From Attention Is All You Need to GPT, BERT, and Some More

Munia Humaira
University of Waterloo, ON, Canada

***Abstract*—Transformers have revolutionized NLP by enabling parallelized training through self-attention processes and circumventing the limitations of traditional sequential models such as RNNs and LSTMs. This report critically examines the Transformer architecture, from its introduction in the seminal paper ”Attention is All You Need” to its adaptation in state-of-the-art models such as GPT and BERT. The study first examines the Transformer's basics, including its encoder-decoder structure and self-attention mechanism, to highlight its benefits and drawbacks, including its reliance on hyperparameter tuning and computational complexity. The extension of the Transformer by GPT and BERT is next examined, with BERT incorporating a bidirectional encoder for enhanced context understanding and GPT utilizing a unidirectional decoder for auto-egressive language modeling. These models' innovations and trade-offs are evaluated critically by discussing how they affect NLP tasks and what problems they still have. By connecting these three pieces, readers will understand how Transformers has impacted modern natural language processing.**

***Index Terms*—Transformers, Natural Language Processing, BERT, GPT**

I. Introduction

Transformers have set a new benchmark in Natural Language Processing (NLP) by enabling parallelized training with self-attention mechanisms. With the help of this mechanism, they process sequential data, allowing them to capture long-range dependencies more effectively than previous models like Recurrent Neural Networks (RNNs) [1], Long Short-Term Memory Networks (LSTMs) [2], and so on. This breakthrough was first introduced in the seminal paper ”Attention is All You Need” by Vaswani et al. (2017) [3], which laid the foundation of NLP models, including Generative Pre-trained Transformer (GPT) and Bidirectional Encoder Representations from Transformers (BERT).

In this critical summary report, I will begin by discussing the Transformer architecture and highlighting its key pros and weaknesses. I'll then explain how GPT and BERT evolved from the Transformer, emphasizing the trade-offs made in their design as well as their unique contributions. My purpose in combining these three studies is to provide readers with a comprehensive understanding of how Transformers have affected contemporary NLP and where the field may be headed.

II. Transformer Architecture

The Transformer model is auto-regressive; it consumes the previously generated symbols as additional input when generating the next. The two main components of the Transformer are the encoder and the decoder. The encoder converts an input sequence of symbol representations $(x1, ..., xn)$ to a sequence of continuous representations $z = (z1, ..., zn)$. Since z is given, the decoder creates an output sequence $(y1, ..., ym)$ of symbols individually. In Figure 1, the Transformer model architecture has been shown. One encoder layer out of $N = 6$ layers has been displayed on the left. A position-wise fully connected feed-forward network and a multi-head self-attention layer are the two sub-layers that make up each layer. Layer normalization and residual connections are used for each sub-layer. The decoder also includes $N = 6$ stacked layers. The diagram's right side shows one layer of the decoder. Each layer is composed of three sub-layers, two of which have the same encoder. The third layer performs multi-head attention over the output of the encoder stack.

The Transformer's self-attention mechanism, which enables the model to consider the relative relevance of several words in a sequence while encoding or decoding, is its key innovation. The paper took advantage of ”Scaled Dot Product Attention,” a modified dot product attention mechanism that is exceptionally fast to compute, especially on GPU. The attention matrix, given dimension d_k queries and keys and dimension d_v [4] values, is computed as follows

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}V\right), \qquad (1)$$

where $Q, K, \&\ V$ are queries, keys, and values all packed together as matrices, respectively. Multi-head attention allows this attention to be computed in parallel, which helps to focus on different positions, and because there are more attention heads, it also helps to attend to information from different subspaces.

While Transformers introduced groundbreaking innovation at that time, it had several notable shortcomings. The GitHub code repository [5] of the paper revealed that the model relied on a large number of hyper-parameters, many of which were not explicitly detailed in the paper, making replication and fine-tuning challenging. Including these details would have improved clarity and reproducibility. Unlike LSTMs, Transformers require a carefully tuned learning rate schedule, as simple optimization methods like stochastic gradient descent (SGD) failed to work effectively. The computational complexity of the Transformer could act as a bottleneck, especially when dealing with lengthy sequences. The self-attention mechanism's time complexity is $\mathcal{O}(n^2)$, where n is the sequence length [6]. For tasks involving extremely long

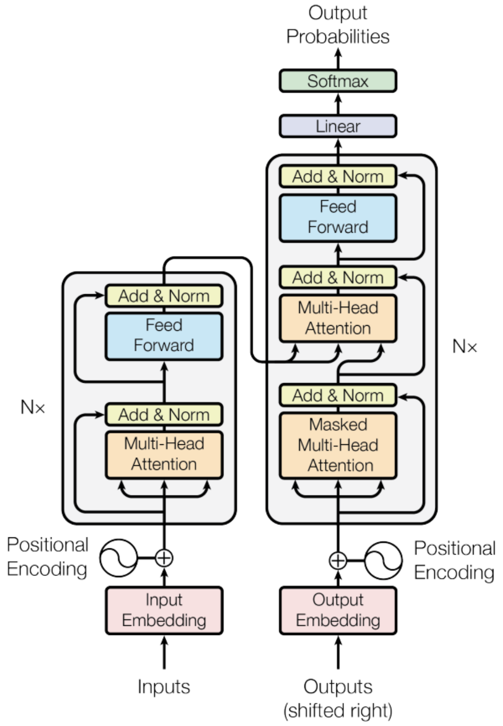


Fig. 1. The Transformer model architecture from [3]. The left panel shows the encoder architecture consisting of a multi-head self-attention module, and the right panel shows the decoder architecture with both having N = 6 stacked layers.

sequences, like processing entire books or lengthy documents, this reduces the model's efficiency.

Moreover, the paper lacked precise mathematical definitions and background explanations for key components, such as multi-head attention. A more formal treatment would enhance readability and comprehension. While the authors claimed that self-attention aids long-range dependencies due to its constant path length, empirical validation of this claim, particularly on long sentences, was missing. Despite these limitations, the architecture of the Transformer served as the foundation for later models such as GPT and BERT. For instance, BERT utilizes a bidirectional Transformer encoder to capture context from both directions, whereas GPT uses a unidirectional Transformer decoder for language modeling. The following section will discuss how these two models were developed based on Transformers.

III. Evolution of GPT & BERT

In 2018, the first OpenAI paper about GPT [7] proposed a semi-supervised method that used a single task-agnostic model to demonstrate improved performance on a wide range of tasks, including textual entailment, question answering, and semantic similarity text classification. In nine of the twelve tasks that they examined, the model markedly outperformed the state-of-the-art (at the time).

The study assumed that a sizable corpus of unlabeled text and many datasets containing hand-annotated training examples (the goal tasks) were available. The training process was divided into two phases. The unlabeled data was first subjected to a language modeling (LM) objective to determine the model's initial parameters. Then, to maximize the likelihood, a standard forward LM objective was used

$$L_1(\mathcal{U}) = \sum_i \log P(u_i|u_{i-k},...,ui-1;\Theta), \tag{2}$$

where $\mathcal{U} = u_1, u_2, ..., u_n$ was the unsupervised corpus of tokens, k was the context window size, and the conditional probability P was modeled using a network with parameters Θ. SGD was used to train the parameters, and the relevant supervised objective was then used to modify these parameters to a target task.

The fundamental component was a Transformer, more precisely, a Transformer decoder. According to their studies, transformers performed better than LSTMs because they were better at capturing long-term dependencies, which led to reliable transfer performance across a variety of workloads. The input context tokens were subjected to the multi-head self-attention. Position-wise feedforward layers were then used to generate an output probability distribution across the target tokens. Following model training using optimization L_1, the parameters were then modified to fit the supervised target task. The labeled dataset was $\mathcal{C}$, where each instance was a sequence of input tokens, $x^1, \ldots, x^m$, along with a label y. The inputs were passed through the pre-trained model to obtain the final transformer block's activation function h_l^m, which was then fed into an added linear output layer with parameters W_y to predict y.

$$P(y|x^1, \ldots, x^m) = softmax(h_l^m W_y). \tag{3}$$

The objective to be maximized was as follows

$$L_2(\mathcal{C}) = \sum_{x,y} \log P(y|x^1, \ldots, x^m) \tag{4}$$

As an additional goal to the fine-tuning, an LM objective improved the supervised model's generalization and accelerated its convergence. One way to express a final objective was mentioned as,

$$L_3(\mathcal{C}) = l_2\mathcal{C} + \lambda \cdot L_1\mathcal{C} \tag{5}$$

where λ was the weight. This paper had enormous repercussions. The Internet's vast amount of data suddenly became usable and helpful—not just for one particular task, but possibly for all of them. This paved the way for the development of large language models.

Released shortly after the GPT publication, the BERT model [8] is unique in that it uses a bidirectional transformer architecture that stacks encoders from the original transformer on top of one another, as seen in Figure 02. Unlike past models trained for particular language tasks, both BERT and GPT pre-trained their models semi-supervised on vast text datasets, such as BooksCorpus or Wikipedia, which contain over 3 billion words overall. After being refined on labeled datasets for NLP tasks, including named entity recognition, question answering, and sentiment analysis, BERT greatly exceeded prior state-of-the-art (SOTA) results in many well-known benchmarks.

BERT's training consists of two phases: pre-training and fine-tuning. Pre-training uses unlabeled data with two

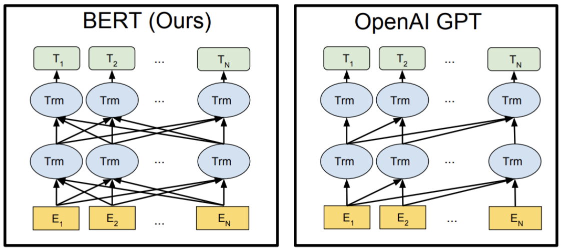
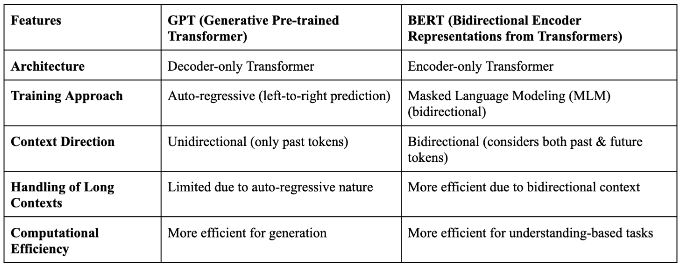


Fig. 2. Differences in pre-training model architectures from [8]. The left part shows the BERT model using a bidirectional Transformer, whereas the right part shows the OpenAI GPT using a left-to-right Transformer decoder architecture
.

tasks—Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In MLM, random tokens are replaced with a ”mask” token, and the model predicts them using a softmax over the vocabulary with cross-entropy loss. NSP is a binary task to learn sentence relationships. Fine-tuning adapts the pre-trained model to specific downstream tasks using labeled data, with each task having a separately fine-tuned model but sharing the same pre-trained parameters.



Features	**GPT (Generative Pre-trained Transformer)**	**BERT (Bidirectional Encoder Representations from Transformers)**
Architecture	Decoder-only Transformer	Encoder-only Transformer
Training Approach	Auto-regressive (left-to-right prediction)	Masked Language Modeling (MLM) (bidirectional)
Context Direction	Unidirectional (only past tokens)	Bidirectional (considers both past & future tokens)
Handling of Long Contexts	Limited due to auto-regressive nature	More efficient due to bidirectional context
Computational Efficiency	More efficient for generation	More efficient for understanding-based tasks

Fig. 3. Key Differences Between GPT and BERT in Architecture, Training, and Efficiency
.

Even though pre-training is beneficial, it wasn’t really helpful on its own. Rather, the goal of both BERT and GPT is to improve on certain linguistic problems. BERT’s architecture had several drawbacks. There is typically no straightforward method for training BERT for auto-regressive tasks predicting one token at a time during inference because it is bidirectional, and entering the target during training would result in target leakage. Smaller variations, such as DistillBERT [9], a speedier alternative, can be used in a smaller setting because BERT is a preset model, and its training is much more costly.

Some disadvantages of GPT are, even if it demonstrated clear improvements in language comprehension. One major criticism is that it is unidirectional, so restricting its ability to capture bidirectional context is one of BERT’s main benefits [8]. Moreover, GPT is computationally expensive and less fit for smaller-scale uses since it depends on large pretraining [3]. Furthermore, it has been questioned how well GPT performs on tasks requiring sophisticated thinking or long-range interdependence since it often generates believable yet factually erroneous or inconsistent solutions [10].

IV. Conclusion

Using self-attention to capture long-range dependencies and parallelized processing, the Transformer architecture has radically changed natural language processing. The field has evolved because models like GPT and BERT achieve SOTA results. These models have transformed conversational AI, powering virtual assistants and customer service chatbots that understand and generate human-like responses, improving user experience across various industries. However, they have some drawbacks: BERT has trouble with auto-regressive tasks, and GPT has trouble with bidirectional context. Furthermore, the dependence on hyperparameter tuning and computational complexity remains a problem. Notwithstanding these difficulties, the Transformer’s scalability and flexibility have opened the door for more extensive language models, such as GPT-3 [10], T5 [11], Vision Transformer (ViT) [10], etc. Addressing these constraints will be essential as NLP develops to create more effective and adaptable models that can manage linguistic tasks that get more complicated.

References

[1] Rumelhart, David E, Hinton, Geoffrey E, Williams, Ronald J (Sept. 1985). Learning internal representations by error propagation. Tech. Rep. ICS 8504. San Diego, California: Institute for Cognitive Science, University of California.

[2] S. Hochreiter and J. Schmidhuber, ”Long Short-Term Memory,” in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[3] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, Polosukhin, Illia *Attention is All you Need*. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS’17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[4] Britz, Denny, Goldie, Anna, Luong, Minh-Thang, Le, Quoc *Massive Exploration of Neural Machine Translation Architectures*. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, Sep 2017, 1442–1451.

[5] Transformers GitHub Repository: https://github.com/tensorflow/tensor2tensor

[6] Keles, Feyza Duman, Wijewardena, Pruthuvi Mahesakya, Hegde, Chinmay. *On The Computational Complexity of Self-Attention*. In: Proceedings of Machine Learning Research. 2023; Vol. 201. pp. 597-619

[7] Radford, Alec, Narasimhan, Karthik, Salimans, Tim, Sutskever, Ilya *Improving Language Understanding by Generative Pre-Training*. https://api.semanticscholar.org/CorpusID:49313245, 2018.

[8] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, Toutanova, Kristina *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), June 2019, Minneapolis, Minnesota, Association for Computational Linguistics, 4171–4186.

[9] Sanh, Victor, Debut, Lysandre, Chaumond, Julien, Wolf, Thomas*DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. https://api.semanticscholar.org/CorpusID:203626972, 2019

[10] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

[11] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140), 1-67.

[12] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ArXiv, abs/2010.11929.