



Transformers I

CS 4391 Introduction Computer Vision

Professor Yu Xiang

The University of Texas at Dallas

Machine Translation

- Translate a phrase from one language to another
 - E.g., English phrase to French phrase

Google Translation

The screenshot shows the Google Translate interface. At the top, there are two dropdown menus for language selection: 'English' on the left and 'French' on the right, separated by a double-headed arrow. Below the 'English' menu is a text input field containing the sentence: 'UT Dallas is a rising public research university in the heart of DFW.' To the right of this text is a small 'x' icon. Below the input field, it says '13 words'. To the right of the input field is a light blue shaded area containing the translated text: 'UT Dallas est une université de recherche publique en plein essor au cœur de DFW.' Below this translated text, it says '15 words'.

English

French

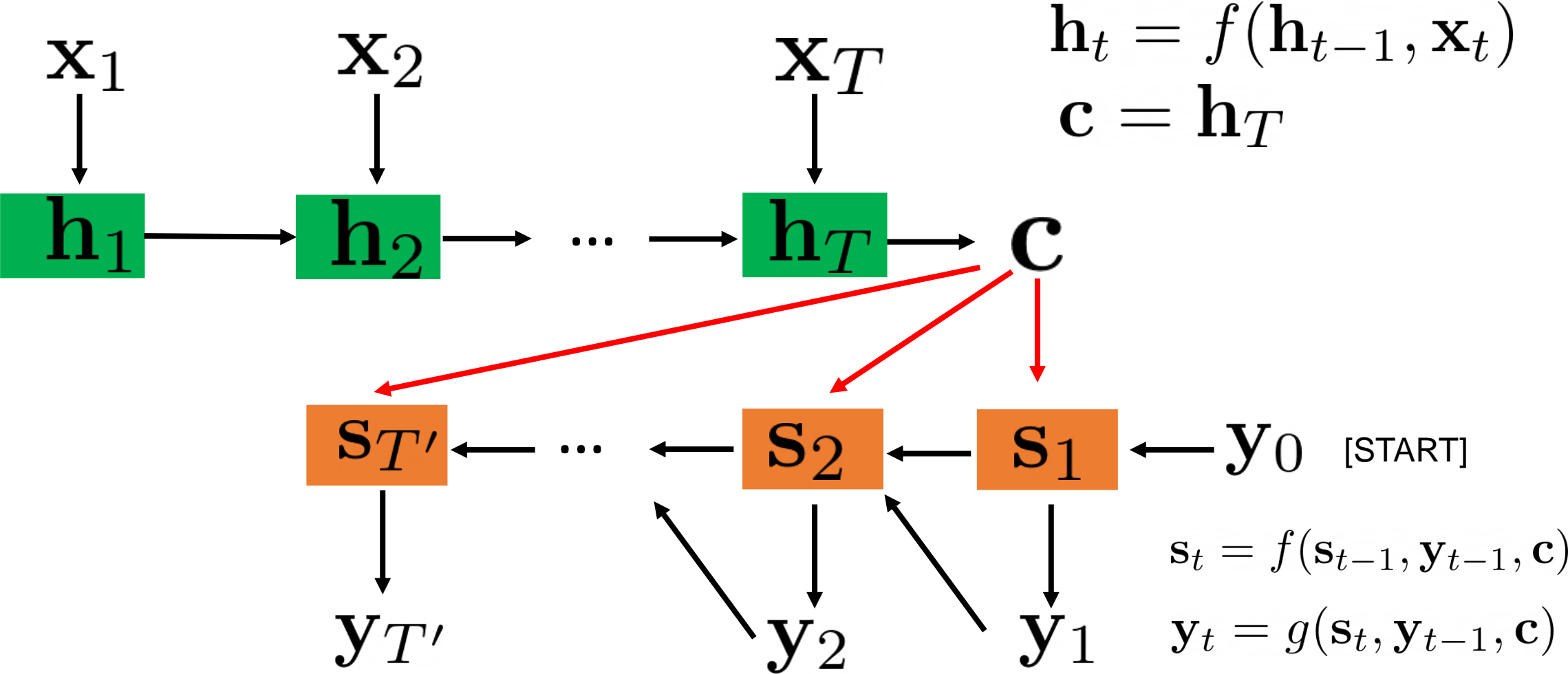
UT Dallas is a rising public research university in the heart of DFW.

13 words

UT Dallas est une université de recherche publique en plein essor au cœur de DFW.

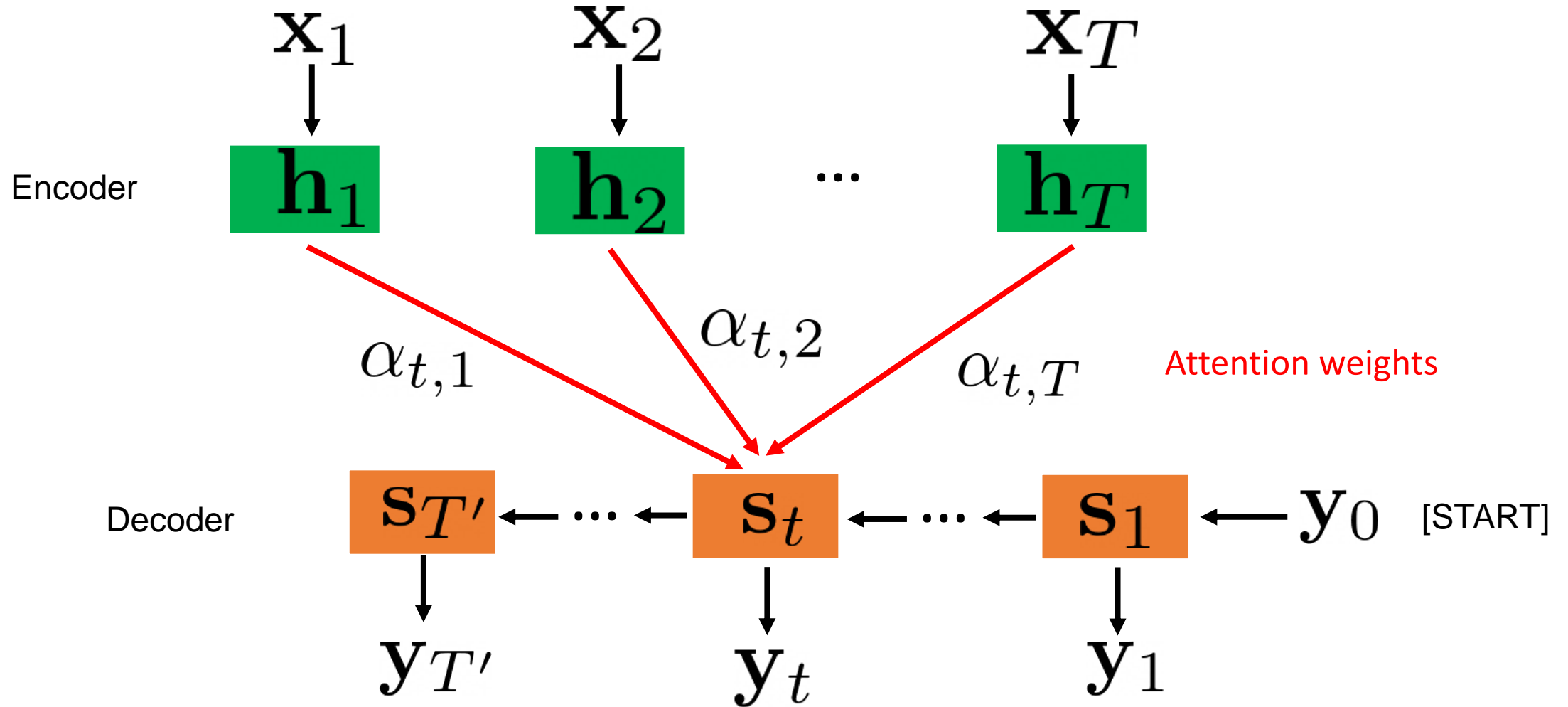
15 words

RNN Encoder-Decoder



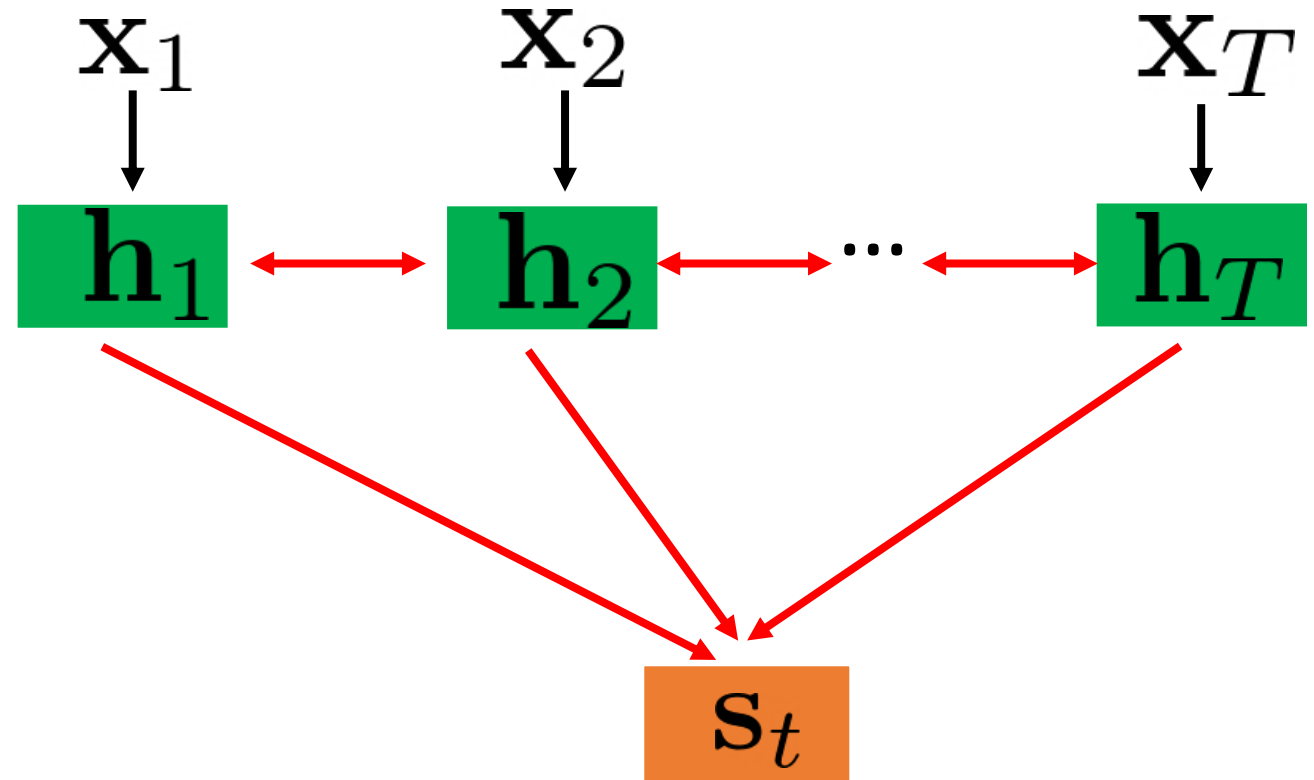
Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Cho et al., EMNLP'14

Transformer: Encoder-Decoder with Attention



Transformer: Attention

- Two types of attentions
 - Self-attention
 - Cross-attention



Transformer: Attention

- Input
 - (key, value) pairs (think about python dictionary)
 - A query
- Output
 - Compare the query to all the keys to compute weights
 - Weighted sum of the values

Attention is all you need. Vaswani et al., NeurIPS'17

Transformer: Attention

- Scaled Dot-Product Attention
 - Keys $K : m \times d_k$
 - Values $V : m \times d_v$
 - n queries $Q : n \times d_k$

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

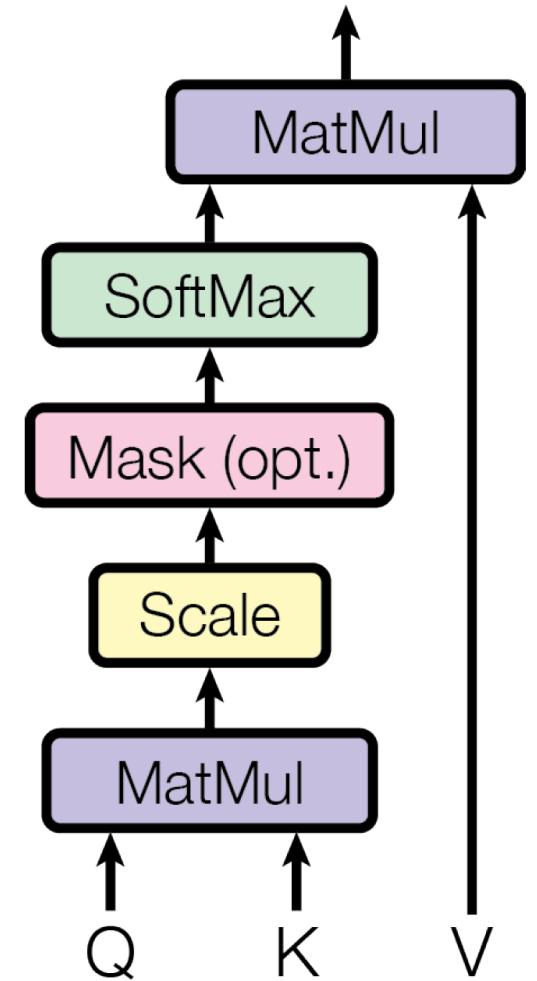
Softmax function

$$\sigma(\mathbf{y})_i = \frac{e^{y_i}}{\sum_j^m e^{y_j}}$$

$$n \times d_v$$

weights

Attention is all you need. Vaswani et al., NeurIPS'17



Transformer: Attention

- Multi-Head Attention

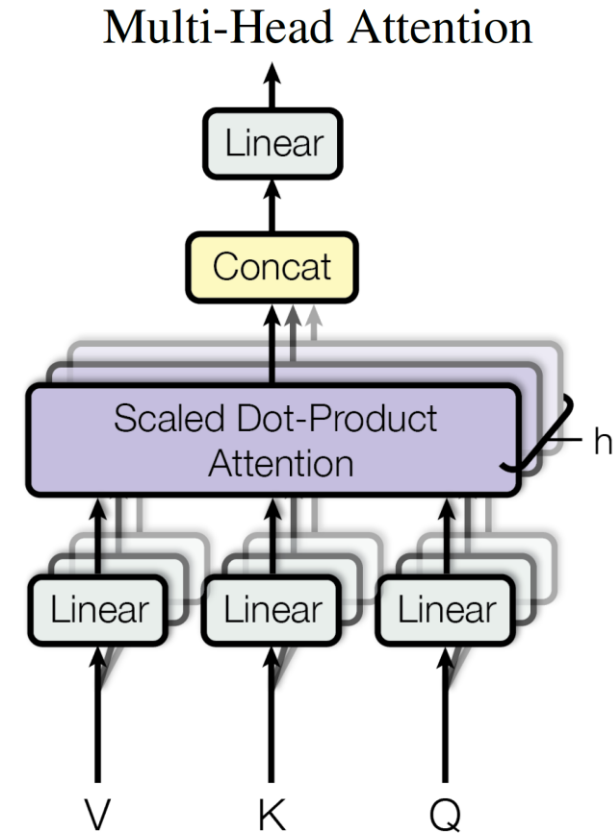
- Suppose the latent vector is with dimension d_{model}

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

$n \times d_v$ $m \times d_{\text{model}}$ $d_{\text{model}} \times d_k$ $n \times d_{\text{model}}$ $d_{\text{model}} \times d_k$ $m \times d_{\text{model}}$ $d_{\text{model}} \times d_v$

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

$n \times d_{\text{model}}$ $n \times hd_v$ $hd_v \times d_{\text{model}}$



Attention is all you need. Vaswani et al., NeurIPS'17

Transformer: Encoder

- Self-attention (repeat N times)
 - Keys, values and queries are all the same
 - n input tokens $n \times d_{\text{model}}$

MultiHead(Q, K, V)

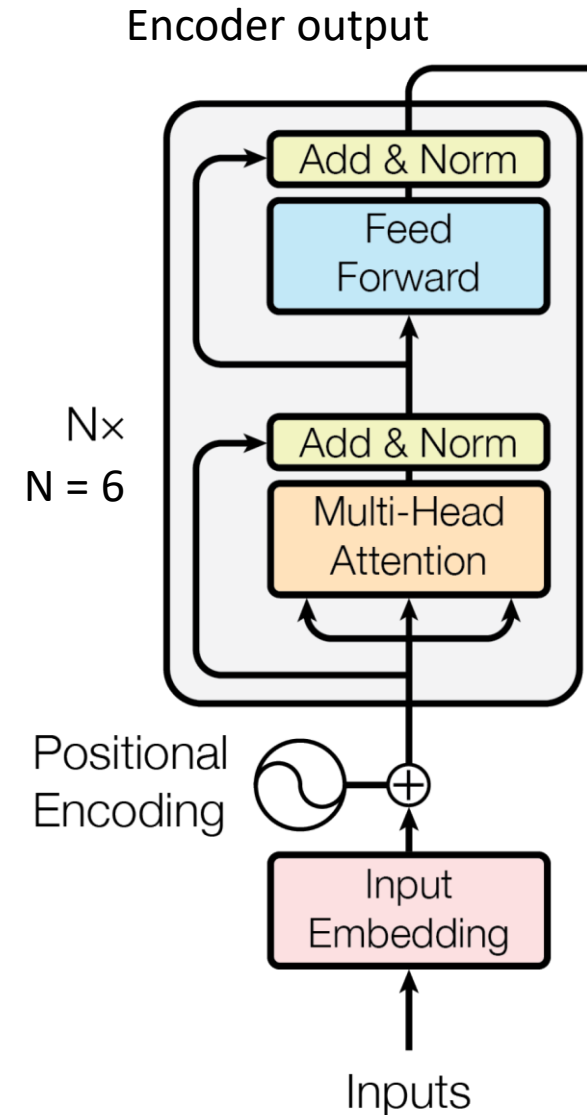
- Residual connection

LayerNorm($x + \text{Sublayer}(x)$)

- Layer normalization

$$\mu^l = \frac{1}{H} \sum_{i=1}^H a_i^l \quad \sigma^l = \sqrt{\frac{1}{H} \sum_{i=1}^H (a_i^l - \mu^l)^2} \quad \frac{a^l - \mu^l}{\sigma^l}$$

Attention is all you need. Vaswani et al., NeurIPS'17



Transformer: Encoder

- Feed Forward Network

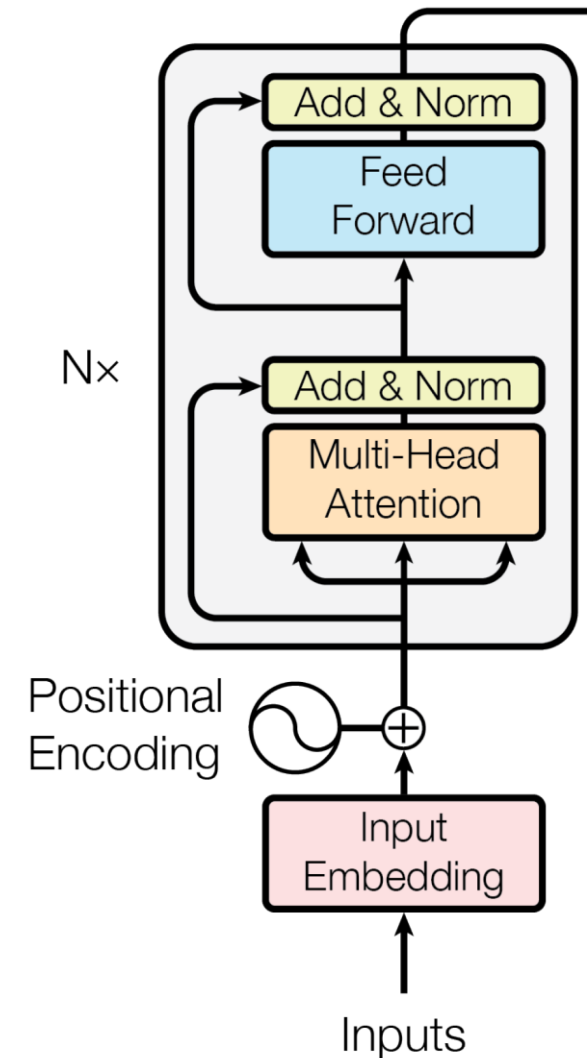
$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

- Positional encoding

- Make use the order of the sequence
- With dimension d_{model} for each input

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

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



Attention is all you need. Vaswani et al., NeurIPS'17

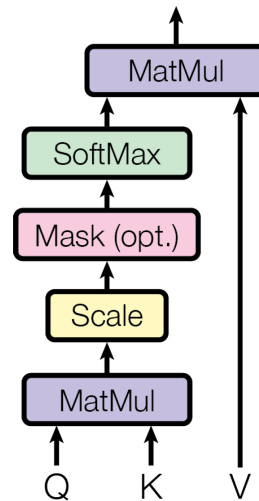
Transformer: Decoder

- Output embedding

[START]

$y_0 \ y_1 \ \dots \ y_{t-1} \ y_t \ y_{t+1} \ \dots \ y_{T'}$

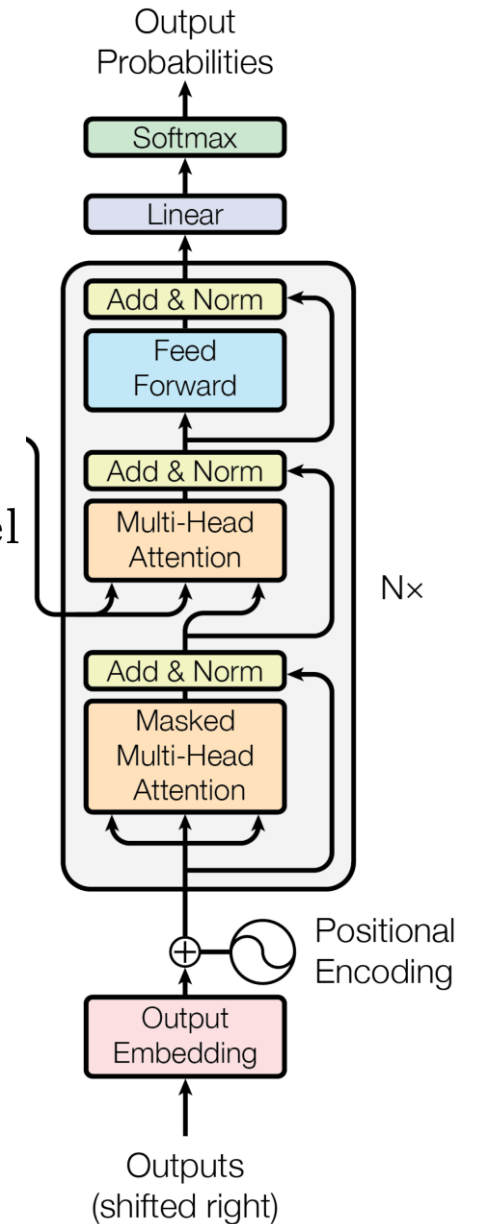
Shifted right by one position and insert the start token



Mask out current and future outputs during training (setting to $-\infty$)

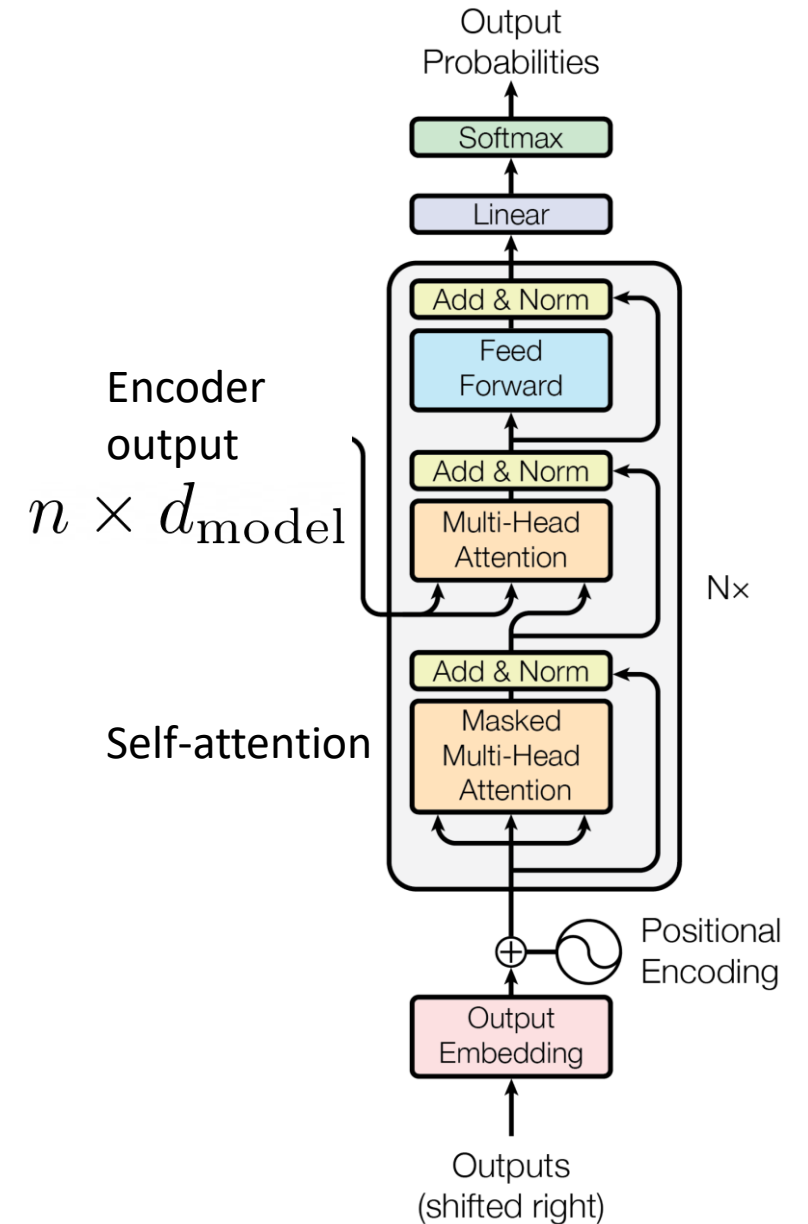
Attention is all you need. Vaswani et al., NeurIPS'17

Encoder output
 $n \times d_{\text{model}}$



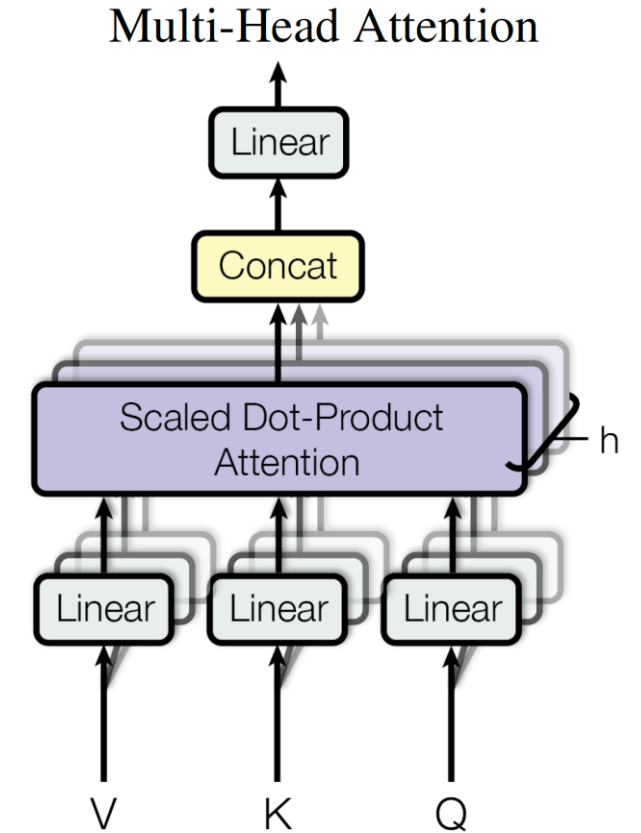
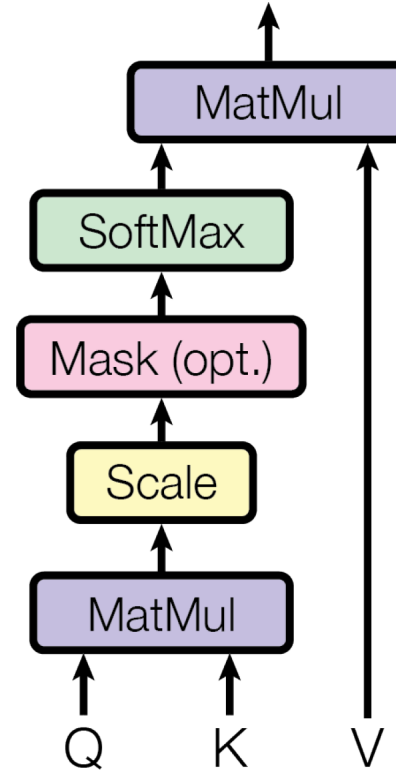
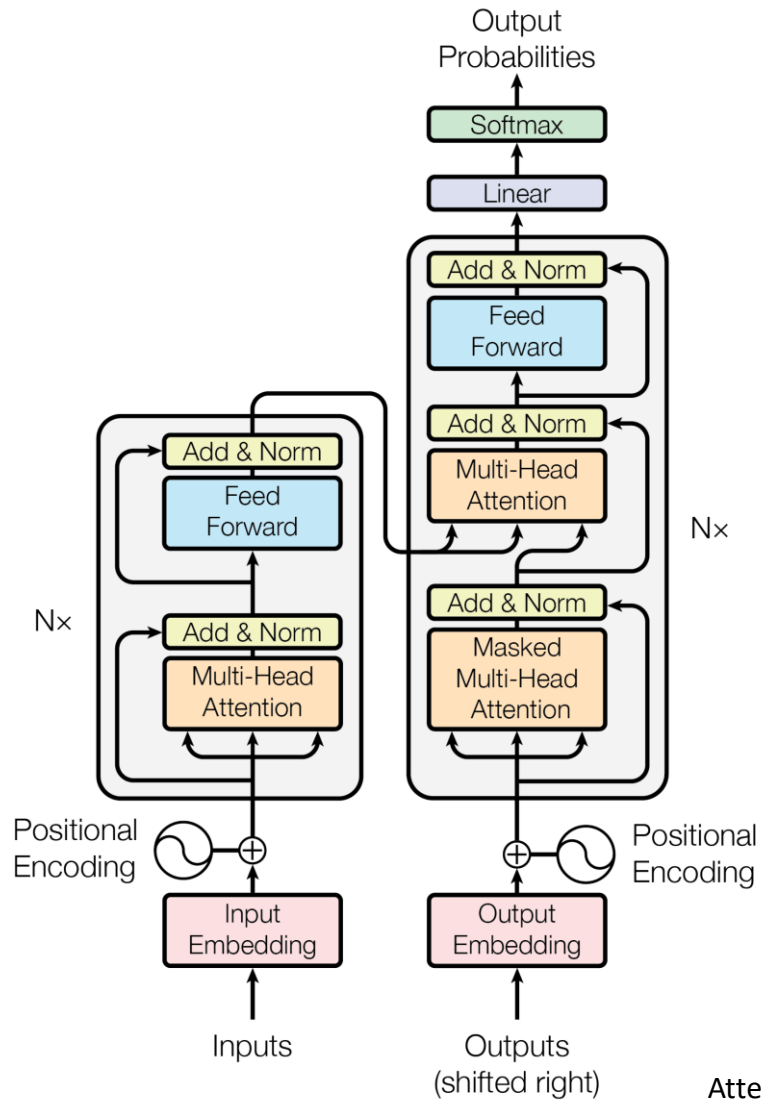
Transformer: Decoder

- Encoder-decoder attention
 - (Key, value): encoder output
 - Queries: decoder output
 - Every position in the decoder attends to all positions in the input sequence
- Softmax
 - Predicts next-token probabilities



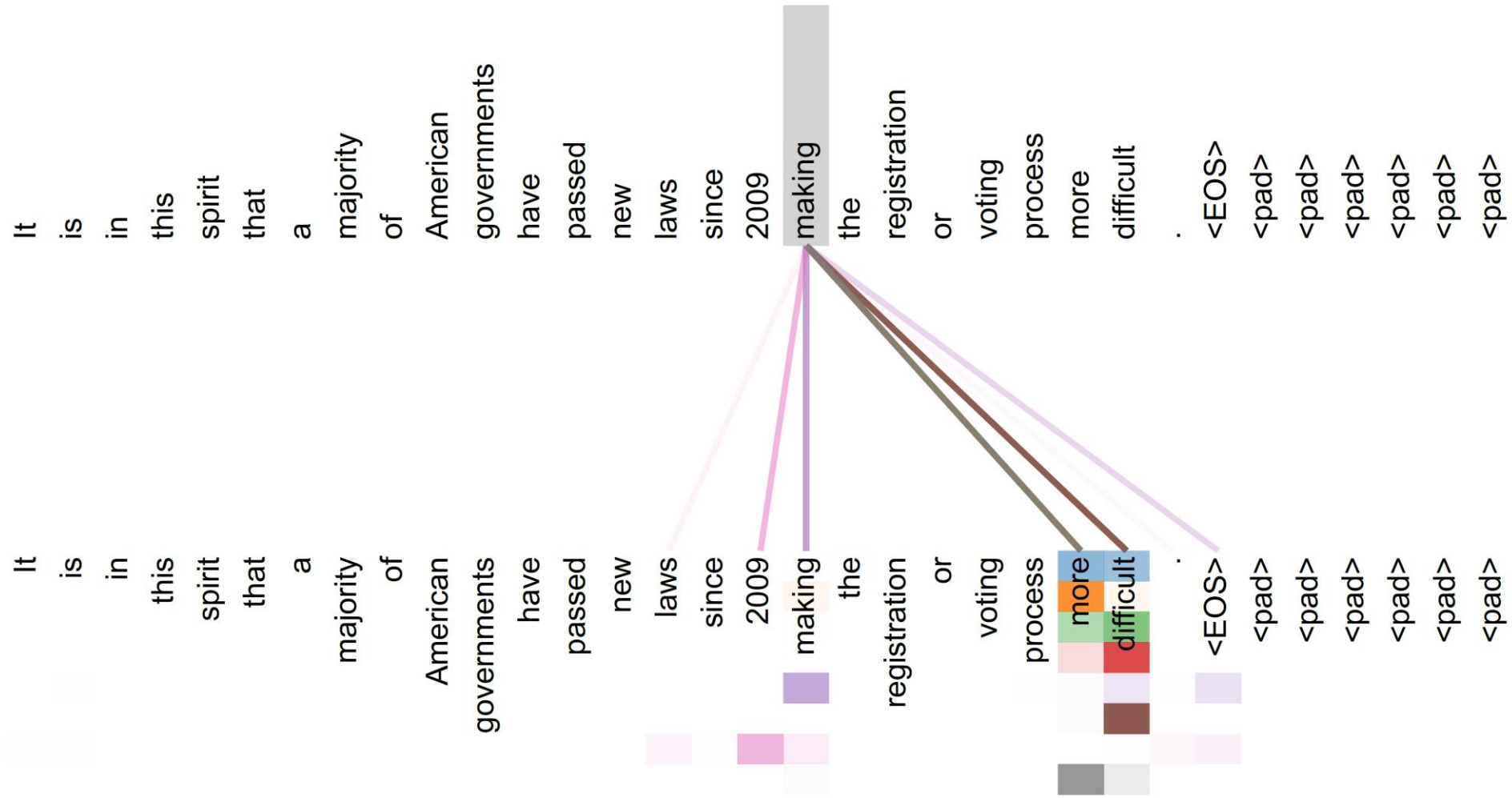
Attention is all you need. Vaswani et al., NeurIPS'17

Transformer



Attention is all you need. Vaswani et al., NeurIPS'17

Transformer: Attention Visualization



Attention is all you need. Vaswani et al., NeurIPS'17

Summary

- Transformers
 - Can capture long-distance dependencies (global attention)
 - Computationally efficient, more parallelizable

Further Reading

- Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation <https://arxiv.org/abs/1406.1078>
- Neural Machine Translation by Jointly Learning to Align and Translate <https://arxiv.org/abs/1409.0473>
- Transformer: Attention is all you need <https://arxiv.org/abs/1706.03762>