applied sciences

MDPI

Review

A Historical Survey of Advances in Transformer Architectures

Ali Reza Sajun *, Imran Zualkernan and Donthi Sankalpa

Computer Science and Engineering Department, American University of Sharjah, Sharjah P.O. Box 26666, United Arab Emirates; izualkernan@aus.edu (I.Z.); dsankalpa@aus.edu (D.S.)
* Correspondence: b00068908@aus.edu

Abstract: In recent times, transformer-based deep learning models have risen in prominence in the field of machine learning for a variety of tasks such as computer vision and text generation. Given this increased interest, a historical outlook at the development and rapid progression of transformer-based models becomes imperative in order to gain an understanding of the rise of this key architecture. This paper presents a survey of key works related to the early development and implementation of transformer models in various domains such as generative deep learning and as backbones of large language models. Previous works are classified based on their historical approaches, followed by key works in the domain of text-based applications, image-based applications, and miscellaneous applications. A quantitative and qualitative analysis of the various approaches is presented. Additionally, recent directions of transformer-related research such as those in the biomedical and timeseries domains are discussed. Finally, future research opportunities, especially regarding the multi-modality and optimization of the transformer training process, are identified.

Keywords: transformers; deep learning; generative deep learning; large language models; GPT; computer vision

Citation: Sajun, A.R.; Zualkernan, I.; Sankalpa, D. A Historical Survey of Advances in Transformer Architectures. *Appl. Sci.* **2024**, *14*, 4316. https://doi.org/10.3390/app14104316

Academic Editors: Andrea Prati, Dongpo Xu, Huisheng Zhang and Jie Yang

Received: 19 March 2024
Revised: 21 April 2024
Accepted: 15 May 2024
Published: 20 May 2024

1. Introduction

Ever since the introduction of the transformer model in June 2017 by Vaswani et al. [1], the world of deep learning has seen a rapid adaptation of the model in pushing the state of the art in a number of previously challenging tasks. Due to its prowess in sequence modeling and machine translation, the transformer architecture was initially widely implemented and indeed emerged as the predominant deep learning model for natural language processing (NLP) and generative deep-learning tasks [2]. Indeed, the introduction of transformers has been a key factor in the development of large language models such as GPT3 and GPT4, which are the basis of culturally significant tools such as ChatGPT [3]. However, inspired by the revolutionary self-attention mechanism in transformers, the architecture has since been implemented in various application domains such as that of images, audio, and time series data [4]. Indeed, in recent times, transformers have been touted as being a potential replacement for Convolutional Neural Networks (CNNs) for vision applications [5], with the introduction of the Vision Transformer (ViT) opening a new realm of architectures which build upon it. Considering the rapid increase in interest in transformer architecture, it becomes pertinent to examine in detail the architecture of the transformer as well as its historical progression from being introduced as an alternative to RNN-like architectures for sequence-to-sequence mapping to being one of the most impactful architectures in the current realm of deep learning. Finally, it may be beneficial to examine the various prevalent transformer architectures applicable to the different data domains.

Prior to the introduction of transformers to the deep learning space, the established state of the art in sequence modeling had long been Long Short-term Memory (LSTMs) [6] and other forms of Recurrent Neural Networks (RNNs) [7]. These were especially prevalent for transduction problems such as language modeling and machine translation due to their recurrence which allows for recent information to be accounted for in order to maintain

sequential information [1]. However, these established models had numerous drawbacks, particularly that the sequential computation involved in the training process prevents parallelization, therefore leading to slower training times [8] in cases of long sentences as they would be processed word by word. Furthermore, RNNs were susceptible to encoder–decoder bottlenecks particularly in sequence-to-sequence tasks because the encoder had to read the entire sequence before developing a hidden state of fixed length which the decoder then decoded [9]. Transformers emerged as an ideal solution to these drawbacks thanks to the self-attention mechanism which disregards the distance between words or output sequences when accounting for dependencies [10], which further allows for parallelization and therefore faster training. The following sections conduct an in-depth outlook at the initial architecture of early transformers. This gives insight into what makes transformers as unique as they are and what features of this architecture contribute to the large success seen by this kind of model.

1.1. Transformers

In order to take a deeper look and investigate the success seen by the transformer model, it is imperative to examine, in detail, the architecture and workings of the solution proposed by Vaswani et al. [1]. Unlike previously proposed sequence transduction models like [11] and [12], transformers maintain the encoder–decoder structure, as seen in Figure 1, but discard the recurrence and convolution aspects. This is made possible thanks to the novel multi-head attention mechanism proposed in addition to the point-wise feedforward networks ingrained in the transformer model. Figure 1 shows the overall transformer architecture as proposed by Vaswani et al. [1]. The following sections describe the various blocks contributing to this architecture in further detail.

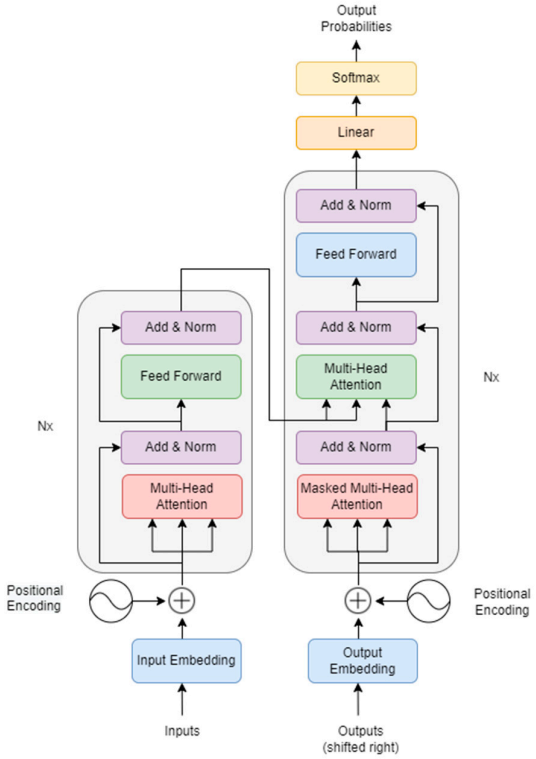

Figure 1. A depiction of transformer architecture.

1.2. Self Attention

The first and most important component of transformer architecture is the self-attention mechanism seen in Figure 2 which allows the model to learn the relationships between the elements of a sequence [13]. In the context of an LLM such as BERT, this would mean that in a sentence such as "The bank of the river is overflowing", the model would

use self-attention to conclude that the "bank" in this case refers to the side of a river as opposed to a financial organization.

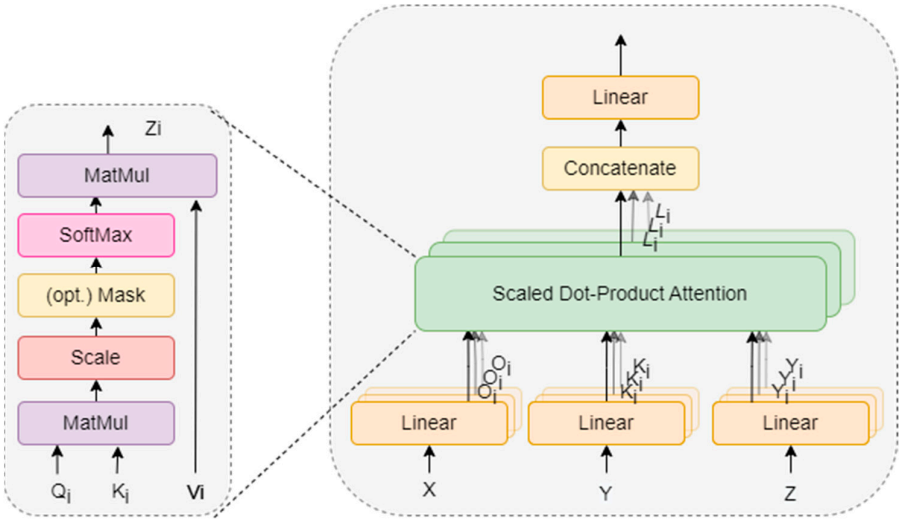

Figure 2. The structure of the attention layer. Left: Scaled Dot-Product Attention. Right: a multi-head attention mechanism.

In the encoder version of this layer, the inputs consist of queries and keys. The attention function is then applied to these vectors as seen in (1).

$$Attention\ (Q, K, V) = \left(\frac{Q \cdot K^{\top}}{\sqrt{d_k}}\right) \cdot V \tag{1}$$

where

- Q is the matrix of the queries;
- K is the matrix of the keys;
- V is the matrix of the values.

The equation is applied in a way that the dot product between the query and the key is first computed to form the score $S = Q \cdot K^{\top}$ These scores are important as they determine how much attention is given to other words when encoding words at the current position. These scores are then normalized in order to ensure the stability of the gradient to enhance training, thereby giving the normalized score $S_n = \frac{S}{\sqrt{d_k}}$. The softmax function is then applied to the normalized scores in order to translate them into probabilities $P = (S_n)$. These probabilities can then be applied to the value matrix to obtain $Z = V \cdot P$. This would mean that vectors with larger probabilities would receive a greater focus from the consequent layers [5]. In transformers, a multi-head attention system is used wherein the original queries, keys, and values are projected into H different sets of learned projections. For each projection, the attention equation from (1) is applied to formulate the output. The output across the H projections is then concatenated to form the multi-head output. The formulation for this process can be found in (2).

$$\begin{gathered} MultiHeadAttn\ (Q, K, V) = (\ head\ _1, \cdots,\ head\ _H) W^O \\ where \\ head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \end{gathered} \tag{2}$$

This process improves upon the performance seen by a single attention layer as it allows the model to focus on multiple equally important words based on different criterion instead of simply attributing a single word per input. This allows for multiple

complex relationships among different elements in a sequence to effectively be captured by the model [13] and therefore enhances the diversity of the subspace. The original transformer model proposed uses eight different heads; however, consequential works have experimented with optimizing the heads to retain the ones which provide the most important information [14].

1.3. Feedforward Networks

Another important component in the functioning of transformers is the feedforward network which is applied after the self-attention layers in the encoder and decoder. This network consists of two linear transformations and a non-linear ReLU activation function which is applied to each position separately and identically. This allows the model to ensure the same treatment across all positions in the input, meaning the token is processed in isolation. This allows the model to learn the complex transformations of the data at each position. Going back to the example mentioned in the previous section, the feedforward network in BERT would fine-tune the embeddings by adding additional layers of abstraction and complexity. So, if there was an example sentence like "The bank of the river is slippery", the self-attention would help give context and recognize it is not a financial organization as discussed previously while the feedforward network would capture the nuance about the bank being slippery due to it being close to water. The formulation for this network can be seen in Equation (3).

$$FFN(x) = max(0, xW_1 + b_1) \cdot W_2 + b_2 \tag{3}$$

1.4. Residual Connections

The transformer also implements residual connections [15] around each module followed by layer normalization [16] which applies normalization layer by layer. This helps mitigate the vanishing gradient problem by allowing gradients to flow directly, bypassing several layers. We can therefore represent each transformer block using the formulation seen in Equation (4).

$$H' = (Self Attention\ (X) + X) \cdot H = \left(FFN\left(H'\right) + H'\right) \tag{4}$$

This residual connection boosts the flow of data by relaying the information forward and therefore serves to enhance the model's performance. The '+' operator in this equation refers to element-wise addition which helps combat the vanishing gradient problem. In the context of the example discussed, these residual connections would make sure essential characteristics of the word "bank" are not lost in the depth of the model's layers.

1.5. Position Encodings

As the self-attention process of the transformer discards with the sequential way in which RNNs or LSTMs handle input embeddings and instead treats all inputs simultaneously and identically, it means that the self-attention layer is not able to account for the position of words in a sentence. However, since the words are sequential, a mechanism is needed which maintains the positions of the words within the encoded information and, therefore, the transformer model makes use of position encodings which are added to the input embedding. In the context of the example, this would mean the position encoding helps maintain the sequential context that the word "bank" is related to "river" and "slippery". The formulation for the added embeddings is seen in Equation (5).

$$\begin{aligned} PE(pos, 2i) &= sin\sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \\ PE(pos, 2i+1) &= cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \end{aligned} \tag{5}$$

Wherein *pos* is the position of a word within a sentence, d_{model} is the dimension and i is the current dimension of the position encoding. Using this, each element of the positional encoding corresponds to a sinusoid, thereby allowing the transformer model to learn to pay attention based on relative positions as well, consequently allowing it to extrapolate to longer sequences. These encodings have indeed been a focal point of the consequent research aiming to optimize the learning process. Indeed numerous works have proposed modifications such as a learning process for the encodings [17,18] or a relative form of position encoding [19].

Having discussed the importance and the working of transformer architecture, and given the rapid advances in the field of deep learning brought forth due to this model, it might be noteworthy to examine the historical progression since its introduction in 2017 leading up to transformers taking over many of the state-of-the-art techniques. While there exist surveys on the various types of transformer architectures that have been proposed, there seems to be a gap in the analysis from a historical viewpoint. Therefore, the rest of the paper examines a historical perspective on the progression of notable transformer architectures in addition to discussing the state-of-the-art techniques and architectures for data of different types.

2. Survey Methodology

The search for sources for this work was done following the PRISMA checklist [20]. The following subsections illustrate the points focused on for the survey's methodology.

2.1. Information Sources

Impactful works to be added to the survey were identified by searching online databases and scanning through the list of references within the main papers. The search was applied mainly to google scholar, OpenAI, Papers with Code, and arxiv as it was found that majority of the works on transformers were published through Arxiv. As the survey is based on the history of transformers, the search was not limited by year, but it was found that works were present only from the year 2017 to the present. The last search for sources was done on the 29 September 2023.

2.2. Search

The following search terms were used through all the above-mentioned databases: Transformers, State-of-The-Art Transformers, Key Transformer Architectures, Transformer Deep Learning, Transformer Vision, Transformer NLP, BERT.

2.3. Study Selection

The works were first shortlisted by their impact factor and number of citations. They were then further filtered based on their usefulness to the subject of this survey.

2.4. Data Collection and Data Items

A data extraction Excel spreadsheet was created that consisted of the following columns: Name of paper, Author, Date, Proposed Model, Datasets, Models Benchmarked Against, Results, and Key notes. This Excel spreadsheet was connected via a paper serial number to a word document that consisted of further key points summarized from the papers.

3. Survey Results

3.1. Early Transformer Implementations

3.1.1. Introductory Works

Since the introduction of the aforementioned transformer model in 2017, a vast array of works have aimed to build upon its novel architecture in order to optimize its performance for a variety of domains. Indeed, the work proposing the transformer model has been cited more than 90,500 times as of 29 September 2023, according to Google Scholar [21]. Among

the thousands of consequential works, a few emerge as notable models which have consequently contributed to pushing the overall state-of-the-art techniques and have established themselves as standards in their fields. Figure 3 displays a timeline of these notable works arranged chronologically and coded according to the domain of implementation.

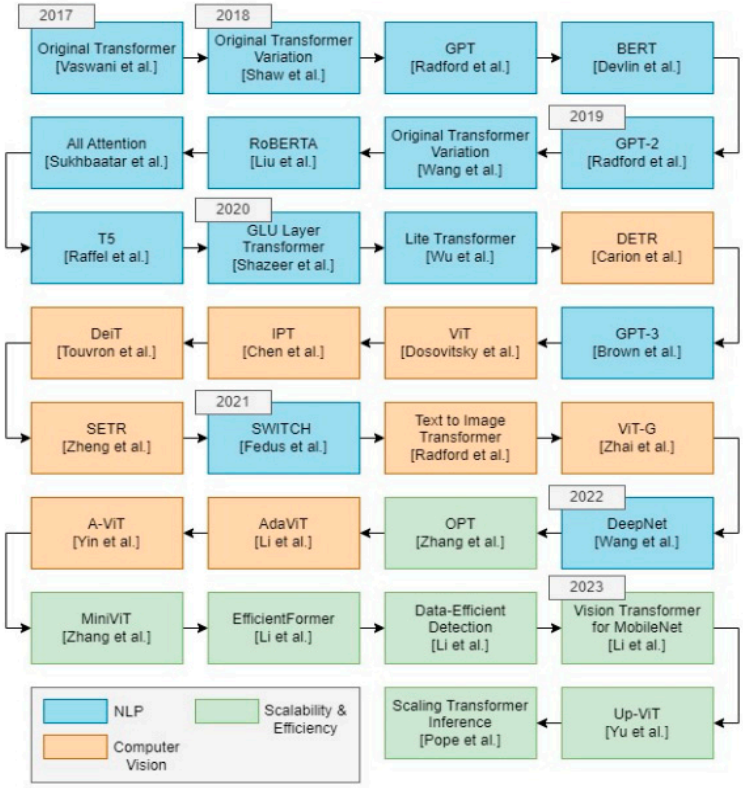

Figure 3. The timeline of the state-of-the-art transformer models [1,17,19,22–47].

In order to benchmark these works, a number of datasets have been utilized by the various works. A few of the commonly used datasets are BookCorpus [48], WMT 2014 [49], Wikipedia [50], C4 [22], ImageNet [51], and COCO [52].

An early work building upon the transformer model was that of Shaw et al. [19], which simply involved extending the self-attention mechanism of transformers to efficiently consider representations of the relative positions or distances between sequence elements. This is done by modeling the input as a labeled, fully connected graph with the edges between input elements x_i and x_j represented by vectors $a_{ij}^V, a_{ij}^K \in R^{d_a}$. A modification is then made to the transformer equation wherein edge information is then propagated to the sublayer output as seen in Equation (6).

$$z_i = \sum_{j=1}^{n} \alpha_{ij} \left(x_j W^V + a_{ij}^V \right) \tag{6}$$

Using these improved embeddings, the authors were able to report improvements in both the EN-DE and EN-FR tasks over the vanilla transformer architecture.

Another early and majorly consequential work was that of Radford et al. [23] who proposed the famous Generative Pre-Training (GPT) model. The base model used for the work was the transformer architecture as it allowed the authors to capture long-range linguistic structures. The idea proposed by the authors was one where the model can perform more optimally for small amounts of labeled text data when it is generatively trained in an unsupervised manner on a large unlabeled text corpus consisting of diverse samples and then discriminatively fine-tuned on the specific task at hand. They do this by utilizing a multi-layer transformer-decoder [53] architecture which applies a multi-

headed self-attention operation over the input context tokens followed by position-wise feedforward layers to produce an output distribution over the target tokens. These trained weights can then be used with an auxiliary objective for classification tasks. The architecture used by the model can be seen in Figure 4.

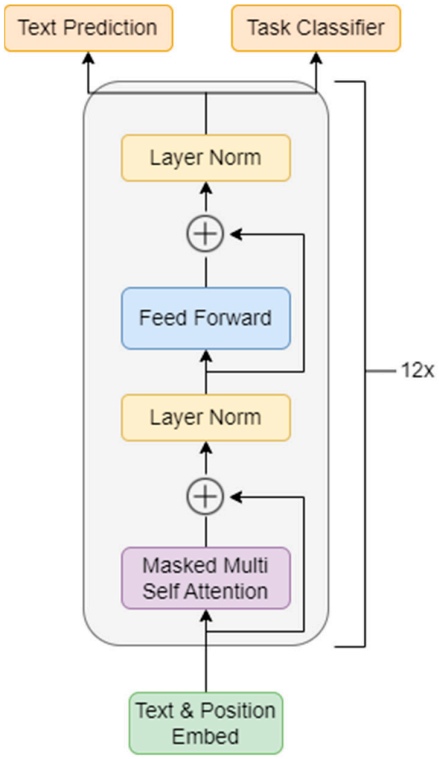

Figure 4. GPT Architecture.

A similar approach to that of GPT was seen by the Bidirectional Encoder Representations from Transformers (BERT) model proposed by Devlin et al. [17], where unlabeled data is used to pre-train the transformer model in an unsupervised fashion before the model is fine-tuned using representative samples from the problem at hand. The major improvement proposed by the authors is the use of bidirectional encoder representations unlike previous solutions, which involved unidirectional models being used in the learning process such as GPT using a left-to-right architecture where each token in the self-attention layer was only able to attend to previous tokens. The BERT model achieves bidirectional learning by using a masked language model (MLM) pre-training objective which the authors adapted from the Cloze task [54]. This model randomly masks some of the tokens from the input with the objective of predicting the original vocabulary ID of the masked work based on the context. This allows the representation to join the left and right context, thereby allowing a bidirectional training process. To further the MLM objective, the authors also implement a next-sentence prediction task which jointly pre-trains text-pair representations. Thereby, the authors outline two distinct processes in training the model, the pre-training and the fine-tuning. During the pre-training, the model is given various tasks when training on unlabeled data, whereas for fine-tuning, the model is initialized with the parameters from the pre-training and all of the parameters are fine-tuned using labeled data from the downstream tasks. Each of these tasks has a separate fine-tuned model; however, in general, there is no architectural difference between the pre-training and the fine-tuning process except for the output layers. Figure 5, adapted from [17], shows the pre-training and fine-tuning procedures.

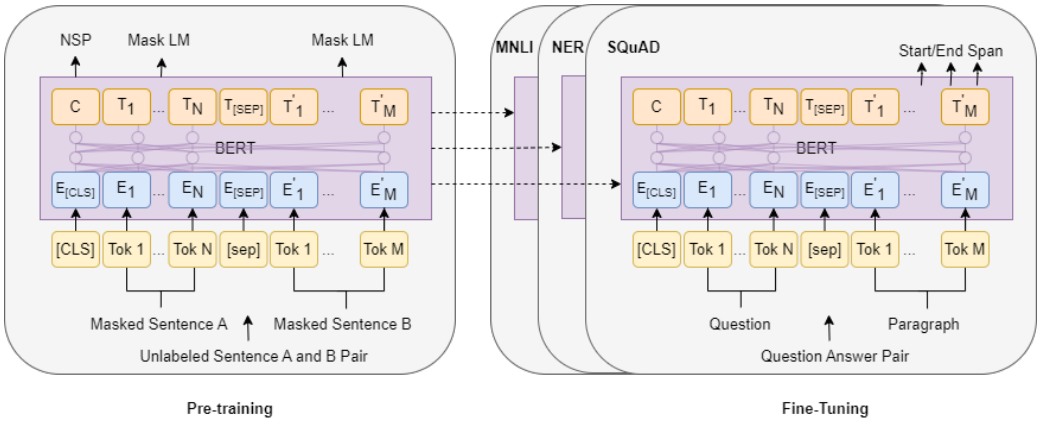

Figure 5. The pre-training and fine-tuning process of the BERT model.

Using this relatively simple conceptual approach, the BERT model was able to obtain state-of-the-art results on eleven natural language processing (NLP) tasks, thereby establishing it as a notable work which numerous consequent models have been built upon.

It was soon after that, in the beginning of 2019, that Radford et al. followed up their proposed GPT model with a model they called GPT-2 which followed a similar philosophy of multi-task learning which they based on a framework proposed by Caruana [53]. In their work, Radford et al. aimed to unify the two dominant approaches, namely, pre-training followed by supervised fine-tuning as well as a technique with unsupervised approaches towards specific tasks such as commonsense reasoning [55] and sentiment analysis [24]. They achieve this by performing language modeling where, in addition to conditioning a model on the input, it is also conditioned on the task. They train their model in an unsupervised manner on a dataset consisting of millions of web pages, called WebText, producing GPT-2, which is an enormous 1.5 billion parameter model which achieved state-of-the-art results on seven language modeling tasks in a zero-shot system. The authors hypnotized that a large enough model would learn tasks embedded within language and would not require explicit, supervised training, which was proven by their results.

Meanwhile, Wang et al. [25], in 2019, proposed a direct improvement upon the transformer model itself by formulating a deep transformer model which they claimed would bypass the prevalent big transformer counterpart. They achieved this using a dual approach where, firstly, they implemented the proper use of layer normalization in addition to introducing a novel way to pass the combinations of previous layers to the next ones. Furthermore, they trained a 30-layer encoder, which they claim was the deepest at the time. Using this approach, the authors were able to outperform the results of both the shallow and the big transformers on the WMT'16 EN-DE, the NIST OpenMT'12 Chinese-English, and the WMT'18 Chinese-English tasks.

Liu et al.'s proposed Robustly Optimized BERT Pre-training Approach (RoBERTa) model [26] was introduced with the idea of improving the limitations of the BERT model which were caused by significant undertraining. The authors achieved this by training the model over a larger dataset, which consisted of CC-News and OpenWebText in addition to the two datasets used to train the original BERT model, and training on longer sequences. The performance was further improved by making the following changes on the original model: dynamically changing the masking pattern that was applied to the training data and removing the Next Sentence Prediction (NSP) objective. Unlike in the BERT model, where the mask was generated only once during the data preprocessing stage, for the RoBERTa, the authors generate a masking pattern every time a sequence is fed into the model. The authors came to the conclusion that removing NSP matched or slightly improved the downstream task performance after comparing the training of their model with and without NSP. Throughout their experimentation, for a more accurate comparison, the original optimization hyperparameters of the BERT model were initially maintained. The

model was able to achieve state-of-the-art results on GLUE [56], RACE [57] and the Stanford Question-Answering Dataset (SQuAD) [58], which are notable NLP tasks.

Another notable proposed modification of the transformer model is that outlined by Sukhbaatar et al. [27], which suggests removing the feedforward layer from the transformer architecture and solely using the attention layers. This is done by augmenting the attention layers with persistent memory vectors which serve the same purpose as the feedforward layers. On the first level, they first show that a feedforward sublayer can be viewed as an attention layer. This argument can then be used to merge them into a single layer which performs both functions by applying the attention mechanism simultaneously on the sequence of input vectors, as in the attention layer, as well as a set of vectors not conditioned on the input. Using this approach, they report outperforming models of similar sizes on the enwik8 and WikiText-103 datasets.

An interesting work published in late 2019 that explored the NLP landscape is that of Raffel et al.'s T5 model [22]; the researchers followed a transfer learning approach in introducing a unified framework which converted all text-based language problems into a text-to-text format. They experiment with a variety of pre-training objectives, architectures, datasets and transfer approaches in addition to developing a new dataset they call the Colossal Clean Crawled Corpus. Using this pre-training regime, they report having achieved state-of-the-art results on a number of prevalent challenges in summarization, question answering, and text classification.

3.1.2. Further Progression

In early 2020, Shazeer [28] proposed an improvement to the transformer model, which involved variants of Gated Linear Units (GLUs) [59] being applied to the feedforward sublayers of the transformer model. These variations were implemented using different linear and non-linear activation functions in place of sigmoids, and the authors report an improvement in performance over the generally used ReLU activation function when evaluating on the SQuAD, GLUE, and SuperGlue [60] tasks.

It was in April of 2020 that a key architecture in the form of the Lite Transformer was introduced by Wu et al. [29]. The reasoning behind the introduction of this architecture was that the authors argued that transformers require an enormous amount of computation in order to achieve high performance and, therefore, they would not be suitable for mobile applications that are constrained by hardware and battery resources. Therefore, they proposed the Lite Transformer specifically to be deployed to perform NLP on mobile devices. They introduce Long-Short Range Attention (LSRA), where one group of heads specialize in local context modeling using convolution while the other specializes in long-distance relationship modeling using attention. They report that this approach shows improvement over the vanilla transformer in three established language tasks, namely, machine translation, abstractive summarization, and language modeling. The Lite Transformer block can be seen in Figure 6.

Using this approach, the proposed model reduces the computation of the transformer base model by 2.5× with only a 0.3 BLEU score degradation. Furthermore, the authors report implementing pruning and quantization processes to compress the model size by 18.2×.

Carion et al. [30] propose a ground-breaking object detecting transformer named DETR that views object detection as a direct set prediction problem. The main components of the model are a set-based loss that forces predictions via a bipartite matching and a transformer encoder and decoder. The overall architecture of the model is illustrated in the following Figure 7.

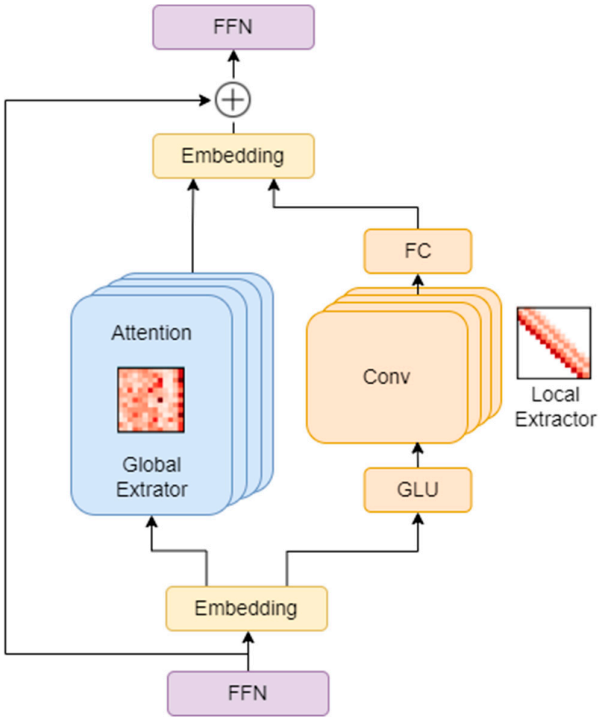

Figure 6. The Lite Transformer block.

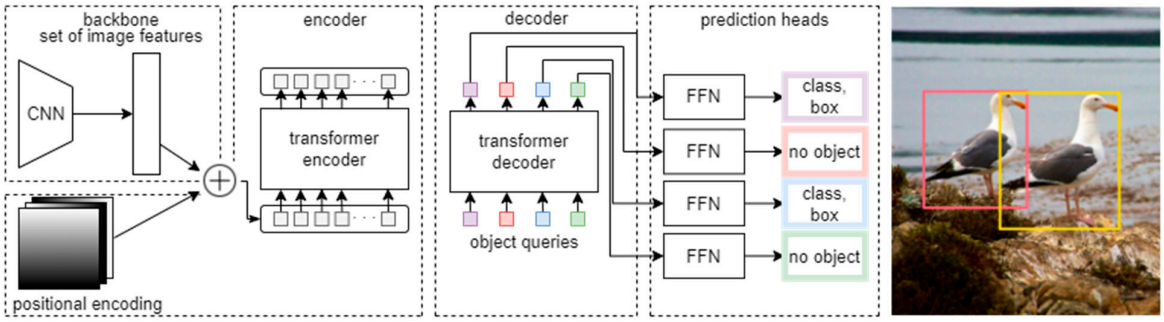

Figure 7. The DETR's architecture.

The CNN is used to extract a compact feature representation of the input image by generating a low-resolution activation map. The transformer's encoder and decoder follow the model architecture of Vaswani et al. [1]. The decoder output encodings are decoded into box coordinates and class labels by the feedforward network. The object detection set prediction loss produces a bipartite matching between the predicted and the ground truth objects and then optimizes the object-specific losses. This model is on par with state-of-the-art Faster R-CNN baseline on the famous COCO object detection dataset. The Faster R-CNN was a model proposed by Ren et al. which used a Region Proposal Network to generated region proposals which were then used by a Fast R-CNN for detection [61].

Around mid-2020, Brown et al. [31] proposed a work which improved on the state-of-the-art NLP transformer model by proposing their improved GPT-3 model. The authors scale-up the model by training it with 175 billion parameters which results in a model which can perform a variety of tasks without requiring task-specific gradient updates or fine-tuning, unlike the previous generations of the model. The other variation from the architecture of GPT-2 is that of the use of alternating dense and locally banded sparse attention patterns in the layers of the transformer. The model is able to perform well and even achieve SOTA results on famous NLP dataset tasks with few-shot demonstrations which are specified purely via text interactions with the model.

Dosovitskiy et al. [32] introduced the Vision Transformer (ViT) in late 2020, which caused a shift in the research field. In order to adapt the transformer for image tasks, the authors applied a standard transformer to images by splitting an image into patches and providing the sequence of the linear embeddings of the patches as the input to the transformer. The overview of the ViT model can be seen in Figure 8.

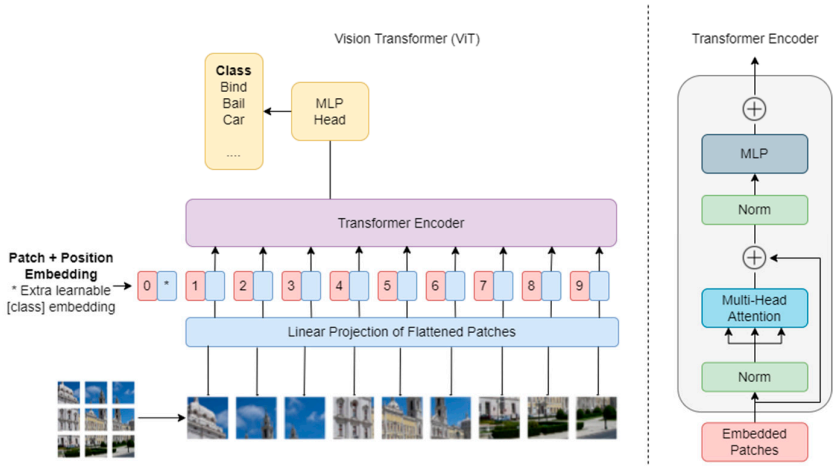

Figure 8. An overview of the ViT model's architecture.

The image is first broken down into patches which are passed through a trainable linear projection resulting in a D-dimension latent vector where D is the latent vector size used by the transformer in its layers. An additional embedding at position 0 is added, which serves as a class label. A classification head consisting of simple, dense layers is added with a hidden layer during pre-training and a single linear layer while fine-tuning. The authors report improvements on the state-of-the-art results achieved by CNN-based models for a range of benchmark datasets such as ImageNet [51], CIFAR10, CIFAR100 [62] and Oxford-IIIT Pets [63].

An interesting implementation using transformer architecture was that created by Zheng et al. [33], who proposed a segmentation model named the Segmentation Transformer (SETR). They implement a solution wherein semantic segmentation is treated as a sequence-to-sequence prediction task with a transformer being deployed to encode an image as a sequence of patches. They combine the encoder with a single decoder by modeling the global context in each layer of the transformer. Figure 9 shows the architecture of their proposed system.

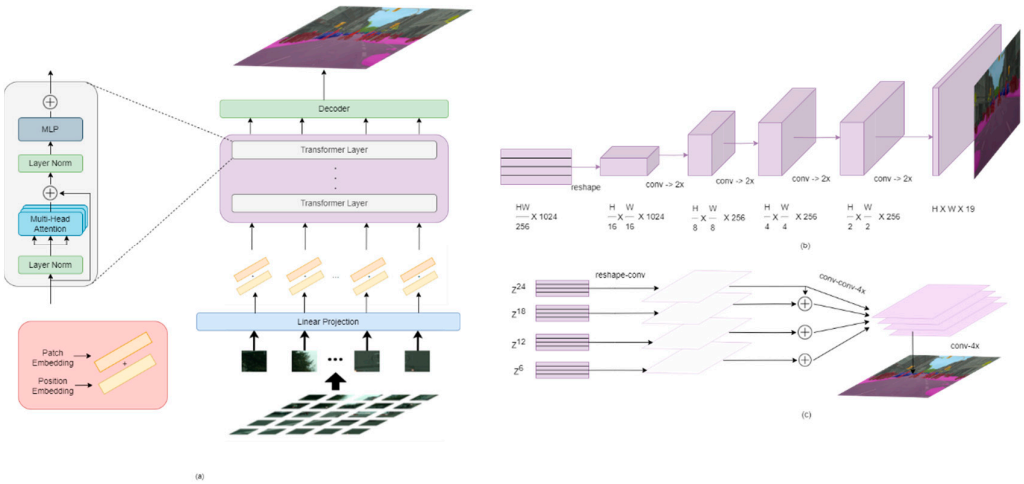

Figure 9. The SETR's architecture.

In the system, the image is first split into fixed patches which are linearly embedded with position encodings added. The resulting sequence of vectors is then fed into a standard transformer encoder. They propose two different decoder designs for pixel-wise segmentation, as can be seen in parts (b) and (c) of Figure 9. They then put these features together through a multi-level feature aggregation system as seen in part (c) in Figure 9. Using this methodology, they were able to achieve state-of-the-art results on the ADE20K [64] and Pascal Context [65] challenges.

To study the low-level vision tasks like denoising, super-resolution, and deraining, Chen et al. [34] worked on developing a new pre-trained model using transformer architecture, called the image processing transformer (IPT). The entire network is composed of multiple pairs of heads and tails corresponding to different tasks and a single shared body, so the pre-trained model becomes more compatible with different image processing tasks. Multiple corrupted counterparts were generated for each image in the famous benchmark ImageNet dataset using several carefully designed operations. The model was then trained on the dataset's original images in addition to the newly generated images, and it outperformed the current state-of-the-art methods on several low-level benchmarks.

Touvron et al. [35] proposed a major non-convolutional transformer model, called the DeiT, that has fewer parameters than the ResNet model, which makes it trainable on a single computer in less than 3 days. Furthermore, a teacher–student strategy which relies on a distillation token procedure was used to ensure that the student learns from the teacher through attention. Using the distillation technique enables image transformers to learn more from a convent than from another comparably performing transformer. Therefore, a combination of those techniques results in a top accuracy of 85.2% on ImageNet with no external data. Consequently, transferring these models to a different downstream task, such as a fine-grained classification on popular benchmark datasets like CIFAR-10, Oxford-102 flowers, and Stanford Cars, achieved competitive results.

3.1.3. Recent Advancements

Fedus et al. [36], in early 2021, found that the widespread adoption of the mixture of experts (MoE) model has been obstructed by the complexity, communication costs, and training instability of the model. As a result, they introduced the switch transformer to simplify the MoE routing algorithm and reduce the communication and computational costs. This is done by distilling the sparse pre-trained and specialized fine-tuned models into small, dense models while preserving 30% of the quality grains. To increase the scale of the neural language model, data, model, and expert parallelism was combined to build models with a trillion parameters which improved the pre-training speed four times for a strongly tuned T5-XXL baseline model.

Radford et al. [37], meanwhile, aimed to leverage a much broader source of supervision by utilizing the raw text about the images to train a model instead of training it on a fixed set of predetermined object categories. They have shown that learning a SOTA image representation from scratch can be efficiently done by pre-training a model to match the captions with the corresponding images. Following the pre-training phase, natural language is used to reference the learned visual concepts or express new ones, which, in turn, enables the zero-shot transfer of the models to downstream tasks. This approach was tested on 30 different existing computer vision datasets and has proven its competitivity with the fully supervised baseline models without the need for any dataset specific training.

In a recent implementation, Zhai et al. [38] aim on scaling-up the original ViT model to achieve better results, generating a model that they named ViT-G. Through the improvements made, the authors were able to train their model using data parallelism alone and were able to fit the entire model on a single TPUv3 core. The model was scaled-up with two billion parameters. The authors removed the class token to save memory and additionally equated the number of multi-head attention-pooling heads to the number of attention heads in the model. Finally, they removed the final nonlinear projection before the final prediction layer, which was present in the original ViT model. The authors also scaled-up

the data by using a larger version of the JFT-300M dataset, namely, the JFT-3B dataset. Using this model, the authors were able to achieve a new state-of-the-art result on the ImageNet dataset with a top accuracy of 90.45%. They also proved that they achieved a decent accuracy of 84.86% with few-shot learning, limiting to only 10 examples per class from the ImageNet dataset for fine-tuning.

To make large-scaled language models more accessible, less complex and resources less expensive, Zhang et al. [39] propose a suite of eight decoder-only pre-trained transformers that consist of 125 million to 175 billion parameters, namely, Open Pre-trained Transformers (OPTs). Their model is comparable to the state-of-the-art GPT-3 model with only 1/7th of the carbon footprint. The model is directly developed from the GPT-3 model with a change in the number of layers and attention heads to vary the parameter size. The smallest model, consisting of 125M parameters, consists of 12 layers and 12 attention heads, while the biggest model, consisting of 175B parameters, consists of 96 layers with 96 attention heads. The batch size is varied from the original model to increase computational efficiency. While training the OPT-175B model, the authors faced an issue of loss divergence, which they fixed by lowering the learning rate and restarting the training from an earlier checkpoint. The authors noticed a correlation between the loss divergence, the dynamic loss scalar crashing to zero, and the l^2-norm of the activations of the final layer spiking. From this, the authors derived a conclusion to pick restart points where the dynamic scalar loss was still in the healthy state, which is greater than 1. The models were also additionally trained with a larger set of data, including datasets that were used to train the RoBERTa, The Pile dataset, and the PushShift.io Reddit dataset. The models were evaluated across 14 NLP tasks, and it was seen that for zero-shot, the average performance follows the trend of GPT-3 for 10 tests.

3.2. Text-Based Applications

Transformers have revolutionized the realm of text-based applications and natural language processing (NLP) through providing solutions to a variety of problems such as text classification, question answering, text summarization, machine translation, and text generation [66]. The first prevalent model in text-based applications is one already analyzed previously—the BERT model proposed in 2018 by Devlin et al. [17]. Despite being a number of years old, this architecture is still relevant to this day due to how groundbreaking it was when it was proposed. Indeed, the BERT model's NLP transformer has been the base for various other prevalent models such as the RoBERTa [26] in 2019, which achieved excellent results by proposing a variation of the BERT model, ETC [67], in 2020, which reported high performance when building upon the BERT model and using the weights provided by the RoBERTa as well as Big Bird [68] in 2021, which was proposed as a variation of the BERT model for longer sequences. Another notable implementation of transformers for the text domain is that of TENER [69], proposed in 2019 as a solution to using transformers for the named entity recognition task—which is the task of finding the start and end of an entity in a sentence and assigning a class for this entity. This is especially useful in applications such as question generation [70], relation extraction [71], and coreference resolution [72]. This model adapts the transformer encoder to model character-level features and word-level features.

3.3. Image-Based Applications

In the realm of image-based applications, an early implementation was that of the Image Transformer proposed by Parmar et al. in 2018 [73]. This model restricted the transformer's self-attention to attend to local neighborhoods. However, in the domain of images, one model reigns supreme, which is that of the Vision Transformer introduced by Dosovitsky et al. in 2020 [32], which was discussed earlier in this work. Numerous consequential works have been derived from this proposed model. A work based on ViTs, which outperformed it, was that of Touvron et al. [35], which has also been previously described in this paper. An alternative framework based on ViTs is that of the Feature Fusion Vision Transformer (FFVT) proposed by Wang et al. [74] in 2021, which adopts

the patch generation process employed by ViTs but modifies it to avoid overlap. An extremely recent solution making use of transformers in the field of vision is that of the Unsupervised Semantic Segmentation Transformer (STEGO), proposed in March 2022 by Hamilton et al. [75]. This model makes use of transformers to localize semantically meaningful categories within image corpora without any form of annotation. This is done by using a novel loss function that encourages features to form compact clusters while preserving their relationships across the corpora.

3.4. Miscellaneous Applications

In addition to the previously identified domains, a couple of miscellaneous approaches are discussed. The first of them is that of audio classification, for which a number of audio transformers have been proposed over the years. The first of these was proposed by Dong et al. in 2018 [76] with the idea of applying a two-dimensional attention block in the proposed audio transformer model. A consequential model for audio captioning was that of the Audio Captioning Transformer (TRACKE) proposed by Koizumi et al. [77] in 2020. The TRACKE estimates keywords, which comprise a word set corresponding to audio events/scenes in the input audio, and generates the caption while referring to the estimated keywords to reduce word-selection indeterminacy. Following this, in 2021, the Audio Spectrogram Transformer was proposed by Gong et al. [78] as a convolution-free, purely attention-based model for audio classification.

The second miscellaneous set of approaches are those of time series modeling, first introduced through the work proposed by Liu et al. in 2021 [79]. This is done by adding gating to the vanilla transformer in an approach they call Gated Transformer Networks. Another model proposed in 2021 was for time series forecasting, by Zhou et al., which they call the Frequency Enhanced Decomposed Transformer (FEDformer) [80]. An interesting time-series-based implementation using transformers is that of the TranAD proposed by Tuli et al. in 2022 for anomaly detection in time series data [81]. The TranAD uses focused score-based self-conditioning to enable robust multi-modal feature extraction and adversarial training to gain stability. The results obtained by these models are highlighted in the next section.

3.5. Recent Directions

The recent increase in the relevance of transformers and the work being conducted in exploring the uses of these versatile models has resulted in transformers becoming more accessible for implementation in various real-world applications. Indeed, one of the applications which has greatly seen the use of transformers is that of medical image analysis [82]. While numerous works have previously aimed at applying a variety of artificial intelligence algorithms towards solving key issues within the realm of medicine, such as COVID-19 detection [83] and the extraction and detection of a fetal electrocardiogram [84,85], with the introduction of transformers for vision, a large number of techniques such as image synthesis/reconstruction, registration, segmentation, detection, and diagnosis have been unlocked. Indeed, as Li et al. [86] discuss, the ability of transformers to capture long-range dependencies as well as the scalability of self-attention enables their diverse usage within the medical field. In addition to the capabilities of transformers to be used within medical imaging, Shamshad et al. [87] discuss their implementations in various other medical applications such as leveraging their text generation ability to generate medical reports as well as using it for regression tasks such as survival outcome prediction.

With the increase in the general depth and complexity of transformers, a number of researchers have chosen to focus on the stability of extremely deep transformers. One such approach relying on scaling is that of the DeepNet proposed by Wang et al. [40], which introduces a new normalization function to modify the residual connections in transformers along with having a theoretically derived initialization process. Using this technique, they report being able to successfully scale transformers up to 1000 layers.

Furthermore, with the rise in the adaptation and use of transformers, an increase in the focus on developing a lighter version of transformers has been noted. This is because, while transformers have produced revolutionary results, it has been at a huge computation cost, thereby preventing the models from being as easily adapted as earlier deep-learning techniques such as CNNs [88]. To this end, numerous researchers have proposed works aiming to scale or slim the weights of a traditional transformer. A notable attempt is that of the EfficientFormer and EfficientFormerV2 proposed by Li et al. [41,42]. These models make use of a process called latency-driven slimming to reduce the time taken for inferencing using the trained transformers. The EfficientFormerV2 work further introduces a fine-grained joint-search strategy that can find efficient architectures by optimizing the latency and the number of parameters simultaneously. A similar work aiming to achieve efficient image recognition was that of the AdaViT proposed by Meng et al. [43], which serves as a computational framework learning to derive policies on which patches, self-attention heads, and transformer blocks to use throughout the backbone on a per-input basis. This is done by attaching a lightweight decision network to the backbone to produce on-the-fly decisions. A similar thought process was seen in the case of the A-ViT method proposed by Yin et al. [44] that adaptively adjusts the inference cost for images of different complexities. This is done by reducing the number of tokens in the ViT as the inference proceeds. Using the proposed method requires no extra parameters or sub-networks, unlike the AdaViT, as the learning of the adaptive halting is based on the original network parameters. A recent work aiming to improve the efficiency of transformer inference is that of Pope et al. [45], who develop an analytical model for inference efficiency to select the best multi-dimensional partitioning techniques. These are combined with low-level optimizations to achieve a Pareto frontier on latency and FLOPS utilization tradeoffs.

Another key work was that of Zhang et al. in the introduction of the MiniViT model [46], which applies weight multiplexing to reduce the complexity of the traditionally immense vision transformer. This is done by multiplexing the weights of consecutive transformer blocks, wherein weights are shared across layers, while imposing a transformation on the weights to increase diversity. Furthermore, the weight distillation over self-attention is also applied to transfer knowledge from the large ViT models to the weight-multiplexed compact models.

Yu and Wu [47] proposed a pruning framework to be applied to ViTs in order to simplify all components in a transformer without altering the structure. This framework, called the UP-ViT, estimates the importance score of each filter in a pre-trained ViT model before removing redundant channels. Furthermore, they propose a progressive block-pruning method that removes the least important block and proposes new hybrid blocks for ViTs.

An interesting area of recent work has been in making the training of transformers a more data-efficient process. An early work in this space was that of the previously discussed DeiT model proposed by Touvron et al. [35], who proposed using what they called a distillation token to effectively learn from a teacher in a teacher–student method employed to train transformers. This distillation token is learned through backpropagation, through the interaction with the class and patch tokens through self-attention layers. A more recent approach towards achieving data-efficient training is proposed by Wang et al. [89], who aim to achieve this by claiming that the sparse feature sampling from local image areas is key and, therefore, they propose a procedure where they alternate how key and value sequences are constructed in the cross-attention layer. Furthermore, they also introduce a label augmentation method which provides richer supervision, in turn, achieving greater data efficiency.

4. Discussion

4.1. Historical Insight

Table 1 summarizes the historical works discussed in the previous section. The works are color-coded in the timeline, wherein the works targeted towards text and NLP tasks are color-coded in blue and the works targeted at image-related tasks are color-coded in orange.

Table 1. A summary of the history of transformer studies.

Name of Paper	Author	Date	Proposed Model	Datasets	Models Benchmarked Against	Results	No. of Citations
Attention Is All You Need [1]	Vaswani et al.	Jun 2017	Transformer	WMT 2014 English-to-German translation task, WMT 2014 English-to-French translation task	ByteNet, Deep-Att + PosUnk, GNMT + RL, ConvS2S, MoE, Deep-Att + PosUnk Ensemble, GNMT + RL Ensemble, ConvS2S Ensemble	28.4 BLEU for EN-DE and 41.8 BLEU for EN-FR with Transformer (Big)	90,568
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [17]	Devlin et al.	Oct 2018	NLP Transformer	BooksCorpus, English Wikipedia	GLUE, SQuAD v1.1, SQuAD v2.0	BERT Large average score of 82.1 on GLUE testing	78,786
Self-Attention with Relative Position Representations [19]	Shaw et al.	Mar 2018	Translation NLP Transformer	WMT 2014 English–German, 2014 WMT English–French	Original transformer	English-to-German improved over the baseline by 0.3 and 1.3 BLEU for the base and big configurations, respectively, and English-to-French improved by 0.5 and 0.3 BLEU for the base and big configurations, respectively	1882
Improving Language Understanding by Generative Pre-Training [23]	Radford et al.	Jun 2018	GPT	Unsupervised training—BooksCorpus fine-tuning based on task-natural language inference, question answering, semantic similarity, and text classification, all present in GLUE	NLI-ESIM + ELMo, CAFE, Stochastic Answer Network, GenSen, Multi-task BiLSTM + Attn QA-val-LS-skip, Hidden Coherence Model, Dynamic Fusion Net, BiAttention MRU SS, Classification-sparse byte mLSTM, TF-KLD, ECNU, Single-task BiLSTM + ELMo + Attn, Multi-task BiLSTM + ELMo + Attn	Best results: NLI-SNLI-89.9 QA-Story Cloze-86.5 SS-STSB-82.0 Classification-CoLA-45.4 GLUE-72.8	6642
RoBERTa: A Robustly Optimized BERT Pre-training Approach [26]	Liu et al.	Jul 2019	Variant of BERT	BooksCorpus, English Wikipedia, CC-News, OpenWebText, Stories	BERT Large, XLNet Large Enseambles-ALICE, MT-DNN, XLNet	Best results: SQUAD 1.1-F1-94.6 Race-Middle-86.5 GLUE-SST-96.4	8926
Language Models are Unsupervised Multitask Learners [90]	Radford et al.	Feb 2019	(GPT2) GPT variation	Created own dataset called WebText	Baseline models, in general	55 F1 on CoQa, matches or exceeds 3 of 4 baselines, has state-of-the-art results on 7/8 datasets	6954
Learning Deep Transformer Models for Machine Translation [25]	Wang et al.	Jun 2019	Translation NLP Transformer	WMT'16 English–German (En-De) and NIST'12 Chinese–English (Zh-En-Small)	Original transformer	Avg. BLEU scores [%] on NIST'12 Chinese-English translation: 52.11 BLEU scores [%] on WMT'18 Chinese-English translation: newtest17-26.9, newstest18-27.4	548
Augmenting Self-attention with Persistent Memory [27]	Sukhbaatar et al.	Jul 2019	Introduction of new Layer for transformer	Character level modeling—enwik8, text8 Word level modeling—wikiText-103	Character-LN HM-LSTM, Recurrent highway networks, Large mLSTM, T12, Transformer+adaptive span Word-LSTM, TCN, GCNN-8, LASTM+nEURAL CACHE, 4-LAYER QRNN, LSTM+Hebbian+Cache, Transformer XL Standard	enwik8-1.01 text8-1.11 wiki-18.3	94

Table 1. *Cont.*

Name of Paper	Author	Date	Proposed Model	Datasets	Models Benchmarked Against	Results	No. of Citations
GLU Variants Improve Transformer [28]	Shazeer N	Feb 2020	Variation of original and T5 by adding GLU layers	C4	T5	GLUE best average score of 84.67-FFNReGLU	141
Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer [22]	Raffel et al.	Jan 2020	NLP Transformer	C4, fine-tuning using GLUE and SuperGLUE	Self-trained Baseline experimental setup	GLUE-85.97 CNNDM-20.90 SQuAD-85.44 SGLUE-75.64 EnDe-28.37 EnFr-41.37 EnRo-28.98	10,117
Lite Transformer With Long-Short Range Attention [29]	Wu et al.	Apr 2020	Lightweight translation transform-ers to deploy on end devices	IWSLT'14 German–English, WMT English to German, WMT English to Franch (En-Fr)	Original Transformer, adaptive inputs (Baevski and Auli)	CNN-DailyMail-F1-Rouge-R-1:41.3, R-2:18.8, R-L:38.3 (did not beat original but lighter) WIKITEXT-103-Valid ppl.-21.4, Test ppl.-22.2	234
End-to-End Object Detection with Transformers [30]	Carion et al.	May 2020	Object detection Transformer	COCO 2017	Different variations of Faster RCNN for detection Panoptic FNN, UPSnet for panoptic segmentation	Panoptic Quality-45.1 Able to classify classes in general without being biased to the training images	7829
Language Models are Few-Shot Learners [31]	Brown et al.	May 2020	GPT-3	Common Crawl, WebText, Books1, Books2, Wikipedia	QA-RAG, T5-11B (2 variants)	QA-Beats SOTA IN TriviaQA-71.2, GPT-3-FewShot LAMBADA-FEW SHOT-86.4 (BEATS SOTA) PIQA-few-shot-82.8	14,698
An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale [32]	Dosovitsky et al.	Oct 2020	Vision Transformer (ViT)	Trained on ILSVRC-2012, ImageNet-21k, JFT Transfered on ReaL labels, Cifar10/100, Oxford-IIT Pets, Oxford Flowers-102	BiT-L (ResNet152x4), Noisy Student (EfficientNet-L2)	ImageNet-88.55-ViT-H ReaL-90.72-ViT-H CIFAR-10-99.50-ViT-H CIFAR-100-94.55-ViT-H Oxford-IIIT-Pets-97.56-ViT-H Oxford Flowers-99.74-ViT-L VTAB (19 tasks)-77.63-ViT-H	21,833
Pre-Trained Image Processing Transformer [34]	Chen et al.	Dec 2020	Image Processing Transformer (IPT)	ImageNet	Super-resolution-VDSR, EDSR, RCAN, RDN, OISR-RK3, RNAN, SAN, HAN, IGNN image denoising-CBM3D, TNRD, DnCNN, MemNet, IRCNN, FFDNet, SADNet, RDN image deraining-DSC, GMM, JCAS, Clear, DDN, RESCAN, PReNet, JORDER.E, SPANet, SSIR, RCDNet	Super resolution: set5-38.37, set14-34.43, B100-32.48, Urban100-33.76 image denoising: BSD68-30-30.75, 50-28.39, Urban100 30-32.00, 50-29.71 deraining: Rain100L-PSNR-41.62, SSIM-0.9880	1129
Training data-efficient image transformers and distillation through attention [35]	Touvron et al.	Dec 2020	based on ViT (DeiT)	ImageNet	ResNet, RegNetY, EfficientNet, KDforAA, ViT (all versions)	DeiT-B 384/1000 epochs outperforms ViT and EffecientNet-85.2 acc	4021
Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers [33]	Zheng et al.	Dec 2020	Semantic Segmenta-tion Transformer (SETR)	CityScapes, ADE20K, Pascal Context, all trained separately	SCN, Semantic FPN ADE20K Dataset-FCN, CCNet, Strip pooling, DANet, OCRNet, UperNet, Deeplab V3+ Pascal Context-DANet, EMANet, SVCNet, Strip pooling, GFFNet, APCNet Cityscapes validation-FCN, PSPNet, DeepLab-V3, NonLocal, CCNet, GCNet, Axial-DeepLab-XL, Axial-DeepLab-L	ADE20K-mIoU = 50.28 Pascal = 55.83 Cityscapes = 82.15	2030

Table 1. *Cont.*

Name of Paper	Author	Date	Proposed Model	Datasets	Models Benchmarked Against	Results	No. of Citations
Switch Transformers: Scaling To Trillion Parameter Models With Simple And Efficient Sparsity [36]	Fedus et al.	Jan 2021	Transformer	Colossal Clean Crawled Corpus (C4)	MoE, T5	Negative Log Perplexity (quality threshold) -1.534 Best average score on SQuAD with score of 88.6 vs. T5	894
Learning Transferable Visual Models From Natural Language Supervision [37]	Radford et al.	Feb 2021	Text-to-Image transformer	Created own dataset called WIT (WebImageText)—has the similar wordcount to WebText	Visual N-Grams for comparison on zero-shot transfer	aYahoo-98.4 ImageNet-76.2 SUN-58.5	8121
Scaling Vision Transformers [38]	Zhai et al.	Jun 2021	Scaled-up ViT (ViT-G)	JFT-3B	NS, MPL, CLIP, ALIGN, BiT-L (ResNet), ViT-H	ImageNet-90.45 INet V2-88.33 VTAB (light)-78.29	573
OPT: Open Pre-trained Transformer Language Models [39]	Zhang et al.	May 2022	Pre-trained NLP transformers, architecture followed GPT-3	BookCorpus, Stories, CCNews v2, CommonCrawl, DM Mathematics, Project Gutenberg, HackerNews, OpenSubtitles, OpenWebText2, USPTO, Wikipedia, dataset Baumgartner et al.	Dialogue Evaluations—Reddit 2.7B, BlenderBot1, R2C2 BlenderBot Hate Speech detection—Davinci CrowS-Pairs-GPT-3 SteroSet-Davinci Dialogue Responsible AI Evaluations—Reddit 2.7B, BlenderBot1, R2C2 BlenderBot	Outperforms Davinci in hate speech detection, best is few-shot (multiclass) with F1-score of 0.812, CroS-Pairs—better than GPT-3 only in two categories, Religion and Disability, with an accuracy of 68.6% and 76.7%, respectively, StereoSet—Almost same as Davinci	719

The table above summarizes key information from the history of the studies discussed in the previous section. In addition to the name of the study, the author and the date, the table also outlines the approach presented as well as the datasets evaluated upon, the models benchmarked against, and the obtained results. Finally, the number of citations attained by the paper as of the writing of this paper are also listed in order to emphasize the importance of some of the presented studies.

In general, it can be seen that a number of works have chosen to add or modify layers of the base transformer models, which has overall been seen to achieve good performance. Indeed, such an approach is seen in works such as those of Shaw et al. [19], Wang et al. [25], Sukhbaatar et al. [27], and Shazeer [28].

Another common approach for NLP tasks which has been shown to work really well is to increase the size of the model to a very large number of parameters and to pre-train it in an unsupervised fashion on a large corpus of data. This has been seen in numerous state-of-the-art models such as the GPT [23], BERT [17], GPT-2 [90], RoBERTa [26], T5 [22], and the GPT-3 [31] models, in the work of Radford et al. [37], and in the OPT model [39].

Yet another form of approaches involves the addition or modification of the loss functions associated with the transformer model. Such an approach was seen in the case of the work performed by Carion et al. [30].

When it comes to images, the general procedure followed by the previous studies was to split images into patches and apply position embeddings on these patches, much like what is done for texts. This was indeed the process followed by the Vision Transformer (ViT) [32]. Other vision models implemented varied decoders such as the work proposed by Zheng et al. [33]. Similarly, studies such as that by Chen et al. [34] make use of multiple pairs of heads and tails corresponding to different low-level vision tasks. The ViT-G model proposed by Zhai et al. [38] followed a procedure where the class token was removed and the non-linear projection before the final layer was removed.

4.2. Application-Based Implementations

4.2.1. Text-Based Applications

As can be seen from the types of models used for these applications, an important aspect in the implementation of text-based transformers is the encoder and the encoding of the input. Indeed, the model achieving the most widespread usage has been the BERT

model [17], which involves modifying the input encodings to make them bidirectional. The RoBERTa [26] builds upon this by adding an optimized pre-training process. Indeed, most of the other NLP-solving approaches have involved modifications to input encodings such as the TENER [69], ETC [67], and the Big Bird [68] models, thereby demonstrating the importance of encodings to the NLP process. Table 2 below displays a summary of notable transformer studies in the domain of NLP.

Table 2. A summary of the transformer studies in the domain of NLP.

Name of Paper	Author	Date	Proposed Model	Datasets	Models Benchmarked Against	Results
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [17]	Devlin et al.	2018	NLP Transformer	BooksCorpus, English Wikipedia	GLUE, SQuAD v1.1, SQuAD v2.0	BERT Large average score of 82.1 on GLUE testing
RoBERTa: A Robustly Optimized BERT Pre-training Approach [26]	Liu et al.	2019	Variant of BERT	BooksCorpus, English Wikipedia, CC-News, OpenWebText, Stories	BERT Large, XLNet Large Enseambles-ALICE, MT-DNN, XLNet	Best results: SQUAD 1.1-F1-94.6 Race-Middle-86.5 GLUE-SST-96.4
TENER: Adapting Transformer Encoder for Named Entity Recognition [69]	Yan et al.	2019	Sequence labeling (NER) transformer	English NER-CoNLL2003, OntoNotes 5.0 Chinese NER-Chinese part of OntoNotes 4.0, MSRA, Weibo NER, Resume NER	Chinese NER-BiLSTM, 1D-CNN, CAN-NER, Transformer english NER-BiLSTM-CRF, CNN-BiLSTM-CRF, BiLSTM-BiLSTM-CRF, CNN-BiLSTM-CRF, 1D-CNN, LM-LSTM-CRF, CRF + HSCRF, BiLSTM-BiLSTM-CRF, LS + BiLSTM-CRF, CN^3, GRN	F1-scores Chinese NER-Weibo-58.17, Resume-95.00, OntoNotes4.0-72.43, MSRA-92.74 English NER-ontoNotes 5.0-88.43, model+CNN-char get 91.45 for CoNLL 2003
ETC: Encoding Long and Structured Inputs in Transformers [67]	Ainslie et al.	2020	Variation of BERT-lifted weights from RoBERTa	BooksCorpus, English Wikipedia	BERT, RoBERTa	Leaderboard results SOTA (1ST) NQ long answer-77.78 HOTPOT QA SUP.F1-89.09 WikiHop-82.25 OpenKP-42.05
Big Bird: Transformers for Longer Sequences [68]	Zaheer et al.	2021	Variation of BERT	MLM	HGN, GSAN, ReflectionNet, RikiNet-v2, Fusion-in-decoder, SpanBERT, MRC-GCN, MultiHop, Longformer	Answering QA task-Best results (F1 SCORE) HotpotQA-Sup-89.1 NaturalIQ-LA-77.8 TriviaQA-Verified-92.4 WikiHop-82.3 (accuracy)

4.2.2. Image-Based Applications

Table 3 below successfully illustrates that the vision domain of transformers is very extensive and is used for many different kinds of applications such as image classification and segmentation. To date, the greatest model is the ViT [32], and many other significant models are based on improving its performance by tweaking its architecture, such as the study by Wang et al. [74] where they just modify the patch generation by avoiding overlap. The recent introduction of the work by Hamilton et al. [75] opens the door to unsupervised

segmentation, proven through their decent results of an accuracy of 76.1%. This solution would solve a lot of real-world-based problem applications, as those datasets are often unbalanced or have less amounts of labeled data. A concrete quantitative analysis across the previous studies is difficult to achieve due to the fact that all the authors report results on different datasets and also report different evaluation metrics.

Table 3. A summary of the transformer-related works in the domain of computer vision.

Name of Paper	Author	Date	Proposed Model	Datasets	Models Benchmarked Against	Results
Image Transformer [73]	Parmar et al.	2018	Attention Transformer	Cifar10	Generative Image Modeling-Pixel CNN, Row Pixel RNN, Gated Pixel CNN, Pixel CNN+, PixelSNAIL Further Inference-ResNet, srez GAN, Pixel Recursive	GIM-4.06 bits/dim CIFAR10-VAlidation, second best with 3.77 on ImageNet, very close to Pixel RNN with 3.86
An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale [32]	Dosovitsky et al.	2020	Vision Transformer (ViT)	Trained on ILSVRC-2012, ImageNet-21k, JFT Transfered on ReaL labels, Cifar10/100, Oxford-IIT Pets, Oxford Flowers-102	BiT-L (ResNet152x4), Noisy Student (EfficientNet-L2)	ImageNet-88.55-ViT-H ReaL-90.72-ViT-H CIFAR-10-99.50-ViT-H CIFAR-100-94.55-ViT-H Oxford-IIIT-Pets-97.56-ViT-H Oxford Flowers-99.74-ViT-L VTAB (19 tasks)-77.63-ViT-H
Training data-efficient image transformers and distillation through attention [35]	Touvron et al.	2020	Based on ViT	ImageNet	ResNet, RegNetY, EfficientNet, KDforAA, ViT (all versions)	DeiT-B 384/1000 epochs outperforms ViT and EffecientNet-85.2 acc
Feature Fusion Vision Transformer for Fine-Grained Visual Categorization [74]	Wang et al.	2021	Introduction of MAWS (mutual attention weight selection)	CUB-200-2011, Stanford Dogs and iNaturalist2017	CUB-200-2011-ResNet-50, RA-CNN, GP-256, MaxExt, DFL-CNN, NTS-Net, Cross-X, DCL, CIN, DBTNet, ASNet, S3N, FDL, PMG, API-Net, StackedLSTM, MMAL-Net, ViT, TransFG & PSM INaturalist2017-Resnet152, SSN, Huang et al., IncResNetv2, TASN, ViT, TransFG&PSM Standford Dogs-MaxEnt, FDL, RA-CNN, SEF, Cross-X, API-Net, ViT, TransFG & PSM	CUB-91.3% accuracy iNaturalist2017-68.5% Standford Dogs-92.4%
Unsupervised Semantic Segmentation By Distilling Feature Correspondences [75]	Hamilton et al.	2022	Unsupervised Semantic Segmentation Transformer (STEGO)	27 class COCOStuff, 27 classes of Cityscapes	ResNet50, MoCoV2, DINO, Deep Cluster, SIFT, Doersch et al., Isola et al. AC, InMARS, IIC, MDC, PiCIE, PiCIE + H	Unsupervised Accuracy-56.9, mIoU-28.2 Linear Probe Accuracy-76.1, mIoU-41.0

4.2.3. Miscellaneous Applications

Through Table 4 below it is well illustrated that the contributions towards transformer models are not just limited to the domain of NLP and images, but they have also been

recently used in audio and time series domains. Here, too, it is difficult to do a concrete quantitative analysis as the specific application domains of the works summarized above are all different. An interesting work to note is that of Koizumi et al. [77], which merges NLP analysis within the audio domain and is quite successful in outperforming the results of the traditional LSTM model that is usually used for such an application, with a best score of 52.1 for the BLUE-1 dataset. Dong et al. [76] achieve a WER score of 10.9 on the eval92 subset of the Wall Street Journal dataset, and Gong et al. [78] achieve their best results on the Speech commands v2 dataset with an accuracy of 98.11% without adding additional audio data while training. The second half of the table demonstrates different areas in the domain of time series using transformers. The domains illustrated are those of the Time Series Classification by Liu et al. [79], who were able to beat the state-of-the-art results on 7 out of 13 competitive datasets, those of the Time Series forecasting proposed by Zhou et al. [80], who achieved SOTA results in all 6 datasets, and the Time Series Anomaly Detection proposed by Tuli et al. [81], who also beat the SOTA results in their domain on 7 out of 10 competitive datasets.

Table 4. A summary of the transformer-related works in the audio and time series domains.

Name of Paper	Author	Date	Proposed Model	Datasets	Models Benchmarked Against	Results
Speech-Transformer: A No-Recurrence Sequence-To-Sequence Model For Speech Recognition [76]	Dong et al.	2018	Audio Transformer	Wall Street Journal dataset	CTC, seq2seq, seq2seq + deep convolutional, seq2seq + Unigram LS	WER 10.9 on eval92
A Transformer-based Audio-Captioning Model with Keyword Estimation [77]	Koizumi et al.	2020	Audio-Captioning Transformer (TRACKE)	Cllotho dataset	Baseline LSTM, Transformer from same challenge	Beats in BLUE-1 with 52.1, BLUE-2-30.9, BLUE-3-18.8, BLUE-4-10.8, CIDEr-25.8, METEOR-14.9, ROGUE-L-34.5, SPICE-9.7 SPIDEr-17.7
AST: Audio Spectrogram Transformer [78]	Gong et al.	2021	Audio Transformer	Converted pre-trained ViT to AST, used DeiT weights	AudioSet dataset-Baseline, PANN, PSLA single, PSLA Ensemble-S, PSLA Ensemble-M ESC-50, speech comands V2-SOTA-S (without additional audio data) SOTA-P (with additional audio data)	AudioSet-AST (Ensamble-M) -> Balanced mAP-0.378, full mAP-0.485 ESC-50-AST-P (trained using additional audio data)-95.6% Speech Commands V2-AST-S (trained without additional audio data)-98.11%
Gated Transformer Networks for Multivariate Time Series Classification [79]	Liu et al.	2021	Time series classification transformer	AUSLAN, ArabicDigits, CMUsubject1, CharacterTrajectories, ECG, JapeneseVowels, KickvsPunch, Libras, NetFlow, UWave, Wafer, WalkvsRun, PEMS	MLP, FCN, ResNet, Encoder, MCNN, t-LeNet, MCDCNN, Time-CNN, TWIESN	Best SOTA results in 7/13 datasets, with best scores of 100% for CMUsubject1, NetFlow and WalkvsRun
TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data [81]	Tuli et al.	2022	Anomaly Detection Time Series Transformer	NAB, UCR, MBA, SMAP, MSL, SWaT, WADI, SMD, MSDS	MERLIN, LSTM-NDT, DAGMM, OmniAnomaly, MSCRED, MAD-GAN, USAD, MTAD-GAT, CAE-M, GDN	Beats the SOTA results in 7/10 datasets for both f1score and AUC best score is AUC of 0.9994 and F1 of 0.9694 for the UCR dataset

5. Gaps and Future Work

As the above discussion illustrates, the realm of transformer architectures is one that has exploded with the new and existing works being rapidly proposed ever since Vaswani et al.'s revolutionary publication [1]. Figure 10 is presented to highlight the progression in the architecture and the complexity of transformer models ever since, with the architecture of notable transformer implementations visualized in order to give the reader a perspective of the rapid rise.

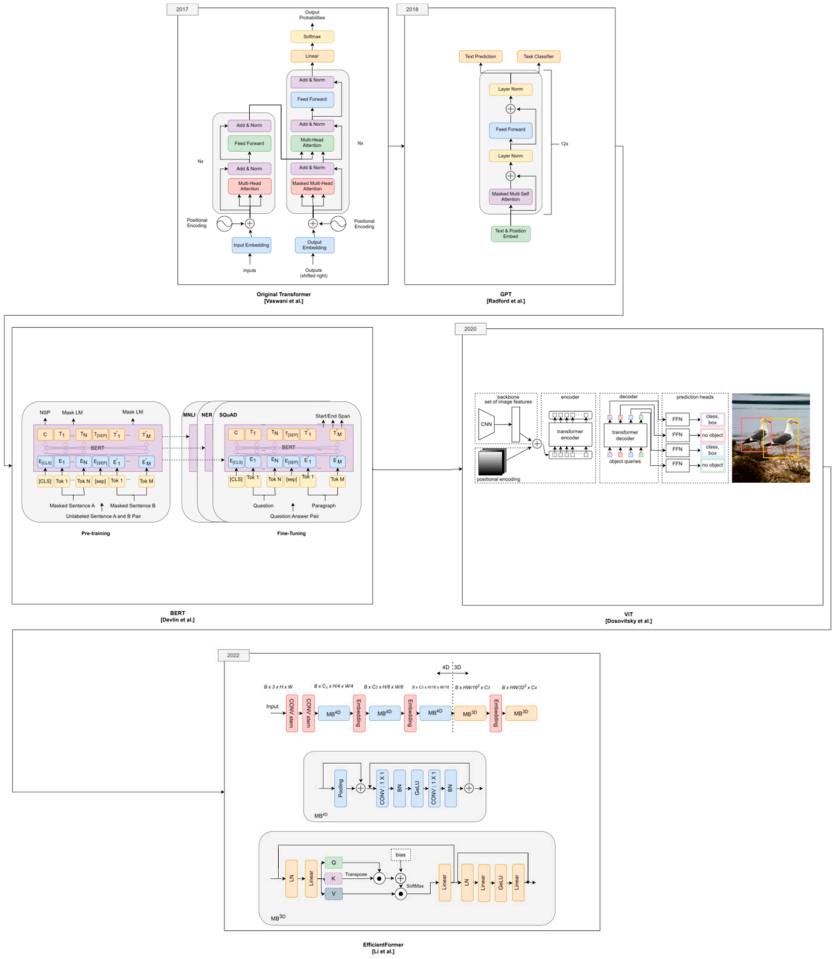

Figure 10. The progression of transformer architectures [1,17,23,32,41].

However, despite this rapid progression, certain gaps in the field remain. One major gap seen in contemporary research is that transformers generally have a quadratic computation and memory complexity due to their being required to model arbitrary long dependencies [91]. This has presented a major issue in the accessibility of the use of transformers and has led to a promising avenue of research aimed at simplifying the training process of transformer models [92]. Indeed, the Lite Transformer [29] discussed earlier was introduced with the intention of addressing this very issue, as were implementations such as the Longformer [93], Reformer [94], Linformer [95], Performer [96], and the OPT [39]. However, these models are a start to what is a vast potential research space in optimizing transformer-training procedures. This is a pressing issue, as many of the state-of-the-art models aim to simply increase a model's size (GPT-4, for instance) [97], and, therefore, make it impractical for that model to be used in many real-world applications.

Another interesting research issue is the problem of integrating all modalities without changing the architecture towards a single modality. Early implementations of this have been seen in models such as the Perceiver [98], which accepts all kinds of input but can only

generate fixed outputs such as class probabilities, and the Perceiver IO, which has flexible inputs and outputs but still relies on the specifics of the modalities, such as augmentation or position encoding, to properly learn [99]. This research area is ripe for expansion, as a model that is truly adaptable to anything would lead to massive progress in the field of deep learning and would broaden the scope of the real-world applications that could be improved with artificial intelligence.

A final research area which can be worked upon is that, generally, large amounts of data are needed to train a good transformer. This is less than ideal as many real-world applications do not contain adequate amounts of labeled data and therefore would not be able to leverage this powerful model. Promising research towards achieving this is that of the ViT-G [38], which reports having achieved few-shot learning by training with just 10 examples per class in the ImageNet dataset. More work needs to be done in this realm to truly make transformers accessible for wide implementations. A possible avenue to achieve this could be exploring ways to train transformers in a semi-supervised fashion [100]. With the successful exploration of these avenues of research, it might be possible to leverage the great power and achievements attained by transformers in real work applications which would affect our daily lives.

Author Contributions: Conceptualization, A.R.S. and I.Z.; methodology, A.R.S., I.Z. and D.S.; investigation, A.R.S. and I.Z.; resources, I.Z.; writing—original draft preparation, A.R.S. and D.S.; writing—review and editing, I.Z.; visualization, D.S.; supervision, I.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The work in this paper was supported, in part, by the Open Access Program from the American University of Sharjah [grant number: OAPCEN-1410-E00291].

Acknowledgments: This paper represents the opinions of the authors and does not mean to represent the position or opinions of the American University of Sharjah.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 6000–6010.
2. Li, X.; Metsis, V.; Wang, H.; Ngu, A.H.H. TTS-GAN: A Transformer-Based Time-Series Generative Adversarial Network. In *Artificial Intelligence in Medicine*; Michalowski, M., Abidi, S.S.R., Abidi, S., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2022; Volume 13263, pp. 133–143. ISBN 978-3-031-09341-8.
3. Myers, D.; Mohawesh, R.; Chellaboina, V.I.; Sathvik, A.L.; Venkatesh, P.; Ho, Y.-H.; Henshaw, H.; Alhawawreh, M.; Berdik, D.; Jararweh, Y. Foundation and large language models: Fundamentals, challenges, opportunities, and social impacts. *Cluster Comput.* **2024**, *27*, 1–26. [CrossRef]
4. Liu, Y.; Zhang, Y.; Wang, Y.; Hou, F.; Yuan, J.; Tian, J.; Zhang, Y.; Shi, Z.; Fan, J.; He, Z. A Survey of Visual Transformers. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–21. [CrossRef] [PubMed]
5. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A Survey on Vision Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 87–110. [CrossRef] [PubMed]
6. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
7. Rumelhart, D.E.; McClelland, J.L. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*; MIT Press: Cambridge, MA, USA, 1987; pp. 318–362. ISBN 978-0-262-29140-8.
8. Zeyer, A.; Bahar, P.; Irie, K.; Schlüter, R.; Ney, H. A Comparison of Transformer and LSTM Encoder Decoder Models for ASR. In Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Singapore, 14–18 December 2019; pp. 8–15.
9. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 17–19 June 2013; Dasgupta, S., McAllester, D., Eds.; PMLR: Atlanta, GA, USA, 2013; Volume 28, pp. 1310–1318.
10. Kim, Y.; Denton, C.; Hoang, L.; Rush, A.M. Structured Attention Networks. *arXiv* **2017**, arXiv:1702.00887.
11. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *arXiv* **2014**, arXiv:1409.3215.
12. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learning Syst.* **2017**, *28*, 2222–2232. [CrossRef] [PubMed]

13. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *ACM Comput. Surv.* **2021**, *54*, 1–41. [CrossRef]
14. Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; Titov, I. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. *arXiv* **2019**, arXiv:1905.09418.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
16. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer Normalization. *arXiv* **2016**, arXiv:1607.06450.
17. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. 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, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Pittsburgh, PA, USA, 2019; Volume 1, (Long and Short Papers), pp. 4171–4186.
18. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional Sequence to Sequence Learning. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Precup, D., Teh, Y.W., Eds.; PMLR: London, UK, 2017; Volume 70, pp. 1243–1252.
19. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-Attention with Relative Position Representations. *arXiv* **2018**, arXiv:1803.02155.
20. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, 71. [CrossRef] [PubMed]
21. Attention Is All You Need Search Results. Available online: https://scholar.google.ae/scholar?q=Attention+Is+All+You+Need&hl=en&as_sdt=0&as_vis=1&oi=scholart (accessed on 5 June 2022).
22. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
23. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://api.semanticscholar.org/CorpusID:49313245 (accessed on 14 May 2024).
24. Radford, A.; Jozefowicz, R.; Sutskever, I. Learning to Generate Reviews and Discovering Sentiment. *arXiv* **2017**, arXiv:1704.01444.
25. Wang, Q.; Li, B.; Xiao, T.; Zhu, J.; Li, C.; Wong, D.F.; Chao, L.S. Learning Deep Transformer Models for Machine Translation. *arXiv* **2019**, arXiv:1906.01787.
26. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
27. Sukhbaatar, S.; Grave, E.; Lample, G.; Jegou, H.; Joulin, A. Augmenting Self-attention with Persistent Memory. *arXiv* **2019**, arXiv:1907.01470.
28. Shazeer, N. GLU Variants Improve Transformer. *arXiv* **2020**, arXiv:2002.05202.
29. Wu, Z.; Liu, Z.; Lin, J.; Lin, Y.; Han, S. Lite Transformer with Long-Short Range Attention. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
30. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2020; Volume 12346, pp. 213–229. ISBN 978-3-030-58451-1.
31. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 1877–1901.
32. Kolesnikov, A.; Dosovitskiy, A.; Weissenborn, D.; Heigold, G.; Uszkoreit, J.; Beyer, L.; Minderer, M.; Dehghani, M.; Houlsby, N.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4 May 2021.
33. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.; et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; IEEE: Piscataway, NJ, USA, 2022; pp. 6877–6886.
34. Chen, H.; Wang, Y.; Guo, T.; Xu, C.; Deng, Y.; Liu, Z.; Ma, S.; Xu, C.; Xu, C.; Gao, W. Pre-Trained Image Processing Transformer. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 12294–12305.
35. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jegou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021; Meila, M., Zhang, T., Eds.; PMLR: London, UK, 2021; Volume 139, pp. 10347–10357.
36. Fedus, W.; Zoph, B.; Shazeer, N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *arXiv* **2022**, arXiv:2101.03961.
37. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. *arXiv* **2021**, arXiv:2103.00020.
38. Zhai, X.; Kolesnikov, A.; Houlsby, N.; Beyer, L. Scaling Vision Transformers. *arXiv* **2021**, arXiv:2106.04560.
39. Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X.V.; et al. OPT: Open Pre-trained Transformer Language Models. *arXiv* **2022**, arXiv:2205.01068.

40. Wang, H.; Ma, S.; Dong, L.; Huang, S.; Zhang, D.; Wei, F. DeepNet: Scaling Transformers to 1000 Layers 2022. *arXiv* **2022**, arXiv:2203.00555. [CrossRef]
41. Li, Y.; Yuan, G.; Wen, Y.; Hu, J.; Evangelidis, G.; Tulyakov, S.; Wang, Y.; Ren, J. Efficientformer: Vision transformers at mobilenet speed. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 12934–12949.
42. Li, Y.; Hu, J.; Wen, Y.; Evangelidis, G.; Salahi, K.; Wang, Y.; Tulyakov, S.; Ren, J. Rethinking Vision Transformers for MobileNet Size and Speed. In Proceedings of the IEEE International Conference on Computer Vision, Paris, France, 2–3 October 2023.
43. Meng, L.; Li, H.; Chen, B.-C.; Lan, S.; Wu, Z.; Jiang, Y.-G.; Lim, S.-N. AdaViT: Adaptive Vision Transformers for Efficient Image Recognition. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 12299–12308.
44. Yin, H.; Vahdat, A.; Alvarez, J.M.; Mallya, A.; Kautz, J.; Molchanov, P. A-ViT: Adaptive Tokens for Efficient Vision Transformer. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Orleans, LA, USA, 18–24 June 2022; pp. 10799–10808.
45. Pope, R.; Douglas, S.; Chowdhery, A.; Devlin, J.; Bradbury, J.; Heek, J.; Xiao, K.; Agrawal, S.; Dean, J. Efficiently Scaling Transformer Inference. *Proc. Mach. Learn. Syst.* **2023**, *5*.
46. Zhang, J.; Peng, H.; Wu, K.; Liu, M.; Xiao, B.; Fu, J.; Yuan, L. MiniViT: Compressing Vision Transformers with Weight Multiplexing. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Orleans, LA, USA, 18–24 June 2022; pp. 12135–12144.
47. Yu, H.; Wu, J. A unified pruning framework for vision transformers. *Sci. China Inf. Sci.* **2023**, *66*, 179101. [CrossRef]
48. Zhu, Y.; Kiros, R.; Zemel, R.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; Fidler, S. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
49. Bojar, O.; Buck, C.; Federmann, C.; Haddow, B.; Koehn, P.; Leveling, J.; Monz, C.; Pecina, P.; Post, M.; Saint-Amand, H.; et al. Findings of the 2014 Workshop on Statistical Machine Translation. In Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, MD, USA, 26–27 June 2014; Association for Computational Linguistics: Pittsburgh, PA, USA, 2014; pp. 12–58.
50. Lim, D.; Hohne, F.; Li, X.; Huang, S.L.; Gupta, V.; Bhalerao, O.; Lim, S.-N. Large Scale Learning on Non-Homophilous Graphs: New Benchmarks and Strong Simple Methods. *arXiv* **2021**, arXiv:2110.14446. [CrossRef]
51. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Li, F.F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
52. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2014; Volume 8693, pp. 740–755. ISBN 978-3-319-10601-4.
53. Caruana, R. Multitask Learning. *Mach. Learn.* **1997**, *28*, 41–75. [CrossRef]
54. Taylor, W.L. "Cloze procedure": A new tool for measuring readability. *J. Q.* **1953**, *30*, 415–433. [CrossRef]
55. Schwartz, R.; Sap, M.; Konstas, I.; Zilles, L.; Choi, Y.; Smith, N.A. Story Cloze Task: UW NLP System. In Proceedings of the LSDSem 2017, Valencia, Spain, 3 April 2017.
56. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Brussels, Belgium, 1 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 353–355.
57. Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; Hovy, E. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 785–794.
58. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; Association for Computational Linguistics: Austin, TX, USA, 2016; pp. 2383–2392.
59. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language Modeling with Gated Convolutional Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 933–941.
60. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
61. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:1506.01497. [CrossRef]
62. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. 2009. Volume 7, pp. 32–33. Available online: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf (accessed on 14 May 2024).
63. Parkhi, O.M.; Vedaldi, A.; Zisserman, A.; Jawahar, C.V. Cats and Dogs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.

64. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene Parsing through ADE20K Dataset. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017; pp. 5122–5130.
65. Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.-G.; Lee, S.-W.; Fidler, S.; Urtasun, R.; Yuille, A. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
66. Patwardhan, N.; Marrone, S.; Sansone, C. Transformers in the Real World: A Survey on NLP Applications. *Information* **2023**, *14*, 242. [CrossRef]
67. Ainslie, J.; Ontanon, S.; Alberti, C.; Cvicek, V.; Fisher, Z.; Pham, P.; Ravula, A.; Sanghai, S.; Wang, Q.; Yang, L. ETC: Encoding Long and Structured Inputs in Transformers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; Association for Computational Linguistics: Pittsburgh, PA, USA, 2020; pp. 268–284.
68. Zaheer, M.; Guruganesh, G.; Dubey, K.A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. Big bird: Transformers for longer sequences. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17283–17297.
69. Yan, H.; Deng, B.; Li, X.; Qiu, X. TENER: Adapting Transformer Encoder for Named Entity Recognition. *arXiv* **2019**, arXiv:1911.04474.
70. Zhou, Q.; Yang, N.; Wei, F.; Tan, C.; Bao, H.; Zhou, M. Neural Question Generation from Text: A Preliminary Study. *arXiv* **2017**, arXiv:1704.01792.
71. Miwa, M.; Bansal, M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; Association for Computational Linguistics: Pittsburgh, PA, USA, 2016; pp. 1105–1116.
72. Fragkou, P. Applying named entity recognition and co-reference resolution for segmenting English texts. *Prog. Artif. Intell.* **2017**, *6*, 325–346. [CrossRef]
73. Parmar, N.J.; Vaswani, A.; Uszkoreit, J.; Kaiser, L.; Shazeer, N.; Ku, A.; Tran, D. Image Transformer. In Proceedings of the International Conference on Machine Learning (ICML), Vienna, Austria, 10–15 July 2018.
74. Wang, J.; Yu, X.; Gao, Y. Feature Fusion Vision Transformer for Fine-Grained Visual Categorization. *arXiv* **2021**, arXiv:2107.02341.
75. Hamilton, M.; Zhang, Z.; Hariharan, B.; Snavely, N.; Freeman, W.T. Unsupervised Semantic Segmentation by Distilling Feature Correspondences. *arXiv* **2022**, arXiv:2203.08414.
76. Dong, L.; Xu, S.; Xu, B. Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5884–5888.
77. Koizumi, Y.; Masumura, R.; Nishida, K.; Yasuda, M.; Saito, S. A Transformer-based Audio Captioning Model with Keyword Estimation. *arXiv* **2020**, arXiv:2007.00222.
78. Gong, Y.; Chung, Y.-A.; Glass, J. AST: Audio Spectrogram Transformer. *arXiv* **2021**, arXiv:2104.01778.
79. Liu, M.; Ren, S.; Ma, S.; Jiao, J.; Chen, Y.; Wang, Z.; Song, W. Gated Transformer Networks for Multivariate Time Series Classification. *arXiv* **2021**, arXiv:2103.14438.
80. Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; Jin, R. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. *arXiv* **2022**, arXiv:2201.12740.
81. Tuli, S.; Casale, G.; Jennings, N.R. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. *arXiv* **2022**, arXiv:2201.07284. [CrossRef]
82. He, K.; Gan, C.; Li, Z.; Rekik, I.; Yin, Z.; Ji, W.; Gao, Y.; Wang, Q.; Zhang, J.; Shen, D. Transformers in medical image analysis. *Intell. Med.* **2023**, *3*, 59–78. [CrossRef]
83. Sajun, A.R.; Zualkernan, I.; Sankalpa, D. Investigating the Performance of FixMatch for COVID-19 Detection in Chest X-rays. *Appl. Sci.* **2022**, *12*, 4694. [CrossRef]
84. Ziani, S. Enhancing fetal electrocardiogram classification: A hybrid approach incorporating multimodal data fusion and advanced deep learning models. *Multimed. Tools Appl.* **2023**, *83*, 55011–55051. [CrossRef]
85. Ziani, S.; Farhaoui, Y.; Moutaib, M. Extraction of Fetal Electrocardiogram by Combining Deep Learning and SVD-ICA-NMF Methods. *Big Data Min. Anal.* **2023**, *6*, 301–310. [CrossRef]
86. Li, J.; Chen, J.; Tang, Y.; Wang, C.; Landman, B.A.; Zhou, S.K. Transforming medical imaging with Transformers? A comparative review of key properties, current progresses, and future perspectives. *Med. Image Anal.* **2023**, *85*, 102762. [CrossRef] [PubMed]
87. Shamshad, F.; Khan, S.; Zamir, S.W.; Khan, M.H.; Hayat, M.; Khan, F.S.; Fu, H. Transformers in medical imaging: A survey. *Med. Image Anal.* **2023**, *88*, 102802. [CrossRef]
88. Fournier, Q.; Caron, G.M.; Aloise, D. A Practical Survey on Faster and Lighter Transformers. *ACM Comput. Surv.* **2023**, *55*, 1–40. [CrossRef]
89. Wang, W.; Zhang, J.; Cao, Y.; Shen, Y.; Tao, D. Towards Data-Efficient Detection Transformers. In Proceedings of the Computer Vision—ECCV 2022; Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T., Eds.; Springer Nature Switzerland: Cham, Switzerland, 2022; pp. 88–105.
90. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.

91. Liu, P.J.; Saleh, M.; Pot, E.; Goodrich, B.; Sepassi, R.; Kaiser, L.; Shazeer, N. Generating Wikipedia by Summarizing Long Sequences. *arXiv* **2018**, arXiv:1801.10198.
92. Tay, Y.; Dehghani, M.; Bahri, D.; Metzler, D. Efficient Transformers: A Survey. *ACM Comput. Surv.* **2023**, *55*, 1–28. [CrossRef]
93. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *arXiv* **2020**, arXiv:2004.05150.
94. Kitaev, N.; Kaiser, Ł.; Levskaya, A. Reformer: The Efficient Transformer. *arXiv* **2020**, arXiv:2001.04451.
95. Wang, S.; Li, B.Z.; Khabsa, M.; Fang, H.; Ma, H. Linformer: Self-Attention with Linear Complexity. *arXiv* **2020**, arXiv:2006.04768.
96. Choromanski, K.; Likhosherstov, V.; Dohan, D.; Song, X.; Gane, A.; Sarlos, T.; Hawkins, P.; Davis, J.; Mohiuddin, A.; Kaiser, L.; et al. Rethinking Attention with Performers. *arXiv* **2021**, arXiv:2009.14794.
97. OpenAI GPT-4 Technical Report 2023. *arXiv* **2023**, arXiv:2303.08774. [CrossRef]
98. Jaegle, A.; Gimeno, F.; Brock, A.; Zisserman, A.; Vinyals, O.; Carreira, J. Perceiver: General Perception with Iterative Attention. *arXiv* **2021**, arXiv:2103.03206.
99. Jaegle, A.; Borgeaud, S.; Alayrac, J.-B.; Doersch, C.; Ionescu, C.; Ding, D.; Koppula, S.; Zoran, D.; Brock, A.; Shelhamer, E.; et al. Perceiver IO: A General Architecture for Structured Inputs & Outputs. *arXiv* **2022**, arXiv:2107.14795.
100. Weng, Z.; Yang, X.; Li, A.; Wu, Z.; Jiang, Y.-G. Semi-supervised vision transformers. In Proceedings of the ECCV 2022, Tel Aviv, Israel, 23–27 October 2022.