Core-GPT Generatively Pre-Trained Transformer

Arundhati Warade, Aditi Agrawal, Kritika Tanwar

***Abstract*—There are several existing text summarization techniques available in today's world such as T5 used by Google, GPT3 etc. We propose a core-GPT, which is built from the scratch on the logic of a transformer and implemented with several combinations of Attention techniques to conclude the structure of the proposed transformer is better using simulations. We show that the complete transformer with complete attention layers performs better for a given chunk of data in terms of cross entropy loss.**

***Index Terms*—Text summarization, Attention, transformers, GPT.**

I. Introduction

Text summarization is a crucial process in natural language processing (NLP) that involves creating a condensed version of a longer text while retaining its essential information. This process can save time and effort in reading and understanding large amounts of text, and it has various applications in fields such as journalism, finance, and education. However, text summarization still faces challenges due to the complexity and ambiguity of natural language, as well as the need to ensure the generated summary is both informative and coherent.

Before Transformers [1], RNN [2], LSTM [2] were used for tasks like text summarization, NER [3], Question and Answering, Translation, Chatbot and other NLP tasks. However they are slower to train and still face the problem of vanishing gradient. Therefore the concept of Transformers was introduced [1].

The transformer architecture, which includes an encoder and decoder, has revolutionized NLP tasks such as text classification, machine translation, and text summarization. The self-attention mechanism in the transformer model is a core component that calculates the importance of each word in the input sequence based on its relationship with other words, while multi-head attention allows the model to attend to different aspects of the input sequence simultaneously. The Transformer model has been used in various NLP tasks and has achieved state-of-the-art results, outperforming previous models that relied on recurrent or convolutional layers. It has also been used to generate natural language text, such as chatbots and language models, due to its ability to attend to different parts of the input sequence, making it well-suited for generating coherent and contextually relevant responses. Overall, the transformer architecture and model have significantly improved the performance of text summarization and other NLP tasks.

A. Motivation

With increasing advantages of Transformer models over traditional RNNs and LSTMs we are motivated to build a transformer model.

First, Transformers are more efficient in processing long sequences of text compared to RNNs and CNNs. RNNs have a sequential nature, which makes it difficult to parallelize the computation and results in slow training and inference times for long sequences. CNNs, on the other hand, are limited by the fixed-length receptive field and may not be able to capture long-range dependencies. Transformers use self-attention mechanisms to attend to different parts of the input sequence, making them more effective in processing long sequences.

Second, Transformers have been shown to outperform RNNs and CNNs in various NLP tasks such as machine translation, text summarization, and question-answering. This is due to the ability of Transformers to capture global dependencies and relationships between different parts of the input sequence, which is important for understanding the context and generating accurate predictions. Finally, the pre-trained Transformer models such as GPT-2 have achieved state-of-the-art results in various natural language generation tasks, including text generation, dialogue systems, and language modeling. This makes them a powerful tool for researchers who want to build on top of pre-existing language models and fine-tune them for their specific use case. Overall, the efficiency, effectiveness, and versatility of Transformer models make them a popular choice for researchers in the field of NLP.

B. Novelty

The main novelty of the Transformer model compared to RNNs and CNNs is its ability to perform parallel processing of input sequences, allowing it to attend to all parts of the input sequence simultaneously. This is achieved through the use of the self-attention mechanism, which calculates the importance of each word in the input sequence based on its relationship with other words.

In contrast, RNNs process input sequences sequentially, which can lead to issues with long-term dependencies and make them slower to train. CNNs, on the other hand, are typically used for image processing and require fixed-length inputs, which can be limiting for natural language processing tasks where the length of input sequences can vary.

The Transformer model's ability to perform parallel processing and attend to different parts of the input sequence at the same time has led to significant improvements in various NLP tasks, including machine translation, text summarization, and question-answering.

II. Related Work

The Transformer model is a type of neural network architecture that has revolutionized the field of Natural Language Processing (NLP). It was introduced by Vaswani et al. in 2017 and has since become the state-of-the-art model for a variety of NLP tasks, including machine translation, question answering, and language generation. The Transformer model is based on the idea of self-attention, which allows it to capture long-range dependencies between words in a sentence. Unlike traditional recurrent neural networks (RNNs) and convolutional neural networks (CNNs), the Transformer model does not rely on sequential processing or fixed-length context windows. Instead, it can process entire sequences of words in parallel, making it much faster than other models.

Natural Language Processing (NLP) is a subfield of artificial intelligence that focuses on enabling computers to understand, interpret, and generate human language. It involves a range of techniques, including machine learning, deep learning, and linguistics, and has applications in areas such as chatbots, sentiment analysis, and speech recognition. One of the key challenges in NLP is dealing with the complexity and ambiguity of natural language. Words can have multiple meanings depending on the context, and grammar rules are often violated in everyday speech. This makes it difficult to develop algorithms that can accurately process and understand text.

Neural networks have been used in NLP for several decades, but early models were limited by their inability to capture long-term dependencies between words. Recurrent neural networks (RNNs) were developed to address this issue, but they suffer from vanishing gradient problems and are slow to train. Convolutional neural networks (CNNs) were also explored for NLP, but they are better suited for tasks that involve local patterns rather than global relationships between words. The Transformer model overcomes these limitations by using self-attention mechanisms to allow each word in a sequence to attend to all other words, regardless of their position in the sequence.

III. Methodology

A. Transformer

The Transformer is a deep learning model architecture introduced in 2017 by Vaswani et al. that has become widely used in natural language processing (NLP) tasks such as machine translation, language modeling, and text classification. The Transformer architecture is based on the use of self-attention mechanisms, which allows the model to selectively focus on different parts of the input sequence during processing.

The architecture consists of an encoder and a decoder, each composed of multiple layers of self-attention and feedforward neural networks. The encoder takes as input a sequence of tokens, and the self-attention mechanism computes a weighted sum of the embeddings of all the tokens in the sequence, where the weights are determined by the similarity between the tokens. The resulting output is then passed through a feedforward neural network to produce a new representation for the sequence.

The decoder takes the encoded input sequence and generates an output sequence by predicting the next token in the sequence based on the previously generated tokens. The self-attention mechanism is used to attend to the encoder output and the previous decoder output to generate a context vector for each step in the decoding process.

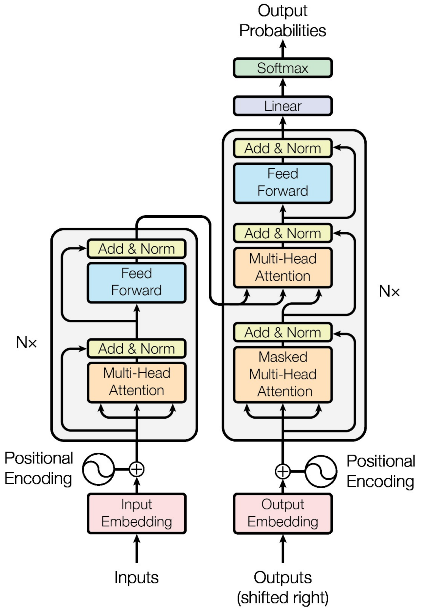

Fig. 1: Basic Transformer Model

The final Linear and Softmax layer in a Transformer model projects the vector produced by the decoder stack into a logits vector, which is then turned into probabilities by the Softmax layer. During training, the loss function compares the output probabilities with the desired probabilities to optimize the model's weights. The model produces the outputs one at a time, and two decoding methods are discussed: greedy decoding and beam search.

Overall, the Transformer architecture has proven to be highly effective for a wide range of NLP tasks and has become a foundational component of many state-of-the-art NLP models.

B. Attention Layers

Attention layers are inspired by human ideas of attention, but is fundamentally a weighted mean reduction. The attention layer takes in three inputs: the query, the values, and the keys. These inputs are often identical, where the query is one key and the keys and the values are equal.

1) Self Attention: Self-attention is achieved when the query, values, and the keys are equal. The input and output dimensions of a self-attention layer are always the same.

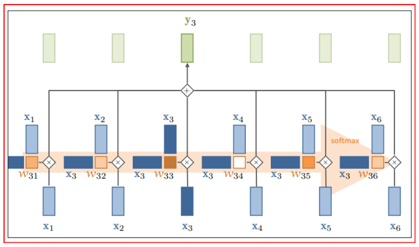

Fig. 2: Self Attention

2) Multi Head Attention: Multi-head attention block is a group of layers that splits to multiple parallel attentions.

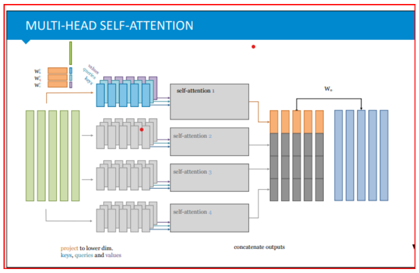

Fig. 3: Multi Head Attention

3) Masked Multi-Head Attention: We compute the attention weights, we mask out any attention from the current token to future tokens in the sequence. We need to set the raw attention weights to negative infinity, so that after the softmax operation, they become 0.

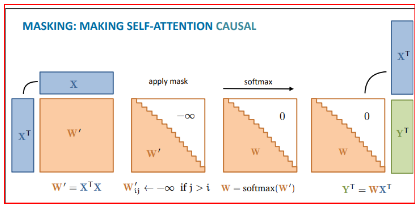

Fig. 4: Masked Multi Head Attention

C. Softmax

The Softmax output function transforms a previous layer's output into a vector of probabilities. It is commonly used for multiclass classification.

$$SoftMax\left(\frac{QK^2}{\sqrt{d_k}}\right)V$$

D. Encodings

Encodings [4] are different vectorization techniques that can represent text into numerical form to the model.

E. LayerNorm

Layer normalization (LayerNorm) [5] is a type of normalization technique used in deep neural networks. It is similar to batch normalization, but instead of normalizing over the batch dimension, it normalizes over the features dimension.

F. Dropout

Dropout [6] is a technique to deal with overfitting by combining the predictions of many different large neural nets at test time.The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much. During training, dropout samples from an exponential number of different "thinned" networks. At test time, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single unthinned network that has smaller weights.

G. Dataset

The Harry Potter novel text file is used as input for summarization and compare its performance with state-of-the-art methods.

IV. Results

We tested different combination of Attention models(Self Attention, Multi Attention, Masked Multi Attention) with Feed Forward, LayerNorm, Sequential Model, Dropout, Linear Transformation and computed the cross entropy loss of the models.

Model	**Training Loss**	**Validation Loss**
Bigram	2.4738	2.4911
Self Attention	2.3980	2.4084
Masked Multi Head Attention	2.2748	2.2858
Self Attention + Feed Forward	2.2286	2.412
Masked Multi Head + Feed Forward	1.9993	2.0808
Layer Norm + Feed Foward + Multi Head	1.9829	2.0607
Scaling Transformer	1.0763	1.4873

TABLE I: Cross Entropy Loss for Combination of Models

From Table 1. it is evident that cross entropy loss with different combination of models have different results.

V. Conclusion

In this paper we have proposed the methodology for text summarization using transformer model from scratch. After applying all the attention levels one by one and computing the loss for each one them. the results are found that after applying all the layers the loss obtained was minimum and was the best result of all the layers of attention and feed forwards applied.

VI. Future Scope

In future work, we plan to explore the use of word to word generation for other natural language processing task, such as Text summarization done in this paper. We also plan to investigate the use of word sense disambiguation for the detection and removal of offensive content which is given by our GPT.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 06 2017.

[2] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, mar 2020. [Online]. Available: https://doi.org/10.1016%2Fj.physd.2019.132306

[3] B. Mohit, *Named Entity Recognition*, 03 2014, pp. 221–245.

[4] M. K. Dahouda and I. Joe, "A deep-learned embedding technique for categorical features encoding," *IEEE Access*, vol. 9, pp. 114 381–114 391, 2021.

[5] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, p. 1929–1958, jan 2014.