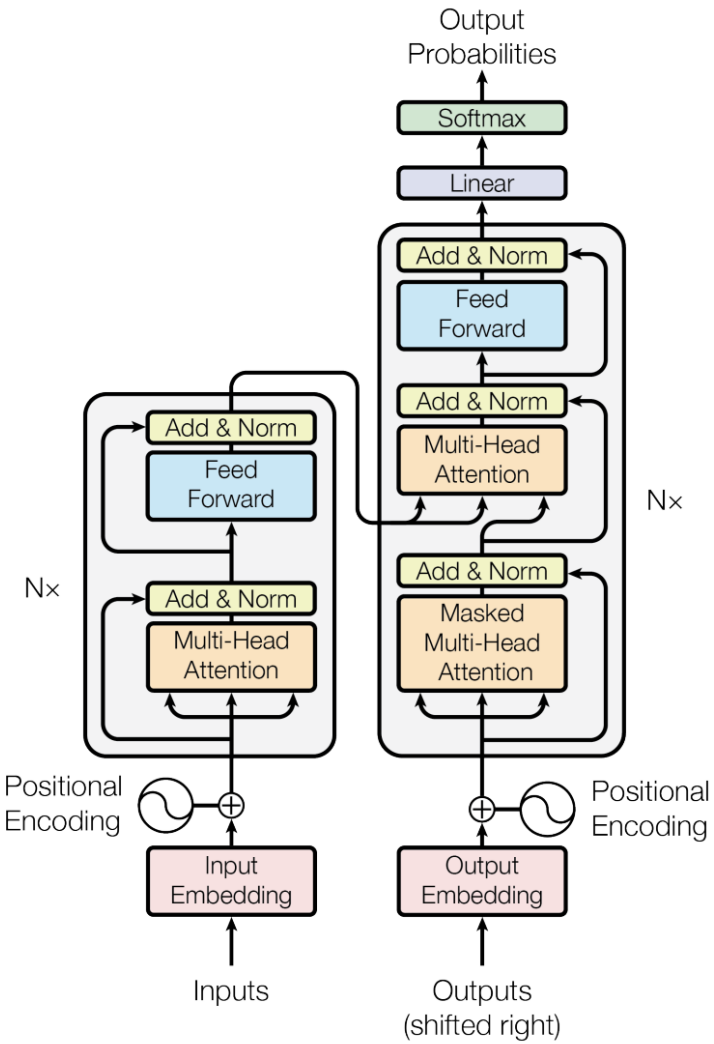


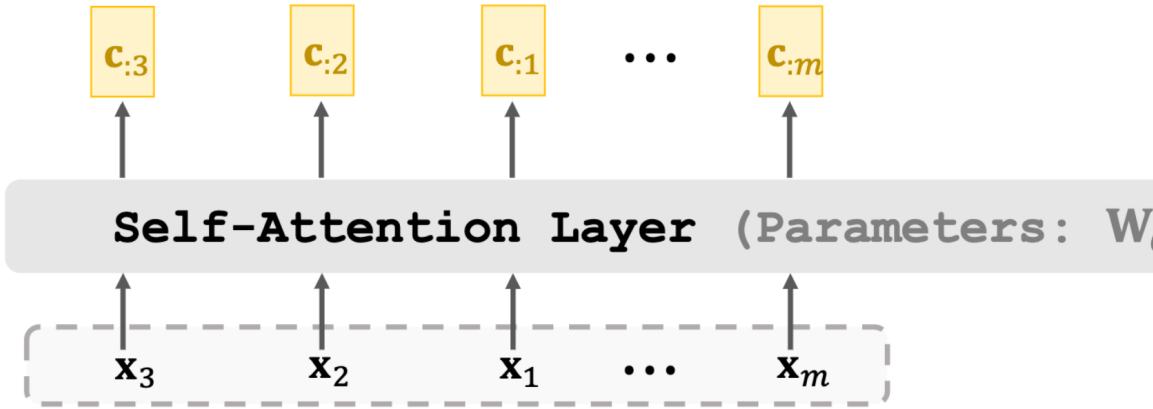


- The order of vector elements doesn't matter - the output c will match the input,
- Hence, positional information is introduced, called **positional encoding**, assigning each input vector element a specific position.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

Positional Encoding





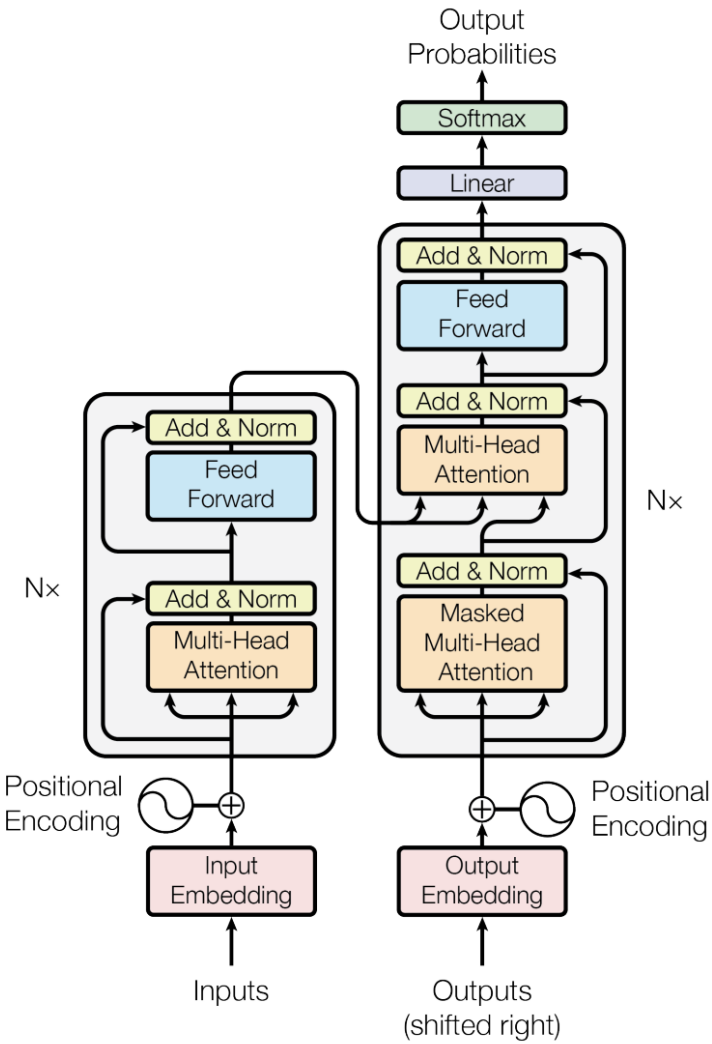
- The order of vector elements doesn't matter - the output c will match the input,
- Hence, positional information is introduced, called **positional encoding**, assigning each input vector element a specific position.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

Demo

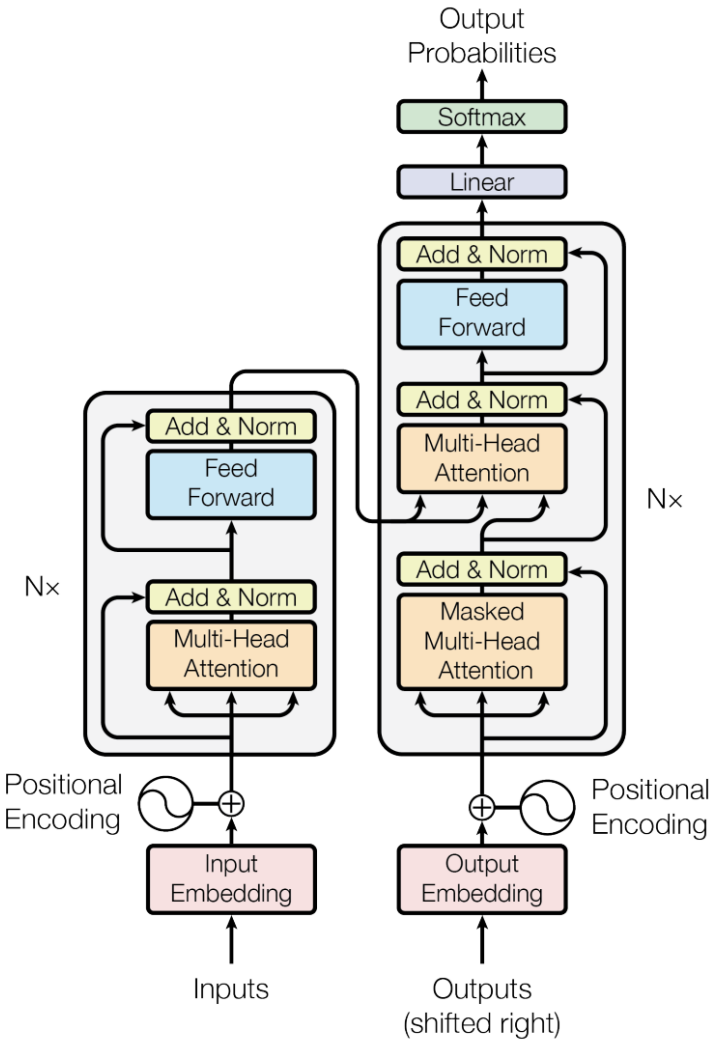
TransfPositional.ipynb

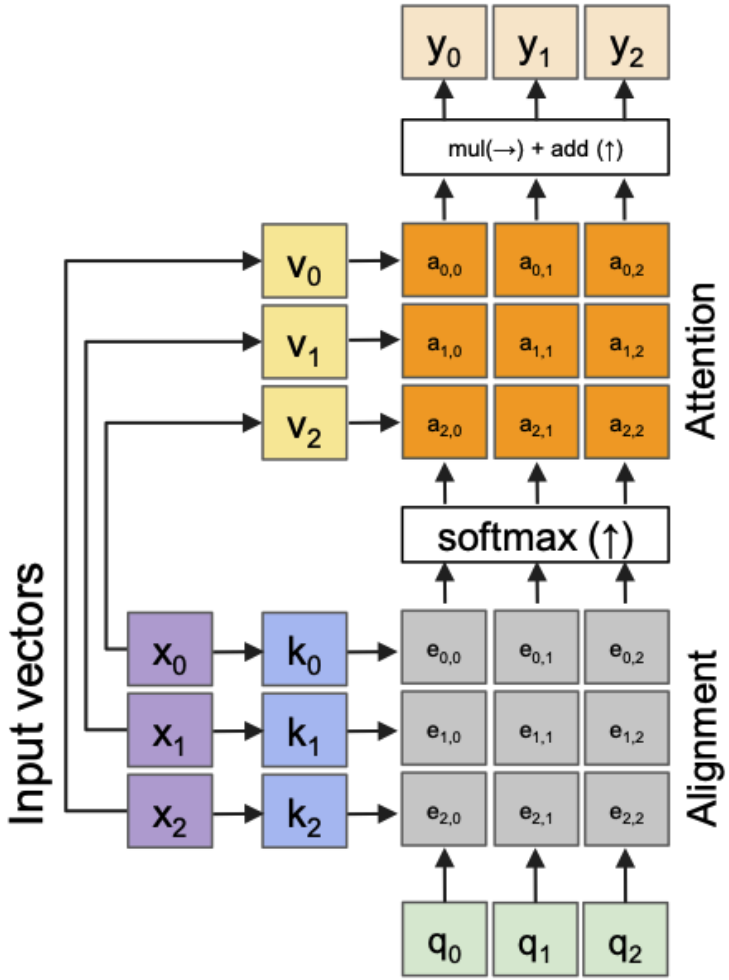
Masked Multi-head Attention



A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

Masked Multi-head Attention

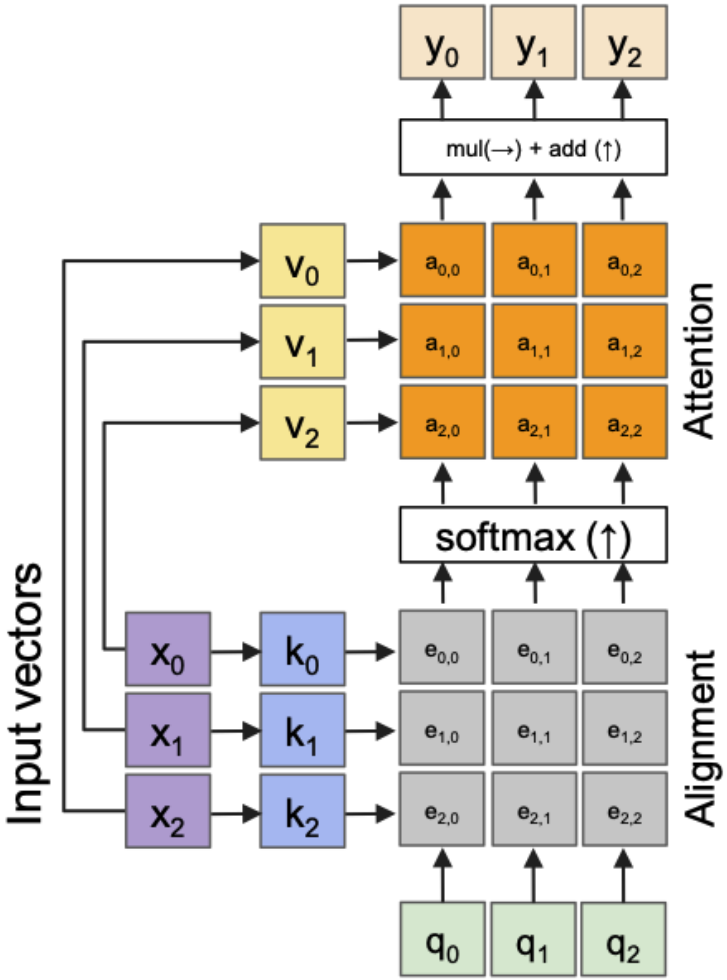




Deep Learning for Computer Vision, Stanford lectures
https://cs231n.stanford.edu

Masked Multi-head Attention

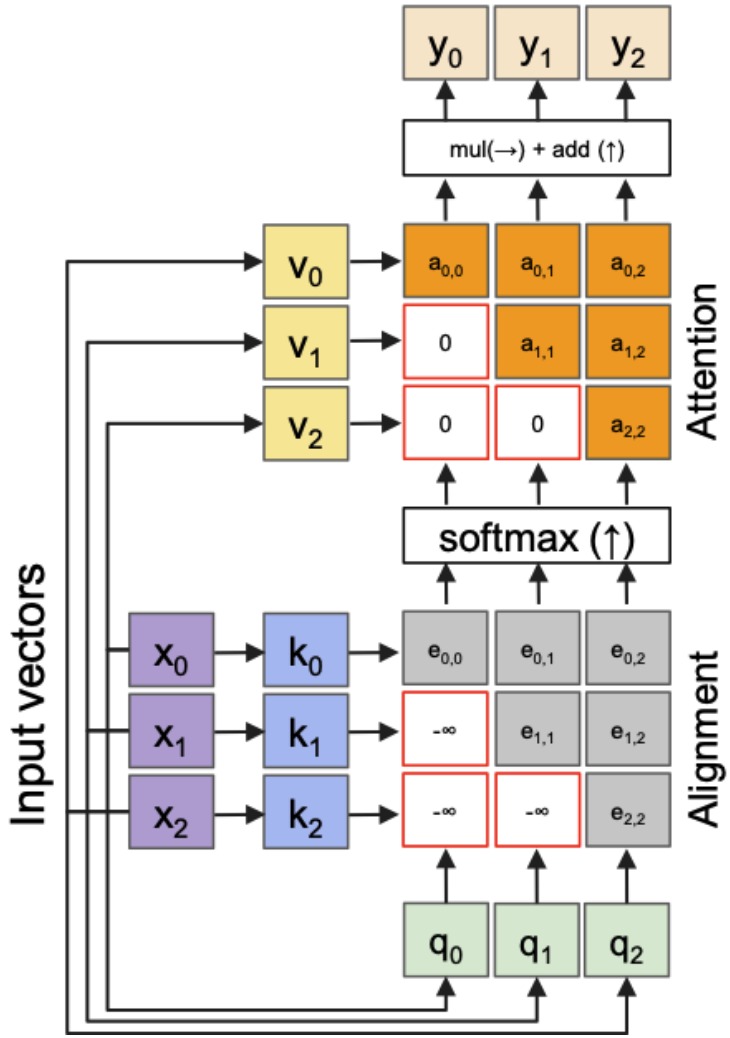
- For parallel computing the future information is available
- But we don't want the transformer to "cheat" by looking ahead at future positions during training
- Future parts of the input sequence are hidden by manually setting the alignment scores to -Inf



Deep Learning for Computer Vision, Stanford lectures https://cs231n.stanford.edu

Masked Multi-head Attention

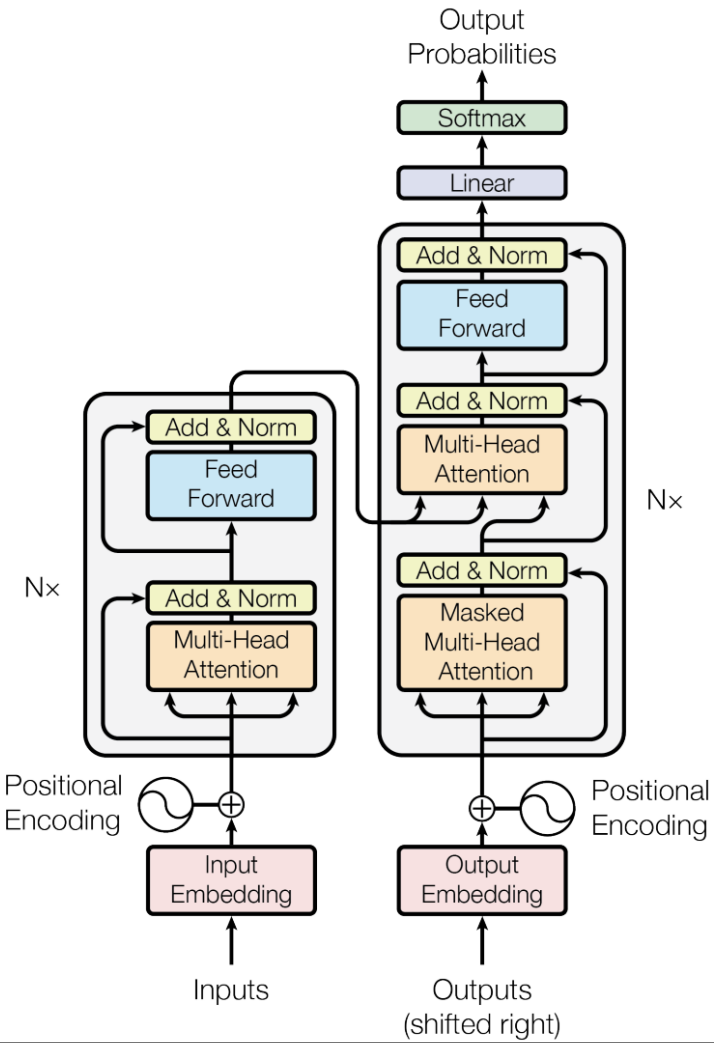
- For parallel computing the future information is available
- But we don't want the transformer to "cheat" by looking ahead at future positions during training
- Future parts of the input sequence are hidden by manually setting the alignment scores to -Inf

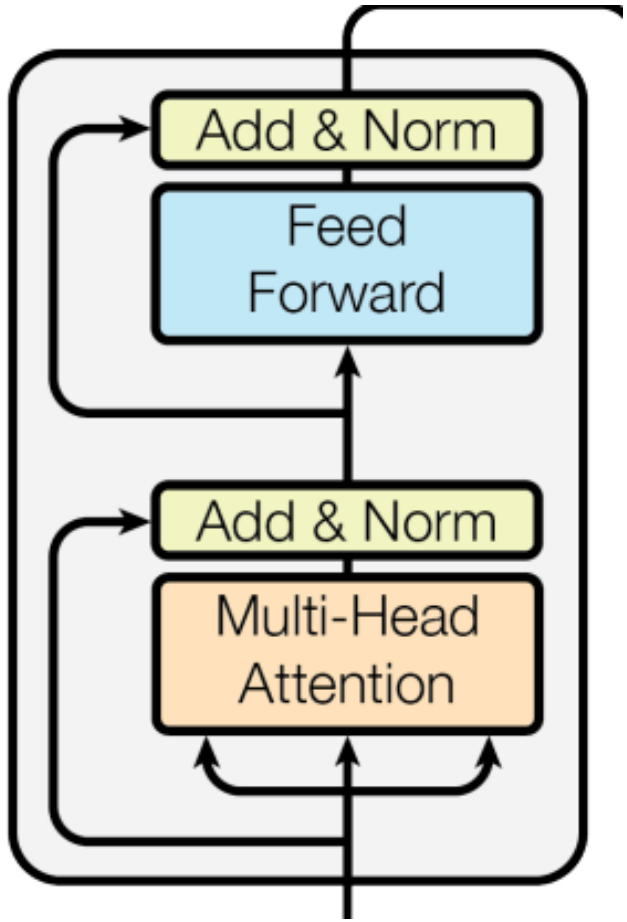


Deep Learning for Computer Vision, Stanford lectures
https://cs231n.stanford.edu

Transformer

Transformer Block:



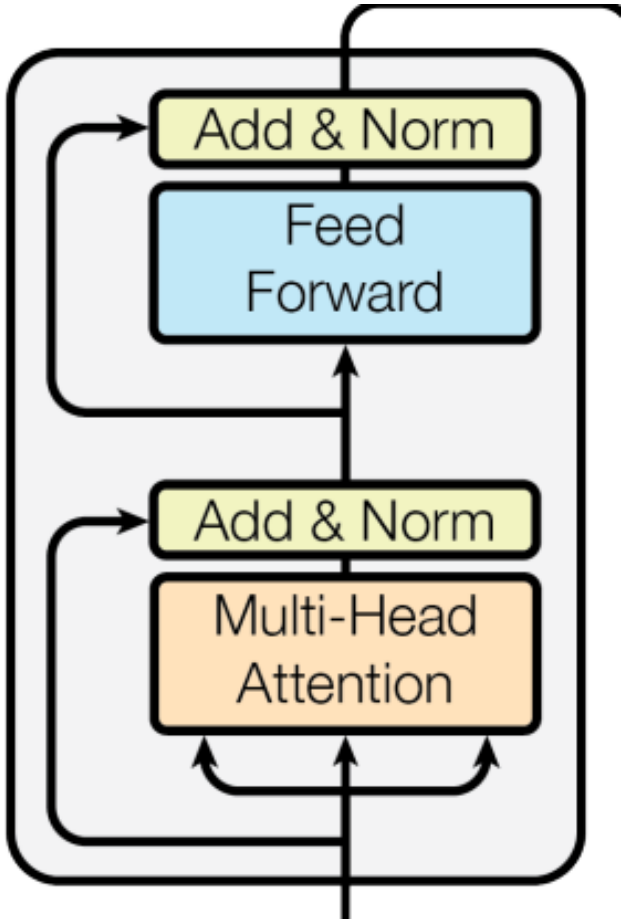


A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

Transformer

Transformer Block:

1. Multi-Head Self Attention Layer (MSP) - the only interaction between input vectors,
2. Multi-Layer Perceptrons Layer (MLP) - operates independently on input vectors,
3. Layer Norm (LN) - operates independently on input vectors,
4. Skip connections - optimization of the training process.



A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31 International Conference on Neural Information Processing Systems (NIPS17).

Demo

Transformer1.ipynb

TO DO:

1. Predict the next token - check whether GPT-2 has learned typical or commonly known phrases
2. Block the prediction of specific words
3. Inspect Attention Head - Check whether sentences with semantically correlated but spatially separated words exhibit appropriate attention patterns.