applied sciences

MDPI

*Review*

# From Word Embeddings to Pre-Trained Language Models: A State-of-the-Art Walkthrough

**Mourad Mars** [1,2]

[1] College of Computers and Information Systems, Umm Al-Qura University, Mecca 24382, Saudi Arabia
[2] Higher Institute of Computer Sciences and Mathematics, Monastir University, Monastir 5000, Tunisia

**Abstract:** With the recent advances in deep learning, different approaches to improving pre-trained language models (PLMs) have been proposed. PLMs have advanced state-of-the-art (SOTA) performance on various natural language processing (NLP) tasks such as machine translation, text classification, question answering, text summarization, information retrieval, recommendation systems, named entity recognition, etc. In this paper, we provide a comprehensive review of prior embedding models as well as current breakthroughs in the field of PLMs. Then, we analyse and contrast the various models and provide an analysis of the way they have been built (number of parameters, compression techniques, etc.). Finally, we discuss the major issues and future directions for each of the main points.

**Keywords:** artificial intelligence; NLP; pre-trained language model

check for updates

**Citation:** Mars, M. From Word Embeddings to Pre-Trained Language Models: A State-of-the-Art Walkthrough. *Appl. Sci.* **2022**, *12*, 8805. https://doi.org/10.3390/app12178805

Academic Editor: Valentino Santucci

Received: 19 April 2022
Accepted: 9 June 2022
Published: 1 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Copyright:** © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Word embeddings (i.e., distributed representations, word vectors) are dense feature vector representations of words in a specific dimensional space [1], which are usually learned by an unsupervised algorithm when fed with large amounts of tokens ($t_1$, $t_2$, …, $t_n$) [2]. Numerous variations have been proposed: context-free embeddings representation [3–5], and contextual representation (pre-trained language models) [6]. Traditional context-free word embedding approaches aim to learn a global word embedding matrix $E \in \mathbb{R}^{V*d}$, where $V$ is the vocabulary size and $d$ is the number of dimensions. In fact, each row $e_i$ of $E$ corresponds to the embedding of the token $t_i$ in the vocabulary $V$. Alternatively, methods that learn contextual embeddings can associate to a token $t_i$ different vectors (different meanings) depending on the context in which the token appears [7].

The above studies have tried to improve word representations, but they still have a key limitation, which is producing one embedding vector for each word in almost all word embeddings, despite the fact that a word may have different meanings. With the large amounts of available data (e.g., Wikipedia dumps, news collections, Web Crawl, The Pile, etc.) and the recent advances in deep learning over recent years such as transformers, model capacity, and computational efficiency, a different number of methods have been introduced to overcome drawbacks of word embeddings [8–11].

The rest of this paper is organized as follows. Section 2 provides background concepts and distinguishes between the different categorizations of word embeddings (frequency-based and context-free word embeddings). In Section 3, we give a deep review of the different SOTA pre-trained language models. Section 4 describes the application ways, parameters, and different compression methods used to create PLMs. Section 5 discusses the current challenges and suggests future directions. Finally, we draw the conclusion of this research paper.

## 2. Language Models and Word Embeddings

There are a variety of pre-training general language representations. We distinguish two generations of word embeddings: frequency-based word vectors and context-free embedding representation. An overview of the most popular approaches is presented here.

**Latent Semantic Analysis (LSA)** is a representation method of word meaning introduced in [12]. It is a fully automated statistical method to extract relationships between words based on their contexts of use in documents or sentences. LSA is an unsupervised learning technique (Mathematical details: [13]). Fundamentally, the proposed model is concretized through four main steps: (1) Term-Document Matrix calculation also known as the bag of words model; (2) Transformed Term-Document Matrix; (3) Dimension Reduction [14]; and (4) Retrieval in Reduced Space.

**Neural Network Language Model (NNLM)**

The Neural Network Language Model (NNLM) [2,15] jointly learns a word vector representation and a statistical language model with a feedforward neural network that contains a linear projection layer and a non-linear hidden layer. N-dimensional one-hot vector that represents the word is used as the input, where N is the size of the vocabulary. The input is first projected onto the projection layer. Afterwards, a softmax operation is used to compute the probability distribution over all words in the vocabulary. As a result of its non-linear hidden layers, the NNLM model is very computationally complex. To lower the complexity, an NNLM is first trained using continuous word vectors learned from simple models. Then, another N-gram NNLM is trained from the word vectors.

**Max–Margin Loss (MML)** Collobert and Westion [16] proposed word embeddings by training a model on a large dataset. They consider a window approach network. Max margin or hinge loss is a technique that tries to get a higher score for words that are correct than for words that are not.

**Word2Vec** is a model, developed by Google, to build word embeddings [3]. An intrinsic difference with previously developed methods is that Word2vec is a predictive model, while LSA and MML are statistical. The Word2vec architecture is based on a feedforward neural network language model [2]. Word2vec can be obtained from two different models (both involving Neural Networks): Skip-Gram and Common Bag Of Words (CBOW). The Skip-Gram model trains by trying to predict the context words from the target, while the CBOW model aims at predicting the current word from the sum of all the word vectors in its surrounding context. Using a contextual window, Word2Vec is able to learn semantic meaning and similarity between words in a fully unsupervised manner. This means that words with similar meanings (e.g., king, queen) tend to occur in a semantic space in close proximity to each other. The CBOW model is quicker as it treats the entire context as one entity, while the skip-gram model produces various training pairs for each context word. The Skip-gram model does a great job of capturing rare words due to the way it manages the context.

After the emergence of Skip-Gram and CBOW-based word embedding models, two other models based on these methods were announced. In the rest of this section, we will present global vectors and fastText.

**Global vectors (Glove)** is an unsupervised learning algorithm, developed by Stanford University, based on a word-word co-occurence matrix that is used to create the embeddings [4]. Each row of the matrix represents a word, while each column represents the contexts that words can appear in. The matrix values represent the frequency with which a word appears in a given context. Finally, an embedding matrix where each row will be a word's embedding vector for the corresponding word is created by applying a dimensionality reduction [14].

**FastText** is a method for learning word embeddings for large datasets [17,18]. It can be seen as an improvement of Word2Vec. Instead of considering words as distinct entities, FastText aims to learn vector representations for each word and to exploit the character and morphological structure of the word by considering every word as a composition of n-grams character (e.g., "unbalanced" = "un" + "balance" + "ed"). Thus, words can be

represented by averaging the embeddings of their n-grams (sub-words). While this will increase the training cost, the sub-word representation permits fastText to represent words more efficiently, allowing the estimation of out-of-vocabulary (OOV) and rare words since their character-based n-grams should occur across other words in the training dataset.

**CoVe** (Context Vectors model), introduced by McCann et al. [19] from SalesForce Research, learns contextual vectors using a neural machine language translation system. It is so viewed as a sequence-to-sequence model. CoVe shows that the learned representations are transferable to other NLP tasks. In order to obtain contextual embeddings, CoVe uses a deep LSTM encoder from a sequence-to-sequence model trained for machine translation. Empirical results revealed that boosting non-contextualized word representations (e.g., Word2Vec, Glove) with CoVe embeddings (LM + CoVe) improves performance over a variety of common NLP tasks.

**FLAIR** embeddings were introduced by Zalando research [20]. FLAIR embeddings are character-level language models. FLAIR proposed embeddings have the distinct properties of being trained without any explicit concept of words and thus fundamentally modeling words as sequences of characters using the BPEmb technique presented by Heinzerling and Strube [21]. In FLAIR, the same word may have different embeddings depending on the context.

## 3. SOTA Pre-Trained Language Models (PLMs)

### *3.1. List of Recent Pretrained Language Models*

With the large amounts of available data (e.g., Wikipedia dumps, news collections, Web Crawl, and The Pile) and the recent advances in deep learning over recent years, such as transformers (Figure 1), model capacity, and computational efficiency, a different number of methods have been introduced to overcome the drawbacks of word embeddings.

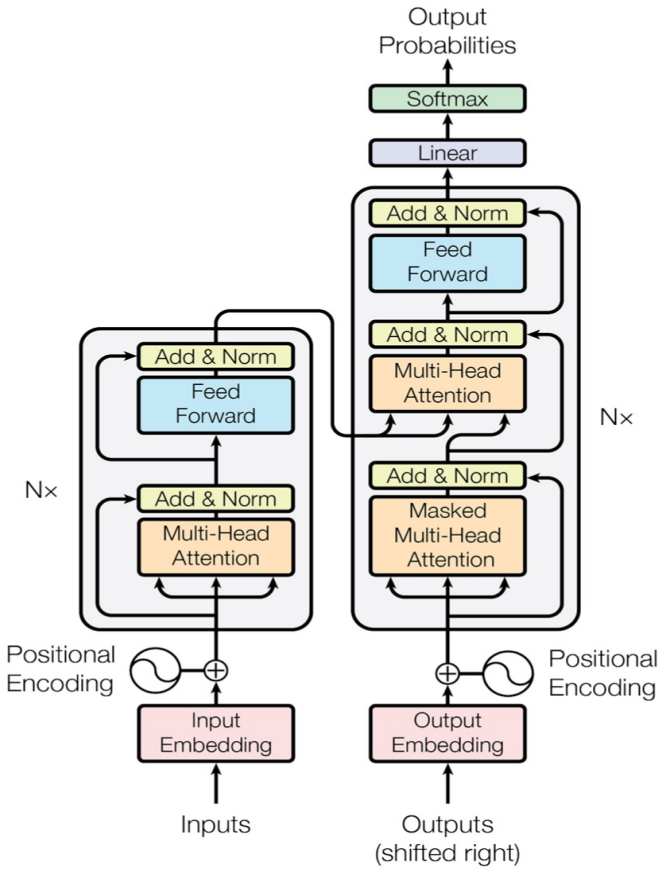


**Figure 1.** An illustration of Transformer Architecture: Figure source [22].

Language pre-training has been highly successful, with state-of-the-art models such as GPT, BERT, XLNet, ERNIE, ELECTRA, and T5, among many others [8,19,20,23–25]. These methods share the same idea, pre-training the language model in an unsupervised fashion on vast amounts of datasets, and then using this pre-trained model for fine-tuning for various downstream NLP tasks. Fine-tuning [8,23–26] typically adds an extra layer(s) for the specific task and further trains the model using a task-specific annotated dataset, starting from the pre-trained back-bone weights (Figure 2).

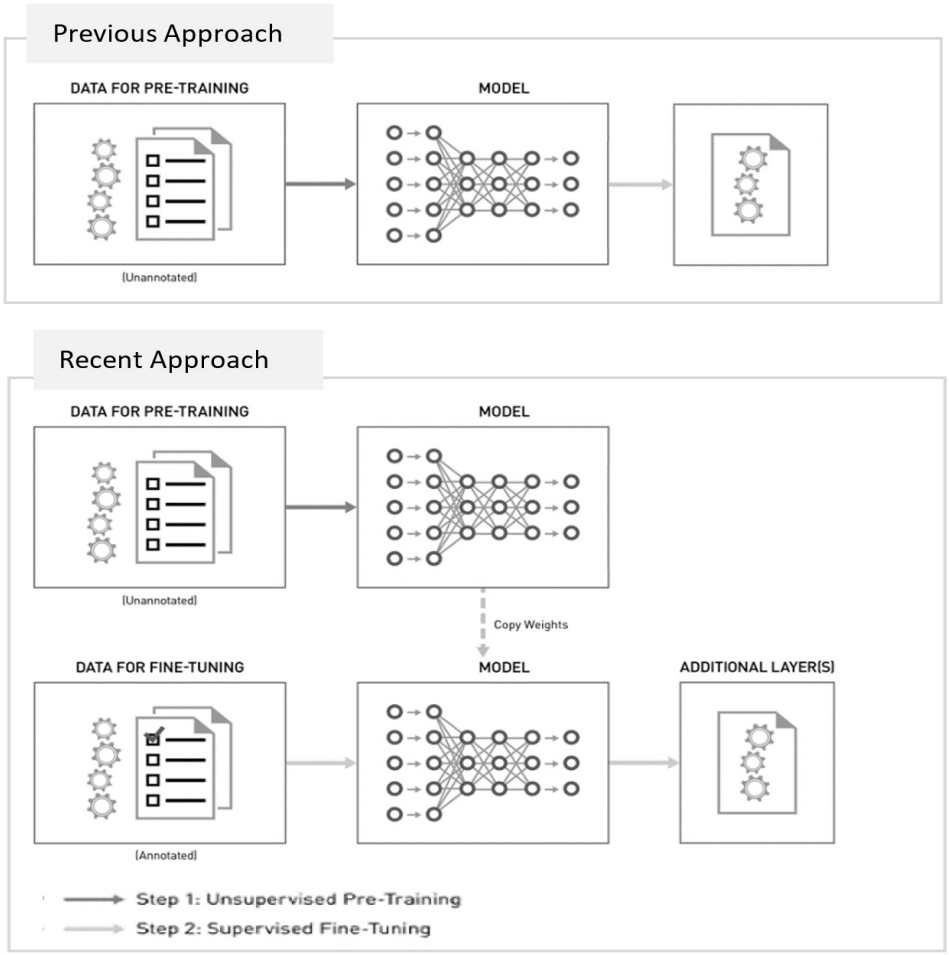


**Figure 2.** Previous vs. recent approaches.

Recently, different unsupervised pre-training objectives have been explored in the literature (autoregressive (AR) language modelling, autoencoding (AE), Language Model (LM), Masked Language Model (MLM), Next Sentence Prediction (NSP), Replaced Text Detection (RTD), etc.). In this section, we will present recent pre-trained language models that have significantly advanced the SOTA over a variety of NLP tasks.

**ELMO**, short for the Embeddings from Language Model [27], comes to overcome the limitations of previous word embeddings. ELMo representations are a function of the internal layers of the bi-directional Language Model (biLM). While the input is a sequence of tokens, the language model learns to predict the probability of the next token given the previous ones (a task called "Language Modeling"). In the forward pass, the sequence contains tokens before the target token. In the backward pass, the history contains words after the target token. The deep BiLSTM architecture helps ELMo learn more context-dependent aspects of the meanings of words in the top layers along with syntax aspects in the lower layers. This leads to better word embeddings and different representations of a word depending on the context in which it appears. Peters et al. [27] demonstrated that the biLM-based representations outperformed CoVe in all considered tasks.

**ULMfit** (Universal Language Model Fine-tuning). Howard and Ruder [28], from FastAI, have suggested ULMfit. The core concept behind this model is that a single model architecture can be used for language pre-training and fine-tuned for supervised downstream classification tasks by using the weights acquired through pre-training. Howard and Ruder [28] demonstrate the importance of a number of novel techniques, including discriminative fine-tuning, slanted triangular learning rate, and gradual unfreezing, for maintaining prior knowledge and ensuring stable fine-tuning.

**OpenAI GPT Transformer**, short for Generative Pre-training Transformer [24], is a multi-layer transformer decoder [22]. Its approach is a combination of two existing ideas: transformers [22] and unsupervised pretraining [28]. GPT is a large auto-regressive language model pretrained with language modelling (LM), a simple architecture that can be trained faster than an LSTM-based model. GPT uses the BookCorpus dataset [29], which contains more than 7000 books from various genres. It is able to learn complex patterns in the data by using the attention mechanism. The total number of trained parameters is 110M parameters.

**BERT** (Bidirectional Encoder Representations from Transformers) introduced by Devlin et al. [8] is built on a number of clever ideas that have been emerging in the NLP recently including Semi-supervised Sequence Learning [26], ELMo [27], ULMFiT [28], the Transformer [22], and the OpenAI transformer [24]. BERT is the first deeply bidirectional, unsupervised language representation, pretrained using only a large text corpus and jointly conditioning both the left and right context of each token. BERT is pretrained with a linear combination of two objectives: (1) masked language modelling (MLM) objective, which randomly masks 15% of the input sequence and tends to predict these masked tokens. (2) Next Sentence Prediction (NSP) which is given two sentences, does the second sentence truly follow the first in the original text? BERT comes in different versions: a 12-layer BERT-base model (with 12 attention heads and 110M parameters) and a 24-layer BERT-large model (with 16 attention heads and 340M parameters) [8].

**Transformer-XL:** Attentive Language Models Beyond a Fixed-Length Context [30]. The problem of long-distance dependence is a challenging problem, and it is a task where RNN fails. Transformer-XL is an auto-regressive pre-trained language model developed by the Google AI team that introduces architectural modifications enabling transformers to understand context beyond that fixed-length limitation without disrupting temporal coherence via segment-level recurrence and relative positional encoding schemes. With its 257M parameters, Transformer-XL is more than 1800 times faster than a typical transformer. The XL here refers to extra long, which means that it has very good performance on the long-distance dependency problem in language modelling (80% longer than RNN). At the same time, it also implies that it was born to solve the problem of long-distance dependence.

**GPT2** [31], OpenAI GPT's successor, is a large auto-regressive language model. This large transformer-based language model has 1.5 billion parameters. GPT-2 was trained on a dataset of 8 million web pages (40 GB of Internet text) called WebText ($10 \times$ larger than GPT), and it was trained simply to predict the next word (Language Model objective). GPT2 showcased zero-shot task transfer capabilities for various tasks such as machine translation and reading comprehension.

**ERNIE 2.0** [32], a continual pre-training framework for language understanding, is a pre-trained language model based on Baidu's own deep learning framework, PaddlePaddle, which can incorporate lexical, syntactic, and semantic information at the same time. We distinguish two general pre-training tasks: word-aware pre-training tasks (which include the Knowledge Masking Task, Capitalization Prediction Task and the Token-Document Relation Prediction Task) and semantic-aware pre-training tasks (which include Sentence Reordering, Sentence Distance, Discourse Relation Task, and IR Relevance Task). ERNIE was trained for English and Chinese languages. The experimental results show that significant improvements have been made in different knowledge-driven tasks, and they are comparable to the existing BERT and XLNET models on other common tasks. ERNIE 2.0 version ranks first in the GLUE rankings (as of January 2020).

**XLNet** is proposed by researchers at Google Brain and CMU [33]. It borrows ideas from Transformer-XL autoregressive language modeling [33] and autoencoding (e.g., BERT). XLNet avoids the drawbacks of previous techniques by introducing a variant of language modeling called permutation language modeling (PLM) during the training process, where the order of the next token prediction can be right to left or left to right (randomly). This drives the model to capture the bidirectional dependencies (bidirectional contexts) and make it a generalized order-aware autoregressive language model. With its 340M parameters, XLNet beats BERT on 20 NLP downstream tasks and succeeds in the SOTA results on 18 tasks.

**RoBERTa:** Robustly Optimized BERT Pretraining Approach [34], developed by Facebook, is built on BERT's language masking strategy and makes some changes related to the key hyperparameters in BERT. RoBERTa uses 160 GB of training data instead of the 16 GB dataset originally used with BERT. To improve the training procedure [34], RoBERTa eliminates the Next Sentence Prediction (NSP) task. It also introduces dynamic masking so that a new masking pattern is generated each time a sentence is fed into training, whereas BERT use a fixed masked token during training. It was also trained more iteratively (more epochs and larger mini-batch sizes). The total number of parameters reached 355M parameters.

**XLM** (Cross-lingual Language Model Pretraining) was developed by Facebook [35]. XLM uses a preprocessing technique known as Byte Pair Encoding (BPE) and a dual-language training mechanism with BERT in order to learn relations between words in different languages without any cross-lingual supervision. Various objectives were used to pretrain XLM, including the causal language modeling (CLM) objective (next token prediction), a masked language modeling (MLM) objective (BERT-like), or a Translation Language Modeling (TLM) object (extension of BERT's MLM to multiple language inputs). The model outperforms previous models in a multi-lingual classification task and significantly improves machine translation when a pretrained model is used for the initialization of the translation model. XLM shows the effectiveness of pretrained representations for cross-lingual language modeling (both on monolingual data and parallel data) and cross-lingual language understanding (XLU).

**CTRL** (Conditional Transformer Language) is a 1.63B parameter PLM [36]. A research team at Salesforce trained CTRL on a very large corpus of around 140 GB of text data with a causal language modeling (CLM) objective. It has a powerful and controllable artificial text generation function, allowing it to generate syntactically coherent writing (predicting the next token in a sequence). CTRL can also improve other NLP applications by fine-tuning specific tasks or transferring the learned representation of the model.

**Megatron-ML** was proposed by NVIDIA in "Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism" [37]. Megatron-LM is an NLP model released by NVIDIA. They train an 8.3 billion parameter transformer language model similar to GPT-2 and a 3.9 billion parameter model similar to BERT. It is a huge model that is over 24× the size of BERT and approximately 6× the size of OpenAI's GPT-2. Increasing the size of both GTP-2 and BERT's models made them faster and more accurate.

**ALBERT:** In conjunction with the trend of larger and larger models (as with the huge Megatron, trained on 512 GPUs), it is fascinating to observe how new architecture enhancements, such as ALBERT (a lite version of BERT), provide higher accuracy with substantially fewer parameters [38]. It is the outcome of a collaboration between Google's research team and Chicago's Toyota Technological Institute. The goal is to introduce two parameter-reduction techniques (decomposed parameterized embedding and cross-layer parameter sharing) to address the problems of training length and memory restrictions. When compared to the original version of BERT, this model scales substantially better. ALBERT has 18× fewer parameters and can be trained about 1.7× faster than BERT-large. For pretraining, ALBERT utilizes its own training method called Sentence-Order Prediction (SOP), which differs from the NSP objective for BERT. The problem with NSP, according to the authors, is that it conflates topic prediction and coherence prediction.

**DistilBERT**: Knowledge distillation method, which was initially described in 2006 [39] and 2015 [40], is a compression technique in which a compact model is trained to replicate the behaviour of a larger model or an ensemble of models. Developed by HuggingFace [41], DistilBERT learns a distilled (approximate) version of BERT, retains 95% performance on GLUE while employing half the number of parameters (only 66 million parameters, instead of 110M/340M). The idea is that when a large neural network has been trained, its whole output distributions of the network can be approximated by using a smaller network. DistilBERT is smaller, faster, cheaper, and lighter compared to BERT, yet its performance is about equivalent to that of BERT. Sharing the same tokenizer as BERT's bert-base-uncased, DistilBERT processes the sentence and passes along some information to the next model. DistilBERT is a smaller version of BERT developed and open sourced by the team at HuggingFace.

There are numerous tentative models, such as TinyBERT [42], BERT-48 [43], BERT-192 [43], MobileBERT [44], MiniBERT [45], and others.

**DistilGPT2** is a model that has been pretrained under the supervision of GPT2 (the smallest of the GPT-2 variants) using the OpenWebTextCorpus dataset. With 82M parameters, DistilGPT-2 is smaller than even the smallest version of GPT-2 (124M parameters). GPT2 has been compressed into DistilGPT2 using the same approach as the knowledge distillation technique. DistilGPT2 is approximately twice as fast as GPT2 OpenAI. A primary goal is to keep the weight down as much as possible.

**T5** is a large pre-trained encoder–decoder model that converts all NLP problems, whether they are unsupervised or supervised, into a text-to-text format [46]. It is trained using teacher forcing. This means that when we train, we always need an input sequence and a target sequence that goes with it. T5 performs well on a wide range of tasks out-of-the-box by prepending a different prefix to the input corresponding to each task (e.g., translation: translate, English to Arabic, summarization: summarize, etc.). Various T5 variants (T5-small, T5-base, T5-large, T5-3b, T5-11b.) can be trained/fine-tuned both in a supervised and unsupervised fashion.

**BART** is a type of PLM. Facebook first described it in the paper "BART: Denoising Sequence-to-Sequence Natural Language Generation, Translation, and Comprehension Pre-training" [47]. BART is a transformer encoder-encoder (seq2seq) model. It comes with two main parts: a bidirectional encoder (BERT-like) and an autoregressive decoder (GPT-like). BART is learned by first corrupting text using an arbitrary noise function and developing a model to restore the original text. BART is most effective when fine-tuned for text generation (e.g., summarization, translation), but it is equally effective for comprehension tasks (e.g., text classification, and question answering).

**XLM-R** is a new cross-lingual language model from Facebook AI, with the letter "R" standing for Roberta [48]. XLM-R is a transformer-based multilingual masked language model that has been pre-trained on a dataset in 100 languages (2.5 TB of newly created clean Common Crawl data). It outperforms prior released multi-lingual models, such as mBERT or XLM, on the GLUE and XNLI benchmarks (classification, sequence labelling, and question answering).

**PEGASUS**: Pre-training with Extracted Gap-sentences for Abstractive Summarization. Based on Transformer, Google's PEGASUS is a generative language model capable of generating words to complete open text tasks [49]. With a new self-supervised objective called gap-sentence generation (GSG) for Transformer encoder–decoder models, the fine-tuning performance on abstractive summarization was improved, achieving state-of-the-art results across a wide range of summarization datasets.

**Reformer** architecture [50] extends the boundaries of long sequence modeling by processing up to 500,000 tokens at once. The Reformer authors came up with four new features (Reformer Self-Attention Layer, Chunked Feed Forward Layers, Reversible Residual Layers and Axial Positional Encodings). Many NLP tasks, e.g., summarization and question answering, can benefit from this model as it is designed to efficiently handle very long sequences of data.

**MT-DNN** (Multi-Task Deep Neural Network), introduced by Microsoft [9], is intended to enable quick customisation for a broad range of natural language understanding tasks, utilizing a variety of algorithms (classification, regression, structured prediction) and text encoders (e.g., RNNs, UniLM, RoBERTa, BERT). The adversarial multi-task learning paradigm is a key aspect of MT-DNN that enables robust and transferable learning. MT-DNN enables multi-task knowledge distillation, which compresses a DNN model significantly without sacrificing performance.

**REALM** model was proposed in REALM: Retrieval-Augmented Language Model Pre-Training [51]. In this retrieval-augmented language model, documents are retrieved from a textual knowledge corpus and used to process question–answer tasks.

**T-NLG** (Turing Natural Language Generation) is a Transformer-based generative language model developed by the Microsoft Turing team. It is a type of generative language model [52]. Using its 17B learned parameters (78 transformer layers with a hidden size of 4256 and 28 attention heads), T-NLG can generate words to perform open-ended textual tasks, enhancing incomplete sentences, producing answers to questions and summarizing documents. In a nutshell, the goal is to answer as correctly and smoothly as possible, as humans are capable of doing in any particular situation. T-NLG was the biggest model ever issued (in early 2020). This research would not have been possible without the use of the DeepSpeed library and the ZeRO optimizer.

**ELECTRA** is an acronym that stands for "Efficiently Learning an Encoder that Classifies Token Replacements Accurately" [53]. ELECTRA uses a new pretraining task, called "replaced token detection" (RTD), to train a bidirectional model (similar to a masked language model (MLM)) that learns from all input positions (like a LM). Instead of training a model to recover masked tokens "[MASK]" such as BERT, ELECTRA trains a discriminator model to distinguish between "real" and "fake" input tokens (corrupted tokens were replaced by a generator network: GANs).

**TAPAS**: Weakly Supervised Table Parsing via Pre-training, is a BERT-like transformer model using just raw tables and their related texts [54]. More precisely, it was pretrained with two particular objectives: Masked Language Model (MLM) and Intermediate pre-training to predict (classify) whether a sentence is supported or refuted by the contents of a table. In this manner, the model learns an inner representation of the English language used in tables and associated texts, which can then be used to extract features useful for downstream tasks such as answering questions about a table or determining whether a sentence is entailed or refuted by the contents of a table. To fine-tune the pre-trained model, one or more classification heads were added.

**LongFormer** model (The Long-Document Transformer) was proposed by Allen AI [55]. Previous models had a significant disadvantage in that they could not handle longer sequences (e.g., BERT, which is limited to a max of 512 tokens at a time). LongFormer is suggested to overcome these long sequence issues. Self-attention computing is reformulated in this transformer-based design in order to decrease model complexity. The long-former model is better than the Transformer-XL model and better than RoBERTa when it comes to long document tasks.

**MPNet** (Masked and Permuted Pre-training for Language Understanding) [56] is a revolutionary pre-training approach that retains the strengths of BERT and XLNet while avoiding the drawbacks (limitations of Masked Language Modeling in BERT and Permuted Language Modeling in XLNet), achieving a better level of accuracy.

**RAG**, short of Retrieval-augmented generation [57], models combine the capabilities of pre-trained dense retrieval (Facebook AI's DPR system) with seq2seq (BART model) generator models to produce a more powerful model. These models perform well on QA tasks and language generation tasks.

**GPT-3**, an autoregressive model, developed by researchers at OpenAI [10]. About 3 trillion words were utilized to train GPT-3, including data from the Common Crawl corpus, as well as data from the web, books, and Wikipedia. In GPT-3, there are 175 billion parameters. There are ten times more parameters than the MT-NLG language model and

100 times more parameters than GPT-2. GPT-3 performs effectively in downstream NLP tasks under zero-shot and few-shot settings due to the huge number of parameters and large dataset training.

**DeBERTa** (Decoding-enhanced BERT with disentangled attention) was pre-trained on vast volumes of raw text using self-supervised learning [58]. As with previous PLMs, the goal of DeBERTa is to provide generic representations of language that may be used in a variety of downstream NLU applications. With the help of three advanced techniques, DeBERTa enhances SOTA PLMs (including BERT, RoBERTa, and UniLM). These techniques are a disentangled attention mechanism, an improved mask decoder, and a fine-tuning virtual adversarial training method.

**MARGE** is a seq2seq model that has been trained with an unsupervised paraphrasing objective as an alternative to the MLM objective [59]. With a little fine-tuning, this leads to very good results on a wide range of discriminative and generative tasks in a lot of different languages.

**GShared**: introduced in "Scaling Giant Models with Conditional Computation and Automatic Sharding" [60]. Gshared (a sequence-to-sequence Transformer model with 600B parameters) attempts to improve the training efficiency of large-scale models using sparsely-gated Mixture-of-Experts (MoE) layers. The Gshared model aids in scaling up multilingual neural machine translation.

**MT5** by Google [61] is a variation of T5 (multilingual T5). It is a massively multilingual, pretrained text-to-text transformer model. MT5 is trained in a unsupervised manner on the mC4 corpus [61] following a similar way as T5. MT5 supports a total of 101 languages. MT5 demonstrated that the T5 recipe is easily adaptable to a multilingual environment and achieved strong performance on a diverse set of benchmarks.

**KEPLER** is an acronym for Knowledge Embedding (KE) and Pre-trained LanguagE Representation [62], and it is used to enhance the integration of factual knowledge into PLMs as well as to develop effective text-enhanced KE with strong PLMs. Experimental results show that KEPLER achieves state-of-the-art performances on various NLP tasks and also works remarkably well as an inductive KE model on knowledge graphs (KG) link prediction. The authors also built Wikidata5M for pretraining and evaluation. KEPLER does well as an inductive KE model when it comes to predicting KG links. It also does well on different NLP tasks.

**XLM-E** [63], released by Microsoft, is an across-lingual language model that has been pre-trained by ELECTRA-style tasks [53]. It is pretrained with the masked LM, translation LM, multilingual replaced token detection (MRTD), and translation placed token detection (TRTD). The XLM-E model beats the baseline models on a variety of cross-lingual comprehension tests while consuming significantly less computing time than the baseline models. Furthermore, the results of the investigation reveal that XLM-E has a higher likelihood of achieving superior cross-lingual transferability.

**The Megatron-Turing NLG** (MT-NLG) pre-trained language model, a joint effort from NVIDIA and Microsoft, has 530 billion paremeters [64]. It is trained using Megatron and DeepSpeed on DGX SuperPod. In comparison to the existing largest model, it has three times the number of parameters as the current biggest model of this type (Turing-NLG 17B and Megatron-LM). In zero-, one-, and few-shot situations, the 105-layer transformer-based MT-NLG outperformed previous state-of-the-art models and established a new benchmark for large-scale language models in both model scale and quality across a wide range of natural language tasks (e.g., completion prediction, reading comprehension, commonsense reasoning, NL inferences, and word sense disambiguation).

**T0**: Multitask Prompted training Enables Zero-Shot Task Generalization [65]. It is a variation of the T5 encoder–decoder model that receives textual inputs and generates target answers. It is trained on a multitask combination of natural language processing datasets that have been divided into distinct tasks. Each dataset has numerous prompt templates that are used to structure example instances as input/target pairs. After training on a varied set of tasks, the T0 model is evaluated on zero-shot generalization to previously

unseen tasks. This method is an excellent alternative to unsupervised language model pretraining since it allows the T0 model to achieve exceptional zero-shot performance on numerous standard datasets and outperform models many times its size.

**KnGPT2** [66] is a compressed version of GPT-2 using Kronecker decomposition followed by an extremely little pre-training on a tiny percentage of the training data with intermediate layer knowledge distillation (ILKD). A comparison of KnGPT2's performance on benchmark tasks for language modeling and the General Language Understanding Evaluation reveals that it outperforms its competitor (DistilGPT2).

**GLaM** (Generalist Language Model), is a series of language models from Google AI [67]. The largest GLaM has 1.2 trillion parameters (7× larger than GPT-3). GaLM scales model capacity via a sparsely active mixture-of-experts (MoE) architecture, resulting in better overall zero-shot and one-shot performance across 29 natural language processing tasks.

**Gopher**: DeepMind trained a series of transformer language models of various sizes, spanning from 44 million parameters to 280 billion parameters. The goal is to identify which model size (scale) substantially enhances performance the most (the largest model named Gopher [68]).

**XGLM** is a multilingual language models [69] developed by Meta AI. It is trained using a large-scale multilingual dataset with tokens from 30 different languages, for a total of 500 billion tokens. XGLM largest release has 7.5B parameters. It is demonstrated that XGLM7.5B outperforms a GPT-3 model of equivalent size (6.7B parameters), especially in the less-resourced languages. A variety of reasoning and natural language inference tests show that XGLM7.5B has high zero and few-shot learning capabilities.

**LaMDA** is a family of neural language models based on transformers that are optimized for dialog applications [70]. It contains around 137B parameters and has been pre-trained on 1.56T words of public conversation data and online content. Quality, safety, and groundedness are the three main factors (metrics) used to evaluate LaMDA. Their findings indicate that model scaling enhances the quality (sensibility, specificity, and interestingness), safety, and groundedness metrics to some extent. On the other hand, combining scaling with fine-tuning yields significant gains in performance on all metrics.

**GPT-NeoX-20B**, a collaborative initiative between EleutherAI and CoreWeave in early 2022 [11]. According to its architecture, it is an autoregressive transformer decoder model that was developed in the same manner as the Generative Pre-trained Transformer 3 (GPT-3). With its 20 billion parameters trained on a curated dataset of 825 GB named the Pile [71], EleutherAI aims to make GPT-NeoX-20B the largest publicly available pretrained general-purpose autoregressive LM. An evaluation of the model was performed using Gao et al. [71], an open source codebase for language model evaluation, which includes a variety of model APIs. GPT-NeoX-20B excelled in knowledge-based and mathematical tasks.

**Chinchilla** (70B Parameter), DeepMind's new pre-trained language model, contributes to the development of an effective training paradigm for large auto-regressive language models with low computational complexity [72]. In order to determine the optimal model size and training length, three techniques have been presented: experimenting with different model sizes and training token numbers, as well as IsoFLOP profiles and parametric loss functions to fit a model. Chinchilla significantly outperforms Gopher (280B), GPT-3 (175B), and Megatron-Turing NLG (530B).

**PaLM**: is a Pathways Language Model recently unveiled by Google AI [73]. It is a 540-billion parameter transformer language model trained using Pathways. Pathways is a novel ML approach recently presented by Barham et al. [74]. It enables extremely efficient training of very large neural networks. This pre-trained language model showcases the impact of scale in the context of few-shot learning.

**OPT**: Open Pre-trained Transformer Language Models [75]. Proposed by Meta AI, it is a suite of decoder-only pre-trained transformers ranging from 125M to 175B parameters. OPT-175B is trained on public datasets (CCNewsV2, a subset of the Pile, PushShift.io Reddit, etc.). Results show that OPT-175B (96 layers, 96 attention heads, 12,288 embedding size, 2M global battch size) performance is comparable to OpenAI's GPT-3, while requiring

only 1/7th of the carbon footprint to develop. To enable researchers to study the effect of scale alone, Meta AI released a smaller-scale baseline models (125M, 350M, 1.3B, 2.7B, 13B, 30B), trained on the same dataset and using similar settings as OPT-175B (weight initialization: Megatron-LM, optimizer: AdamW, and a dropout of 0.1 throughout).

### 3.2. Summary and Comparison

Recent breakthroughs in neural architectures, such as the Transformer, coupled with the increased availability of large-scale datasets, have revolutionized the field of large language models. This advanced the state-of-the-art in a variety of NLP tasks and increased the use of PLMs (BERT, GPT-3, XLNet, GPT-NEO, T5, etc.) in production contexts. Different architectures with different settings have been proposed. Table 1 lists the settings for training selected pre-trained language models.

**Table 1.** Model architecture details in terms of number of layers, number of attention heads, dimension of contextual embedding and trained parameters.

| Model | Layers | Attention Heads | CE Dimension | Parameters |
|---|---|---|---|---|
| GPT | 12 | 12 | 768 | 117M |
| BERT-Large | 24 | 16 | 1024 | 340M |
| GPT-2-1.5B | 48 | 12 | 1600 | 1.5B |
| RoBERTa | 24 | 16 | 1024 | 355M |
| DistilBERT-Base | 6 | 12 | 768 | 66M |
| ALBERT-Base | 12 | 12 | 768 | 12M |
| ALBERT-Large | 24 | 16 | 1024 | 18M |
| XLNet | 24 | 16 | 1024 | 340M |
| ELECTRA | 24 | 16 | 1024 | 335M |
| Megatron-LM | 72 | 32 | 3072 | 8.3B |
| T5-11B | 24 | 128 | 1024 | 11B |
| CTRL | 48 | 16 | 1280 | 1.63B |
| Longformer-Large | 24 | 16 | 1024 | 435M |
| Pegasus | 16 | 16 | 1024 | 568M |
| Turing-NLG | 78 | 28 | 4256 | 17.2B |
| OPT-125M | 12 | 12 | 768 | 125M |
| OPT-175B | 96 | 96 | 12,288 | 175B |

As previously stated, the availability of public datasets has risen in recent years. As a result, the size of PLMs training datasets increased. Table 2 contains an overview of selected PLMs. We have provided all of the datasets used for training, including (Wikipedia, the Pile, Common Crawl, etc.) of selected language models.

**Table 2.** Summary of major datasets used in pre-training language models.

| Name | Lab | Param. | Dataset Sources | Data Size |
|---|---|---|---|---|
| Megatron MT-NLG | NVIDIA and Microsoft | 530B | The Pile v1 + more:<br>OpenWebText2 (Reddit links)<br>Stack Exchange and PubMed Abstracts<br>Wikipedia and Books3<br>Gutenberg (PG-19) and BookCorpus2<br>NIH ExPorter<br>Pile-CC<br>ArXiv, GitHub<br>Common Crawl 2020 and2021<br>+ RealNews (120 GB).<br>+ CC-Stories (31 GB). | >825 GB |

**Table 2.** *Cont.*

| Name | Lab | Param. | Dataset Sources | Data Size |
|---|---|---|---|---|
| GPT-2 | Open AI | 1.5B | WebText (40 GB) | 40 GB |
| BERT | Google | 345M | BooksCorpus [4] 800M words<br>Wikipedia 2500M<br>English Wikipedia (12 GB)<br>+ BookCorpus (4 GB). | 16 GB |
| Megatron-11B | FAIR | 11B | English Wikipedia (12 GB)<br>+ BookCorpus (4 GB)<br>+ CC-News(76 GB).<br>+ OpenWebText/Reddit upvoted (38 GB).<br>+ Stories CC (31 GB). | 161 GB |
| RoBERTa.-Base | Facebook AI and Univ. of Wash. | 125M | English Wikipedia (12 GB)<br>+ BookCorpus (4 GB)<br>+ CC-News (76 GB).<br>+ OpenWebText/Reddit (38 GB).<br>+ Stories CC (31 GB). | 161 GB |
| Megatron-LM | NVIDIA | 8.3B | Wikipedia and OpenWebText<br>RealNews<br>+ CC-Stories. | 174 GB |
| Fairseq. | Meta AI | 13B | English Wikipedia (12 GB)<br>+ BookCorpus (4 GB)<br>+ CC-News (76 GB).<br>+ OpenWebText/Reddit upvoted (38 GB).<br>+ Stories, 1M story documents from the CC (31 GB).<br>+ English CC100 (292 GB). | 453 GB |
| GPT-3 | Open-AI | 175B | Wikipedia<br>Books and Journals<br>Common Crawl (filtered)<br>WebText2 and Others. | 45 TB |
| OPT-175 | Meta AI | 175B | BookCorpus and CC-Stories<br>ThePile:<br>+ Pile-CC + USPTO<br>+ OpenWebText2 + Project Gutenberg<br>+ OpenSubtitles + Wikipedia<br>+ DMMathematics<br>+ HackerNews– Pushshift.io<br>Reddit dataset + CCNewsV2<br>CommonCrawl News dataset | 800 GB |

At the end of this section, we presented a full list of the current state-of-the-art of key pre-trained language models. Table 3 provides a summary of the PLMs, architectures and training parameters for each.

**Table 3.** Summary of recent PLMs.

| Year | Resource (PLMs) | Team | Architecture | Parameters |
|---|---|---|---|---|
| Mar-2018 | ELMO | AI2 | Bi-directional LM | 94M |
| Jun-2018 | GPT | OpenAI | Transformer Dec. | 117M |
| Oct-2018 | BERT-base | Google | Transformer Enc. | 110M |
| Oct-2018 | BERT-large | Google | Transformer Enc. | 340M |
| Jan-2019 | Transformer-XL | Google AI | Transformer | 257M |
| Jan-2019 | XLM | Facebook AI | Transformer | 570M |
| Feb-2019 | GPT-2-large | OpenAI | Transformer Dec. | 774M |
| Feb-2019 | GPT-2-medium | OpenAI | Transformer Dec. | 345M |
| Feb-2019 | GPT-2-small | OpenAI | Transformer Dec. | 124M |
| Feb-2019 | GPT-2-XL | OpenAI | Transformer Dec. | 1.5B |
| May-2019 | UNILM | Microsoft | Transformer Enc. | 340M |
| May-2019 | MASS | Microsoft Research Asia | Transformer | 120M |
| Jun-2019 | XLNET | Google Brain and CMU | Transformer Enc. | 340M |
| Jul-2019 | ERNIE 2.0 | Baidu | Transformer Enc. | 114M |
| Jul-2019 | RoBERTa (base) | Facebook AI | Transformer Enc. | 109M |
| Jul-2019 | RoBERTa (large) | Facebook AI | Transformer Enc. | 355M |
| Sep-2019 | ALBERT-B | Google and Toyota | Transformer Enc. | 12M |
| Sep-2019 | ALBERT-L | Google and Toyota | Transformer Enc. | 18M |
| Sep-2019 | CTRL | SalesForce | Transformer | 1.63B |
| Sep-2019 | Megatron-LM | Nvidia | Seq2Seq | 8.3B |
| Sep-2019 | TinyBERT | Huawei | Transformer Enc. | 14.5M |
| Oct-2019 | BART | Facebook | Transformer | 460M |
| Oct-2019 | DistilBERT | HuggingFace | Transformer Enc. | 66M |
| Oct-2019 | DistilGPT2 | HuggingFace | Transformer Dec. | 82M |
| Oct-2019 | T5 | Google | Transformer | 11B |
| Nov-2019 | XLM-R (base) | Facebook AI | Transformer | 270M |
| Nov-2019 | XLM-R (large) | Facebook AI | Transformer | 550M |
| Jan-2020 | Reformer | Google research | Transformer | 149M |
| Fev-2020 | MT-DNN | Microsoft | Transformer | 330M |
| Feb-2020 | T-NLG | Microsoft | Transformer | 17.2B |
| Mar-2020 | ELECTRA | Google Brain | Transformer Enc. | 335M |
| Apr-2020 | Longformer (base) | AllenAI | Transformer Enc. | 149M |
| Apr-2020 | Longformer (large) | AllenAI | Transformer Enc. | 435M |
| May-2020 | GPT-3 | OpenAI | Transformer Dec. | 175B |
| Jun-2020 | DeBERTa (base) | Microsoft | Transformer Dec. | 140M |
| Jun-2020 | DeBERTa (large) | Microsoft | Transformer Dec. | 400M |
| Jun-2020 | DeBERTa (xlarge) | Microsoft | Transformer Dec. | 750M |
| Jun-2020 | DeBERTa (xlarge-v2) | Microsoft | Transformer Dec. | 900M |
| Jun-2020 | DeBERTa (xxlarge-v2) | Microsoft | Transformer Dec. | 105B |
| Jun-2020 | MARGE | Facebook AI | Seq2Seq | 960M |
| Jun-2020 | GShared | Google | Transformer | 600B |
| Mar-2021 | GPT-NEO | EleutherAI | Transformer Dec. | 2.7B |
| Jun-2021 | XLM-E | Microsoft | Transformer | 279M |
| Oct-2021 | MT-NLG | Nvidia and Microsoft | Transformer | 530B |
| Oct-2021 | T0 | Researchers | Transformer | 11B |
| Dec-2021 | GLaM | Google AI | Transformer | 1.2 Trillion |
| Dec-2021 | Gopher | Google AI | Transformer | 280B |
| Dec-2021 | XGLM | Meta AI | Transformer | 7.5B |
| Jan-2022 | LaMDA | Google | Transformer Dec. | 137B |
| Feb-2022 | GPT-NeoX-20B | EleutherAI and CoreWeave | Transformer Dec. | 20B |
| Mar-2022 | Chinchilla | DeepMind | Transformer | 70B |
| Apr-2022 | PaLM | Google AI | Transformer | 540B |
| May-2022 | OPT | Meta AI | Transformer Dec | 175B |

## 4. PLMs Applications, Parameters, Objectives, and Compression Methods

### *4.1. PLMs Applications*

With the recent advances in deep learning and the large number of pre-trained language models, a wide range of NLP tasks can be solved efficiently. PLMs can be taken advantage of by fine-tuning them for the task, prompting them to execute the intended task, or reshaping the task as a text generation issue and applying PLMs to solve it appropriately.

- Fine-tuning: involves performing general-purpose pre-training with a large unlabeled corpus, then adding an extra layer(s) for the specific task and further training the model using a task-specific annotated dataset, starting from the pretrained backbone weights. PLMs are now being used to solve a wide range of NLP tasks (e.g., sentiment analysis, textual entailment, question answering, common sense reasoning, translation, summarization, named entity recognition [8,24–26], stance detection [76], semantic keyphrase extraction [77], etc.);
- Prompt-based learning: reducing an NLP challenge to a task comparable to the PLM's pre-training objective (e.g., word prediction, textual entailment, classification, etc.). Few-shot/one-shot/zero-shot approaches can be achieved with prompting since they can better utilize the knowledge stored in PLMs [78,79];
- Text generation: reducing an NLP challenge to a text generation task in order to exploit knowledge encoded in generative language models such as GPT-3 [31], T5 [46], and GPT-Neox-20B [11].

### *4.2. PLMs Parameters*

PLMs have advanced rapidly in recent years due to the availability of large-scale computers and datasets. At the same time, recent research has demonstrated that large language models are excellent few-shot learners, achieving high accuracy on a wide range of NLP datasets. As a result, the number of cutting-edge NLP models has increased at an exponential rate (e.g., T5: 11B parameters, GPT-3: 175B parameters, LaMDA: 137B parameters, MT-NLG: 530B parameters, GShard: 600B parameters, etc.). Figure 3 shows some of the most popular large pre-trained language models.

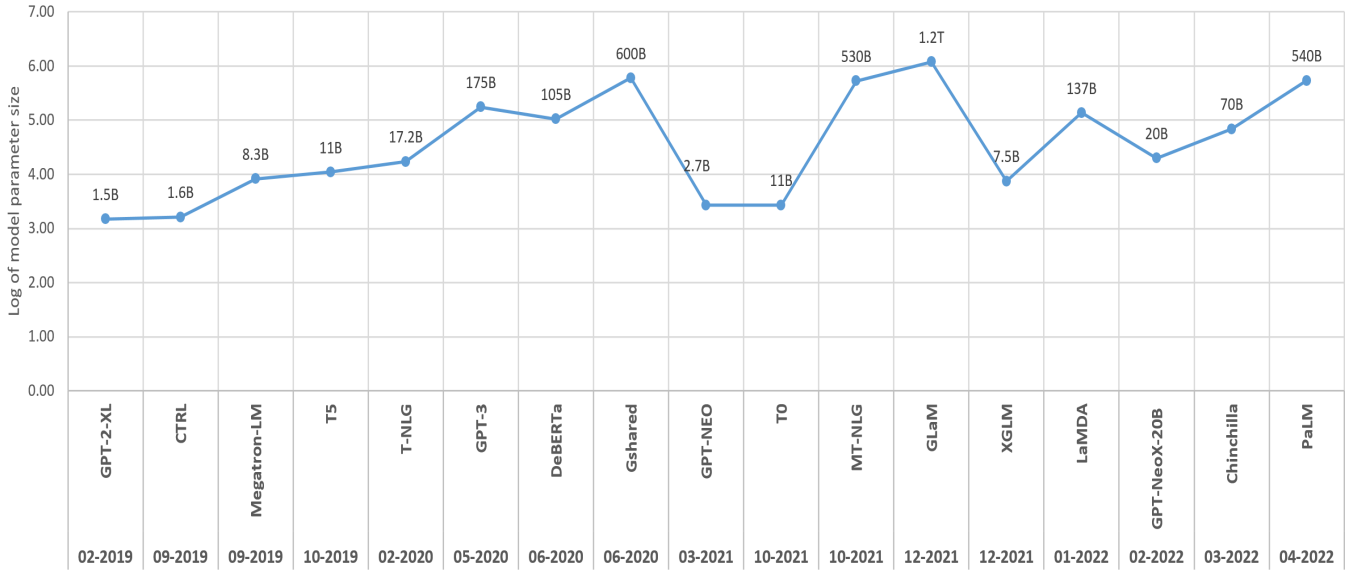


**Figure 3.** PLMs vs trained parameters (log parameter).

### *4.3. Compression Methods*

State-of-the-art PLMs are often made up of millions or even billions of parameters. This requires a substantial amount of memory and compute (GPUs and TPUs infrastructure). The topic of model compression emerges as a result of the obvious need to minimize

the size and complexity of neural networks for deployment. Variant strategies have been used to compress models and determine the ideal mix of model sizes versus accuracy. Pruning [80] (weights, neurons, heads and layers, blocks), quantization, knowledge distillation [81–84] (DistilBERT, TinyBERT), parameter sharing (ALBERT), matrix (tensor) decomposition, weight squeezing [85], and dynamic inference acceleration are among the emerged techniques. For further information, please see details here [86,87].

## 5. Challenges, and Future Directions

Though PLMs have proven their power for various NLP tasks, challenges still exist due to complexity of languages and the need of computation. In this section, we discuss some challenges and expose future research directions.

### *5.1. More Data, More Parameters*

Currently, PTMs have not yet reached their upper bounds. The introduction of the GPT-3, or Megatron-Turing NLG, has proved that the more data a model is trained on, the better the representations are at transferring to other NLP problems. We think that in the future, big companies such as Nvidia, Google, and Open-AI will keep working on the huge model as a research goal, but that it will be out of reach for most people.

### *5.2. Others Compression Alternatives*

Model compression (in size and/or latency) has been widely adopted to obtain light-weighted models without significantly diminished accuracy. There are a number of well-known methods, including pruning, quantization, low-rank approximation, knowledge distillation, and neural architecture search (NAS). This line of research should be looked into more deeply to come up with powerful ways to make light-weight models from heavier ones.

### *5.3. New Architectures of PLMs*

For pre-training purposes, the transformer has been shown to be a highly successful architectural solution in capturing contextualised representations and encode relations. The primary restriction on the transformer, on the other hand, is the complexity of its computations.

Most transformer models are inefficient at processing long sequences. The limit is derived from the transformer architecture's positional embeddings, for which a maximum length must be set (The magnitude of such a size is related to the amount of memory needed to handle texts; most PLMs are trained to handle sequences up to 512 tokens). However, the model is requested to analyze lengthier sequences for a variety of NLP tasks. The model will only process 512 tokens at a time, truncating anything longer to be processed later or in parallel (if memory allows. A Titan RTX with 24 GB of GPU RAM, for example, can barely fit 24 samples of 512 tokens in length at the same time). This issue arises due to the computation and memory complexity of the self-attention module: attention layers scale quadratically with the sequence length, posing a problem with long texts. As a result, the use of more GPUs and TPUs is compulsory in order to have enough memory to speed-up training (Table 4). Large PLMs, such as GPT-3, T5, Megatron-LM, and Turing-NLG, for example, require roughly 10,000 GPUs for training and may potentially cost tens of millions of dollars (to my knowledge, exact numbers are not available).

To overcome this restriction, the transformer's architecture must be improved in upcoming PLMs. Moreover, it is critical to investigate more efficient non-transformer architectures for PLMs.

**Table 4.** Training parameters vs Training duration.

| Model Comparison | Number of Parameters | Hardware | Training Time |
|---|---|---|---|
| ELMo | 94M | P100 × 3 | 14 days |
| BERT-Base | 110M | 8 × V100 | 12 days |
| BERT-Large | 340M | 64 TPU Chips | 4 days |
| RoBERTa-Large | 340M | 1024 × V100 | 1 day |
| DistlBERT-Base | 66M | 8 × V100 | 3.5 days |
| XLNET-Large | 340M | 512 TPU Chips | 2.5 days |
| GPT-2-Large | 570M | TPUv3 × 32 | 7 days |

### *5.4. Responsible Compute and GreenAI*

In the field of PLMs, significant progress has been made. With the increased availability of large-scale datasets (Wikipedia, the Pile, Common Crawl, etc.) and computational capabilities (GPUs, TPUs), the energy consumed by large PLMs (GPT-3, Megatron-LM, T5, GlaM, LaMDA, PaLM, etc.) is becoming a growing concern [88]. An empirical study [89] has estimated the carbon footprint of several NLP models and concluded that this trend is both environmentally unfriendly and prohibitively expensive [90]. To summarize, it is critical to use data-centric techniques to make AI greener, more inclusive, and to democratize Green AI [88–90].

## 6. Conclusions

This paper presents an overview of the recent advances achieved in pre-trained language models for NLP. For the most part, we compiled a thorough list of current PLMs, including their architectures, number of parameters, training data, training objectives, compression methods, application strategies, etc. Additionally, we reviewed several challenges and suggested several possible future research directions for PLMs.

**Funding:** This research was funded by the Deanship of Scientific Research at Umm Al-Qura University grant number 22UQU4350491DSR01.

**Acknowledgments:** MM would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by grant code: (22UQU4350491DSR01). We also would like to thank anonymous reviewers for their constructive feedback on the initial manuscript.

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

## References

1. Harris, Z. Distributional structure. *Word* **1954**, *10*, 146–162. https://doi.org/10.1007/978-94-009-8467-7_1.
2. Bengio, Y.; Ducharme, R.; Vincent, P.; Janvin, C. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
3. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; Curran Associates, Inc.: Red Hook, NY, USA, 2013.
4. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014), Doha, Qatar, 25–29 October 2014.
5. Levy, O.; Goldberg, Y. Neural Word Embedding as Implicit Matrix Factorization. In *Advances in Neural Information Processing Systems 27, Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; MIT Press: Cambridge, MA, USA, 2014; pp. 2177–2185.
6. Liu, Q.; Huang, H.; Gao, Y.; Wei, X.; Tian, Y.; Liu, L. Task-oriented Word Embedding for Text Classification. In Proceedings of the 27th International Conference on Computational Linguistics (COLING), Santa Fe, NM, USA, 20–26 August 2018.
7. Doan, T.M. Learning Word Embeddings. Ph.D. Thesis, University Jean Monnet, Saint-Etienne, France, 2018.
8. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
9. Liu, X.; He, P.; Chen, W.; Gao, J. Multi-Task Deep Neural Networks for Natural Language Understanding. *arXiv* **2019**, arXiv:1901.11504.
10. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.

11. Black, S.; Biderman, S.; Hallahan, E.; Anthony, Q.; Gao, L.; Golding, L.; He, H.; Leahy, C.; McDonell, K.; Phang, J.; et al. GPT-NeoX-20B: An Open-Source Autoregressive Language Model . *arXiv* **2022**, arXiv:2204.06745.
12. Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Deerwester, S.; Harshman, R. Using Latent Semantic Analysis to Improve Access to Textual Information. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Washington, DC, USA, 15–19 May 1988; pp. 281–285.
13. Deerwester, S.; Dumais, S.; Furnas, G.; Landauer, T.; Harshman, R. Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.* **1990**, *41*, 391–407.
14. van der Maaten, L.J.P.; Hinton, G.E. Visualizing High-Dimensional Data Using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
15. Bengio, Y.; Lauzon, V.; Ducharme, R. Experiments on the application of IOHMMs to model financial returns series. *IEEE Trans. Neural Netw.* **2001**, *12*, 113–123. https://doi.org/10.1109/72.896800.
16. Collobert, R.; Weston, J. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In Proceedings of the 25th International Conference on Machine Learning (ICML '08), Helsinki, Finland, 5–9 July 2008; ACM: New York, NY, USA, 2008; pp. 160–167. https://doi.org/10.1145/1390156.1390177.
17. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146.
18. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of Tricks for Efficient Text Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017; Short Papers; Association for Computational Linguistics: Valencia, Spain, 2017; Volume 2, pp. 427–431.
19. McCann, B.; Bradbury, J.; Xiong, C.; Socher, R. Learned in Translation: Contextualized Word Vectors. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
20. Akbik, A.; Blythe, D.; Vollgraf, R. Contextual String Embeddings for Sequence Labeling. In Proceedings of the 27th International Conference on Computational Linguistics (COLING), Santa Fe, NM, USA, 20–26 August 2018.
21. Heinzerling, B.; Strube, M. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018; European Language Resources Association (ELRA): Miyazaki, Japan, 2018.
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
23. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2018), New Orleans, LA, USA, 1–6 June 2018.
24. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. OpenAI. 2018. Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 19 August 2022).
25. Howard, J.; Ruder, S. Fine-tuned Language Models for Text Classification. *arXiv* **2018**, arXiv:1801.06146.
26. Dai, A.M.; Le, Q.V. Semi-supervised Sequence Learning. *arXiv* **2015**, arXiv:1511.01432.
27. Peters, M.; Ammar, W.; Bhagavatula, C.; Power, R. Semi-supervised sequence tagging with bidirectional language models. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), Vancouver, BC, Canada, 30 July–4 August 2017.
28. Howard, J.; Ruder, S. Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Long Papers; Association for Computational Linguistics: Melbourne, Australia, 2018; Volume 1, pp. 328–339. https://doi.org/10.18653/v1/P18-1031.
29. Zhu, Y.; Kiros, R.; Zemel, R.S.; 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.
30. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.G.; Le, Q.V.; Salakhutdinov, R. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *arXiv* **2019**, arXiv:1901.02860.
31. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. *OpenAI Blog* **2019**, *1*, 9.
32. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Chen, X.; Zhang, H.; Tian, X.; Zhu, D.; Tian, H.; Wu, H. ERNIE: Enhanced Representation through Knowledge Integration. *arXiv* **2019**, arXiv:1904.09223.
33. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.G.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv* **2019**, arXiv:1906.08237.
34. 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.
35. Lample, G.; Conneau, A. Cross-lingual Language Model Pretraining. *arXiv* **2019**, arXiv:1901.07291.
36. Keskar, N.S.; McCann, B.; Varshney, L.R.; Xiong, C.; Socher, R. CTRL: A Conditional Transformer Language Model for Controllable Generation. *arXiv* **2019**, arXiv:1909.05858.
37. Shoeybi, M.; Patwary, M.; Puri, R.; LeGresley, P.; Casper, J.; Catanzaro, B. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. *arXiv* **2019**, arXiv:1909.08053.

38. 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:1909.11942.
39. Bucilua Cristian, C.; Caruana, R.; Niculescu-Mizil, A. Model Compression. In *KDD '06, Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006*; Association for Computing Machinery: New York, NY, USA, 2006; pp. 535–541. https://doi.org/10.1145/1150402.1150464.
40. Hinton, G.E.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
41. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
42. Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; Liu, Q. TinyBERT: Distilling BERT for Natural Language Understanding. *arXiv* **2019**, arXiv:1909.10351.
43. Zhao, S.; Gupta, R.; Song, Y.; Zhou, D. Extreme language model compression with optimal subwords and shared projections. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.
44. Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; Zhou, D. Mobilebert: Task-agnostic compression of bert for resource limited devices. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 2158–2170.
45. Tsai, H.; Riesa, J.; Johnson, M.; Arivazhagan, N.; Li, X.; Archer, A. Small and practical BERT models for sequence labeling. *arXiv* **2019**, arXiv:1909.00100.
46. 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. *arXiv* **2019**, arXiv:1910.10683.
47. 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. *arXiv* **2019**, arXiv:1910.13461.
48. 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. *arXiv* **2019**, arXiv:1911.02116.
49. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P.J. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *Proc. Mach. Learn. Res.* **2019**, *119*, 11328–11339.
50. Kitaev, N.; Kaiser, L.; Levskaya, A. Reformer: The Efficient Transformer. *arXiv* **2020**, arXiv:2001.04451.
51. Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; Chang, M. REALM: Retrieval-Augmented Language Model Pre-Training. *arXiv* **2020**, arXiv:2002.08909.
52. Rosset, C. Turing-NLG: A 17-Billion-Parameter Language Model by Microsoft. Microsoft Blog. 2020. Available online: https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/ (accessed on 19 April 2022).
53. Clark, K.; Luong, M.; Le, Q.V.; Manning, C.D. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *arXiv* **2020**, arXiv:2003.10555.
54. Herzig, J.; Nowak, P.K.; Müller, T.; Piccinno, F.; Eisenschlos, J.M. TAPAS: Weakly Supervised Table Parsing via Pre-training. *arXiv* **2020**, arXiv:2004.02349.
55. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *arXiv* **2020**, arXiv:2004.05150.
56. Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T. MPNet: Masked and Permuted Pre-training for Language Understanding. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 16857–16867.
57. Lewis, P.S.H.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
58. He, P.; Liu, X.; Gao, J.; Chen, W. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *arXiv* **2020**, arXiv:2006.03654.
59. Lewis, M.; Ghazvininejad, M.; Ghosh, G.; Aghajanyan, A.; Wang, S.I.; Zettlemoyer, L. Pre-training via Paraphrasing. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 18470–18481.
60. Lepikhin, D.; Lee, H.; Xu, Y.; Chen, D.; Firat, O.; Huang, Y.; Krikun, M.; Shazeer, N.; Chen, Z. {GS}hard: Scaling Giant Models with Conditional Computation and Automatic Sharding. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
61. Xue, L.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; Raffel, C. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv* **2020**, arXiv:2010.11934.
62. Wang, X.; Gao, T.; Zhu, Z.; Liu, Z.; Li, J.; Tang, J. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 176–194.
63. Chi, Z.; Huang, S.; Dong, L.; Ma, S.; Singhal, S.; Bajaj, P.; Song, X.; Wei, F. XLM-E: Cross-lingual Language Model Pre-training via ELECTRA. *arXiv* **2021**, arXiv:2106.16138.
64. Smith, S.; Patwary, M.; Norick, B.; LeGresley, P.; Rajbhandari, S.; Casper, J.; Liu, Z.; Prabhumoye, S.; Zerveas, G.; Korthikanti, V.; et al. Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model. *arXiv* **2022**, arXiv:2201.11990.
65. Sanh, V.; Webson, A.; Raffel, C.; Bach, S.H.; Sutawika, L.; Alyafeai, Z.; Chaffin, A.; Stiegler, A.; Scao, T.L.; Raja, A.; et al. Multitask Prompted Training Enables Zero-Shot Task Generalization. *arXiv* **2021**, arXiv:2110.08207.
66. Edalati, A.; Tahaei, M.S.; Rashid, A.; Nia, V.P.; Clark, J.J.; Rezagholizadeh, M. Kronecker Decomposition for GPT Compression. *arXiv* **2021**, arXiv:2110.08152.

67. Du, N.; Huang, Y.; Dai, A.M.; Tong, S.; Lepikhin, D.; Xu, Y.; Krikun, M.; Zhou, Y.; Yu, A.W.; Firat, O.; et al. GLaM: Efficient Scaling of Language Models with Mixture-of-Experts. *arXiv* **2021**, arXiv:2112.06905.
68. Rae, J.W.; Borgeaud, S.; Cai, T.; Millican, K.; Hoffmann, J.; Song, H.F.; Aslanides, J.; Henderson, S.; Ring, R.; Young, S.; et al. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. *arXiv* **2021**, arXiv:2112.11446.
69. Lin, X.V.; Mihaylov, T.; Artetxe, M.; Wang, T.; Chen, S.; Simig, D.; Ott, M.; Goyal, N.; Bhosale, S.; Du, J.; et al. Few-shot Learning with Multilingual Language Models. *arXiv* **2021**, arXiv:2112.10668.
70. Thoppilan, R.; Freitas, D.D.; Hall, J.; Shazeer, N.; Kulshreshtha, A.; Cheng, H.; Jin, A.; Bos, T.; Baker, L.; Du, Y.; et al. LaMDA: Language Models for Dialog Applications. *arXiv* **2022**, arXiv:2201.08239.
71. Gao, L.; Biderman, S.; Black, S.; Golding, L.; Hoppe, T.; Foster, C.; Phang, J.; He, H.; Thite, A.; Nabeshima, N.; et al. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *arXiv* **2021**, arXiv:2101.00027.
72. Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; Casas, D.d.L.; Hendricks, L.A.; Welbl, J.; Clark, A.; et al. Training Compute-Optimal Large Language Models. *arXiv* **2022**, arXiv:2203.15556. https://doi.org/10.48550/ARXIV.2203.15556.
73. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. PaLM: Scaling Language Modeling with Pathways. *arXiv* **2022**, arXiv:2204.02311. https://doi.org/10.48550/ARXIV.2204.02311.
74. Barham, P.; Chowdhery, A.; Dean, J.; Ghemawat, S.; Hand, S.; Hurt, D.; Isard, M.; Lim, H.; Pang, R.; Roy, S.; et al. Pathways: Asynchronous Distributed Dataflow for ML. *Proc. Mach. Learn. Syst.* **2022**, *4*, 430–449. https://doi.org/10.48550/ARXIV.2203.12533.
75. 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. https://doi.org/10.48550/ARXIV.2205.01068.
76. Karande, H.; Walambe, R.; Benjamin, V.; Kotecha, K.; Raghu, T.S. Stance Detection with BERT Embeddings for Credibility Analysis of Information on Social Media. *PeerJ Comput. Sci.* **2021**, *7*, e467.
77. Devika, R.; Vairavasundaram, S.; Mahenthar, C.S.J.; Varadarajan, V.; Kotecha, K. A Deep Learning Model Based on BERT and Sentence Transformer for Semantic Keyphrase Extraction on Big Social Data. *IEEE Access* **2021**, *9*, 165252–165261. https://doi.org/10.1109/ACCESS.2021.3133651.
78. Sun, Y.; Zheng, Y.; Hao, C.; Qiu, H. NSP-BERT: A Prompt-based Zero-Shot Learner Through an Original Pre-training Task-Next Sentence Prediction. *arXiv* **2021**, arXiv:2109.03564.
79. Xu, H.; Chen, Y.; Du, Y.; Shao, N.; Wang, Y.; Li, H.; Yang, Z. ZeroPrompt: Scaling Prompt-Based Pretraining to 1000 Tasks Improves Zero-Shot Generalization. *arXiv* **2022**, arXiv:2201.06910.
80. Frankle, J.; Carbin, M. The Lottery Ticket Hypothesis: Training Pruned Neural Networks. *arXiv* **2018**, arXiv:1803.03635.
81. Liu, D.; Cheng, P.; Dong, Z.; He, X.; Pan, W.; Ming, Z. A General Knowledge Distillation Framework for Counterfactual Recommendation via Uniform Data. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Online, 25–30 July 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 831–840.
82. Ding, F.; Luo, F.; Hu, H.; Yang, Y. Multi-level Knowledge Distillation. *arXiv* **2020**, arXiv:2012.00573.
83. Sun, S.; Cheng, Y.; Gan, Z.; Liu, J. Patient Knowledge Distillation for BERT Model Compression. *arXiv* **2019**, arXiv:1908.09355.
84. Wang, T.; Zhu, J.; Torralba, A.; Efros, A.A. Dataset Distillation. *arXiv* **2018**, arXiv:1811.10959.
85. Chumachenko, A.; Gavrilov, D.; Balagansky, N.; Kalaidin, P. Weight squeezing: Reparameterization for extreme compression and fast inference. *arXiv* **2020**, arXiv:2010.06993.
86. 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. https://doi.org/10.1162/tacl_a_00413.
87. Gupta, M.; Agrawal, P. Compression of Deep Learning Models for Text: A Survey. *ACM Trans. Knowl. Discov. Data (TKDD)* **2020**, *16*, 1–55.
88. Schwartz, R.; Dodge, J.; Smith, N.A.; Etzioni, O. Green AI. *Commun. ACM* **2020**, *63*, 54–63.
89. Verdecchia, R.; Cruz, L.; Sallou, J.; Lin, M.; Wickenden, J.; Hotellier, E. Data-Centric Green AI: An Exploratory Empirical Study. *arXiv* **2022**, arXiv:2204.02766. https://doi.org/10.48550/ARXIV.2204.02766.
90. Strubell, E.; Ganesh, A.; McCallum, A. Energy and Policy Considerations for Deep Learning in NLP. *arXiv* **2019**, arXiv:1906.02243.