information

Review

A Systematic Review of Transformer-Based Pre-Trained Language Models through Self-Supervised Learning

Evans Kotei and Ramkumar Thirunavukarasu *

School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, India; evans.kotei2019@vitstudent.ac.in

* Correspondence: ramkumar.thirunavukarasu@vit.ac.in; Tel.: +91-944-242-1674

Abstract: Transfer learning is a technique utilized in deep learning applications to transmit learned inference to a different target domain. The approach is mainly to solve the problem of a few training datasets resulting in model overfitting, which affects model performance. The study was carried out on publications retrieved from various digital libraries such as SCOPUS, ScienceDirect, IEEE Xplore, ACM Digital Library, and Google Scholar, which formed the Primary studies. Secondary studies were retrieved from Primary articles using the backward and forward snowballing approach. Based on set inclusion and exclusion parameters, relevant publications were selected for review. The study focused on transfer learning pretrained NLP models based on the deep transformer network. BERT and GPT were the two elite pretrained models trained to classify global and local representations based on larger unlabeled text datasets through self-supervised learning. Pretrained transformer models offer numerous advantages to natural language processing models, such as knowledge transfer to downstream tasks that deal with drawbacks associated with training a model from scratch. This review gives a comprehensive view of transformer architecture, self-supervised learning and pretraining concepts in language models, and their adaptation to downstream tasks. Finally, we present future directions to further improvement in pretrained transformer-based language models.

Keywords: transformer network; transfer learning; pretraining; natural language processing; language models

Citation: Kotei, E.; Thirunavukarasu, R. A Systematic Review of Transformer-Based Pre-Trained Language Models through Self-Supervised Learning. *Information* **2023**, *14*, 187. https://doi.org/10.3390/info14030187

Academic Editors: Katsuhide Fujita and Paulo Quaresma

Received: 30 January 2023
Revised: 9 March 2023
Accepted: 14 March 2023
Published: 16 March 2023

1. Introduction

The transformer network is a novel architecture that produces optimal performance in language processing applications. Its success depends on its abstraction of long-range dependencies from large datasets. This transformer network does not require hand-crafted features, which is a bottleneck in machine-learning models. The advancements in computer hardware, availability of larger datasets, and advanced word embedding algorithms have increased the adaptation of DL models for vision tasks [1] and solving NLP problems [2]. Transfer learning techniques provide optimal results in NLP tasks through pretraining. Some of these include language representation [3], natural language generation [4], language understanding [5–8], text reconstruction [9], and abstractive text summarization [10,11]. The learning techniques require more labelled and annotated data to yield good performance. It is a drawback because much time is required to generate these annotated datasets. Transfer learning [12] is one technique used to address this drawback. In transfer learning, models trained on larger datasets, such as ImageNet [13], are used as the base model to train target models with few datasets. Classification and detection problems in vision processing are mostly solved based on transfer learning [14–17] in vision processing tasks. Transfer learning (TL) for image registration and segmentation cannot be left out depending on the outstanding performance [18,19].

For example, a base model VGG-16 [20] extracts information for all tasks, and the knowledge gained during the training is transferred to downstream tasks through fine-tuning [21] with optimal performance [22–24]. The approach eliminates the problem of

overfitting, which is common in deep learning applications when the training dataset is few. Despite the outstanding performance of RNN and CNN models in sequential and vision tasks, they suffer in modelling log–range dependencies and locality bias. The Transformer network [25] deals with these drawbacks. Based on the encoder/decoder layers and self-attention in the transformer network, this ensures the parallelization of a long-range relationship. As mentioned earlier, the backbone of deep learning models is labelled data, which are few in quantity. Notwithstanding, unlabeled datasets are available. Transformer networks can learn from unlabeled datasets through a self-supervised learning approach with pseudo-supervision. Through transfer learning, several models for NLP tasks have emerged [8–10]. The application of transfer learning for NLP applications transcends through multiple disciplines such as financial communication [26], public law [27,28], task-oriented dialogue [29], academia [30–32], and the medical sector [33–35].

This study discusses the efficacy of pre-trained TL approaches for language processing with the under-listed highlights.

- An overview of the transformer network architecture and its core concepts.
- Self-supervised learning based on unlabeled datasets for transformer-based pretrained models.
- Explains the fundamental principles of pre-training techniques and activities for downstream adaption.
- Future trends for pretrained transformer-based language models.

The study follows the below structure:

Retrieving publications for this study following PRISMA reporting standards are in Section 2. The description of the core structure of the transformer network is in Section 3. Self-supervised learning and its application in pretraining are explained in Section 4. Section 5 discusses the various pretrained models proposed in the literature. Pretraining downstream tasks with the transformer network are in Section 6. Challenges with future directions for efficient pretrained models are in Section 7. Section 8 concludes the review.

2. Materials and Methods

This section describes methods and techniques employed in writing this paper, following PRISMA reporting standards.

2.1. Review Planning

This review was planned and executed by, first, formulating research questions that address the set objectives of the study. Based on the research objectives, we set up a search strategy and criteria, which served as a guide to include or reject papers or publications.

Objectives and Research Questions

Deep TL passes on knowledge gained from one domain and is transferred to another target domain to deal with the problem of overfitting due to a few training datasets. Pretraining is a transfer learning technique widely used in language processing (LP) tasks. The BERT pretrained model has given birth to other variants to handle different language tasks instead of the traditional deep learning algorithms such as RNN due to its efficacy in dealing with long-range sequences. This review seeks to understand the various pretrained language models proposed in the literature. The following research questions aid in achieving the aim of this paper:

RQ1: What are the various transformer-based pretrained models available for NLP processing?
RQ2: What are the various pretraining techniques available?
RQ3: What datasets or corpora are used for pretraining language models?
RQ4: What are the challenges associated with transformer-based language model pretraining based on self-supervised learning?
RQ5: How and when to choose a pretraining model for an NLP task?

2.2. Search Strategy

We searched for relevant publications or literature about NLP applications based on transformer networks for pretrained language models. The search strings for article retrieval were formulated based on study objectives and research questions. Three main categories of keywords "transformer-based natural language processing", "pretrained language models", and "transfer learning approaches for natural language processing" were formulated. The selected set of keywords used in the publication search is in Table 1.

Table 1. Search keywords.

Category	Keyword
Transformer-based natural language processing	Transformer network for NLP application, natural language processing, attention-based NLP models, representation learning from transformers
Pretrained language models	BERT models for natural language processing, intermediate fine tuning on language models, pretraining text models.
Transfer learning approaches for NLP	NLP-based self-supervised learning, transfer learning for language tasks, deep transfer learning for NLP

The keywords were linked using Boolean operators such as "OR" and "AND" to complete the search string for retrieving articles. The search strings had to be modified based on individual database requirements without compromising the selected keywords. This review considered publications on transformer networks proposed for NLP applications from 2018 to 2022. Electronic databases such as SCOPUS, Google Scholar, SpringerLink IEEE Xplore, ACM Digital Library, and ScienceDirect were the sources of articles used in this study.

2.2.1. Snowballing Approach

The snowballing technique [36] was used in retrieving research articles in conjunction with database searches. Articles retrieved from the various digital databases (Primary studies) assisted in getting additional publications using the reference list (backwards snowballing—BSB) and citations (forward snowballing—FSB). The approach helped in touching on all the relevant articles needed for this study without missing some key publications.

2.2.2. Screening Criteria

The retrieved publications from databases and also through snowballing approach for this study were screened by two authors (Ramkumar T. and Evans Kotei). Only transformer-based publications were considered during the screening process.

2.2.3. Exclusion Criteria

This review does not include publications with less than four pages, symposium papers, conference keynotes, and tutorials. Publications downloaded multiple times due to articles having multiple database indexing were identified and removed accordingly. Only relevant articles were selected for the study to form the Primary studies. Figure 1 is the PRISMA flow diagram to explain the search process.

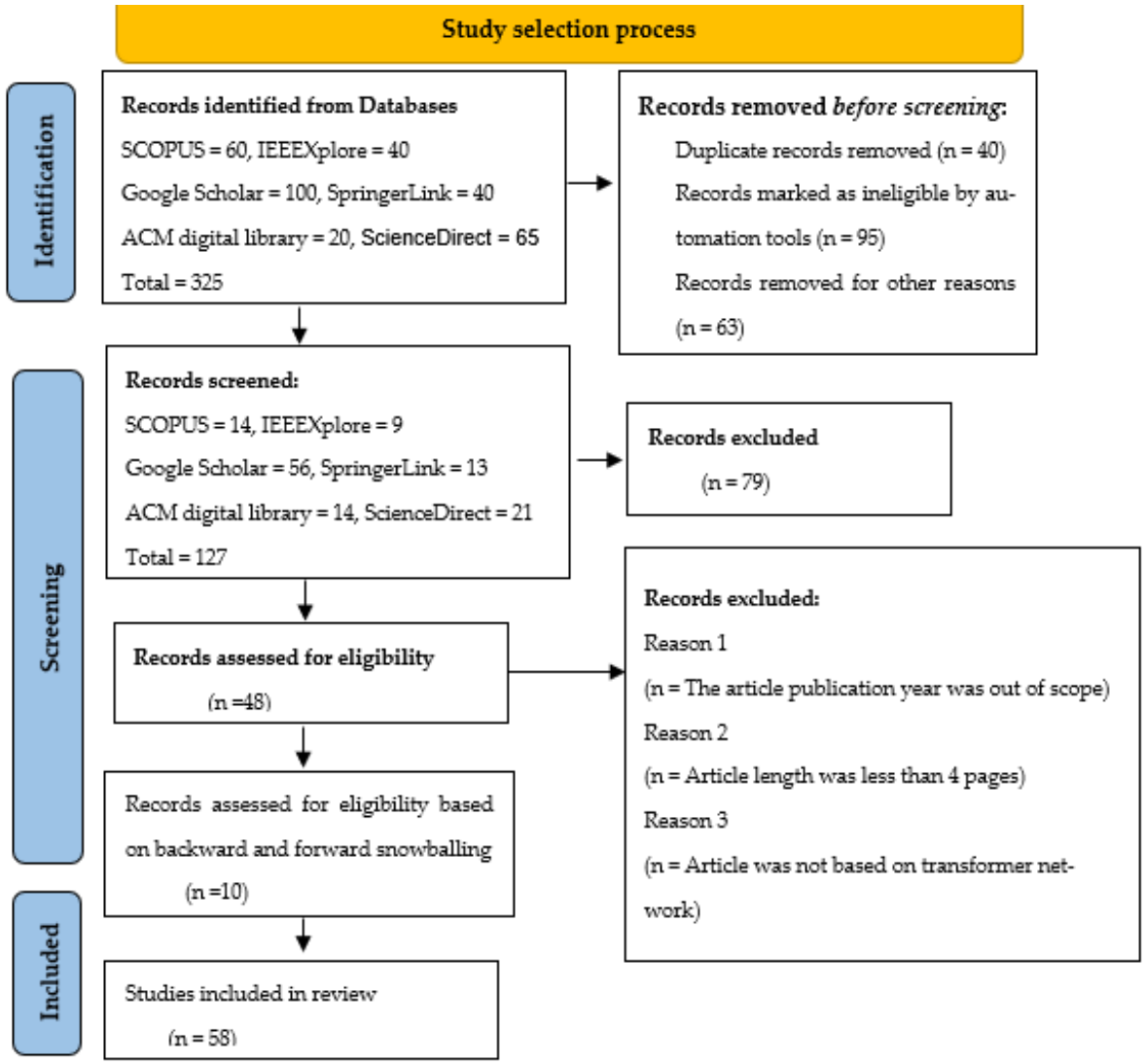

Figure 1. Article retrieval and selection process based on PRISMA reporting standard.

The selected 58 publications consist of 26 (46%) conference publications and 32 (54%) journal articles published from 2018 to 2022. A summary of retrieved articles is in Figure 2.

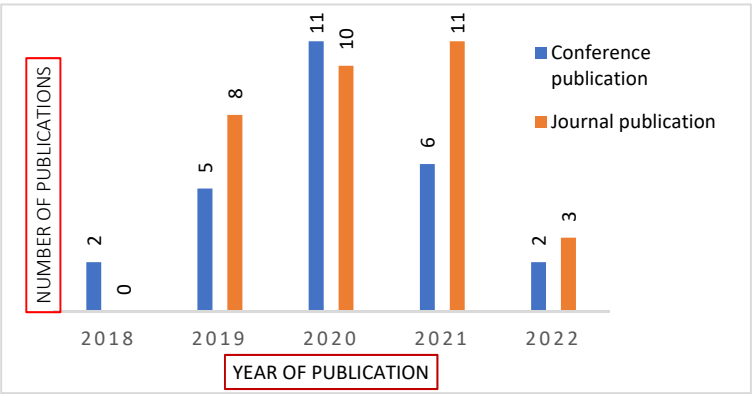

Figure 2. Selected article distribution.

3. Transformer Network

The transformer network has two parts (the encoder and the decoder), with self-attention for neural sequence transduction [37,38]. The encoder architecture in the transformer network handles symbolic relationships of an input categorization ($x1, \ldots, xn$) to an incessant relation, $z = (z1, \ldots, zn)$. On the other hand, the decoder part of the transformer model engenders an output sequence ($y1, \ldots, ym$) one after the other. Each stage is auto-degenerating and exploits the earlier input as supplementary to the next word. Figure 3 is the transformer network.

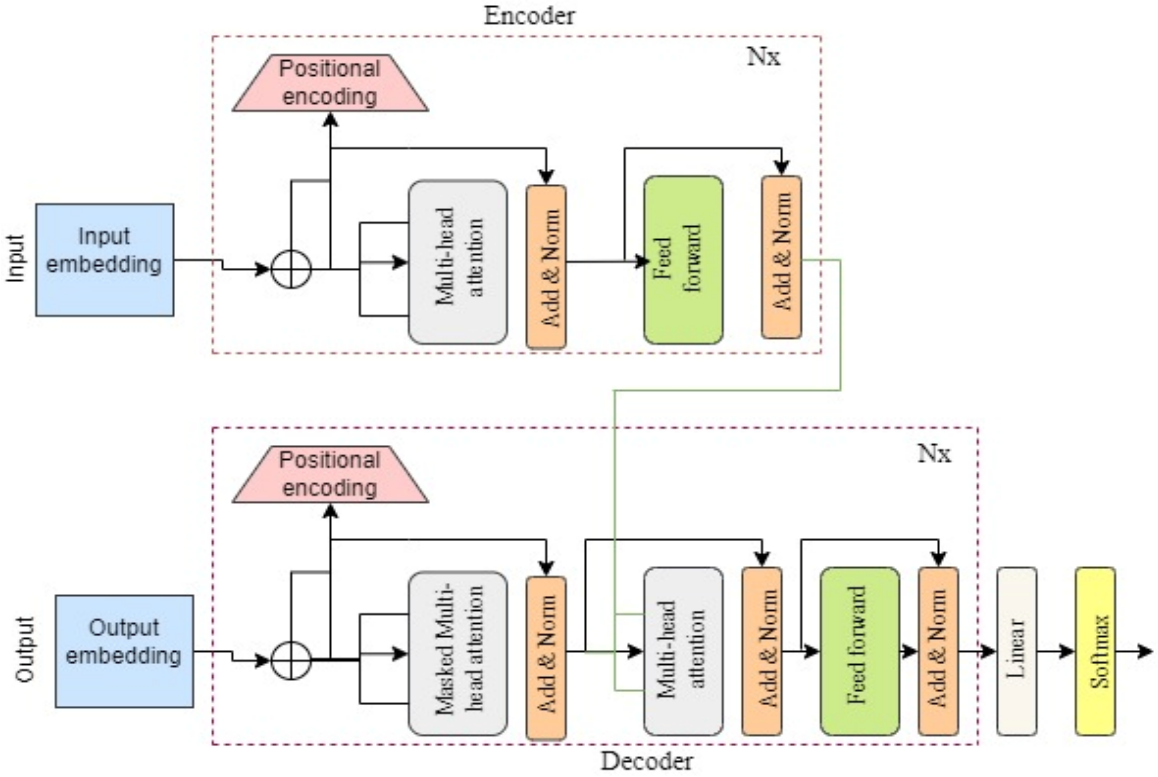

Figure 3. Transformer model (An input sequence is converted into a series of continuous representations by the encoder component of the transformer's architecture before being supplied to the decoder. The decoder combines the encoder's output with the decoder's output from the preceding time step to produce an output sequence).

3.1. Encoder and Decoder Stacks

A position-wise fully connected feed-forward network and multi-head self-attention form part of the encoder/decoder layers. Additionally, there is a residual layer with a normalization function to ensure the models' optimal performance [25].

3.2. Attention

Attention models produce good results through their query (Q), key (K), and value-pairs (V), which are in vector form. Predictions are based on these three variables, as shown in Figure 3. Attention uses the scaled dot-attention function for value localization based on two input pairs (queries and keys). The dimensions of the input pairs are denoted *dk* (*dimensional key*) and *dv* (dimension value). Figure 4 is scaled dot-attention and multi-head attention.

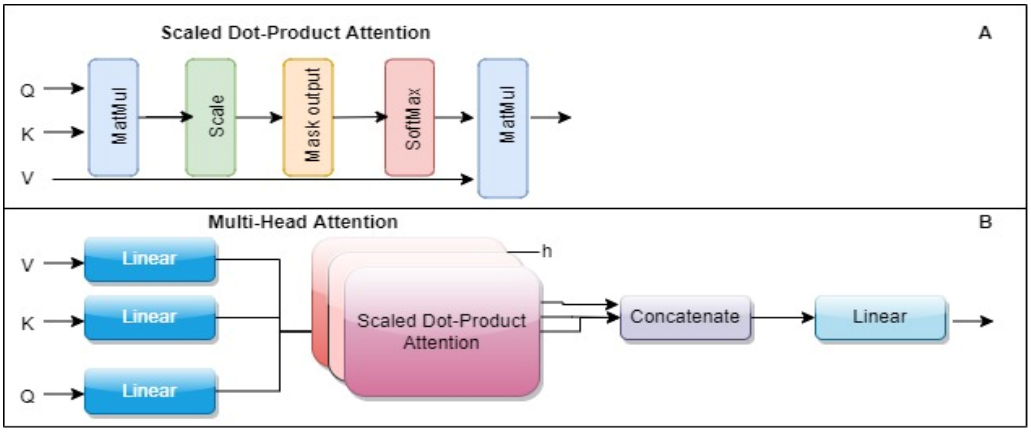

Figure 4. (**A**,**B**) Attention mechanism in transformer network (This is performed in the multi-head attention mechanism, which concurrently implements several single attention functions by masking the output of the scaled multiplication of the *Q* and *K* matrices. The multi-head self-attention is comparable to the encoder's first sublayer. This multi-head mechanism receives the keys and values from the encoder's output and the queries from the preceding decoder sublayer on the decoder side. The decoder then focuses on every word in the input sequence).

The weight value is the dot-product of *Q* and $K/\sqrt{dk}$ and a SoftMax function. There are two kinds of attention (additive attention and dot-product attention) [39]. Comparatively, the dot-product attention mechanism is faster and more efficient because of the multiplication code in its architecture. When the *dk* value is small, the additive attention performs better than the dot product attention [40]. This is because an increase in the *dk* values due to dot product computation pushes SoftMax to a lesser gradient space. Masked Multi-Head Attention within the transformer architecture is defined by:

$$hi = Attention\ (QW_i^Q, KW_i^K, VW_i^V),$$

$$H = Concat\ (h_1, h_2, \ldots, h_n),$$

$$O = HW_h, \text{ where, } W_i^Q \in R^{dmodel \times d}, W_i^K \in R^{dmodel \times d}, \text{ and } W_i^V \in R^{dmodel \times dv}.$$

The output from the attention model is h_i is concatenated together and projected to the same magnitude by multiplying it with W_h, such that $W_h \in R^{(n \times dv) \times dmodel}$ and $O \in R^{L \times dmodel}$.

The attention mechanism is used in several tasks [41–43].

4. Self-Supervised Learning (SSL)

This is a novel technique of acquiring collective information or knowledge based on unlabeled datasets through pseudo-supervision. Even though self-supervised learning is new, its patronage cuts across several disciplines, such as NLP, computer vision, speech recognition, and robotics [44–48].

4.1. Why Self-Supervised Learning?

Most deep learning applications are trained on supervised learning, which requires human-annotated instances to learn. Supervised learning depends on labelled data, but good and quality data are hard to come by, specifically for complex issues such as object detection [49,50] and image segmentation [51,52], where detailed information is required. Meanwhile, the unlabeled data are readily accessible in abundance. The advantage of a supervised learning application is that models perform very well on specific datasets. Generating human-annotated labels is a cumbersome process and requires a domain expert, who is scarce and not readily available, especially in the medical sector. Models trained through supervised learning suffer from generalization errors and fake correlations because the model only knows the training pattern and struggles with the unseen dataset. Despite

the supervised learning approach being dominant in developing deep learning applications, it has some drawbacks. Below is a summary of them:

- Supervised learning requires a human-annotated dataset, which is expensive to generate, especially a domain-specific dataset.
- Poor generalization because the model tries to memorize the training data and suffers from unseen data during classification.
- Limitation of deep learning applications in domains where labelled data are less example, in the medical health sector.

Based on these drawbacks, some solutions have been provided through extensive research. One is self-supervised learning, which eliminates the requirement for human-annotated labels. The labels are generated automatically by the algorithm. The intuition behind self-supervised learning is to study representations from a given unlabeled dataset using self-supervision and fine-tuned with a few labelled datasets for the supervised downstream task such as classification, segmentation or object detection.

4.2. Self-Supervised Learning—Explained

In this type of learning, a model learns from part of the input dataset and evaluates itself with the other part of the dataset. The basic idea for SSL is to transform the unsupervised problem into a supervised problem by generating some auxiliary pre-text tasks for the model from the input data such that while solving the problem, the model learns the underlying structure of the data. Transformer models such as BERT [5], ELECTRA [9], and T5 [11] produce optimal results in NLP tasks. The models are, first, trained on larger datasets and later fine-tuned with a few labelled data examples.

Self-supervised learning for pretraining models comes in multiple forms. For example, the models presented in [5,6] employed masked language modelling (MLM) with cross entropy as a loss function and next sentence prediction (NSP) using sigmoid loss. Through pretraining from the larger unlabeled dataset, the model extracts general language representations making downstream tasks to achieve better performance in a less labelled dataset. Pretraining over larger unlabeled datasets through SSL provides low-level information or background knowledge, which optimizes model performance even on lesser labelled data.

The paradigm of self-supervised learning shares similarities with supervised and unsupervised learning. For example, SLL does not require human-annotated data for learning, which is not the case with unsupervised learning with supervision. The variance between SSL and SL is learning meaningful representations from the unlabeled dataset, whereas unsupervised learning finds hidden patterns. On the other hand, SSL is synonymous with supervised learning because both require supervision. SSL offers general language representations for downstream models through transfer learning. It has better generalization through learning from unlabeled text data.

4.3. Self-Supervised Applications in NLP Applications

This learning approach began with NLP tasks in language models such as document processing applications, text suggestion, and sentence completion. The narrative changed after *the Word2Vec* paper [53] was introduced. The BERT (Bidirectional Encoder Representations from the Transformers) [5] model and its variants are the widely used language models based on SSL. Most of the variants of the BERT model were developed through modification in the last layers to handle a variety of NLP scenarios.

5. Pretrained Language Models Based on Transformer Network

The intuition of TL has become a standard method in NLP applications. Typical examples of NLP pretrained models include BERT [5], RoBERTa [6], ELECTRA [9], T5 [11], and XLNet [7]. Pretrained models present several opportunities such as:

- Pretrained models extract low-level information from unlabeled text datasets to enhance downstream tasks for performance optimization.

- The disadvantages of building models from scratch with minimal data sets are eliminated via transfer learning.
- Fast convergence with optimized performance even on smaller datasets.
- Transfer learning mitigates the overfitting problem in deep learning applications due to limited training datasets [54].

5.1. Transformer-Based Language Model Pretraining Process

In transferring knowledge from a pretrained model to a downstream application in natural language processing, it follows the under-listed steps:

Corpus identification: Identifying the best corpus to train the model in any pretraining model context is vital. A corpus is an unlabeled benchmark dataset, usually adopted to train a model for better performance, similar to BERT [5], which is pretrained by English Wikipedia and BooksCorpus. For a model to perform well, it must train on different text corpora [6,7].

Create vocabulary: Creating or generating the vocabulary is the next step. The step is mostly with varieties of tokenizers such as Google's Neural Machine Translation (GNMT) [55], byte pair encoding [56], and SentencePiece [57]. A tokenizer generates the vocabulary based on a selected corpus. Table 2 is a list of the size and the vocabulary used

Table 2. Summary of dataset, type of vocabulary and tokenizer for pretrained models.

Reference	Model	Dataset (Corpus)	Vocabulary	Vocabulary Size	Tokenizer
Lan et al., [3]	ALBERT	English Wikipedia and Books Corpus [58]	WordPiece	30,000	SentencePiece [57]
Devlin et al., [5]	BERT	English Wikipedia and Books Corpus [58]	WordPiece	30,000	SentencePiece [57]
Liu et al., [6]	RoBERTa	Books Corpus [58], English Wikipedia, CC-news, Open webtext	Byte-Pair Encoding (BPE)	50,000	-
Conneau and Lample [59]	Cross-lingual XLMs	Wikipedia, EUbookshop corpus, OpenSubtitles, GlobalVoices [60]	BPE	95,000	Kytea4 and PyThaiNLP5
Liu et al., [61]	mBART	CCNet Datasets [62]	bi-texts	250,000	SentencePiece
Wang et al., [63]	StuctBERT	English Wikipedia and Books Corpus	WordPiece	30,000	WordPiece
Joshi et al., [64]	SpanBERT	English Wikipedia and Books Corpus	WordPiece	30,000	-

Pretrained models such as XLM [59] and mBART [61] had larger vocabulary sizes because they modelled various languages. Pre-training a language model on big data increases the model size but ensures optimal performance. CharacterBERT, CANINE, ByT5, and Charformer [65–68]. The models do not use the WordPiece system; rather, a Character-CNN module makes the model lighter and more efficient, especially in specialized domains such as biomedical.

Construct the learning framework: The learning models learn by minimizing the loss function for convergence. Pretrained models such as [3,6,63] extract sentence semantics and should work on a downstream task for optimal performance. For example, the SpanBERT [64] is a variant of BERT proposed for content-masked prediction without using masked token representations, as performed in [69].

Pre-training approach: One approach to pretrain a language model is to start from scratch. The method is good but computationally expensive and requires a larger dataset. The drawback limits its application in language model training since it is unaffordable. The authors in [70,71] proposed a pre-training framework known as "knowledge inheritance"

(KI) that aids in the development of new pretrained models from already existing pretrained models. Based on this framework, less computational power and lesser time are required to pretrain the new model through self-supervised learning.

Parameter and hyperparameter settings: Model parameters and hyper-parameters such as learning rate, batch size [72] mask, and input sequence must be carefully set for quicker convergence and improved performance.

5.2. Dataset

Pretraining language models based on self-supervised learning require a larger unlabeled training dataset. In dealing with NLP tasks, the training dataset can be general, social media, language-based and domain-specific categories. Each category has different text characteristics to make it suitable for a particular language task. The dataset belonging to the general category is a clean text written by experts. The dataset obtained from the social media category is noisy and unstructured because it came from the public, not experts. Text datasets belonging to the domain-specific category, for example, biomedical, finance, and law, have texts not used in the general domain category. Domain-specific datasets are few in quantity, which makes it challenging when developing an NLP model for domain-specific tasks such as BioBER [33], ClinicalBERT [34], BLUE [73], and DAPT [74]. Pretraining on a larger dataset offers performance optimization, with the BERT model being an example. To affirm this point, the models developed in [6,7] with 32.89 B texts produced good performances. Based on this notion, larger datasets emerged for pretraining language models. A typical example is the CommonCrawl corpus [75]. Models such as IndoNLG [76], MuRIL [77], IndicNLPSuite [78], mT5 [79], mT6 [80], XLM-R [81], XLM-E [82], and INFOXLM [83] are multilingual pretrained models trained on larger datasets producing optimal performance. A summary of pretraining models with their datasets is in Table 3.

Table 3. A summary of dataset for pretraining models based on Transformer network.

Category	Model	Dataset	Focus	Evaluation Metrics
General				
	RoBERTa [6]	Books Corpus [58], English Wikipedia, Open webtext, and Stories	Pretrain a model on a larger dataset with bigger batch sizes for optimal performance.	GLUE [84], RACE, and SQuAD
	T2T Transformer [11]	Colossal Clean Crawled Corpus (C4) [11]	Developed a common framework to convert a variety of text-based language problems into a text-to-text format	GLUE and SQuAD
Social media				
	HateBERT [85]	RAL-E	Developed to analyze offensive language singularities in English	Macro F1 Class—F1
	SentiX [86]	Amazon review [87] and Yelp 2020 dataset	Analysis of consumer sentiments from different domains	Accuracy

Table 3. *Cont.*

Category	Model	Dataset	Focus	Evaluation Metrics
Domain Specific				
Biomedical	BioBERT [33]	BooksCorpus PMC articles and PubMedAbstracts	Question and answering model for the biomedical field	F1 score, MRR
	BLUE [73]	BC5CDR, MedSTS, and BIOSSES [73]	Developed the BLUE evaluation framework to access the performance of biomedical pretrained models	Pearson, Accuracy, and micro F1
	ClinicalBERT [34]	MIMIC-III v1.4 database [88]	Demonstrate that clinical-specific contextual embeddings improve domain results	Accuracy, Exact F1
News and academia	DAPT [74]	Amazon review [87] and RealNews [89]	Developed an efficient model to analyze small corpus with improved performance	F1-Score
Language based				
Monolingual	IndoNLG [76]	Indo4B [76]	Developed the IndoNLU model for complex sentence classification	F1-Score
	DATM [90]	GermEval 2017 data [90]	Developed a transformer-based model to explore model efficiency on German customers	F1-Score
	PTT5 [91]	BrWac [92] and ASSIN 2 [93]	Improved the T5 model to translate the Portuguese language to Brazilian Portuguese	Precision, Pearson, Recall, and F1
	RoBERTa-tiny-clue [94]	CLUECorpus2020 [94]	Developed the Chinese CLUECorpus2020 to pretrain Chinese language models	Accuracy
	Chinese-Transformer-XL [95]	WuDaoCorpora [95]	Developed a 3 TB Chinese Corpora for word embedding model pre-training	Per-word perplexity (ppl)
Multi-lingual	IndoNLG [76]	Indo4B-Plus	Introduced the IndoNLG model to translate multiple languages (Indonesian, Sundanese, and Javanese)	BLEU, ROUGE, and F1 score
	MuRIL [77]	OSCAR [75] and Wikipedia	Introduced the MuRIL multilingual LM for Indian languages translation	Accuracy
	IndicNLPSuite [78]	IndicGLUE benchmark	Developed a large-scale, dataset for Indian language translation	Accuracy

Table 3. *Cont.*

Category	Model	Dataset	Focus	Evaluation Metrics
Multi-lingual	mT5 [79]	mC4 derived from Common Crawl corpus [75]	Introduced the mT5 multilingual variant of the T5 model pretrained on the Common Crawl dataset, which covers 101 languages	Accuracy and F1 score
	mT6 [80]	CCNet [62]	The proposed MT6 is an improved version of MT5 for corruption analysis	Accuracy and F1 score
	XLM-R [81]	CommonCrawl Corpus [75]	Developed a multilingual model for a wide range of cross-lingual transfer tasks	Accuracy and F1 score
	XLM-E [82]	CommonCrawl Corpus [75]	Developed two techniques for token recognition and replacement for cross-lingual pre-training	Accuracy and F1 score
	INFOXLM [83]	CommonCrawl Corpus [75]	Proposed an info-theoretic model for cross-lingual language modelling to maximize the mutual information between multi-granularity texts	Accuracy

5.3. Transformer-Based Language Model Pretraining Techniques

This section introduces various pretraining techniques based on transformer networks proposed in the literature for NLP tasks using SSL.

5.3.1. Pretraining from Scratch

Pretraining a language model from scratch was used in elite models such as BERT [5], RoBERTa [6], and ELECTRA [9] for language processing tasks. The method is data driven because the training process is through self-supervised learning based on a larger unlabeled test dataset. Pretraining from scratch is computationally intensive and expensive because computers with high processing power technologies, such as graphical processing units (GPUs), are required.

5.3.2. Incessant Pretraining

In this method, a new language model is initialized from an existing pretrained language model for further pretraining. The initialized weights are not learned from scratch, as in pretraining from scratch models. Figure 5 illustrates the transmission of preexisting weights or parameters from a base pretrained model to a target domain for tuning. This approach is a common phenomenon in developing models for domain-specific tasks. Transformer-based language models such as ALeaseBERT [27], BioBERT [33], infoXLM [83], and TOD-BERT [29] are examples of models initialized on existing pretrained models and later finetuned for specific NLP tasks. A key observation of this method of pretraining is that it is cost-effective in terms of computational power since it is trained on already pretrained parameters. Additionally, less training time is required compared to training from scratch.

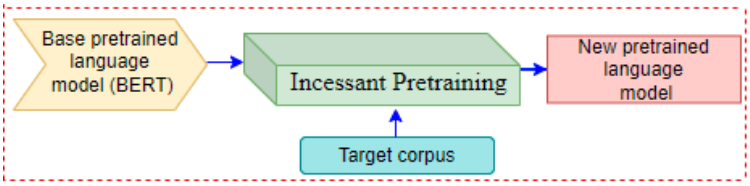

Figure 5. Incessant pretraining process (in this case, the pre-training task is progressively constructed, and the models are pre-trained and fine-tuned to respond to different language understanding tasks).

The BioBERT method was initialized using BERT's weights, which were pre-trained with general domain corpora (English Wikipedia and BooksCorpus). Next, BioBERT is finetuned on corpora from the biomedical area (PubMed abstracts and PMC full-text articles).

5.3.3. Pretraining Based on Knowledge Inheritance

As previously indicated, pretraining a language model based on self-supervised learning requires a larger dataset, which makes the method computationally expensive and time-consuming. As knowledge acquisition from a people perspective is from human experience, the same phenomenon is in language model training. The authors in [70] proposed a model known as "knowledge inheritance pretrained transformer" (KIPT), which is similar to knowledge distillation (KD). Refer to Figure 6 for the training process. The model learns how knowledge distillation provides supervision during pre-training to target models.

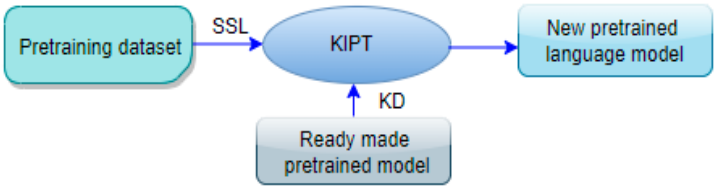

Figure 6. Pretrained model based on knowledge transfer.

A new language model is pretrained using the knowledge from an existing pretrained model. The equation below explains the learning process. L_{SSL} and L_{KD} are losses from self-supervised learning and knowledge distillation, respectively, and L_{KIPT} is the model's loss function.

$$L_{kIpT} = \sigma \times L_{ssL} + (1 - \sigma) \times L_{kD}$$

The proposed knowledge inheritance model operates on the "teacher and student" scenario, where the "student" learns from the "teacher" by encoding the knowledge acquired from the "teacher". The student model extracts knowledge through SSL and from the "teacher" to enhance model efficiency. The approach requires less datasets, making it less computationally expensive with minimal training time compared to only self-supervised pretraining methods. The CPM-2 model introduced in [71] is a Chinese–English bilingual model developed based on knowledge inheritance with optimized performance.

5.3.4. Multi-Task Pre-Training

With this technique, a model extracts relevant information across multiple tasks concurrently to minimize the need for a labelled dataset in a specific target task. The authors in [11] utilized a multi-task-pretraining approach to optimize model performance. Multi-Task Deep Neural Network (MT-DNN) was used for learning representations across several natural language understanding (NLU) tasks. The proposed model depends on a significant quantity of cross-task data with a regularization effect that results in more general representations to aid in adapting to new domains [96]. Two steps make up the MT-DNN training process: pre-training and multi-task learning. The pre-training phase is the same as the BERT model. The parameters of all shared task-specific layers were

learned during the multi-task learning stage using mini-batch-based stochastic gradient descent (SGD). Finding a single training dataset that includes all the necessary slot types, such as domain classification, intents categorization, and slot tagging for named entity identification, is challenging in the health domain. A multi-task transformer-based neural architecture for slot tagging solves the issues [97]. As a multi-task learning problem, the slot taggers were trained using many data sets encompassing various slot kinds. In terms of time and memory and efficiency and effectiveness, the experimental findings in the biomedical domain were superior to earlier state-of-the-art systems for slot tagging on the various benchmark biomedical datasets. The multi-task approach was used in [98] to extract eight different tasks in the biomedical field. The Clinical STS [99] dataset was subjected to multi-task fine-tuning, and the authors repeatedly selected the optimal subset of related datasets to produce the best results. To further improve the model's performance after multi-task fine-tuning, the model can be further fine-tuned on the target particular dataset. The Multi-task Learning (MTL) [100] model's outstanding performance represents the pinnacle of the multi-task pre-training technique in NLP applications.

Table 4 is a summary of the various pretraining techniques employed in the development of language models for different NLP tasks. Multi-task pretraining is appropriate for domain-specific applications with outstanding performance. On the other hand, knowledge inheritance is as good as the multi-task pretraining technique. Its adaptation is suitable for edge scenario devices since it is computationally less expensive. The information and the suggested literature support researchers in selecting the appropriate pretraining technique for new applications.

Table 4. Summary of pretraining techniques employed for language modelling.

Method	Model	Focus	Pros	Limitations	Model Evaluation
Pretraining from scratch	BERT [5]	Designed to pretrain deep bidirectional representations from unlabeled text.	It is a straightforward model to generate cutting-edge models for a variety of tasks, including QA and language inference, with minimal architectural adjustments.	The BERT model was severely undertrained and may match or outperform some models published after it.	GLUE score = 80.5%, accuracy 86.7, F1 score = 93.2
	RoBERTa [6]	Improvements to the original BERT architectural design combined with alternatives and training methods that improve downstream task performance.	The architectural and training advancements demonstrate a competitive advantage of masked language model pretraining, with all other state-of-the-art models.	Model is computationally expensive since the training dataset is large (160 GB data).	SQuAD = 94.6/89.4, MNLI-m = 90.2, SST-2 = 96.4, QNLI = 98.9%
	ELECTRA [9]	Introduces discriminative and generator models for prediction.	Outstanding performance on downstream tasks with less computing power.	Requires high computer power for training	MNLI = 90.7, CoLA = 68.1

Table 4. *Cont.*

Method	Model	Focus	Pros	Limitations	Model Evaluation
Incessant pretraining	ALeaseBERT [27]	Introduced a new benchmark dataset, trained on the ALeaseBERT language model, and generated ground-breaking outcomes.	The suggested model detects two elements (entities and red flags), crucial in a contract review with excellent performance.	The precision at high recall for the red flag detection requires improvement for end-user and professional satisfaction.	MAP = 0.5733, Precision = 0.62, Recall = 0.48, F1 = 0.54
	BioBERT [33]	Introduced model for pre-trained language representation for biomedical text mining.	The first domain-specific BERT-based model pretrained on biomedical corpora with improved performance.	It is expensive to generate domain-specific corpora because of specific vocabulary not found in general corpora.	NER = (0.62% F1 score = 2.80%, MRR = 12.24%
	TOD-BERT [29]	Introduced a task–conversation model, trained on nine human and multi-turn task-oriented datasets, spanning more than 60 domains.	Four tasks involving dialogue that TOD-BERT performs better than BERT are answer selection, dialogue act prediction, dialogue state tracking, and intention categorization.	Implementation can be computationally expensive.	MWOZ = 65.8% 1-to-100 accuracy and 87.0% 3-to-100 accuracy
	infoXLM [83]	Presents a framework that defines a cross-linguistic language model to maximize multilingual and multi-granularity texts.	A cross-lingual comparative learning task and a single cross-lingual pretraining are successful with the model from an information-theoretic perspective	Due to specialized vocabulary that is absent from broad corpora, creating domain-specific corpora is costly.	XNLI = 76.45, MLQA = 67.87/49.58
Multi-task pretraining	MT-DNN [97]	To integrate multi-task learning with language model pretraining for language representation learning.	MT-DNN has remarkable generalization capabilities, archiving outstanding results on 10 NLU tasks using three well-known benchmarks: GLUE, SNLI, and SciTail.	The model requires improvement to include the linguistic structure of the text more clearly and understandably.	MNLI = 87.1/86.7, CoLa = 63.5, Accuracy = 91.6%
	MT-BioNER [98]	Present a slot tagging neural architecture based on a multi-task transformer network for the biomedical field.	The suggested strategy outperforms the most recent cutting-edge techniques for slot tagging on several benchmark biomedical datasets.	Investigate the effects of dataset overlap on the model's performance on larger unlabeled datasets	Recall = 90.52, Precision = 88.46, F1 = 89.5

Table 4. *Cont.*

Method	Model	Focus	Pros	Limitations	Model Evaluation
Multi-task pretraining	MT-Clinical BERT [99]	Developed the Multitask-Clinical BERT, which uses shared representations to carry out eight clinical tasks.	The suggested approach is resilient enough to incorporate new activities while concurrently supporting future information extraction.	Adding larger tasks may need rigorous ablation tests to determine the overall benefits of each such work.	Micro-F1 = 84.1 (+0.2)
	Multi-task learning [100]	Developed a multi-task learning model with decoders for a variety of biological and clinical NLP tasks.	The MT-BERT-Fine-Tuned model proposed eight tasks from various text genres that displayed outstanding performance.	Further investigation is required on task relationship characterization on data qualities.	Accuracy = 83.6%
Knowledge inheritance pretraining	KIPM [70]	Present the KI pretraining architecture to effectively learn bigger pretrained language models.	The proposed architecture uses already trained larger models to teach smaller ones by transferring information across several language models.	Selecting an appropriate teacher model for KI can be difficult sometimes, limiting model performance.	F1 = 84.5%
	CPM-2 [71]	A cost-effective pipeline for large-scale pre-trained language models based on KI.	The framework is memory-efficient for quick tuning, achieving outstanding performance on full-model tuning.	The model needs further optimization.	Accuracy = 91.6%

NER—Named Entity Recognition, RE—Relation Extraction, IR—Information Retrieval, QA—Question Answering.

5.4. Word Embedding Types in Transformer-Based Pretraining Models

Word embedding converts character-based datasets into matrix format for a language model to process. There are two major embedding types: primary embedding and secondary embedding. Primary embeddings are characters or sub-words and word embeddings to form a vocabulary fed as input to the NLP model for processing. Word embedding vocabulary consists of every word selected in the pretraining dataset. Meanwhile, the character-embedding vocabulary entails only the characters that form the pretraining corpus. Secondary embeddings contain secondary information, such as the position and language of the pretraining model. The model size and the vocabulary with primary and secondary embeddings are equal [101].

5.4.1. Text/Character Embeddings

The input dataset for most NLP models is a sequence of characters, a combination of characters (sub-word), numbers, and symbols. The CharacterBERT [65], CHARFORMER [68], and AlphaBERT [102] are typical examples of character-based embedding pretrained models that utilize characters instead of words for pretraining. On the other hand, the novel BERT model [5], BART [4], RoBERTa [6], and XLNet [7] are pretrained on sub-word embeddings, even though they have varying tokenizers for vocabulary generation. The generated vocabulary consists of letters, symbols, punctuation, and numbers mapped to a dense low-dimensional vector. The learning process is through the random initialization of each character in the vocabulary.

5.4.2. Code Embeddings

This type of embedding is domain-specific, for example, in the medical sector, where special codes represent cases or concepts such as disease, drug prescription, prognosis, therapy, and surgery. Patient information is stored in codes instead of plain text so that only clinical professionals can interpret it. The authors in [103] proposed a transformer-based bidirectional representation learning model on EHR sequences to diagnose depression. The input dataset for the model was code embedding extracted from an electronic health record (EHR). Med-BERT [104] and BeHRt [105] also uses code embeddings as input vocabulary for pretraining through random initialization.

5.4.3. Sub-Word Embeddings

Byte Pair Encoding Byte Level BPE (bBPE), Unigram, and SentencePiece are employed to generate the vocabulary, which serves as the input data for pretraining the language model. A summary of tokenizers used in literature is in Table 2. It is very critical when choosing the vocabulary size when using sub-word embeddings. A smaller-sized vocabulary can generate long sequences because multiple sub-words will emerge. The case is a bit different with models such as IndoNLG [76], MuRIL [77], and IndicNLPSuite [78], which are developed for multilingual language processing because such models require a large vocabulary to handle different kinds of languages.

5.5. Secondary Embeddings

Secondary embedding contains specific information with a purpose about the pretrained model. Positional embedding and sectional embedding are examples of secondary embeddings used in general models to describe the position and also differentiate tokens forming various sentences, especially in language models such as RoBERTa-tiny-clue [94], Chinese-Transformer-XL [95], and XLM-E [82]. There are specific secondary embedding types used in domain-specific transformer-based language models. A few are below.

5.5.1. Positional Embeddings

Transformer-based language models require positional information about the text dataset to make predictions without regard to the text location in the vocabulary. The situation varies with CNN and RNN models because predictions are consecutive to each character following the other in RNN. Sequential processing does not use positional information. Transformer networks do not process information sequentially, hence the need-to-know order and positional details of characters for prediction. The positional information is sometimes learned together with other parameters during pretraining [5,9].

5.5.2. Sectional Embeddings

In sentence-pair models, both sentence tokens are taken as input simultaneously and differentiated with sectional embedding. Positional embedding varies with tokens in the input sentences, but sectional embedding remains constant.

5.5.3. Language Embeddings

This type of secondary embedding works in cross-lingual pretrained language models [106,107] to provide vivid information to the model on the input sentence language. For instance, the XLM model is pretrained on MLM, which contains sentences in one language on monolingual text data in 100 languages. MLM sentences come from one language where the language embedding is constant for all the input sentence tokens.

6. Knowledge Transfer Techniques for Downstream Tasks

The techniques employed to transfer knowledge, parameters and pretrained corpus to a downstream task for natural language processing include: (word feature-based transfer, fine-tuning, and prompt-based tuning).

6.1. Word Feature Transfer

The input data to traditional natural language architectures such as RNN embedding models such as Word2Vec [53] generate the input set (word features). Transformer-based pretrained models such as BERT [5], generate contextual word vectors (word features) similar to Word2Vec. The BERT model supports encoding more information in word vectors due to the deepness of transformer architecture with stacked attention. Due to this, downstream tasks benefit from the word vectors from any part of the network layer.

The process involves training the downstream model from the initial stages without the labelled embedding instances. The innovative BERT model is improved upon by the DeBERTa model suggested [108]. The variance between the two is that DeBERTa uses a disentangled attention mechanism where the words are in two-vector form (content and position). The second unique technique of DeBERTa is a mask decoder for prediction during pretraining. All these combined make this model superior to BERT and RoBERTa. The ConvBET model [105] is also an advancement of the BERT model because it uses less memory and is computationally efficient.

6.2. Fine-Tuning

Current work has shown that fine-tuning a base model produces optimal performance on target tasks to training with only target task data [109]. The advantage of pretraining is that it provides universal inference of a language [110]. The work in [111] proved that fine-tuning yields optimal performance by examining the English BERT variants. It is also evident that fine-tuning does not change the representation but rather fine-tunes it to a downstream task. Fine-tuning was used in [112] to understand how representation space changes during fine-tuning for downstream tasks.

The study capitalized on three NLP tasks; dependency parsing, NLP inference, and reading comprehension. Fine-tuning adds massive changes to domain instances but looks out-of-domain similar to the pre-trained model. To evaluate fine-tuning effects on representations learned by pretrained language models, the authors in [113] proposed a sentence-level probing model to ascertain the changes. BERT was fine-tuned based on two indicators [114]. The first indicator was to evaluate the attention mode in the transformer network based on the Jensen–Shannon divergence during fine-tuning of the BERT model. The second indicator measured feature extraction changes during model fine-tuning based on Singular Vector Canonical Correlation Analysis (SVCCA) [115].

6.3. Intermediate-Task Transfer Learning

Compared with more established deep learning techniques such as RNN, the top pretrained networks BERT and RoBERTa perform extraordinarily well. The current performance of these models is optimized by further training the model on a curated dataset for the intermediate task through fine-tuning. The work proposed in [116] employed an intermediate fine-tuning approach to improving the performance of the RoBERTa pretrained language model with 110 intermediate–target task combinations.

Intermediate fine tuning on a semi-supervised pretrained language model performs well in domain-specific tasks such as medical question–answer pairs [117] to extract medical question resemblances. Figure 7 depicts the training process. In dealing with medical domain NLP applications, a model pretrained on a different problem in similar domain beats models pre-trained on an analogous task in a dissimilar field. Pretraining a language model based on a biomedical dataset produces optimal performance in domain-specific languages [118]. For example, an optimized performance was achieved in the biomedical question and answering (QA) task through the transfer of knowledge from BioBERT, based on natural language inference (NLI) [119]. Fine-tuning works well when the source and target datasets come from the same domain but in different tasks. In this case, fine-tuning happens on domain datasets before transferring to in-domain datasets. In [118], the authors showed that teaching a domain-specific language model on rich biological corpora has a considerable impact. Training the model on larger NLI datasets such as

MultiNLI [120] and SNLI [121] aids in efficient task-specific reasoning with optimized performance. Fine-tuning is possible when the source and target datasets are from the same task and domain. However, the target dataset is more specialized, whereas the source dataset is more general [122]. Fine-tuning is also feasible for many tasks and domains where source and target datasets come from different fields. The BioBERT model was tuned on a generic MultiNLI database biomedical question and answer (QA) [119]. The performance was outstanding in learning to reason at the phrase level for biomedical QA.

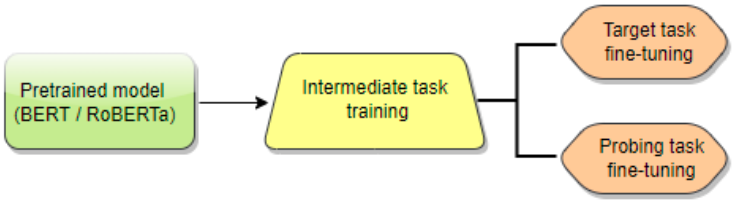

Figure 7. Intermediate-task transfer learning and subsequent fine-tuning (a pre-trained model (BERT), is fine-tuned on the target task for intermediate task training. The model is then fine-tuned separately for each target and probing task. The target tasks offer great importance to NLP applications.).

7. Discussion, Open Challenges, and Future Directions

This section highlights findings from literature based on transfer learning techniques, such as pretraining on transformer networks for natural language models. We also shed light on some future directions that are vital to the progression of the field.

7.1. Optimized Pretraining Techniques

As there are billions of parameters involved, pretraining transformer-based language models using unlabeled datasets over SSL is costly and makes it impractical to train a language model from scratch. According to the literature, models such as [69,70] acquired knowledge from language models that had already undergone pretraining using a knowledge distillation technique. As compared to models created for equivalent tasks, the newly designed KPIT's efficiency was exceptional. The KPIT model possesses rich features such as a faster convergence rate and less pretraining time requirements, making it appropriate for downstream tasks.

7.2. Domain Specific Pretraining

Mixed-Domain Pretraining is a popular strategy frequently used in the literature to produce domain-specific assignments.

The method relies on a larger domain-specific dataset, which unintentionally necessitates more computing capacity. Despite its efficacy, pretraining is unaffordable due to hardware requirements and energy usage. Task Adaptive Transfer Learning (TATL) was proposed in [118] to address this. Another technique was through pseudo-labelling in-domain data and iterative training [123], which keeps the distribution of pseudo-labelled instances closer to that of the in-domain data to achieve optimal performance.

7.3. Dataset/Corpus

Pretrained models require a larger volume of labelled datasets or text corpus for optimal performance. Labelled datasets are expensive to generate in larger quantities. Self-supervised learning is one approach that utilizes the voluminous unlabeled dataset for contemporary NLP tasks. Language models such as ALBERT [3], BER [5], and RoBERTa [6] were pretrained on benchmark general corpora such as English Wikipedia and Books Corpus [58]. On the other hand, developing task or domain-specific language models is challenging since the dataset in specific domains are scanty for the transformer model to produce good results. For example, training models for the biomedical field require domain-specific datasets, which are not readily available in larger quantities.

7.4. Model Efficacy

The cost of pretraining on unlabeled text data is expensive in terms of hardware and dataset acquisition. The second issue is that datasets for domain-specific areas (biomedical) are few, even though unlabeled datasets are abundantly available. The DeBERTa model [108] and ConvBERT [124] are examples of models that produce good performance compared to earlier pretrained models such as BERT [5], RoBERTa [6], and ELECTRA [9]. For instance, DeBERTa is pretrained on fewer datasets compared to BERT, reducing computational power with improved performance as well. Moreover, the ConvNet model created employing a mixed attention mechanism outperforms ELECTRA utilizing just a quarter of the dataset used to pretrain the ELECTRA model. Modern pretrained language models require such models to operate on edge devices with less processing power and have optimal performance.

7.5. Model Adaptation

Through incessant pretraining, knowledge gained in general pretrained models was adapted to specific domains such as biomedical and multilingual models. Despite the success of incessant pretraining in domain-specific tasks, there are some performance issues due to inadequate domain-specific datasets. Models such as ALeaseBERT [27], BioBERT [33], infoXLM [83], and TOD-BERT [29] are examples of incessant pretrained models whose main aim is to reduce computational cost and provide optimal performance for domain-specific models. There is a need to research novel adaptation methods for pretrained language models.

7.6. Benchmarks

Evaluating a transformer-based pretrained model is vital, as model efficacy is paramount in its patronage. Some benchmarking frameworks have been proposed in this regard for general [84] and specific domain models [35,73]. In [76,78], there is some benchmarks to evaluate monolingual and multilingual language models. Despite these benchmarks being available, they are not adequate to cover all domains. Most of these benchmarks are developed for the performance of literature-based datasets, hence the need for other ones for electronic health records and domain-specific corpus.

7.7. Security Concerns

Security is of much concern in pretrained transformer models since there are some identified risks, such as data leakage occurring during pretraining. This usually happens on datasets containing confidential information about people. Training a model over a long period subjects it to retrieve vital information, such as personally identifiable information. Due to this drawback, models pretrained on datasets containing confidential information are not released into the public domain. A typical example is the model presented in [125], which extracted precise text classifications of personal information from the GPT-2 model's training data. We recommend that the KART (Knowledge, Anonymization, Resource, and Target) framework [126], which deals with real-world privacy leakages, be adapted and improved for better performance.

8. Conclusions

This review follows the PRISMA reporting standards for review to retrieve relevant publications to form the Primary studies. Additionally, backwards and forward snowballing was employed to retrieve additional publications from the Primary studies. This study reviews transfer learning-based pretrained models for NLP based on deep transformer networks. The study shows the recent trends of transformer networks in solving language problems compared to traditional deep learning algorithms such as RNN. The paper explains the transformer model and the various core concepts behind its operation. The work focused on self-supervised learning using labelled data for later tasks rather than unlabeled datasets for model pretraining. The study also examined several benchmarking

systems for assessing the effectiveness of pretrained models. Some challenges identified in the literature from transformer-based pretrained models have been discussed, with possible recommendations to deal with those challenges. We also provide future directions to help researchers focus on developing improved NLP applications using transformer networks and self-supervised learning.

Author Contributions: Conceptualization, E.K. and R.T.; methodology, R.T.; software, E.K.; validation, E.K. and R.T.; formal analysis, E.K.; investigation, E.K.; resources, R.T.; data curation, E.K.; writing—original draft preparation, E.K.; writing—review and editing, R.T.; visualization, E.K.; supervision, R.T.; project administration, R.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data are contained within this article.

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

References

1. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A Convolutional Neural Network for Modelling Sentences. *arXiv* **2014**. [CrossRef]
2. Liu, P.; Qiu, X.; Xuanjing, H. Recurrent neural network for text classification with multi-task learning. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 2873–2879.
3. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942v6.
4. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7871–7880. [CrossRef]
5. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the NAACL HLT 2019—2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
6. 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:1907.11692v1.
7. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–11.
8. Fausk, H.; Isaksen, D.C. *t*-model structures. *Homol. Homotopy Appl.* **2007**, *9*, 399–438. [CrossRef]
9. Clark, K.; Luong, M.-T.; Le, Q.V.; Manning, C.D. ELECTRA: Pre-training text encoders as discriminators rather than generators. In Proceedings of the ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020; pp. 1–18.
10. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P.J. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In Proceedings of the 37th International Conference on Machine Learning (ICML 2020), Virtual Event, 13–18 July 2020; Volume PartF16814, pp. 11265–11276.
11. 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.
12. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
14. Kotei, E.; Thirunavukarasu, R. Ensemble Technique Coupled with Deep Transfer Learning Framework for Automatic Detection of Tuberculosis from Chest X-ray Radiographs. *Healthcare* **2022**, *10*, 2335. [CrossRef]
15. Zhong, Z.; Li, Y.; Ma, L.; Li, J.; Zheng, W.-S. Spectral–Spatial Transformer Network for Hyperspectral Image Classification: A Factorized Architecture Search Framework. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [CrossRef]
16. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High Quality Object Detection and Instance Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1483–1498. [CrossRef] [PubMed]
17. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable transformers for end-to-end object detection. In Proceedings of the ICLR 2021, Virtual Event, 3–7 May 2021; pp. 1–16.
18. Balakrishnan, G.; Zhao, A.; Sabuncu, M.R.; Guttag, J.; Dalca, A.V. VoxelMorph: A Learning Framework for Deformable Medical Image Registration. *IEEE Trans. Med. Imaging* **2019**, *38*, 1788–1800. [CrossRef] [PubMed]
19. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *15*, 12077–12090.
20. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015-Conference Track Proceedings), San Diego, CA, USA, 7–9 May 2015; pp. 1–14.

21. Chouhan, V.; Singh, S.K.; Khamparia, A.; Gupta, D.; Tiwari, P.; Moreira, C.; Damaševičius, R.; de Albuquerque, V.H.C. A Novel Transfer Learning Based Approach for Pneumonia Detection in Chest X-ray Images. *Appl. Sci.* **2020**, *10*, 559. [CrossRef]
22. Coccia, M. Deep learning technology for improving cancer care in society: New directions in cancer imaging driven by artificial intelligence. *Technol. Soc.* **2019**, *60*, 101198. [CrossRef]
23. Fang, X.; Liu, Z.; Xu, M. Ensemble of deep convolutional neural networks based multi-modality images for Alzheimer's disease diagnosis. *IET Image Process.* **2020**, *14*, 318–326. [CrossRef]
24. Apostolopoulos, I.D.; Mpesiana, T.A. COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Phys. Eng. Sci. Med.* **2020**, *43*, 635–640. [CrossRef]
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5999–6009.
26. Yang, Y.; Uy, M.C.S.; Huang, A. FinBERT: A Pretrained language model for financial communications. *arXiv* **2020**, arXiv:2006.08097v2.
27. Leivaditi, S.; Rossi, J.; Kanoulas, E. A Benchmark for lease contract review. *arXiv* **2020**, arXiv:2010.10386v1.
28. Chalkidis, I.; Fergadiotis, M.; Malakasiotis, P.; Aletras, N.; Androutsopoulos, I. LEGAL-BERT: The muppets straight out of law school. *arXiv* **2020**, arXiv:2010.02559v1, 2898–2904. [CrossRef]
29. Wu, C.-S.; Hoi, S.; Socher, R.; Xiong, C. TOD-BERT: Pre-trained Natural Language Understanding for. In Proceedings of the Emnlp2020, Online, 16–20 November 2020; pp. 917–929.
30. Liu, X.; Yin, D.; Zheng, J.; Zhang, X.; Zhang, P.; Yang, H.; Dong, Y.; Tang, J. OAG-BERT: Towards a Unified Backbone Language Model for Academic Knowledge Services. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2022, Washington, DC, USA, 14–18 August 2022. [CrossRef]
31. Beltagy, I.; Lo, K.; Cohan, A. SciBERT: Pretrained contextualized embeddings for scientific text. *arXiv* **2019**, arXiv:1903.10676.
32. Peng, S.; Yuan, K.; Gao, L.; Tang, Z. MathBERT: A pre-trained model for mathematical formula understanding. *arXiv* **2021**, arXiv:2105.00377v1.
33. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2019**, *36*, 1234–1240. [CrossRef] [PubMed]
34. Alsentzer, E.; Murphy, J.; Boag, W.; Weng, W.-H.; Jindi, D.; Naumann, T.; McDermott, M. Publicly available clinical BERT embeddings. In Proceedings of the 2nd Clinical Natural Language Processing Workshop, Minneapolis, MN, USA, 6–7 June 2019. [CrossRef]
35. Yuxian, G.; Robert Tinn, R.; Hao Cheng, H.; Lucas, M.; Usuyama, N.; Liu, X.; Naumann, T.; Gao, J.; Poon, H. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *arXiv* **2020**, arXiv:abs/2007.15779.
36. Badampudi, D.; Petersen, K. Experiences from using snowballing and database searches in systematic literature studies Categories and Subject Descriptors. In Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering, Nanjing, China, 27–29 April 2015; pp. 1–10.
37. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *4*, 3104–3112.
38. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
39. Bahdanau, D.; Cho, K.H.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
40. Britz, D.; Goldie, A.; Luong, M.-T.; Le, Q. Massive Exploration of Neural Machine Translation Architectures. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017. [CrossRef]
41. Cheng, J.; Dong, L.; Lapata, M. Long Short-Term Memory-Networks for Machine Reading. *arXiv* **2016**, arXiv:1601.06733.
42. Lin, Z.; Feng, M.; Santos, C.N.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A structured self-attentive sentence embedding. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France, 24–26 April 2017; pp. 1–15.
43. Lewis, J.C.; Floyd, I.J. Reorientation effects in vitreous carbon and pyrolytic graphite. *J. Mater. Sci.* **1966**, *1*, 154–159. [CrossRef]
44. Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; Tang, J. Self-supervised Learning: Generative or Contrastive. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 857–876. [CrossRef]
45. Baevski, A.; Zhou, H.; Mohamed, A.; Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12449–12460.
46. Liu, Q.; Kusner, M.J.; Blunsom, P. A Survey on contextual embeddings. *arXiv* **2020**, arXiv:2003.07278v2.
47. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *ACM Comput. Surv.* **2022**, *54*, 1–41. [CrossRef]
48. Yang, J.; Li, C.; Zhang, P.; Dai, X.; Xiao, B.; Yuan, L.; Gao, J. Focal Self-attention for Local-Global Interactions in Vision Transformers. *arXiv* **2021**, arXiv:2107.00641.

49. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seoul, Korea, 16–18 June 2020; pp. 9759–9768.
50. Park, D.; Chun, S.Y. Classification based grasp detection using spatial transformer network. *arXiv* **2018**, arXiv:1803.01356v1.
51. Kirillov, A.; He, K.; Girshick, R.; Rother, C.; Dollár, P. Panoptic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9404–9413.
52. Prangemeier, T.; Reich, C.; Koeppl, H. Attention-Based Transformers for Instance Segmentation of Cells in Microstructures. In Proceedings of the 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Seoul, Republic of Korea, 16–19 December 2020; pp. 700–707. [CrossRef]
53. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. In Proceedings of the 1st International Conference on Learning Representations (ICLR 2013), Scottsdale, AZ, USA, 2–4 May 2013; pp. 1–12.
54. Erhan, D.; Courville, A.; Bengio, Y.; Vincent, P. Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.* **2010**, *9*, 201–208.
55. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv* **2016**, arXiv:1609.08144v2.
56. Sennrich, R.; Haddow, B.; Birch, A. Neural machine translation of rare words with subword units. *arXiv* **2016**, arXiv:1508.07909.
57. Kudo, T.; Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Brussels, Belgium, 31 October–4 November 2018. [CrossRef]
58. 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 2015, Washington, DC, USA, 7–13 December 2015; pp. 19–27. [CrossRef]
59. Conneau, A.; Lample, G. Cross-lingual language model pretraining. *arXiv* **2019**, arXiv:1901.07291v1.
60. Tiedemann, J. Parallel data, tools and interfaces in OPUS. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey, 23–25 May 2012; pp. 2214–2218.
61. Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; Zettlemoyer, L. Multilingual Denoising Pre-training for Neural Machine Translation. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 726–742. [CrossRef]
62. Wenzek, G.; Lachaux, M.A.; Conneau, A.; Chaudhary, V.; Guzmán, F.; Joulin, A.; Grave, E. CCNet: Extracting high quality monolingual datasets from web crawl data. In Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020), Marseille, France, 11–16 May 2020; pp. 4003–4012.
63. Wang, W.; Bi, B.; Yan, M.; Wu, C.; Bao, Z.; Xia, J.; Peng, L.; Si, L. StructBERT: Incorporating language structures into pre-training for deep language understanding. *arXiv* **2019**, arXiv:1908.04577v3.
64. Joshi, M.; Chen, D.; Liu, Y.; Weld, D.S.; Zettlemoyer, L.; Levy, O. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 64–77. [CrossRef]
65. El Boukkouri, H.; Ferret, O.; Lavergne, T.; Noji, H.; Zweigenbaum, P.; Tsujii, J. CharacterBERT: Reconciling ELMo and BERT for Word-Level Open-Vocabulary Representations From Characters. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 6903–6915. [CrossRef]
66. Clark, J.H.; Garrette, D.; Turc, I.; Wieting, J. Canine: Pre-training an Efficient Tokenization-Free Encoder for Language Representation. *Trans. Assoc. Comput. Linguist.* **2022**, *10*, 73–91. [CrossRef]
67. Xue, L.; Barua, A.; Constant, N.; Al-Rfou, R.; Narang, S.; Kale, M.; Roberts, A.; Raffel, C. ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. *Trans. Assoc. Comput. Linguist.* **2022**, *10*, 291–306. [CrossRef]
68. Tay, Y.; Tran, V.Q.; Ruder, S.; Gupta, J.; Chung, H.W.; Bahri, D.; Qin, Z.; Baumgartner, S.; Yu, C.; Metzler, D. Charformer: Fast character transformers via gradient-based subword tokenization. *arXiv* **2021**, arXiv:2106.12672v3.
69. Di Liello, L.; Gabburo, M.; Moschitti, A. Efficient pre-training objectives for Transformers. *arXiv* **2021**, arXiv:2104.09694v1.
70. Qin, Y.; Lin, Y.; Yi, J.; Zhang, J.; Han, X.; Zhang, Z.; Su, Y.; Liu, Z.; Li, P.; Sun, M.; et al. Knowledge Inheritance for Pre-trained Language Models. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, WA, USA, 10–15 July 2022; pp. 3921–3937. [CrossRef]
71. Zhang, Z.; Gu, Y.; Han, X.; Chen, S.; Xiao, C.; Sun, Z.; Yao, Z.S.Y.; Qi, F.; Guan, J.; Ke, P.; et al. CPM-2: Large-scale cost-effective pre-trained language models. *AI Open* **2021**, *2*, 216–224. [CrossRef]
72. You, Y.; Li, J.; Reddi, S.; Hseu, J.; Kumar, S.; Bhojanapalli, S.; Song, X.; Demmel, J.; Keutzer, K.; Hsieh, C.J. Large batch optimization for deep learning: Training BERT in 76 minutes. *arXiv* **2019**, arXiv:1904.00962v5.
73. Peng, Y.; Yan, S.; Lu, Z. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In Proceedings of the 18th BioNLP Workshop and Shared Task, Florence, Italy, 1 August 2019. [CrossRef]
74. Gururangan; Marasovi, A.; Lo, K.; Beltagy, I.; Downey, D.; Smith, N.A. Don't stop pretraining: Adapt language models to domains and tasks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8342–8360.
75. Suárez, P.J.O.; Sagot, B.; Romary, L. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In Proceedings of the 7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7), Cardiff, UK, 22 July 2019.

76. Cahyawijaya, S.; Winata, G.I.; Wilie, B.; Vincentio, K.; Li, X.; Kuncoro, A.; Ruder, S.; Lim, Z.Y.; Bahar, S.; Khodra, M.; et al. IndoNLG: Benchmark and Resources for Evaluating Indonesian Natural Language Generation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Virtual Event, 7–11 November 2021; pp. 8875–8898. [CrossRef]
77. Khanuja, S.; Bansal, D.; Mehtani, S.; Khosla, S.; Dey, A.; Gopalan, B.; Margam, D.K.; Aggarwal, P.; Nagipogu, R.T.; Dave, S.; et al. MuRIL: Multilingual representations for Indian languages. *arXiv* **2021**, arXiv:2103.10730v2.
78. Kakwani, D.; Kunchukuttan, A.; Golla, S.; Gokul, N.C. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and Pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 4948–4961. [CrossRef]
79. Xue, L.; Constant, N.; Roberts, A.; Kale, M. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 483–498.
80. Chi, Z.; Dong, L.; Ma, S.; Huang, S.; Singhal, S.; Mao, X.-L.; Huang, H.-Y.; Song, X.; Wei, F. mT6: Multilingual Pretrained Text-to-Text Transformer with Translation Pairs. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Virtual Event, 7–11 November 2021; pp. 1671–1683. [CrossRef]
81. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8440–8451. [CrossRef]
82. Chi, Z.; Huang, S.; Dong, L.; Ma, S.; Zheng, B.; Singhal, S.; Bajaj, P.; Song, X.; Mao, X.-L.; Huang, H.-Y.; et al. XLM-E: Cross-lingual Language Model Pre-training via ELECTRA. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 6170–6182. [CrossRef]
83. Chi, Z.; Dong, L.; Wei, F.; Yang, N.; Singhal, S.; Wang, W.; Song, X.; Mao, X.-L.; Huang, H.-Y.; Zhou, M. InfoXLM: An Information-Theoretic Framework for Cross-Lingual Language Model Pre-Training. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 3576–3588. [CrossRef]
84. 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. [CrossRef]
85. Caselli, T.; Basile, V.; Mitrović, J.; Granitzer, M. HateBERT: Retraining BERT for Abusive Language Detection in English. In Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021), Online, 6 August 2021; pp. 17–25. [CrossRef]
86. Zhou, J.; Tian, J.; Wang, R.; Wu, Y.; Xiao, W.; He, L.S. ENTI X: A Sentiment-aware pre-trained model for cross-domain sentiment analysis. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 568–579.
87. Ni, J.; Li, J.; McAuley, J. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 188–197.
88. Johnson, A.E.W.; Pollard, T.J.; Shen, L.; Lehman, L.-W.H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L.A.; Mark, R.G. MIMIC-III, a freely accessible critical care database. *Sci. Data* **2016**, *3*, 160035. [CrossRef]
89. Zellers, Y.C.R.; Holtzman, A.; Rashkin, H.; Farhadi, Y.B.A.; Roesner, F. Defending against neural fake news. *arXiv* **2020**, arXiv:1905.12616v3.
90. Idrissi-Yaghir, A.; Schäfer, H.; Bauer, N.; Friedrich, C.M. Domain Adaptation of Transformer-Based Models Using Unlabeled Data for Relevance and Polarity Classification of German Customer Feedback. *SN Comput. Sci.* **2023**, *4*, 1–13. [CrossRef]
91. Carmo, D.; Piau, M.; Campiotti, I.; Nogueira, R.; Lotufo, R. PTT5: Pretraining and validating the T5 model on Brazilian Portuguese data. *arXiv* **2020**, arXiv:2008.09144v2.
92. Filho, J.A.W.; Wilkens, R.; Idiart, M.; Villavicencio, A. The BRWAC corpus: A new open resource for Brazilian Portuguese. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018; pp. 4339–4344.
93. Gonçalo Oliveira, H.; Real, L.; Fonseca, E. (Eds.) Organizing the ASSIN 2 Shared Task. In Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese, Salvador, BA, Brazil, 15 October 2019; Volume 2583.
94. Xu, L.; Zhang, X.; Dong, Q. CLUECorpus2020: A large-scale Chinese corpus for pre-training language model. *arXiv* **2020**, arXiv:2003.01355v2.
95. Yuan, S.; Zhao, H.; Du, Z.; Ding, M.; Liu, X.; Cen, Y.; Zou, X.; Yang, Z.; Tang, J. WuDaoCorpora: A super large-scale Chinese corpora for pre-training language models. *AI Open* **2021**, *2*, 65–68. [CrossRef]
96. Liu, X.; He, P.; Chen, W.; Gao, J. Multi-Task Deep Neural Networks for Natural Language Understanding. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019. [CrossRef]
97. Khan, M.R.; Ziyadi, M.; AbdelHady, M. MT-BioNER: Multi-task learning for biomedical named entity recognition using deep bidirectional transformers. *arXiv* **2020**, arXiv:2001.08904v1.
98. Mulyar, A.; Uzuner, O.; McInnes, B. MT-clinical BERT: Scaling clinical information extraction with multitask learning. *J. Am. Med. Inform. Assoc.* **2021**, *28*, 2108–2115. [CrossRef]

99. Wang, Y.; Fu, S.; Shen, F.; Henry, S.; Uzuner, O.; Liu, H. The 2019 n2c2/OHNLP Track on Clinical Semantic Textual Similarity: Overview. *JMIR Public Health Surveill.* **2020**, *8*, e23375. [CrossRef]
100. Peng, Y.; Chen, Q.; Lu, Z. An Empirical Study of Multi-Task Learning on BERT for Biomedical Text Mining. In Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing, Online, 9 July 2020. [CrossRef]
101. Ganesh, P.; Chen, Y.; Lou, X.; Khan, M.A.; Yang, Y.; Sajjad, H.; Nakov, P.; Chen, D.; Winslett, M. Compressing Large-Scale Transformer-Based Models: A Case Study on BERT. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 1061–1080. [CrossRef]
102. Chen, Y.-P.; Chen, Y.-Y.; Lin, J.-J.; Huang, C.-H.; Lai, F. Modified Bidirectional Encoder Representations From Transformers Extractive Summarization Model for Hospital Information Systems Based on Character-Level Tokens (AlphaBERT): Development and Performance Evaluation. *JMIR Public Health Surveill.* **2020**, *8*, e17787. [CrossRef] [PubMed]
103. Meng, Y.; Speier, W.; Ong, M.K.; Arnold, C.W. Bidirectional Representation Learning From Transformers Using Multimodal Electronic Health Record Data to Predict Depression. *IEEE J. Biomed. Health Inform.* **2021**, *25*, 3121–3129. [CrossRef]
104. Rasmy, L.; Xiang, Y.; Xie, Z.; Tao, C.; Zhi, D. Med-BERT: Pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ Digit. Med.* **2021**, *4*, 86. [CrossRef] [PubMed]
105. Li, Y.; Rao, S.; Solares, J.R.A.; Hassaine, A.; Ramakrishnan, R.; Canoy, D.; Zhu, Y.; Rahimi, K.; Salimi-Khorshidi, G. BEHRT: Transformer for Electronic Health Records. *Sci. Rep.* **2020**, *10*, 7155. [CrossRef] [PubMed]
106. Huang, H.; Liang, Y.; Duan, N.; Gong, M.; Shou, L.; Jiang, D.; Zhou, M. Unicoder: A Universal Language Encoder by Pre-training with Multiple Cross-lingual Tasks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 2485–2494. [CrossRef]
107. Yang, J.; Ma, S.; Zhang, D.; Wu, S.; Li, Z.; Zhou, M. Alternating Language Modeling for Cross-Lingual Pre-Training. *Proc. Conf. AAAI Artif. Intell.* **2020**, *34*, 9386–9393. [CrossRef]
108. He, P.; Liu, X.; Gao, J.; Chen, W. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *arXiv* **2020**, arXiv:2006.03654v6.
109. Phang, J.; Févry, T.; Bowman, S.R. Sentence Encoders on STILTs: Supplementary training on intermediate labeled-data tasks. *arXiv* **2019**, arXiv:1811.01088v2.
110. Howard, J.; Sebastian, R. Universal Language Model Fine-tuning for Text Classificatio. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 328–339. [CrossRef]
111. Zhou, Y.; Srikumar, V. A Closer Look at How Fine-tuning Changes BERT. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 1046–1061. [CrossRef]
112. Merchant, A.; Rahimtoroghi, E.; Pavlick, E.; Tenney, I. What Happens To BERT Embeddings During Fine-tuning? In Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, Online, 20 November 2020; pp. 33–44. [CrossRef]
113. Mosbach, M.; Khokhlova, A.; Hedderich, M.A.; Klakow, D. On the Interplay Between Fine-tuning and Sentence-Level Probing for Linguistic Knowledge in Pre-Trained Transformers. In Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, Online, 20 November 2020; pp. 68–82. [CrossRef]
114. Hao, Y.; Dong, L.; Wei, F.; Xu, K. Investigating learning dynamics of BERT fine-tuning. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Suzhou, China, 4–7 December 2020; pp. 87–92.
115. Raghu, M.; Gilmer, J.; Yosinski, J.; Sohl-Dickstein, J. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6077–6086.
116. Pruksachatkun, Y.; Phang, J.; Liu, H.; Htut, P.M.; Zhang, X.; Pang, R.Y.; Vania, C.; Kann, K.; Bowman, S.R. Intermediate-Task Transfer Learning with Pretrained Language Models: When and Why Does It Work? In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 5231–5247. [CrossRef]
117. Mccreery, C.H.; Chablani, M.; Amatriain, X. For Medical Question Similarity. In Proceedings of the Machine Learning for Health (ML4H) at NeurIPS 2019, Vancouver, BC, Canada, 13 December 2019; pp. 1–6.
118. Cengiz, C.; Sert, U.; Yuret, D. KU_ai at MEDIQA 2019: Domain-specific Pre-training and Transfer Learning for Medical NLI. In Proceedings of the 18th BioNLP Workshop and Shared Task, Florence, Italy, 1 August 2019. [CrossRef]
119. Jeong, M.; Sung, M.; Kim, G.; Kim, D. Transferability of natural language inference to biomedical question answering. *arXiv* **2021**, arXiv:2007.00217v4.
120. Williams, A.; Nangia, N.; Bowman, S. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018. [CrossRef]
121. Bowman, S.R.; Angeli, G.; Potts, C.; Manning, C. A large annotated corpus for learning natural language inference. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015. [CrossRef]
122. Sun, C.; Yang, Z.; Wang, L.; Zhang, Y.; Lin, H.; Wang, J. Biomedical named entity recognition using BERT in the machine reading comprehension framework. *J. Biomed. Inform.* **2021**, *118*, 103799. [CrossRef]
123. Wang, Y.; Verspoor, K.; Baldwin, T. Learning from Unlabelled Data for Clinical Semantic Textual Similarity. In Proceedings of the 3rd Clinical Natural Language Processing Workshop, Online, 19 November 2020; pp. 227–233. [CrossRef]
124. Jiang, Z.; Yu, W.; Zhou, D.; Chen, Y.; Feng, J.; Yan, S. ConvBERT: Improving BERT with span-based dynamic convolution. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12837–12848.

125. Carlini, N.; Tramer, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.B.; Song, D.; Erlingsson, U.; et al. Extracting training data from large language models. In Proceedings of the 30th USENIX Security Symposium, Online, 11–13 August 2021; pp. 2633–2650.
126. Nakamura, Y.; Hanaoka, S.; Nomura, Y.; Hayashi, N.; Abe, O.; Yada, S.; Wakamiya, S.; Aramaki, E. KART: Privacy leakage framework of anguage models pre-trained with clinical records. *arXiv* **2022**, arXiv:2101.00036v2.