PAPER DISSECTION

’Attention is All You Need’: The Blueprint of Modern NLP

Omkar Ingale and Aiqi Shuai

University of Zurich, Switzerland

Abstract

The Transformer architecture introduced in “Attention is All You Need” has redefined the field of natural language processing and sequence modeling by eliminating the need for recurrent and convolutional structures. This dissection delves into the key innovations of the Transformer, including the self-attention mechanism, multi-head attention, and positional encoding, which enable parallelization and efficient handling of long-range dependencies.

Keywords: Transformer Architecture, Deep Learning, Natural Language Processing, Attention Mechanism

1. Motivation for Choice of Paper

The paper “Attention is All You Need”, introduced by Vaswani et al. in 2017[a], revolutionized the field of natural language processing (NLP) by proposing the Transformer architecture. Unlike traditional models relying on recurrent or convolutional layers, the Transformer utilizes an attention mechanism as its core building block, enabling parallelization and more efficient processing of sequential data. This work not only addressed limitations in handling long-range dependencies but also set new benchmarks in tasks such as machine translation. This dissection aims to provide a detailed summary and analysis of the concepts and methodologies presented in the original paper. We will also try to explore how the transformer model has evolved through the years as newer research was published.

2. ML Methods and Innovation

The Transformer model, introduced in "Attention is All You Need," leverages several key innovations that have redefined sequence-to-sequence modeling:

2.1 Self-Attention Mechanism

The self-attention mechanism is the Transformer’s key ability to process sequences. Unlike recurrent or convolutional models, which rely on sequential data processing, self-attention computes pairwise dependencies across the entire input sequence simultaneously. This allows the model to focus on relevant parts of the sequence, regardless of their distance from the current token. Mathematically, self-attention is computed as:

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

where Q, K, and V represent query, key, and value matrices derived from the input embeddings, and d_k is the dimensionality of the keys.

2.2 Multi-Head Attention

To enhance the model’s ability to capture diverse relationships within the data, the Transformer employs multi-head attention. By projecting the input into multiple subspaces and applying self-attention independently in each, the model can capture a wide range of contextual relationships. The outputs from all attention heads are concatenated and linearly transformed to produce the final output:

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

where W^O is the output projection matrix.

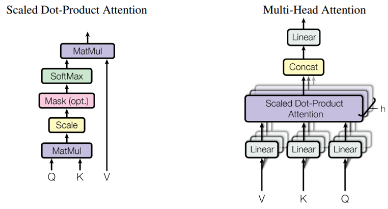

Figure 1. (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

2.3 Positional Encoding

Transformers do not process tokens sequentially, so they require a mechanism to incorporate positional information. Positional encodings are added to the input embeddings to provide this context. These encodings are computed using sinusoidal functions:

$$\text{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right),$$

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

where pos is the position index and i represents the dimension.

[a] Vaswani et al. 2017

2.4 Feed-Forward Networks

Each Transformer layer includes a position-wise feedforward network (FFN) that applies two linear transformations with a ReLU activation in between. The FFN operates independently on each position, enhancing the model's representational capacity:

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

2.5 Layer Normalization and Residual Connections

To improve training stability and convergence, the Transformer incorporates layer normalization and residual connections around each sublayer (e.g. self-attention and FFN). This ensures effective gradient flow during backpropagation and facilitates the learning process.

2.6 Encoder-Decoder Structure

The Transformer adopts an encoder-decoder architecture, where:

- The encoder processes the input sequence into a sequence of continuous representations.
- The decoder generates the output sequence by attending to the encoder's output and previously generated tokens.

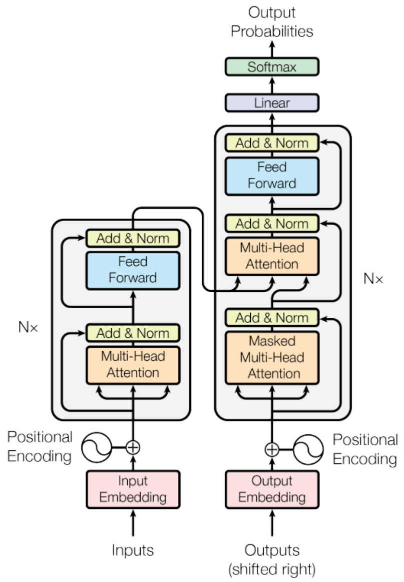

Figure 2. Encoder and Decoder Structure

2.7 Parallelization and Efficiency

By eliminating recurrent connections, the transformer enables parallel processing of input sequences. This significantly reduces training time and allows the model to handle long-range dependencies more effectively than traditional architectures, such as RNNs or LSTMs.

3. Takeaways

The transformer architecture showcased improved scores on major benchmarks such as BLEU. The following table summarises their reported scores.

Table 1. BLEU Scores as reported in original paper

Model	EN-DE	EN-FR
ByteNet (Kalchbrenner et al. 2017)	23.75	
Deep-Att + PosUnk (Zhou et al. 2016)		39.2
GNMT + RL (Wu et al. 2016)	24.6	39.2
ConvS2S (Gehring et al. 2017)	25.16	40.46
MoE (Shazeer et al. 2017)	26.03	40.56
Deep-Att + PosUnk Ensemble (Zhou et al. 2016)		40.4
GNMT + RL Ensemble (Wu et al. 2016)	26.30	41.16
ConvS2S Ensemble (Gehring et al. 2017)	26.36	**41.29**
Transformer (Base)	27.3	38.1
Transformer (big)	**28.4**	**41.8**

The key takeaway from the paper is the Transformer's self-attention mechanism, which allows the model to focus on different parts of an input sequence simultaneously, enabling efficient handling of long-range dependencies. By eliminating recurrent and convolutional layers, the Transformer achieves parallelization, significantly reducing training times compared to sequential models like RNNs and LSTMs. The multi-head attention mechanism enhances the model's capacity to capture relationships at different levels of abstraction, while positional encodings provide important information about the order of input tokens. These innovations, coupled with the encoder-decoder structure, set new benchmarks in machine translation tasks such as English-to-German and English-to-French translations. Apart from translation, the Transformer has contributed to the advancements in NLP, including large-scale language models like BERT and GPT.

4. Problems of Approach

While the paper contributed to significant advances in NLP, there are certain problems/drawbacks in it's architecture. Some of them are listed below.

4.1 Quadratic Computational Complexity

According to the paper, the self-attention mechanism has a quadratic computational complexity which scales quadratically with the length of the input sentence. As a result, processing longer input sentences becomes computationally expensive and memory intensive.

4.2 Lack of Explicit Recurrence

The Transformer model does not have a explicit recurrence mechanism. It compensates for this using positional encoding that injects a sense of sequential order into the architecture. This works in most cases but does not model sequential relationships as well as recurrent architectures.

5. Conclusion and Developments

The paper by Vaswani et al. has practically redefined the field of NLP by introducing the Transformer architecture. Innovations like the attention mechanism and the use of positional encodings to pave the way for parallelization and eliminate sequential processing have changed the field forever. This paper has contributed significantly to the AI and LLM revolution by being the foundation stone for GPT, BERT, contextualized encodings and much more.

Another paper that builds on this progress in "Tensor Product Attention Is All You Need"[b] which tries to shrink the large KV caches required by the original transformer. They do this by using tensor decompositions to represent queries, keys and values compactly, thereby significantly reducing the required memory at inference time. Such innovations could not be possible without the original Transformer paper.

6. Helpful Resources

1. ChatGPT: For understanding and resolving doubts about the paper.
2. YouTube: Channels like Rasa, 3Blue1Brown that help breakdown and visualise the workings of the transformer.
3. Medium Articles: Several medium articles dive in-depth and provide a great understanding of the Transformer architecture.

References

Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122v2.*

Kalchbrenner, Nal, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2017. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099v2.*

Shazeer, Noam, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: the sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538.*

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems,* vol. 31.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144.*

Zhang, Yifan, Yifeng Liu, Zhen Qin Huizhuo Yuan, Yang Yuan, Quanquan Gu, and Andrew Chi-Chih Yao. 2025. Tensor product attention is all you need. In *Arxiv:2501.06425v1.*

Zhou, Jie, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep recurrent models with fast-forward connections for neural machine translation. *CoRR* abs/1606.04199.

[b] Zhang et al. 2025