"Attention Is All You Need" Digested

Sepehr Sameni
Department of Artificial Intelligence
University of Tehran
sepehr.sameni@gmail.com

Abstract

In this short document, I will discuss the strengths and weaknesses of "Attention is all you need"[31] by looking at the follow-up papers and conclude with some potential improvements on top of transformers based on high computational complexity.

1 Sequence to Sequence Models

One of the most powerful formulations of many machine learning problems is seq2seq. Given an input sequence, we seek to generate the corresponding output sequence with a possibly different length than the input. Although this is a vague definition, It used to be implemented as a recurrent encoder followed by a recurrent decoder[29]. This vanilla solution has four major problems:

1. long training time caused by the sequential nature of the encoder and decoder.
2. long inference time caused by the autoregressive decoder.
3. order dependency in encoder and decoder, even though it's not a problem for text and we can almost always read and write words from left to right(this has also been challenged recently by [12]), it is not clear how to best read pixels of a picture[7]
4. the information bottleneck; encoding the whole sequence into one fixed size vector is not efficient and posses hard restrictions on the encoder.

There has been a lot of research to solve these problems including: [3] that introduced an attention mechanism in the decoder and solved the information bottleneck(we should note that the tradeoff here is that they also increased the memory consumption of the network during inference), convolutional encoder and decoder[13] that addressed the first issue by introducing a non recurrent building block for encoding and decoding, the tradeoff here is the depth, in order for the network to be able to model long dependencies of the input, they had to increase the depth of the network.

The next step toward improving seq2seq is using something without order dependence in the encoder which Transformers achieve pretty well and interestingly enough they obtain better results with shorter training times.

It is also important to note that even though this paper is mainly focused on NMT, seq2seq has many more applications, including image caption generation[33], video captioning[32], Speech recognition[8], speech generation[30], language modeling[11], document classification[22] and even multi task learning[19].

2 Transformer

In this paper, authors ditched rnn and cnn altogether and replaced them with only self-attention and cross attention, by doing this they were able to achieve state of the art results on NMT with a fraction of computing power compared to the previous state of the art results. In the following sections, I will

talk about different components of their model, talk about their pros and cons and cite papers about those individual parts.

2.1 Attention

Bahdanau et al[3] introduced attention to decoders and since then we have seen a lot of improvements caused by attention[18] and it makes sense to assume that it is a crucial part of modern NMT and using it in the encoder(which is a novel thing in this paper) should yield higher accuracy(BLEU score). Although this is a sensible assumption, it does not have any mathematical proofs supporting it and might be wrong altogether as challenged in You May Not Need Attention[23]

Another reason in favor of using attention is the more explainability of the network that for years had been assumed to be true. Even though by using multi-head attention they have weakened this point, explainability is still possible in transformer networks[2]. But this assumption can be wrong, in the recently published paper titled Attention is not Explanation[17] authors show that attention weights are not necessary interpretable.

The silver bullet is that attention mechanism helps the vanishing gradient by reducing the maximum dependency length between tokens and that is enough to explain their training efficiency. But are they here to stay? Do we have any mathematical proofs for their expressibility? First, in the paper called Universal Transformers[10] authors show that under mild assumptions, Recurrent Transformers can be Turing Complete and it might mean that RNNs can be replaced with their Transformer counterpart.

Another interesting line of research is to prove that Transformers (or a variant of them) are Universal function approximators like [36] that shows any translation equivariant function can be approximated arbitrarily well by a convolutional neural network given that it is sufficiently wide, in direct analogy to the classical universal approximation theorem.

One of the missing things in the paper is that they talked about restricted attention and compared them with other models in the first table but there are no experiments on them. I know that implementing them efficiently is really hard (and maybe impossible given the current memory layouts) but they could implement them via attention masking and compare the results to show whether full sequence attention in the base model is leveraged or not.(see figure 1)

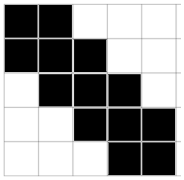

Figure 1: Resctricted Attenttion Map with $n = 5$ and $r = 3$

Another missing experiment for the Attention module is the lack of self-attention visualization in the decoder, it makes sense for the encoder to learn meaningful attention weights (like coreference resolution and parsing) but what does the decoder's masked self-attention show? Is it a masked version of the same things? Or something non-interpretable for humans?. Also, attention visualization in figure 5 is not that informative and the caption of the figure is rather vague.

2.2 Multi-Head Attention

Explanation of multi-head attention is pretty clear, and it's a great way to combat the lower resolution precision that comes with attention averaging and there is an ablation study on the number of heads but they didn't visualize how increasing or decreasing the number of heads can hurt the network's performance.

Implicitly different heads learn different concepts and that is interesting but the tradeoff here is that it increases the possibility of overfitting and probably increasing the training time. One can easily combat this with another implicit method like Weighted Transformers[1] by introducing a few trainable parameters and encouraging difference between heads. It is also shown in[20], applying a disagreement regularizer to explicitly encourage different heads to attend to different locations

helps to overcome overfitting and it even helps to encode our prior knowledge about the sentence in heads[28]

2.3 Layer-Norm

The paper does not explain the necessity of layer norm in training and the intuition on choosing layer norm instead of the batch norm(we know that layer norm is better than the batch norm when we have recurrent modules but what about transformers?)

Another general issue with normalization layers is their lack on understanding, as it is shown recently in[24] common belief about batch norm, that they control the change of the layers' input distributions during training to reduce the so-called "internal covariate shift" does not hold true and indeed it makes the loss landscape more smooth. But we don't have such a theory for layer norm and using it is just practical but not mathematically sound. On top of these, it has recently been shown[39] that with careful initialization we can drop batch norm and layer norm layers and train even deeper networks without any issues and obtain better results than the base transformer in NMT.

2.4 Position Embedding

The necessity of position embedding is clear but it is not clear how the addition of them to the word vectors changes the behavior of word vectors and helps the network to understand the position. They also justified their choice of position embedding based on sine and cosine with dependence on only starting position vector and distance(but not the other position vector) but they didn't do any experiments on this and didn't show that the network is capable of generating and consuming sequences longer than the ones seen during training.

As future studies show, position embedding is much more important and explicitly modeling the relative distance can improve the results[25] and improved in [9] by including an inductive bias raised from the sinusoid encoding matrix.

2.5 Feed Forward Network

It is not clear how the FFN helps the training and why it is needed, it can be argued that they are inserted to make the whole network nonlinear but it is not true because attention is not linear and even though the output of the network, if we only used self-attention layers, is just a linear combination of all tokens but it is input dependent and thus nonlinear. Once again this shows that we need a better theoretical understanding of Transformers in order to justify the FFN. Another point worth mentioning is that it seems FFN play an important role in the Transformer because they are not replaced in The Evolved Transformer[27].

2.6 Masked Decoder

Decoding is inherently sequential and forcing our network to work sequentially via masking is a bit unnatural and it forces the decoder to work in a strange way, as I said replacing calculated self-attention with uniform averaging does not decrease accuracy and it means that we can do better, as it is stated in [6]: "On the other hand, lacking a memory component (as present in the RNN models) prevents the network from modeling a state space,reducing its theoretical strength as a sequence model, ..." they showed that combining a self-attentive encoder and a recurrent decoder improves accuracy (with the cost of increased training time) and I think it's a reasonable middle ground.

2.7 Inference

Even though it's not part of this paper's goals but it worth mentioning that autoregressive models like this will not improve the inference time and as a matter of fact it will increase the memory budget for inference because of the self-attention in decoder that needs to store all intermediate presentations for previous tokens for the current token which can be solved with an embarrassingly simple idea like [37] which is surprising, it seems that self-attention in the decoder is not that important and once again it raises the question that what does the self-attention in decoder learns and bothers the reader with this lack of experiment.

Another interesting idea to improve inference time is to use a Non-Autoregressive NMT by replacing the teacher forcing method of training with token fertilities and using the full power of decoder(no masked attention) to generate the whole output sequence in one pass and reduce the latency by an order of magnitude[14]

2.8 Experiments

The ablation study is good and it aligns well with the paper claims and authors' intuitions. But I would love it if there were a multilingual translation study to show that the self-attention mechanism can be language independent. It was perfect if they also showed something like CoVE[21] and show the power of the leaned encoder as a contextualized word embedding extractor. The constituency parsing is not well explained and they are just there to make the paper a bit longer, I think there are many better applications for seq2seq that could have been shown to impress the reader. But overall the experiments are sound and they are well developed.

3 Possible Improvements

3.1 Capsule Attention

3.1.1 Problem Statement

The computational complexity of transformer networks are quadratic in sequence length($\mathcal{O}(n^2d)$) and that prevents them to be used on mobile devices or large audio sequences

3.1.2 Possible Solution

I argue that the full attention is not necessary for the network and it suffices for the network to only attend to the most important concepts of the sequence. So instead of generating a K(key) and V(value) with size n for the attention, we can generate them with much lower size k(which can be a constant number like 16) via an EM algorithm like [40] or even vanilla soft K-means over the full K and V. So as in the base transformer, we calculate K, V, and Q then instead of calculating the dot product between K and Q we first summarize K from size $n * d$ to $k * d$ via weight-less dynamic routing and we use the same weights to also summarize the V into $k * d$, afterward we calculate attention between Q and the summarized K and multiply by the summarized V which yields a sequence of size $n * d$ as before but with a much lower computational complexity of $\mathcal{O}(nkd)$

3.1.3 Prevoius Work

As shown in Star Transformer[15], we can propagate information into different nodes with a single intermediate node, which is like setting k in CAT to 1, but one should note that Start Transformer is applied for T times in each layer and those calculations are dependent and cannot be parallelized, in contrast in CAT all k centers are calculated in parallel and there are no iterations(except for the kmeans which can take as little as 2 iterations)

Another similar work is[34] which uses capsule networks at the last step of encoder but CAT is the extreme case of that idea and applies this idea in all the layers of both encoder and decoder's self-attention

[26] also tried to make attention linear in sequence length by changing the order of attention calculation and come up with an $\mathcal{O}(nd^2)$ algorithm which is more compute expensive than CAT for typical network sizes.

As shown in [27] and [35] convolution is still relevant and combining convolution networks for local relation extraction and CAT for global context extraction seems legit.

3.1.4 Possible Application

Not only we might be able to train larger BERT[11] or GPT-2 models and improve the current state of the art NLP models. With such an algorithm we can easily apply self-attention to much longer inputs like pixels and audio and improve GAN results[38]. In theory, we should be able to use CAT in all seq2seq tasks and improve their results by training larger models and better-representing context.

3.2 Dilated Attention

3.2.1 Problem Statement

The computational complexity of transformer networks are linear in depth($\mathcal{O}(ln^2d)$) and that prevents them to be used on mobile devices or large audio sequences.

3.2.2 Possible Solution

We can use self-attention dilation to reduce their complexity to $\mathcal{O}(2n^2d) = \mathcal{O}(n^2d)$ by using exponentially grown dilation rate in upper layers of the network as shown in figure 2. Once again I argue that at least with the current methods, full-sequence attention is not being utilized and most transformer networks are under fitted and this is great opportunity to weaken their structure and fully utilize the resulting network.

Figure 2: Dilated Self Attention(nodes with the same color are analyzed together)

3.2.3 Prevoius Work

Time[4] and time again[5] it has been shown that dilation improves parameter efficiency and final accuracy of the networks.

3.3 Mobile Transformer

3.3.1 Problem Statement

As I mentioned multi-head attention is overparameterized and it can be easily regularized either via implicit methods over explicit ones.

3.3.2 Possible Solution

I want to implement a regularizer by architecture design that resembles separable convolutions[16] and light convolutions[35]. Instead of having different heads that use different projection weights we first split the input and project resulting parts and at the end instead of using another linear layer on the concatenated vector, I rely on the FFN to fuse inputs (like mobile net's 1x1 cnn)

3.3.3 Calculations

for the vanilla transformer model we have $3(key, value, query) * h(heads) * d_{model}(input) * d_{qkv}(intermediate) + h(concat) * d_{qkv}(intermediate) * d_{model}(final) + d_{model}(ffn_{input}) * d_{ff}(intermediate) + d_{ff} * d_{model}(final)$ which is 3145728 for the base model. for the mobile version(with a moved residual connection) we have $3 * h * d_{model}//h(splited) * d_{qkv} + 0$(no need to resize intermediate representation)$+h * d_{qkv} * d_{ff} + d_{ff} * d_{model}$ which is equal to 2195456, one million parameters saved per layer for 12 layers of encoder and decoder in base-transformer we can reduce the total parameters by 20%

References

[1] Karim Ahmed, Nitish Shirish Keskar, and Richard Socher. Weighted Transformer Network for Machine Translation. *arXiv e-prints*, art. arXiv:1711.02132, Nov 2017.

[2] Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. On The Alignment Problem In Multi-Head Attention-Based Neural Machine Translation. *arXiv e-prints*, art. arXiv:1809.03985, Sep 2018.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv e-prints*, art. arXiv:1409.0473, Sep 2014.

[4] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv e-prints*, art. arXiv:1803.01271, Mar 2018.

[5] Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark Hasegawa-Johnson, and Thomas S. Huang. Dilated Recurrent Neural Networks. *arXiv e-prints*, art. arXiv:1710.02224, Oct 2017.

[6] Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation. *arXiv e-prints*, art. arXiv:1804.09849, Apr 2018.

[7] Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. PixelSNAIL: An Improved Autoregressive Generative Model. *arXiv e-prints*, art. arXiv:1712.09763, Dec 2017.

[8] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. State-of-the-art Speech Recognition With Sequence-to-Sequence Models. *arXiv e-prints*, art. arXiv:1712.01769, Dec 2017.

[9] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *arXiv e-prints*, art. arXiv:1901.02860, Jan 2019.

[10] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal Transformers. *arXiv e-prints*, art. arXiv:1807.03819, Jul 2018.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, art. arXiv:1810.04805, Oct 2018.

[12] Nicolas Ford, Daniel Duckworth, Mohammad Norouzi, and George E. Dahl. The Importance of Generation Order in Language Modeling. *arXiv e-prints*, art. arXiv:1808.07910, Aug 2018.

[13] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional Sequence to Sequence Learning. *arXiv e-prints*, art. arXiv:1705.03122, May 2017.

[14] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. Non-Autoregressive Neural Machine Translation. *arXiv e-prints*, art. arXiv:1711.02281, Nov 2017.

[15] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-Transformer. *arXiv e-prints*, art. arXiv:1902.09113, Feb 2019.

[16] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv e-prints*, art. arXiv:1704.04861, Apr 2017.

[17] Sarthak Jain and Byron C. Wallace. Attention is not Explanation. *arXiv e-prints*, art. arXiv:1902.10186, Feb 2019.

[18] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *arXiv e-prints*, art. arXiv:1611.04558, Nov 2016.

[19] Lukasz Kaiser, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. One Model To Learn Them All. *arXiv e-prints*, art. arXiv:1706.05137, Jun 2017.

[20] Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. Multi-Head Attention with Disagreement Regularization. *arXiv e-prints*, art. arXiv:1810.10183, Oct 2018.

[21] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in Translation: Contextualized Word Vectors. *arXiv e-prints*, art. arXiv:1708.00107, Jul 2017.

[22] Nikolaos Pappas and Andrei Popescu-Belis. Multilingual Hierarchical Attention Networks for Document Classification. *arXiv e-prints*, art. arXiv:1707.00896, Jul 2017.

[23] Ofir Press and Noah A. Smith. You May Not Need Attention. *arXiv e-prints*, art. arXiv:1810.13409, Oct 2018.

[24] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How Does Batch Normalization Help Optimization? *arXiv e-prints*, art. arXiv:1805.11604, May 2018.

[25] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-Attention with Relative Position Representations. *arXiv e-prints*, art. arXiv:1803.02155, Mar 2018.

[26] Zhuoran Shen, Mingyuan Zhang, Shuai Yi, Junjie Yan, and Haiyu Zhao. Factorized Attention: Self-Attention with Linear Complexities. *arXiv e-prints*, art. arXiv:1812.01243, Dec 2018.

[27] David R. So, Chen Liang, and Quoc V. Le. The Evolved Transformer. *arXiv e-prints*, art. arXiv:1901.11117, Jan 2019.

[28] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-Informed Self-Attention for Semantic Role Labeling. *arXiv e-prints*, art. arXiv:1804.08199, Apr 2018.

[29] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. *arXiv e-prints*, art. arXiv:1409.3215, Sep 2014.

[30] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. *arXiv e-prints*, art. arXiv:1609.03499, Sep 2016.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv e-prints*, art. arXiv:1706.03762, Jun 2017.

[32] Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to Sequence – Video to Text. *arXiv e-prints*, art. arXiv:1505.00487, May 2015.

[33] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and Tell: A Neural Image Caption Generator. *arXiv e-prints*, art. arXiv:1411.4555, Nov 2014.

[34] Mingxuan Wang, Jun Xie, Zhixing Tan, Jinsong Su, Deyi xiong, and Chao bian. Towards Linear Time Neural Machine Translation with Capsule Networks. *arXiv e-prints*, art. arXiv:1811.00287, Nov 2018.

[35] Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. Pay Less Attention with Lightweight and Dynamic Convolutions. *arXiv e-prints*, art. arXiv:1901.10430, Jan 2019.

[36] Dmitry Yarotsky. Universal approximations of invariant maps by neural networks. *arXiv e-prints*, art. arXiv:1804.10306, Apr 2018.

[37] Biao Zhang, Deyi Xiong, and Jinsong Su. Accelerating Neural Transformer via an Average Attention Network. *arXiv e-prints*, art. arXiv:1805.00631, May 2018.

[38] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-Attention Generative Adversarial Networks. *arXiv e-prints*, art. arXiv:1805.08318, May 2018.

[39] Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Fixup Initialization: Residual Learning Without Normalization. *arXiv e-prints*, art. arXiv:1901.09321, Jan 2019.

[40] Suofei Zhang, Wei Zhao, Xiaofu Wu, and Quan Zhou. Fast Dynamic Routing Based on Weighted Kernel Density Estimation. *arXiv e-prints*, art. arXiv:1805.10807, May 2018.