Attention is all you need

Paper by Vaswani et al, 2017

Presented by Andreas Munk, Curtis Huebner, Martin Wang

Background

- In machine translation, conventional approach is to use a seq2seq encoder-decoder network.
- Sequential data modelled using RNNs or LSTMs, as seen in class.

However, this has limitations!

Difficult to take into account **long term dependencies**:

- Makes the model hard to parallelize. (inefficient)
- Efficiency and performance drops for longer sentences (sequences).

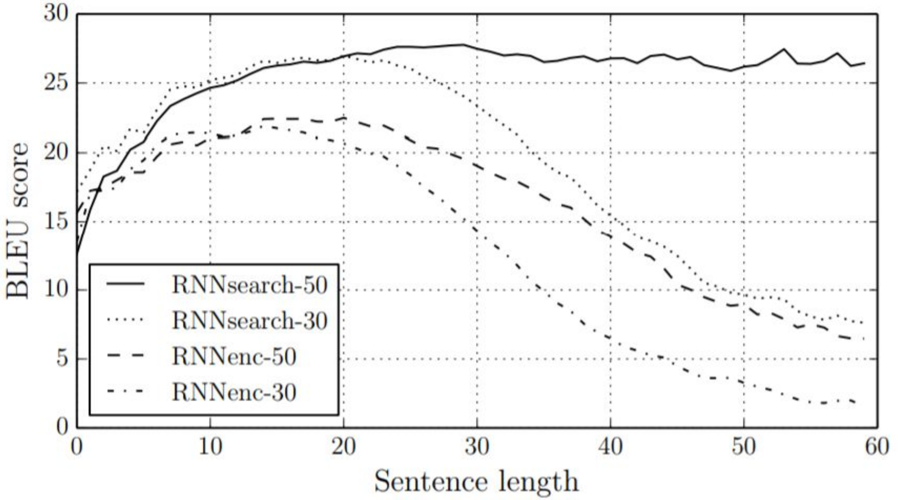


Fig 2, *Neural Machine Translation by Jointly learning to align and translate*, (Bahdanau et al), ICLR 2015

“Classic” attention - was shown in lecture

- **Neural Machine translation by Jointly learning to align and translate**

Attention intuition: Think of it as a weighted sum of inputs, where the weights are learnt through a simple neural network.

- When decoding, we take a weighted sum of all the encoder inputs so far, and pass it into the decoder hidden state.
- This lets us **selectively** use past state information, and helps utilize long term dependencies.

More on the “classic” attention approach

Classic Encoder-Decoder:

$$h_t = f(x_t, h_{t-1})$$

$$c = q(\{h_1, \cdots, h_{T_x}\}),$$

$$p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c),$$

Classic attention:

$$p(y_i | y_1, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$

Notice, we have c_i now:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j. \qquad \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$h_j = \left[\overrightarrow{h}_j^\top ; \overleftarrow{h}_j^\top\right]^\top.$$

Equations from *Neural Machine Translation by Jointly learning to align and translate*, (Bahdanau et al), ICLR 2015

“Classic” attention - diagram from lecture 9 slides

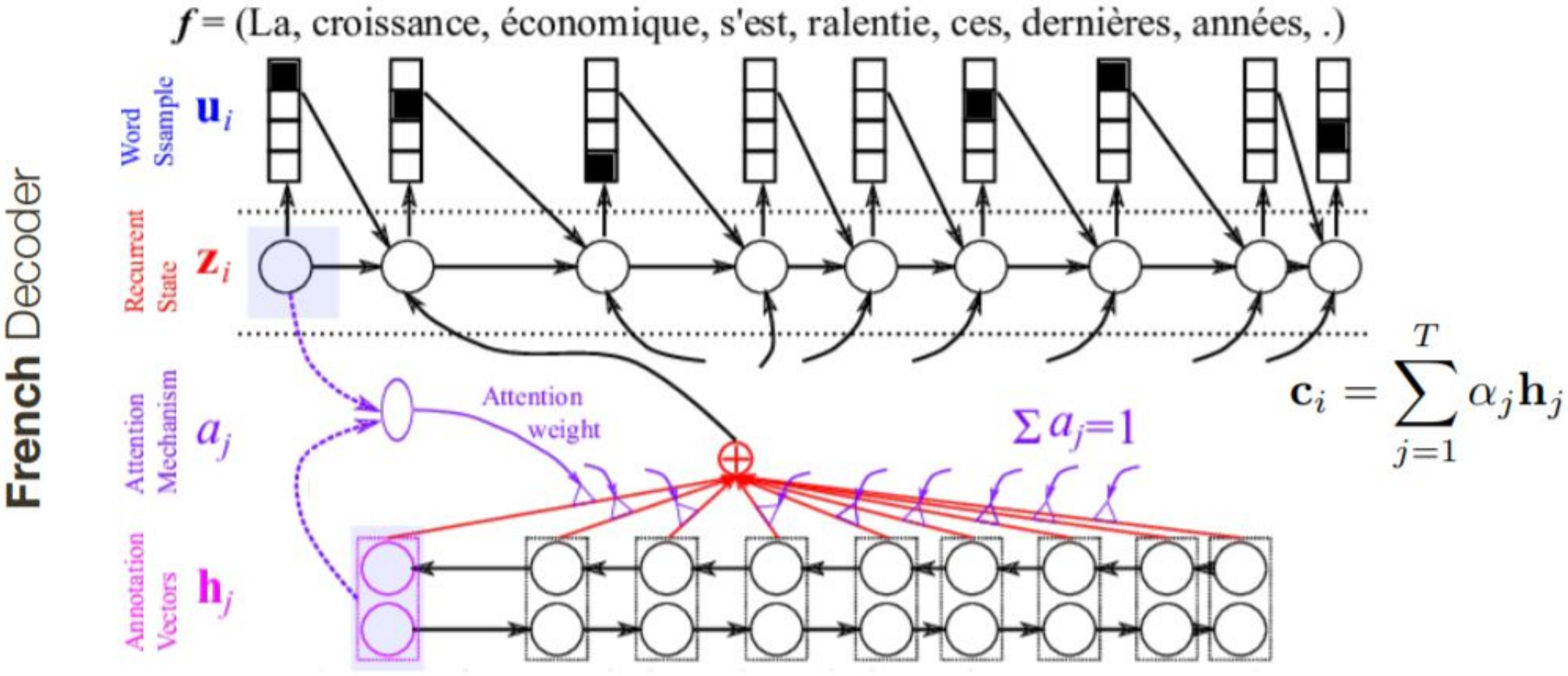


Diagram from lecture 9 slides CPSC 532S, originally from
https://devblogs.nvidia.com/introduction-neural-machine-translation-gpus-part-3/

Previous work, non-attention based

- Previous work has tried to address the challenges mentioned previously regarding long-term dependencies.
- Bytenet (Kalchbrenner et al, 2017)
- ConvS2S (Gehring et al, 2017)

Both of these models use CNN’s for encoding and decoding, eliminating the need for recurrence (RNN, LSTM)

- Enables parallelization
- Intuitively, similar to attention
- Still hard to take into account long term dependencies

Transformer Intuition

- In classic attention, during the decoding process, we weight all the encoder hidden states. Can this be extended? Turns out it can.
- **We can eliminate recurrence altogether.**
- In seq2seq, we “unrolled” a recurrent network. When we started to decode, the “last hidden state of the encoder” included information about the long-term dependencies in the sequence. (this was passed through the recurrent encoder)
- Now, instead of using recurrent hidden states, we use attention. The crucial difference is that each output prediction word is its own “prediction problem”

More intuition

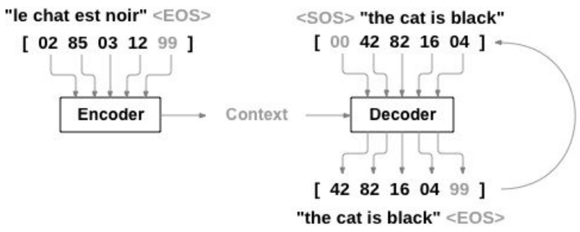
- Didn't the old network also use attention **alongside** recurrence?
- Yes - but this paper introduces a more sophisticated attention mechanism - multi-headed attention: the idea that we have multiple passes of attention, and then combine them.

Andreas will now talk about these in detail

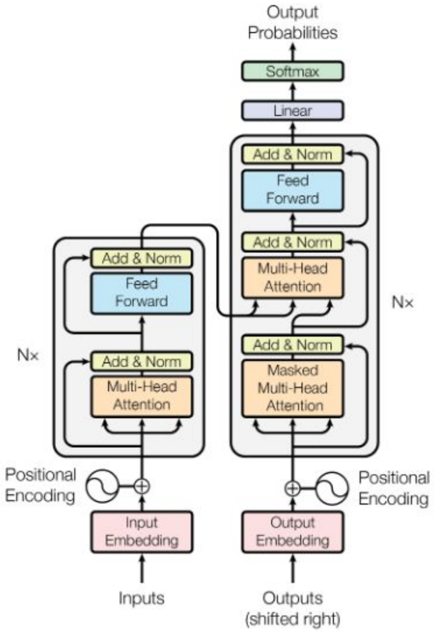
Architecture - The Transformer

- Similar to the encoder-decoder network (SUTSKEVER, Ilya; VINYALS, Oriol; LE, Quoc V. Sequence to sequence learning with neural networks. In: *Advances in neural information processing systems*. 2014. p. 3104-3112)
- Encoder-decoder is a mapping:

$$(x_1, \ldots, x_n) \rightarrow (z_1, \ldots, z_n) \rightarrow (y_1, \ldots, y_m)$$

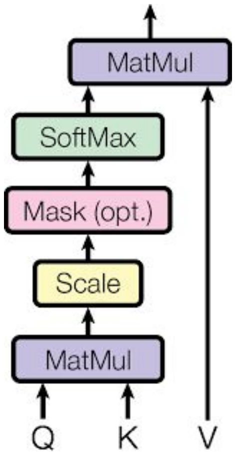


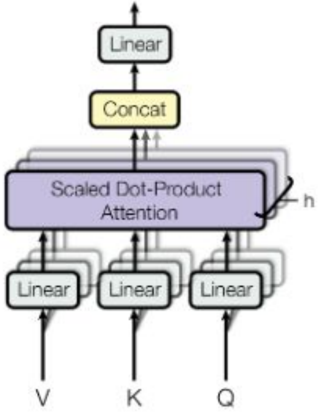
https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html



The transformer (Vaswani, Ashish, et al. "Attention is all you need." *Advances in Neural Information Processing Systems*. 2017.)

Scaled Dot-Product Attention





Vaswani, Ashish, et al. "Attention is all you need." *Advances in Neural Information Processing Systems*. 2017.

$$Q \in \mathbb{R}^{q \times k} \; K \in \mathbb{R}^{k \times l} \; V \in \mathbb{R}^{l \times v}$$

We choose the dimensions q, k, and v. l is the number of elements to attend to.

Dot product attention

$$Attention(Q, K, V) = softmax\left(\frac{QK}{\sqrt{k}}\right)V$$

- Weighted sum over elements to attend to
- Scaling to counteract saturating the softmax function, leading to small gradients.

Multi-Head Attention

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O$$

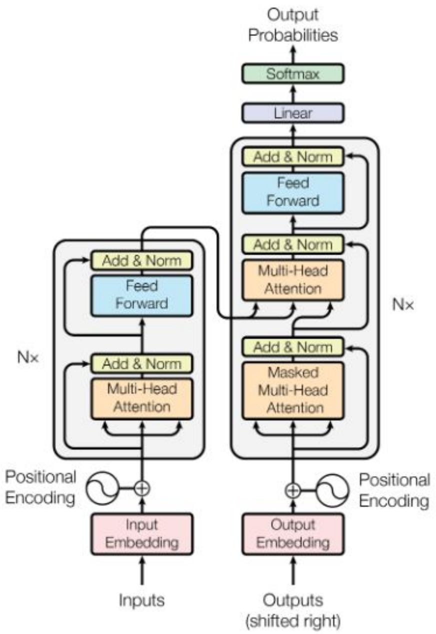
$$head_i = Attention(QW_i^Q, W_i^K K, VW_i^V)$$

- Projection into smaller dimensions instead of the $d_{model} = 512$ dimensional output from previous layers
- We can consider these as intermediate embeddings
- They choose $h = 8$ parallel attention layers
- $k = v = d_{model}/h = 64$

Positional encoding

- Sum the input/output encoding and positional encoding
- $PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$

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



The transformer (Vaswani, Ashish, et al. "Attention is all you need." *Advances in Neural Information Processing Systems*. 2017.)

Why self-attention

n is sequence length, d is representation dimension, k convolution kernel size, and

r is size of neighborhood in restricted attention

Layers Type	Complexity per Layer	Sequential Operations
Self-Attention	$O(n^2 d)$	$O(1)$
Recurrent	$O(nd^2)$	$O(n)$
Convolutional	$O(knd^2)$	$O(1)$
Self-Attention (restricted)	$O(rnd)$	$O(1)$

Why self-attention

- Computational complexity (when sequence length is smaller than representation dimension)
  - Restricted self-attention
- Parallelization
- Long term dependencies
- Interpretability

Experiments

- The model was tested on the WMT english to german and english to french translation tasks.
  - English to German consists of 4.5M sentence pairs
  - English to French consists of 36M sentence pairs
- The model was also tested on english constituency parsing.
  - 40K sequences from the WSJ portion of the Penn Treebank dataset.

WMT EtoG and EtoF

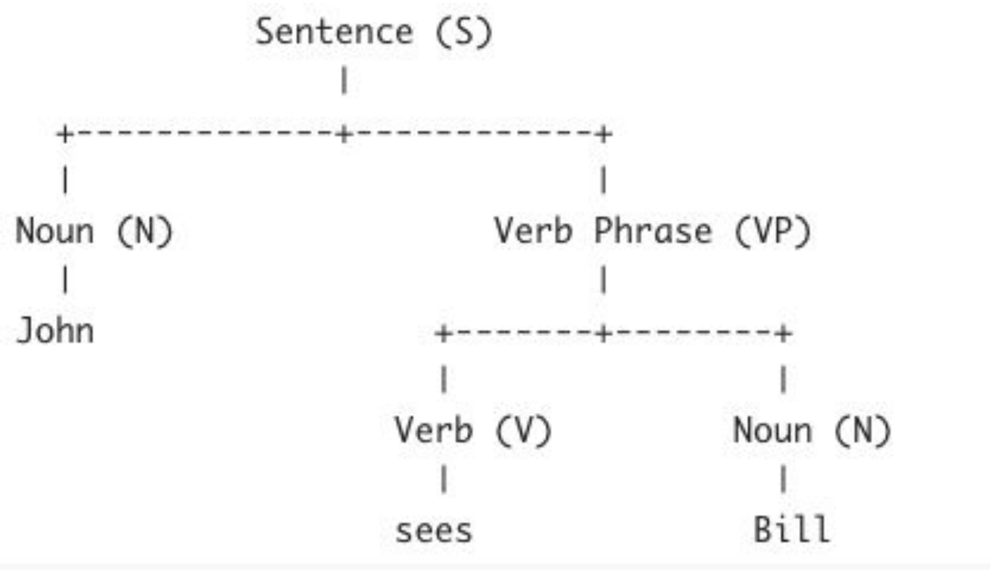
- BLEU score used as a metric.
- Sentence to sentence translation.
- Uses label smoothing to get better BLEU scores at the cost of perplexity

Sample sentences: “I declare resumed the session of the European Parliament adjourned on Friday 17 December 1999, and I would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period.”

Target Translation: “Je déclare reprise la session du Parlement européen qui avait été interrompue le vendredi 17 décembre dernier et je vous renouvelle tous mes vux en espérant que vous avez passé de bonnes vacances.”

English Constituency Parsing

- Was tested to see how the architecture performs on other domains
- F1 score used to measure performance (EVALB)
- Unclear from the paper how the parse tree is generated



Ablations/Model Variations

- Evaluated on WMT EtoG
- Paper tries out various model sizes to characterize performance with different model parameters
- In general, they find that “bigger is better” but that there is an optimal # of read heads.

	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

Vaswani et al. 2017

Results: WMT

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**	$2.3 \cdot 10^{19}$	

Vaswani et al. 2017

Results: Constituency Parsing

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

Vaswani et al. 2017

References

- Attention is all you need, (Vaswani et al.) , 2017.
- Neural Machine Translation by Jointly Learning to Align and Translate, (Bahdanau et al) ICLR 2015
- Neural Machine Translation in Linear Time, (Kalchbrenner et al), 2017.
- Convolutional sequence to sequence learning (Gehring et al), 2017.
- SUTSKEVER, Ilya; VINYALS, Oriol; LE, Quoc V. Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. 2014. p. 3104-3112