Pre-Trained Models: Past, Present and Future

Xu Han[1*], **Zhengyan Zhang**[1*], **Ning Ding**[1*], **Yuxian Gu**[1*], **Xiao Liu**[1*], **Yuqi Huo**[2*], **Jiezhong Qiu**[1], **Liang Zhang**[2], **Wentao Han**[1†], **Minlie Huang**[1†], **Qin Jin**[2†], **Yanyan Lan**[3†], **Yang Liu**[1,3†], **Zhiyuan Liu**[1†], **Zhiwu Lu**[2†], **Xipeng Qiu**[4†], **Ruihua Song**[2†], **Jie Tang**[1†], **Ji-Rong Wen**[2†], **Jinhui Yuan**[5†], **Jun Zhu**[1†], **Wayne Xin Zhao**[2†]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[3] Institute for AI Industry Research, Tsinghua University, Beijing, China
[4] School of Computer Science, Fudan University, Shanghai, China
[5] OneFlow Inc., Beijing, China

{hanxu17,zy-z19,dingn18,gu-yx17,liuxiao17}@mails.tsinghua.edu.cn,
{bnhony,batmanfly}@ruc.edu.cn

Abstract

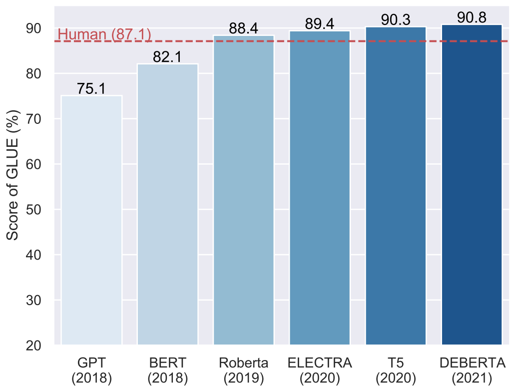
Large-scale pre-trained models (PTMs) such as BERT and GPT have recently achieved great success and become a milestone in the field of artificial intelligence (AI). Owing to sophisticated pre-training objectives and huge model parameters, large-scale PTMs can effectively capture knowledge from massive labeled and unlabeled data. By storing knowledge into huge parameters and fine-tuning on specific tasks, the rich knowledge implicitly encoded in huge parameters can benefit a variety of downstream tasks, which has been extensively demonstrated via experimental verification and empirical analysis. It is now the consensus of the AI community to adopt PTMs as backbone for downstream tasks rather than learning models from scratch. In this paper, we take a deep look into the history of pre-training, especially its special relation with transfer learning and self-supervised learning, to reveal the crucial position of PTMs in the AI development spectrum. Further, we comprehensively review the latest breakthroughs of PTMs. These breakthroughs are driven by the surge of computational power and the increasing availability of data, towards four important directions: designing effective architectures, utilizing rich contexts, improving computational efficiency, and conducting interpretation and theoretical analysis. Finally, we discuss a series of open problems and research directions of PTMs, and hope our view can inspire and advance the future study of PTMs.

1 Introduction

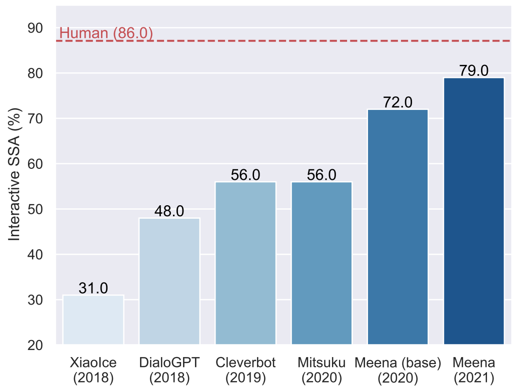
Deep neural networks, such as convolutional neural networks (CNNs) (Krizhevsky et al., 2012; Kim, 2014; Kalchbrenner et al., 2014; He et al., 2016), recurrent neural networks (RNNs) (Sutskever et al., 2014; Donahue et al., 2015; Liu et al., 2016; Wu et al., 2016), graph neural networks (GNNs) (Kipf and Welling, 2016; Veličković et al., 2018; Schlichtkrull et al., 2018), and attention neural networks (Jaderberg et al., 2015; Wang et al., 2017), have been widely applied for various artificial intelligence (AI) tasks in recent years. Different from previous non-neural models that largely relied on hand-crafted features and statistical methods, neural models can automatically learn low-dimensional continuous vectors (*a.k.a.,* distributed representations) from data as task-specific features, thereby getting rid of complex feature engineering. Despite the success of deep neural networks, a number of studies have found that one of their critical challenges is data hungry. Since deep neural networks usually have a large number of parameters, they are thus easy to overfit and have poor generalization ability (Belkin et al., 2019; Xu et al., 2021) without sufficient training data.

Considering this issue, over the same period of developing deep neural networks, massive efforts have been devoted to manually constructing high-quality datasets for AI tasks (Deng et al., 2009; Lin et al., 2014; Bojar et al., 2014), making it possible to learn effective neural models for specific tasks that are superior to conventional non-neural models. However, it is expensive and time-consuming to manually annotate large-scale data. For example, utilizing crowdsourcing to segment images costs about \$6.4 per image (Liu et al., 2020b). Some complex tasks that require expert annotations may charge much more to build their datasets. Several tasks such as visual recognition (Deng et al., 2009) and machine translation (Bojar et al., 2014) have

[*] The first six authors contribute equally to organize this paper. The order is determined by dice rolling.
[†] All faculty authors are alphabetically sorted.



(a) The results of various NLP PTMs on the language understanding benchmark GLUE.



(b) The results of the manual evaluation for various dialogue systems using NLP PTMs.

Figure 1: Figure 1(a) and Figure 1(b) show the performance improvements on language understanding and language generation respectively after using large-scale PTMs. From these figures, we can find that the methods based on PTMs achieve significant improvements on both language understanding and language generation.

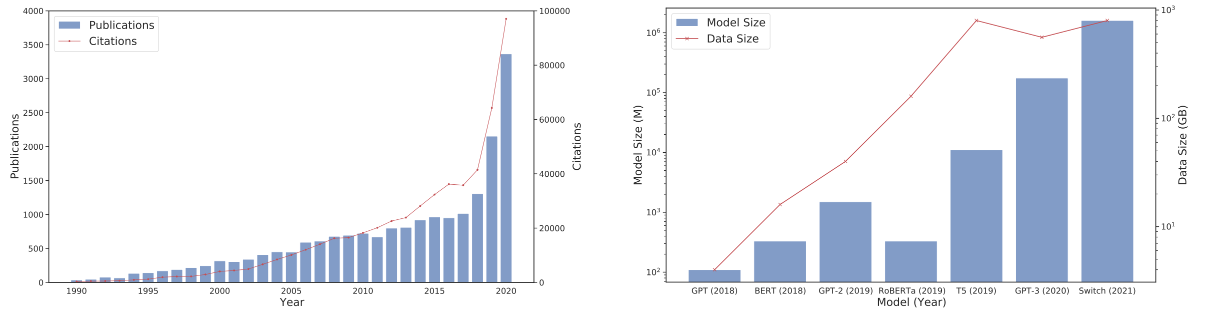
datasets containing millions of samples, yet it is impossible to build such large-scale datasets for all AI tasks. More generally, the dataset of a specific AI task usually has a limited size. Hence, for a long time until now, it has been a key research issue: *how to train effective deep neural models for specific tasks with limited human-annotated data*.

One milestone for this issue is the introduction of transfer learning (Thrun and Pratt, 1998; Pan and Yang, 2009). Instead of training a model from scratch with large amounts of data, human beings can learn to solve new problems with very few samples. This amazing learning process is motivated by the fact that human beings can use previously learned knowledge to handle new problems. Inspired by this, transfer learning formalizes a two-phase learning framework: a pre-training phase to capture knowledge from one or more source tasks, and a fine-tuning stage to transfer the captured knowledge to target tasks. Owing to the wealth of knowledge obtained in the pre-training phase, the fine-tuning phase can enable models to well handle target tasks with limited samples.

Transfer learning provides a feasible method for alleviating the challenge of data hungry, and it has soon been widely applied to the field of computer vision (CV). A series of CNNs (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; Szegedy et al., 2015; He et al., 2016) are pre-trained on the human-annotated visual recognition dataset ImageNet (Deng et al., 2009). Benefiting from the strong visual knowledge distributed in ImageNet, fine-tuning these pre-trained CNNs with a small amount of task-specific data can perform well on downstream tasks. This triggers the first wave of exploring pre-trained models (PTMs) in the era of deep learning. In this wave, PTMs are used for almost all CV tasks such as image classification (He et al., 2016), object detection (Sermanet et al., 2013; Ren et al., 2016), image segmentation (Long et al., 2015), and image captioning (Vinyals et al., 2015).

The natural language processing (NLP) community was also aware of the potential of PTMs and started to develop PTMs for NLP tasks (Qiu et al., 2020). To take full advantage of large-scale unlabeled corpora to provide versatile linguistic knowledge for NLP tasks, the NLP community adopts self-supervised learning (Liu et al., 2020b) to develop PTMs. The motivation of self-supervised learning is to leverage intrinsic correlations in the text as supervision signals instead of human supervision. For example, given the sentence “Beijing is the capital of China”, we mask the last word in the sentence, and then require models to predict the masked position with the word “China”. Through self-supervised learning, tremendous amounts of unlabeled textual data can be utilized to capture versatile linguistic knowledge without labor-intensive workload. This self-supervised setting in essence follows the well-known language model learning (Bengio et al., 2003).

For a long time, the problem of vanishing or exploding gradients (Bengio et al., 1994) is the pain point of using deep neural networks for NLP tasks. Therefore, when the CV community advances the



(a) The number of publications about "language models" and their citations in the recent years.

(b) The model size and data size applied by NLP PTMs in recent years. A base-10 log scale is used for the figure.

Figure 2: Figure 2(a) shows the number of publications with the keyword "language model" as well as their citations in different years. Figure 2(b) shows the parameter size of large-scale PTMs for NLP tasks and the pre-training data size are increasing by 10 times per year. From these figures, we can find that, after 2018, when large-scale NLP PTMs begin to be explored, more and more efforts are devoted to this field, and the model size and data size used by the PTMs are also getting larger.

research of deep PTMs, the early exploration of the NLP community focuses on pre-training shallow networks to capture semantic meanings of words, like Word2Vec (Mikolov et al., 2013b,a,c) and GloVe (Pennington et al., 2014). Although these pre-trained word embeddings play an important role in various NLP tasks, they still face a major limitation to represent polysemous words in different contexts, as each word is represented by only one dense vector. A famous example in NLP is that the word "bank" has entirely different meanings in the sentences "open a bank account" and "on a bank of the river". This motivates pre-training RNNs to provide contextualized word embeddings (Melamud et al., 2016; Peters et al., 2018; Howard and Ruder, 2018), yet the performance of these models is still limited by their model size and depth.

With the development of deep neural networks in the NLP community, the introduction of Transformers (Vaswani et al., 2017) makes it feasible to train very deep neural models for NLP tasks. With Transformers as architectures and language model learning as objectives, deep PTMs GPT (Radford and Narasimhan, 2018) and BERT (Devlin et al., 2019) are proposed for NLP tasks in 2018. From GPT and BERT, we can find that when the size of PTMs becomes larger, large-scale PTMs with hundreds of millions of parameters can capture polysemous disambiguation, lexical and syntactic structures, as well as factual knowledge from the text. By fine-tuning large-scale PTMs with quite a few samples, rich linguistic knowledge of PTMs brings awesome performance on downstream NLP tasks. As shown in Figure 1(a) and Figure 1(b), large-scale PTMs well perform on both language understanding and language generation tasks in the past several years and even achieve better results than human performance. As shown in Figure 2(a), all these efforts and achievements in the NLP community let large-scale PTMs become the focus of AI research, after the last wave that PTMs allow for huge advances in the CV community.

Up to now, various efforts have been devoted to exploring large-scale PTMs, either for NLP (Radford et al., 2019; Liu et al., 2020d; Raffel et al., 2020; Lewis et al., 2020a), or for CV (Lu et al., 2019; Li et al., 2019; Tan and Bansal, 2019). Fine-tuning large-scale PTMs for specific AI tasks instead of learning models from scratch has also become a consensus (Qiu et al., 2020). As shown in Figure 2(b), with the increasing computational power boosted by the wide use of distributed computing devices and strategies, we can further advance the parameter scale of PTMs from million-level to billion-level (Brown et al., 2020; Lepikhin et al., 2020) and even trillion-level (Fedus et al., 2021). And the emergence of GPT-3 (Brown et al., 2020), which has hundreds of billions of parameters, enables us to take a glimpse of the latent power distributed in massive model parameters, especially the great abilities of few-shot learning like human beings (shown in Figure 3).

The existing large-scale PTMs have improved the model performance on various AI tasks and even subverted our current perception of the performance of deep learning models. However, several fundamental issues about PTMs still remain: it is still not clear for us the nature hidden in huge

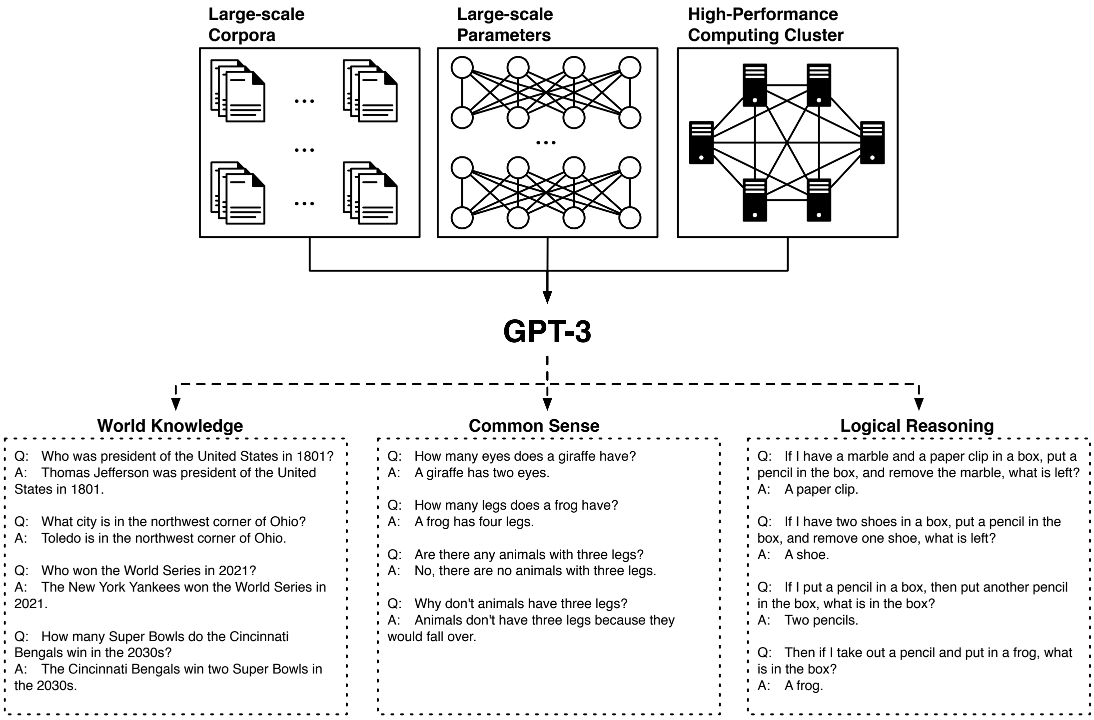


Figure 3: GPT-3, with 175 billion parameters, uses 560 GB data and 10,000 GPUs for its training. It has shown the abilities of learning world knowledge, common sense, and logical reasoning.

amounts of model parameters, and huge computational cost of training these behemoths also prevents us from further exploration. At this moment, these PTMs have pushed our AI researchers to a crossroad, with a number of open directions to go.

"*Rome wasn't built in a day*"— PTMs also experience a long development before achieving the latest success. To this end, we try to trace the development history of PTMs and draw their positions in the AI spectrum, which can give us a clear understanding of the core research issues of PTMs. Then, we introduce the details of various latest PTMs, following four important lines that are currently being advanced, including designing effective architectures, utilizing rich contexts, improving computational efficiency, and conducting interpretation and theoretical analysis. By integrating the current development of PTMs into the context of the historical spectrum, we discuss several open problems and conclude promising future directions for PTMs. We hope our efforts in this paper can advance further development of PTMs. In what follows, we will introduce the background of pre-training in Section 2 and Section 3, the model architectures of PTMs in Section 4, using multi-source heterogeneous data for PTMs in Section 5, the computational efficiency optimization of PTMs in Section 6, and the theoretical analysis of PTMs in Section 7. Finally, we will briefly discuss a series of open problems and promising directions towards better PTMs in the future.

2 Background

Although effective PTMs have recently gained the attention of researchers, pre-training is not a novel machine learning tool. In fact, pre-training has been developed for decades, as a typical machine learning paradigm. In this section, we introduce the development of pre-training in the AI spectrum, from early supervised pre-training to current self-supervised pre-training, which can lead to a brief understanding of the background of PTMs.

2.1 Transfer Learning and Supervised Pre-training

The early efforts of pre-training are mainly involved in transfer learning (Thrun and Pratt, 1998). The study of transfer learning is heavily motivated by the fact that people can rely on previously learned knowledge to solve new problems and even achieve better results. More formally, transfer learning aims to capture important knowledge from multiple source tasks and then apply the knowledge to a target task.

In transfer learning, source tasks and target tasks may have completely different data domains and

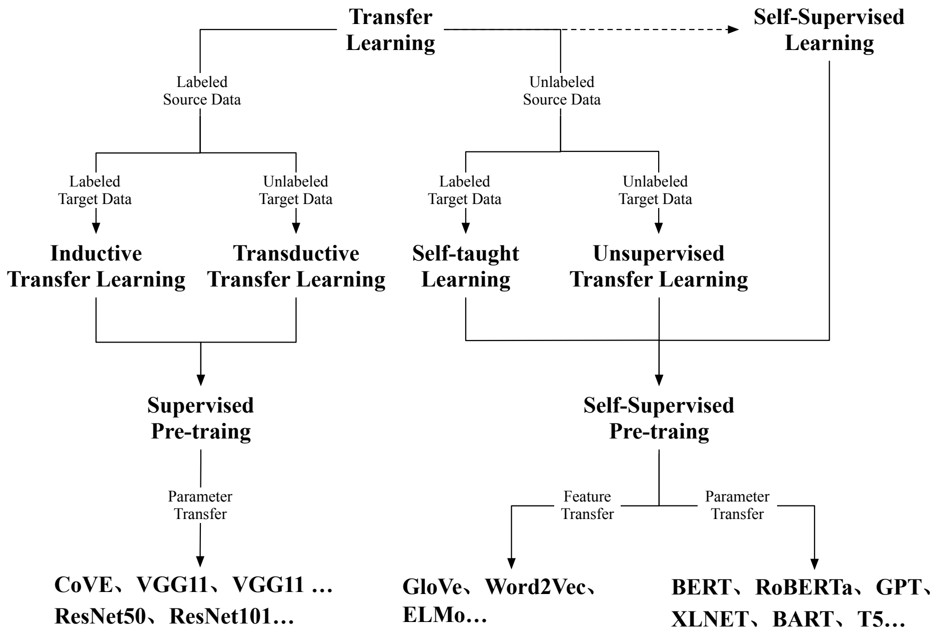


Figure 4: The spectrum of pre-training methods from transfer learning, self-supervised learning to the latest pre-training neural models.

task settings, yet the knowledge required to handle these tasks is consistent (Pan and Yang, 2009). It is thus important to select a feasible method to transfer knowledge from source tasks to target tasks. To this end, various pre-training methods have been proposed to work as the bridge between source and target tasks. Specifically, these methods first pre-train models on the data of multiple source tasks to pre-encode knowledge and then transfer the pre-encoded knowledge to train models for target tasks.

Generally, two pre-training approaches are widely explored in transfer learning: feature transfer and parameter transfer. Feature transfer methods pre-train effective feature representations to pre-encode knowledge across domains and tasks (Johnson and Zhang, 2005; Evgeniou and Pontil, 2007; Dai et al., 2007; Raina et al., 2007). By injecting these pre-trained representations into target tasks, model performance of target tasks can be significantly improved. Parameter transfer methods follow an intuitive assumption that source tasks and target tasks can share model parameters or prior distributions of hyper-parameters. Therefore, these methods pre-encode knowledge into shared model parameters (Lawrence and Platt, 2004; Evgeniou and Pontil, 2004; Williams et al., 2007; Gao et al., 2008), and then transfer the knowledge by fine-tuning pre-trained parameters with the data of target tasks.

To some extent, both representation transfer and parameter transfer lay the foundation of PTMs. Word embeddings, widely used as the input of NLP tasks, are built on the framework of feature transfer. Inspired by parameter transfer, pre-trained CNNs are applied as the backbone of most state-of-the-art CV models. Some recent well-known PTMs are also based on representation transfer and parameter transfer, e.g., ELMo (Peters et al., 2018) and BERT apply representation transfer and parameter transfer respectively.

Since AlexNet (Krizhevsky et al., 2012), a series of deep neural networks have been developed for AI tasks. As compared with those conventional machine learning models, deep neural models have more parameters and show better capabilities of fitting complex data. Therefore, from AlexNet to later VGG (Simonyan and Zisserman, 2015) and GoogleNet (Szegedy et al., 2015), the architecture of these neural networks becomes deeper and deeper, and their performance accordingly becomes better and better. Although the network depth is important, training a deep network is not easy, as stacking more network layers inevitably brings the problem of vanishing or exploding gradients (Bengio et al., 1994). Besides the gradient issues, model performance may soon meet a ceiling and then degrade rapidly with continually increasing network depths.

By adding normalization to parameter initialization (LeCun et al., 2012; Saxe et al., 2013) and hidden states (Ioffe and Szegedy, 2015), and introducing shortcut connections with residual layers, ResNet (He et al., 2016) effectively tackles these problems. As we mentioned before, deep neural networks require large amounts of data for training. To provide sufficient data to train deep models, some large-scale supervised datasets have also been built (Russakovsky et al., 2015; Lin et al., 2014; Krishna et al., 2017; Chen et al., 2015; Cordts et al., 2016), and the most representative one is ImageNet. ImageNet contains millions of images divided into thousands of categories, representing a wide variety of everyday objects. Based on the combination of effective model ResNet, informative dataset ImageNet, as well as mature knowledge transfer methods, a wave of pre-training models on labeled data emerges.

The CV community benefits a lot from this wave. By applying ResNet pre-trained on ImageNet as the backbone, various CV tasks have been quickly advanced, like image classification (He et al., 2016; Lee et al., 2015), object detection (Ren et al., 2016; Sermanet et al., 2013; Gidaris and Komodakis, 2015), image segmentation (Long et al., 2015; Zheng et al., 2015), image caption (Vinyals et al., 2015; Johnson et al., 2016), visual question answering (Antol et al., 2015; Gao et al., 2015; Xiong et al., 2016), etc. Utilizing PTMs like ResNet50 [1] has proven to be a crucial step to obtain highly accurate results on most CV tasks. Inspired by the success of PTMs for CV tasks, some NLP researchers also explore supervised Pre-training, and the most representative work is CoVE (McCann et al., 2017). CoVE adopts machine translation as its pre-training objective. After pre-training, the encoder of source languages can work as a powerful backbone for downstream NLP tasks.

2.2 Self-Supervised Learning and Self-Supervised Pre-training

As shown in Figure 4, transfer learning can be categorized under four sub-settings, inductive transfer learning (Lawrence and Platt, 2004; Mihalkova et al., 2007; Evgeniou and Pontil, 2007), transductive transfer learning (Shimodaira, 2000; Zadrozny, 2004; Daume III and Marcu, 2006), self-taught learning (Raina et al., 2007; Dai et al., 2008) [2], and unsupervised transfer learning (Wang et al., 2008).

Among these four settings, the inductive and transductive settings are the core of research, as these two settings aim to transfer knowledge from supervised source tasks to target tasks. Although supervised learning is always one of the core issues of machine learning research, the scale of unlabeled data is much larger than that of manually labeled data. Recently, more and more researchers have noticed the importance of large-scale unlabeled data and are committed to extracting information from unlabeled data. Self-supervised learning has been proposed to extract knowledge from large-scale unlabeled data by leveraging input data itself as supervision.

Self-supervised learning and unsupervised learning have many similarities in their settings. To a certain extent, self-supervised learning can be regarded as a branch of unsupervised learning because they both apply unlabeled data. However, unsupervised learning mainly focuses on detecting data patterns (e.g., clustering, community discovery, and anomaly detection), while self-supervised learning is still in the paradigm of supervised settings (e.g., classification and generation) (Liu et al., 2020b).

The development of self-supervised learning makes it possible to perform pre-training on large-scale unsupervised data. Compared to supervised pre-training working as the cornerstone of CV in the deep learning era, self-supervised pre-training allows for huge advances in the field of NLP. Although some supervised pre-training methods like CoVE have achieved promising results on NLP tasks, it is nearly impossible to annotate a textual dataset as large as ImageNet, considering annotating textual data is far more complex than annotating images. Hence, applying self-supervised learning to utilize unlabeled data becomes the best choice to pre-train models for NLP tasks. The recent stunning breakthroughs in PTMs are mainly towards NLP tasks, more specifically pre-trained language models.

The early PTMs for NLP tasks exist in the form of well-known word embeddings (Collobert and Weston, 2008; Mikolov et al., 2013b; Pennington et al., 2014), which apply self-supervised methods to transform words into distributed representations. As these pre-trained word representations capture syntactic and semantic information in the text, they are often used as input embeddings and initialization parameters for NLP models and offer significant improvements over random initialization pa-

[1] ResNet50 is a PTM with 50 layers.

[2] Self-study learning can be viewed as a variant of inductive transfer learning without available labeled data

rameters (Turian et al., 2010). Since these word-level models often suffer from the word polysemy, Peters et al. (2018) further adopt a sequence-level neural model to capture complex word features across different linguistic contexts and generates context-aware word embeddings. Using word embeddings as the input of neural models has almost become the common mode for NLP tasks.

After Vaswani et al. (2017) propose Transformers to deal with sequential data, PTMs for NLP tasks have entered a new stage, because it is possible to train deeper language models compared to conventional CNNs and RNNs. Different from those word-level PTMs used as input features, the Transformer-based PTMs such as GPT and BERT can be used as the model backbone of various specific tasks. After pre-training these Transformer-based PTMs on large-scale textual corpora, both the architecture and parameters of PTMs can serve as a starting point for specific NLP tasks, i.e., just fine-tuning the parameters of PTMs for specific NLP tasks can achieve competitive performance. So far, these Transformer-based PTMs have achieved state-of-the-art results on almost all NLP tasks. Inspired by GPT and BERT, many more effective PTMs for NLP tasks have also been proposed, like XLNET (Yang et al., 2019), RoBERTa (Liu et al., 2020d), BART (Lewis et al., 2020a), and T5 (Raffel et al., 2020).

With the recent advance of PTMs for NLP tasks, applying Transformer-based PTMs as the backbone of NLP tasks has become a standard procedure. Motivated by the success of self-supervised learning and Transformers in NLP, some researchers explore self-supervised learning (Wu et al., 2018; Chen et al., 2020c; Chen and He, 2020; He et al., 2020) and Transformers (Carion et al., 2020; Liu et al., 2021c) for CV tasks. These preliminary efforts have shown that self-supervised learning and Transformers can outperform conventional supervised CNNs. Furthermore, Transformer-based multimodal PTMs (Lu et al., 2019; Li et al., 2019; Tan and Bansal, 2019) have also been proposed and shown promising results. After the last wave of supervised pre-training, self-supervised pre-training has become the focus of current AI research.

Looking back at the pre-training in the AI spectrum, it is not difficult to find that pre-training has been developed for decades, focusing on how to acquire versatile knowledge for various downstream tasks. Next, we will comprehensively introduce the

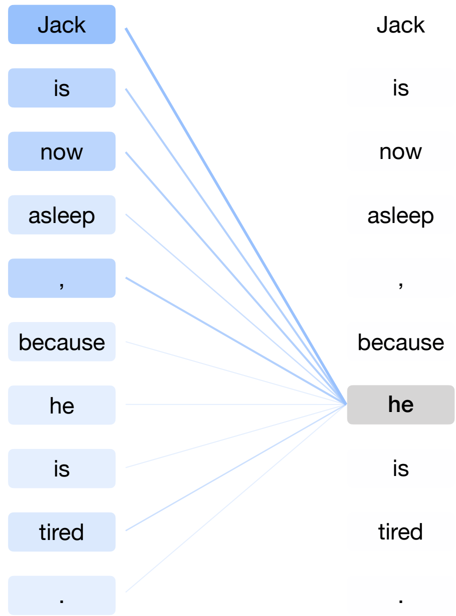


Figure 5: An illustration of the self-attention mechanism of Transformer. The figure shows the self-attention results when encoding the word "he", where the darker the color of the square is, the larger the corresponding attention score is.

latest breakthroughs of PTMs in this wave of self-supervised pre-training. Considering that almost all the latest PTMs are related to pre-trained language models, "PTMs" in the following sections refers to pre-trained language models or multimodal models. For those conventional PTMs based on supervised pre-training, we refer to the papers of He et al. (2019) and Zoph et al. (2020).

3 Transformer and Representative Pre-trained Models

As we mentioned before, the key to the success of recent PTMs is an integration of self-supervised learning and Transformers. Hence, this section begins with the dominant basic neural architecture, Transformer. Then, we will introduce two landmark Transformer-based PTMs, GPT and BERT, which respectively use autoregressive language modeling and autoencoding language modeling as the pre-training objective. All subsequent PTMs are variants of these two models. The final part of this section gives a brief review of typical variants after GPT and BERT to reveal the recent development of PTMs.

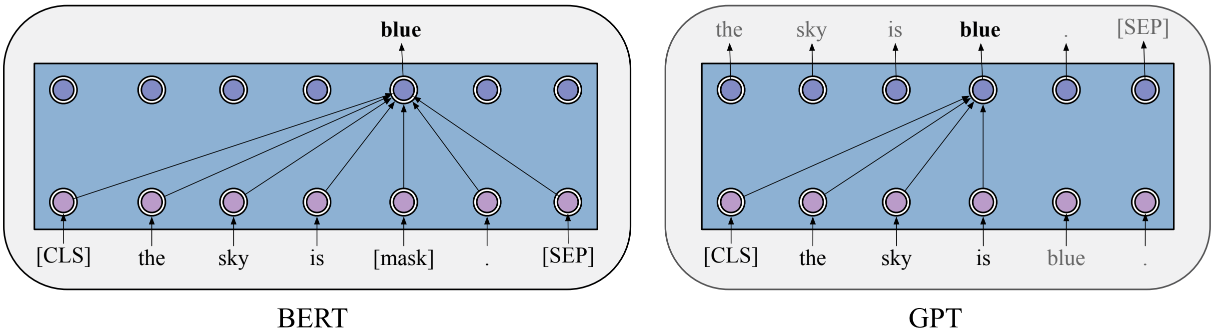


Figure 6: The difference between GPT and BERT in their self-attention mechanisms and pre-training objectives.

3.1 Transformer

Before Transformer, RNNs have been typical neural networks for processing sequential data (especially for natural languages) for a long time. As RNNs are equipped with sequential nature, they read a word at each time step in order and refer to the hidden states of the previous words to process it. Such a mechanism is considered to be difficult to take advantage of the parallel capabilities of high-performance computing devices such as GPUs and TPUs.

As compared to RNNs, Transformer is an encoder-decoder structure that applies a self-attention mechanism, which can model correlations between all words of the input sequence in parallel. Hence, owing to the parallel computation of the self-attention mechanism, Transformer could fully take advantage of advanced computing devices to train large-scale models. In both the encoding and decoding phases of Transformer, the self-attention mechanism of Transformer computes representations for all input words. Next, we dive into the self-attention mechanism more specifically.

In the encoding phase, for a given word, Transformer computes an attention score by comparing it with each other word in the input sequence. And such attention scores indicate how much each of the other words should contribute to the next representation of the given word. Then, the attention scores are utilized as weights to compute a weighted average of the representations of all the words. We give an example in Figure 5, where the self-attention mechanism accurately captures the referential relationships between "Jack" and "he", generating the highest attention score. By feeding the weighted average of all word representations into a fully connected network, we obtain the representation of the given word. Such a procedure is essentially an aggregation of the information of the whole input sequence, and it will be applied to all the words to generate representations in parallel. In the decoding phase, the attention mechanism is similar to the encoding, except that it only decodes one representation from left to right at one time. And each step of the decoding phase consults the previously decoded results. For more details of Transformer, please refer to its original paper (Vaswani et al., 2017).

Due to the prominent nature, Transformer gradually becomes a standard neural structure for natural language understanding and generation. Moreover, it also serves as the backbone neural structure for the subsequently derived PTMs. Next, we introduce two landmarks that completely open the door towards the era of large-scale self-supervised PTMs, GPT and BERT. In general, GPT is good at natural language generation, while BERT focuses more on natural language understanding.

3.2 GPT

As introduced in Section 2, PTMs typically consist of two phases, the pre-training phase and the fine-tuning phase. Equipped by the Transformer decoder as the backbone [3], GPT applies a generative pre-training and a discriminative fine-tuning. Theoretically, compared to precedents of PTMs, GPT is the first model that combines the modern Transformer architecture and the self-supervised pre-training objective. Empirically, GPT achieves significant success on almost all NLP tasks, including natural language inference, question answering, commonsense reasoning, semantic similarity and classification.

Given large-scale corpora without labels, GPT optimizes a standard autoregressive language mod-

[3]Since GPT uses autoregressive language modeling, the encoder-decoder attention in the original Transformer decoder is removed.

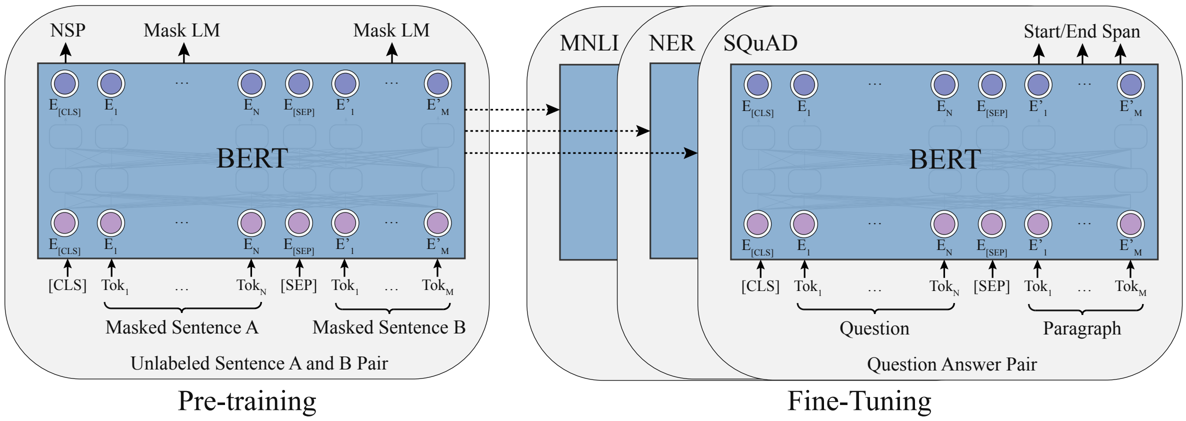


Figure 7: The pre-training and fine-tuning phases for BERT.

eling, that is, maximizing the conditional probabilities of all the words given their corresponding previous words as contexts. In the pre-training phase of GPT, the conditional probability of each word is modeled by Transformer. As shown in Figure 6, for each word, GPT computes its probability distributions by applying multi-head self-attention operations over its previous words followed by position-wise feed-forward layers.

The adaptation procedure of GPT to specific tasks is fine-tuning, by using the pre-trained parameters of GPT as a start point of downstream tasks. In the fine-tuning phase, passing the input sequence through GPT, we can obtain the representations of the final layer of the GPT Transformer. By using the representations of the final layer and task-specific labels, GPT optimizes standard objectives of downstream tasks with simple extra output layers. As GPT has hundreds of millions of parameters, it is trained for 1 month on 8 GPUs, which is fairly the first "large-scale" PTM in the history of NLP. And undoubtedly, the success of GPT pave the way for the subsequent rise of a series of large-scale PTMs. In the next part, we introduce another most representative model BERT.

3.3 BERT

The emergence of BERT has also greatly promoted the development of the PTM field. Theoretically, compared with GPT, BERT uses a bidirectional deep Transformer as the main structure. There are also two separate stages to adapt BERT for specific tasks, pre-training and fine-tuning (see Figure 7).

In the pre-training phase, BERT applies autoencoding language modeling rather than autoregressive language modeling used in GPT. More specifically, inspired by cloze (Taylor, 1953), the objective masked language modeling (MLM) is designed. As shown in Figure 6, in the procedure of MLM, tokens are randomly masked with a special token [MASK], the objective is to predict words at the masked positions with contexts. Compared with standard unidirectional autoregressive language modeling, MLM can lead to a deep bidirectional representation of all tokens.

Besides MLM, the objective of next sentence prediction (NSP) is adopted to capture discourse relationships between sentences for some downstream tasks with multiple sentences, such as natural language inference and question answering. For this task, a binary classifier is used to predict whether two sentences are coherent. In the pre-training phase, MLM and NSP work together to optimize the parameters of BERT.

After pre-training, BERT can obtain robust parameters for downstream tasks. By modifying inputs and outputs with the data of downstream tasks, BERT could be fine-tuned for any NLP tasks. BERT could effectively handle those applications with the input of a single sentence or sentence pairs. For the input, its schema is two sentences concatenated with the special token [SEP], which could represent: (1) sentence pairs in paraphrase, (2) hypothesis-premise pairs in entailment, (3) question-passage pairs in question answering, and (4) a single sentence for text classification or sequence tagging. For the output, BERT will produce a token-level representation for each token, which can be used to handle sequence tagging or question answering, and the special token [CLS] can be fed into an extra layer for classification. After GPT, BERT has further achieved significant improvements on 17 different NLP tasks, including SQuAD (better than human performance), GLUE (7.7% point absolute improvements), MNLI (4.6% point absolute improvements), etc.

3.4 After GPT and BERT

After GPT and BERT, some of their improvements have been proposed, such as RoBERTa and ALBERT. RoBERTa(Liu et al., 2020d) is one of the success variants of BERT, which mainly has four simple and effective changes: (1) Removing the NSP task; (2) More training steps, with bigger batch size and more data; (3) Longer training sentences; (4) Dynamically changing the [MASK] pattern. RoBERTa achieves impressive empirical results on the basis of BERT. Moreover, RoBERTa has pointed out that the NSP task is relatively useless for the training of BERT. ALBERT(Lan et al., 2019) is another important variant of BERT, which provides several interesting observations on reducing parameters. First, it factorizes the input word embedding matrix into two smaller ones. Second, it enforces parameter-sharing between all Transformer layers to significantly reduce parameters. Third, it proposes the sentence order prediction (SOP) task to substitute BERT's NSP task. As a sacrifice to its space efficiency, ALBERT has a slower fine-tuning and inference speed.

As shown in Figure 8, besides RoBERTa and ALBERT, there are various PTMs being proposed in recent years towards better capturing knowledge from unlabeled data. Some work improves the model architectures and explores novel pre-training tasks, such as XLNet (Yang et al., 2019), UniLM (Dong et al., 2019), MASS (Song et al., 2019), SpanBERT (Joshi et al., 2020) and ELECTRA (Clark et al., 2020). Besides, incorporating rich data sources is also an important direction, such as utilizing multi-lingual corpora, knowledge graphs, and images. Since the model scale is a crucial success factor of PTMs, researchers also explore to build larger models to reach over hundreds of billions of parameters, such as the series of GPT (Radford et al., 2019; Brown et al., 2020), Switch Transformer (Fedus et al., 2021), and meanwhile conduct computational efficiency optimization for training PTMs (Shoeybi et al., 2019; Rajbhandari et al., 2020; Ren et al., 2021). In the following sections, we will further introduce all these efforts for PTMs in detail.

4 Designing Effective Architecture

In this section, we dive into the after-BERT PTMs deeper. The success of Transformer-based PTMs has stimulated a stream of novel architectures for modeling sequences for natural language and beyond. Generally, all the after-BERT Transformer architectures for language pre-training could be categorized according to two motivations: toward **unified sequence modeling** and **cognitive-inspired architectures**. Besides, we also take a glimpse over other important BERT variants in the third subsection, which mostly focus on improving natural language understanding.

4.1 Unified Sequence Modeling

Why is NLP so challenging? One of the fundamental reasons is that it has versatile downstream tasks and applications, which could be generally categorized into three genres:

- Natural language understanding: includes grammatical analysis, syntactic analysis, word/sentence/paragraph classification, question answering, factual/commonsense knowledge inference and etc.
- Conditional generation: includes machine translation, abstract summarizing, blank filling and etc.
- Unconditional generation: includes context-free text generation and etc.

Nevertheless, the differences between them are not so significant. As Feynman's saying goes, "What I cannot create, I do not understand". On one hand, a model that can not understand must not fluently generate; on the other hand, we can easily turn understanding tasks into generation tasks (Schick and Schütze, 2020). Recent studies also show that GPTs can achieve similar and even better performance on understanding benchmarks than BERTs (Liu et al., 2021b). The boundary between understanding and generation is vague.

Based on the observation, a bunch of novel architectures has been seeking for unifying different types of language tasks with one PTM. We will take a look over its development and discuss the inspirations they bring towards a unified foundation of natural language processing.

Combining Autoregressive and Autoencoding Modeling. The pioneer work to unify GPT-style unidirectional generation and BERT-style bidirectional understanding is XLNet (Yang et al., 2019), which proposes the permutated language modeling. The masked-recover strategy in BERT naturally contradicts with its downstream application, where no [MASK] is seen. XLNet solves the problem by permutating tokens' order in the pre-training and then applying the autoregressive prediction paradigm, which endows XLNet with the ability for both understanding and generation. An important follower of permutation language modeling is MPNet (Song et al., 2020), which amends the XLNet's discrepancy that in pre-training XLNet does not know the sentence's length while in downstream it knows.

Besides permutated language modeling, another stream would be multi-task training. UniLM (Dong et al., 2019) proposes to jointly train different language modeling objectives together, includes unidirectional, bidirectional, and seq2seq objectives. This can be achieved by changing the attention masks in Transformers. UniLM performs quite well in generative question answering and abstract summarization.

Recently, GLM (Du et al., 2021) proposes a more elegant approach for combining autoregressive and autoencoding. Given a variable-length masked span, instead of providing the number of [MASK] to model as BERT and SpanBERT (Joshi et al., 2020) do, GLM asks the Transformer to autoregressively generate the masked tokens. And to preserve the information of [MASK]s' number, GLM proposes a 2D positional encoding strategy. GLM is the first model to achieve the best performance on all types of tasks including natural language understanding, conditional generation, and unconditional generation at the same time.

Applying Generalized Encoder-Decoder. Before GLM, both encoder structure (e.g., BERT) or decoder structure (e.g., GPT) can not solve an important problem: to fill in blanks with variable lengths (Du et al., 2021; Shen et al., 2020b). The decoder can not make it because it can only generate at the end of the sequence, while the encoder can not because the number of [MASK]s will leak information. A natural idea is to turn to encoder-decoder architecture originally designed for machine translation, which would produce variable lengths of target sequences conditioned on the sources.

The pioneer of this genre is MASS (Song et al., 2019), which introduces the masked-prediction strategy into the encoder-decoder structure. However, MASS does not touch the problem of filling variable-length blanks. T5 (Raffel et al., 2020) solves the problem by masking a variable-length of span in text with only one mask token and asks the decoder to recover the whole masked sequence. BART (Lewis et al., 2020a) introduces the interesting idea of corrupting the source sequence with multiple operations such as truncation, deletion, replacement, shuffling, and masking, instead of mere masking. There are following works that specify in typical seq2seq tasks, such as PEGASUS (Zhang et al., 2020a) for abstract summarization and PALM (Bi et al., 2020).

Table 1: Three fundamental types of framework and their suitable downstream tasks. "NLU" refers to natural language understanding. "Cond. Gen." and "Uncond. Gen." refer to conditional and unconditional text generation, respectively. "✓" means "is good at", "—" means "could be adapted to", and "×" means "cannot be directly applied to". We define unconditional generation as the task of generating text without further training as in a standard language model, while conditional generation refers to seq2seq tasks such as text summarization. Taken from (Du et al., 2021).

Framework	NLU	Cond. Gen.	Uncond. Gen.
Autoregressive	—	—	✓
Autoencoding	✓	×	×
Encoder-Decoder	—	✓	—

However, several challenges lie in front of encoder-decoder architectures. First, the encoder-decoder introduces much more parameters compared to a single encoder/decoder. Although this problem could be alleviated by parameter-sharing of the encoder and decoder, its parameter-efficiency is still doubtful. Second, encoder-decoder structures generally do not perform very well on natural language understanding. Despite reported improvements over similar-sized vanilla BERT, well-trained RoBERTa/GLM encoder performs much better than them.

4.2 Cognitive-inspired Architectures

Is the current Transformer a good enough implementation of human beings' cognitive system? Of course not. Attention mechanism, the core module in Transformer architecture, is inspired by the micro and atom operation of the human's cognitive system and only responsible for the perceptive function. However, human-level intelligence is far more complex than the mere understanding of the

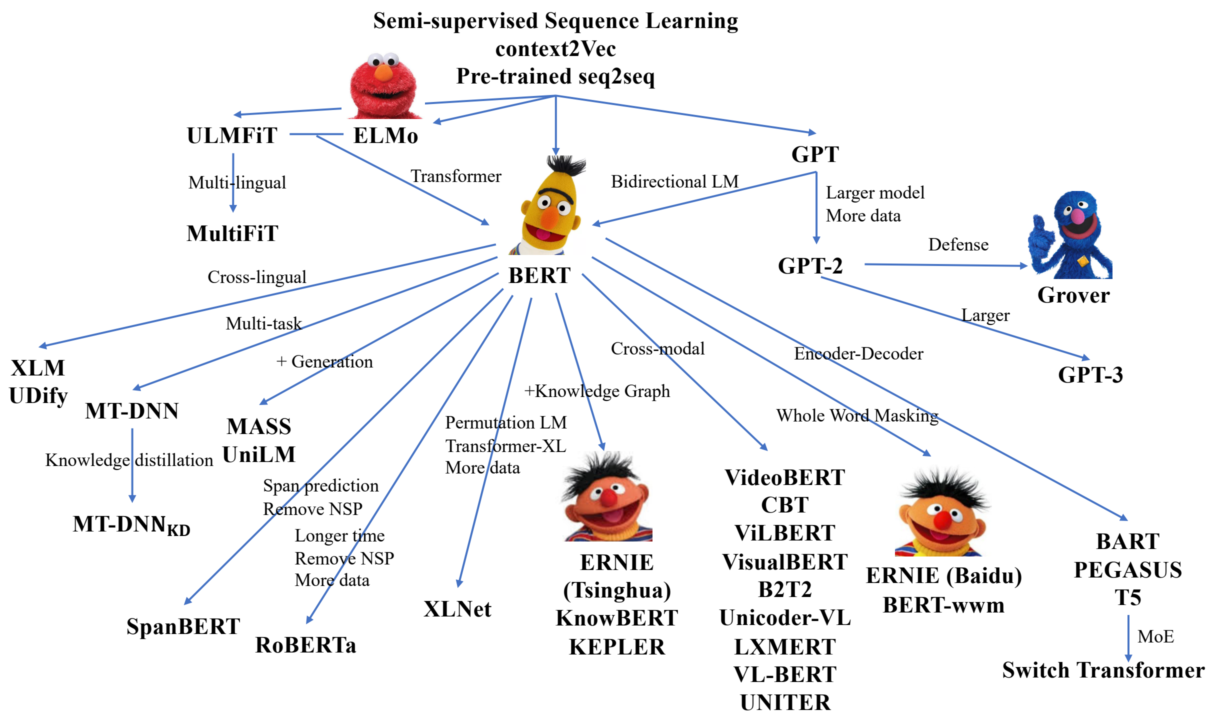


Figure 8: The family of recent typical PTMs, including both pre-trained language models and multimodal models.

association between different things.

In pursuit for human-level intelligence, understanding the macro architecture of our cognitive functions including decision making, logical reasoning, counterfactual reasoning and working memory (Baddeley, 1992) is crucial. In this subsection, we will take a look over the novel attempts inspired by advances of cognitive science, especially on maintainable working memory and sustainable long-term memory.

Maintainable Working Memory. A natural problem of Transformer is its fixed window size and quadratic space complexity, which significantly hinders its applications in long document understanding.

Despite the bunch of modifications on approximate computing of the quadratic growing pointwise attention (Tay et al., 2020), a question is that we humans do not present such a long-range attention mechanism. As an alternative, cognitive scientists have revealed that humans could maintain a working memory (Baddeley, 1992; Brown, 1958; Barrouillet et al., 2004; Wharton et al., 1994), which not only memorizes and organizes but also forgets. The conventional long-short term memory network is an exemplar practice for such a philosophy.

For Transformer-based architecture, the Transformer-XL (Dai et al., 2019) is the first to introduce segment-level recurrence and relative positional encoding to fulfill this goal. However, the recurrence only implicitly models the working memory. As a more explicit solution, CogQA (Ding et al., 2019) proposes to maintain a cognitive graph in the multi-hop reading. It is composed of two systems: a system 1 based on PTMs and a system 2 based on GNNs to modeling the cognitive graph for multi-hop understanding.

A limitation of CogQA is that its use of system 1 is still based on fixed window size. To endow working memory with the ability to understand long documents, CogLTX (Ding et al., 2020) leverages a MemRecall language model to select sentences that should be maintained in the working memory and another model for answering or classification.

Sustainable Long-term Memory. The success of GPT-3 (Brown et al., 2020) and recent studies on language models' ability in recalling factual knowledge (Petroni et al., 2019; Wang et al., 2020a; Liu et al., 2021b) has revealed the fact that Transformers can memorize. But how does Transformers make it?

In (Lample et al., 2019), the authors provide some inspiring evidence on how Transformers memorize. They replace the feed-forward networks in a Transformer layer with large key-value memory networks, and find it to work pretty well. This somehow proves that the feed-forward networks in Transformers is equivalent to memory networks.

Nevertheless, the memory capacity in Transformers is quite limited. For human intelligence, besides working memory for deciding and reasoning, the long-term memory also plays a key role in recalling facts and experiences. REALM (Guu et al., 2020) is a pioneer to explore how to construct a sustainable external memory for Transformers. The authors tensorize the whole Wikipedia sentence by sentence, and retrieve relevant sentences as context for masked pre-training. The tensorized Wikipedia is asynchronously updated for a given number of training steps. RAG (Lewis et al., 2020b) extends the masked pre-training to autoregressive generation, which could be better than extractive question answering.

Besides tensorizing the text corpora, (Verga et al., 2020; Févry et al., 2020) propose to tensorize entities and triples in existing knowledge bases. When entities appear in contexts, they replace entity tokens' embedding in an internal Transformer layer with the embedding from outer memory networks. (Dhingra et al., 2020; Sun et al., 2021) maintain a virtual knowledge from scratch, and propose a differentiable reasoning training objective over it. All of these methods achieve promising improvement on many open-domain question answering benchmarks.

4.3 More Variants of Existing PTMs

Besides the practice to unify sequence modeling and construct cognitive-inspired architectures, most current studies focus on optimizing BERT's architecture to boost language models' performance on natural language understanding.

A stream of work aims at improving the masking strategy, which could be regarded as a certain kind of data augmentation (Gu et al., 2020). SpanBERT (Joshi et al., 2020) shows that masking a continuous random-length span of tokens with a span boundary objective (SBO) could improve BERT's performance. Similar ideas have also been explored in ERNIE (Sun et al., 2019b,c) (where a whole entity is masked), NEZHA (Wei et al., 2019), and Whole Word Masking (Cui et al., 2019).

Another interesting practice is to change the masked-prediction objective to a harder one. ELECTRA (Clark et al., 2020) transform MLM to a replace token detection (RTD) objective, in which a generator will replace tokens in original sequences and a discriminator will predict which are replaced.

5 Utilizing Multi-Source Data

In this section, we introduce some typical PTMs that take advantage of multi-source heterogeneous data, including multilingual PTMs, multimodal PTMs, and knowledge-enhanced PTMs.

5.1 Multilingual Pre-Training

Language models trained on large-scale English corpora have achieved great success in many benchmarks. However, we live in a multilingual world, and training a large language model for each language is not an elegant solution considering the cost and the amount of data required. In fact, people from all over the world can express the same meaning, although they use different languages. This suggests that semantics is irrelevant with symbols. Additionally, researchers of (Lample and Conneau, 2019; Huang et al., 2020b) found that they could get even better performance on benchmarks when training one model with several languages comparing with training several uni-lingual models. Thus, training one model to learn multilingual representations may be a better way.

Before BERT, some researches have been conducted in exploring multilingual representations. There are mainly two ways to learn multilingual representations. One way is through parameter sharing. For example, training multilingual LSTMs with several language pairs together to achieve multilingual translation. Another way is learning language-agnostic constraint, such as decoupling language representations into language-specific and language-agnostic representations utilizing the WGAN (Arjovsky et al., 2017) framework. These methods enable models to be applied to multilingual scenarios, but they are limited to specific tasks. Because the model in each of them is trained with one specific task from beginning to end, and cross-lingual knowledge cannot be generalized to other tasks. Hence, for other multilingual tasks, training new models from scratch is still required. Learning new models from scratch needs a large volume of task-specific data.

The appearance of BERT shows that the framework of PTMs with general self-supervised learning tasks and then fine-tuning on specific downstream tasks is feasible. This inspires researchers to design tasks to pre-train multilingual models. According to training objectives, multilingual tasks could be divided into understanding tasks and generation tasks. Understanding tasks focus on sentence or word-level classification, which are of help for downstream classification tasks like natural language inference (Conneau et al., 2018b). Generation tasks focus on sentence generation, which is crucial in generation downstream tasks such as machine translation.

Some understanding tasks are first trained through monolingual corpora. For example, multilingual BERT (mBERT) released by Devlin et al. (2019) is pre-trained with the multilingual masked language modeling (MMLM) task using monolingual Wikipedia corpora in 104 languages. The research conducted by (Pires et al., 2019) shows that mBERT has the ability to generalize cross-lingual knowledge for zero-shot scenarios. This indicates that even with the same structure of BERT, using multilingual data can enable the model to learn cross-lingual representations. XLM-R (Conneau et al., 2020) build a monolingual dataset called CC-100, which supports 100 languages. The scale of CC-100 is much larger than the Wikipedia corpora used in mBERT, especially for those low-resource languages (Conneau et al., 2020). XLM-R is pre-trained with MMLM as the only task on CC-100 and gets better performance on several benchmarks than mBERT, which indicates that larger scale of monolingual training corpora will bring better performance.

However, MMLM cannot handle parallel corpora, which is quite important for some NLP tasks like neural machine translation. Intuitively, parallel corpora are very helpful to directly learn cross-lingual representations for those sentences in different languages with same meanings. From this point, XLM (Lample and Conneau, 2019) leverages bilingual sentence pairs to perform translation language modeling (TLM). Similar to MLM in BERT, TLM combines two semantically matched sentences into one and randomly masks words in both parts of the sentence. Comparing with MLM, TLM requires models to predict the masked words depending on the contexts of both source and target languages. This encourages models to align the representations of two languages together.

Instead of applying MMLM to learn the representations of parallel sentence pairs, Unicoder (Huang et al., 2019a) provides two other pre-training tasks based on parallel corpora: cross-lingual word recovery (CLWR) and cross-lingual paraphrase classification (CLPC). CLWR uses target language embeddings to represent source language embeddings by leveraging attention mechanisms, and the objective is to recover the source language embeddings (Huang et al., 2019a). This task enables models to learn word-level alignments of different languages. CLPC treats aligned sentences as positive pairs and sample misaligned pairs to perform sentence-level classification, letting the model predict whether the input pair is aligned or not, which focus on learning sentence-level alignments. ALM (Yang et al., 2020) generates code-switched sequences automatically from parallel sentences and performs MLM on it, which forces models to make predictions based only on contexts of other languages. InfoXLM (Chi et al., 2020b) analyzes MMLM and TLM from the perspective of information theory and proposes a pre-training objective XLCO, which encourages models to distinguish aligned sentence pairs with misaligned negative examples in the framework of contrastive learning. HICTL (Wei et al., 2021) extends the idea of using contrastive learning to learn both sentence-level and word-level cross-lingual representations. ERNIE-M (Ouyang et al., 2020) proposes back-translation masked language modeling (BTMLM), which expands the scale of parallel corpora through back-translation mechanisms. These works show that leveraging parallel corpora can bring much help towards learning cross-lingual representations.

Researches have also widely explored generation models. Normally, a generation model consists of a Transformer encoder and a Transformer decoder. For example, MASS (Song et al., 2019) extends MLM to language generation. It randomly masks a span of words in the input sentence to the encoder and predicts the masked words in an autoregressive manner with the decoder. A typical generation task is denoising autoencoding (DAE), which applies noise functions to the input sentence and then restores the original sentence by the decoder. Noise functions usually contain two operations: replacing a span of words with a mask token, as well as permuting the order of words. mBART (Liu et al., 2020c) extends DAE to support multiple languages by adding special symbols. It adds a language symbol both to the end of the encoder input and the beginning of the decoder input. This enables models to know the languages to be encoded and generated.

Although DAE in mBART (Liu et al., 2020c) is trained with multiple languages, the encoding input and the decoding objective are always in the same language. This leads the model to capture spurious correlations between language symbols and generated sentences. In other words, the model may ignore language symbols and generate sentences in the same language of the input. To address this issue, XNLG (Chi et al., 2020a) proposes cross-lingual autoencoding (XAE) task. Different from DAE, the encoding input and the decoding objective are in different languages, which is similar to machine translation. In addition, XNLG (Chi et al., 2020a) optimize parameters in two-stage manner. It trains the encoder with the MLM and TLM tasks in the first stage. Then, it fixes the encoder and trains the decoder with the DAE and XAE tasks in the second stage. All parameters are well pre-trained by this way, and the gap between pre-training with MLM and fine-tuning with autoregressive decoding is also filled.

5.2 Multimodal Pre-Training

Large-scale pre-training and its downstream applications have cascaded impactful research and development with diverse real-world modalities. We see objects, hear sounds, and speak languages. Modalities, such as audio, video, image, and text, refer to how something happens or is experienced. Tasks include multiple modalities that are developing in a fast-paced. More recently, large-scale pre-training has enhanced research interests in the intersection of the multiple modalities area, such as image and text or video and text. Specifically, this kind of modalities can all be classified as vision and language (V&L), as images and videos belong to vision, as well as text and speech (audio) belong to language. V&L tasks can be divided into image-text-based, video-text-based, and video-audio-based according to their specific modalities being used.

We now present a detailed overview of the previous trends in research pre-training to visual and language modalities. First, for the image-text-based V&L PTMs, the most current solutions are to adopt visual-linguistic BERT. By leveraging BERT, the main difficulty relies upon integrating non-text information into the framework of BERT. ViLBERT(Lu et al., 2019) is a model to learn task-agnostic joint representations of image content and natural language. It extends the BERT architecture to a multimodal model that supports two streams of input, preprocessing text and visual information separately. After two encoders, it uses the Transformer layer to obtain united attention. ViLBERT first provides a new mind for learning the relationship between vision and linguistic, which is no longer limited to learn a specific task but takes the relationship between vision and language as a pre-trainable and transferable ability of the model. It uses three pre-training tasks: MLM, sentence-image alignment(SIA), and masked region classification (MRC), and is evaluated on five downstream tasks: visual question answering (VQA), visual commonsense reasoning (VCR), grounding referring expressions (GRE), image-text retrieval (IR), and zero-shot image-text retrieval (ZSIR). LXMERT (Tan and Bansal, 2019) has similar architecture compared to VilBERT but uses more pre-training tasks: MLM, SIA, MRC, masked region feature regression (MRFR), and VQA, and is tested on three downstream tasks: VQA, graph question answering (GQA), and natural language for visual reasoning (NLVR2).

VisualBERT (Li et al., 2019), on the other side, extends the BERT architecture at the minimum. It can be regarded as a simple and effective baseline for V&L pre-training. The Transformer layer implicitly aligns elements in the input text and image regions. It uses two pre-training tasks: MLM and IA, and is tested on four downstream tasks: VQA, VCR, NLVR2, and Flickr30k. Unicoder-VL(Li et al., 2020a) moves the offsite visual detector in VisualBERT into end-to-end, it designs the image token as the sum of bounding box and object label features. It uses MLM, SIA, masked object classification (MOC) as the pre-training tasks and uses IR, ZSIR, and VCR as the downstream task. VL-BERT(Su et al., 2020) uses a similar architecture. Each input element is either a word from the input sentence or a region-of-interest (RoI) from the input image. It uses MLM and MOC as the pre-training tasks and finds that adding SIA will decrease the performance. It is evaluated on three downstream tasks: VQA, VCR, and GRE.

Some multimodal PTMs are designed to solve specific tasks, such as VQA. B2T2(Alberti et al., 2019) is the model that mainly focuses on VQA. It designs a model for early fusion of the co-reference between textual tokens and visual object features, and then uses MLM and SIA as the pre-training tasks. VLP(Zhou et al., 2020a) focuses on VQA and image captioning. It uses a shared multi-layer Transformer for both encoding and decoding, different from many existing methods whose encoder and decoder are implemented using separate models. It is pre-trained on bidirectional masked language prediction (BMLP) and sequence to sequence masked language prediction (s2sMLP). Among them, UNITER(Chen et al., 2020e) learns the unified representation between the two modalities. It tries many pre-trained tasks, such as MLM, SIA, MRC, MRFR, and is tested on various downstream tasks: VQA, IR, VCR, NLVR2, referring expression comprehension (REC), and visual entailment (VE).

ImageBERT (Qi et al., 2020) is as same as Unicoder-VL. It designs a novel weakly supervised approach to collect large-scale image-text data from the website, whose volume and quality are essential to vision-language pre-train tasks. The collecting steps include web-page collection, image filtering, sentence detection, sentence cleaning, image-text semantic scoring, and image-text aggregation. The resulting dataset contains 10M images and their descriptions with an average length of 13 words, which shows benefits to pre-training. The pre-training tasks include MLM, SIA, MOC, and MRFR, while only being tested on one downstream task: IR. Lu et al. (2020) investigate relationships between nearly all vision-and-language tasks by developing a large-scale, multi-task training regime. It classifies the common tasks into four groups: VQA, caption-based image retrieval, grounding referring expressions, and multimodal verification. It adopts two pre-training tasks by masking multimodal modeling only for aligned image-caption pairs and masking overlapped image regions, while performing well on five downstream tasks: VQA, GQA, IR, RE, and NLVR2.

X-GPT(Xia et al., 2020) finds that while previous BERT-based cross-modal PTMs produce excellent results on downstream understanding tasks, they cannot be applied to generation tasks directly. It is then proposed to pre-train text-to-image caption generators through three novel generation tasks, including image-conditioned masked language modeling (IMLM), image-conditioned denoising autoencoding (IDA), and text-conditioned image feature generation (TIFG). For downstream tasks, it focuses only on image captioning (IC). Oscar(Li et al., 2020c) uses object tags detected in images as anchor points to ease the learning of alignments significantly. It is motivated by the observation that the salient objects in an image can be accurately detected, and are often mentioned in the paired text. It performs well on six downstream tasks: IR, IC, novel object captioning(NOC), VQA, GCQ, and NLVR2.

Recently, CLIP (Radford et al., 2021) and WenLan (Huo et al., 2021) explore enlarging web-scale data for vision and language pre-training with big success. Comparing to previous works, they are faced with a large-scaled distributed pre-training challenge. WenLan(Huo et al., 2021) solves the problem by leveraging DeepSpeed(Rajbhandari et al., 2020).

5.3 Knowledge-Enhanced Pre-Training

PTMs can extract plenty of statistical information from large amounts of data. Besides, external knowledge, such as knowledge graphs, domain-specific data and extra annotations of pre-training data, is the outcome of human wisdom which can be a good prior to the modeling of statistics. In this section, we classify external knowledge according to the knowledge format and introduce several methods attempting to combine knowledge with PTMs.

The typical form of structured knowledge is a knowledge graph. Many works try to enhance PTMs by integrating entity and relation embeddings (Zhang et al., 2019b; Liu et al., 2020a; Peters et al., 2019; Sun et al., 2020; Rosset et al., 2020) or their alignments with the text (Xiong et al., 2019; Sun et al., 2019b). However, real-world knowledge graphs like Wikidata contain more information than entities and relations. Wang et al. (2021) pre-train models based on the descriptions of Wikidata entities, by incorporating a language model loss and a knowledge embedding loss together to get knowledge-enhanced representations. Some works regard the paths and even sub-graphs in knowledge graphs as a whole, and directly model them and the aligned text to retain more structural information. Since aligning entities and relations to raw text is often troublesome and can introduce noise in data pre-processing, another line of work (Bosselut et al., 2019; Guan et al., 2020; Chen et al., 2020d) can directly convert structural knowledge into the serialized text and let models learn knowledge-text alignments by themselves. Another interesting attempt is OAG-BERT (Liu et al., 2021a), which integrates heterogeneous structural knowledge in the open academic graph (OAG) (Zhang et al., 2019a), which covers 0.7 billion heterogeneous entities and 2 billion relations.

Compared to structured knowledge, unstructured knowledge is more intact but also noisier. How to effectively model this kind of knowledge from the data is also worth being explored. The data of a specific domain or task can be considered as a kind of unstructured knowledge. Many works (Beltagy et al., 2019; Lee et al., 2020) further pre-train the general PTMs on this data to get better domain-specific or task-specific models. Since there are some domain-specific and task-specific human annotations, Ke et al. (2020) incorporate these extra annotations to get better domain-specific and task-specific language representations. For all the above-mentioned works, knowledge is implicitly stored in their model parameters. To model external knowledge in a more interpretable way, some works (Lewis et al., 2020b; Guu et al., 2020) design retrieval-based methods to use structured knowledge on downstream tasks. Another kind of works (Wang et al., 2020b) can use adapters trained on different knowledge sources with extra annotations to distinguish where the knowledge is from.

6 Improving Computational Efficiency

As introduced in Section 1, a major trend of PTMs is that the number of parameters is getting larger and larger. Increasing the size of a neural network typically improves accuracy, but it also increases the memory and computational requirements for training the model.

6.1 System-Related Optimization

An effective and practical way to reduce computational requirements is system-related optimization towards computation efficiency and memory usage. System-related optimization methods are often model-agnostic and do not change the underlying optimization algorithm. Therefore, they are widely used in learning large-scale PTMs. The system-related optimizations can be divided into single-device optimization and multi-device optimization.

Single-Device Optimization. Current PTMs usually cost a lot of memory for pre-training due to their redundancy. One of the redundancy is the representation of floating points. Modern deep learning systems use a single-precision (FP32) format. But the weights of a model usually fall in a limited range, using a half-precision floating-point (FP16) format can accomplish most of the computation with little loss of the precision (Gupta et al., 2015).

However, in some cases, FP16 training may fail because of the truncation and overflow. To tackle this problem, mixed-precision training (Micikevicius et al., 2018) preserves some critical weights in FP32 to avoid overflow and uses dynamic loss scaling to get rid of truncation. Sufficient experiments have shown that mixed-precision training is more stable, and brings significant reduction towards training time and memory usage. However, when the model parameters are not initialized well, mixed-precision training is still likely to lead to unstable training. More exploration on low-precision training is still worth exploring.

Another kind of redundancy is the activation states saved for computing gradients. Taking a Transformer-based model as an example, apart from the weights of attention layers and linear layers, the device also stores the hidden states of each layer for the efficiency of the chain rule used in the gradient backpropagation. These hidden states can consume even much more memory than pure model parameters. Gradient checkpointing is an algorithm to save memory by storing only a part of the activation states after forward pass. The discarded activation states are recomputed during the backward pass if necessary.

When pre-training recent large-scale PTMs, the memory consumption can be too large to fit in a single GPU. Therefore, some works (Huang et al., 2020a) attempt to swap some of the model parameters or training states from the GPU memory to the CPU memory since the CPU memory is usually much larger. Some works design delicate strategies to select which memory part to swap and overlap the transition with the computation as much as possible (Ren et al., 2021).

Multi-Device Optimization. Recently, distributed training is commonly used in pre-training, where multiple GPUs distributed in lots of computational nodes are used together to train a single model. Data parallelism (Li et al., 2020b) is a simple but effective approach to accelerate training when a large batch size is needed. As shown in Figure 9, when using data parallelism, a large batch is partitioned to different devices, and thus forward pass can be parallelized. But at the backward pass, the

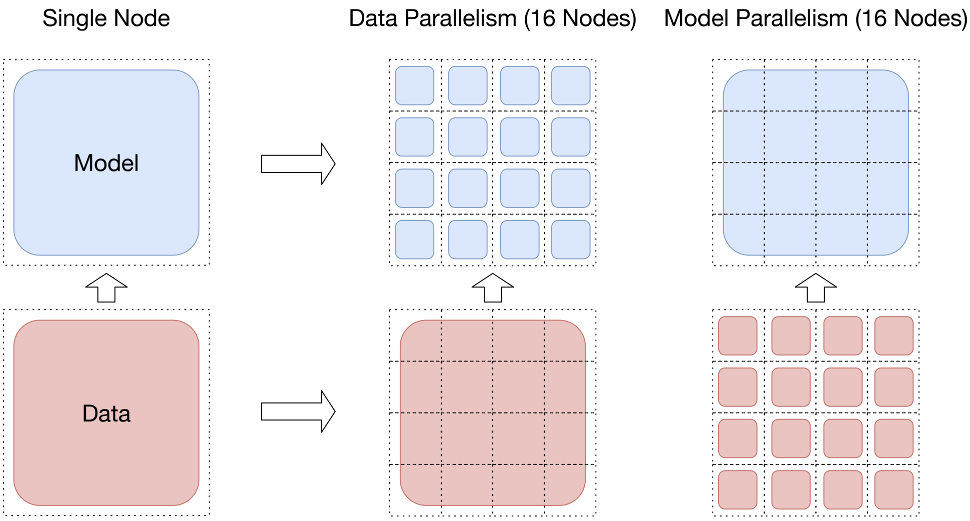


Figure 9: An illustration of the data parallelism and model parallelism with 16 nodes.

gradients on different devices should be aggregated with an all-reduce operation to ensure the parameter optimization consistency, which may introduce additional communication overhead.

When pre-training models with billions to trillions of parameters, traditional data parallelism brings challenges of fitting the whole model parameters into one GPU memory, even with half-precision training. Although one can overcome this problem is with GPUs of larger memory, for ordinary researchers, the expenses can be hard to afford, which limits the use of PTMs. Model parallelism is an effective way to tackle this problem(Shazeer et al., 2018). As shown in Figure 9, when doing model parallelism, model parameters of the same layer can be distributed to multiple nodes. The communication operations between these devices like reduce-scatter and all-gather guarantee the correctness of forward and backward. Megatron-LM (Shoeybi et al., 2019) adopts model parallelism to Transformer-based PTMs. They divide the self-attention heads as well as the weights of the feed-forward layer equally to different GPUs, reducing the memory burden of a single GPU. Mesh-Tensorflow (Shazeer et al., 2018) enables the users to split tensors along any tensor dimensions. This brings more options for model parallelism.

Although model parallelism enables different nodes to store different parts of the model, it has to insert collective communication primitives during both forward and backward pass, which can not be overlapped by computation. On the contrary, the all-reduce collective communication operation in data parallelism usually can be overlapped by the backward computation. As a result, data parallelism is preferred as long as it can conquer the excessive requirement of memory capacity. In the standard implementation of data parallelism, optimizer states are usually copied along different nodes to guarantee synchronized optimization across data parallelism units. This redundancy leads to the additional overhead of GPU memory, especially when the model is trained in a mixed-precision manner since the optimizer needs to store a 32-bit master state to ensure accuracy. Observing that the redundancy brought by optimizer states and parameters can be eliminated, ZeRO (Rajbhandari et al., 2020) equally partitions and distributes the optimizer states to each data parallelism node, such that each node only updates the optimizer states corresponding to its partition. At the end of a training step, all states are gathered across data parallelism nodes.

The above-mentioned model parallelism techniques try to partition and parallelize matrices operations across different nodes. Another effective method for model parallelism is pipeline parallelism which partitions a deep neural network into multiple layers and then puts different layers onto different devices. After the computation, each device sends the output to the next device where the next layer computation takes place. Since pipeline parallelism only needs to communicate the intermediate activation between devices performing adjacent stages of pipelining, the communication cost is relatively small. Existing pipeline methods include GPipe (Huang et al., 2019b), which can send smaller parts of samples within a mini-batch to different devices, and TeraPipe (Li et al., 2021), which can apply token-level pipeline mechanisms for Transformer-based models where each token in a sequence is processed by different devices. Both these pipeline methods speed up the large-scale PTMs. However, they should be stopped at the end of each batch since the backward propagation to complete, leading to pipeline bubbles.

6.2 More Efficient Pre-Training

Besides some system-related optimization methods, various efforts have been devoted to exploring more efficient pre-training methods, so that we can pre-train a large-scale PTMs with a less costly solution.

Efficient Training Methods. Conventional pre-training tasks can be sample-inefficient. For example, for MLM which is widely used to pre-train recent PTMs, the model is required to predict masked tokens according to contexts. The masked tokens are usually a subset (typically 15%) of the input tokens, i.e., the model can only learn from a small set of tokens. To tackle this problem, ELECTRA (Clark et al., 2020) proposes the replaced token detection task. This task forces the model to distinguish whether an input token is replaced by a generator. This task can leverage more supervision information from each sample since all the tokens in the input need to be distinguished. ELECTRA takes much fewer pre-training steps when it reaches the similar performance of the model with MLM. Furthermore, traditional MLM randomly selects tokens in a document to predict. Since the difficulty of predicting different tokens varies a lot, the random selecting strategy makes the training aimless and inefficient. Therefore, some works selectively mask the tokens based on their importance (Gu et al., 2020) or gradients (Chen et al., 2020b) in back-propagation to speed up model training.

Apart from the tasks, the current pre-training dynamics are also sub-optimal. Recent large-scale PTMs usually require a large batch size. But in an early work (Goyal et al., 2017), researchers find that naively increasing the batch size may cause difficulty in optimization. Therefore, they propose a warmup strategy that linearly increases the learning rate at the beginning of training. This strategy is commonly used in recent large PTMs. Another feature of recent PTMs is that they are usually composed of multiple stacks of a base structure like Transformer or ResNet. The conventional training paradigm optimizes each layer simultaneously using the same hyper-parameters. However, recent work studies Transformer-based models and claim that different layers can share similar self-attention patterns. Therefore, a shallow layer model can firstly be trained and then duplicated to construct a deep model (Gong et al., 2019). Some layers can also be dropped during training to reduce the complexity of back-propagation and weight update (Zhang and He, 2020). In addition, (You et al., 2017) and (You et al., 2020) find that adaptively using different learning rates at different model layers can also speed up convergence when the batch size is large.

Efficient Model Architectures. Besides efficient pre-training tasks, more variants of model architectures can also reduce the computational complexity to improve the efficiency of PTMs. For most Transformer-based PTMs, as the input sequence goes longer, the efficiency is limited by the computation of attention weights due to its quadratic time and space complexity of the sequence length. Therefore, many works attempt to reduce the complexity of Transformers. Some works (Peng et al., 2021; Choromanski et al., 2021; Wang et al., 2020c; Katharopoulos et al., 2020) design low-rank kernels to theoretically approximate the original attention weights and result in linear complexity. Some works (Child et al., 2019) introduce sparsity into attention mechanisms by limiting the view of each token to a fixed size and separating the tokens into several chunks so that the computation only takes place in every single chunk rather than the whole sequence. Compared to pre-defined chunks, some works (Roy et al., 2021; Kitaev et al., 2020) find that using learnable parameters to assign tokens into chunks results in better performance. Another kind of methods (Guo et al., 2019; Lee et al., 2019; Beltagy et al., 2020; Ainslie et al., 2020; Zaheer et al., 2020) combine global and local attention mechanisms and use global nodes to gather the tokens in a sequence. In this way, the long sequence is compressed into a small number of elements which reduces the complexity.

Keeping the same theoretical computation complexity as the original Transformer, more variants to the structures can also accelerate model convergence. Mix-of-experts (MoE) has been proved early (Shazeer et al., 2017) to increase the parameters of the deep neural models while keeping the computational overhead nearly unchanged. Recently, Switch Transformers (Fedus et al., 2021) employ this technique in pre-training. They add multiple experts to each layer in the Transformer model and select only one expert for computation during each forward and backward step and thus the training and inference time remain similar. They find that MoE-based models converge faster than the ordinary ones due to the significantly larger model capacity brought by multiple experts.

6.3 Model Compression

Another way to improve the efficiency of PTMs is model compression. In this setting, a large model is compressed to a small one to meet the demand for faster inference and deployment on resource-constrained devices.

Parameter Sharing. PTMs can be compressed with sharing parameters across similar units. ALBERT (Lan et al., 2019) uses factorized embedding parameterization and cross-layer parameter sharing to reduce the parameters of PTMs. Using the same weights across all Transformer layers, ALBERT achieves a significant reduction of the parameters based on the BERT model, keeping the same or even better performance of the original BERT model. This indicates that PTMs can be extremely over-parameterized.

Model Pruning. To take more advantage of the over-parameterized feature of current PTMs, another way to reduce the model parameters is model pruning, which refers to cut off some useless parts in PTMs to achieve accelerating while maintaining the performance. In (Fan et al., 2019), Transformer layers are selectively dropped during training resulting in a more shallow model during inference. In (Michel et al., 2019) and (Voita et al., 2019), researchers study the redundancy of the attention heads in Transformers and find that only a small part of them is enough for good performance. Most of them can be removed with little impact on the accuracy. Other trials such as CompressingBERT (Gordon et al., 2020) try to prune the attention matrices and the weight of linear layers to reduce the number of parameters in PTMs and also maintained comparable performance as the original model.

Knowledge Distillation. Although ALBERT saves the memory usage of PTMs, the inference time is not significantly decreased since the features still need to go through the layers with the same number as the original model. Another method called knowledge distillation trains a small model to reproduce the behavior of the large teacher model. The memory usage and the time overhead both decreased when using the small model for inference. Some typical works employing knowledge distillation in PTMs includes DistillBERT (Sanh et al., 2019), TinyBERT (Jiao et al., 2019), BERT-PKD (Sun et al., 2019a) and MiniLM (Wang et al., 2020d). In these works, a small version model is trained to mimic the output probability, the hidden states of each Transformer layer, and the attention matrices of the large model during both the pre-training and fine-tuning stages. With knowledge distillation, the "Modelegy" in the teacher model is transferred into the student model, which leads to performance increase compared to training a student model alone. However, the knowledge distillation methods mentioned above require the data used for large model pre-training, which is usually not released in consideration of the data copyright and privacy. Moreover, the teacher needs to forward over the entire pre-training data to produce its logits or intermediate representations, causing even longer training time.

Model Quantization. To get a more compressed model, model quantization is also a useful technique, which has been widely explored in CNN-based models (Stock et al., 2020; Polino et al., 2018). Model quantization refers to the compression of higher precision parameters to lower precision. Conventional PTMs are usually represented in 32 bits or 16 bits, while models after quantization can be in 8 bits or even 1 or 2 bits. For recent Transformer-based models, 8-bit quantization has been proved to be effective for model compression in Q8BERT (Zafrir et al., 2019), with little impact on the model performance. Despite this, training 1 or 2 Bits models remains challenging due to the significant decrease in model capacity. To alleviate the performance degradation, other methods to preserve the accuracy can also be employed. Q-BERT (Shen et al., 2020a) uses mixed-bits quantization in which the parameters with higher Hessian spectrum require higher precision while those with lower Hessian spectrum need lower. TernaryBERT (Zhang et al., 2020b) applies knowledge distillation in quantization, forcing the low-bit model to imitate the full-precision model. Both methods result in ultra low-bit models. However, low-bit representation is a highly hardware-related technique, which means quantization often requires specific hardware and can not generalize to other devices.

7 Interpretation and Theoretical Analysis

Beyond the superior performance of PTMs on various NLP tasks, researchers also explore to interpret the behaviors of PTMs, including understanding how PTMs work and uncovering the patterns that PTMs capture. These works cover several important properties of PTMs: knowledge, robustness, and structural sparsity/modularity. Moreover, there are some pioneering works on building the theoretical analysis for PTMs.

7.1 Knowledge of Pre-Trained Models

The implicit knowledge captured by PTMs can be roughly divided into two categories: linguistic knowledge and world knowledge.

Linguistic Knowledge. The linguistic knowledge of PTMs attracts most of attentions among all topics of PTMs' interpretation. Compared to conventional neural models such as CNNs and RNNs which have fewer layers and parameters, large-scale PTMs can learn rich linguistic knowledge from massive pre-training data. In order to study PTMs' linguistic knowledge, researcher design several approaches: (1) Representation Probing: Fix the parameters of PTMs and train a new linear layer on the hidden representations of PTMs for a specific probing task. It is the most popular approach because it can be easily adapted to any probing task without particular design. (2) Representation Analysis: Use the hidden representations of PTMs to compute some statistics such as distances or similarities. According to these statistics, we can construct the relation between different words, phrases, or sentences. (3) Attention analysis: similar to representation analysis, attention analysis compute statistics about attention matrices and is more suitable to discover the hierarchical structure of texts. (4) Generation Analysis: Use language models to directly estimate the probabilities of different sequences or words. The target texts could be correct or incorrect in some linguistic phenomenons.

Representation probing have been widely applied to analyze NLP neural models from word embeddings to PTMs (Köhn, 2015; Ettinger et al., 2016; Shi et al., 2016; Adi et al., 2017; Conneau et al., 2018a; Hewitt and Manning, 2019; Glavaš and Vulić, 2021). Liu et al. (2019) conduct comprehensive probing experiments on 11 linguistic tasks and find that the representations given by large-scale PTMs are competitive compared to previous task-specific models, which indicates that the models have already learned knowledge about tokens, chunks, and pairwise relations. To further investigate how PTMs represent sentence structures about syntactic, semantic, local, and long-range information, Tenney et al. (2019b) design a new edge probing task and examine PTMs on a broad suite of sub-sentence tasks and show that PTMs have strong ability to encode syntactic informative while they bring little improvement on semantic tasks. Similarly, several works also reveal the strong syntax encoding of PTMs (Vilares et al., 2020; Warstadt and Bowman, 2020; Hewitt and Manning, 2019). To analyze the function of different layers, Jawahar et al. (2019a) and Tenney et al. (2019a) show that PTMs encode linguistic information with phrase features at the bottom, syntactic features in the middle and semantic features at the top. Compared to non-contextual representations (e.g., word2vec), PTMs' representations are better in encoding sentence-level properties (Miaschi and Dell'Orletta, 2020). Furthermore, Manning et al. (2020) explore to reconstruct the sentence tree structures given by linguists using a linear transformation of PTMs' embeddings and achieve promising results.

Besides representation probing, researchers try to uncover the structure and relation among different representations. Kim et al. (2020) propose to leverage the concept of Syntactic Distance to construct the constituency trees of sentences from word representations. Rosa and Mareček (2019) analyze how the deletion of one word in a sentence changes representations of other words to reveal the influence of one word on other words.

There are also several works on interpreting PTMs via attention matrices. Lin et al. (2019) quantitatively evaluate attention matrices for subject-verb agreement and anaphor-antecedent dependencies, and show that PTMs tend to encode positional information in lower layers and capture hierarchical information in higher layers. To better characterize the behaviors of PTMs' attention matrices, Htut et al. (2019) propose to take the maximum attention weight and compute the maximum spanning tree as two statistics. Based on the experimental results, they find that fine-tuning has little impact on the self-attention patterns.

Since PTMs can be directly used to generate tokens or estimate the probabilities of different sentences, it is intuitive to construct analysis tasks based on generation (Goldberg, 2019). Perturbed Masking (Wu et al., 2020) recovers syntactic trees from PTMs without any extra parameter and the structure given by PTMs are competitive with a human-designed dependency schema in some downstream tasks. To analysis the gain of pre-training on estimating the probabilities of ungrammatical words, Schijndel (Schijndel et al., 2019) show that expanding the training corpus yields diminishing returns and the training corpus would need to be unrealistically large to make PTMs match human performance.

World Knowledge. In addition to linguistic knowledge, PTMs also learn rich world knowledge from pre-training, mainly including commonsense knowledge and factual knowledge (Zhou et al., 2020b; Bouraoui et al., 2020).

For the commonsense knowledge, Ettinger (Ettinger, 2020) first evaluates PTMs' knowledge in the aspect of psycholinguists and find that the models perform well in the situation of shared category or role reversal but fail with challenging inferences and role-based event. Then, to extract commonsense from PTMs, Davison et al. (2019) propose to first transform relational triples into masked sentences and then rank these sentences according to the mutual information given by PTMs. In the experiments, the PTM-based extraction method without further training even generalizes better than current supervised approaches. Similarly, Da and Kasai (2019) also find that PTMs have learned various commonsense features in their representation space based on a series of probing tasks. In addition to the commonsense features/attributes, the implicit relations between different attributes are important and Forbes et al. (2019) show that current PTMs' representations cannot model the implicit relations well, which requires further exploration.

For factual knowledge, Petroni et al. (2019) propose to formulate the relational knowledge generation as the completion of fill-in-the-blank statements. According to the experimental results, they find that PTMs significantly outperform previous supervised baselines on this task without any fine-tuning. However, the construction of these fill-in-the-blank statements is non-trivial. To extract more factual knowledge from PTMs, LPAQA (Jiang et al., 2020b) have been propose to automatically search better statements/prompts through mining-based and paraphrasing-based methods. AutoPrompt (Shin et al., 2020) proposes to train discrete prompts for knowledge probing. In P-tuning (Liu et al., 2021b), the authors discover that the better prompts lie in continuous embedding space, rather than discrete space. The P-tuning boosts the P@1 performance on LAMA to 64%, which is 20% higher than AutoPrompt. Moreover, Roberts et al. (2020) fine-tune PTMs for the task of open-domain question answering and find that fine-tuning can further benefit the knowledge generation of PTMs. However, Pörner et al. (2020) find that the success of knowledge generation may rely on learning neural stereotypical associations, i.e., a person with an Italian-sounding name will be predicted to Italian by PTMs. For understanding the number in texts, Wallace et al. (2019c) find that ELMo captures numeracy the best for all pre-trained methods, which is a character-based model, but BERT, which uses sub-word units, is less exact. (Wang et al., 2020a) investigates the knowledge stored in Transformer's feed-forward attention matrices, and therefore proposes a framework to unsupervisedly construct open knowledge graphs using PTMs.

7.2 Robustness of Pre-Trained Models

Recent works have identified the severe robustness problem in PTMs using adversarial examples. Adversarial attacks aims to generate new samples, which are misclassified by models, by small perturbation on the original inputs. For example, PTMs can be easily fooled by synonym replacement (Jin et al., 2020; Zang et al., 2020). Meanwhile, irrelevant artifacts such as form words can mislead the PTMs into making wrong predictions (Niven and Kao, 2019; Wallace et al., 2019a). Current works mainly utilize the model prediction, prediction probabilities, and model gradients of the models to search adversarial examples. However, it is difficult to maintain the quality of the adversarial examples generated by machines. Recently, human-in-the-loop methods (Wallace et al., 2019b; Nie et al., 2020) have been applied to generate more natural, valid, and diverse adversarial examples, which brings larger challenge and expose more properties and problems of PTMs. In conclusion, the robustness of PTMs has become a serious security threat when people deploy PTMs for real-world applications.

7.3 Structural Sparsity of Pre-Trained Models

Following BERT, most PTMs adopt Transformer as the architecture backbone. Although people can easily train a deep Transformer and achieve significant improvement over previous works using CNN and RNN, Transformer meets the problem of overparameterization. Researchers have shown that the multi-head attention structures are redundant in the tasks of machine translation (Michel et al., 2019), abstractive summarization (Baan et al., 2019), and language understanding (Kovaleva et al., 2019), i.e., when removing part of attention heads, we can achieve better performance. This phenomenon is consistent to the observation in (Clark et al., 2019) where they find that most heads in the same layer have similar self-attention patterns. Furthermore, Kovaleva et al. (2019) conduct a qualitative and quantitative analysis of the information encoded by PTMs' heads. Their findings suggest that the attention behaviors of different heads can be categorized into a limited set of patterns. Besides the multi-head attention, several other works explore to identify the sparsity of parameters. Gordon et al. (2020) show that low levels of pruning (30-40%) do not affect pre-training loss or the performance on downstream tasks at all. Targeting the sparsity during fine-tuning, Prasanna et al. (2020) validate the lottery ticket hypothesis on PTMs and find that it is possible to find sub-networks achieving performance that is comparable with that of the full model. Surprisingly, Kao et al. (2020) show that we can improvement the performance by simply duplicating some hidden layers to increase the model capacity, which suggests that the redundant parameters may benefit the fine-tuning.

7.4 Theoretical Analysis of Pre-Trained Models

Since pre-training has achieved great success in deep learning, researchers try to investigate how pre-training works, especially unsupervised pre-training. In the early days of deep learning, people found that it is effective to train a deep belief network by greedy layer-wise unsupervised pre-training followed by supervised fine-tuning (Hinton et al., 2006). Recently, pre-training based on contrast learning including language modeling has become the mainstream approach. In this section, we will introduce some theoretical explanatory hypotheses or frameworks for pre-training.

Erhan et al. (2010) propose two hypotheses to explain the effect of pre-training: (1) better optimization and (2) better regularization. In the aspect of better optimization, the network with pre-training is closer to the global minimum compared to the models randomly initialized. In the aspect of better regularization, the training error of PTMs is not necessarily better than the random models while the test error of PTMs is better, which means better generalization ability. Then, the experimental results lean towards the second hypothesis. They find that the PTM doesn't achieve lower training error. Moreover, compared to other regularization approaches such as L1/L2, the unsupervised pre-training regularization is much better.

Towards the recent development of pre-training objective, Saunshi et al. (2019) conduct a theoretical analysis of contrastive unsupervised representation learning. Contrastive learning treats the pairs of text/images appearing in the same context as the semantically similar pairs and the randomly sampled pairs as the semantically dissimilar pairs. Then, the distance between the similar pair should be close and the distance between the dissimilar pair should be distant. In the prediction process of language modeling, the context and the target word are the similar pair and the other words are negative samples (Kong et al., 2020). Saunshi et al. (2019) first provide a new conceptual framework to bridge the gap between pre-training and fine-tuning. Specifically, they introduce the concept of latent classes and the semantically similar pairs are from the same latent class. For example, the latent class can be "'happy" to include all texts including happy sentiments. The latent classes cover all possible classes and the classes defined by downstream tasks are from the set of latent classes. Then, they prove that the loss of contrastive learning is the upper bound of the downstream loss. Hence, when optimizing the pre-training loss, we can expect a lower loss in downstream tasks.

8 Future Direction

So far, we have comprehensively reviewed the past and the present of PTMs. In this section, we speculate on future directions of PTMs from the following perspectives: new architectures and methods (section 8.1), theoretical foundation (section 8.2), computational efficiency (section 8.3), multi-source data (section 8.4), knowledge bases (section 8.5), and cognitive learning (section 8.6).

Hopefully, this section could take a glance at the future of PTMs.

8.1 New Architectures and Methods

From the perspective of architectures and methods of PTMs, we belive the following directions worth further exploring in the future:

New Architectures. The Transformer has been proved to be an effective architecture for pre-training. However, the main limitation of the Transformer is its computation complexity. Limited by the memory of GPUs, most of current PTMs cannot deal with the sequence longer than 512 tokens. Therefore, searching for more efficient model architecture for PTMs is important to capture longer-range contextual information. However, the design of deep architecture is challenging, and we may seek help from some automatic methods, such as neural architecture search (NAS). Besides, although larger PTMs can usually lead to better performance, a practical problem is how to leverage these huge PTMs on special scenarios, such as low-capacity devices and low-latency applications, where the efficiency of PTMs is a key factor. Moreover, different downstream tasks prefer different architectures. For example, the Transformer encoder is suitable for natural language understanding tasks while the Transformer decoder is suitable for NLG tasks. Therefore, we can carefully design the specific architecture according to the type of downstream tasks.

New Pre-training Tasks. The general-purpose PTMs are always our pursuits for learning the intrinsic universal knowledge of languages (even world knowledge). However, such PTMs usually need deeper architecture, larger corpus, and challenging pre-training tasks, which further result in higher training costs. Moreover, training huge models is also a challenging problem, which needs more sophisticated and efficient training techniques such as distributed training, mixed precision, gradient accumulation, etc. Therefore, a more practical direction is to design more efficient model architecture, self-supervised pre-training tasks, optimizers, and training skills using existing hardware and software. ELECTRA (Clark et al., 2020) is a good attempt towards this direction.

Beyond Fine-tuning. Currently, fine-tuning is the dominant method to transfer PTMs' knowledge to downstream tasks, but one deficiency is its parameter inefficiency: every downstream task has its own fine-tuned parameters. An improved solution is to fix the original parameters of PTMs and by adding small fine-tunable adaption modules for specific task. Thus, we can use a shared PTM to serve multiple downstream tasks. Recently, as the emerging of GPT-3 (Brown et al., 2020), a novel genre for model tuning, namely prompt-based tuning, is getting more attention. By designing, generating and searching discrete (Petroni et al., 2019; Gao et al., 2021) or continuous (Liu et al., 2021b; Han et al., 2021; Lester et al., 2021) prompts and let PTMs conduct pre-training tasks like MLM, these models could (1) bridge the gap between pre-training and model tuning, thereby perform more efficiently on downstream tasks; (2) reduce the computation cost on fine-tuning the tremendous amounts of parameters. To sum up, prompt-based tuning is a promising way to stimulate the linguistic and world knowledge distributed in PTMS.

Reliability. The reliability of PTMs is also becoming an issue of great concern with the extensive use of PTMs in production systems. The studies of adversarial attacks against PTMs help us understand their capabilities by fully exposing their vulnerabilities. Adversarial defenses for PTMs are also promising, which improve the robustness of PTMs and make them immune against adversarial attack. Overall, as key components in many NLP applications, the interpretability and reliability of PTMs remain to be explored further in many respects, which helps us understand how PTMs work and provides a guide for better usage and further improvement.

8.2 Theoretical Foundation

In this section, we analyze the future directions in a more fundamental way. In the perspective of theoretical foundation, we give the following research directions:

Uncertainty. One under-addressed issue with PTMs (as well as other deep neural networks) is that they are often over-confident in predictions, i.e., “don't know what they don't know". For instance, GPT-3 can be used to answer questions with promising performance on benchmark datasets. However, if you ask a simple question like "How many eyes does my foot have?", GPT-3 would certainly produce an answer like "Your foot has two eyes", which looks counter-intuitive.[4] Of course,

[4]More examples of the Turing test of GPT-3 can

the above question is not often asked by normal humans. It is generally a challenging task to deal with such out-of-distribution (OOD) data in machine learning.

To address the above challenge, one promising direction is to adopt Bayesian methods that explore probabilistic tools to capture the uncertainty of both data and model (also known as aleatoric uncertainty and epistemic uncertainty, respectively) (Der Kiureghian and Ditlevsen, 2009) or derive some testing statistics. Such uncertainty or statistics is helpful to detect outliers (Wang et al., 2020f). Recently, much work has been done on the theory, algorithms and programming libraries of Bayesian deep learning, which conjoins Bayesian methods and deep networks (e.g., see (Shi et al., 2017) for more details). Such progress can be further extended to large-scale PTMs to properly characterize uncertainty and avoid over-confident outputs. Of course, improving the computational efficiency of Bayesian deep learning is a key factor to succeed.

Generalization and Robustness. Another important issue with PTMs is on generalization. As an important advancement of deep learning, it inherits the advantages as well as challenges of deep neural networks. It has been observed that classical learning theory is not sufficient to understand the behavior of deep networks (Zhang et al., 2017), thereby calling for new tools in learning theory. As for PTMs, besides theoretical understanding of the neural models themselves (e.g., Transformer and BERT), new questions arise. For example, it is important to theoretically understand the roles of pre-training in improving the generalization of downstream tasks. The recent work (Saunshi et al., 2019) provides a fruitful attempt on understanding contrastive learning with particular assumptions. However, it is still largely open to analyze under more realistic settings.

The adversarial robustness also raises new questions. In previous work, it was shown that a higher sample complexity is needed in order to achieve adversarial robustness for neural networks (Schmidt et al., 2018). Such analysis has inspired further improvements (e.g., (Pang et al., 2020)). However, it is generally unknown how the large-scale PTMs can help in this aspect. Are there effective ways to explore PTMs as extra data resources to improve the robustness of downstream tasks? Also, the robustness of PTMs themselves is an unsolved issue, as mentioned before.

be found at: https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html

8.3 Computational Efficiency

Deep learning models have become increasingly complicated and large (Devlin et al., 2019; Brown et al., 2020; Kaplan et al., 2020; Fedus et al., 2021). The novel requirements of large-scale DL models bring severe challenges to the existing DL frameworks such as TensorFlow (Abadi et al., 2016) and PyTorch (Paszke et al., 2019), which were designed in the early days without initially foreseeing the emerging requirements (e.g., model/pipeline parallelism of large models (Brown et al., 2020; Huang et al., 2019b; Wang et al., 2019)). To develop more efficient framework, the following directions are helpful:

Data Movement. Developing an efficient distributed DL framework faces various challenges. One has to carefully manage the data movement between devices, which may otherwise become the performance bottleneck (Narayanan et al., 2019; Jiang et al., 2020a). A well-defined parallelism strategy is needed to place and schedule compute tasks on inter-connected devices, minimizing the communication cost, maximizing computing and memory resources, and optimizing computation-communication overlap. In the best case, the efficient strategy is generated automatically.

Parallelism Strategy. Particular to the choices of parallelism strategy, data parallelism, model parallelism, pipeline parallelism, and various hybrid parallelism approaches can find their best usage depending on the structure of neural networks and hardware configuration (Ben-Nun and Hoefler, 2019). Data parallelism is especially suitable for DL models with a relatively small set of parameters (usually less than tens of million parameters) where near-linear speed-up can be achieved when the back-propagation maximally overlaps with gradient/parameter communication (Hashemi et al., 2019; Peng et al., 2019; Jiang et al., 2020a). Model parallelism and pipeline parallelism are for models with a more significant number of parameters, which probably cannot fit into a single device. In current practice, a user must consider the network structure given a DL model and the inter-device communication bandwidth thoroughly to decide the most appropriate parallelism strategies or switch between different strategies (Shazeer et al., 2018).

Large-scale Training. Given the poor support to model parallelism and pipeline parallelism by existing DL frameworks, some emerging open-source projects develop dedicated frameworks for large-scale training. For example, HugeCTR (Oldridge et al., 2020) is used for large-scale click-through rate estimation. Megatron-LMs (Shoeybi et al., 2019; Narayanan et al., 2021) and DeepSpeed (Rajbhandari et al., 2021, 2020) target large-scale NLP PTMs. InsightFace (ins, 2021) trains large-scale face recognition models. However, these frameworks are restricted to limited application cases and cannot serve as a general solution. Further, these approaches cannot work together to constitute a complete solution due to the compatibility issue.

Wrapper and Plugin. Without a mechanism to supporting model parallelism and pipeline parallelism, one has to develop various libraries dedicated to some particular algorithms via inserting the data routing ops by hand between computing ops on top of existing frameworks. Further, communication and computation need to be manually overlapped to maximize the system throughput. Manually programming the communication operations is prohibitively complicated and only can solve problem case by case, leading to a significant obstacle in applying model parallelism to new DL models. If communication operations can be automatically managed transparently to users by the DL framework, more DL models and applications can benefit from the distributed DL. To support more complicated parallelism strategies, many schemes are proposed as wrappers or plugins based on some mainstream DL frameworks such as TensorFlow and PyTorch. Mesh-TensorFlow (Shazeer et al., 2018), FlexFlow (Jia et al., 2019), OneFlow (one, 2021), MindSpore (min, 2021) and GShard (Lepikhin et al., 2020) provide APIs for developers to express a wide range of parallel computation patterns for different components of DNN models. The SBP configuration in OneFlow could be still too complex for the users to set. However, directly programming with communication primitives for a different kind of parallelism is more complicated. OneFlow transforms the manually programming to just setting *SBP* signatures. Moreover, in OneFlow, the user could just set the *SBP* signatures of a subset of ops instead of the whole set, and leave the rest *SBP* to be inferred with heuristic approach like GShard (Lepikhin et al., 2020), in which users provide some initial annotations or use default annotations as seed, then the algorithm propagates the sharding information to the un-annotated tensors. The approach in FlexFlow (Jia et al., 2019) can also be used here.

8.4 Multi-Modal and Multi-Lingual PTMs

Although multi-modal pre-training has witnessed numerous advances in the last two years (see section 5.2), it still has the following ongoing research lines:

More Modalities. In addition to image and text, video and audio can also be exploited for multi-modal pre-training. The main challenge thus lies in how to model the temporal context involved in these two modalities. In particular, for large-scale pre-training over video-text pairs, the conventional self-supervised learning (SSL) methods are not suitable due to their high computation loads, and our focus is then on developing more efficient SSL algorithms (e.g., SSL directly over compressed videos).

More Insightful Interpretation. It is still unknown why bridging vision and language works. For example, regardless of the advantages brought by multi-modal pre-training, does it lead to any harms to the single modality (image or text)? It the answer is yes, can we overcome this drawback during multi-modal pre-training? Along this research line, the latest visualization tools for deep learning can be exploited for the interpretation of multi-modal pre-training.

More Downstream Applications. It is well-known that multi-modal pre-training can be applied to image-text retrieval, image-to-text generation, text-to-image generation, and other downstream tasks. However, it is still challenging to find a 'true' real-world application scenario for multi-modal pre-training, since many effective engineering tricks can be leveraged instead (even with less cost). A closer collaboration with the industry is thus needed.

Transfer Learning. Currently, in order for a multimodal multilingual model to be able to handle different languages, data for each language is required during pre-training. It is not flexible to add unseen languages during pre-training. Therefore, a new pre-training framework should be explored so that it can use data in limited languages during pre-training, but can easily adapt to other unseen languages. Besides, current multimodal multilingual models are not able to process audio data. For example, to translate English audio to Chinese audio, we need to first transfer English audio to English text by an extra speech recognition system. After translation with a cross-lingual model, we need to further transfer Chinese text to Chinese audio by an extra text-to-speech tool. How to directly transfer audio of the source language to the target language text or target language audio by multimodal multilingual pre-training is worthy of exploring.

8.5 PTMs as Knowledge Bases

As introduced in section 7, PTMs can achieve a surge of improvements for a wide range of NLP tasks because they learn linguistic knowledge from large unlabeled corpora. As opposed to knowledge represented by discrete symbols, which is interpretable to humans, knowledge stored in PTMs is represented as real-valued vectors that are unexplainable. For example, given a triple $\langle h, r, t\rangle$ in a knowledge graph, it is easy to know that the head entity h has a relation r to the tail entity t. In contrast, you seem to have difficulty knowing what a representation produced by a PTM (eg., BERT) means. Therefore, we can refer to the knowledge stored in PTMs as continuous knowledge, which is distinguished from discrete knowledge stored in conventional knowledge bases.

Knowledge-aware Tasks. While the use of discrete knowledge is effective, it is time-consuming and labor-intensive to manually construct knowledge bases. With the rapid advance of researches on PTMs, there emerge various PTMs such as GPT, BERT, and BART. More and more researchers have probed into what knowledge do PTMs learn from the data, and why they perform so well on downstream tasks (Jawahar et al., 2019b; Ethayarajh, 2019). Petroni *et al.* (Petroni et al., 2019) state that PTMs can be seen as knowledge bases and study how to apply PTMs to the knowledge completion task. Wang *et al.* (Ethayarajh, 2019) also claim that PTMs would be open knowledge graphs and propose an unsupervised method to build knowledge graphs based on PTMs.

Store and Manage Knowledge. As existing PTMs are built on varying architectures and may be trained with different corpora, they contain diverse knowledge. As a result, how to store and manage various continuous knowledge in PTMs becomes a new challenge. There are two kinds of straightforward ideas. The first is to pre-train a huge model on extra-large scale data. Then, PTMs will have the extraordinary ability to cover almost all knowledge in existing PTMs. This method is simple and effective while it requires extremely high computing power and storage resources. For example, GPT-3 (Brown et al., 2020) uses about 175 billion parameters. The second is to combine multiple models into one large model based on mixture of experts (MoE) (Jacobs et al., 1991). For example, Fedus *et al.* (Fedus et al., 2021) improve MoE to propose Switch Transformers. This method is easy to contain new models but the requirement of memory grows as the number of models increases. Considering that there are similarities and differences among the existing PTMs, we have an important question that needs to be answered: Is it possible to build a universal continuous knowledge base (UCKB) that stores knowledge from various PTMs? The UCKB can not only store continuous knowledge imported from existing PTMs but also can blend different knowledge and then export the fused knowledge to a model to make it more powerful. Chen *et al.* (Chen et al., 2020a) first propose the concept of UCKB and make some preliminary explorations. They regard neural networks as parameterized functions and use knowledge distillation (Hinton et al., 2014) to import and export knowledge through the interface. UCKB overcomes the redundancy of model storage and stores the knowledge of various models into a common knowledge base. However, how to design the storage architecture of UCKB and the interface remains challenging.

8.6 Cognitive and Knowledgeable PTMs

Making PTMs more knowledgeable is an important topic for the future of PTMs. We divide the future development of knowledgeable PTMs into the following three directions:

Knowledge Augmentation. For an input text, there is rich related external knowledge, which can be used to augment the input. Considering the formats of knowledge and plain texts are very different, it is important to bridge the gap between text representation and knowledge representation (including symbols or vectors) and use their information uniformly as input. The solution on this problem requires both unified model architectures and knowledge-guided pre-training objectives.

Knowledge Support. Current model architectures are manually designed and usually very regular. With prior knowledge about the input, we can train different sub-module to process different kinds of input, which may accelerate the training and inference process and benefit the model efficiency. This process is similar to the human behavior that different brain regions correspond to different activity functions.

Knowledge Supervision. Knowledge bases store amounts of structural data, which can be used as an complementary source during pre-training. By learning from both knowledge bases and large-scale corpora, PTMs can have more language understanding and generation ability compared to only using plain texts. Through these three directions, we hope the future PTMs can easily understanding the meanings beyong the words and achieve better performance on various downstream tasks.

In terms of cognitive PTMs, we believe the following directions would be helpful:

Cognitive Architecture. Since neural networks are inspired by micro structure of human neural system, it is expected to see how macro function and organization of human cognitive system can enlighten the design of next generation of intelligence system, such as the Global Working Theory (GWT). The success of CogQA and CogLTX may provide some thoughts on this challenge.

Explicit and Controllable Reasoning with System 2. While deep learning has achieved success in many perceptive tasks, how to conduct complex dicision making and efficient multi-step reasoning is still unsolved, which may require machines to automatically plan the decision making process into a cognitive graph and do explicit reasoning over the factors in graph as human do.

Interactions of Model Knowledge. Though our PTMs are getting bigger and more general, what knowledge it has learned from pre-training is largely unexplored. Moreover, since our brains are working with the collaboration of different function zones, it is important to see if our PTMs have shaped different inner function modules and how they would interact with each other.

8.7 Applications

PTMs have been successfully applied in a wide variety of domains and tasks. In this section, we will highlight some of these applications.

Domain-specific PTMs. When large-scale domain-specific corpora are cheaply available, we can train domain-specific PTMs on such data. Some notable works include BioBERT (Lee et al., 2020) and SciBERT (Beltagy et al., 2019), which are trained respectively on biological and scientific literature texts. These models are expected and verified to learn more domain-specific knowledge and language use than those trained on general texts. Such domain expertise is usually regarded as important for solving many domain-specific problems.

Natural Language Generation (NLG). Many NLG tasks have been dominated by PTMs, such as GPT-2, BART, T5, UniLM, and many more. These tasks include machine translation, summarization, dialog generation, story generation, poetry generation, and other long text generation. Since the prevalent trend of PTMs, the backbone models have moved from RNN/GRU to transformers or transformer-based PTMs. As large-scale PTMs have been trained on so many data, they have innate advantages for NLG, particularly low-resourced NLG.

Dialog Systems. Many recent open-domain dialog systems are built upon large-scale transformer structures. These examples include Meena (Adiwardana et al., 2020), Blender (Roller et al., 2021), CDial-GPT (Wang et al., 2020e), Plato (Bao et al., 2020), and Plato-2 (Bao et al., 2021), which are trained on large-scale conversation data, commonly with the sequence-to-sequence framework. These models have shown capability of delivering natural and engaging conversations, some of which have been reported to be close to human-level performance (Adiwardana et al., 2020). However, dialog-specific pre-training tasks are yet to be explored, comparing to pre-training tasks in other applications.

Multi-modality Understanding and Generation. PTMs have also been successfully applied to multi-modality understanding and generation. Trained on text-image parallel data, these models have been shown strong in applications such as multi-modal information retrieval, visual question answering, image-to-text generation, and text-to-image generation.

Domain Adaptation and Task Adaptation. Large-scale PTMs learn general knowledge from large-scale general texts, which provide a good initial point to further learn domain-specific knowledge by fine-tuning or other techniques. Although PTMs are becoming larger and larger, the domain-specific data are always limited. Therefore, domain adaptation is becoming crucial for domain-specific applications. It has been evident that simple fine-tuning of the large-scale PTMs is not sufficient for domain-specific applications (Gururangan et al., 2020; Ke et al., 2020). The most essential reason for this is the distribution shift: the data distribution in a domain may be substantially different from that in general pre-training text.

Another important issue for the success of domain-specific applications goes to task adaptation. Most often, domain applications have a small set of labeled data, which can empower supervised learning to learn domain expertise more efficiently. However, for super-large PTMs, simply fine-tuning on labeled data seems to be inefficient in computation, nor effective in performance. Thus, how to bridge the gap between the pre-training phase and task-specific fine-tuning becomes crucial. Moreover, efficient and effective task-specific fine-tuning is also an important research direction.

9 Conclusion

In this paper, we take a look into the history of pre-training to indicate the core issue of PTMs, and meanwhile reveal the crucial position of PTMs in the AI development spectrum. Furthermore, we comprehensively review the latest efforts towards better PTMs, including designing effective architectures, utilizing rich contexts, improving computational efficiency, and conducting interpretation and theoretical analysis. All these works contribute to the recent wave of developing PTMs. Although existing PTMs have achieved promising results, especially those large-scale PTMs showing amazing abilities in zero/few-shot learning scenarios, how to develop PTMs next is still an open question. We briefly discuss a series of open problems and promising directions for future PTMs, and hope our view could inspire more efforts in this field and advance the development of PTMs.

Note and Contribution

This paper originates from a 3-day closed-door workshop initiated by Jie Tang, Ji-Rong Wen and Minlie Huang held in Beijing from January 1 to January 3, 2021, supported by China Computer Federation (CCF). All authors of this paper organized or participated in this workshop, and this paper can be regarded as a summary and extension of the discussion in the workshop.

The contributions of all authors are listed as follows: Zhiyuan Liu and Xu Han designed the structure of this paper, Xu Han drafted the abstract, Section 1 and Section 2, Ning Ding drafted Section 3, Xiao Liu and Jiezhong Qiu drafted Section 4, Yuqi Huo and Liang Zhang drafted Section 5, Yuxian Gu drafted Section 6, Zhengyan Zhang drafted Section 7. All faculty authors drafted various topics in Section 8, including Xipeng Qiu for Section 8.1, Jun Zhu and Yanyan Lan for Section 8.2, Jinhui Yuan and Wentao Han for Section 8.3, Ji-Rong Wen, Ruihua Song and Yang Liu for Section 8.4, Yang Liu for Section 8.5, Jie Tang and Zhiyuan Liu for Section 8.6, Minlie Huang and Jie Tang for Section 8.7. Wayne Xin Zhao, Xipeng Qiu provided comments to the manuscript, and Xu Han, Ning Ding and Zhengyan Zhang proofread the whole paper.

References

2021. Insightface project. https://github.com/deepinsight/insightface.

2021. MindSpore Deep Learning Framework. https://github.com/mindspore-ai/mindspore.

2021. OneFlow Deep Learning Framework. https://github.com/Oneflow-Inc/oneflow.

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of OSDI*, pages 265–283.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of ICLR*.

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Philip Pham, Anirudh Ravula, and Sumit Sanghai. 2020. ETC: Encoding long and structured inputs in transformers. In *Proceedings of EMNLP*, pages 268–284.

Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. 2019. Fusion of detected objects in text for visual question answering. In *Proceedings of EMNLP-IJCNLP*, pages 2131–2140.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of ICCV*, pages 2425–2433.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of ICML*, pages 214–223.

Joris Baan, Maartje ter Hoeve, Marlies van der Wees, Anne Schuth, and Maarten de Rijke. 2019. Understanding multi-head attention in abstractive summarization. *arXiv preprint arXiv:1911.03898*.

Alan Baddeley. 1992. Working memory. *Science*, 255(5044):556–559.

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. PLATO: Pre-trained dialogue generation model with discrete latent variable. In *Proceedings of ACL*.

Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhen Guo, Zhibin Liu, and Xinchao Xu. 2021. Plato-2: Towards building an open-domain chatbot via curriculum learning. In *Proceedings of ACL*.

Pierre Barrouillet, Sophie Bernardin, and Valérie Camos. 2004. Time constraints and resource sharing in adults' working memory spans. *Journal of Experimental Psychology: General*, 133(1):83–100.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *PNAS*, 116(32):15849–15854.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of EMNLP-IJCNLP*, pages 3615–3620.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Tal Ben-Nun and Torsten Hoefler. 2019. Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. *CSUR*, 52(4):1–43.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *JMLR*, 3:1137–1155.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *TNNLS*, 5(2):157–166.

Bin Bi, Chenliang Li, Chen Wu, Ming Yan, and Wei Wang. 2020. Palm: Pre-training an autoencoding&autoregressive language model for context-conditioned generation. In *Proceedings of EMNLP*, pages 8681–8691.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of WMT*, pages 12–58.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of ACL*, pages 4762–4779.

Zied Bouraoui, José Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from BERT. In *Proceedings of AAAI*, pages 7456–7463.

John Brown. 1958. Some tests of the decay theory of immediate memory. *Quarterly journal of experimental psychology*, 10(1):12–21.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*, pages 1877–1901.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Proceedings of ECCV*, pages 213–229.

Gang Chen, Maosong Sun, and Yang Liu. 2020a. Towards a universal continuous knowledge base. *arXiv preprint arXiv:2012.13568*.

Liang Chen, Tianyuan Zhang, Di He, Guolin Ke, Liwei Wang, and Tie-Yan Liu. 2020b. Variance-reduced language pretraining via a mask proposal network. *arXiv preprint arXiv:2008.05333*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020c. A simple framework for contrastive learning of visual representations. In *Proceedings of ICML*, pages 1597–1607.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020d. Kgpt: Knowledge-grounded pre-training for data-to-text generation. In *Proceedings of EMNLP*, pages 8635–8648.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Xinlei Chen and Kaiming He. 2020. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020e. Uniter: Universal image-text representation learning. In *Proceedings of ECCV*, pages 104–120.

Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. 2020a. Cross-lingual natural language generation via pre-training. In *Proceedings of AAAI*, pages 7570–7577.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020b. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. 2021. Rethinking attention with performers. In *Proceedings of ICLR*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of BlackboxNLP*, pages 276–286.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *Proceedings of ICLR*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*, pages 160–167.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*, pages 8440–8451.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018a. What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of ACL*, pages 2126–2136.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018b. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of EMNLP*, pages 2475–2485.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of CVPR*, pages 3213–3223.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Jeff Da and Jungo Kasai. 2019. Cracking the contextual commonsense code: Understanding commonsense reasoning aptitude of deep contextual representations. In *Proceedings of EMNLP Workshop*.

Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2007. Co-clustering based classification for out-of-domain documents. In *Proceedings of KDD*, pages 210–219.

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. 2008. Self-taught clustering. In *Proceedings of ICML*, pages 200–207.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of ACL*, pages 2978–2988.

Hal Daume III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *JAIR*, 26:101–126.

Joe Davison, Joshua Feldman, and Alexand er M. Rush. 2019. Commonsense knowledge mining from pretrained models. In *Proceedings of EMNLP-IJCNLP*, pages 1173–1178.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of CVPR*, pages 248–255.

Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Bhuwan Dhingra, Manzil Zaheer, Vidhisha Balachandran, Graham Neubig, Ruslan Salakhutdinov, and William W Cohen. 2020. Differentiable reasoning over a virtual knowledge base. In *Proceedings of ICLR*.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of ACL*, pages 2694–2703.

Ming Ding, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Cogltx: Applying bert to long texts. In *Proceedings of NeurIPS*, volume 33, pages 12792–12804.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of CVPR*, pages 2625–2634.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Proceedings of NeurIPS*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. All nlp tasks are generation tasks: A general pretraining framework. *arXiv preprint arXiv:2103.10360*.

Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. 2010. Why does unsupervised pre-training help deep learning? In *Proceedings of AISTATS*, pages 201–208.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of EMNLP-IJCNLP*, pages 55–65.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *TACL*, 8:34–48.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of RepEval*, pages 134–139.

An Evgeniou and Massimiliano Pontil. 2007. Multi-task feature learning. In *Proceedings of NeurIPS*.

Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *Proceedings of KDD*, pages 109–117.

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. In *Proceedings of ICLR*.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of EMNLP*, pages 4937–4951.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do neural language representations learn physical commonsense? In *Proceedings of CogSci*, pages 1753–1759.

Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. 2015. Are you talking to a machine? dataset and methods for multilingual image question answering. In *Proceedings of NeurIPS*, pages 2296–2304.

Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. 2008. Knowledge transfer via multiple model local structure mapping. In *Proceedings of KDD*, pages 283–291.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of ACL*.

Spyros Gidaris and Nikos Komodakis. 2015. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of ICCV*, pages 1134–1142.

Goran Glavaš and Ivan Vulić. 2021. Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *Proceedings of EACL*, pages 3090–3104.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Linyuan Gong, Di He, Zhuohan Li, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Efficient training of BERT by progressively stacking. In *Proceedings of ICML*, pages 2337–2346.

Mitchell A. Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing BERT: studying the effects of weight pruning on transfer learning. In *Proceedings of RepL4NLP*, pages 143–155.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2020. Train no evil: Selective masking for task-guided pre-training. In *Proceedings of EMNLP*, pages 6966–6974.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pre-training model for commonsense story generation. *TACL*, 8:93–108.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-transformer. In *Proceedings of HLT-NAACL*, pages 1315–1325.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. 2015. Deep learning with limited numerical precision. In *Proceedings of ICML*, pages 1737–1746.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*.

Sayed Hadi Hashemi, Sangeetha Abdu Jyothi, and Roy H Campbell. 2019. Tictac: Accelerating distributed deep learning with communication scheduling. In *Proceedings of MLSys*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of CVPR*, pages 9729–9738.

Kaiming He, Ross Girshick, and Piotr Dollár. 2019. Rethinking imagenet pre-training. In *Proceedings of ICCV*, pages 4918–4927.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of CVPR*, pages 770–778.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of NAACL-HLT*, pages 4129–4138.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014. Distilling the knowledge in a neural network. In *Proceedings of NeurIPS*.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of ACL*, pages 328–339.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R Bowman. 2019. Do attention heads in bert track syntactic dependencies? *arXiv preprint arXiv:1911.12246*.

Chien-Chin Huang, Gu Jin, and Jinyang Li. 2020a. Swapadvisor: Pushing deep learning beyond the gpu memory limit via smart swapping. In *Proceedings of ASPLOS*, page 1341–1355.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019a. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. In *Proceedings of EMNLP-IJCNLP*, pages 2485–2494.

Haoyang Huang, Lin Su, Di Qi, Nan Duan, Edward Cui, Taroon Bharti, Lei Zhang, Lijuan Wang, Jianfeng Gao, Bei Liu, et al. 2020b. M3p: Learning universal representations via multitask multilingual multimodal pre-training. *arXiv preprint arXiv:2006.02635*.

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Mia Xu Chen, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. 2019b. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Proceedings of NeurIPS*, pages 103–112.

Yuqi Huo, Manli Zhang, Guangzhen Liu, Haoyu Lu, Yizhao Gao, Guoxing Yang, Jingyuan Wen, Heng Zhang, Baogui Xu, Weihao Zheng, et al. 2021. Wenlan: Bridging vision and language by large-scale multi-modal pre-training. *arXiv preprint arXiv:2103.06561*.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of ICML*, pages 448–456.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. 2015. Spatial transformer networks. In *Proceedings of NeurIPS*, pages 2017–2025.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019a. What does BERT learn about the structure of language? In *Proceedings of ACL*, pages 3651–3657.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019b. What does bert learn about the structure of language? In *Proceedings of ACL*, pages 3651–3657.

Zhihao Jia, Matei Zaharia, and Alex Aiken. 2019. Beyond data and model parallelism for deep neural networks. In *Proceedings of MLSys*.

Yimin Jiang, Yibo Zhu, Chang Lan, Bairen Yi, Yong Cui, and Chuanxiong Guo. 2020a. A unified architecture for accelerating distributed DNN training in heterogeneous gpu/cpu clusters. In *Proceedings of OSDI*, pages 463–479.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020b. How can we know what language models know. *TACL*, 8:423–438.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. In *Proceedings of EMNLP*, pages 4163–4174.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of AAAI*, pages 8018–8025.

Justin Johnson, Andrej Karpathy, and Li Fei-Fei. 2016. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of CVPR*, pages 4565–4574.

Rie Johnson and Tong Zhang. 2005. A high-performance semi-supervised learning method for text chunking. In *Proceedings of ACL*, pages 1–9.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *TACL*, 8:64–77.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL*, pages 655–665.

Wei-Tsung Kao, Tsung-Han Wu, Po-Han Chi, Chun-Cheng Hsieh, and Hung-Yi Lee. 2020. Further boosting bert-based models by duplicating existing layers: Some intriguing phenomena inside bert. *arXiv preprint arXiv:2001.09309*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of ICML*, pages 5156–5165.

Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. 2020. Sentilare: Linguistic knowledge enhanced language representation for sentiment analysis. In *Proceedings of EMNLP*, pages 6975–6988.

Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. 2020. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. In *Proceedings of ICLR*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, pages 1746–1751.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *Proceedings of ICLR*.

Arne Köhn. 2015. What's in an embedding? analyzing word embeddings through multilingual evaluation. In *Proceedings of EMNLP*, pages 2067–2073.

Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. 2020. A mutual information maximization perspective of language representation learning. In *Proceedings of ICLR*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of EMNLP-IJCNLP*, pages 4364–4373.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123:32–73.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of NeurIPS*, pages 1097–1105.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Proceedings of NeurIPS*.

Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2019. Large memory layers with product keys. In *Proceedings of NeurIPS*, pages 8546–8557.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *Proceedings of ICLR*.

Neil D Lawrence and John C Platt. 2004. Learning to learn with the informative vector machine. In *Proceedings of ICML*.

Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. 2012. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. 2015. Deeply-supervised nets. In *Proceedings of AISTATS*, pages 562–570.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of ICML*, pages 3744–3753.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proceedings of ACL*, pages 7871–7880.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of NeurIPS*, pages 9459–9474.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020a. Unicoder-vl: A universal encoder for vision and language by cross-modal pretraining. In *Proceedings of AAAI*, pages 11336–11344.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, and Soumith Chintala. 2020b. Pytorch distributed: Experiences on accelerating data parallel training. In *Proceedings of PVLDB*, page 3005–3018.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020c. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Proceedings of ECCV*, pages 121–137.

Zhuohan Li, Siyuan Zhuang, Shiyuan Guo, Danyang Zhuo, Hao Zhang, Dawn Song, and Ion Stoica. 2021. Terapipe: Token-level pipeline parallelism for training large-scale language models. *arXiv preprint arXiv:2102.07988*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proceedings of ECCV*, pages 740–755.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside bert's linguistic knowledge. In *Proceedings of BlackboxNLP*, pages 241–253.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of NAACL-HLT*, pages 1073–1094.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of IJCAI*, pages 2873–2879.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020a. K-bert: Enabling language representation with knowledge graph. In *Proceedings of AAAI*, pages 2901–2908.

Xiao Liu, Da Yin, Xingjian Zhang, Kai Su, Kan Wu, Hongxia Yang, and Jie Tang. 2021a. Oag-bert: Pretrain heterogeneous entity-augmented academic language model. *arXiv preprint arXiv:2103.02410*.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. 2020b. Self-supervised learning: Generative or contrastive. *arXiv preprint arXiv:2006.08218*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020c. Multilingual Denoising Pre-training for Neural Machine Translation. *TACL*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020d. Roberta: A robustly optimized bert pretraining approach. In *Proceedings of ICLR*.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021c. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of CVPR*, pages 3431–3440.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of NeurIPS Reproducibility Challenge*.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of CVPR*, pages 10437–10446.

Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *PNAS*, 117(48):30046–30054.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Proceedings of NeurIPS*, pages 6294–6305.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of CoNLL*, pages 51–61.

Alessio Miaschi and Felice Dell'Orletta. 2020. Contextual and non-contextual word embeddings: an in-depth linguistic investigation. In *Proceedings of RepL4NLP*, pages 110–119.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Proceedings of NeurIPS*, pages 14014–14024.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2018. Mixed precision training. In *Proceedings of ICLR*.

Lilyana Mihalkova, Tuyen Huynh, and Raymond J Mooney. 2007. Mapping and revising markov logic networks for transfer learning. In *Proceedings of AAAI*, pages 608–614.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of ICLR Workshop*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NeurIPS*.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, pages 746–751.

Deepak Narayanan, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R. Devanur, Gregory R. Ganger, Phillip B. Gibbons, and Matei Zaharia. 2019. Pipedream: Generalized pipeline parallelism for dnn training. In *Proceedings of SOSP*.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. Efficient large-scale language model training on gpu clusters. *arXiv preprint arXiv:2104.04473*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of ACL*, pages 4885–4901.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of ACL*, pages 4658–4664.

Even Oldridge, J. Perez, Ben Frederickson, Nicolas Koumchatzky, M. Lee, Z.-H. Wang, Lei Wu, F. Yu, Rick Zamora, O. Yılmaz, Alec M. Gunny, Vinh Phu Nguyen, and S. Lee. 2020. Merlin: A gpu accelerated recommendation framework. In *Proceedings of IRS*.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-m: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. *arXiv preprint arXiv:2012.15674*.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *TKDE*, 22(10):1345–1359.

Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, and Jun Zhu. 2020. Rethinking softmax cross-entropy loss for adversarial robustness. In *Proceedings of ICLR*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of NeurIPS*.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. 2021. Random feature attention. In *Proceedings of ICLR*.

Yanghua Peng, Yibo Zhu, Yangrui Chen, Yixin Bao, Bairen Yi, Chang Lan, Chuan Wu, and Chuanxiong Guo. 2019. A generic communication scheduler for distributed dnn training acceleration. In *Proceedings of SOSP*, pages 16–29.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of EMNLP-IJCNLP*, pages 43–54.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of EMNLP-IJCNLP*, pages 2463–2473.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of ACL*, pages 4996–5001.

Antonio Polino, Razvan Pascanu, and Dan Alistarh. 2018. Model compression via distillation and quantization. In *Proceedings of ICLR*.

Nina Pörner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: efficient-yet-effective entity embeddings for BERT. In *Proceedings of EMNLP*, pages 803–818.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT plays the lottery, all tickets are winning. In *Proceedings of EMNLP*, pages 3208–3229.

Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. 2020. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *arXiv preprint arXiv:2001.07966*.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63:1872––1897.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sand hini Agarwal, Girish Sastry, Amand a Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. *OpenAI Blog*.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21:1–67.

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. 2007. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of ICML*, pages 759–766.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *Proceedings of SC*.

Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, Shaden Smith, and Yuxiong He. 2021. Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning. *arXiv preprint arXiv:2104.07857*.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. 2021. Zero-offload: Democratizing billion-scale model training. *arxiv preprint arXiv:2101.06840*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster r-cnn: towards real-time object detection with region proposal networks. *PAMI*, 39(6):1137–1149.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of EMNLP*, pages 5418–5426.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of EACL*.

Rudolf Rosa and David Mareček. 2019. Inducing syntactic trees from bert representations. *arXiv preprint arXiv:1906.11511*.

Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Knowledge-aware language model pretraining. *arXiv preprint arXiv:2007.00655*.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient content-based sparse attention with routing transformers. *TACL*, 9:53–68.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of NeurIPS*.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khand eparkar. 2019. A theoretical analysis of contrastive unsupervised representation learning. In *Proceedings of ICML*, pages 5628–5637.

Andrew M Saxe, James L McClelland, and Surya Ganguli. 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Marten Van Schijndel, Aaron Mueller, and Tal Linzen. 2019. Quantity doesn't buy quality syntax with neural language models. In *Proceedings of EMNLP-IJCNLP*, pages 5830–5836.

Michael Sejr Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of ESWC*, pages 593–607.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. 2018. Adversarially robust generalization requires more data. In *Proceedings of NeurIPS*.

Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Noam Shazeer, Youlong Cheng, Niki Parmar, Dustin Tran, Ashish Vaswani, Penporn Koanantakool, Peter Hawkins, HyoukJoong Lee, Mingsheng Hong, Cliff Young, et al. 2018. Mesh-tensorflow: Deep learning for supercomputers. In *Proceedings of NeurIPS*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proceedings of ICLR*.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020a. Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of AAAI*, pages 8815–8821.

Tianxiao Shen, Victor Quach, Regina Barzilay, and Tommi Jaakkola. 2020b. Blank language models. In *Proceedings of EMNLP*, pages 5186–5198.

Jiaxin Shi, Jianfei. Chen, Jun Zhu, Shengyang Sun, Yucen Luo, Yihong Gu, and Yuhao Zhou. 2017. ZhuSuan: A library for Bayesian deep learning. *arXiv preprint arXiv:1709.05870*.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of EMNLP*, pages 1526–1534.

Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of EMNLP*, pages 4222–4235.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Proceedings of ICLR*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *Proceedings of ICML*, pages 5926–5936.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Proceedings of NeurIPS*, pages 16857–16867.

Pierre Stock, Armand Joulin, Rémi Gribonval, Benjamin Graham, and Hervé Jégou. 2020. And the bit goes down: Revisiting the quantization of neural networks. In *Proceedings of ICLR*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. Vl-bert: Pre-training of generic visual-linguistic representations. In *Proceedings of ICLR*.

Haitian Sun, Pat Verga, Bhuwan Dhingra, Ruslan Salakhutdinov, and William W Cohen. 2021. Reasoning over virtual knowledge bases with open predicate relations. *arXiv preprint arXiv:2102.07043*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019a. Patient knowledge distillation for bert model compression. In *Proceedings of EMNLP-IJCNLP*, pages 4323–4332.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. In *Proceedings of COLING*, pages 3660–3670.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019b. Ernie: Enhanced representation through knowledge integration. In *Proceedings of ACL*, pages 1441–1451.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019c. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NeurIPS*, pages 3104–3112.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of CVPR*, pages 1–9.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of EMNLP-IJCNLP*, pages 5103–5114.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*.

Wilson L Taylor. 1953. Cloze procedure: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of ACL*, pages 4593–4601.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *Proceedings of ICLR*.

Sebastian Thrun and Lorien Pratt. 1998. *Learning to learn: Introduction and overview*. Springer Science & Business Media.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*, pages 384–394.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of ICLR*.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William W Cohen. 2020. Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge. *arXiv preprint arXiv:2007.00849*.

David Vilares, Michalina Strzyz, Anders Søgaard, and Carlos Gómez-Rodríguez. 2020. Parsing as pretraining. In *Proceedings of AAAI*, pages 9114–9121.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of CVPR*, pages 3156–3164.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of ACL*, pages 5797–5808.

Eric Wallace, Shi Feng, Nikhil Kand pal, Matt Gardner, and Sameer Singh. 2019a. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of EMNLP-IJCNLP*, pages 2153–2162.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019b. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *TACL*, 7:387–401.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019c. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of EMNLP-IJCNLP*, pages 5306–5314.

Chenguang Wang, Xiao Liu, and Dawn Song. 2020a. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*.

Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. 2017. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Minjie Wang, Chien-chin Huang, and Jinyang Li. 2019. Supporting very large models using automatic dataflow graph partitioning. In *Proceedings of EuroSys*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. 2020b. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. 2020c. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020d. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of NeurIPS*.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. Kepler: A unified model for knowledge embedding and pre-trained language representation. *TACL*, 9:176–194.

Yida Wang, Pei Ke, Yinhe Zheng, Kaili Huang, Yong Jiang, Xiaoyan Zhu, and Minlie Huang. 2020e. A large-scale chinese short-text conversation dataset. In *NLPCC*.

Zheng Wang, Yangqiu Song, and Changshui Zhang. 2008. Transferred dimensionality reduction. In *Proceedings of ECML-PKDD*, pages 550–565.

Ziyu Wang, Bin Dai, David Wipf, and Jun Zhu. 2020f. Further analysis of outlier detection with deep generative models. In *Proceedings of NeurIPS*.

Alex Warstadt and Samuel R. Bowman. 2020. Can neural networks acquire a structural bias from raw linguistic data? In *Proceedings of CogSci*.

Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. 2019. Nezha: Neural contextualized representation for chinese language understanding. *arXiv preprint arXiv:1909.00204*.

Xiangpeng Wei, Yue Hu, Rongxiang Weng, Luxi Xing, Heng Yu, and Weihua Luo. 2021. On learning universal representations across languages. In *Proceedings of ICLR*.

Charles M Wharton, Keith J Holyoak, Paul E Downing, Trent E Lange, Thomas D Wickens, and Eric R Melz. 1994. Below the surface: Analogical similarity and retrieval competition in reminding. *Cognitive Psychology*, 26:64–101.

Chris Williams, Edwin V Bonilla, and Kian M Chai. 2007. Multi-task gaussian process prediction. In *Proceedings of NeurIPS*, pages 153–160.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via nonparametric instance discrimination. In *Proceedings of CVPR*, pages 3733–3742.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of ACL*, pages 4166–4176.

Qiaolin Xia, Haoyang Huang, Nan Duan, Dongdong Zhang, Lei Ji, Zhifang Sui, Edward Cui, Taroon Bharti, Xin Liu, and Ming Zhou. 2020. Xgpt: Cross-modal generative pre-training for image captioning. *arXiv preprint arXiv:2003.01473*.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *Proceedings of ICML*, pages 2397–2406.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *Proceedings of ICLR*.

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2021. How neural networks extrapolate: From feedforward to graph neural networks. In *Proceedings of ICLR*.

Jian Yang, Shuming Ma, D. Zhang, Shuangzhi Wu, Zhou jun Li, and M. Zhou. 2020. Alternating language modeling for cross-lingual pre-training. In *Proceedings of AAAI*, pages 9386–9393.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of NeurIPS*.

Yang You, Igor Gitman, and Boris Ginsburg. 2017. Scaling sgd batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.03888*.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning: Training bert in 76 minutes. In *Proceedings of ICLR*.

Bianca Zadrozny. 2004. Learning and evaluating classifiers under sample selection bias. In *Proceedings of ICML*.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. In *Proceedings of NeurIPS*.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. In *Proceedings of NeurIPS*, pages 17283–17297.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of ACL*, pages 6066–6080.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *Proceedings of ICLR*.

Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Bin Shao, Rui Li, et al. 2019a. Oag: Toward linking large-scale heterogeneous entity graphs. In *Proceedings of KDD*, pages 2585–2595.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of ICML*, pages 11328–11339.

Minjia Zhang and Yuxiong He. 2020. Accelerating training of transformer-based language models with progressive layer dropping. In *Proceedings of NeurIPS*, pages 14011–14023.

Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. 2020b. Ternarybert: Distillation-aware ultra-low bit bert. In *Proceedings of EMNLP*, pages 509–521.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019b. Ernie: Enhanced language representation with informative entities. In *Proceedings of ACL*, pages 1441–1451.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. 2015. Conditional random fields as recurrent neural networks. In *Proceedings of ICCV*, pages 1529–1537.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020a. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of AAAI*, pages 13041–13049.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dand an Huang. 2020b. Evaluating commonsense in pretrained language models. In *Proceedings of AAAI*, pages 9733–9740.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. *Proceedings of NeurIPS*, 33.