Attention Is All You Need (Vaswani *et al.*, 2017)

Presenters: *Vikrant Yadav, Dhruba Pujary, Tarun Krishna*

University of Amsterdam

Motivation

* RNNs/LSTMs model dependencies along a (long) recurrent path.
* Even if the gradient play nice this does not necessarily mean that they model interactions correctly → **credit assignment** problem.
* $\mathbf{h}_{t+1}$ really depend upon $\mathbf{x}_0$ or $\mathbf{x}_1$ or both or neither ?
* In general attention mechanism was motivated by the difficulty of storing large amounts of information into a single, fixed size vector.
* Any long-distance dependency can suffer from having to squeeze large amounts of information into fixed sized representations.

Self-Attention

– **Self-attention** computes attention between elements of the same sequence.
  * can replace RNNs as sequence model
  * shortens paths of credit assignment
  * at the core of Google's Transformer NMT system.

– **Self-attention** is bidirectional (like a Bi-RNN), but no recurrent connections between time steps. It can be visualized as shown below:

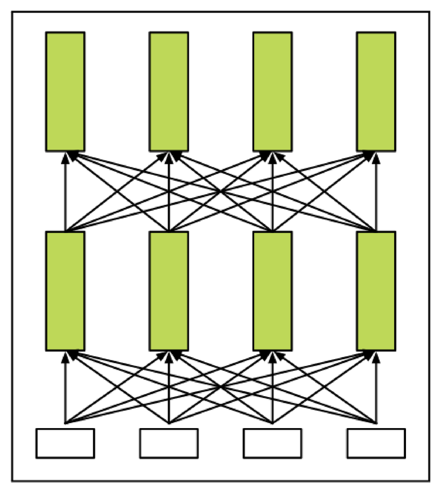


Figure 1: An schematic for self-attention.

Model Architecture

The Transformer follows an **Encoder-Decoder** architecture using stacked self-attention and point-wise, fully connected layers on both the ends with slight changes in decoder Figure 2.

Encoder and Decoder Stacks

– **Encoder**
  * A stack of $N = 6$ identical layers, each layer has two sub-layers
    * a multi-head self-attention mechanism
    * a simple, position-wise fully connected feed-forward network
  * a residual connection around two sub-layers, followed by layer normalization

– **Decoder**
  * A stack of $N = 6$ identical layers, each layer has three sub-layers
    * a multi-head self-attention mechanism
    * a multi-head attention over the output of the encoder stack
    * a simple, position-wise fully connected feed-forward network
  * a residual connection around two sub-layers, followed by layer normalization
  * modified self-attention sub-layer in the decoder stack to prevent positions from attending to subsequent positions using masking.

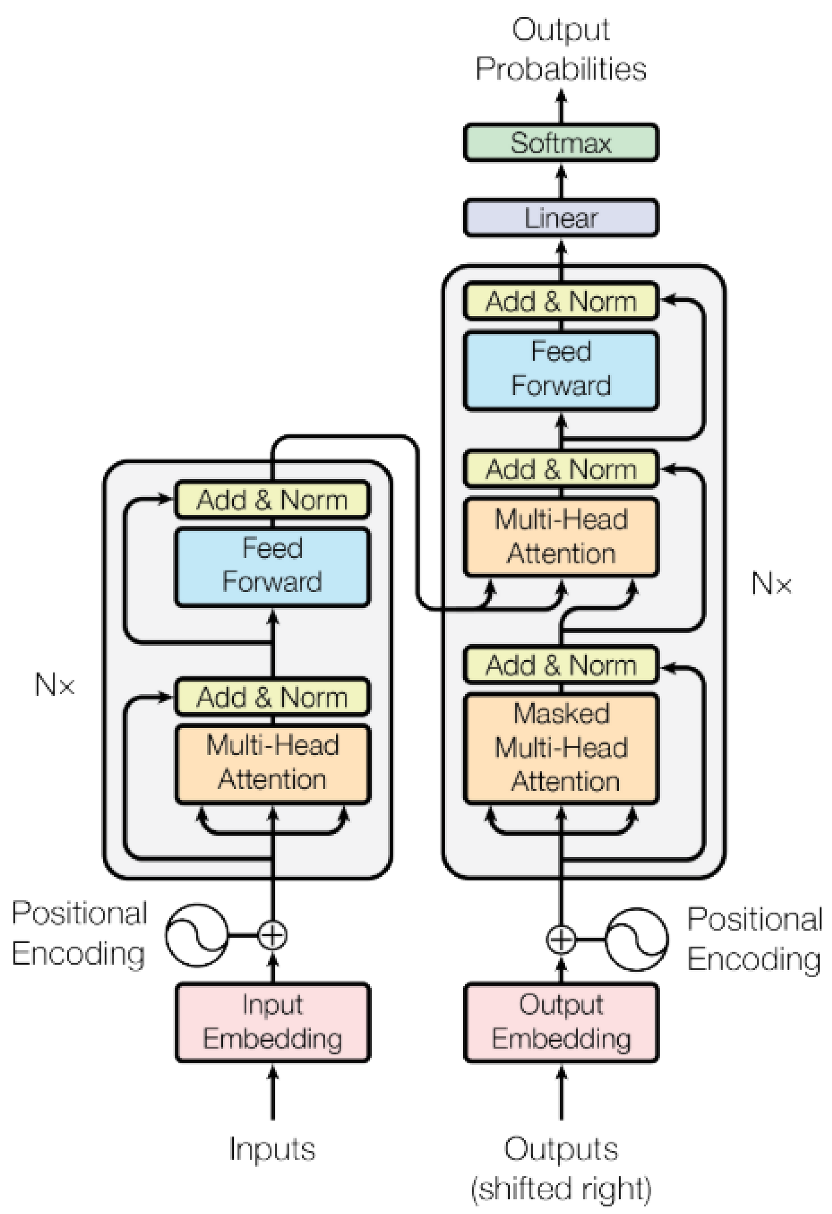
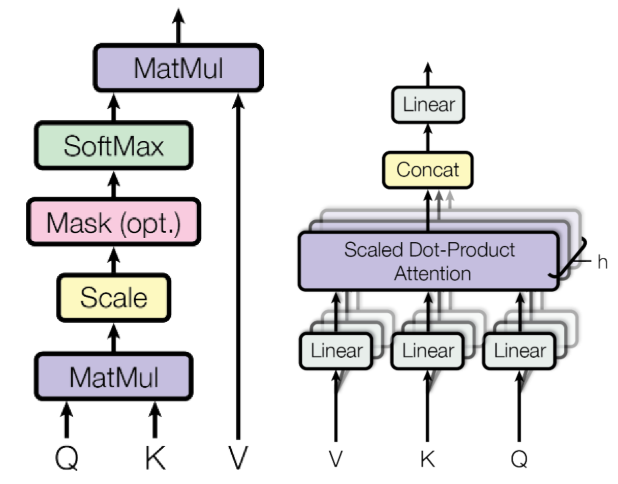


Figure 2: The Transformer - model architecture.

Building Blocks of Transformer



(a) Scaled Dot-Product Attention (b) Multi-Head Attention

Figure 3: Scaled Dot-Product Attention (left), Multi-Head Attention (right) consists of several attention layers running in parallel.

– **Scaled Dot-Product Attention**$(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_K}})V$,Q, K, V are *Query*, *Key* and *Value* matrix respectively, d_k dimension for keys Figure 3.

– **Multi-Head Attention** allows the model to jointly attend to information from different representation subspaces at different positions.

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

– **Position-wise Feed-Forward Networks**: Each of the layers in the encoder and decoder contains a fully connected feed-forward network, which is applied to each position separately and identically as:

$$\mathrm{FFN}(x) = \mathrm{ReLU}(xW_1 + b_1)W_2 + b_2$$

– **Positional Encoding** was introduced to inject some information about the relative or absolute position of the tokens in the sequence. Where *pos* is the position and i is the dimension.

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

Results

The animal didn't cross the street because ***it*** *was too tired*

The animal didn't cross the street because ***it*** *was too wide*

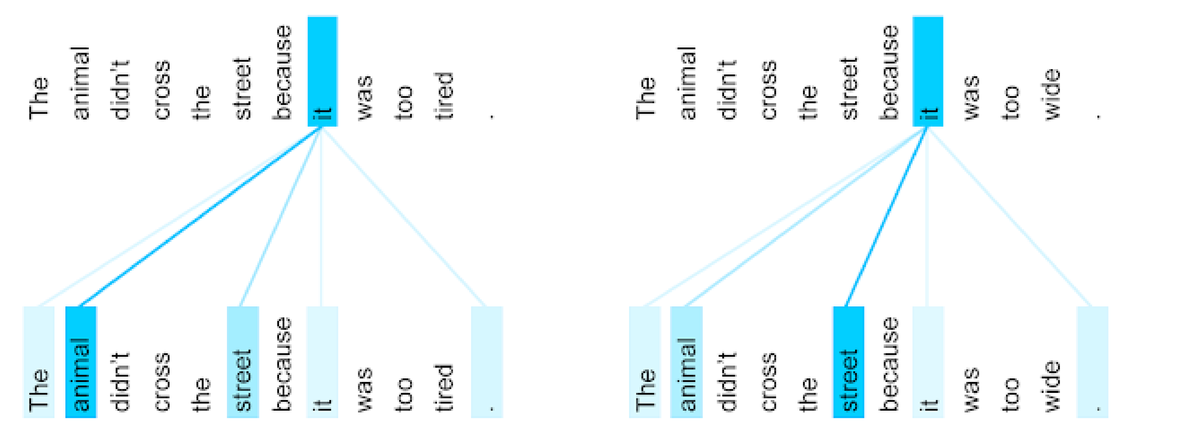


Figure 4: The encoder self-attention distribution for the word **it** while doing translation from English for 2 different context sentences

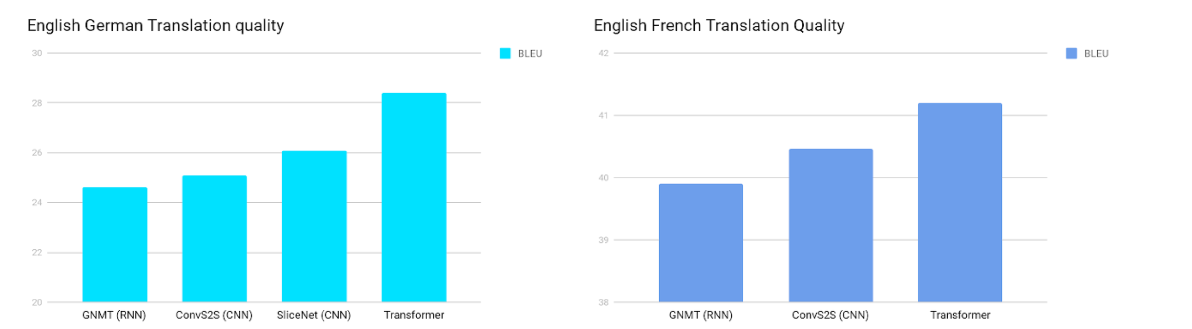


Figure 5: Comparison of Transformer with previous state of art methods

Transformer's Mutation

* core idea of *Transformer* is to find correlation between 2 features which can be distant apart in same domain or be in different domains . For example *(speech and text)* or *(image and text)*
* Different mutated version's of Transformer can be created to find also correlation between cross domain features
* Investigate local, restricted attention mechanisms to efficiently handle large inputs and outputs