Attention Is All You Need

1. Summary

1.1 Abstract

The abstract talks about the common sequence transduction models such as CNN and RNN used for generation of sequential data such as speech, text or language translation. These models typically use an encoder-decoder architecture often combined with an attention mechanism. The research paper aims to introduce a new network architecture called Transformer.

1.2 Introduction

The introduction mostly summarizes the classical ways used for sequential data handling and generation such as RNN, LSTM and Gated Recurrent Units which have established a state-of-art approach. It also talks about the attention mechanism which has improved the efficiency manyfolds and has become an integral part. However, in the older mechanisms such as RNN or LSTM, input tokens are still processed sequentially which make it slower for large inputs due to the unavailability of parallelization. The research paper introduces Transformer model architecture as the new state-of-the-art mechanism for processing sequential data using parallelization and reducing sequential computation with short training periods such as twelve hours on eight P100 GPUs.

1.3 Background

Older models like ByteNet and ConvS2S used parallel processing and CNNs to compute all parts of the sentence at once rather than word-by-word like RNNs. However, as words get farther apart in the sentence, these models struggle more to understand their relationships.

The Transformer architecture fixes this issue by using self-attention. Self-attention mechanism lets each word look at all other words in a sequence including itself. It calculates how much attention a word needs to give to another to understand the sentence better. It can relate all words to each other in just a few steps, irrespective of their distance. However, there is a tradeoff. Attention may average across many words which can blur the details. To fix this, Transforms use Multi-Head Attention which uses multiple sets of self-attention in parallel to learn different relationships.

The Transformer is the first model to drop the usage of both RNN and CNN completely, using only self-attention to process input and output which is a big innovation.

1.4 Model Architecture

The model architecture section dives deep into the working of Transformers and various components that make them.

Encoder and Decoder Stacks

Like many other models, Transformer also contains encoders and decoders. Encoders process the input and Decoders generate the output by looking at what it has generated so far and what the Encoder said about the input.

Encoder

It has six identical layers stacked on top of each other. Each layer has two parts:

Multi-Head Self-Attention: each word looks at all other words in the input to understand the full context.

Feedforward Network: a mini neural network applied to each word's representation. This applies two linear transformations with a Rectified Linear Unit (ReLU) in between. This helps to add non-linearity and deepen the model's ability to learn.

Each of these two parts has a Residual connection which acts as a shortcut from input to output and a Layer normalization to keep values stable during training.

Decoder

It also has six identical layers but each layer has three parts:

Masked Multi-Head Self-Attention: to stop the model from looking ahead at future words during generation.

Encoder-Decoder Attention: allows each output word to attend to the entire input sentence.

Feedforward Network: a mini neural network applied to each word's representation (same as in encoder)

All the three parts have Residual Connection and Layer Normalization same as the Encoder.

Attention

An attention function can be described as mapping a query and a set of key-value pairs to an output. The output is a weighted sum of the values, where the weight of each value is computed by a compatibility function of the query with the corresponding key.

Scaled Dot-Product Attention

In this attention mechanism, each word (query, key, value) is turned into a vector and an Attention score is computed by computing the similarity between query and keys. These scores are used to weigh the value vector and get the final result. The scaling factor $\sqrt{1/dk}$ prevents extremely large values that would mess up the training.

Multi-Head Attention (MHA)
Instead of doing the attention once, the Transformer does it eight times in parallel (known as 8 heads). Each head looks at the sentence in a slightly different way. The results are combined to form a richer understanding.

Applications of Attention in our Model
The Transformer uses attention in three places:

1. **Encoder self-attention:** Each word in the input attends to all others
2. **Decoder self-attention:** Each output word attends only to earlier words and not the future ones
3. **Encoder-decoder attention:** The decoder attends to the encoder output

Embeddings and Softmax
Words (tokens) are converted to vectors using embedding layers. After decoding, the final output is passed through a softmax to get probabilities for the next word. In this model, the same weight matrix is shared between two embedding layers and the pre-softmax linear transformation.

Positional Encoding
The model doesn't have recurrence or convolutions, so it doesn't know what order the words are in. To fix that, positional information is added using sin and cos patterns. The encoding helps the model to determine the position of each word and learn patterns based on relative positions.

1.5 Why Self-Attention?

- Self-attention can connect all words in just one step, and it's highly parallelizable and fast
- Every word can attend to every other word directly. This makes it easier to learn long-range relationships
- RNNs process one word at a time and hence are slow and require many steps to relate distant words, therefore self-attention is better
- CNNs need many layers to connect distant words, and they're more computationally expensive unless optimized, therefore self-attention is better

1.6 Training
This section summarizes how the training for Transformer model was done:

Training Data and Batching

- For English to German, a standard dataset was used with 4.5 million sentence pairs

- For English to French, a larger dataset with 36 million sentence pairs was used
- To convert words into model-readable form, byte-pair encoding (BPE) was used
- Sentence pairs of similar lengths were batched together to optimize memory and computation
- Each batch had ~25,000 tokens (words/subwords) from both the input and output sides

Hardware and Training Time
Training ran on 1 machine with 8 NVIDIA P100 GPUs.

Base model:
- Each training step = **0.4 seconds**
- Total training = **100,000 steps ≈ 12 hours**

Big model:
- Each step = **1 second**
- Total training = **300,000 steps ≈ 3.5 days**

Optimizer
Used the Adam optimizer, which is great for NLP tasks.
Tweaked learning rate using a "warm-up" schedule:

- Gradually increase the learning rate for the first 4000 steps
- After that, decrease it based on step number
- This approach helps stabilize training early on and prevent big jumps in updates

Regularization

Dropout (10%):

- Randomly “drop” parts of the network during training to prevent over-reliance on certain paths

- Applied to:
  - Each sub-layer output.
  - Embeddings + positional encodings.

Label Smoothing ($\epsilon = 0.1$):

- Instead of training the model to be 100% sure of the correct answer, it was intentionally made a bit uncertain
- This improves generalization and BLEU score, even if it slightly worsens "perplexity" (a confidence metric)

1.7 Results and Conclusion

The Transformer model outperforms all prior models on translation tasks, setting new records on the WMT 2014 English–German (BLEU 28.4) and English–French (BLEU 41.0) benchmarks, even with less training cost. It generalizes well across configurations — larger models perform better, attention heads have a sweet spot, and techniques like dropout and label smoothing help. The model also adapts well beyond translation: it achieved top-tier results on English constituency parsing, outperforming many traditional models even with limited data. Overall, the Transformer proves that attention alone, without recurrence or convolution, is highly effective for sequence tasks, and it opens the door for attention-based models across diverse domains like text, images, audio, and video.

1.8 References

[1] Attention Is All You Need - Research Paper

Summary By:
Kartikey Vijayakumar Hebbar
Northeastern University, Boston
Intro to AI Agents - Summer 2025