Transformer in Deep Learning

唐梓烨	李自航	刘睿哲	蒋一凡
58122310	09022213	58122309	57122208

***Abstract*—The Transformer model is a revolutionary neural network architecture that has achieved remarkable success in various fields, including natural language processing (NLP) and computer vision (CV). This paper aims to introduce the structure of the Transformer model and its applications in the fields of NLP and CV. It delves into the commonalities and differences of the Transformer model across different domains. We first provide a detailed overview of the core structure of the Transformer model, followed by discussions on Transformer variants in NLP tasks (such as BERT and GPT). Finally, we explore the emerging applications of the Transformer model in computer vision (e.g., Vision Transformer). Through this paper, readers will gain a deep understanding of the working principles of the Transformer model and its advantages and improvements in various domains.**

***Index Terms*—Transformer, Self-Attention, Bert, GPT, ViT**

I. Introduction

Natural language processing (NLP) and computer vision (CV) have always been two key domains in the field of artificial intelligence. The development of these domains has long been constrained by the performance and capabilities of models, and the emergence of the Transformer model has disrupted the dominance of RNN and CNN. In recent years, the Transformer model has become one of the preferred models for NLP and CV tasks, demonstrating a momentum towards unifying research approaches in both NLP and CV. First, we will provide a detailed introduction to the basic structure of the Transformer model, including self-attention mechanism. This mechanism enables the model to process individual elements in the input sequence simultaneously, whether they are words, pixels, or other forms of data. The strength of this idea lies in its generality, allowing the Transformer to succeed in various domains. In NLP, the Transformer model has achieved tremendous success in tasks such as text classification, named entity recognition, and machine translation. In the field of computer vision, the Transformer has also made its mark in image classification and object detection. However, while the application of the Transformer model spans across different domains, it comes with unique challenges and characteristics in each domain. For instance, in NLP, the BERT model improves pre-trained word embeddings by allowing the model to obtain bidirectional input and grasp contextual information. In CV, the ViT model tackles the complexity of image data by partitioning the images into blocks. Through an in-depth exploration of these similarities and differences, we can better understand the multi-domain applicability of the Transformer model and how to elevate it to new heights.

II. transformer is powerful

A. what is transformer? [1]

Please see the figure 1. Transformer is a neural network model based on the attention mechanism. The Transformer model consists of an encoder and a decoder, both of which are composed of multiple identical layers. Each layer has two sub-layers: multi-head self-attention mechanism and fully connected feed-forward network. The Transformer model captures the relationship between each word in the input sequence and the target sequence by using self-attention mechanism, and improves the performance of the model by using multi-head mechanism. I will now introduce the transformer model from input to output in a bottom-up manner.

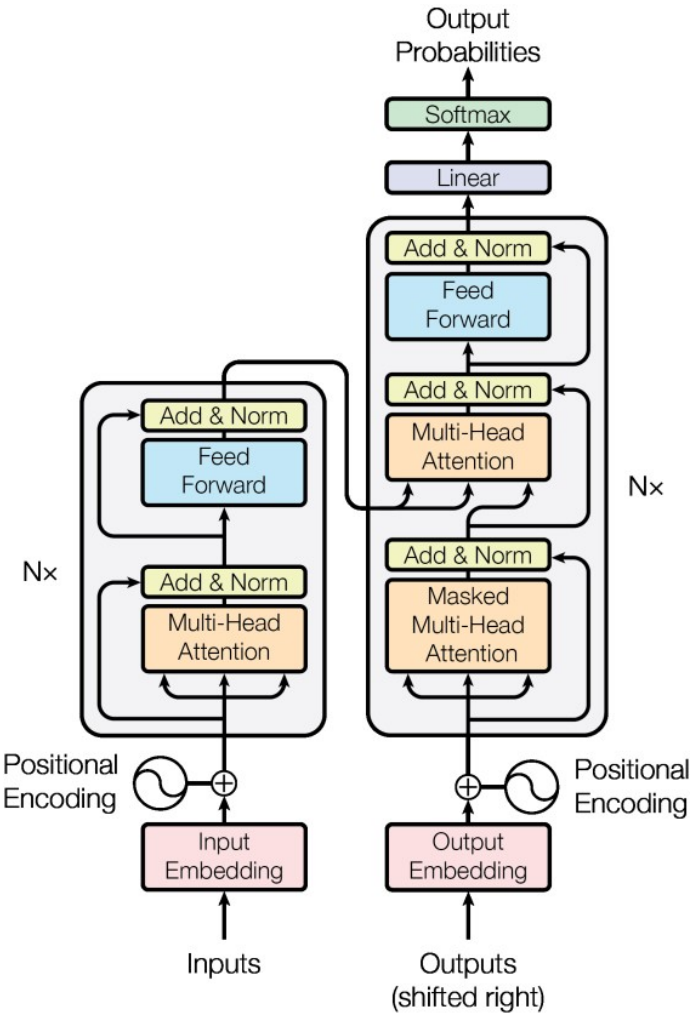


Figure 1: Transformer

1) Input Embedding and Output Embedding: Word Embedding is the process of converting discrete symbolic sequences, such as words or characters, into continuous vector representations, allowing neural networks to effectively process textual data. It enables the model to understand the semantics and contextual information of textual data, providing input for subsequent self-attention layers and feedforward neural networks.

2) Positional Encoding: Since the Transformer model does not include recurrent or convolutional operations for processing input sequences, it cannot inherently grasp the order information of words. Therefore, it is necessary to inject information about the relative or absolute positions of tokens into the sequence. To achieve this, we introduce "positional encoding" to embed positional information into the model. Positional encoding is a matrix related to both position and dimensions, and it is added to the input word embeddings. In the original paper, the authors used sinusoidal functions to perform positional encoding.

$$PE_{(pos,2i)} = \sin(pos/10000_{2i/dmodel})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000_{2i/dmodel})$$

where pos is the position and i is the dimension. That is, each dimension of the positional encoding corresponds to a sinusoid. The wavelengths form a geometric progression from 2π to $10000 \cdot 2\pi$. We chose this function because we hypothesized it would allow the model to easily learn to attend by relative positions, since for any fixed offset *k*, PE_{pos+k} can be represented as a linear function of PE_{pos}. [1]

B. Attention

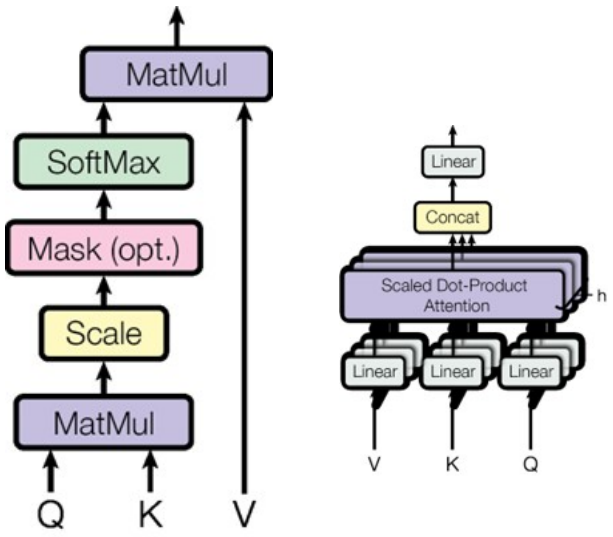
Please see the figure 2. An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

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

1) Scaled Dot-Product Attention: Self-attention mechanism is a crucial component of the Transformer model, allowing it to establish weighted connections between different positions in a sequence, enabling it to consider information from all positions simultaneously in a single step. The computation of self-attention involves the following steps:

Q, K, V: For an input sequence, first, query, key, and value vectors are obtained through three linear transformations. These vectors are used to calculate attention weights and generate output.

Attention Weights: To measure the degree of association between each position and other positions, attention scores are calculated by taking the dot product between the



(a) figure 1 (b) figure 2

Figure 2: Attention

query and key, followed by scaling (usually with a scaling factor, such as $\sqrt{d_k}$). This results in attention scores.

Normalization of Attention Scores: The softmax function is applied to convert attention scores into a probability distribution, ensuring that each position's contribution to other positions is weighted to sum to 1, forming normalized attention weights.

Weighted Sum (MatMul): The attention weights are applied to the value vectors, and then all weighted values are summed to produce the final output. This output contains information from all positions, with each position's contribution determined by its degree of association with other positions.

Self-attention mechanism allows the model to dynamically allocate weights to different positions, making it flexible in capturing contextual information for various tasks. This is one of the key reasons why the Transformer model excels in a wide range of natural language processing tasks.

2) Multi-Head Attention: The multi-head attention mechanism in the Transformer is an extension of self-attention, enabling the model to learn self-attention in different representation subspaces. This mechanism enhances the model's representational capacity, allowing it to simultaneously focus on different aspects of the input sequence, better capturing complex relationships within the sequence. Multi-head attention is achieved by combining multiple parallel self-attention mechanisms, with each self-attention mechanism referred to as a "head." Each head has its set of query, key, and value parameters, learned through training. The outputs of all attention heads are concatenated and transformed through a linear transformation to produce the final multi-head attention output. Its advantages include improved learning capabilities: different heads can learn to capture different relationships, enhancing the model's representational capacity. For example, some heads may focus on syntactic relationships, while others may focus on semantic relationships. It also improves training efficiency: multi-head attention can be computed in parallel because each head is independent, enhancing both training and inference efficiency. Lastly, it enhances model stability: multi-head attention helps alleviate some instabilities in attention mechanisms, as the combination of multiple heads can smooth the distribution of attention weights.

C. Add & Norm

Transformer introduces residual connections and layer normalization between the input and output of each sub-layer (self-attention layer and feedforward neural network layer). Residual connections allow the flow of information by skipping a certain amount of information through a layer, facilitating the direct flow of gradients. Layer normalization transforms the data to have a mean of 0 and a variance of 1, ensuring the stability of data feature distributions. These techniques help mitigate the vanishing gradient problem during training.

D. Feed-Forward Neural Network

Each encoder and decoder layer in the Transformer includes a feed-forward neural network. It maps the output of the self-attention layer to a higher-dimensional space and then maps it back to the original dimension. This process involves two linear transformations and a nonlinear activation function, typically RelU. Its role is to map data to a high-dimensional space and then back to a lower-dimensional space, extracting deeper-level features through linear transformations. And add some non-linear transform to enhance the model's learning capacity.

E. Masked Multi-Head Attention

Masked Multi-Head Attention is a crucial technique for handling variable-length sequences. Its primary role is to ensure that the model only attends to information before the current position and prevents information leakage from future positions during self-attention calculations. When dealing with sequence data, especially in decoders, a masking mechanism is used to ensure that the information generated at each position depends only on the current and preceding positions. This is because future information is not visible when generating an output sequence. Specifically, the masking mechanism creates a mask matrix, which is multiplied with attention scores, setting the scores for future positions to negative infinity (or zero after softmax), effectively eliminating the influence of future positions in attention calculations. This allows the model to focus solely on the current and past positions.

F. Why Transformer

Compared to RNN and CNN, the Transformer offers several advantages:

Better Parallelism: The self-attention mechanism in Transformer can be computed in parallel, while RNN and CNN with their recurrent and convolutional operations are inherently sequential and harder to parallelize. For example, RNN require capturing temporal dependencies in sequences, and CNN require gradually increasing the receptive-field from small to large regions.

Better Long-Distance Dependency Modeling: The self-attention mechanism in Transformer can capture relationships between any two positions in an input sequence, while RNN can only capture relationships between adjacent positions, and CNN are limited to local relationships.

Better Global Feature Extraction: The self-attention mechanism in Transformer allows attention to be calculated for every position in the input sequence, capturing global context information, while RNN and CNN are limited to capturing local context information. Given these advantages that are better than the former models, Transformer quickly gained popularity and found applications in various domains, including NLP and CV.

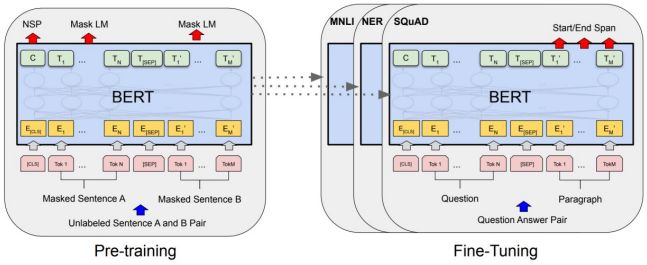


Figure 3: Bert

III. Transformer in NLP

When Transformer was first introduced, it was primarily used for machine translation. It addressed the shortcomings of RNN in terms of parallelism and handling long texts, resulting in significant improvements in translation accuracy. As researchers recognized the advantages of the Transformer and attention mechanisms in NLP, they sought to apply the Transformer to various fields, such as text classification, semantic analysis, and language generation. They combined the Transformer with pre-training, training a model and then fine-tuning it for different tasks. This approach reduced training time and data requirements for specific tasks, reducing the risk of overfitting and improving model accuracy and reliability. Consequently, in NLP, pre-trained Transformers became widely adopted.

A. Bert:Bidirectional Encoder Representation from Transformers [2]

Please see the figure 3. Bert is a pre-trained language representation model that primarily utilizes the Transformer's Encoder module. During pre-training, it employs the Masked Language Model (MLM) and Next Sentence Prediction (NSP) methods, emphasizing bidirectional information acquisition and leveraging contextual information to enhance the model's understanding. Now, let's discuss some of the improvements made to Bert. Bert is a pre-trained language representation model that primarily utilizes the Transformer's Encoder module. During pre-training, it employs the Masked Language Model (MLM) and Next Sentence Prediction (NSP) methods, emphasizing bidirectional information acquisition and leveraging con-

textual information to enhance the model's understanding. Now, let's discuss some of the improvements made to Bert.

1) Encoder-Only: Bert serves as a feature extractor, primarily focusing on self-supervised pre-training to learn deep semantic information from text. Therefore, it retains only the Encoder part of the Transformer. Leveraging the Encoder's ability not to mask information from later positions, Bert enables the model to learn bidirectional textual information, making predictions based on both preceding and subsequent context.

2) Masked Language Model (MLM): MLM is one of Bert's pre-training tasks. In the MLM task, a portion of the input text is randomly masked, and the model's objective is to predict these masked words based on context. Initially, 15% of the words in the input text are randomly replaced with a special "[MASK]" token to indicate that they are masked. Then, a Transformer-based encoder processes this masked text, generating hidden state vectors for each position. Finally, these hidden state vectors are fed into a softmax function to produce a probability distribution over all words in the vocabulary, enabling the model to predict the masked words. The MLM task allows Bert to process text bidirectionally, unlike previous language models that operated strictly in a left-to-right or right-to-left manner. This has proven helpful for various downstream tasks, such as machine reading comprehension, natural language inference, and question-answering systems.

3) Next Sentence Prediction (NSP): NSP is another pre-training task in Bert. In the NSP task, the model is given two sentences, and it needs to determine whether these two sentences are contextually related. Initially, two random sentences are selected from a text corpus, with the first sentence labeled as "A" and the second as "B." A special separator token "[SEP]" is added between the two sentences, and a special token "[CLS]" is added at the beginning of the input sequence. A Transformer-based encoder then processes this input sequence to generate hidden state vectors for each position. Finally, a binary classifier is used to predict whether sentence "B" is the next sentence in the original document, given sentence "A." During training, 50% of the sentence pairs are from the original document, while the other 50% are randomly composed unrelated sentence pairs [2]. NSP helps Bert capture the context between sentences, which is beneficial for various downstream tasks.

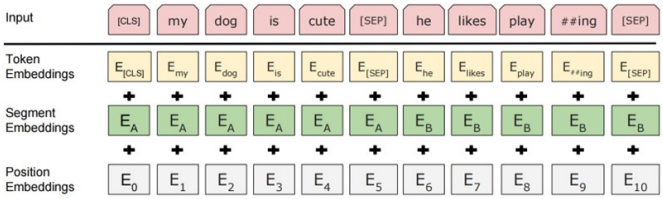


Figure 4: Embeddings

4) Improve of input embeddings: Please see the figure 4. Bert's input embeddings have been improved compared to the Transformer model. It no longer relies on a fixed formula for Position Embeddings; instead, it allows the model to learn how to encode positions dynamically. Additionally, Bert introduces Segment Embeddings, which are used to distinguish between two different sentences, aligning with its pretraining tasks. Bert also innovatively incorporates special tokens in its input embeddings, such as [CLS] and [SEP]. The output after passing through the self-attention mechanism for [CLS] represents the model's attention over the entire sentence, i.e., the model's feature extraction for the sentence. Due to the unique nature of attention mechanisms, placing [CLS] at any position within the sentence yields the same effect. On the other hand, [SEP] is used to separate different sentences. These special characters hold specific meanings within the model, and this approach has been adopted by subsequent models like ViT [3] to enhance their understanding capabilities.

5) Fine-Tuning: As mentioned earlier, Bert is fundamentally a pre-trained feature extractor. Its main purpose is self-supervised pre-training to learn deep semantic information from text. To apply these learned sentence features to various natural language processing tasks, fine-tuning is required. This involves selecting specific objective functions, deciding which parts of sentence features to use as inputs for downstream tasks, and fine-tuning the model's parameters accordingly. Figure 3 illustrates how Bert can be fine-tuned for a question-answering task on the SQuAD

dataset [2].

6) The Contribution of Bert: The contribution of the Bert model lies in its introduction of the pre-training and fine-tuning paradigm and the pioneering of the bidirectional semantic understanding pre-training task. Bert only requires unsupervised training on a large amount of unlabeled data, followed by model transfer and supervised fine-tuning for various tasks. This significantly reduces the training cost for specific tasks and the need for labeled data. The bidirectional semantic understanding pre-training task greatly improves the model's ability to combine context. This makes Bert conceptually simple and empirically powerful, achieving state-of-the-art results on eleven natural language processing tasks [2].

B. GPT: Generative Pre-trained Transformer

GPT is a generative pre-trained language model based on the Transformer architecture. It can generate coherent and natural text based on given input and is applicable to various natural language processing tasks, including machine translation, text summarization, question-answering systems, and more. The core idea of GPT is to leverage a large amount of unlabeled text data for unsupervised pre-training, learning a general representation of text, followed by supervised fine-tuning on specific tasks to adapt the model parameters. GPT has multiple versions, such as GPT-1 [4], GPT-2 [5], GPT-3 [6], each with an increase in model parameters and training data, enhancing generation capabilities and generalization. Now, let's discuss some characteristics of GPT-1 to GPT-3 based on their respective research papers.

1) Decoder only: Please see the figure 5.GPT utilizes only the Decoder part of the Transformer, omitting the Encoder part. This is because GPT's objective is language modeling, predicting the next word based on the previous context. Language modeling requires attention only to the text generated before and doesn't require considering the entire input sequence. Therefore, GPT employs masked self-attention layers to achieve this functionality without using encoder-decoder attention layers [4]. Additionally, GPT removes the multi-head self-attention layers originally

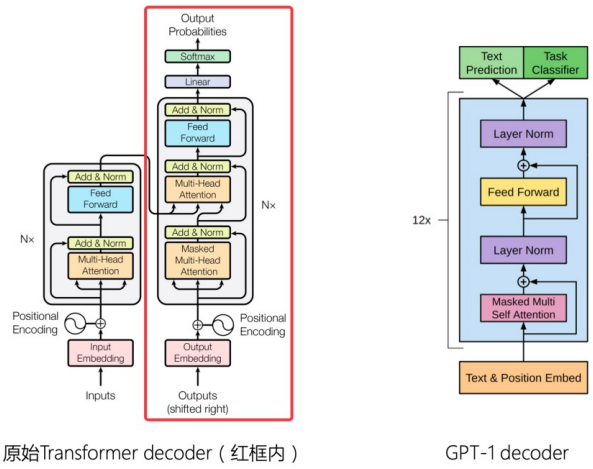


Figure 5: Decoder Only

designed to receive Encoder outputs, as GPT lacks an Encoder part. A structural comparison between GPT and the Transformer Decoder is shown in Figure 5.

2) GPT-2 can be a Zero-Shot Learning Language Model: With the continuous increase in model size and training parameters, GPT-2 achieves zero-shot learning capabilities. It can perform downstream tasks without any annotated data or parameter tuning directly. This is distinct from models like BERT, which require supervised fine-tuning after pretraining for specific tasks. When GPT-2 is tasked with a zero-shot downstream task, it simply generates corresponding text based on task descriptions and examples [5]. For instance, for a machine translation task, you can input "Translate from English to French: Hello, how are you?" and it will output "Bonjour, comment allez-vous?". For a question-answering task, inputting "Who is the president of the U.S.?" results in "Joseph Robinette Biden Jr.". These inputs can be seen as prompts [7], instructing the model on the task without the need for further supervised training, enhancing both performance and generalization. GPT-2 achieves this through pretraining on a vast amount of unlabeled text data, learning general language knowledge and patterns. These data include information related to various tasks such as syntax, logic, and common knowledge, enabling the model to understand natural language semantics. GPT-2 then employs an autoregressive approach to predict the next word from left to

right, facilitating language generation and output.

3) Emergence in Particularly Large GPT-3: Following GPT-2, OpenAI further increases model size and training parameters, resulting in a qualitative leap in model performance through quantitative changes. Notably, accuracy and generalization improve significantly, allowing the model to perform various downstream tasks with minimal examples. The reason for this "emergence" is yet to be fully understood, but it is speculated that larger models possess stronger memory and reasoning capabilities, enabling them to learn more knowledge and patterns from massive datasets. In this paper, GPT-3 achieves remarkable performance with 175 billion parameters, setting new records in the NLP field [6]. The paper also suggests that GPT models have not yet reached a point of diminishing returns with increasing parameter size [6], indicating that further increases in parameters and raw data could lead to improved performance.

4) Where to get so much text data: GPT requires a substantial amount of text data for training, making the acquisition of high-quality text data a notable research area. OpenAI used data from the foreign forum Reddit in GPT-2 and leveraged Reddit's upvote mechanism to filter high-quality data [5]. For GPT-3, training data volume increased by a factor of 1,125, reaching 45 terabytes, equivalent to 10% of all text on the internet. GPT-3 collected data from Common Crawl, a massive web dataset, and applied filtration by utilizing GPT-2's pre-filtered high-quality text as positive examples to train an LR classifier for data selection. Further steps, such as deduplication, were employed to obtain a high-quality dataset [6].

C. The Advantages of Pre-trained Transformers in NLP

As mentioned earlier, Transformer-based pre-trained models can effectively leverage vast amounts of unlabeled text data through self-supervised learning to acquire universal language knowledge and representations. This enhances performance and generalization in downstream tasks. Additionally, pre-trained Transformer models exhibit scalability and flexibility, enabling adjustments in model structure and parameters according to specific task requirements,

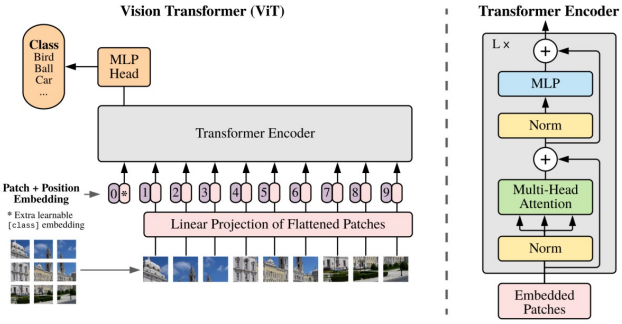


Figure 6: ViT

facilitating various functions and applications. Pre-trained Transformer models are currently one of the most advanced and popular technologies in the NLP field, demonstrating significant potential and prospects in machine translation, text classification, reading comprehension, dialogue generation, text summarization, and more.

IV. Transformer in Computer Vision (CV)

In the field of computer vision (CV), Transformers have also made significant contributions. Initially, researchers combined attention mechanisms from Transformers with convolutional neural networks (CNN) to improve feature extraction in CNN. Later, the Vision Transformer (ViT) introduced a pure Transformer-based structure for pretraining, and models trained using this architecture also demonstrated remarkable performance. [3]

A. Enter the image into the Transformer

Please see the figure 6. How to convert two-dimensional images into one-dimensional sequences for input into the transformer model is the first step to make use of CV. This paper presents an ingenious approach This is achieved by dividing the image into individual patches and arranging them into a one-dimensional sequence, mimicking the process of feeding a sentence into a transformer model in natural language processing. This is also the reason behind the statement in the paper that "AN IMAGE IS WORTH 16X16 WORDS." [3] The decision to avoid flattening each pixel of the image into a one-dimensional sequence was made to prevent excessively long sequences,

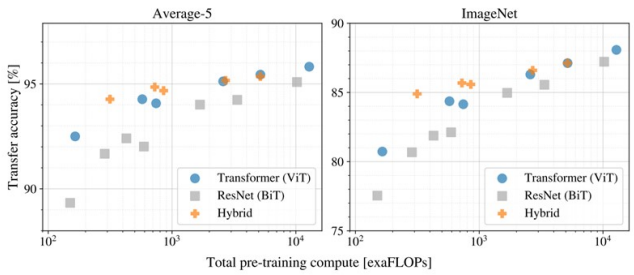


Figure 7: Compared

which would have increased the complexity of model training.

B. Patch and Position Embedding

Segmenting the image into patches inevitably results in the loss of certain positional information. However, the transformer model itself provides a solution through Position Embedding, allowing the Patch Embedding vectors to incorporate positional information about each patch's location within the image. The ViT model primarily draws inspiration from the Bert model in terms of the specific transformation method for Position Embedding, leaving it to the model to autonomously learn this aspect.

C. Extra Learnable [Class] Embedding

In ViT, the authors introduce an additional learnable embedding vector used to represent the category information for the entire image, denoted as "0*" in figure 6 [3]. Its purpose is to capture global features from the embeddings of other image blocks, serving as the representation vector for the image. This is akin to the [CLS] token in the Bert model, both of which are designed to capture information about the entire sequence.

D. Compared with CNN

Please see the Figure 7. Compared to Convolutional Neural Networks (CNN), ViT exhibits several common advantages, such as the ability to leverage unlabeled data, enhanced parallelism, stronger generalization, and improved robustness. Some of these advantages are inherited from the transformer model, as previously mentioned. However, ViT is not without its limitations. When dealing with relatively small training datasets, ViT often performs worse than equivalently sized ResNets. This is because CNN have already matured in the field of computer vision, making it challenging for transformers to utilize certain CNN-specific prior knowledge, such as locality/two-dimensional neighborhood structure and translation equivariance. Nevertheless, as datasets continue to expand, ViT surpasses ResNets in performance across multiple tasks. The paper also discusses a hybrid approach that combines ResNets and transformers during training, specifically by using ResNets for image embedding, where it outperforms both ViT and ResNets on smaller datasets but falls behind ViT on larger datasets, as shown in the Figure7 [3]. This indicates that transformers are gaining momentum as a replacement for CNN in the computer vision domain.

V. Conclusion and Future Work

A. Conclusion

After extensive research and reading numerous papers on transformers, it becomes evident that transformers have continually expanded their scope across various domains. Starting from their inception in machine translation, they have traversed through the entire field of Natural Language Processing (NLP) and extended their reach into Computer Vision (CV) and other areas of deep learning. Transformers are now applied in diverse tasks, either in conjunction with traditional Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) or by pioneering new methods, offering a wider array of choices for solving different tasks in different domains. On a deeper level, over time, transformer models have undergone continuous refinement and adaptation to suit various tasks. This includes the expansion of data volume, an increase in model parameters, and the incorporation of pre-training and fine-tuning principles into transformers. These studies collectively enhance the learning and generalization capabilities of transformer models.

Below, I outline the notable successes of transformer models in various domains:

- Natural Language Processing (NLP): Transformers and their variants have been extensively explored and applied in NLP tasks, such as machine translation,

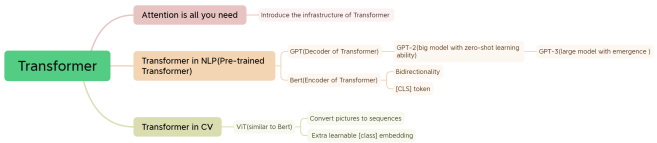


Figure 8: Mind Map

language modeling, and named entity recognition. Significant efforts have been directed towards pre-training transformer models on large-scale text corpora, which we believe is a major contributing factor to their widespread adoption in NLP.

- Computer Vision: Transformers are also applicable to a range of visual tasks, including image classification, object detection, image generation [8], and video processing.
- Audio Applications: Transformers can be extended to applications related to audio, such as speech recognition, speech synthesis, audio enhancement, and music generation.
- Multimodal Applications: Due to their flexible architecture, transformers are being applied in various multimodal scenarios, including visual question answering, visual commonsense reasoning, caption generation, speech-to-text translation, and text-to-image generation. [9])

To compile this report, our team extensively reviewed multiple papers, with a specific focus on Attention is all you need(Transformer) [1],BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding(Bert) [2], Improving Language Understanding by Generative Pre-Training(GPT-1) [4], Language Models are Unsupervised Multitask Learners(GPT-2) [5],Language Models are Few-Shot Learners(GPT-3) [6],AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE(ViT) [3]. Below, I will provide a brief overview of the connections between these papers and the overarching themes of our report through a mind map (8) and some tables (I & II).

Table I: The Difference Between Transformer, Bert, GPT

Model	Network Structure	Pre-training Task	Application Domain
Transformer	Encoder-decoder structure, using self-attention mechanism for sequence modeling	None, it is a general neural network architecture that can be used for different tasks	Machine translation, text generation
Bert	Only uses the encoder part, using bidirectional self-attention mechanism	Masked language model and next sentence prediction	Natural language understanding, such as question answering, text classification, sentiment analysis, etc.
GPT	Only uses the decoder part, using unidirectional self-attention mechanism	Autoregressive language model	Natural language generation, such as text completion, dialogue generation, summarization, etc.

B. Future Work

The research and applications of Transformer models are still evolving, with new developments constantly emerging. For instance, in the field of Computer Vision (CV), the Swin Transformer model [10], based on the ViT model, has introduced innovations. Its main feature is the use of a sliding window mechanism to divide the image into multiple small blocks and then perform self-attention calculations within each block. This approach reduces computational complexity while preserving local information. In addition to this, Transformer has bridged the gap between NLP and CV research, offering hope for multimodal applications. Today, we can already witness the use of various Transformer-based multimodal applications, such as generating code, images, videos, music, and more from natural language descriptions. [11] These applications greatly promote the development of Artificial General Intelligence (AGI) and Artificial General Creative Intelligence (AIGC). I believe that Transformer is a significant breakthrough on the path to achieving strong artificial

Table II: The Similarities And Differences Between Bert And GPT

Aspect	BERT	ViT
Network structure	Only uses the encoder part of the transformer, using bidirectional self-attention mechanism	Uses the same transformer architecture as BERT, but applies it to image patches instead of tokens
Pre-training task	Masked language model and next sentence prediction	Image classification with a learnable class embedding
Similarity [CLS] and Extra Learnable [Class]	Uses a special token [CLS] as the first token of the input sequence, which is used for classification tasks	Uses an extra learnable embedding vector as the first element of the sequence, which is used for image classification
Application domain	Natural language understanding, such as question answering, text classification, sentiment analysis, etc.	Computer vision, such as image recognition, object detection, segmentation, etc.

intelligence. For instance, the powerful capabilities of GPT-4 raise questions about whether it possesses consciousness and understanding.

Therefore, in future research, there are several directions to explore further improvements in Transformer models:

- Theoretical Analysis: Conducting a thorough theoretical analysis of Transformer models to understand and explain their exceptional performance, shedding light on the underlying principles that drive their success.
- Interpretability: Addressing the "black-box" nature of Transformers by developing methods to interpret and explain the learning and generation processes within these models. This will enhance our ability to trust and use them in critical applications. [12]
- Intra- and Cross-Modal Attention: Enhancing the design of intra- and cross-modal attention mechanisms in Transformers to better capture and process information across different modalities, paving the way for more versatile multimodal applications.)

The directions I mentioned above are just a few ideas, and I believe that with the efforts of numerous researchers, Transformer models will continue to evolve and find applications in an even broader range of domains. This ongoing development will ultimately lead to improvements that enhance our daily lives and contribute to the advancement of AI.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[4] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[6] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[7] B. McCann, N. S. Keskar, C. Xiong, and R. Socher, "The natural language decathlon: Multitask learning as question answering," *arXiv preprint arXiv:1806.08730*, 2018.

[8] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *International conference on machine learning*. PMLR, 2018, pp. 4055–4064.

[9] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022.

[10] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[12] H. Chefer, S. Gur, and L. Wolf, "Transformer interpretability beyond attention visualization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 782–791.