Attention is all you need

Paper Summary:

- The paper introduces the transformers for sequence modeling and transduction problems which can potentially replace the **RNNs**, LSTMs and GRUs which are **inherently sequential** and computationally **expensive**.
- The Transformer allows for significantly more parallelization and can reach a new state of the art in translation quality after being trained for as little as twelve hours on eight P100 GPUs.

Network Architecture:

Encoder : Stack of 6 identical layers, each has a **MHSA** and **PWFFNN** layers, all sub-layers in the model, as well as the embedding layers, produce outputs of dimension **512**.
Decoder : Stack of 6 identical layers each has **Masked** MHSA, MHCA and PWFFNN layer. In the masked layer, we make sure the predictions depend on known outputs at positions less than current.
No.of Heads = 8, **Relu** is used as act fn in PWFFNN.

Attention :

- An attention function can be described as **mapping a query** and a set of **key-value pairs** to an output. Most commonly used attention functions are **additive attention**, and **dot-product** attention. While the two are **similar** in theoretical **complexity**, dot-product attention is much faster and more space efficient in practice, since it can be implemented using highly **optimized** matrix **multiplication**.
- Additive attention computes the **compatibility** function using a feed-forward network with a single hidden layer. While dot-product attention is **SDPA without scaling** by root(dk)

Self attention :

Relates different positions of a single sequence in order to compute a representation of the sequence. It has been used successfully in a variety of tasks including reading **comprehension**, **summarization** and learning task-independent sentence representations.
SDPA : **Attention (Q,K,V) = softmax(QK^T/root(dk))V**
For large values of dk, the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients, hence scaling is preferred.
MHA : **MultiHead(Q, K, V) = Concat(head1, ..., headh)W^O,**
Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. Here, dk = dv = dmodel/h to match the SHA.

- In the encoder, all of the keys, values and queries come from the output of the previous layer. Each position in the encoder can attend to all positions in the previous layer of encoder.
- In the decoder, **queries** come from the **previous decoder** layer, and the memory keys and values come from the output of the encoder. This allows **every position** in the decoder to attend over all positions in the input sequence.

Positional encoding : To inject some information about the relative or absolute position of the tokens in the sequence. It can either be learned or fixed.

- To improve computational performance for tasks involving very long sequences, self-attention could be restricted to considering only a neighborhood of size r in the input sequence centered around the respective output position.
- Comparison of attention mechanisms with state-of-art techniques shows, each model has comparative advantage over others in certain areas but still with restricted attention mechanisms it can outperform other models, also as side benefit, self-attention could yield more interpretable models.

Training Process:

Dataset : WMT 2014 **English-German** and **English-French** (using byte-pair encoding).
Hardware: 8 NVIDIA P100 GPUs, **Optimizer**: Adam, **Regularization:** Residual Dropout (Pdrop = 0.1) and Label Smoothing.

Conclusion :

- The Transformer achieves better BLEU scores than previous state-of-the-art models on both the datasets.
- For translation tasks, the Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers.
- It was tested on English constituency parsing as well, where despite the lack of task-specific tuning our model performs surprisingly well, yielding better results than all previously reported models.