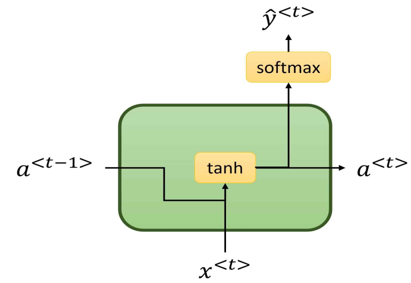
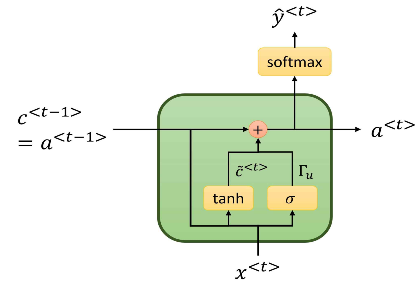
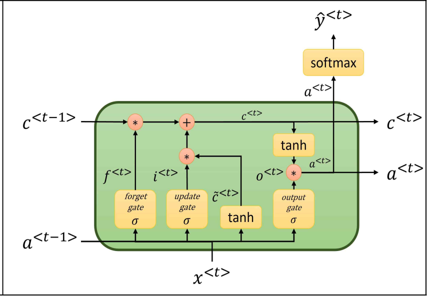
Transformer Network

Transformer Intuition

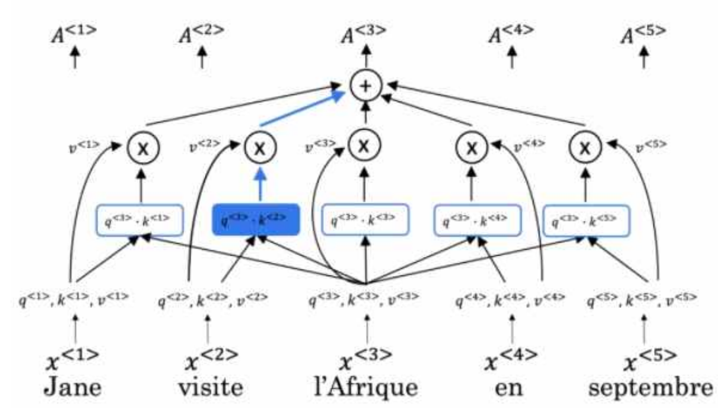
Increased complexity, sequential		
RNN unit	GRU unit	LSTM unit
		

A Transformer Network, like its predecessors RNNs, GRUs, LSTMs, can process information one word at a time.(Sequential architecture
- Attention + CNN - Self-Attention - Multi-Head Attention
- Attention mechanism. - Convolutional Neural Network style of processing.

Self-Attention

$A(q, K, V)$ = attention-based vector representation of a word → calculate for each word $A^{<1>}, A^{<2>}, \cdots, A^{<s>}$	
RNN Attention $$a^{<t,t'>} = \frac{\exp(e^{<t,t'>})}{\sum_{t'=1}^{T_x} \exp(e^{<t,t'>})}$$	Transformers Attention $$A(q, K, V) = \sum_i \frac{\exp(q \cdot k^{<i>})}{\sum_j \exp(q \cdot k^{<j>})} v^{<i>}$$

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



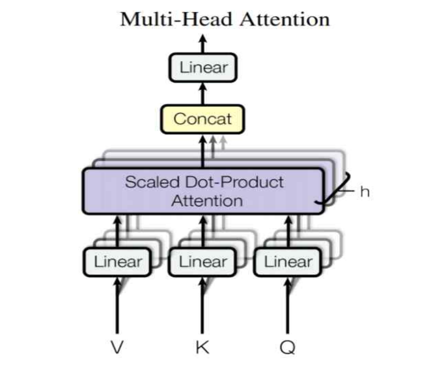
Query(Q)	Key(K)	Value(V)
$q^{<1>}$	$k^{<1>}$	$v^{<1>}$
$q^{<2>}$	$k^{<2>}$	$v^{<2>}$
$q^{<3>}$	$k^{<3>}$	$v^{<3>}$
$q^{<4>}$	$k^{<4>}$	$v^{<4>}$
$q^{<5>}$	$k^{<5>}$	$v^{<5>}$

Given a word, its neighboring words are used to compute its context by summing the word values to map the Attention related to that given word.

Q = interesting questions about the words in a sentence
K = qualities of words given a Q
V = specific representations of words given a Q

$q^{<t>} = W^{Q}x^{<t>}$	$q^{<t>} = W^{K}x^{<t>}$	$q^{<t>} = W^{V}x^{<t>}$

Multi-Head Attention



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

$$MultiHead(Q, K, V) = concat(head_1, head_2, \cdots, head_h)W_0,$$

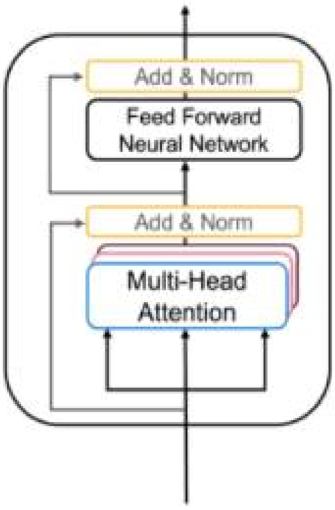
$head_i = Attention(W_i^Q Q, W_i^K K, W_i^V V)$, h = # heads

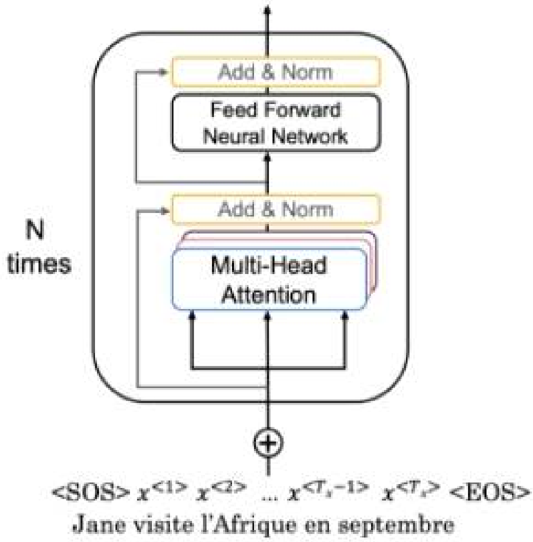
i here represents the computed attention weight matrix associated with ith "head" (sequence).

Transformers

[Vaswani et al. 2017, Attention Is ALl You Need]

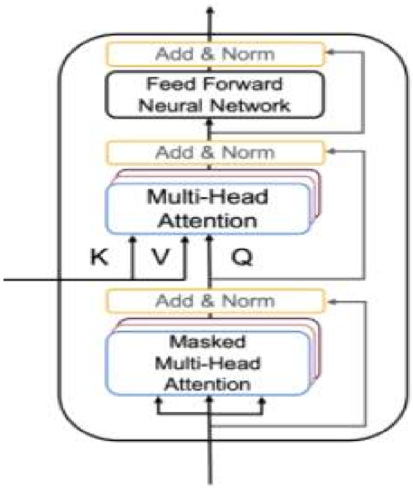
Encoder

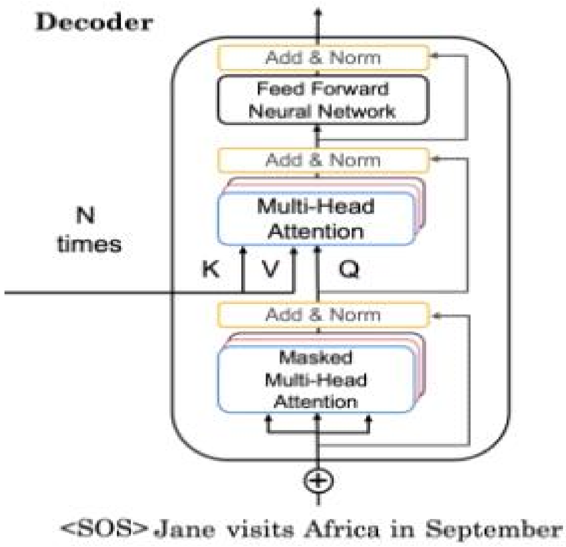
<table>
<tr><th>Transformer encoder layer</th><th>Transformer Encoder</th></tr>
<tr><td>


</td><td>


</td></tr>
</table>

Decoder

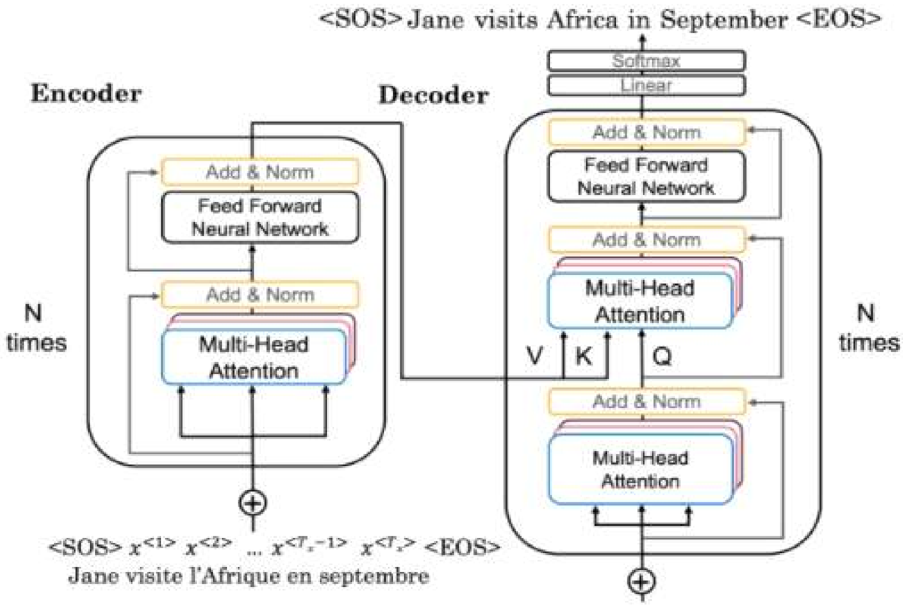
<table>
<tr><th>Transformer decoder layer</th><th>Transformer Decoder</th></tr>
<tr><td>


</td><td>


</td></tr>
</table>

Transformer

Transformer



Positional Encoding

$$PE_{(pos,\,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

$$PE_{(pos,\,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

Why is positional encoding important in the translation process?

1. Position and word order are essential in sentence construction of any language
2. Providing extra information to our model

Good criteria for a good positional encoding algorithm

1. It should output a unique encoding for each time-step (word's position in a sentence)
2. Distance between any two time-steps should be consistent for all sentence lengths.
3. The algorithm should be able to generalize to longer sentence.