Transformer

[CSED490X] Recent Trends in ML: A Large-Scale Perspective

Jungtaek Kim

jtkim@postech.ac.kr

POSTECH
Pohang 37673, Republic of Korea
https://jungtaek.github.io

March 23, 2022

Table of Contents

Introduction

The Transformer era has begun!

Transformer

Figure 1: Sentinel Prime.

- Transformer has been introduced in the work by Vaswani et al. [2017].
- Sequence modeling and sequence transduction problems are solved in this paper.
- Language modeling and machine translation are target applications of the vanilla Transformer architecture.
- Following the modern sequence modeling scheme, it devises an encoder-decoder architecture.

Figure 1 is taken from Wikipedia.

[Vaswani et al., 2017] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems (NeurIPS), volume 30, pages 5998–6008, Long Beach, California, USA, 2017.

Today's Lecture

- We will cover the tasks solved in this paper first.
- Then, we will study datasets for machine translation.
- Before introducing the Transformer model, we will visit traditional models for sequence modeling.
- Eventually, we will study the Transformer model and a learning algorithm, used in this paper.
- Finally, we will investigate the experimental results.

Tasks

Tasks

- The vanilla Transformer is used in solving a machine translation problem.

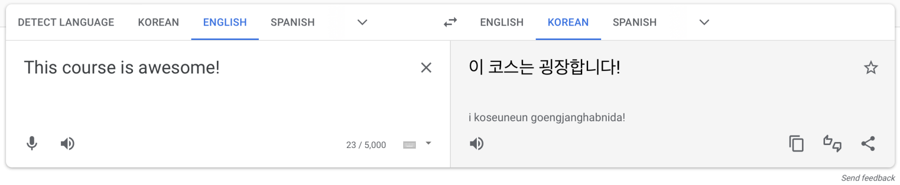


Figure 2: English to Korean Translation.

- The sentence given, “This course is awesome!” is tokenized as “this”, “course”, “is”, “awesome”, “!”.
- After tokenization, each token is expressed as a one-hot encoded vector with a vocabulary of tokens.

Figure 2 is taken from Google Translate.

Vocabulary of Tokens

Vocabulary of Tokens

	⟨EOS⟩	you	they	this	it	is	are	course	class	great	awesome	.	!	?
this	0	0	0	1	0	0	0	0	0	0	0	0	0	0
course	0	0	0	0	0	0	0	1	0	0	0	0	0	0
is	0	0	0	0	0	1	0	0	0	0	0	0	0	0
awesome	0	0	0	0	0	0	0	0	0	0	1	0	0	0
!	0	0	0	0	0	0	0	0	0	0	0	0	1	0
⟨EOS⟩	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3: Tokenization of "This course is awesome!".

Datasets

Datasets

- WMT 2014 English-German and English-French datasets are used.
- The standard WMT 2014 English-German dataset consists of about 4.5 million sentence pairs.
- The WMT 2014 English-French dataset consists of 36 million sentence pairs.
- Each training batch contains a set of sentence pairs containing approximately 25,000 source tokens and 25,000 target tokens.

A Machine Learning Model

Recurrent Neural Networks

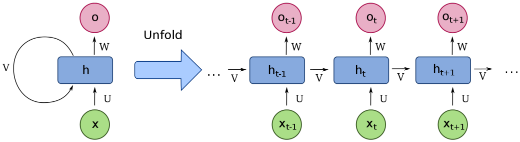


Figure 4: Illustration of recurrent neural networks.

- It is a class of neural networks, which has connections between nodes from a directed graph along a sequence.
- It can learn a temporal dynamic behavior for a sequence.
- Long short-term memory (LSTM) and gated recurrent unit (GRU) have been proposed.

Figure 4 is taken from Wikipedia.

Recurrent Neural Networks

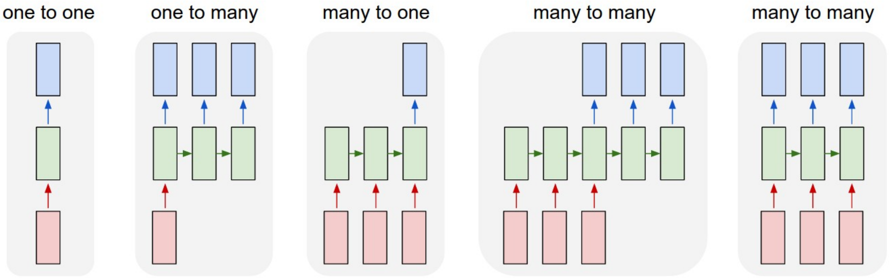


Figure 5: Types of recurrent neural networks.

Taken from `https://karpathy.github.io/2015/05/21/rnn-effectiveness/`.

Sequence-to-Sequence Models

- A sequence-to-sequence model is designed as an encoder-decoder architecture.
- It is defined as a conditional language model, which is conditioned on the previously-generated word sequence of target language and a sentence of source language.

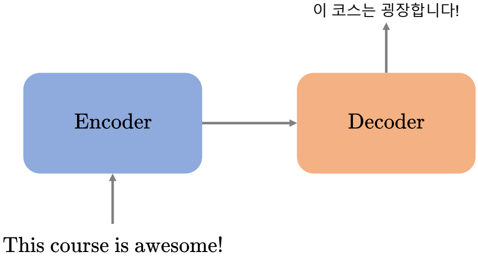
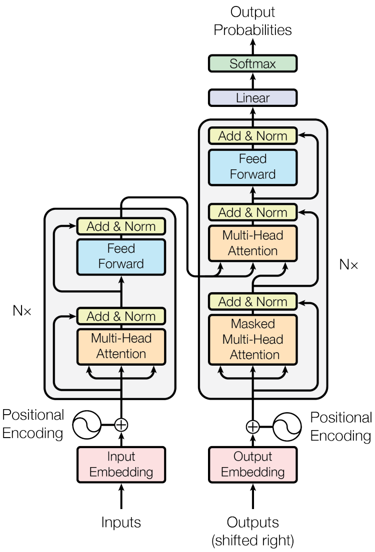


Figure 6: Encoder-decoder architecture.

Transformer



Embedding Layer

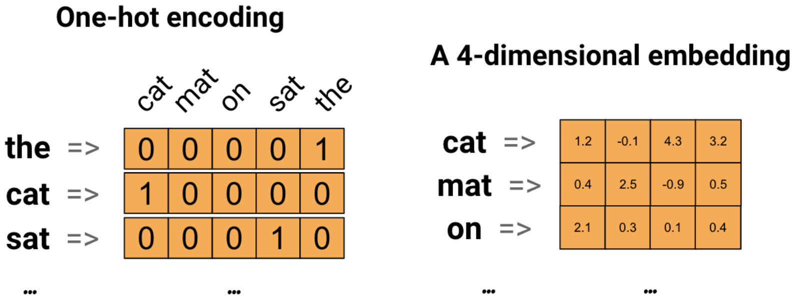
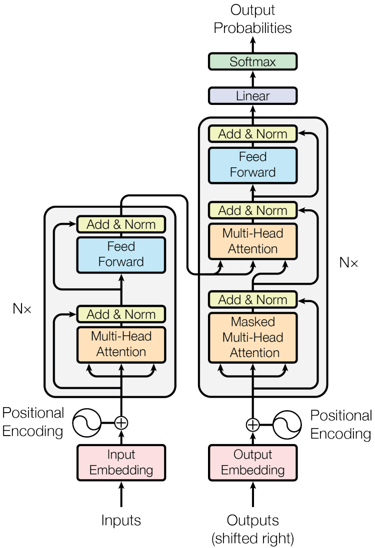


Figure 7: One-hot encoding and a 4-dimensional embedding.

- It transforms a one-hot encoded vector to a d-dimensional embedding vector.
- This transformation is usually initialized at random.

Figure 7 is taken from this link.

Transformer



Positional Encoding

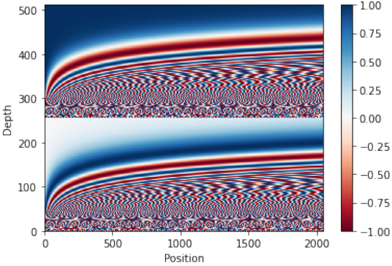


Figure 8: Visualization of positional encoding.

- The Transformer model does not contain recurrence and convolution, even though it is to model a sequence.
- Thus, it requires the information about the relative or absolution position of the tokens in a sequence.
- Positional encoding is defined as

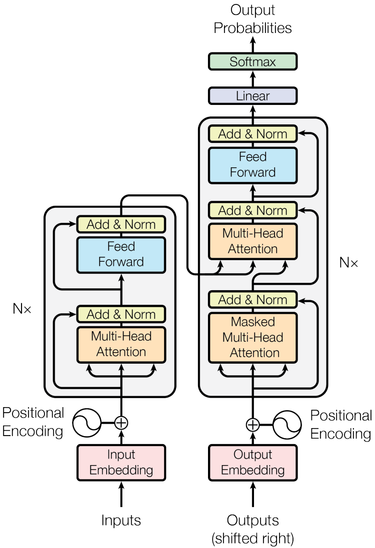
$$\mathrm{PE}_{(\mathrm{pos},2i)} = \sin\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right), \qquad (1)$$

$$\mathrm{PE}_{(\mathrm{pos},2i+1)} = \cos\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right), \qquad (2)$$

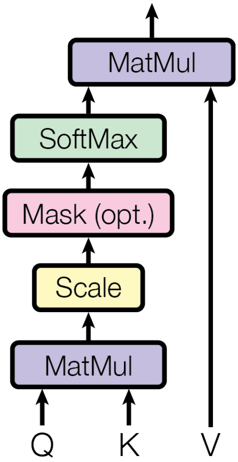
where pos is the position, i is the dimension, and d_{model} is the dimensionality of the model.

Figure 8 is taken from this link.

Transformer



Scaled Dot-Product Attention



- The input consists of queries and keys of dimension d_k, and values of dimension d_v.
- It computes the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values.
- Finally, scaled dot-product attention is defined as

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{3}$$

where Q, K, and V are query, key, and value matrices, respectively.

Dot Product & Cosine Similarity

- Suppose that $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ are d-dimensional vectors.
- Dot product is defined as

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^\top \mathbf{y} = \|\mathbf{x}\| \|\mathbf{y}\| \cos\theta. \tag{4}$$

- Cosine similarity is defined as

$$\cos\theta = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}. \tag{5}$$

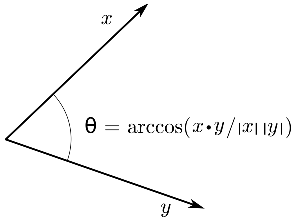


Figure 9: Illustration of dot product and cosine similarity.

Why Does Scaled Dot-Product Attention Rescale by $1/\sqrt{d_k}$?

Why 1 / sqrt(D) rescaling? And not 1 / D or something else?

Let query / key elements be unit Gaussians.

$$Q_{ij} \sim \mathcal{N}(0,1) \quad K_{ij} \sim \mathcal{N}(0,1)$$

Variance of product of two unit Gaussians.

$$\mathrm{Var}(Q_{il}K_{lj}) = 1$$

Variance of query / key inner product.

$$\mathrm{Var}(Q_i K_j^T) = \mathrm{Var}(\textstyle\sum_l^D Q_{il}K_{lj}) = D$$

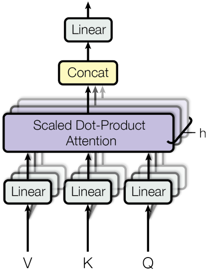
Standard deviation is then

$$\mathrm{Std}(Q_i K_j^T) = \sqrt{D}$$

$\boxed{\text{Therefore } QK^T \rightarrow \frac{QK^T}{\sqrt{D}}}$

Taken from the Misha Laskin's slides.

Multi-Head Attention



- Instead of performing a single attention function with d_{model}-dimensional keys, values, and queries, they are linearly projected h times with different, learned linear projects to d_k, d_k, and d_v dimensions, respectively.
- On each of these projected versions of queries, keys, and values, the attention function is performed in parallel.
- Multi-head attention is defined as

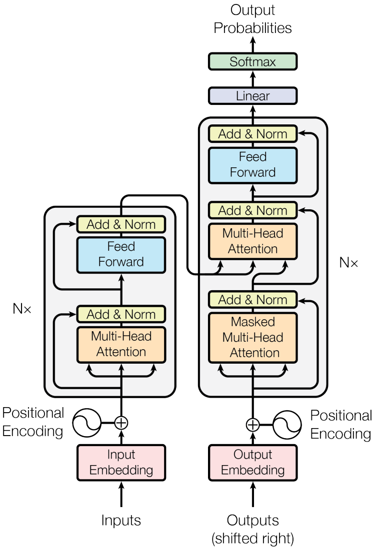
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O, \tag{6}$$

where

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \tag{7}$$

$W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

Transformer



Add & Norm

- It employs a residual connection [He et al., 2016] and and layer normalization:

$$\mathrm{LayerNorm}(x + \mathrm{Sub\text{-}layer}(x)), \tag{8}$$

where $\mathrm{Sub\text{-}layer}$ is a sub-layer for applying some transformations.

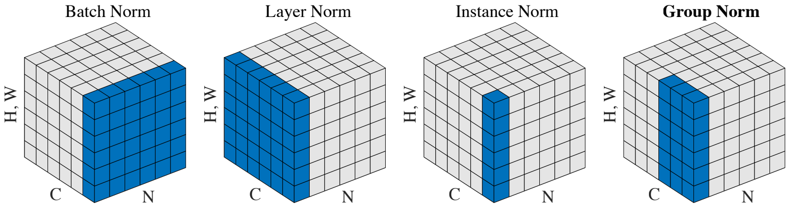


Figure 10: Various normalization techniques.

Figure 10 is taken from [Wu and He, 2018].

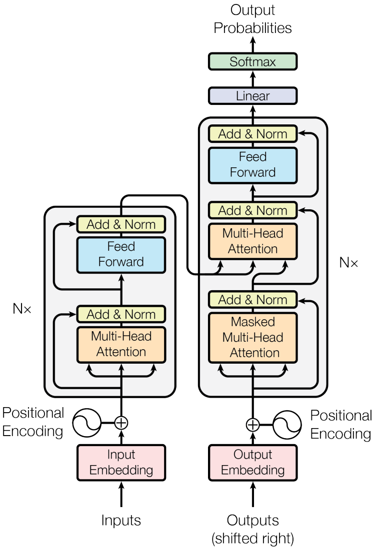
Position-wise Feed-Forward Networks

- Each of the layers in an encoder and a decoder contains a fully-connected feed-forward network.
- It is applied to each position separately and identically.
- A position-wise feed-forward network (FFN) is defined as

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2, \tag{9}$$

where $W_1 \in \mathbb{R}^{d_{\mathrm{model}} \times d_{\mathrm{FFN}}}$, $b_1 \in \mathbb{R}^{d_{\mathrm{FFN}}}$, $W_2 \in \mathbb{R}^{d_{\mathrm{FFN}} \times d_{\mathrm{model}}}$, and $b_2 \in \mathbb{R}^{d_{\mathrm{model}}}$. Note that $\max(0, x)$ is identical to $\mathrm{ReLU}(x)$.

Transformer



Output Probabilities

- Watch this video to check out how it works.

A Learning Algorithm

Hardware & Schedule

- This model is trained on 8 NVIDIA P100 GPUs.
- The base model is trained for a total of 100,000 steps or 12 hours, where each training step takes about 0.4 seconds.
- The big model is trained for 300,000 steps or 3.5 days, where each step takes about 1.0 seconds.

Optimizer & Regularization

Optimizer

- Adam optimizer [Kingma and Ba, 2015] with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$ is used.
- A learning rate is scheduled as

$$\text{learning\_rate} = d_{\text{model}}^{-0.5} \min(t^{-0.5}, t\tau^{-1.5}), \tag{10}$$

at step t, where τ is a warm-up step.

Regularization

- Residual dropout and label smoothing are used as regularization techniques.

Experimental Results

BLEU Score

- BLEU score [Papineni et al., 2002] is an evaluation metric for machine translation, which is capable of replacing expensive human evaluations.
- It is defined as

$$\mathrm{BLEU} = \mathrm{BP}\exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{11}$$

where

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r, \\ \exp(1 - r/c) & \text{otherwise}, \end{cases} \tag{12}$$

$$p_n = \frac{\sum_{\mathcal{C} \in \text{Candidates}} \sum_{\text{n-gram} \in \mathcal{C}} \text{Count}_{\text{clip}}(\text{n-gram})}{\sum_{\mathcal{C}' \in \text{Candidates}} \sum_{\text{n-gram}' \in \mathcal{C}'} \text{Count}(\text{n-gram}')}, \tag{13}$$

w_n is a weight for n-gram. Note that c and r are the length of the candidate translation and the effective reference corpus length, respectively.

BLEU Score

Example 1.

Candidate 1: `It is a guide to action which ensures that the military always obeys the commands of the party.`

Candidate 2: `It is to insure the troops forever hearing the activity guidebook that party direct.`

Reference 1: `It is a guide to action that ensures that the military will forever heed Party commands.`

Reference 2: `It is the guiding principle which guarantees the military forces always being under the command of the Party.`

Reference 3: `It is the practical guide for the army always to heed the directions of the party.`

- Two candidates might be all acceptable.
- However, by comparing to reference sentences, counting the number of shared words indicates Candidate 1 is good and Candidate 2 is bad.
- It implies that Candidate 1 is closer to the reference sentences than Candidate 2.

BLEU Score

Example 2.

Candidate: `the` `the` `the the the the the.`

Reference 1: `The cat is on the mat.`

Reference 2: `There is a cat on the mat.`

- Simple counting does not reflect the quality of candidate sentence properly.
- Thus, the number of shared words is clipped by the maximum count in each of the reference sentences.
- Modified unigram precision is $2/7$.

BLEU Score

Example 3:

Candidate: `of the`

Reference 1: `It is a guide to action that ensures that the military will forever heed Party commands.`

Reference 2: `It is the guiding principle which guarantees the military forces always being under the command of the Party.`

Reference 3: `It is the practical guide for the army always to heed the directions of the party.`

- Since Candidate is too short compared to the reference sentences, modified n-gram precision fails to evaluate properly.
- Modified unigram precision is 1.0 and modified bigram precision is 1.0 as well.

Experimental 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	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Experimental Results

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

Experimental Results

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

Experimental Results

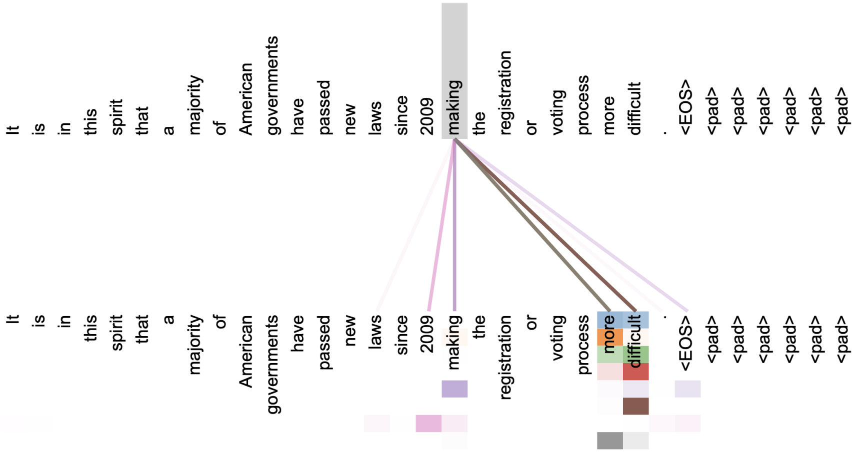


Figure 3: An example of the attention mechanism following long-distance dependencies in the encoder self-attention in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb 'making', completing the phrase 'making...more difficult'. Attentions here shown only for the word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

Experimental Results

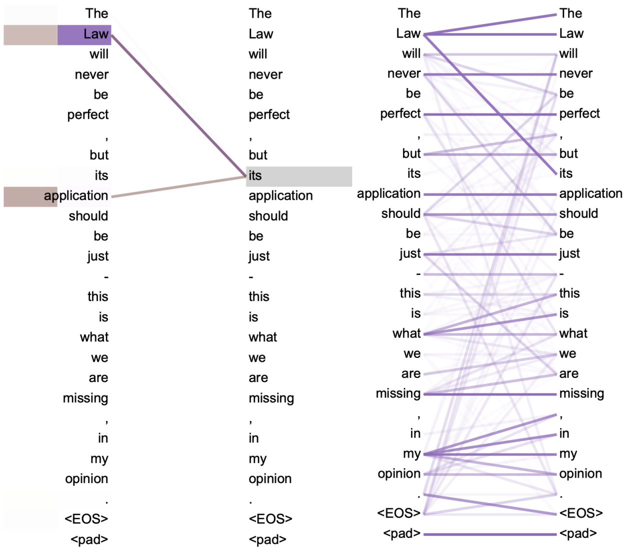


Figure 4: Two attention heads, also in layer 5 of 6, apparently involved in anaphora resolution. Top: Full attentions for head 5. Bottom: Isolated attentions from just the word 'its' for attention heads 5 and 6. Note that the attentions are very sharp for this word.

Experimental Results

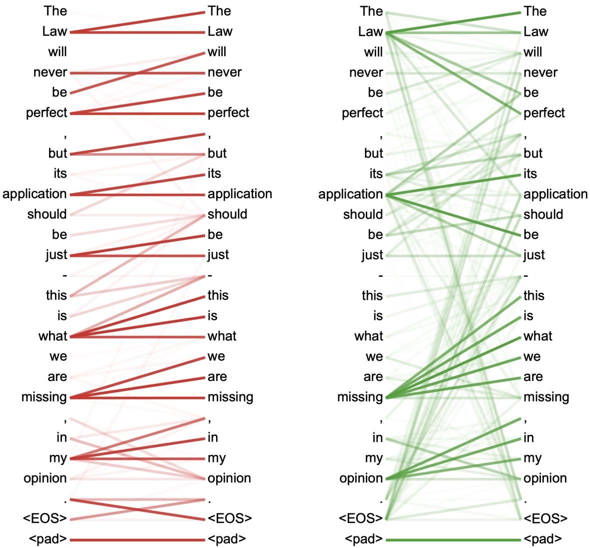


Figure 5: Many of the attention heads exhibit behaviour that seems related to the structure of the sentence. We give two such examples above, from two different heads from the encoder self-attention at layer 5 of 6. The heads clearly learned to perform different tasks.

Any Questions?

References I

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, Nevada, USA, 2016.

D. P. Kingma and J. L. Ba. ADAM: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, California, USA, 2015.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, 2002.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 5998–6008, Long Beach, California, USA, 2017.

Y. Wu and K. He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.