Attention Is All You Need

CARL EDWARDS (CNE2)

Outline

- Introduction and Motivation
- Task
  - BLEU Score
- Background
- Model
- Training
- Results
- Visualizations
- Takeaway

Introduction and Motivation

- Recurrent neural networks (RNN), such as LSTMs and GRUs, are state of the art for language modeling and machine translation
- Recurrent models compute along the sequence's position
  - Cannot be parallelized easily
- Attention models can model dependencies irrespective of distance
  - Generally used with RNNs
- **Key Idea:** Attention is All You Need
- Paper introduces the model "Transformer"

Task – Machine Translation

Goal: Translate from English to German and English to French

Measure: BLEU = BiLingual Evaluation Understudy

BLEU Score

Mathematically, the BLEU score is defined as:

$$\text{BLEU} = \underbrace{\min\Big(1, \exp\big(1 - \frac{\text{reference-length}}{\text{output-length}}\big)\Big)}_{\text{brevity penalty}} \underbrace{\Big(\prod_{i=1}^{4} precision_i\Big)^{1/4}}_{\text{n-gram overlap}}$$

with

$$precision_i = \frac{\sum_{\text{snt}\in\text{Cand-Corpus}} \sum_{i\in\text{snt}} \min(m^i_{cand}, m^i_{ref})}{w^i_t = \sum_{\text{snt'}\in\text{Cand-Corpus}} \sum_{i'\in\text{snt'}} m^{i'}_{cand}}$$

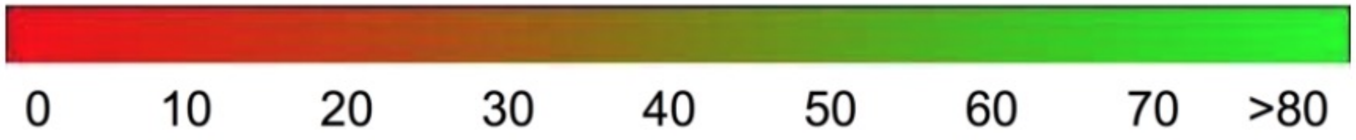
where

- m^i_{cand} is the count of i-gram in candidate matching the reference translation
- m^i_{ref} is the count of i-gram in the reference translation
- w^i_t is the total number of i-grams in candidate translation

BLEU Score Interpretation

BLEU Score	Interpretation
< 10	Almost useless
10 - 19	Hard to get the gist
20 - 29	The gist is clear, but has significant grammatical errors
30 - 40	Understandable to good translations
40 - 50	High quality translations
50 - 60	Very high quality, adequate, and fluent translations
> 60	Quality often better than human

The following color gradient can be used as a general scale interpretation of the BLEU score:



Background

- Some work attempts to reduce sequential computation using convolutional layers
  - Computation is reduced to either linear or logarithmic computation with the distance between sequence symbols.
  - Transformer can do this in constant time
- Self-attention has been used successfully in tasks such as reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations.
- Transformer is the first transduction model to use self-attention without RNNs or convolution.
  - Transduction is used in a linguistic sense.

Encoder-Decoder Background

- Most state of the art models employ an encoder-decoder architecture
  - Input: Sequence of symbolic representations $(x_1, ..., x_n)$
  - Encoder produces latent representations: $\mathbf{z} = (z_1, ..., z_n)$
  - Decoder uses **z** to produce output sequence: $(y_1, ..., y_m)$

Model

- Encoder is on the left; decoder is on the right.
- These layers are stacked 6 times in the Transformer model.
- The decoder looks at the output of the encoder (and the previously generated words)

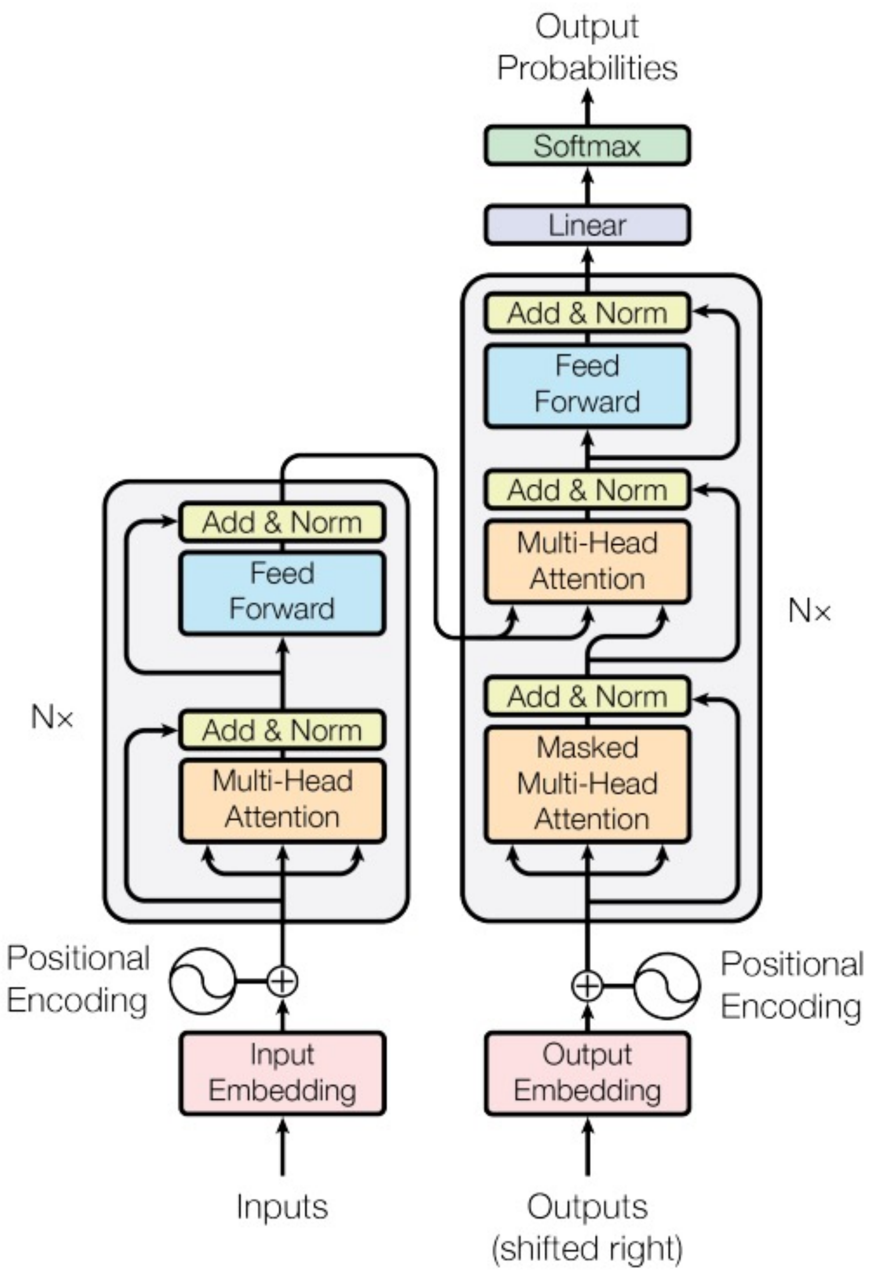
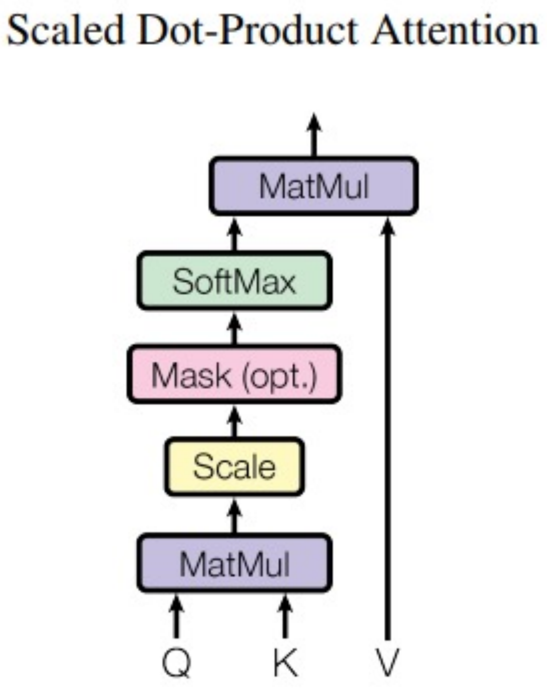


Figure 1: The Transformer - model architecture.

Scaled Dot-Product Attention

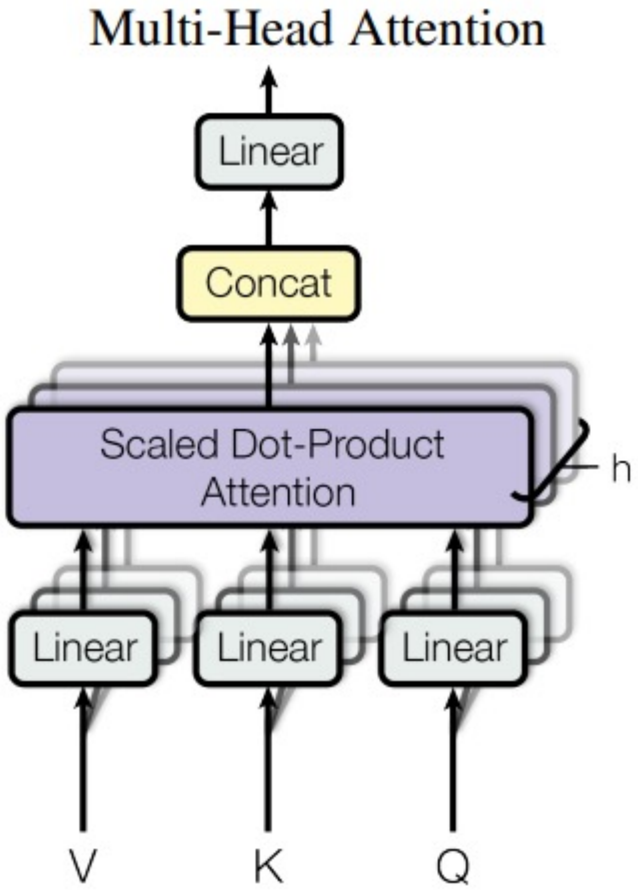
- Learn projections from input representation to
  - Query (Q) (dimension d_k)
  - Key (K) (dimension d_k)
  - Value (V) (dimension d_v)
- Matmul between Q and K are logits for how much attention is needed. Softmax is used to compute weights to average the value representation.
- The paper introduces scaled dot-product attention
  - Dot product attention (multiplicative) had been used without scaling.
  - Observation: Dot product grows too large in magnitude for large number of dimensions, so divide by $\sqrt{d_k}$.



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

Multi-Head Attention

- Averaging inhibits single-head attention from looking at different representation subspaces
- Instead, split single-attention into multiple attentions!
  - Each attention head is computed in parallel



Multi-Head Attention

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

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

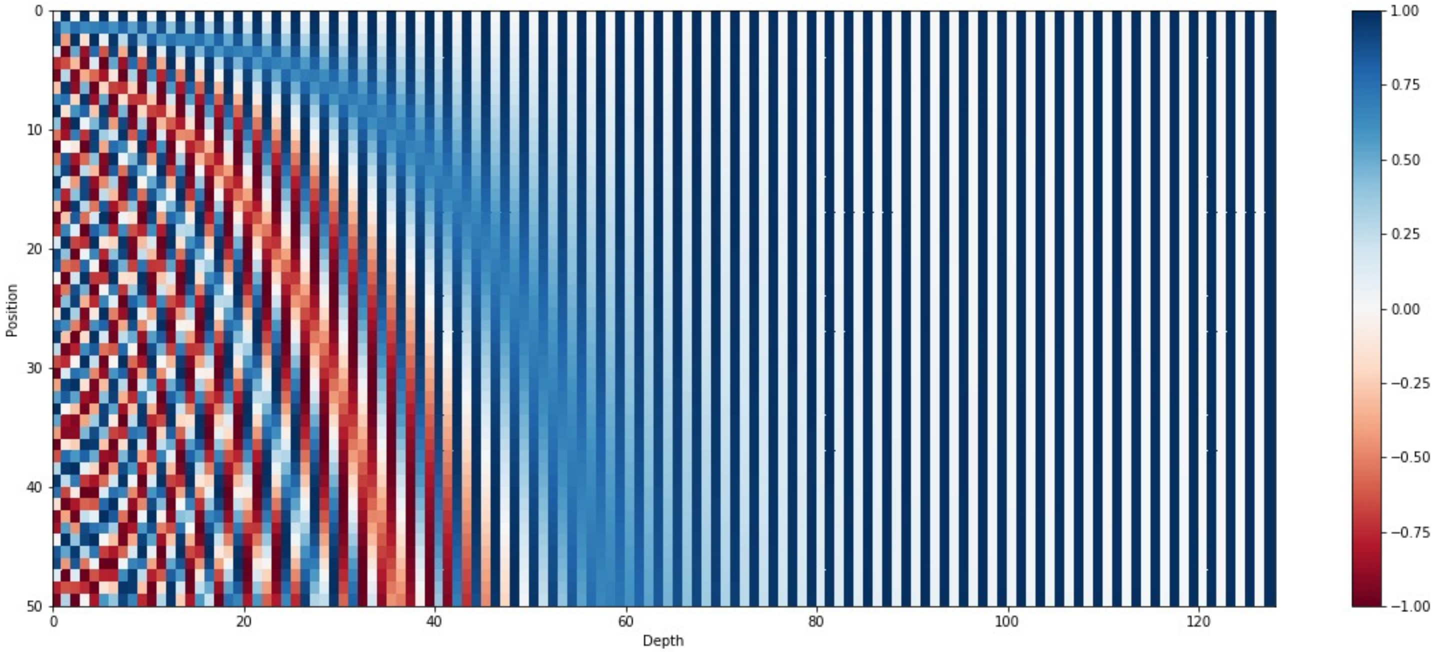
- **h** = 8 attention heads are used in Transformer.
- To maintain the same computation as single-head attention,

$$d_k = d_v = d_{\text{model}}/h = 64$$

Positional Embeddings

- Since Transformer has a constant path length (distance a signal has to travel between positions), the model can't tell what order the inputs are in.
  - To fix this, add positional encodings!
  - Sinosoid is used,
    - but learned positions work just as well.

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



Types of Attention

1. Encoder-decoder attention: Queries come from previous layer of the decoder, keys and values come from encoder.
2. Encoder self-attention layer: Each position can attend every other position in the previous layer of the encoder.
3. Decoder self-attention layer: Mask out all connections in the Softmax that cannot have been seen.
    1. This maintains the autoregressive property by preventing the model from looking at words it hasn't seen yet.

Encoder

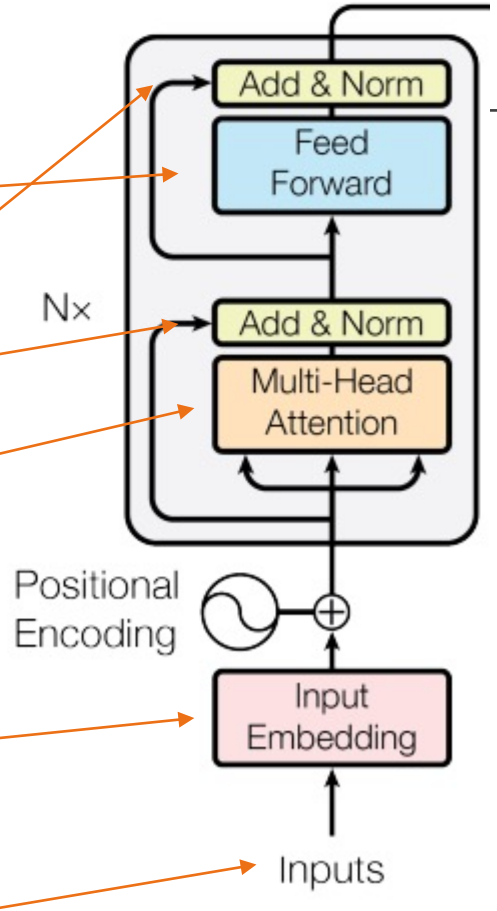
Simple ReLU feedforward network with 1 larger hidden layer

Recurrent connection by adding previous representation then layer normalization

Self-attention over all inputs

Learned embedding layer

Byte-pair or word-piece encoding



Decoder

Usual learned project and Softmax for token prediction

- Shares weights with embedding layer though!

Simple ReLU feedforward network with 1 larger hidden layer

Attention over all inputs

Recurrent connection by adding previous then layer normalization

Self-attention over all outputs (so far)

Learned embedding layer

Byte-pair or word-piece encoding

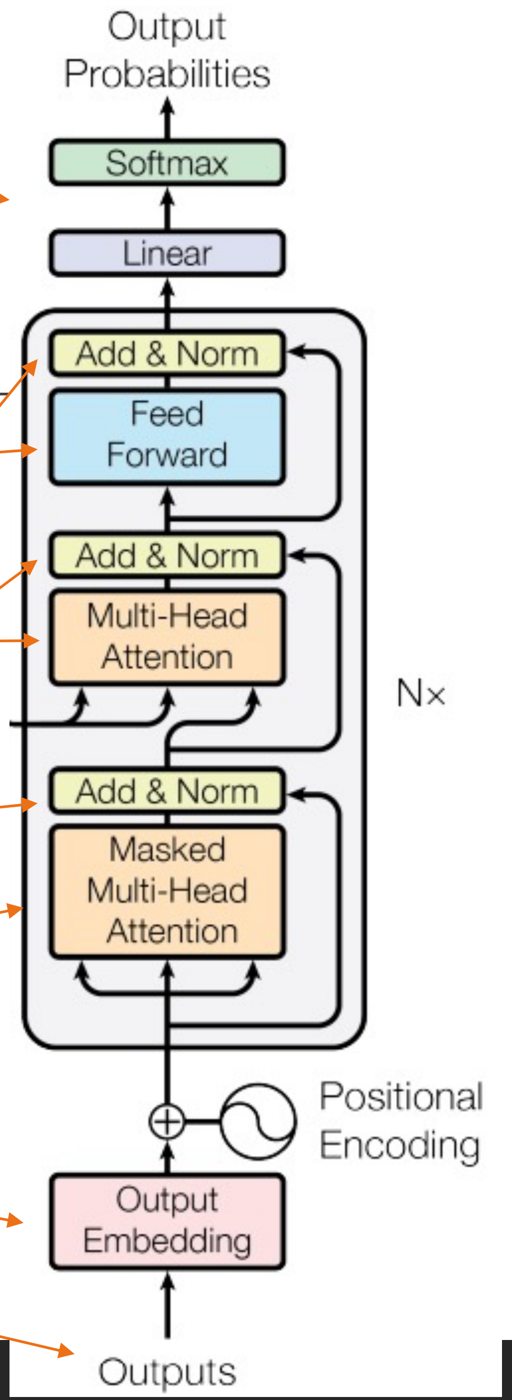


Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

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)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Advantages Over Other Approaches

Multi-head attention can also learn multiple things to look at

Training

- Sentences encoded:
  - English-German uses BytePair encoding for 37,000 tokens on 4.5M sentence pairs.
  - English-French uses WordPiece encoding for 32,000 tokens on 36M sentence pairs.
- Batch size determined in order to have 25,000 source and target tokens.
- 8 NVIDIA T100 GPUs.
  - Base models trained for 12 hours, big models for 3.5 days.
- Adam optimizer with special learning rate: $$lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$
  - Linear warmup followed by inverse square root decay.
- Regularization: Dropout of 0.1 applied to residual connections and sum of positional encoding and embeddings. Label smoothing is performed.
- The last 5 checkpoints are averaged (for base model). Beam search is used to select the best translation.

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 [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	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**	**41.0**	$2.3 \cdot 10^{19}$	

Results - BLEU

Model Variation Experiments

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

Attention Visualizations

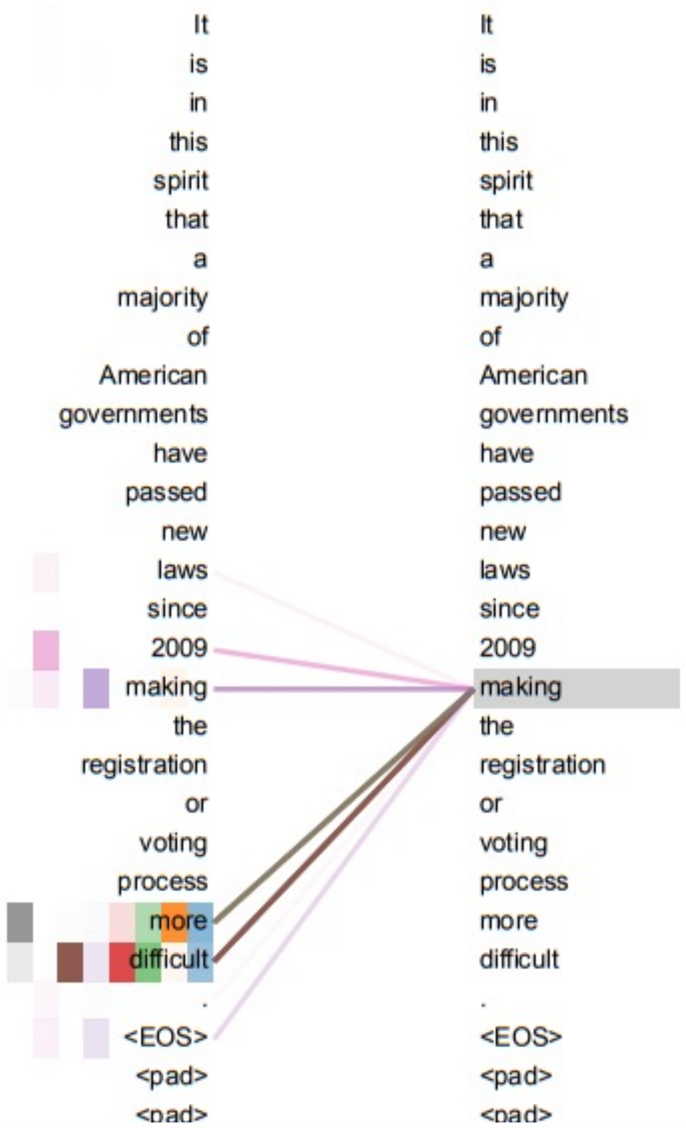


Figure 3: An example of the attention mechanism following long-distance dependencies in the encoder self-attention in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb ‘making’, completing the phrase ‘making...more difficult’. Attentions here shown only for the word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

Attention Visualizations

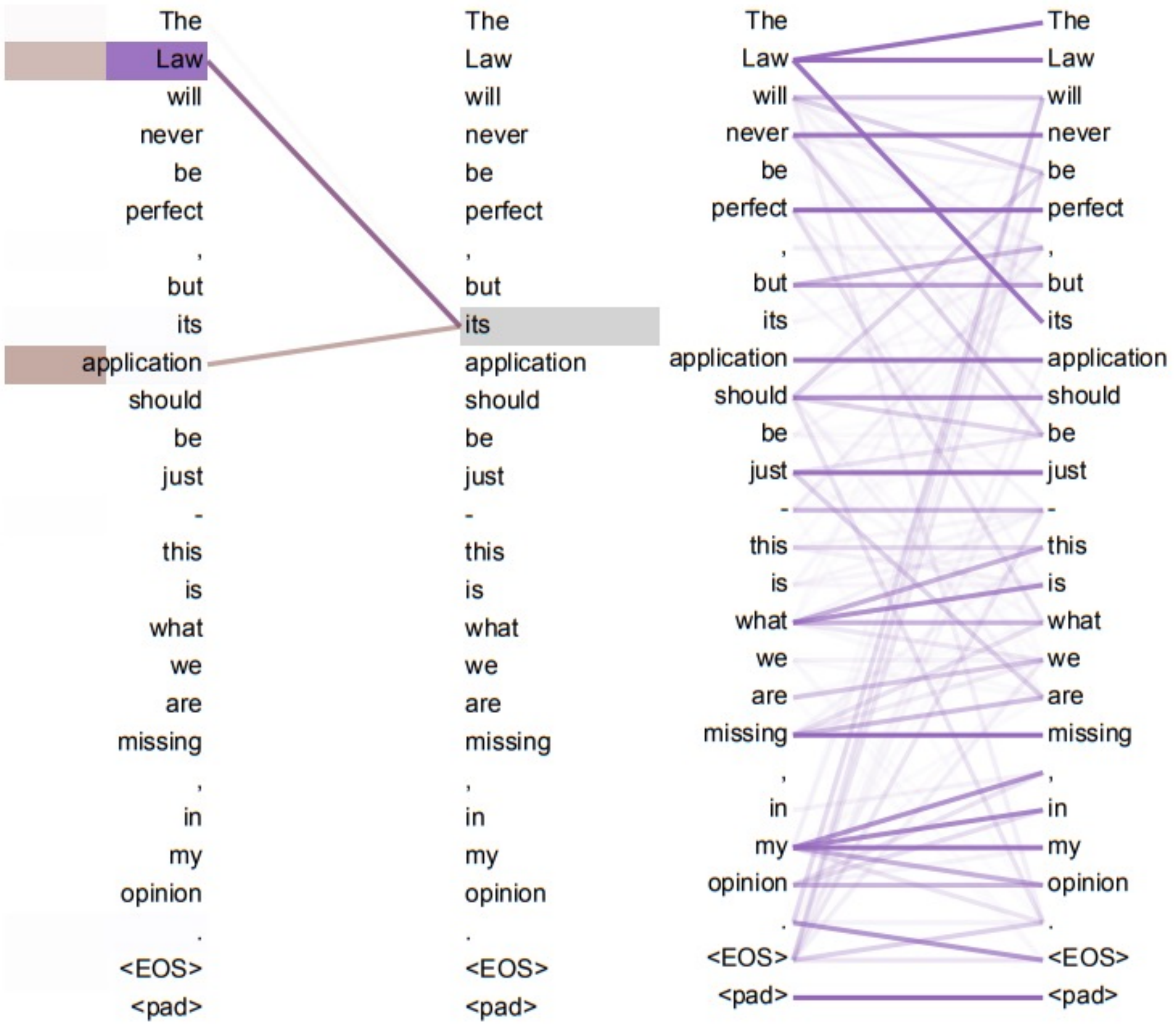


Figure 4: Two attention heads, also in layer 5 of 6, apparently involved in anaphora resolution. Top: Full attentions for head 5. Bottom: Isolated attentions from just the word 'its' for attention heads 5 and 6. Note that the attentions are very sharp for this word.

Attention Visualizations

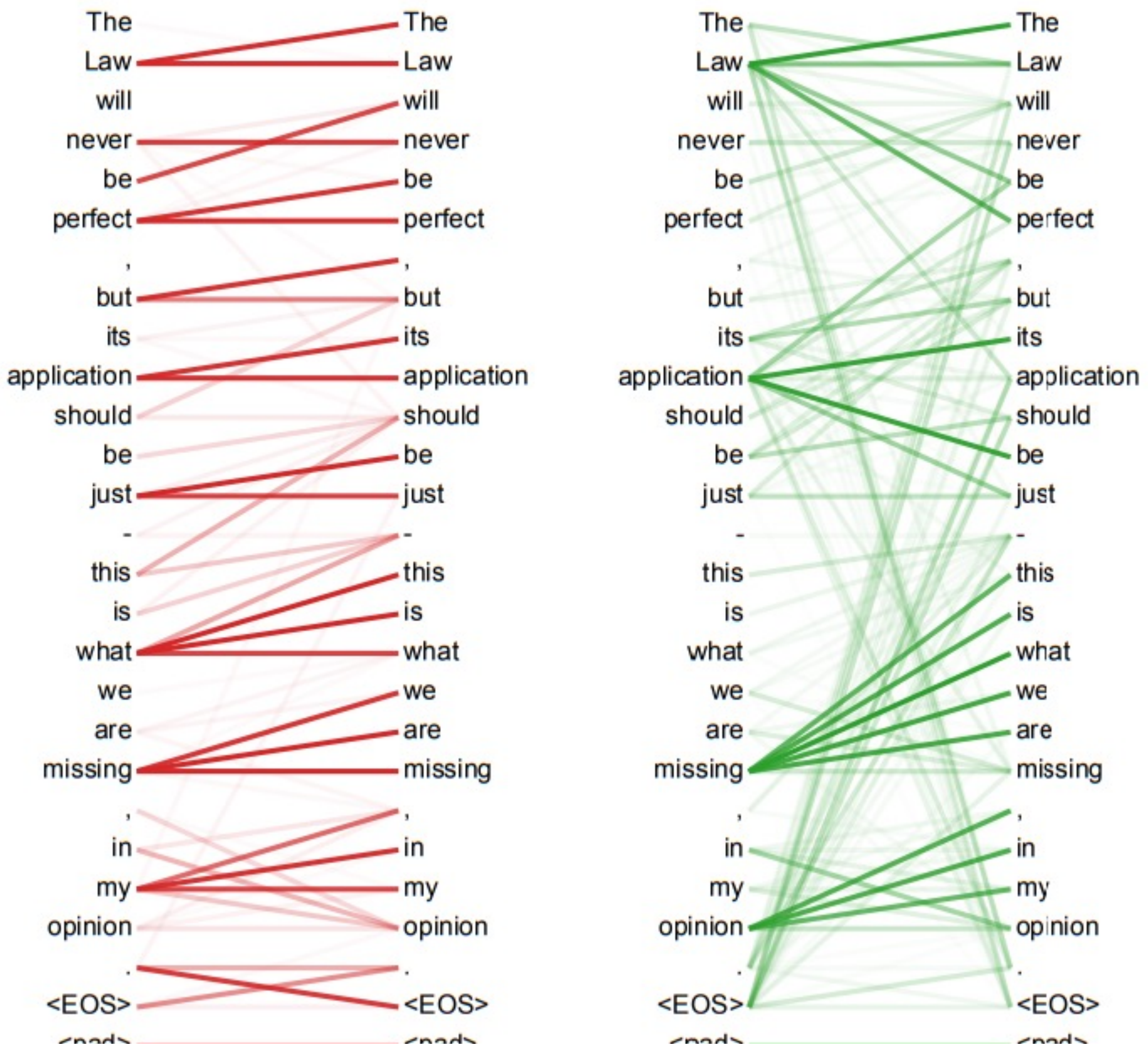


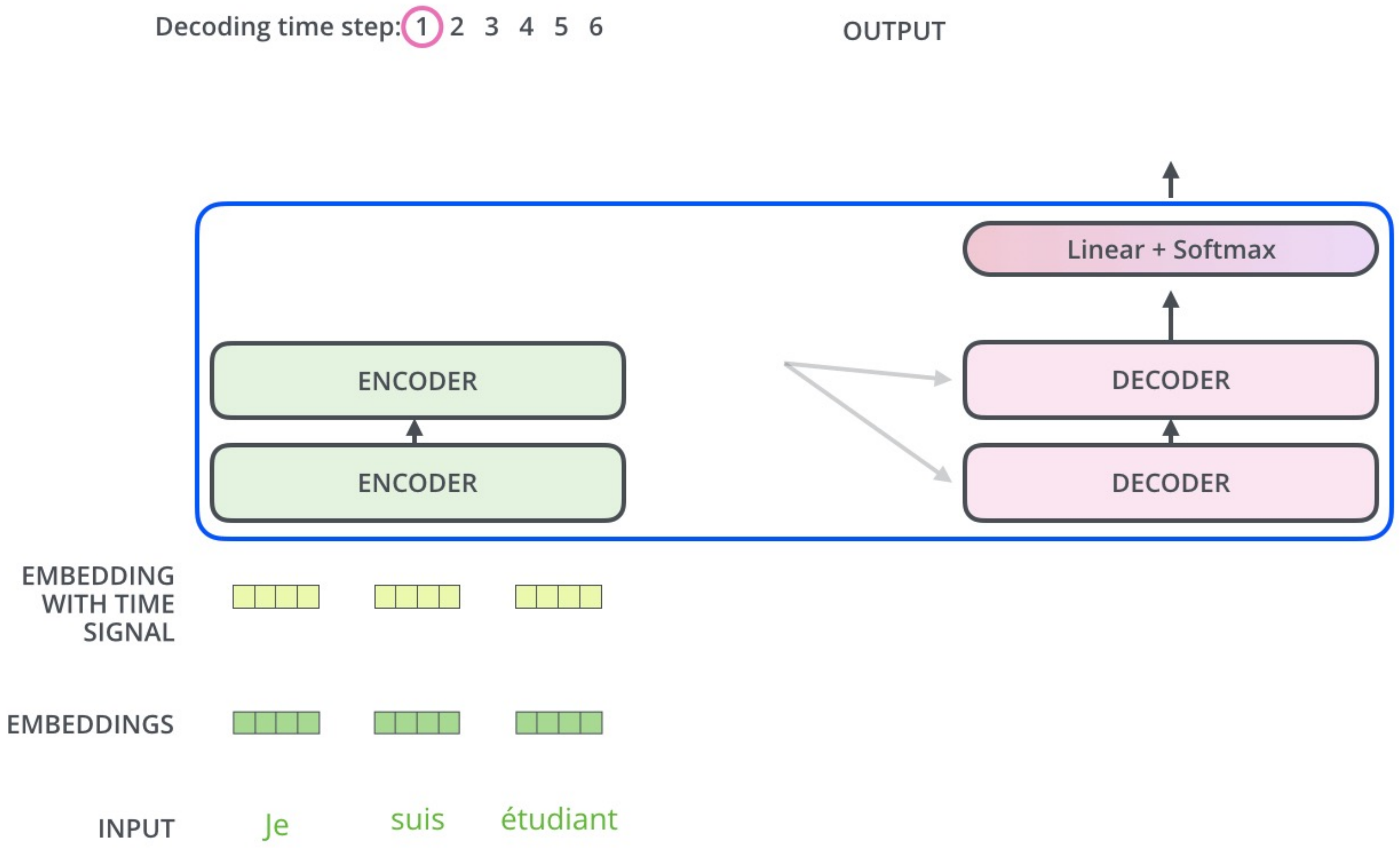
Figure 5: Many of the attention heads exhibit behaviour that seems related to the structure of the sentence. We give two such examples above, from two different heads from the encoder self-attention at layer 5 of 6. The heads clearly learned to perform different tasks.

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

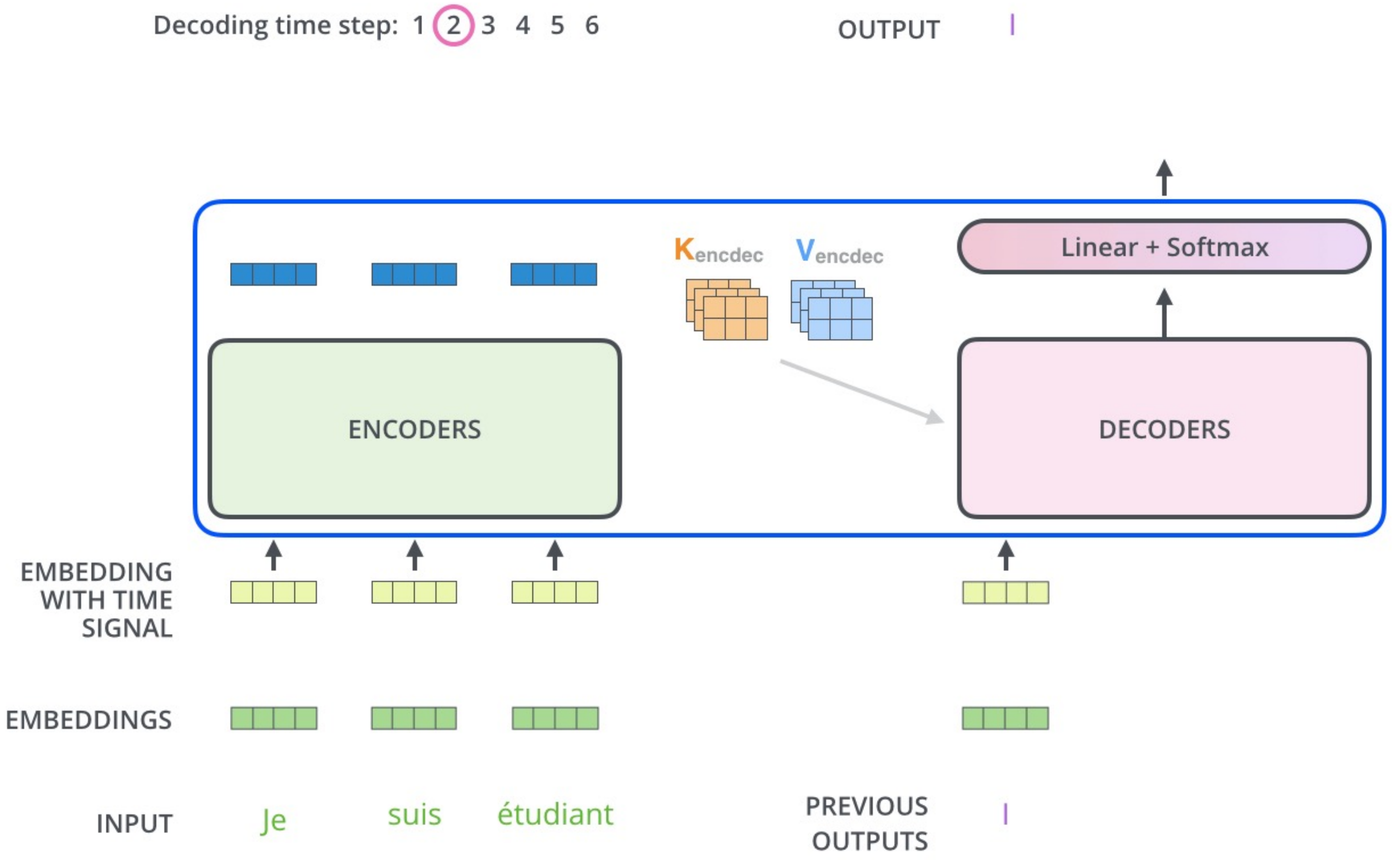
Parser	Training	WSJ 23 F1
Vinyals & Kaiser el al. (2014) [37]	WSJ only, discriminative	88.3
Petrov et al. (2006) [29]	WSJ only, discriminative	90.4
Zhu et al. (2013) [40]	WSJ only, discriminative	90.4
Dyer et al. (2016) [8]	WSJ only, discriminative	91.7
Transformer (4 layers)	WSJ only, discriminative	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

English Constituency Parsing

Example of Model



Example of Autoregressive Property



Takeaway

- Motivation: RNNs are not easily parallelizable and don’t learn long dependencies well.
- Models that only use attention are effective and train faster.
- Transformer generalizes to other tasks.
- Multi-Head attention helps address some of the problems of traditional attention.
- Transformers have a constant time path from one position to any other position.

References

- Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017.
- BLEU score slides: https://cloud.google.com/translate/automl/docs/evaluate
- https://kazemnejad.com/blog/transformer_architecture_positional_encoding/ for picture of positional encodings.
- I great blog on Transformers (that I can't beat): https://jalammar.github.io/illustrated-transformer/