MI2RL Medical Imaging Intelligent Reality Lab

ASAN Medical Center

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

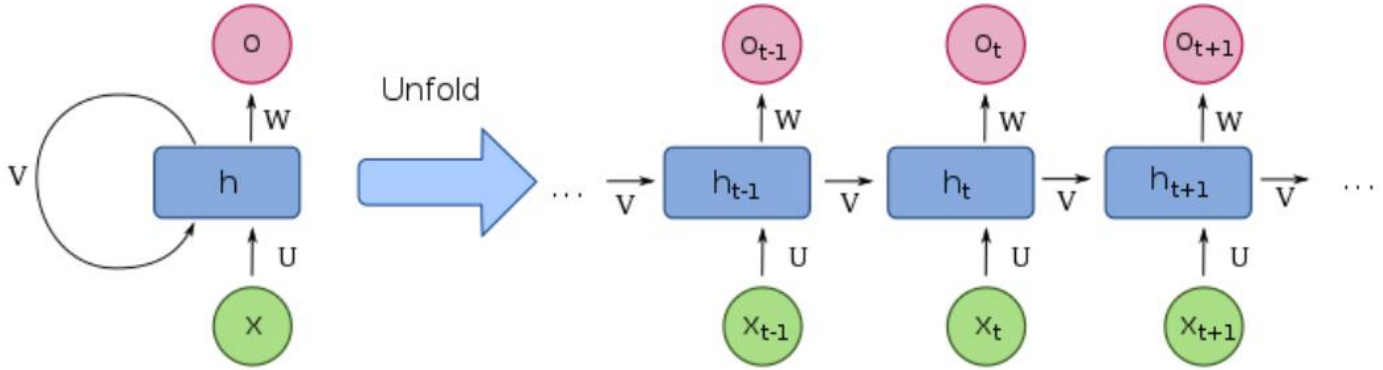
2025.05.28

김연수

Contents

Introduction

- **Sequence Modeling**
  - Sequence를 가지는 데이터로부터 또 다른 Sequence를 가지는 데이터를 생성하는 task
  - Ex) machine translation, chatbot
  - Recurrent neural network, long short-term memory, gated recurrent neural network



- **Attention**
  - Allowed modeling dependencies regardless of distance
  - Mostly used in conjunction with recurrent networks

- **Transformer**
  - Eschews recurrence entirely → relies **only on attention**
  - Draws global dependencies
  - Significantly more **parallelization**
  - **New state-of-the-art** in translation, much faster training

Background

- Prior Approaches (RNNs, LSTMs, GRUs, CNNs like ByteNet/ConvS2S) relied on sequential computation (RNNs) or limited-range convolutions.
- Difficulty in learning long-range dependencies efficiently due to sequential nature or distance-dependent operations.
- **Attention** enabled modeling dependencies regardless of distance, but mostly used as an add-on to recurrent networks.
- The **Transformer**: First model to rely entirely on Self-Attention (Intra-Attention).
- Achieves a constant number of operations for any two positions, drastically **improving parallelization**.
- **Superior performance** and **significantly faster training** by completely eschewing recurrence and convolutions.

Model Architecture

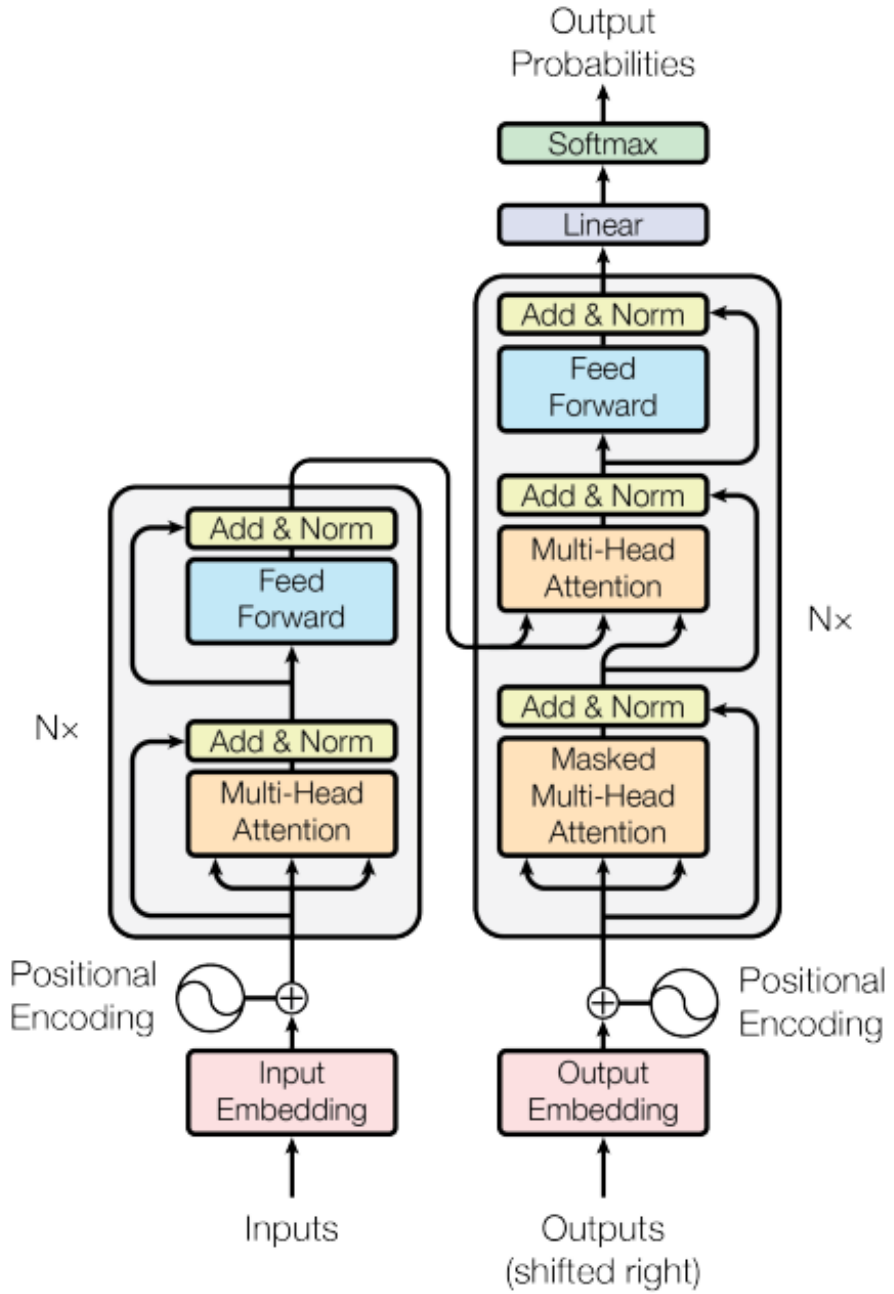
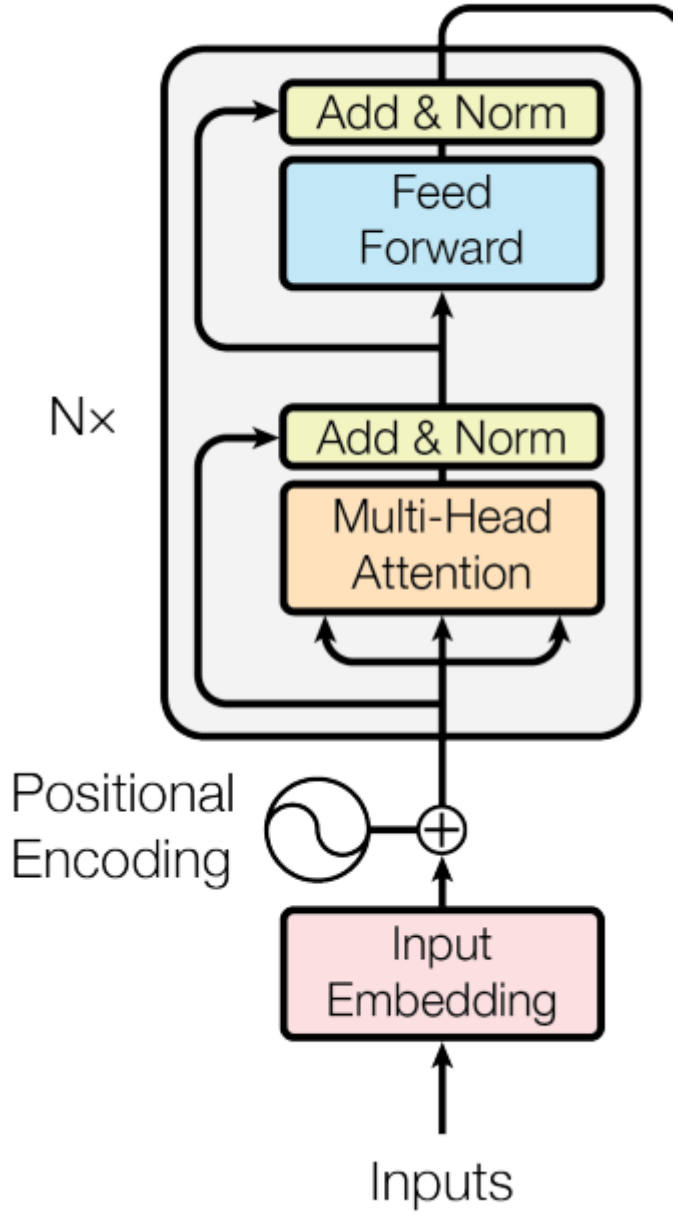


Figure 1: The Transformer - model architecture.

Encoder and Decoder Stacks

Encoder

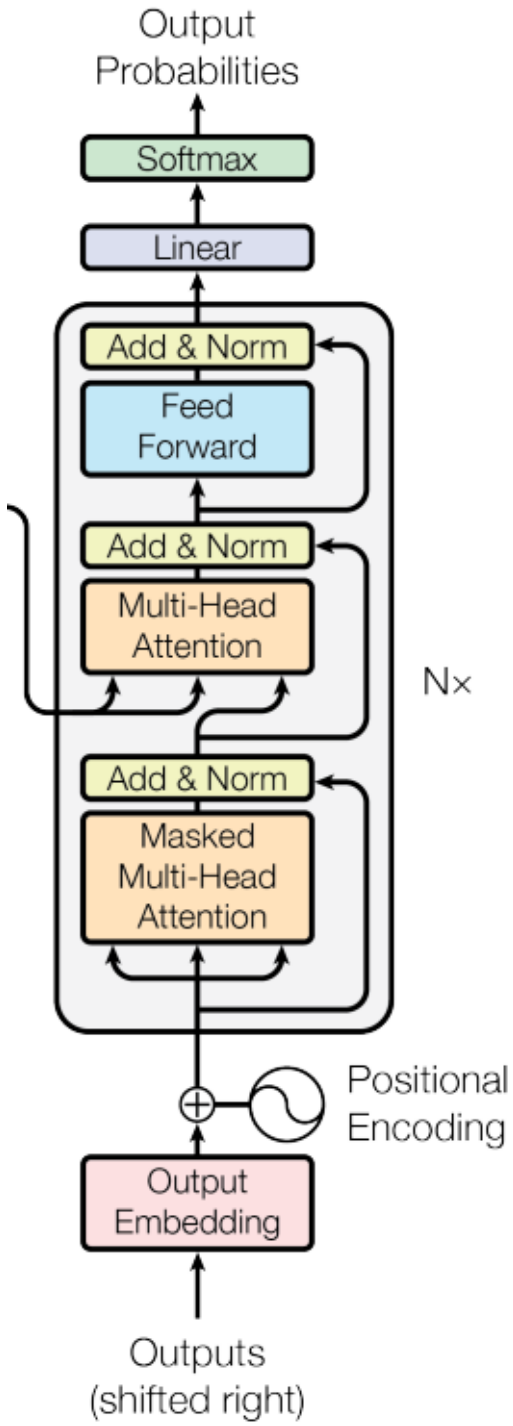
- Composed of a stack of $N = 6$ identical layers
- Each layer has two sub-layers
  - Multi-head self-attention mechanism
  - Simple, position-wise fully connected feed-forward network
- Residual connections & Layer normalization
  - Implemented as LayerNorm$\left(x + Sublayer(x)\right)$
  - Facilitates training of deep networks and ensures stability
  - All sub-layer outputs have dimension d_{model} = 512



Encoder and Decoder Stacks

Decoder

- Generates an output sequence of symbols one element at a time, based on z from the encoder
- Also composed of a stack of $N = 6$ identical layers
- Each layer: has three sub-layers
  - Masked multi-head self-attention
  - Multi-head encoder-decoder attention
  - Position-wise fully connected feed-forward network
- Residual connections & layer normalization



Attention

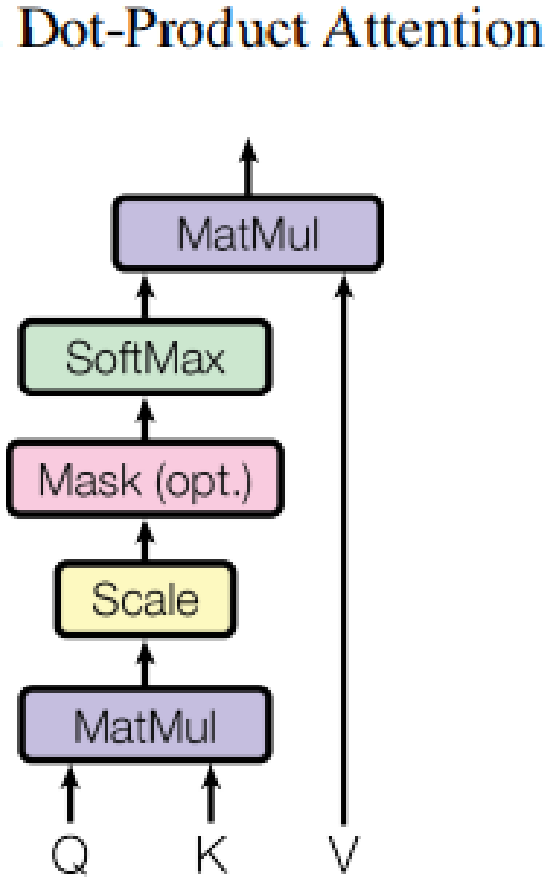
Attention function

- Maps a Query (Q) and a set of Key-Value (K-V) pairs to an output
- All are vectors
- Output = weighted sum of values (weights computed by compatibility of Q with K)

Scaled Dot-Product Attention

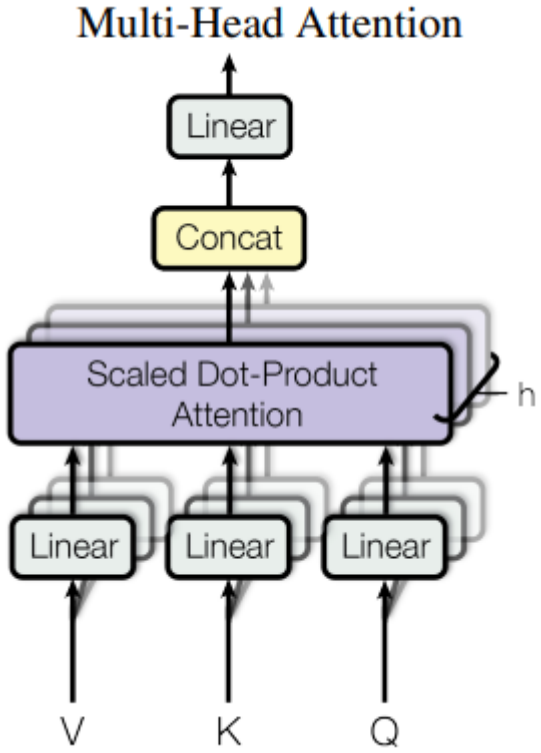
- Computes dot products of Q with all K
- Divides result by $\sqrt{d_k}$ for stable training
- Applies Softmax function to obtain weights, then weighted sum with V
- Outperforms unscaled Dot-Product Attention for larger d_k values

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



Multi-Head Attention

- To overcome limitations of single attention (reduced effective resolution due to averaging)
- Linearly projects Q, K, V h times with different, learned projection into h 'heads'
- Performs attention function in parallel on each projected version
- Concatenates the outputs from each head
- Applies a final linear projection to produce to produce the ultimate result



$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$

$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Applications of Attention in our Model

The Transformer uses multi-head attention in three different ways

1. **Encoder-decoder attention layers**
   - Queries: come from the previous decoder layer
   - Keys/Values: come from the output of the encoder stack
   - Allows every position in the decoder to attend over all positions in the input sequence
2. **Encoder self-attention layers**
   - Queries/Keys/Values: come from the same place(output of the previous layer in the encoder)
   - Each position in the encoder can attend to all positions in the previous layer of the encoder
3. **Decoder self-attention layers**
   - Queries/Keys/Values: come from the output of the previous layer in the decoder
   - Masking applied: prevents positions from attending to subsequent positions
   - Preserves the auto-regressive property(prevents leakage of future information)

Position-wise Feed-Forward Networks

- Contained within each layer of both encoder and decoder
- Applied to each position separately and identically
- Consists of two linear transformations with a ReLU activation in between.

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

- Input/output dimension: $d_{model} = 512$, inner-layer dimension: : $d_{ff} = 2048$

Embeddings and Softmax

- **Embeddings**
  - Learned embeddings convert input and output tokens into d_{model}-dimensional vectors
  - Weight matrix shared between the two embedding layers and the pre-softmax linear transformation
  - Weights in embedding layers are multiplied by $\sqrt{d_{model}}$

- **Softmax**
  - Converts the decoder output to predicted next-token probabilities

Positional Encoding

- Transformer lacks recurrence/convolution, cannot inherently utilize sequence order
- Injects information about the relative or absolute position of tokens in the sequence
- Added to the input embeddings at the bottom of the encoder and decoder stacks
- Same dimension (d_{model} = 512) as embedding ($PE + Embedding$)

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

Why Self-Attention

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Training

- Training Data and Batching
  - English-German: WMT 2014, ~4.5M sentence pairs, 37k shared byte-pair vocabulary.
  - English-French: WMT 2014, 36M sentences, 32k word-piece vocabulary.
  - Batched by approximate sequence length (25k source + 25k target tokens per batch).
- Hardware and Schedule
  - One machine with 8 NVIDIA P100 GPUs
  - Base models: 100,000 steps, 0.4 sec/step
  - Big models: 300,000, 1.0 sec/step
- Optimizer
  - Optimizer: Adam optimizer((β1=0.9, β2=0.98, ϵ=10−9)
  - Learning Rate $lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$
- Regularization
  - Residual dropout
  - Label smoothing

Results

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Table 3: Variations on the Transformer architecture. Unlisted values are identical to those of the base model. All metrics are on the English-to-German translation development set, newstest2013. Listed perplexities are per-wordpiece, according to our byte-pair encoding, and should not be compared to per-word perplexities.

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	positional embedding instead of sinusoids									4.92	25.7	
big	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

Table 4: The Transformer generalizes well to English constituency parsing (Results are on Section 23 of WSJ)

Parser	Training	WSJ 23 F1
Vinyals & Kaiser el al. (2014) [37]	WSJ only, discriminative	88.3
Petrov et al. (2006) [29]	WSJ only, discriminative	90.4
Zhu et al. (2013) [40]	WSJ only, discriminative	90.4
Dyer et al. (2016) [8]	WSJ only, discriminative	91.7
Transformer (4 layers)	WSJ only, discriminative	91.3
Zhu et al. (2013) [40]	semi-supervised	91.3
Huang & Harper (2009) [14]	semi-supervised	91.3
McClosky et al. (2006) [26]	semi-supervised	92.1
Vinyals & Kaiser el al. (2014) [37]	semi-supervised	92.1
Transformer (4 layers)	semi-supervised	92.7
Luong et al. (2015) [23]	multi-task	93.0
Dyer et al. (2016) [8]	generative	93.3

Conclusion

- **Transformer** is the first sequence transduction model based **entirely on attention**, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention

- Can be **significantly faster** than architectures based on recurrent or convolutional layers.
- Achieved **new state-of-the-art** on WMT 2015 English-to-German & English-to-French translation.
- Best model outperformed all previous ensembles on English-to-German

Attention Visualizations

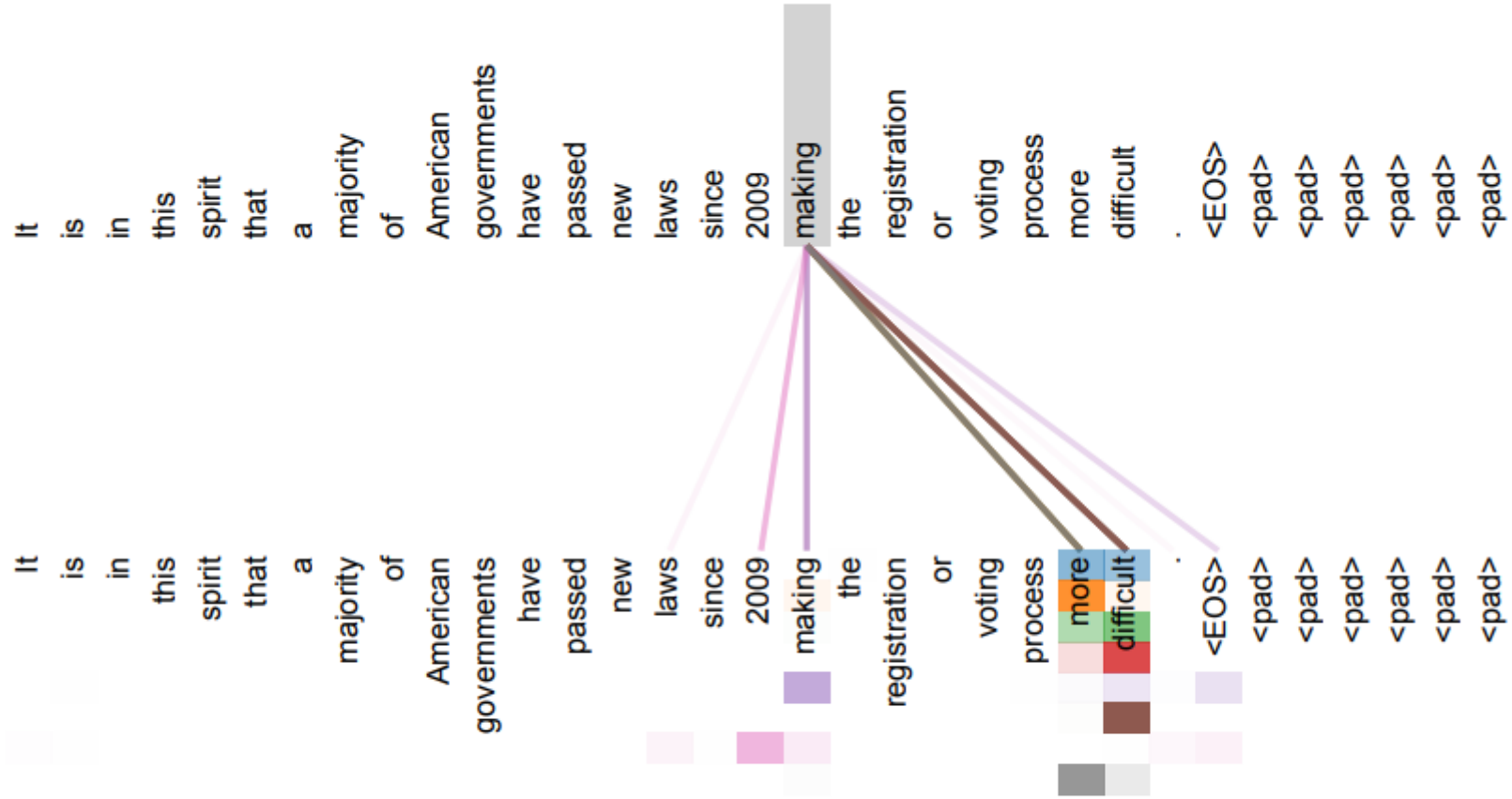


Figure 3: An example of the attention mechanism following long-distance dependencies in the encoder self-attention in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb 'making', completing the phrase 'making...more difficult'. Attentions here shown only for the word 'making'. Different colors represent different heads. Best viewed in color.

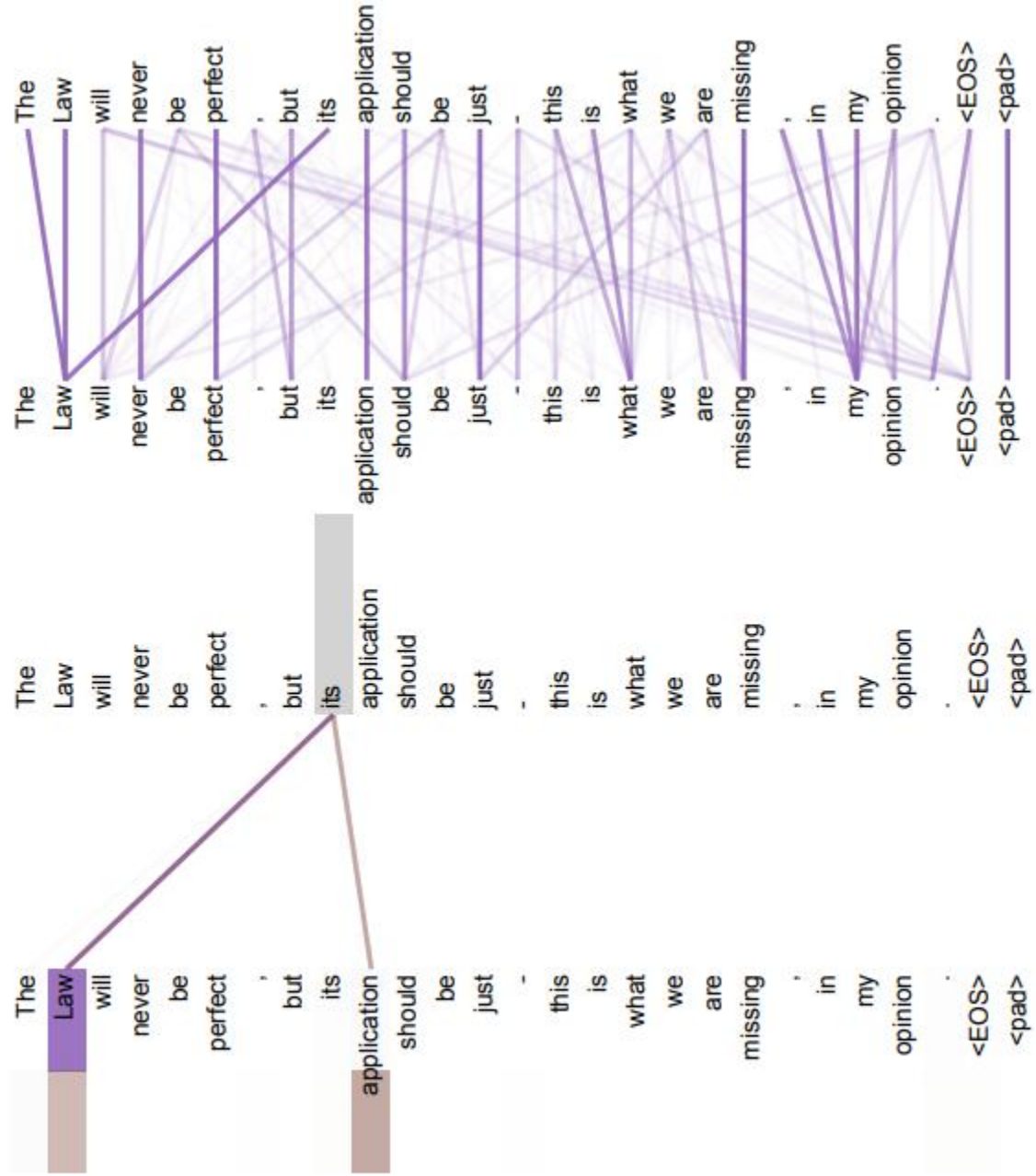


Figure 4: Two attention heads, also in layer 5 of 6, apparently involved in anaphora resolution. Top: Full attentions for head 5. Bottom: Isolated attentions from just the word 'its' for attention heads 5 and 6. Note that the attentions are very sharp for this word.

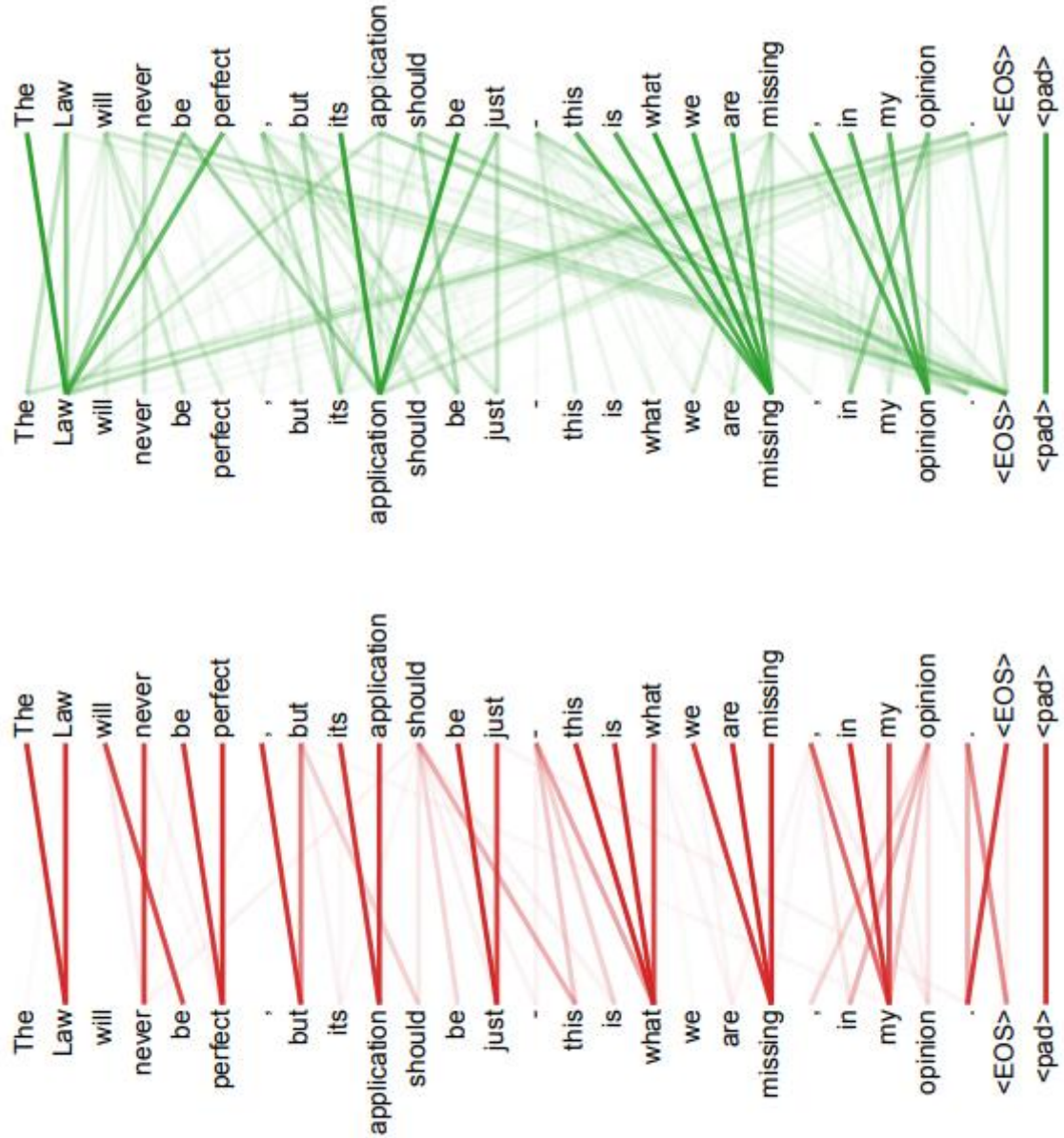


Figure 5: Many of the attention heads exhibit behaviour that seems related to the structure of the sentence. We give two such examples above, from two different heads from the encoder self-attention at layer 5 of 6. The heads clearly learned to perform different tasks.