# Attention & Transformers

Ivo Verhoeven | Advanced Topics in Computational Semantics

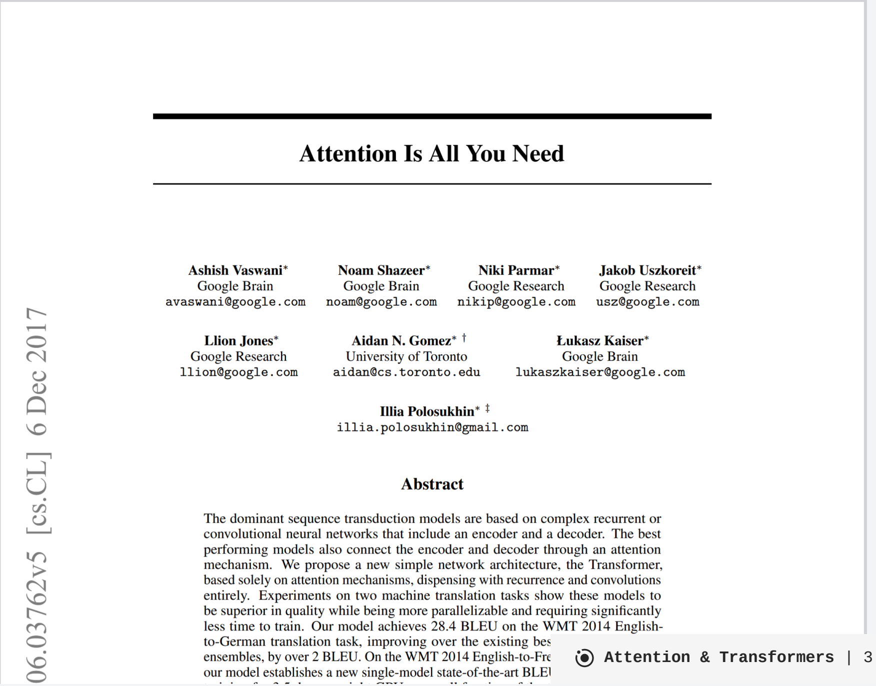
# About Me


- 2017 - 2020: BSc. Liberal Arts & Sciences
- 2020 – 2022: MSc. AI at University of Amsterdam
  - Thesis on meta-learning, morphology and translation
  - Took ATCS in 2021
- 2022 - ???: PhD at ILLC
  - Misinformation detection and generalisation with Katia Shutova

# Vaswani et al.: Attention is All You Need

- Introduces the Transformer architecture in late 2017
  - Google Brain/Google Research collab

Vaswani et al. (2017). Attention is all you need. Advances
neural information processing systems, 30.



# 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



06.03762v5 [cs.CL] 6 Dec 2017


## Abstract

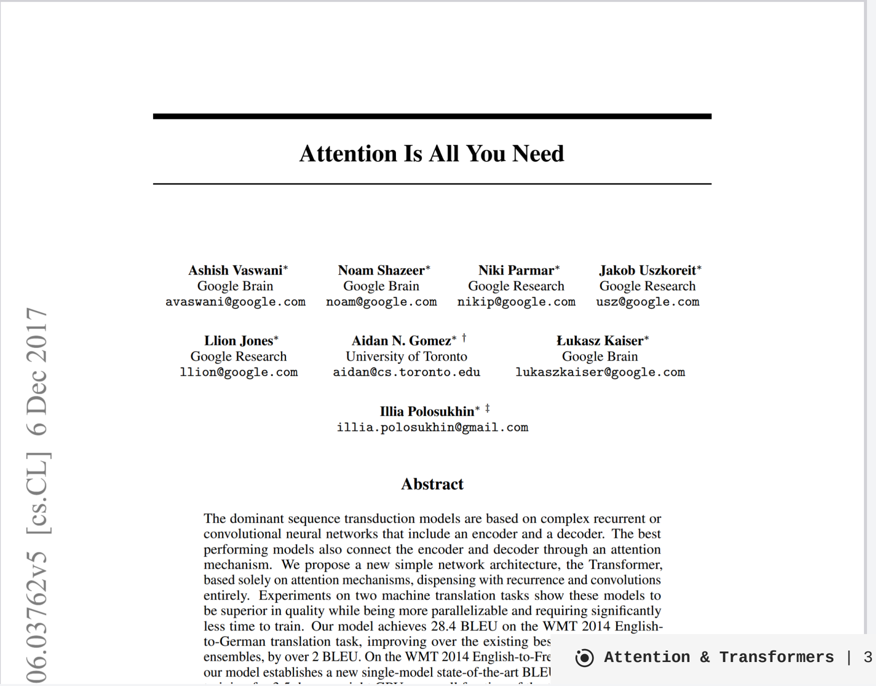

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing bes
ensembles, by over 2 BLEU. On the WMT 2014 English-to-Fre
our model establishes a new single-model state-of-the-art BLE

# Vaswani et al.: Attention is All You Need

- Introduces the Transformer architecture in late 2017
  - Google Brain/Google Research collab

- Paper currently has **169 248** citations
  - Or **~64 citations a day**

`Vaswani et al. (2017). Attention is all you need. Advances in neural information processing systems, 30.`



arXiv:1706.03762v5 [cs.CL] 6 Dec 2017

## 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


### Abstract

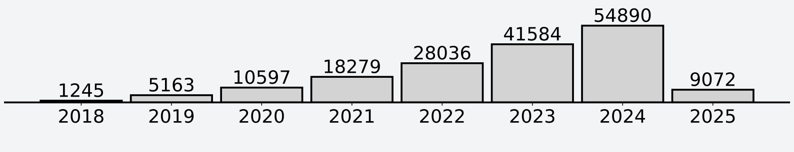

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing bes ensembles, by over 2 BLEU. On the WMT 2014 English-to-Fre our model establishes a new single-model state-of-the-art BLE

# Vaswani et al.: Attention is All You Need

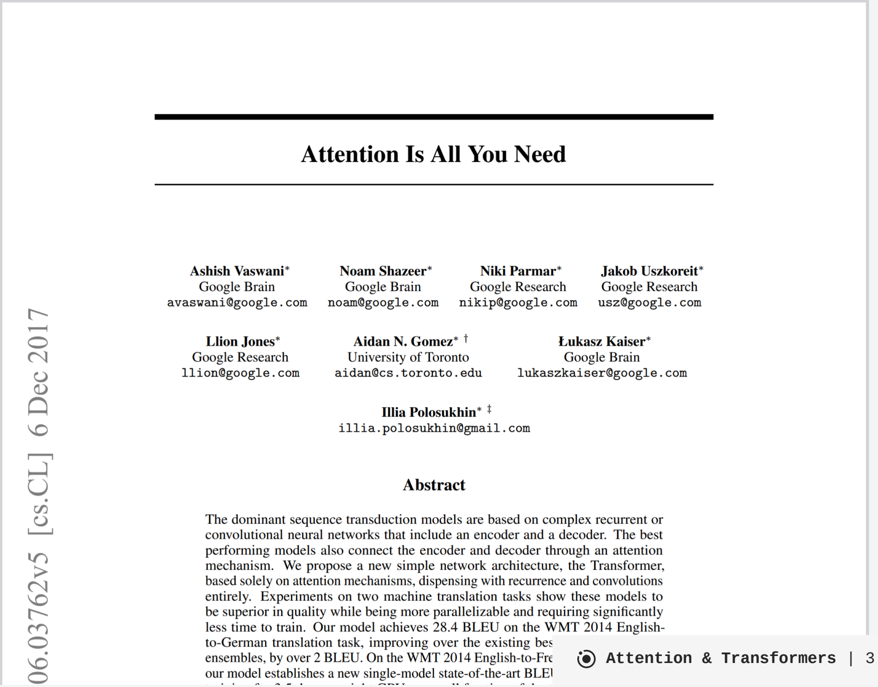
- Introduces the Transformer architecture in late 2017
    - Google Brain/Google Research collab

- Paper currently has **169 248** citations
    - Or **~64 citations a day**

- Number of citations is only accelerating




Vaswani et al. (2017). Attention is all you need. Advances
neural information processing systems, 30.



06.03762v5 [cs.CL] 6 Dec 2017

## 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

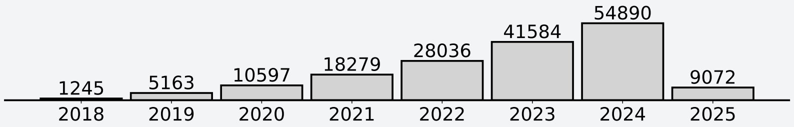


**Abstract**

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing bes
ensembles, by over 2 BLEU. On the WMT 2014 English-to-Fre
our model establishes a new single-model state-of-the-art BLE

# Vaswani et al.: Attention is All You Need

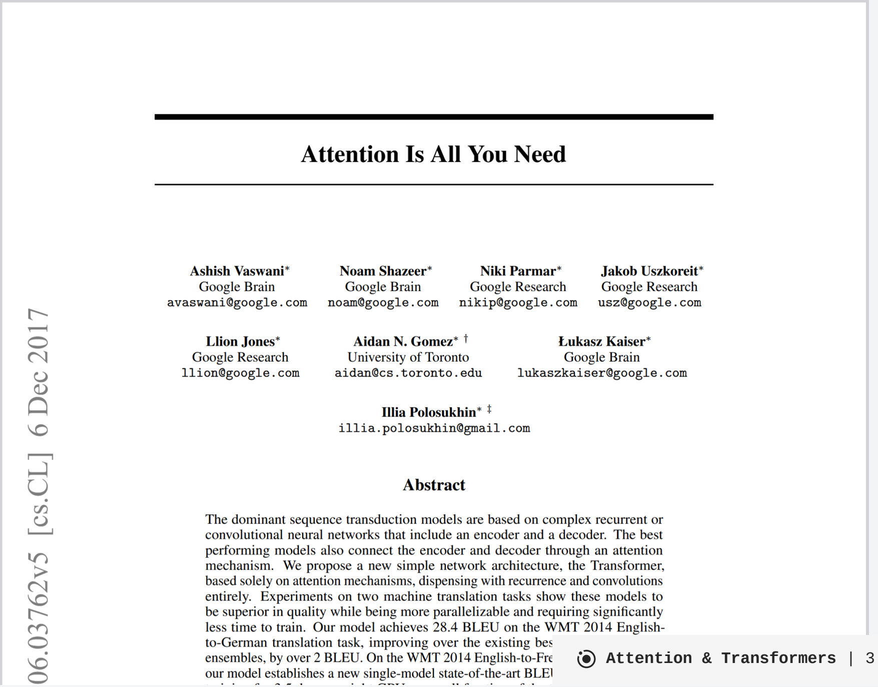
- Introduces the Transformer architecture in late 2017
  - Google Brain/Google Research collab
- Paper currently has **169 248** citations
  - Or **~64 citations a day**
- Number of citations is only accelerating




- Most cited paper ever has **233 829** citations

`Lowry et al. (1951) Protein measurement with the folin phenol reagent.`

`Vaswani et al. (2017). Attention is all you need. Advances neural information processing systems, 30.`



06.03762v5 [cs.CL] 6 Dec 2017

## 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

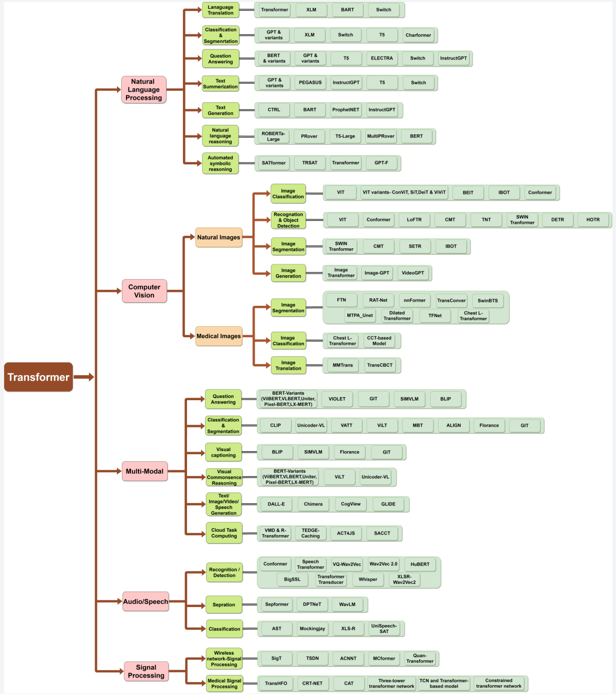


**Abstract**

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing bes ensembles, by over 2 BLEU. On the WMT 2014 English-to-Fre our model establishes a new single-model state-of-the-art BLE

# Vaswani et al.: Attention is All You Need

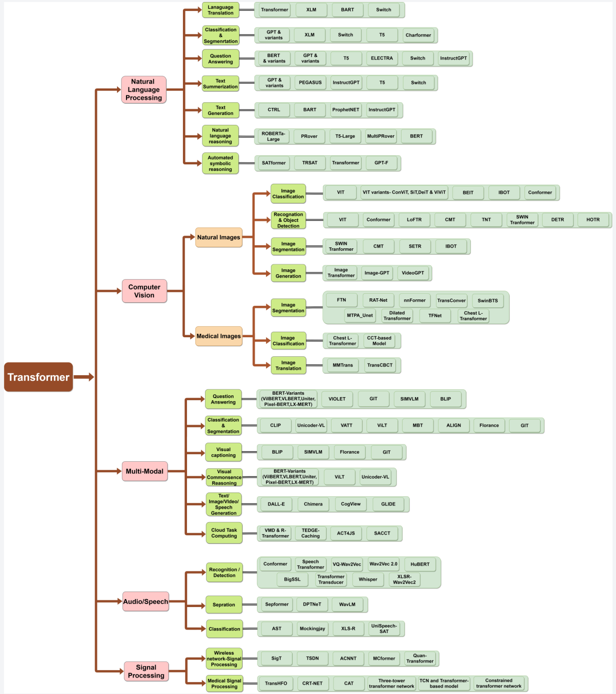
- It’s hard to think of an AI area that hasn’t been affected by the Transformer




Islam, et al. (2023). A Comprehensive Survey on Application Transformers for Deep Learning Tasks. arXiv:2306.07303.

# Vaswani et al.: Attention is All You Need

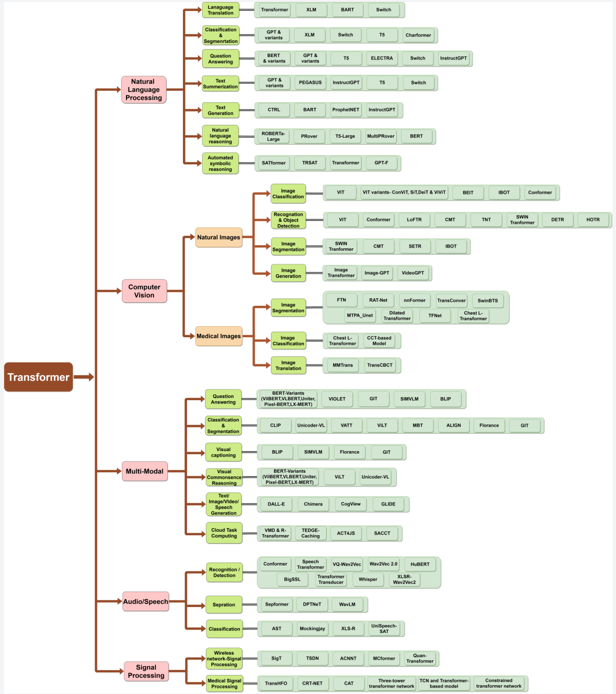
- It’s hard to think of an AI area that hasn’t been affected by the Transformer
  - **NLP:** Transformer > RNN
    - Seq-to-seq: what it was designed for
    - Classification: encoder-only transformers
    - Generation: decoder-only transformers




Islam, et al. (2023). A Comprehensive Survey on Application
Transformers for Deep Learning Tasks. arXiv:2306.07303.

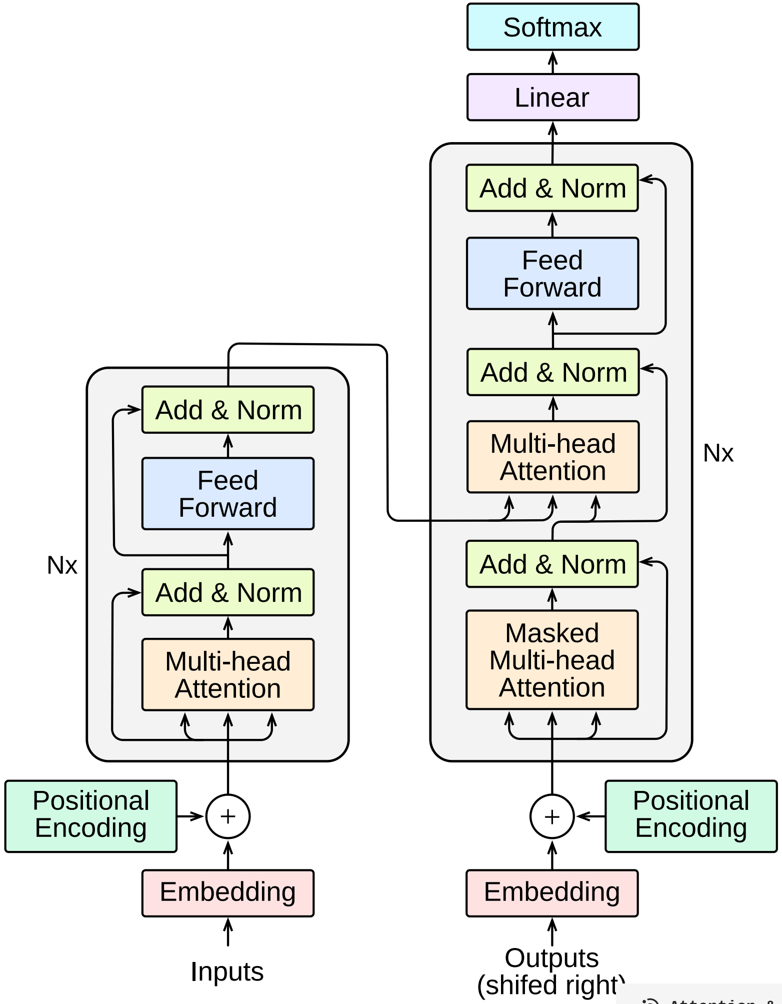
# Vaswani et al.: Attention is All You Need

- It’s hard to think of an AI area that hasn’t been affected by the Transformer
  - **NLP:** Transformer > RNN
    - Seq-to-seq: what it was designed for
    - Classification: encoder-only transformers
    - Generation: decoder-only transformers
  - **CV:** ViT > CNN
  - **Multi-modal:** Transformer > different architectures
  - **Speech:** Transformer > CNN
  - **Graphs:** Transformer/Attention > GCN

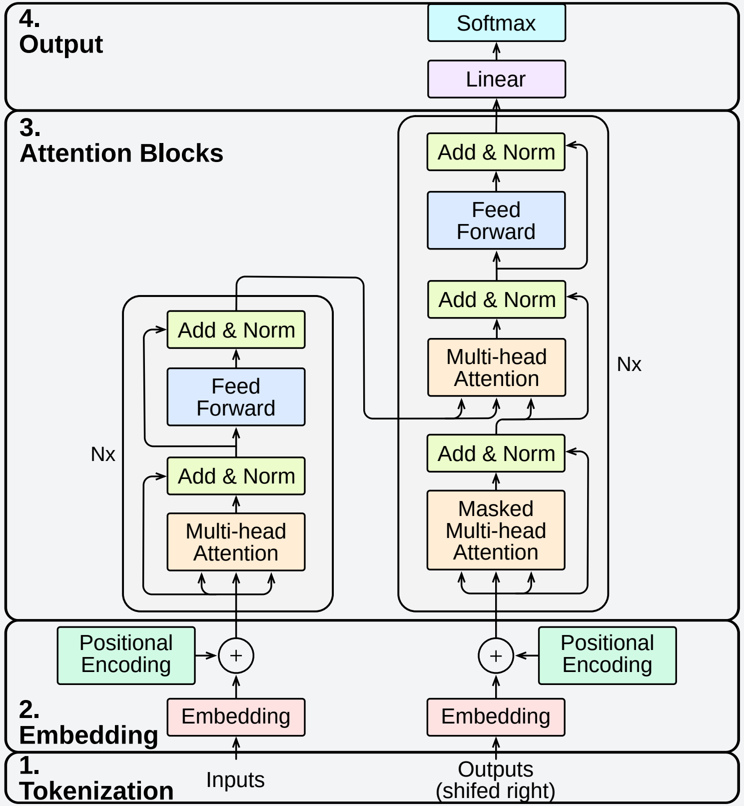



Islam, et al. (2023). A Comprehensive Survey on Application
Transformers for Deep Learning Tasks. arXiv:2306.07303.

# The Transformer

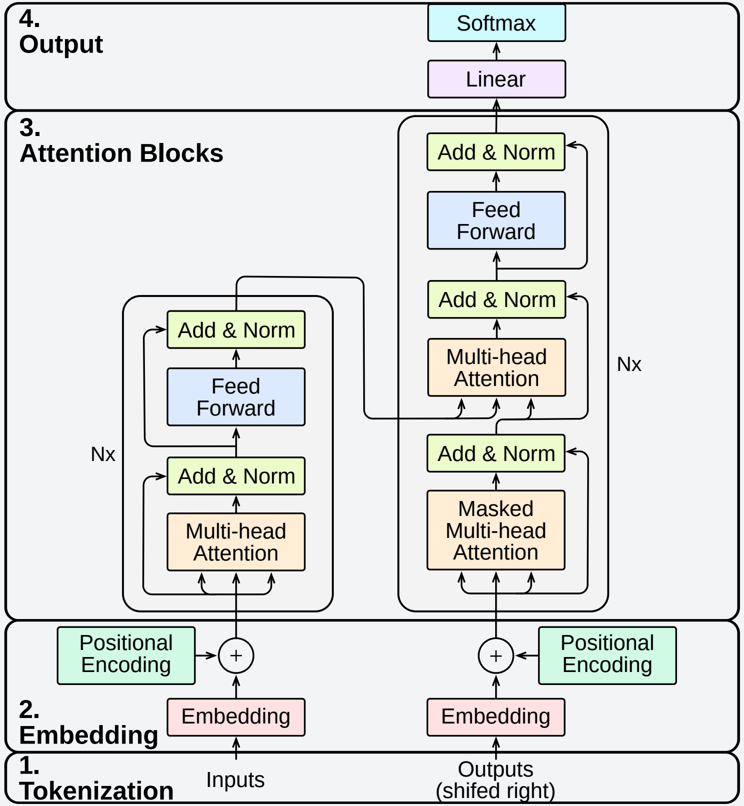


# Breaking the Transformer into modules



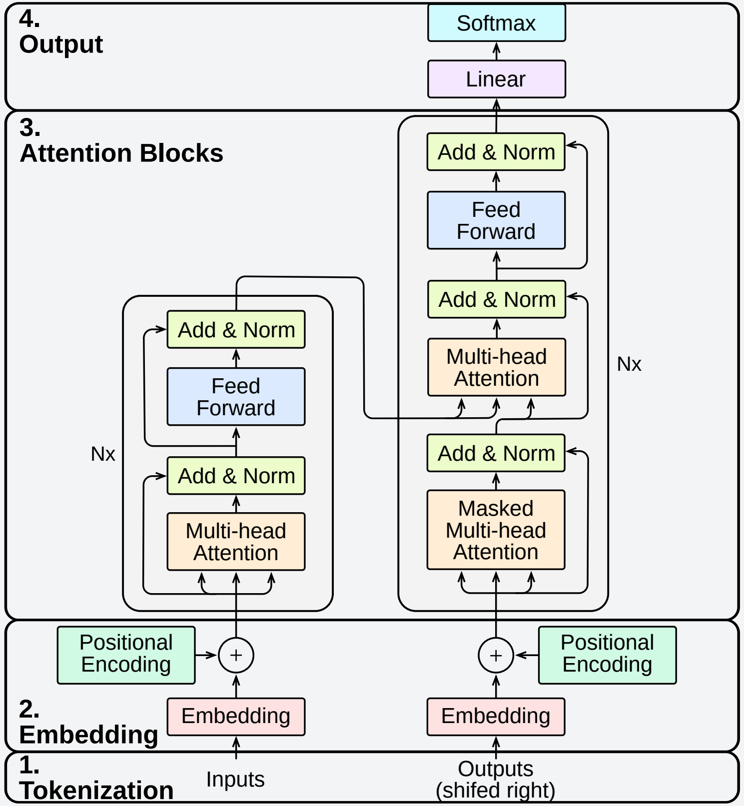
# Breaking the Transformer into modules

4. Output
   - Softmax
   - Linear



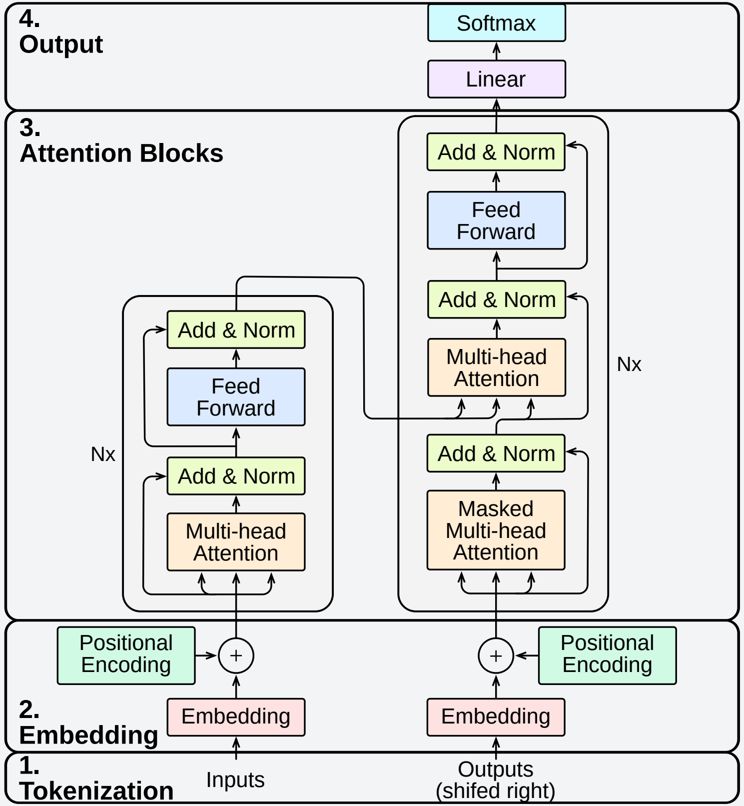
# Breaking the Transformer into modules

4. Output
    - Softmax
    - Linear
3. Attention Blocks
    - Multi-head Attention
    - Add & Norm
    - Feed Forward



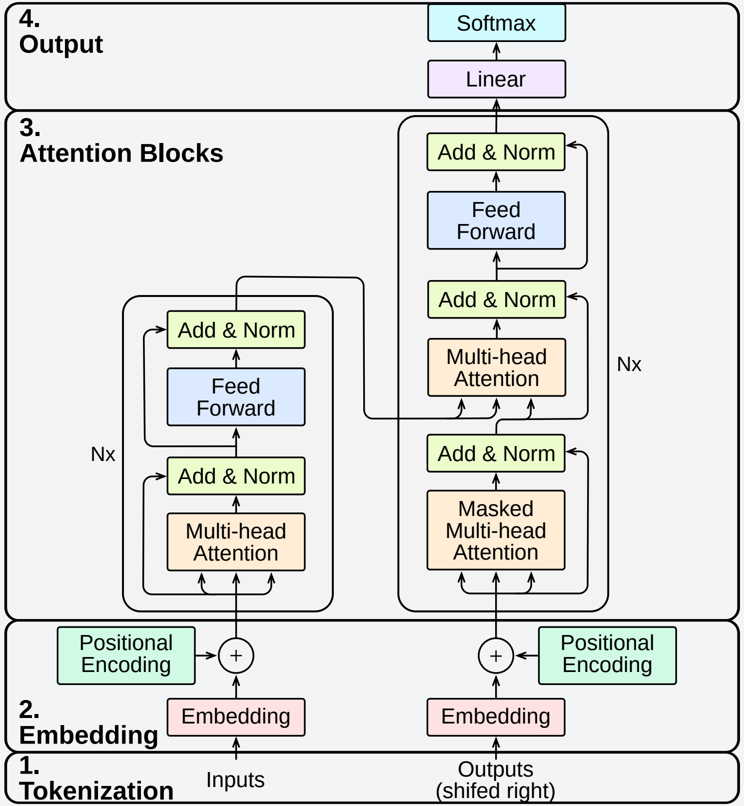
# Breaking the Transformer into modules

4. Output
   - Softmax
   - Linear
3. Attention Blocks
   - Multi-head Attention
   - Add & Norm
   - Feed Forward
2. Embedding
   - Token Embedding
   - Positional Encoding



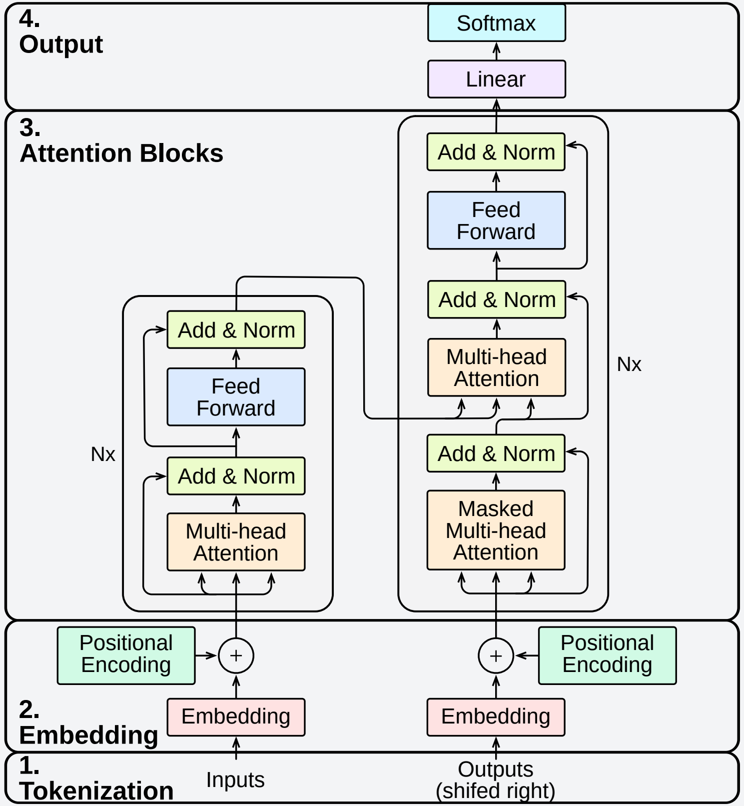
# Breaking the Transformer into modules

4. Output
   - Softmax
   - Linear
3. Attention Blocks
   - Multi-head Attention
   - Add & Norm
   - Feed Forward
2. Embedding
   - Token Embedding
   - Positional Encoding
1. Tokenization
   - (Not pictured)



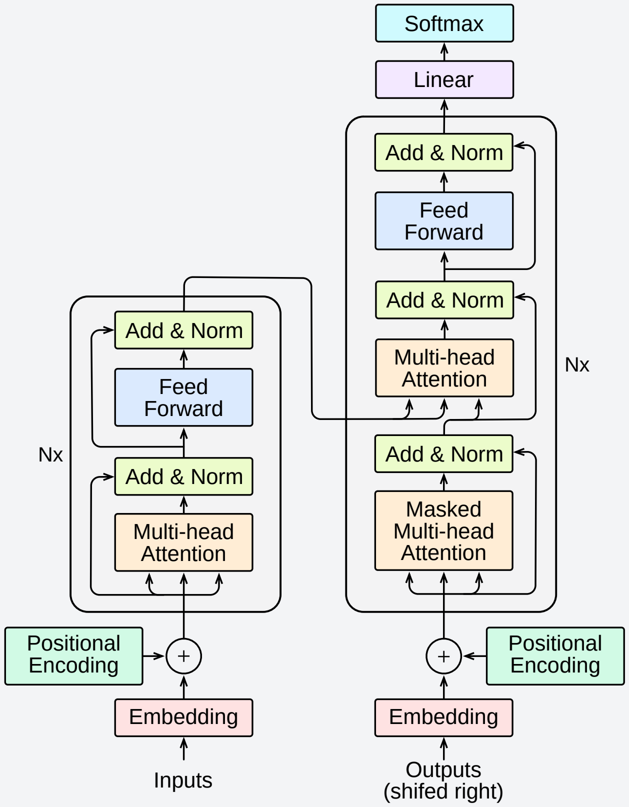
# Breaking the Transformer into modules

4. ~~Output~~
   - ~~Softmax~~
   - ~~Linear~~
3. Attention Blocks
   - Multi-head Attention
   - Add & Norm
   - Feed Forward
2. Embedding
   - ~~Token Embedding~~
   - Positional Encoding
1. Tokenization
   - (Not pictured)



# Table of Contents





# Encoders & Decoders

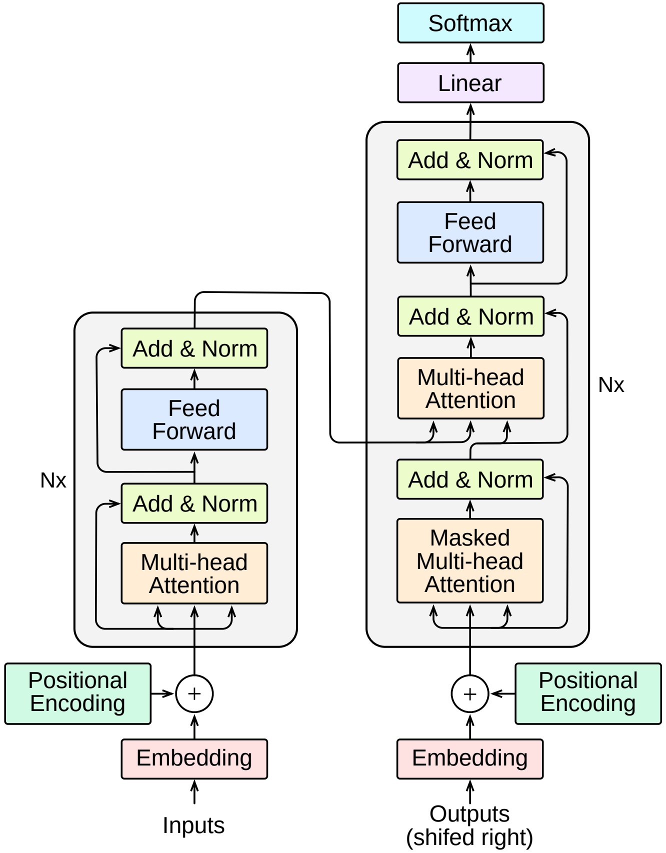
Text comes in, text goes out

Jakob Uszkoreit (August 31, 2017). Transformer: A Novel Neural Network Architecture for Language Understanding.
https://research.google/blog/transformer-a-novel-neural-network-architecture-for-language-understanding/

# Attention Blocks

What makes the Transformer what it is — and where it came from

# Multi-head Attention



# Definition & Properties

Multi-head Attention

- Let $\mathbf{V}$ be a matrix of (word) vectors
  - It has a sequence length of $T_V$
  - It has a dimensionality of $D$

$$\texttt{Attention}(?, ?, \mathbf{V}) = \mathbf{A}\mathbf{V}$$

$$\mathbf{A} \in (0, 1)^{[T_V \times T_V]}$$

$$\mathbf{V} \in \mathbb{R}^{[T_V \times D]}$$

# Definition & Properties

Multi-head Attention

- Let $\mathbf{V}$ be a matrix of (word) vectors
  - It has a sequence length of $T_V$
  - It has a dimensionality of $D$
- $\texttt{Attention}$ is just a matrix product of $\mathbf{V}$ with an attention matrix $\mathbf{A}$
  - $\mathbf{A}$ is a square matrix of size $T_V \times T_V$
  - It's elements are all between $(0, 1)$
  - It's rows sum to $1$

$$\texttt{Attention}(?, ?, \mathbf{V}) = \mathbf{A}\mathbf{V}$$

$$\mathbf{A} \in (0, 1)^{[T_V \times T_V]}$$

$$\mathbf{V} \in \mathbb{R}^{[T_V \times D]}$$

# Definition & Properties

## Multi-head Attention

- Let $\mathbf{V}$ be a matrix of (word) vectors
  - It has a sequence length of $T_V$
  - It has a dimensionality of $D$
- $\mathtt{Attention}$ is just a matrix product of $\mathbf{V}$ with an attention matrix $\mathbf{A}$
  - $\mathbf{A}$ is a square matrix of size $T_V \times T_V$
  - It's elements are all between $(0, 1)$
  - It's rows sum to $1$

$$\mathtt{Attention}(?, ?, \mathbf{V}) = \mathbf{A}\mathbf{V}$$

$$\mathbf{A} \in (0, 1)^{[T_V \times T_V]}$$

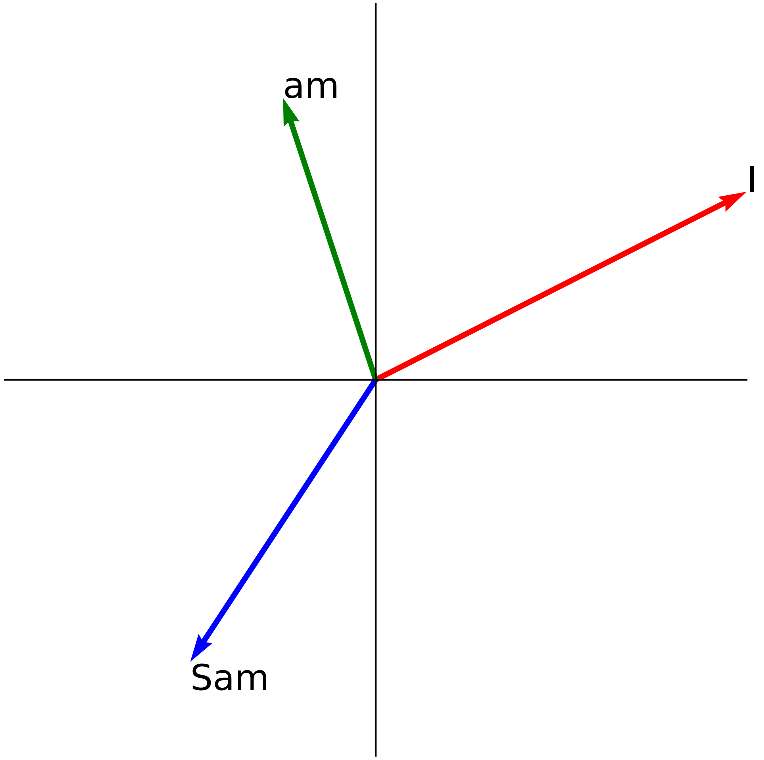
$$\mathbf{V} \in \mathbb{R}^{[T_V \times D]}$$

# Definition & Properties

## Multi-head Attention

- The result of **`Attention`** is just a convex combination of $\mathbf{V}$

$$\overset{\mathbf{A}}{\begin{bmatrix} 0.6 & 0.1 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.1 & 0.7 \end{bmatrix}} \overset{\mathbf{V}}{\begin{bmatrix} 2.0 & 1.0 \\ -0.5 & 2.0 \\ -1.0 & -0.5 \end{bmatrix}} \begin{array}{c} \color{red}{\mathrm{I}} \\ \color{green}{\mathrm{am}} \\ \color{blue}{\mathrm{Sam}} \end{array}$$



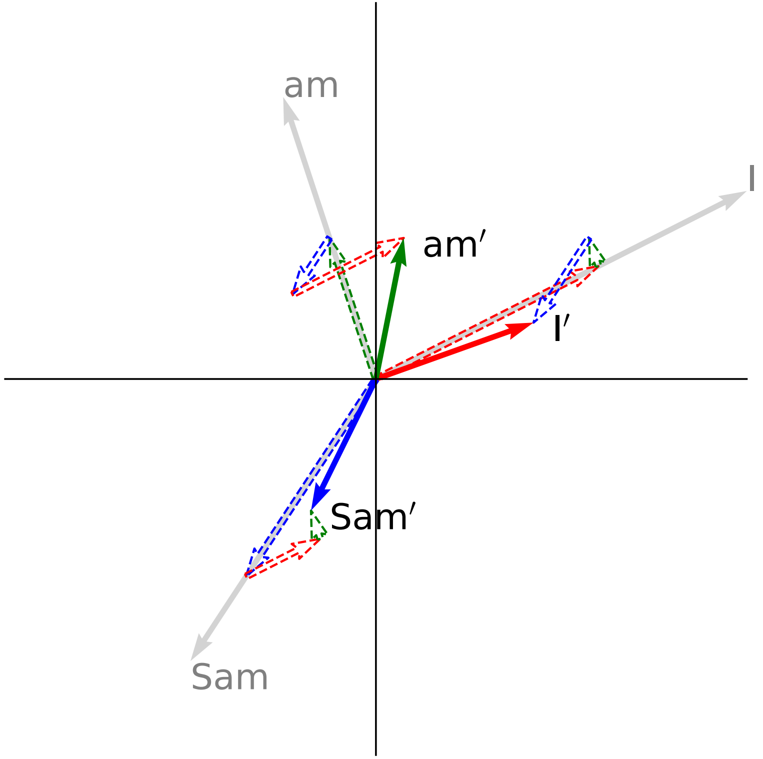
# Definition & Properties

Multi-head Attention

- The result of **`Attention`** is just a convex combination of $\mathbf{V}$

$$\begin{array}{cc} \mathbf{A} & \mathbf{V} & \\ \begin{bmatrix} 0.6 & 0.1 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.1 & 0.7 \end{bmatrix} & \begin{bmatrix} 2.0 & 1.0 \\ -0.5 & 2.0 \\ -1.0 & -0.5 \end{bmatrix} & \begin{array}{c} \color{red}{\mathrm{I}} \\ \color{green}{\mathrm{am}} \\ \color{blue}{\mathrm{Sam}} \end{array} \end{array}$$

$$= \begin{bmatrix} 0.6 * \color{red}{\mathrm{I}} + 0.1 * \color{green}{\mathrm{am}} + 0.3 * \color{blue}{\mathrm{Sam}} \\ 0.3 * \color{red}{\mathrm{I}} + 0.5 * \color{green}{\mathrm{am}} + 0.2 * \color{blue}{\mathrm{Sam}} \\ 0.2 * \color{red}{\mathrm{I}} + 0.1 * \color{green}{\mathrm{am}} + 0.7 * \color{blue}{\mathrm{Sam}} \end{bmatrix}$$

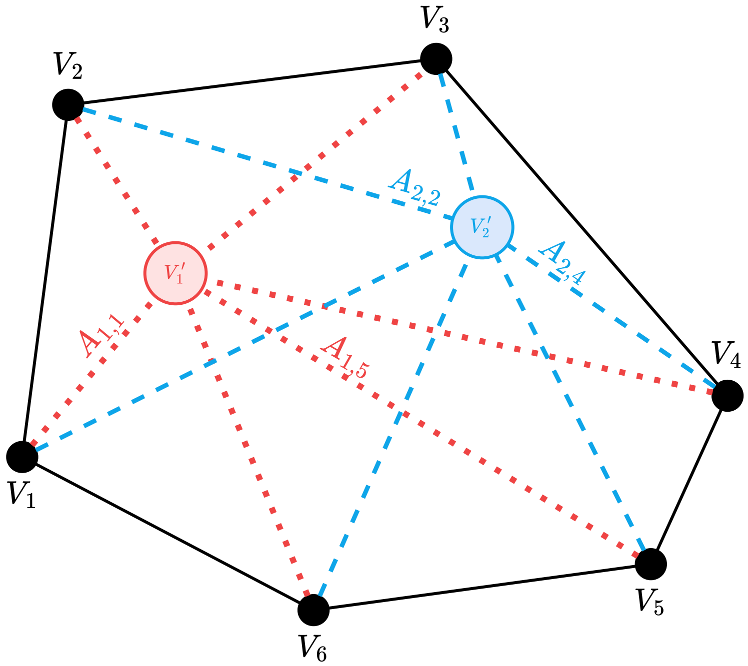


# Definition & Properties

Multi-head Attention

**Convex Combination**

The elements of $V'$ will lie inside the convex hull of all of the elements in $V$



# Definition & Properties

## Multi-head Attention

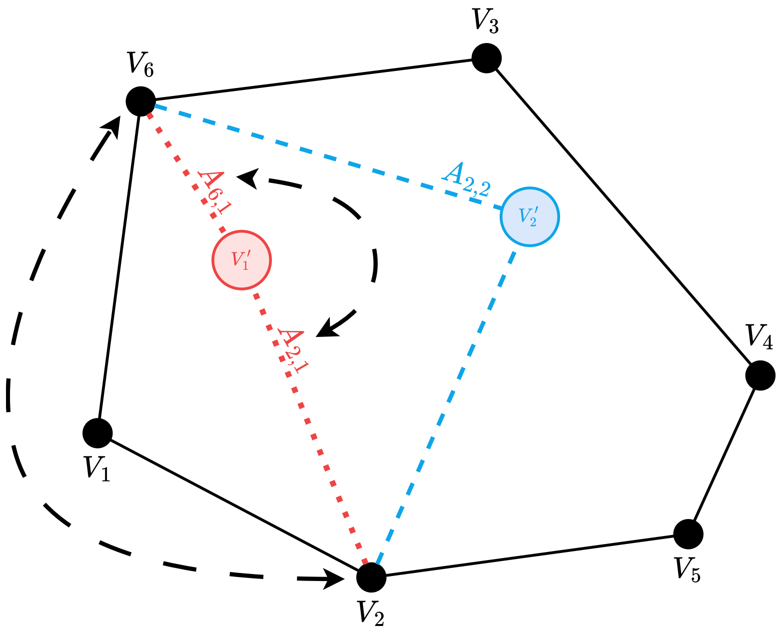
### Convex Combination

The elements of $V'$ will lie inside the convex hull of all of the elements in $V$

### Permutation Equivariance

The elements of $V'$ are *equivariant* to a change in the order of the rows of $\mathbf{A}$

- Attention does not care about word order



# Definition & Properties

Multi-head Attention

So is `Attention` just a linear map?

- Not quite

Linear maps are:

# Definition & Properties

Multi-head Attention

So is `Attention` just a linear map?

- Not quite

Linear maps are:

- Inflexible in terms of sequence length

# Definition & Properties

Multi-head Attention

So is `Attention` just a linear map?

- Not quite

Linear maps are:

- Inflexible in terms of sequence length
- Parameter inefficient

# Definition & Properties

## Multi-head Attention

So is `Attention` just a linear map?

- Not quite

Linear maps are:

- Inflexible in terms of sequence length
- Parameter inefficient
- Invariant to the input content

# Definition & Properties

## Multi-head Attention

- Let $\mathbf{V}$ be a matrix of **value** vectors
  - It has a sequence length of $T_V$
  - It has a dimensionality of $D_V$
- Let $\mathbf{K}$ be a matrix of **key** vectors
  - It has a sequence length of $T_V$
  - It has a dimensionality of $D_Q$
- Let $\mathbf{Q}$ be a matrix of **query** vectors
  - It has a sequence length of $T_Q$
  - It has a dimensionality of $D_Q$

$$\mathtt{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \underbrace{\mathtt{softmax}\left(f\left(\mathbf{Q}, \mathbf{K}\right)\right)}_{\mathbf{A}}\mathbf{V}$$

$$\mathbf{A} \in (0,1)^{[T_Q \times T_V]}$$

$$\mathbf{V} \in \mathbb{R}^{[T_V \times D_V]}$$

$$\mathbf{K} \in \mathbb{R}^{[T_V \times D_Q]}$$

$$\mathbf{Q} \in \mathbb{R}^{[T_Q \times D_Q]}$$

# Definition & Properties

Multi-head Attention

- Let $\mathbf{V}$ be a matrix of **value** vectors
  - It has a sequence length of $T_V$
  - It has a dimensionality of $D_V$
- Let $\mathbf{K}$ be a matrix of **key** vectors
  - It has a sequence length of $T_V$
  - It has a dimensionality of $D_Q$
- Let $\mathbf{Q}$ be a matrix of **query** vectors
  - It has a sequence length of $T_Q$
  - It has a dimensionality of $D_Q$
- Let $f(\mathbf{Q}, \mathbf{K})$ be some kernel function
  - Read: similarity function

$$\mathtt{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \underbrace{\mathtt{softmax}\left(f\left(\mathbf{Q}, \mathbf{K}\right)\right)}_{\mathbf{A}}\mathbf{V}$$

$$\mathbf{A} \in (0,1)^{[T_Q \times T_V]}$$

$$\mathbf{V} \in \mathbb{R}^{[T_V \times D_V]}$$

$$\mathbf{K} \in \mathbb{R}^{[T_V \times D_Q]}$$

$$\mathbf{Q} \in \mathbb{R}^{[T_Q \times D_Q]}$$

# Definition & Properties

Multi-head Attention

- Let $\mathbf{V}$ be a matrix of **value** vectors
  - It has a sequence length of $T_V$
  - It has a dimensionality of $D_V$
- Let $\mathbf{K}$ be a matrix of **key** vectors
  - It has a sequence length of $T_V$
  - It has a dimensionality of $D_Q$
- Let $\mathbf{Q}$ be a matrix of **query** vectors
  - It has a sequence length of $T_Q$
  - It has a dimensionality of $D_Q$
- Let $f(\mathbf{Q}, \mathbf{K})$ be some kernel function
  - Read: similarity function

$$\mathtt{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \underbrace{\mathtt{softmax}\left(f\left(\mathbf{Q}, \mathbf{K}\right)\right)}_{\mathbf{A}}\mathbf{V}$$

$$\mathbf{A} \in (0,1)^{[T_Q \times T_V]}$$
$$\mathbf{V} \in \mathbb{R}^{[T_V \times D_V]}$$
$$\mathbf{K} \in \mathbb{R}^{[T_V \times D_Q]}$$
$$\mathbf{Q} \in \mathbb{R}^{[T_Q \times D_Q]}$$

# Non-Transformer Examples

Multi-head Attention

- $\mathbf{V}$ contains information
- $\mathbf{K}$ contains information about information (i.e, metadata)
- $\mathbf{Q}$ contains metadata about what we want from $\mathbf{V}$
- $f(\mathbf{Q}, \mathbf{K})$ is high when $\mathbf{Q}$ is similar to $\mathbf{K}$

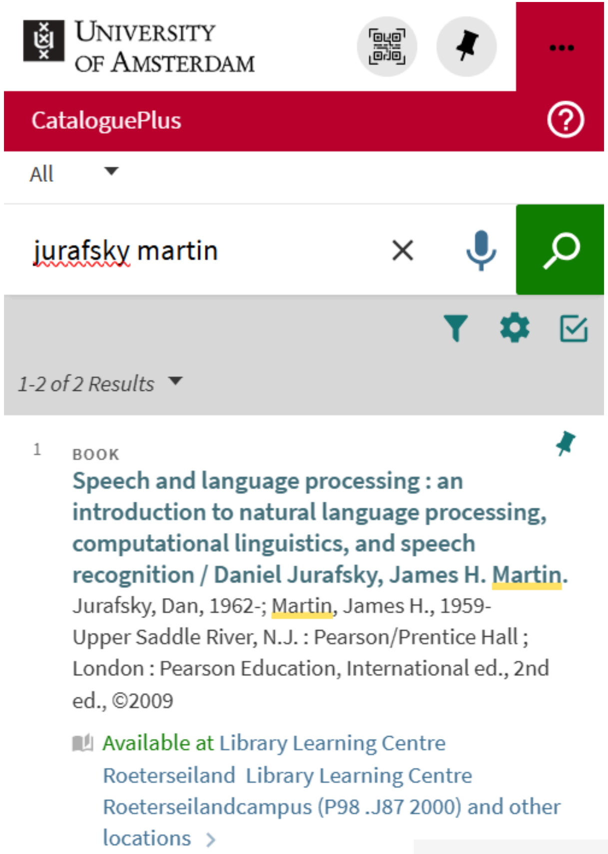
# Non-Transformer Examples

Multi-head Attention

- $\mathbf{V}$ contains information
- $\mathbf{K}$ contains information about information (i.e, metadata)
- $\mathbf{Q}$ contains metadata about what we want from $\mathbf{V}$
- $f(\mathbf{Q}, \mathbf{K})$ is high when $\mathbf{Q}$ is similar to $\mathbf{K}$

**E Soft lookup**

We want to find a textbook about NLP in the library $(\mathbf{V})$. We search for titles $(\mathbf{K})$ with "jurafsky" and "martin" as authors $(\mathbf{Q})$. The computer returns books with similar titles $(f)$



# Non-Transformer Examples

Multi-head Attention

- $f(\mathbf{Q}, \mathbf{K})$ is high when $\mathbf{Q}$ is similar to $\mathbf{K}$
- The output of $f$ must a matrix of size $\mathbf{A} \in (0,1)^{[T_Q \times T_V]}$

# Non-Transformer Examples

Multi-head Attention

- $f(\mathbf{Q}, \mathbf{K})$ is high when $\mathbf{Q}$ is similar to $\mathbf{K}$
- The output of $f$ must a matrix of size $\mathbf{A} \in (0,1)^{[T_Q \times T_V]}$

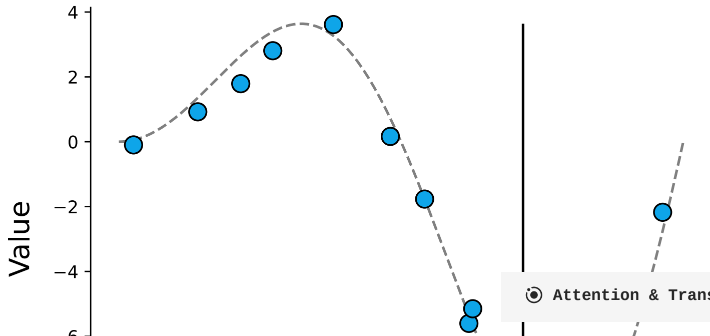
**E Nadaraya-Watson Kernel Regression**

We have some sequence of values
$\mathcal{D} = [(1.36, 1.79), (3.40, -1.77) \ldots, (6.05, -2.17)]$

We want to predict a new sample at $x = 4.21$

We compute the negative Euclidean distance of our new sample with all training samples ($f$). We normalize the outputs to lie between $(0,1)$

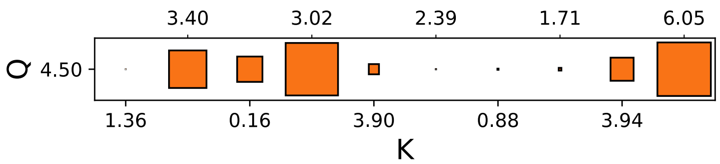
We compute our predicted value as the mean of the seen values, weighted by the computed similarities



# Non-Transformer Examples

Multi-head Attention

- $f(\mathbf{Q}, \mathbf{K})$ is high when $\mathbf{Q}$ is similar to $\mathbf{K}$
- The output of $f$ must a matrix of size $\mathbf{A} \in (0,1)^{[T_Q \times T_V]}$



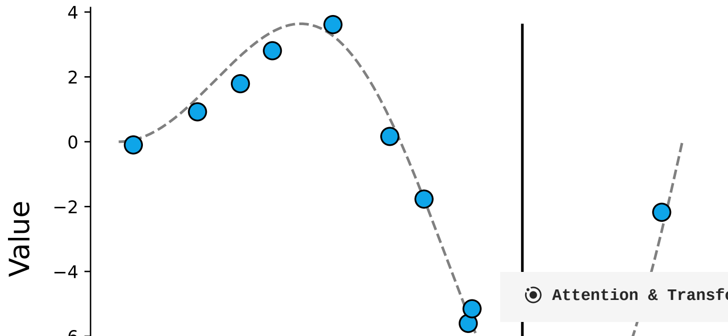

**E Nadaraya-Watson Kernel Regression**

We have some sequence of values
$\mathcal{D} = [(1.36, 1.79), (3.40, -1.77) \dots, (6.05, -2.17)]$

We want to predict a new sample at $x = 4.21$

We compute the negative Euclidean distance of our new sample with all training samples ($f$). We normalize the outputs to lie between $(0, 1)$

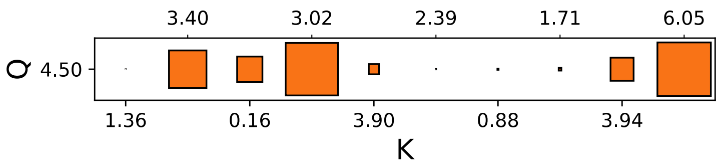
We compute our predicted value as the mean of the seen values, weighted by the computed similarities



# Non-Transformer Examples

Multi-head Attention

- $f(\mathbf{Q}, \mathbf{K})$ is high when $\mathbf{Q}$ is similar to $\mathbf{K}$
- The output of $f$ must a matrix of size $\mathbf{A} \in (0,1)^{[T_Q \times T_V]}$



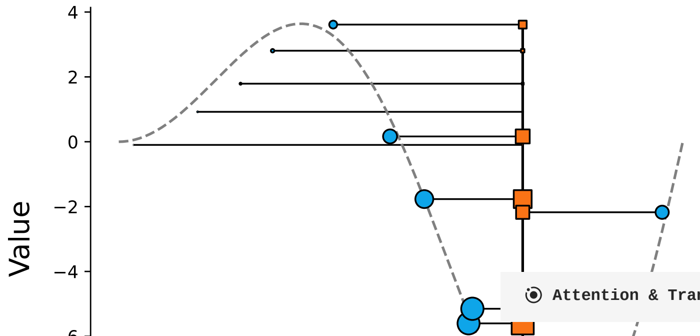

**E Nadaraya-Watson Kernel Regression**

We have some sequence of values
$\mathcal{D} = [(1.36, 1.79), (3.40, -1.77) \ldots, (6.05, -2.17)]$

We want to predict a new sample at $x = 4.21$

We compute the negative Euclidean distance of our new sample with all training samples ($f$). We normalize the outputs to lie between $(0, 1)$

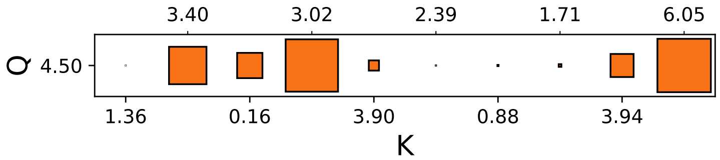
We compute our predicted value as the mean of the seen values, weighted by the computed similarities



# Non-Transformer Examples

Multi-head Attention

- $f(\mathbf{Q}, \mathbf{K})$ is high when $\mathbf{Q}$ is similar to $\mathbf{K}$
- The output of $f$ must a matrix of size $\mathbf{A} \in (0,1)^{[T_Q \times T_V]}$



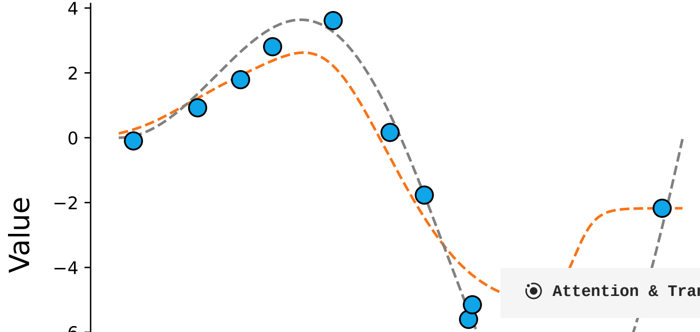

**E Nadaraya-Watson Kernel Regression**

We have some sequence of values
$\mathcal{D} = [(1.36, 1.79), (3.40, -1.77) \ldots, (6.05, -2.17)]$

We want to predict a new sample at $x = 4.21$

We compute the negative Euclidean distance of our new sample with all training samples ($f$). We normalize the outputs to lie between $(0, 1)$

We compute our predicted value as the mean of the seen values, weighted by the computed similarities



# Non-Transformer Examples

## Multi-head Attention

# Non-Transformer Examples

Multi-head Attention

# Attention in Transformers

## Multi-head Attention

# Multi-head Attention

Multi-head Attention

# Add & Norm

# Residual Connections

Add & Norm

# LayerNorm

Add & Norm

These are the equations

$$
\begin{aligned}
\mathbf{X}_l \;=\; & \texttt{LayerNorm}( \\
& \quad \mathbf{X}_{l-1} + \texttt{SubLayer}\left(\mathbf{X}_{l-1}\right) \\
& )
\end{aligned}
$$

# Feed Forward

# Embedding

# Position Encoding

# Tokenization

# Training Transformers

# The End