Engineering 25 (2023) 51–65

Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Research
Artificial Intelligence—Review

Pre-Trained Language Models and Their Applications

Haifeng Wang [a,*], Jiwei Li [b], Hua Wu [a], Eduard Hovy [c], Yu Sun [a]

[a] Baidu Inc., Beijing 100193, China
[b] College of Computer Science and Technology, Zhejiang University, Hangzhou 310058, China
[c] Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ARTICLE INFO

Article history:
Received 10 November 2021
Revised 8 March 2022
Accepted 5 April 2022
Available online 7 September 2022

Keywords:
Pre-trained models
Natural language processing

ABSTRACT

Pre-trained language models have achieved striking success in natural language processing (NLP), leading to a paradigm shift from supervised learning to pre-training followed by fine-tuning. The NLP community has witnessed a surge of research interest in improving pre-trained models. This article presents a comprehensive review of representative work and recent progress in the NLP field and introduces the taxonomy of pre-trained models. We first give a brief introduction of pre-trained models, followed by characteristic methods and frameworks. We then introduce and analyze the impact and challenges of pre-trained models and their downstream applications. Finally, we briefly conclude and address future research directions in this field.

1. A brief history of pre-trained models

The concept of pre-training is related to transfer learning [1]. The idea of transfer learning is to reuse the knowledge learned from one or more tasks and apply it to new tasks. Traditional transfer learning employs annotated data for supervised training, which has been the common practice for at least a decade. Within deep learning, pre-training with self-supervised learning on massive unannotated data has become the dominant transfer learning approach. The difference is that pre-training methods use unannotated data for self-supervised training and can be applied to various downstream tasks via fine-tuning or few-shot learning.

In natural language processing (NLP), model pre-training is based on the task of language modeling. The goal of language modeling is to predict the next token, given a history of unannotated texts [2–4]. The first milestone of neural language modeling appears in Ref. [5], which models *n*-gram probabilities through distributed representations of words and feed-forward neural networks. Since then, deep learning methods have begun to dominate the training paradigm of language modeling. In early methods for neural language modeling, recurrent neural networks (RNNs) were widely used [6,7]. Among the RNN family, long short-term memory (LSTM) [8] stands out due to its advantage of being less prone to the gradient vanishing problem via its well-designed gating mechanism. With the emergence of the model known as transformer [9], considerable efforts have been devoted to building stronger and more efficient language models based on the transformer architecture [10–14]. In neural language modeling, distributed word representations named "word embeddings" that are learned with models such as Word2Vec [15] and GloVe [16] have become common initializations for the word vectors of deep learning models, significantly improving the performance of downstream tasks such as named-entity recognition [16], part-of-speech tagging [17], and question answering [18].

Although methods that leverage static word embeddings for warm startup can improve the performance of downstream NLP tasks, they lack the ability to represent different meanings of words in context. To solve this problem, context–aware language models were proposed to incorporate the complete context information into the training procedure. Dai and Le [19] introduced context–aware language modeling, which uses unannotated data to improve sequence learning with recurrent networks. This achieves significant performance improvement in sentiment analysis, text classification, and object classification tasks. In 2017, contextualized word vectors were proposed, which are derived from an encoder that is pre-trained on machine translation and then transferred to a variety of downstream NLP tasks [20]. However, these studies use a small amount of data for pre-training and do not achieve consistent performance improvement across all NLP tasks. Nonetheless, these pioneering studies greatly motivated follow-up pre-training methods for context modeling.

* Corresponding author.
E-mail address: wanghaifeng@baidu.com (H. Wang).

https://doi.org/10.1016/j.eng.2022.04.024

In another pioneering study on pre-trained models (PTMs), embeddings from language models were proposed to leverage bidirectional LSTMs in order to learn contextual word representations, and the pre-trained contextual embeddings were then applied to downstream tasks [21]. This method demonstrated great improvements in a broad range of NLP tasks, including question answering, textual entailment, sentiment analysis, semantic role labeling, coreference resolution, and named-entity extraction.

Since then, numerous PTMs within the "pre-training then fine-tuning" paradigm have started to emerge. Generative pre-training (GPT) [22] was the first model to use unidirectional transformers as the backbone for the GPT of language models, thereby illustrating the dramatic potential of pre-training methods for diverse downstream tasks. Following GPT [23], the first model to leverage bidirectional transformers was called Bidirectional Encoder Representations from Transformers (BERT); this model learns bidirectional contexts by means of conditioning on both the left and the right contexts in deep stacked layers. BERT introduced a denoising autoencoding pre-training task, termed masked language modeling (MLM), to recover the corrupted tokens of input sentences according to their contexts, in what was akin to a cloze task. This approach greatly boosted the performance gain of downstream natural language understanding (NLU) tasks. In this type of pre-training, which is also known as self-supervised learning, the pre-training labels are derived from unannotated data. By resorting to web-scale unannotated data from the Internet, PTMs can automatically learn syntactic and semantic representations.

The great success of PTMs has attracted a wide range of interest in scaling them up and exploring the boundaries of pre-training techniques; examples include decoding-enhanced BERT with disentangled attention (DeBERTa) [24], text-to-text transfer transformers (T5) [25], GPT-3 [26], large-scale generative Chinese pre-trained language model (CPM) [27], PanGu-α [28], and ERNIE 3.0 Titan [29]. Large-scale PTMs, such as GPT-3, have now demonstrated the powerful capabilities of zero-shot and few-shot learning. With dozens of examples, GPT-3 achieved a performance similar to that of BERT, being fine-tuned with tens of thousands of pieces of data on SuperGLUE [30]. GPT-3 can also generate high-quality creative texts so that even humans cannot determine whether or not the texts are written by a human. The success of GPT-3 makes it possible to use this model for general-purpose text generation, which was considered to be impossible in the past decades.

Another line of pre-training methods has attempted to incorporate knowledge in order to enhance the representation capability of PTM [31]. Some studies employ linguistic knowledge to design entity-related tasks with weak supervision. For example, they corrupt entity spans in texts and use knowledge-masking strategies such as entity-level or phrase-level masking [31] and entity replacement prediction [32] to better learn lexical, syntactic, and semantic information from texts. Another direction of research integrates structured knowledge together with plain texts into pre-training, such as knowledge-enabled BERT (K-BERT) [33], contextualized language and knowledge embedding (CoLAKE) [34], enhanced language representation with informative entities (ERNIE-THU) [35], knowledge-enhanced BERT (KnowBERT) [36], SenseBERT [37], knowledge embedding and pre-trained language representation (KEPLER) [38], and ERNIE 3.0 [39]. ERNIE 3.0, which powers PTMs with knowledge, has achieved new state-of-the-art (SOTA) performances across 54 Chinese NLP benchmarks, as well as some English benchmarks, including SuperGLUE [30]. Moreover, K-Adapter [40] uses multiple adapters for different tasks independently in order to better fuse various knowledge sources and mitigate catastrophic forgetting. Knowledge-based incorporation has dramatically improved knowledge sharing between unstructured text and structured knowledge, greatly promoting the capacity of knowledge memorization and reasoning in PTMs [39].

However, the aforementioned models only focus on rich-resource languages, such as English and Chinese, and thus may overlook numerous low-resource languages. Recent work on multilingual models is aiming to transfer knowledge from rich-resource languages to low-resource languages by modeling the semantic representation of disparate languages in a unified vector space. Inspirited by BERT, multilingual BERT (mBERT) was developed and released; this model is trained via multilingual masked language modeling (MMLM) on multilingual corpora [41]. From an intuitive perspective, the use of parallel corpora is conducive to learning cross-lingual representations in different languages. Therefore, cross-lingual language model (XLM) [42] leverages bilingual sentence pairs to perform translation language modeling (TLM), which encourages models to align the representations of two languages together. Researchers have also released more multilingual language models, such as XLM-RoBERTa (XLM-R) [43], InfoXLM [44], and ERNIE-M [45], by improving MMLM or TLM. These studies have demonstrated that pre-trained multilingual language models can significantly improve performance of multilingual NLP tasks or low-resource language tasks.

Given the success of PTMs in NLP, these models have quickly been extended to other fields such as computer vision [46–48] and speech processing [49]. Although self-supervised pre-training has been the most successful transfer learning method in NLP, the PTMs used for computer vision tasks are diversified. The dominant method in computer vision tasks is still supervised learning. Sun et al. [48] show that representation learning holds promise for advancing model performance based on large-scale (noisy) annotated datasets, such as ImageNet [50] or JTF300M [48]. These methods learn visual representations and significantly improve the performance of various downstream vision tasks [48]. Self-supervised pre-training have also been explored in computer vision [51–56]. Doersch et al. [53] propose various prediction tasks as propse tasks to learn visual representations. Dosovitskiy et al. [57] explore the masked patch prediction task using transformer architecture for images and demonstrates that pre-trained transformers achieve excellent results compared with convolutional neural networks (CNNs).

Recently, contrastive learning has been successfully utilized for visual self-supervised pre-training. Contrastive predictive coding [58] has achieved strong results in various scenarios, including speech, image, and text. These methods [58–60] attempt to maximize the similarity of two augmentations of an image and minimize the similarity of different images with contrastive loss. More recently, pre-training methods have been advanced by utilizing language supervision for visual representation learning [61], achieving a strong performance in image classification tasks and other vision tasks.

Pre-training methods have also been applied to multimodal applications, in which texts are combined with other modalities, such as images [62–65], videos [66,67], and speech [68], enabling a broad application scope of PTMs. Such methods [63] significantly improve the performance of various multimodal tasks by jointly learning task-agnostic representations of images and texts. Based on the transformer architecture, PTMs build cross-modal semantic alignments from large-scale image-text pairs. For image generation, DALL-E [69] and CLIP-guided generation [61] leverage multimodal language and vision input to render compelling visual scenes. Although the most commonly used pre-training tasks for multimodal context are MLM and masked region prediction, Yu et al. [70] propose knowledge-enhanced scene graph prediction to capture the alignments of more detailed semantics. Gan et al. [71] incorporate adversarial training into pre-training and achieves higher performance. Cho et al. [72] formulate multimodal

pre-training as a unified language modeling task based on multimodal context. This demonstrates that PTMs are playing a critical role in the artificial intelligence (AI) community and will potentially promote the unification of the pre-training framework across research fields such as speech, computer vision, and NLP.

There are some existing reviews on PTMs. Some focus on particular types and applications of PTMs, such as transformer-based pre-trained language models [73], BERT-based training techniques [74], prompted-based learning [75], data augmentation [76], text generation [77], and conversational agent design [78]. Another line provides a panoramic perspective of the whole progress of PTMs. For example, Ramponi and Plank [79] provide an overview from early traditional non-neural methods to PTMs in NLP. Qiu et al. [80] systematically categorize existing PTMs from four different perspectives and outlines some potential directions of PTMs for future research. Bommasani et al. [81] propose the concept of foundation models to unify PTMs in different subfields such as NLP, computer vision, and speech, and analyzes their opportunities and challenges in various AI domains. Han et al. [82] take a deep look into the history of PTMs to reveal the crucial position of PTMs in the AI development spectrum. In our review, we mainly focus on the PTMs in NLP: We first provide a detailed analysis of different PTMs and trends in PTMs at scale, discussing their impact on the field of NLP and the main challenges of PTMs; we then focus on our observations of and practices in the industrial applications of PTMs.

In this paper, we will first summarize the methods and taxonomy of pre-trained language models in Section 2, followed by a discussion of the impact and challenges of pre-trained language models in Section 3. Next, we will introduce the industrial applications of pre-training techniques in Section 4. Finally, we will conclude and address potential future work in this area.

2. Methods of PTMs

2.1. Different frameworks and extensions of PTMs

When working with PTMs, it is essential to design efficient training methods that can fully use unannotated data and assist downstream fine-tuning. In this section, we briefly introduce some widely used pre-training frameworks to date. Fig. 1 summarizes the existing prevalent pre-training frameworks, which can be classified into three categories: transformer decoders only; transformer encoders only; and transformer decoder–encoders. A brief description of each category is given below, and more detail is provided in the subsections that follow.

- Transformer decoders only frameworks use a unidirectional (left-to-right) transformer decoder as the pre-training backbone and predict tokens in a unidirectional autoregressive fashion. Here, "auto-regression" refers to predicting the current token based on historical tokens—that is, the partial sequence on the left of the current token. More specifically, given the text sequence $\boldsymbol{x} = (x_1, x_2, x_3, \ldots, x_T)$ (where $\boldsymbol{x}$ is the original sentence, x_t (t = 1, 2, ..., T) is the tth token, and T is the sequence length), an autoregressive model factorizes the likelihood of the input text sequence as $p(\boldsymbol{x}) = \prod_{t=1}^{T} p(x_t|x_{<t})$, where p is the likelihood of the input text sequence.
- Transformer encoder only frameworks leverage a bidirectional transformer encoder and aim to recover corrupted tokens, given the input sentences with randomly masked tokens.
- Transformer encoder–decoder frameworks aim at pre-training a sequence-to-sequence (seq2seq) generation model by masking tokens on the source side and recovering them on the target side. These frameworks consist of two classes: ① seq2seq encoder–decoders, which consist of a bidirectional transformer encoder and a unidirectional decoder with separate parameters; and ② unified encoder–decoders, in which a bidirectional transformer encoder and a left-to-right decoder are simultaneously pre-trained with shared model parameters.

2.1.1. Transformer decoders only

The objective for language modeling is to predict the next token auto-regressively, given its history. The nature of auto-regression entails the future invisibility of input tokens at each position; that is, each token can only attend to the preceding words. GPT [22] was the first model to use the transformer decoder architecture as its backbone. Given a sequence of words as context, GPT computes the probability distribution of the next word with the masked multi-head self-attention of the transformer. In the fine-tuning phase, the pre-trained parameters are set as the initialization of the model for downstream tasks. GPT is pre-trained on the BooksCorpus dataset, which is nearly the same size as the 1B Word Benchmark. It has hundreds of millions of parameters and improves SOTA results on nine out of 12 NLP datasets, showing

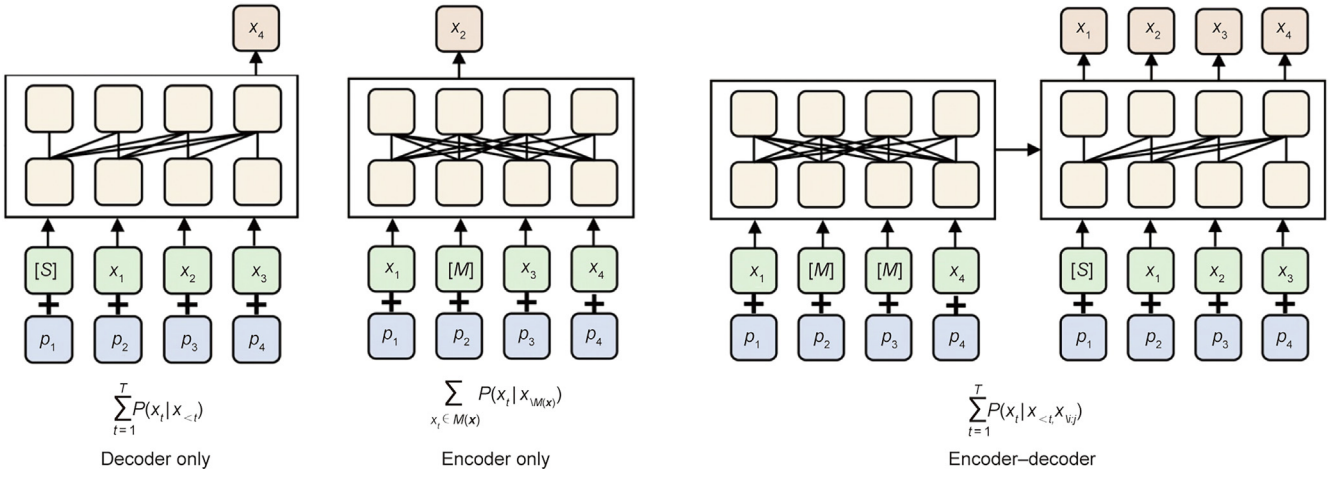

Fig. 1. An illustration of the existing prevalent pre-training frameworks, where $\boldsymbol{x}$ is the original sentence, x_t (t = 1, 2, ..., T) is the tth token, T is the sequence length, and $M(\boldsymbol{x})$ is the set of masked tokens in $\boldsymbol{x}$. S denotes the start token embedding of a sequence. p_1, p_2, p_3, and p_4 denote the position embeddings of the first to fourth tokens. P is the conditional probability. i and j indicate the start and the end indices of input tokens of the encoder, respectively.

the potential of large-scale PTMs. GPT-2 [83] follows the unidirectional framework with a transformer decoder that was trained with a larger corpus, namely, WebText, and 1.5 billion model parameters. GPT-2 achieves SOTA results on seven out of eight tested language modeling datasets in a zero-shot setting. GPT-3 [26] further increases the parameters of the transformer to 175 billion and introduces in-context learning. Both GPT-2 and GPT-3 can be applied to downstream tasks without fine-tuning. They achieve a strong performance by scaling up the model size and dataset size.

Unidirectional language modeling lacks attention on its full contexts on both sides, which may degrade its performance on downstream tasks. To tackle this problem, Yang et al. [84] propose the use of permuted language modeling (PLM), which performs autoregressive modeling on permuting input tokens. For example, a permutation of the sentence "I love the movie" can be "I the movie love." Once the permutation is chosen, the last few tokens of the permuted sentence are the target to predict. In the above example, the token "love" is the target, depending on the visible context "I the movie." An advantage of PLM is that it can fully leverage the contextual information for different masked tokens, thus building dependent context relationships with both preceding and successive words. To enable PLM, Yang et al. [84] propose a novel two-stream self-attention mechanism, with one query stream to compute the query vectors and another content stream to compute the key/context vectors. The two-stream self-attention approach evades the leakage of visible context to the masked positions.

2.1.2. Transformer encoders only

Pre-trained transformer encoders, such as BERT [23], have become the standard in NLP systems. BERT uses an MLM framework with a transformer as the backbone. In the pre-training stage, BERT randomly replaces tokens with a special token [MASK] and tries to recover corrupted words based on their contextual representations. It also adopts an objective of next-sentence prediction (NSP) to capture the discourse relations between two sentences, which is helpful for sentence-level tasks, such as question answering. Devlin et al. [23] refer to this procedure as a cloze task, according to Ref. [85]. BERT was pre-trained on a combination of the BooksCorpus (800 M words) and English Wikipedia (2500 M words), and achieved great improvements on 17 NLP tasks, attaining a level even better than a human performance on some of the downstream tasks. However, BERT's shortcomings are also obvious: Because the [MASK] token does not appear in real data during fine-tuning, it creates a mismatch between pre-training and fine-tuning. To amend this discrepancy, BERT uses a novel method to mask tokens: Among the 15% of the random positions that would have to be masked, only 80% are replaced by the [MASK] token, while 10% are kept as the original tokens, and 10% are replaced by random tokens in the training process. This masking strategy causes the model to take more steps to converge, since only 15% of the tokens in the training batch are predicted. Another problem with BERT is that it predicts tokens independently without considering other masked tokens. The model proposed in Ref. [86], a unified encoder–decoder model, tends to solve this problem by blanking out text spans of input sentences and predicting the masked span auto-regressively, which mitigates the independent assumption of masked tokens within the same span in the pre-training of masked language models.

Following the success of BERT, an enormous amount of research effort has gone into MLM. SpanBERT [87] is designed to predict spans of text. It chooses to mask random contiguous spans instead of random tokens, and a span boundary prediction objective is introduced to force the model to predict masked spans according to the structural information of the span boundaries. It also achieves better performance by replacing the NSP objective in BERT with single-sequence training. SpanBERT outperforms BERT on span-related tasks such as question answering and coreference resolution. Like SpanBERT, which uses lexical analysis and chunking tools to locate the span boundary, enhanced representation through knowledge integration (ERNIE) [31] uses a Chinese tokenizer to obtain phrase information and then replaces the random token masking in BERT with the entity or phrase masking. ERNIE also utilizes a named-entity recognition toolkit to identify the entity boundary and randomly masks tokens at the entity level, thus enabling the integration of external knowledge into model pre-training.

2.1.3. Transformer encoder–decoders

Transformer encoder–decoder architecture is dedicated to natural language generation (NLG) tasks. Unlike NLU, which focuses on comprehending texts, NLG aims to generate a coherent, meaningful, and human-like natural language expression according to specific inputs. For example, the goal of machine translation is to generate a sentence in the target language with the same meaning as the given source language input; for text summarization, the goal is to generate a short version of the input document that captures the core meanings and opinions. The critical point is to model two sequences simultaneously—one for the input and the other for the output.

Song et al. [88] proposes Masked Sequence-to-Sequence Learning (MASS) for language generation, in order to pre-train a seq2seq model. The basic idea of MASS is to take a sentence with a masked fragment (i.e., several consecutive tokens) as input and predict the masked fragment conditioned on the encoder representations. In this way, MASS successfully transforms the transformer encoder framework into an autoregressive framework by masking on the source side and predicting on the target side. MASS uses monolingual data from the News Crawl Datasets of Workshop on Machine Translation (WMT) to pre-train the model, and shows substantial improvement on machine translation quality in comparison with models directly trained on annotated data.

Pre-training on both a transformer encoder and a transformer decoder results in a unified model that can simultaneously deal with both language understanding and language generation. One member of this class is the standard transformer encoder–decoder model that does not share unified encoder and decoder components. Bidirectional and Auto-Regressive Transformers (BART) [89] proposes a similar objective as MASS, but differs in that MASS masks a consecutive series of tokens—that is, *n*-grams of the input—while BART corrupts text with an arbitrary noising function—that is, masking/deleting/replacing/exchanging random tokens in different positions. BART can be viewed as a combination of the above two architectures: The random masking strategy on the source side enables the model to deal with NLU tasks, and the overall seq2seq pre-training framework enables the model to be generalized to NLG tasks. Pre-trained on 160 GB data of news, books, stories, and web text, BART achieves comparable results to RoBERTa [90] and new SOTA results on dialogue and abstractive text summarization. Another member of this category unifies the encoder and decoder as identical transformer blocks. Dong et al. [91] and Bao et al. [92] also propose a unified language model pre-training framework for NLU and generation. These studies partition the self-attention matrix into three components: the bidirectional component, the unidirectional component, and the seq2seq component, which respectively stand for unidirectional, bidirectional, and seq2seq language models. Their experiments show performance gains over using a single pre-training objective. Du et al. [86] propose a variant of the model reported in Ref. [91], putting the masked tokens on the right of the unmasked tokens and conducting autoregressive blank filling. Xiao et al. [93] mask multiple segments at different granularities to encourage the decoder to

rely more on the encoder representations, thus enhancing the correlation between the encoder and the decoder. Zhang et al. [94] adopt a different approach: First, a sentence is removed according to the pre-defined importance criteria from an input document, and then the removed sentence is generated based on the remaining context sentences. This strategy performs auto-regression at the sentence level and prompts whole-document understanding and summary-like generation. Experiments on 12 downstream summarization tasks demonstrate SOTA results, showing the effectiveness of the gap-sentence pre-training method.

2.2. Scaling up PTMs

Recent advances in NLP have demonstrated a promising trend toward scaling up PTMs with billions of parameters. OpenAI researchers trained a model called GPT-3, which has 175 billion parameters [26]. GPT-3 achieves strong performance on many NLP datasets, including question answering, machine translation, and three-digit arithmetic. GPT-3 demonstrates that scaling up language models significantly improves task-agnostic and few-shot performances, sometimes even achieving better results than prior SOTA fine-tuning approaches [26]. Although large PTMs are a promising direction, training large-scale PTMs is a challenging task, which requires massive training data and graphics processing unit (GPU) resources. Thus, efficient model training algorithms play a crucial role in scaling up PTMs. The following section introduces the prevalent large-scale PTMs as well as the training methods used to achieve them.

2.2.1. PTMs at scale

Table 1 [24–28,39,95–102] summarizes the mainstream large-scale PTMs. The size of PTMs has become increasingly larger in recent years, ranging from 2.6 billion to even 175 billion parameters. Large-scale pre-trained language models embrace a potpourri of training recipes including exponentially increased trainable parameters, pre-training architectures, knowledge enhancement, language-specific corpora, and different pre-trained tasks to support the billion-level training of PTMs. Although training methods differ among these models, all the PTMs use transformers [9] as the standard backbone due to the latter's efficient parallel computing performance. Since training large-scale models requires massive unsupervised data, research on scaling up PTMs focuses on high-resource languages such as English and Chinese.

According to the different designs used in pre-training architectures, large-scale PTMs can be generally classified into three classes (as in Section 2.1): encoder only, decoder only, and encoder–decoder. The majority of large PTMs leverage the decoder only or the encoder–decoder architecture, whereas only a few large models adopt an encoder-only design. This is because encoder-only models cannot perform well on generation tasks, such as text summarization and dialogue generation, while decoder-only models that are designed for language generation can shed light on not only NLG but also language understanding tasks via prevalent prompting techniques such as GPT-3 [26].

- Encoder-only models at scale employ a bidirectional transformer encoder to learn contextual representations; they demonstrate impressive performance on NLU tasks. For example, DeBERTa$_{1.5B}$ [24], which consists of 48 transformer layers with 1.5 billion parameters, applied a disentangled attention mechanism and enhanced the mask decoder to surpass human performance on the SuperGLUE [30] benchmark. Since a bidirectional nature makes the model unable to be directly used in NLG tasks, DeBERTa trained another version of a unified encoder–decoder to adapt to NLG tasks.
- Decoder-only models use transformer decoders by applying autoregressive masks to prevent the current token from attending to future tokens. Examples include GPT-3 [26], CPM [27], and PanGu-α [28]. This line of PTMs aims at

Table 1
Summary of large-scale pre-trained language models.

Model	Number of parameters	Model architecture	Knowledge learning	Language	Pre-training data	Training strategy	Training platform	Reference
DeBERTa$_{1.5B}$	1.5 billion	Encoder only	—	English	English data (78 GB)	—	PyTorch	[24]
T5	11 billion	Encoder–decoder (seq2seq)	—	English	C4 (750 GB)	Model/data parallelism	TensorFlow	[25]
GPT-3	175 billion	Decoder only	—	English	Cleaned CommonCrawl, WebText	Model parallelism	—	[26]
CPM	2.6 billion	Decoder only	—	Chinese	Chinese corpus (100 GB)	—	PyTorch	[27]
PanGu-α	200 billion	Decoder only	—	Chinese	Chinese data (1.1 TB, 250 billion tokens)	MindSpore auto-parallel	MindSpore	[28]
ERNIE 3.0	10 billion	Encoder–decoder (unified)	√	Chinese, English	Chinese data (4 TB), English data	Model/pipeline/tensor parallelism	PaddlePaddle	[39]
Turing-NLG	17 billion	Decoder only	—	English	English data	DeepSpeed/ZeRO	—	[95]
HyperCLOVA	204 billion	Decoder only	—	Korean	Korean data	—	—	[96]
CPM-2	11 billion	Encoder–decoder (seq2seq)	—	Chinese, English	WuDao corpus (2.3 TB Chinese + 300 GB English)	—	PyTorch	[97]
CPM-2-MoE	198 billion	Encoder–decoder (seq2seq)	—	Chinese, English	WuDao corpus (2.3 TB Chinese + 300 GB English)	Mixture of Experts (MoE)	PyTorch	[98]
Switch transformers	1751 billion	Encoder–decoder (seq2seq)	—	English	C4 (750 GB)	MoE	TensorFlow	[99]
Yuan 1.0	245 billion	Encoder–decoder (unified)	—	Chinese	Chinese data (5 TB)	Model/pipeline/tensor parallelism	—	[100]
GLaM	1.2 trillion	Encoder only	—	English	English data (1.6 trillion tokens)	MoE/model parallelism	TensorFlow	[101]
Gopher	280 billion	Decoder only	—	English	English data (10.5 TB)	Model/data parallelism	Jax	[102]

ZeRO: zero redundancy optimizer; MoE: mixture-to-expert.

generating human-like texts. Turing-NLG [95] is a 17-billion-parameter language model that has achieved strong performance in language model benchmarks. GPT-3, with 175 billion parameters, can strikingly write samples that deceive human readers, demonstrating that large-scale language models can dramatically advance few-shot learning scenarios with in-context learning. In addition to English large-scale monolingual PTMs, there are also models for other languages such as Chinese and Korean. CPM [27] (2.6 billion parameters) and PanGu-α [28] (200 billion parameters) are two Chinese variants of GPT-3, while HyperCLOVA [96] is a 204-billion-parameter Korean variant.

- Encoder–decoder models can be further categorized into two classes: ① conventional seq2seq encoder–decoders and ② unified encoder–decoders. Conventional seq2seq encoder–decoders adopt the classic transformer encoder–decoder architecture for pre-training. Recent work includes the T5 [25], the multilingual T5 (mT5) [97], and the large-scale cost-effective pre-trained language model (CPM-2) [98]. T5 [25], which has up to 11 billion parameters, unifies the NLP tasks in one framework by casting the language understanding and generation tasks in a text-to-text manner. As the multilingual variant of T5, mT5 [97], which has up to 13 billion parameters, has extended the monolingual data to 101 human languages and outperformed the previous SOTA results on a variety of multilingual benchmarks. CPM-2 [98], with 11 billion parameters, is a bilingual model trained on Chinese and English, whose mixture-of-expert (MoE) version, denoted as CPM-2-MoE, has 198 billion parameters. This model has demonstrated excellent general language intelligence via fine-tuning and prompting. Another kind of encoder–decoder model is the unified encoder–decoder framework, in which the encoder–decoder architecture shares the same module and applies different mask strategies for MLM and autoregressive language modeling. ERNIE 3.0 [39] jointly learns language understanding and generation by designing two separate heads for understanding and generation, which share a task-agnostic representation. As the third-generation PTMs (with ten billion parameters) in the ERNIE series, ERNIE 3.0 combines the merits of both autoregressive causal language models and autoencoding models to train large-scale knowledge-enhanced PTMs. It has outranked the SOTA performance on a variety of NLP benchmarks, including SuperGLUE [30]. These methods have demonstrated superior performance because they all tend to unify multiple NLP tasks in one model and use different kinds of corpora or knowledge to enhance the performance.

Most of the above-mentioned large-scale models are trained on plain texts without integrating knowledge. Therefore, some researchers have attempted to incorporate knowledge such as linguistic knowledge and world knowledge into PTMs. ERNIE 3.0 pre-trained transformers on massive unstructured texts and knowledge graphs to learn lexical, syntactic, and semantic information. It enriched the PTMs through knowledge integration, phrase masking, and named-entity masking.

The dramatic progress in language PTMs has attracted research interest on multimodal pre-training [72,103–107]. Table 2 [69,103,104,107] lists the details of large-scale multimodal PTMs. DALL-E [69] is a 12-billion variant of GPT-3 that was trained on 250 million English text–image pairs to generate images according to language descriptions, thereby improving the zero-shot learning performance. ERNIE-ViLG [107] uses a unified GPT framework for bidirectional image–text generation, formulating both the image and text generation as autoregressive generative tasks. As a result, it outperforms previous methods on generative tasks such as text-to-image generation and image captioning with a ten-billion parameter model pre-trained on 145 million high-quality Chinese text–image pairs. Moreover, the multi-modality-to-multi-modality multi-task mega-transformer (M6) [104] is a 100-billion-parameter transformer encoder, which is trained on over 1.9 TB images and 292 GB Chinese texts. M6 achieved strong performance in visual question answering, image captioning, and Chinese image–text matching. In addition to their improvements on multi-modal tasks, these models can improve the performance of mono-modal tasks, such as text classification, inference, summarization, and question generation [105]. These results show that multimodal pre-training can leverage multimodal information to enhance both image representation and text representation, which in turn improves the performance of both multimodal tasks and NLP tasks.

2.2.2. Efficient training of large-scale models

The exponential increment of the PTMs' size has posed a great challenge for efficient training due to the limited GPU memory and unaffordable training time. Therefore, it is non-trivial to leverage efficient training techniques to speed up large-scale model training.

2.2.2.1. Dense models. Data parallelism is a simple solution that allocates different data partitions to multiple workers and duplicates identical parameters at all workers. However, it usually suffers from a small per-GPU batch size. Another solution is model parallelism, in which model parameters are partitioned over different workers. However, conventional optimization algorithms require extra memory per parameter to store intermediate states, which hinders the model size from being updated efficiently. Pipeline parallelism combines the merits of both model parallelism and data parallelism to reduce time costs. GPipe [108] uses a novel batch-splitting pipelining algorithm by first partitioning a mini-batch of training samples into smaller micro-batches and then aggregating the gradient update simultaneously at the end. Megatron-LM [109] is an intra-layer model parallel approach for transformer networks, which adds a few synchronization primitives on the self-attention and multi-layer perceptron blocks. PTD-P [110] combines pipeline, tensor, and data parallelism across multi-GPU servers with a novel interleaved pipelining scheduling strategy, increasing the throughput by more than 10%. Recently, Colossal-AI [111] implemented a combination of various data, pipeline, sequence, and multiple tensor parallelism for large-scale model training, which can be a good option for training dense models.

Table 2
Large-scale multimodal PTMs.

Model	Number of parameters	Pre-training paradigm		Pre-training Data	Training parallelism	Training platform	Reference
		Denosing auto-encoder	Causal language model				
DALL-E	12 billion	×	√	250 million English text–image pairs	Mixed-precision training	PyTorch	[69]
CogView	4 billion	×	√	30 million English text–image pairs	—	PyTorch	[103]
M6	100 billion	√	×	1.9 TB images + 292 GB Chinese	MoE	—	[104]
ERNIE-ViLG	10 billion	√	√	145 million Chinese text–image pairs	Mixed-precision training	PaddlePaddle	[107]

2.2.2.2. Sparse models. The sparsely gated MoE model [112] achieved more than 1000 times the increment in model capacity using a sparsely gated combination of multiple expert sub-networks. By leveraging the ensemble mechanism, MoE employs the gated unit to determine which top-*k* sub-networks should be activated for prediction.

Switch transformers [91] have advanced the scale of PTMs with up to trillions of parameters by simplifying the sparse routing and replacing the feed-forward fully connected layers with switch routing, in which each sample is routed to only a single expert.

2.2.2.3. Other efficient training strategies. Recent techniques for memory-efficient optimization include mixed-precision training [113] and memory-efficient adaptive optimization. Mixed-precision training utilizes half-precision floating-point numbers without losing model accuracy, which nearly halves the memory requirements. Other studies have aimed at memory-efficient adaptive optimization. For example, the zero redundancy optimizer (ZeRO) [114], which is the catalyst that powers Turing-NLG, consists of ZeRO-data parallelism (DP) and ZeRO-residual (R) algorithms that aim at reducing the memory footprint of the model states and the residual memory consumption, respectively. First, ZeRO-DP optimizes the optimizer states, gradients, and parameters by performing optimizer state partitioning, adding gradient partitioning, and adding parameter partitioning. Then, ZeRO-R optimizes the residual memory through the removal of activation replication, pre-definition of appropriate temporary buffer size, and proactive memory management.

3. Impact and challenges of PTMs

3.1. Impact of PTMs in NLP

The emergence of PTMs has enabled a significant breakthrough in the field of NLP. Before PTMs, many studies focused on designing specialized models for specific NLP tasks, which usually could not be used for other tasks. For example, Kim [115] proposes the TextCNN model for text classification, and Hochreiter and Schmidhuber [8] propose the LSTM model for language generation. Since their emergence, PTMs have started to serve as foundation models in NLP due to their impressive capabilities in representation learning. This has opened up a new "pre-training then fine-tuning" paradigm for NLP. This paradigm can fully exploit unannotated data to train a foundation model and then fine-tune it with limited task-specific annotated data. Even with limited annotated data, the performance of the downstream NLP tasks is greatly improved. Fig. 2 [23,39,116,117] demonstrates the evolution of SOTA results on five NLP benchmarks from supervised models without pre-training to PTMs such as BERT and ERNIE 3.0. It is evident that PTMs significantly outperform the previous non-PTMs, and the knowledge-enhanced ERNIE 3.0 has steadily exceeded BERT on many NLP tasks. Another important trend is to adopt PTMs to unify almost all NLP tasks. For example, T5 [25] casts both language understanding and generation tasks in a text-to-text manner and tackles all NLP tasks using a seq2seq PTM. Thus, the NLP community has witnessed the emerging trend of task unification.

GPT-3 [26] has shown a promising performance in zero-shot learning or few-shot learning. Along with GPT-3, a new prompt-exploiting training [118] has been proposed to reformulate the task paradigm. Thus, pre-training then prompt tuning has initiated a new trend to better leverage PTMs. Instead of adapting PTMs to downstream tasks with fine-tuning, downstream tasks are predefined as "slot-filling" tasks: Given a human-designed template with slots, let the PTMs learn to fill out these templates. This framework has been proven powerful, as it enables language models to adapt to few-shot or zero-shot scenarios; as a result, it has attracted wide attention in the NLP community. We generally describe the impact of PLMs in the following three aspects: NLU, NLG, and dialogue. For dialogue, PTMs focus on response generation. Here, we take dialogue as a separate category due to its large amount of related work.

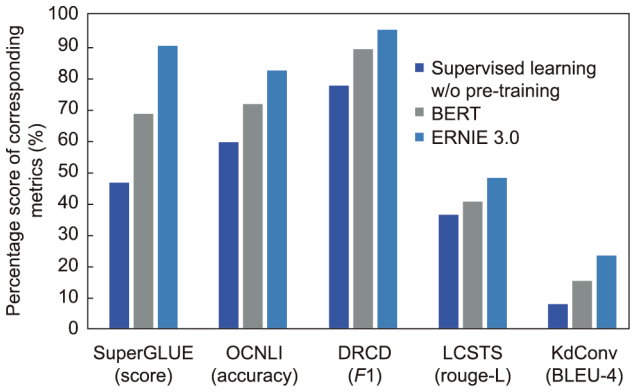

Fig. 2. The evolution shift of representation techniques on various NLP benchmarks. Results are from Refs. [23,39,116,117]. SuperGLUE is an NLU leaderboard consisting of a set of difficult language understanding tasks; an original Chinese natural language inference dataset (OCNLI), a Chinese machine reading comprehension dataset (DRCD), a large scale Chinese short text summarization dataset (LCSTS), and a Chinese multi-domain dialogue dataset towards multi-turn knowledge-driven conversation (KdConv) are evaluation corpora for natural language inference, machine reading comprehension, text summarization, and dialogue generation, respectively. w/o: without.

3.1.1. Natural language understanding

NLU is a broad topic in NLP that contains many tasks, such as named-entity recognition, sentiment analysis, document classification, reading comprehension, semantic matching, natural language inference, and information extraction. Table 3 [39,116,117,119,120] compares the performance of models with and without pre-training techniques on four different NLU tasks. It can be seen that models with pre-training outperform those without pre-training by a clear margin. Thus, PTMs have become the standard backbone in NLU tasks. Numerous researchers have employed PTMs to provide task-agnostic representations and then design task-specific architectures or objectives to enhance the NLU performance. For example, BertGCN [121] combines the representative capacity of BERT and transductive learning from graph convolutional networks to advance its performance of text classification, which increases its accuracy by around 4%.

To compare the performances of the PTMs on NLU tasks, researchers uploaded their results on two benchmarks, GLUE and SuperGLUE. These PTMs now outperform humans on these two leaderboards. In addition, multilingual models such as mBERT [41], XLM [42], mT5 [97], and ERNIE-M [45] use a unified model to represent various languages such that the learned information can be shared among different languages. This technology alleviates the data sparseness problem in low-resource languages and reduces the demand to train specialized language models for each specific language. This new paradigm is changing the focus of research on NLP from designing specialized models for multilingual tasks to studying how PTMs can be used in these tasks.

3.1.2. Natural language generation

NLG tasks, such as text summarization, question generation, and data-to-text generation, are very challenging in NLP. Due to the huge search space, it is difficult for methods before the PTM era, which suffer from insufficient annotation data and limited model parameters, to generate fluent, coherent, and informative text. As shown in Table 4 [94, 122–125], PTMs have played a key

Table 3
SOTA performance with and without pre-training on NLU tasks.

NLU task	Sentiment analysis SST-2 binary classification (accuracy)	Natural language inference OCNLI (*F*1)	Nested named entity recognition GENIA (*F*1)	Machine reading comprehension DRCD (*F*1)
SOTA w/o pre-training	93.2	59.80	74.80	78.03
SOTA w/ pre-training	97.5	82.75	83.75	95.84

Results are from Refs. [39,116,117,119,120]. w/: with; SST-2: Stanford Sentiment Treebank v2; OCNLI: Original Chinese Natural Language Inference; DRCD: Delta Reading Comprehension Dataset.

role in improving the performance of NLG tasks. Large-scale PTMs automatically learn word combinations and sentence expressions from unannotated data, which significantly improves the models' ability in language generation in terms of fluency, coherence, and informativeness. ERNIE-GEN [93] uses an enhanced multi-flow seq2seq pre-training and fine-tuning framework and incorporates a span-by-span generation task to generate consecutive entities, which has achieved new SOTA results on five typical NLG tasks. Researchers and practitioners also pre-train task-specific transformer models on generation tasks, such as MASS [88] and PEGASUS [94]. More specifically, MASS adopts the encoder–decoder framework to reconstruct a sentence fragment, given the remaining part of the sentence, and achieves significant improvements over baselines without pre-training on machine translation. PEGASUS was used to pre-train a large-scale encoder-decoder model with a well-designed pre-training objective, which achieved a SOTA performance on all 12 text-summarization tasks. With the growth of the model size, PTMs gradually show notable ability in creative writing. Models such as GPT-3, HyperCLOVA, and ENRIE 3.0 are capable of generating articles, questions and answers, novels, and program codes via only zero-shot learning. The quality of the generated texts is sometimes comparable with that of human-written texts. For example, humans only achieve 52% accuracy in distinguishing real news from fake news generated by GPT-3.

3.1.3. Dialogue

In the past few years, several representative dialogue-generation models have been pre-trained with human-like conversations collected from social media, including Twitter, Reddit, Weibo, and Baidu Tieba. Based on the general language model GPT-2 [83], DialoGPT [126] has been trained for response generation using Reddit comments. Meena [127] scales up the network to 2.6 billion parameters and employs more social media conversations in the training process, resulting in a significant improvement in response quality. To mitigate undesirable toxic or bias traits in large corpora, Blender [128] further fine-tunes the PTM with human-annotated datasets and emphasizes the desirable conversational skills of engagingness, empathy, and personality. In addition, to alleviate the safe-response problem in open-domain chitchat, PLATO [129] encodes the discrete latent variable into transformers for diverse response generation. Moreover, PLATO-2 [130] further scales up PLATO via curriculum learning for both Chinese and English response generation. The Ninth Dialog System Technology Challenge (DSTC-9) [131] revealed that PLATO-2 delivers a superior performance in multiple conversational tasks, including open-domain chitchat, knowledge-grounded dialogue, and task-oriented conversation. Recently, PLATO-XL [132] was scaled up to 11 billion parameters, with multi-party-aware pre-training being carried out to better distinguish roles in social media conversations. Other Chinese dialogue PTMs that have been developed on a modest scale include Cdial-GPT [133], ProphetNet-X [134], and EVA [135].

With these large-scale dialogue PTMs, some of the problems that plague traditional end-to-end neural approaches [136,137] are alleviated significantly, including deficiencies in response fluency and context relevance. Moreover, in comparison with existing chatbots that rely on complex frameworks, such as Mitsuku [138] and XiaoIce [139], these dialogue PTMs demonstrate superior performance in multi-turn conversations, especially in terms of engagingness and humanness.

3.2. Key research challenges

Although PTMs have significantly improved the performance of NLP tasks, there are still some key challenges for PTM applications, such as interpretability, robustness, reasoning capability, and the deployment of large-scale PTMs. This section describes these challenges in the hope that additional future efforts can be devoted to these directions.

3.2.1. Deployability

One trend in PTMs is the substantial increase in capacity. Since the release of GPT [22] and BERT [23], PTMs have scaled exponentially with respect to both the number of parameters and the size of the pre-training data. For example, the largest version of GPT-3 [26] requires a total training computation of 3.64×10^3 petaflop-days, resulting in a total number of around 3.14×10^{23} flops and costing millions of dollars. The rapid growth in model size raises concerns regarding the tradeoff between scale and deployability. Two types of strategy have been proposed to tackle this issue: ① Large-scale PTMs are only used as the foundation model via application programming interface (API) calls, similar to the way in which the GPT-3 model is used. This strategy enables the efficient use of PTMs and evades model deployment on each device, but significantly limits the model's application scope. ② Large models are compressed to smaller ones [140] for potential deployment. Typical compressing techniques include model compression

Table 4
SOTA performance with and without pre-training on NLG tasks.

NLG task	Text summarization ESLC (ROUGE-L)	Dialogue generation KdConv-film (BLEU-4)	Question generation SQuAD 1.1 (BLEU-4)	Data-to-text generation WebNLG (BLEU)
SOTA w/o pre-training	23.44	5.40	15.87	63.69
SOTA w/ pre-training	36.51	74.44	25.41	66.07

Results are from Refs. [94,122–125]. ESLC: English Skills Learning Center; BLEU: bilingual evaluation understudy; ROUGE-L: recall-oriented understudy for gisting evaluation-longest common subsequence.

and knowledge distillation. Unfortunately, existing compressing techniques are unable to compress super-large PTMs (e.g., GPT-3) to a suitable size for deployment on a single GPU or terminal device such as a laptop or cell phone. Advanced research in model compression is thus imperative in order to make large PTMs available to more users. Another promising direction is to use parameter-efficient techniques, such as prompt tuning [141–146], to reduce the memory budget of deployment; this remains as a large area for further exploration.

3.2.2. Model trustworthiness

Another challenge of PTMs is their trustworthiness, which mainly involves their interpretability [147] and robustness [148]. Although PTMs have achieved SOTA performances across various tasks, how they make decisions are sometimes obscure to humans, which makes PTM models difficult to be applied in fields where model interpretability is essential, such as health-care and law [149]. Consequently, there is a growing interest in interpreting deep neural models [150]. In particular, many studies aim to understand what PTMs have learned in their representations [151].

Some studies have been published on the trustworthiness of deep neural models. These include: linguistic structural analyses on PTMs [152], which aim to analyze the linguistic knowledge that is learned by pre-trained language models and to understand the reason for their success; model behavioral analyses [153], which evaluate model robustness and reliability with multiple test sets; and *post-hoc* explanation analyses [154], which aim to provide understandable explanations for the predictions of deep neural models.

Despite the research that has already been done in this field, the following challenges must be addressed in order to build trustworthy systems: ① general interpretation methods for NLP tasks (existing interpretation methods are designed for classification tasks); ② causal analysis between model prediction and learned knowledge or extracted explanations; and ③ a comprehensive evaluation platform for interpretability, including evaluation data and metrics.

3.2.3. Commonsense knowledge and reasoning

Large-scale PTMs have been found to encode some commonsense knowledge [155]. Nevertheless, appropriate probing tasks need to be designed in order to mine the commonsense knowledge learned in PTMs—such as formulating a relational knowledge-extraction task as the completion of fill-in-the-blank statements—so as to examine the knowledge-learning ability of PTMs [156]. Although PTMs learn some knowledge from texts, there is still a large amount of knowledge that cannot be obtained from texts alone. One possible direction is to have models learn this kind of knowledge from both visual inputs and text inputs.

In addition to commonsense knowledge, other studies are questioning whether PTMs are endowed with reasoning abilities. For example, Talmor et al. [157] design different tasks to evaluate the reasoning abilities of PTMs. The researchers disentangle pre-training from fine-tuning and find that the reasoning capabilities are poor for most PTMs, revealing that existing PTMs lack the ability to reason. To alleviate this problem, one possible direction could be to integrate prior knowledge into the PTMs in order to guide the models to learn reasoning rules implicitly.

3.2.4. Model security

One severe issue with PTMs is their vulnerability to adversarial examples, which can mislead the model into producing a specific wrong prediction when perturbations are injected into the input [158]. This susceptibility exposes PTMs to safety concerns: The models can be easily attacked with adversarial patterns by third parties, resulting in irreparable loss in real-world applications. In addition to adversarial attacks, another form of attack—namely, backdoor attacks—is a threat to PTMs. Unlike adversarial attacks, which usually act during the inference process of a neural model, backdoor attacks hack the model during training [159]. If a model is deliberately trained on backdoor data, it will be extremely dangerous for users to use this model in applications involving privacy and security concerns. Future work could aim to improve the robustness of PTMs toward adversarial attacks. To deal with backdoor attacks, a model should be able to detect in the input the triggers that can activate the backdoor attack and remove the triggers, thus enhancing model security.

4. Applications of PTMs

4.1. Platforms and toolkits for applications

Due to their universality, PTMs have become foundation models in NLP. Many researchers have developed a series of open-source toolkits and platforms to make better use of PTMs. These toolkits and platforms usually contain various PTMs, fine-turning tools, and model-compression tools.

4.1.1. Toolkits

When researchers propose a new pre-trained language model, they often open-source a corresponding toolkit for developers. Such toolkits usually provide codes for downstream task development based on the specific model, and therefore lack generality. Typical toolkits include google-research/bert [160], PaddlePaddle/ERNIE [161], and PCL-Platform.Intelligence/PanGu-α [162]. These toolkits provide a series of open-sourced PTMs, such as BERT, ERNIE, and PanGu-α, along with source code and training data. For example, the ERNIE toolkit provides not only the source code, training data, and PTM of ERNIE but also a couple of enhanced ERNIE series models, such as ERNIE-Doc [163] and ERNIE-ViL [70]. In order to deploy the ERNIE model to online service, the ERNIE toolkit also provides a model-compression tool.

With the intensive publish of PTMs, knowing how to use these models in a unified toolkit has become an urgent need. Given this background, toolkits for general NLP applications have been developed. Typical toolkits include HuggingFace/Transformers [164], Fairseq [165], and PaddleNLP [166]. PTMs are integrated in a user-friendly way into such general-purpose toolkits. Taking HuggingFace as an example, this toolkit integrates the codes for different kinds of PTMs and codes for downstream application developments, including classification, generation, summarization, translation, question answering, and so forth.

4.1.2. Platforms

Besides toolkits, platforms provide users with PTM services for customization. These platforms can provide facilities for developers to build models and deploy them to online services. For example, Baidu Wenxin [167] is a platform to facilitate the use of PTMs. This platform meets the needs of both experienced developers and junior developers. It enables developers to easily build their models with data and model configuration only. It also provides experienced developers with toolkits to train their models that are tailored for applications. Other platforms such as AliceMind [168] provide similar services with no significant differences. OpenAI API [169] is another kind of platform that is used to develop applications based only on PTMs. OpenAI API is based on GPT-3 [26]; it provides specific high-level functions, such as English-to-French translation, grammar correction,

question answering, advertisement generation, and product-name generation.

4.2. Applications

PTMs have been widely deployed in real applications, including document intelligence, content creation, virtual assistant, and intelligent search engines. Below, we describe how PTMs are applied in each field.

4.2.1. Document intelligence

One widely studied application for PTMs is document intelligence, which includes sentiment analysis, news classification, anti-spam detection, and information extraction. Sentiment analysis is widely used to identify sentiment polarity, such as public opinion, for market research, brand reputation analysis, and social media influence. Garg and Chatterjee [170] propose analyzing the sentiment of Twitter feeds using a PTM and classifying them into three categories: positive, negative, and neutral. AlQahtani [171] proposes analyzing customer reviews on products by combining data-mining techniques with PTMs. Recently, Singh et al. [172] analyzed public sentiment on the impact of the coronavirus on social life using a PTM. Chen and Sokolova [173] propose analyzing the sentiments in the coronavirus disease 2019 (COVID-19)-related messages in a popular social media platform, where users share their stories to seek support from other users, especially during the COVID-19 pandemic. Experimental results show that PTMs can achieve significant performance gain in classifying sentiment polarities, demonstrating the effectiveness of PTMs.

News classification and anti-spam detection can also be modeled as classification tasks. Ding et al. [163] apply PTMs to classify news into extreme left-wing or right-wing standpoints. Liu et al. [174] propose classifying the papers published in Arxiv.org into 11 categories, including math, computer science, and so forth. Jwa et al. [175] use BERT to detect fake news by analyzing the relationship between the headline and the body text in news.

Document information extraction is widely used in industry. Many AI cloud services contain tools for information extraction [176], such as Google AI Cloud, Baidu AI Cloud, and Alibaba AI Cloud. Among these services, Baidu has built a PTM-based platform, TextMind, for document information-extraction applications, including receipt analysis for expense reimbursements, information extraction from resumes, financial statement analysis, contract analysis, and legal judgment analysis. One of the world's largest online home retailers, Wayfair, also applies BERT to extract information from customer messages.

Document image understanding is another important research topic in document intelligence for automatically reading, understanding, and analyzing business documents. A series of multimodal document PTMs [177] has been proposed to jointly model interactions between text, image, and layout information in business documents for many document image understanding tasks, such as receipt understanding, document image classification, and document information extraction. Applica proposes a solution to take into consideration layout, graphics, and text in order to enable the extraction of precise answers for complex business processes in financial services, insurance services, life sciences, and so on.

4.2.2. Content creation

Content creation tasks are usually designed to verify the performance of recently proposed large-scale models [22]. For example, Narrativa applies GPT-2 for content automation from just a few words provided by customers and generates high-quality advertisement content [178]. GPT-2 has demonstrated its ability to generate content for e-commerce in order to relieve humans from laborious tasks. Microsoft has also demonstrated that the pre-trained generation model Turing-NLG is beneficial for autosuggest recommendations [95]. Moreover, many researchers have built various demo applications based on GPT-3, including applications for ad generation, AI copywriting, book writing, code generation, customer service, and so forth. As for visual content creation, pre-trained multimodal generative models such as DALL-E [69], CogView [103], and ERNIE-ViLG [107] have greatly improved the quality and fidelity of generated images. The results from CogView have demonstrated this model's capability to generate high-quality images in a single domain such as industrial fashion design, so this model has been deployed in online fashion production.

In addition to these industrial applications, researchers have shown the potential ability of PTMs for creative writing, including poem generation [179], lyrics generation [27], e-mail auto completion [180], to-do generation [181], auto-completion for sentences and paragraphs, and even a long novel generation [22]. Although PTMs exhibit strong generative capabilities, an increasing number of concerns have arisen regarding generative models, including privacy and copyright.

4.2.3. Virtual assistants

Virtual assistants are adopted in many applications nowadays. Typical applications include smart speakers, such as Alexa [182] from Amazon and Xiaodu [129] from Baidu. Such applications have used PTMs and have shown that PTMs can provide excellent language understanding ability for spoken language and voice recognition [183] in smart speakers. With the benefit brought by PTMs, these smart speakers can respond to weather forecast queries, sing songs on demand, and vocally control smart home devices. Moreover, smart speakers can chat with humans on a broad range of topics and thus establish a closer and more stable relationship between users and the system. In addition to the usage of PTMs in smart speakers, PTMs have been deployed in mobile-phone-based virtual assistants, such as Siri and Google Assistant. For example, NDTV [184] proposes that PTMs can improve the interaction quality, while Vincent [185] proposes that PTMs can be used in intelligent customer service robots to recognize customer sentiments.

As PTMs are applied more and more widely in virtual assistants, the responses generated by chatting bots are becoming more human-like. For example, Microsoft has proposed a PLM-based model called DialoGPT that learns from the comment history of Reddit and can fluently reply to users. Google has also suggested the use of PLMs to develop a chatbot application that can "chat about anything" [127]. To make the robots more human-like, Facebook applied PLM to a series of dialogue chatbots named Blender and Blender 2.0 [128]. Shortly afterwards, Baidu proposed PLATO-XL [132], a PLM-based model, to further push the performance of a chatbot and reach the SOTA in terms of both human evaluation and automatic evaluation metrics. Thanks to the performance improvement brought by PTMs, these applications can be very robust in interactions with users [186].

4.2.4. Intelligent search

Aside from the applications mentioned above, PTMs are widely used in search engines. Google has already applied PTMs in its Google Search and achieved significant improvements [187]. Baidu has also applied PTMs, ERNIE 2.0 [188] and ERNIE 3.0 [39], as its backbone to support semantic matching by encoding text into dense representations for better retrieval performance in Baidu Search [189]. Facebook [190] has revealed a unified embedding framework for personalized systems and noted that their future work will contain PTMs.

To address the surging demand for multimedia content searches, the performance of image and video search engines can be enhanced through the utilization of multimodal PTMs. For

example, WenLan [106] developed two real-world applications based on image–text matching, thereby demonstrating the power of multimodal pre-training.

To further improve the performance of search engines, researchers have recently paid an increasing amount of attention to multilingual search engine models. Multilingual models are pre-trained with a multilingual corpus to learn cross-language information [191]. The most significant advantage of multilingual models is their transferability across languages, which improves their performance on low-resource languages.

5. Conclusions and future work

PTMs can fully exploit unannotated data for self-supervised learning and have become the foundation models in NLP, significantly improving the performance of downstream NLP tasks. The emergence of PTMs opens up a new "pre-training then fine-tuning" paradigm for NLP. With the increase of model parameters, PTMs show promising performance in zero-shot learning or few-shot learning. Their success in NLP is triggering more research devoted to PTMs in other fields such as computer vision, speech processing, and multimodal understanding and generation, revealing their potential to act as foundation models in these fields.

Despite the dramatic success of PTMs in NLP, there is still a long way to go to achieve artificial general intelligence. First, PTMs are black boxes that are poorly understood. Their interpretability and robustness have yet to be explored due to the nonlinearity of transformer models. Thus, it is difficult to use PTMs to make reliable decisions and reasoning before we fully understand their principles. It is worth devoting a great deal of effort to researching the uncertainty of PTMs. Furthermore, current multimodal and multilingual pre-training [192] is still in the early stage. Unifying multimodal and multilingual pre-training will emerge as an exciting trend for further exploration, which may improve the performance of these low-resource tasks. Another promising direction is to incorporate prior knowledge into PTMs to improve their reasoning abilities and efficiency. Existing work on knowledge pre-training, such as K-BERT [33] and ERNIE 3.0 [39], has injected knowledge triplets into pre-training or fine-tuning. However, PTMs have demonstrated limited capability for commonsense awareness and reasoning, which require further improvement. Although large-scale PTMs have demonstrated strong generalization capabilities, efficiently deploying them is still an open question. For applications that require low latency, model compression of PTMs remains a promising direction. Existing model-compression methods consist of distillation [193], pruning [194], quantization [195], and so forth. However, how to efficiently build large-scale PTMs with a deployable inference time remains an ongoing challenge. In addition, designing more efficient architecture in place of or to improve transformers remains an open problem.

In summary, there is still a long way to go for PTMs to be able to make reliable decisions and carry out reliable planning, which are essential elements of AI. More efficient and powerful neural networks need to be proposed and developed. Fortunately, the use of PTMs in real applications continues to provide an increased amount of data and address new challenges, potentially promoting the rapid development of new pre-trained methods.

Compliance with ethics guidelines

Haifeng Wang, Jiwei Li, Hua Wu, Eduard Hovy, and Yu Sun declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Bahl LR, Jelinek F, Mercer RL. A maximum likelihood approach to continuous speech recognition. IEEE Trans Pattern Anal Mach Intell 1983;PAMI-5(2):179–90.
[2] Thrun S, Pratt L. Learning to learn. Cham: Springer; 1998.
[3] Nadas A. Estimation of probabilities in the language model of the IBM speech recognition system. IEEE Trans Acoust Speech Signal Process 1984;32(4):859–61.
[4] Chen SF, Goodman J. An empirical study of smoothing techniques for language modeling. Comput Speech Lang 1999;13(4):359–94.
[5] Bengio Y, Ducharme R, Vincent P, Janvin C. A neural probabilistic language model. J Mach Learn Res 2003;3:1137–55.
[6] Sundermeyer M, Schlüter R, Ney H. LSTM neural networks for language modeling. In: Proceedings of the 13th Annual Conference of the International Speech Communication Association (Interspeech 2012); 2012 Sep 9–13; Portland, OR, USA. 2012. p. 194–7.
[7] Mikolov T, Zweig G. Context dependent recurrent neural network language model. In: Proceedings of 2012 IEEE Spoken Language Technology Workshop (SLT); 2012 Dec 2–5; Miami, FL, USA. 2012. p. 234–9.
[8] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[9] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017); 2017 Dec 4–9; Long Beach, CA, USA. 2017. p. 5998–6008.
[10] Shazeer N, Cheng Y, Parmar N, Tran D, Vaswani A, Koanantakool P, et al. Mesh-TensorFlow: deep learning for supercomputers. In: Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS 2018); 2018 Dec 3–8; Montréal, QC, Canada; 2018.
[11] Dai Z, Yang Z, Yang Y, Carbonell J, Le QV, Salakhutdinov R. Transformer-XL: attentive language models beyond a fixed-length context. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019); 2019 Jul 28–Aug 2; Florence, Italy. 2019. p. 2978–88.
[12] Beltagy I, Peters ME, Cohan A. Longformer: the long-document transformer. 2020. arXiv:2004.05150.
[13] Press O, Smith NA, Lewis M. Shortformer: better language modeling using shorter inputs. 2020. arXiv:2012.15832.
[14] Press O, Smith NA, Levy O. Improving transformer models by reordering their sublayers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020); 2020 Jul 5–10; online. 2020. p. 2996–3005.
[15] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Proceedings of the 27th Conference on Neural Information Processing Systems (NIPS 2013); 2013 Dec 5–10; Lake Tahoe, NV, USA. 2013. p. 3111–9.
[16] Pennington J, Socher R, Manning CD. GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29; Doha, Qatar; 2014. p.1532–43.
[17] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. J Mach Learn Res 2011;12:2493–537.
[18] Xiong C, Zhong V, Socher R. DCN+: mixed objective and deep residual coattention for question answering. In: Proceedings of the 6th International Conference on Learning Representations (ICLR 2018); 2018 Apr 30–May 3; Vancouver, BC, Canada; 2018.
[19] Dai AM, Le QV. Semi-supervised sequence learning. In: Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2015); 2015 Dec 7–12; Montréal, QC, Canada. 2015. p. 3079–87.
[20] McCann B, Bradbury J, Xiong C, Socher R. Learned in translation: contextualized word vectors. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017); 2017 Dec 4–9; Long Beach, CA, USA; 2017.
[21] Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2018 Jun 1–6; New Orleans, LA, USA; 2018. p. 2227–37.
[22] Radford A, Narasimhan K, Salimans T, Sutskever I. Improving language understanding by generative pre-training. San Francisco: OpenAI; 2018.
[23] Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2019 Jun 2–7; Minneapolis, MN, USA; 2019. p. 4171–86.
[24] He P, Liu X, Gao J, Chen W. DeBERTa: decoding-enhanced BERT with disentangled attention. In: Proceedings of the 9th International Conference on Learning Representations (ICLR 2021); 2021 May 3–7; Vienna, Austria; 2021.
[25] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res 2019;21(140):1–67.
[26] Brown TB, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. In: Proceedings of the 34th Conference on

Neural Information Processing Systems (NeurIPS 2020); 2020 Dec 7–12; online. 2020. p. 1877–901.
[27] Zhang Z, Han X, Zhou H, Ke P, Gu Y, Ye D, et al. CPM: a large-scale generative Chinese pre-trained language model. AI Open 2021;2:93–9.
[28] Zeng W, Ren X, Su T, Wang H, Liao Y, Wang Z, et al. PanGu-α: large-scale autoregressive pretrained Chinese language models with auto-parallel computation. 2021. arXiv:2104.12369.
[29] Wang S, Sun Y, Xiang Y, Wu Z, Ding S, Gong W, et al. ERNIE 3.0 Titan: exploring larger-scale knowledge enhanced pre-training for language understanding and generation. 2021. arXiv:2112.12731.
[30] Wang A, Pruksachatkun Y, Nangia N, Singh A, Michael J, Hill F, et al. SuperGLUE: a stickier benchmark for general-purpose language understanding systems. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019); 2019 Dec 9–14; Vancouver, BC, Canada. 2019. p. 3266–80.
[31] Sun Y, Wang S, Li Y, Feng S, Chen X, Zhang H, et al. ERNIE: enhanced representation through knowledge integration. 2019. arXiv:1904.09223.
[32] Xiong W, Du J, Wang WY, Stoyanov V. Pretrained encyclopedia: weakly supervised knowledge-pretrained language model. In: Proceedings of the 8th International Conference on Learning Representations (ICLR 2020); 2020 Apr 26–30; Addis Ababa, Ethiopia; 2020.
[33] Liu W, Zhou P, Zhao Z, Wang Z, Ju Q, Deng H, et al. K-BERT: enabling language representation with knowledge graph. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence; 2020 Feb 7–12; New York City, NY, USA. Palo Alto: AAAI Press; 2020. p. 2901–8.
[34] Sun T, Shao Y, Qiu X, Guo Q, Hu Y, Huang X, et al. CoLAKE: contextualized language and knowledge embedding. In: Proceedings of the 28th International Conference on Computational Linguistics; 2020 Dec 8–13; online. 2020. p. 3660–70.
[35] Zhang Z, Han X, Liu Z, Jiang X, Sun M, Liu Q. ERNIE: enhanced language representation with informative entities. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019); 2019 Jul 28–Aug 2; Florence, Italy. 2019. p. 1441–51.
[36] Peters ME, Neumann M, Logan IV RL, Schwartz R, Joshi V, Singh S, et al. Knowledge enhanced contextual word representations. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019 Nov 3–7; Hong Kong, China. 2019. p. 43–54.
[37] Levine Y, Lenz B, Dagan O, Ram O, Padnos D, Sharir O, et al. SenseBERT: driving some sense into BERT. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020); 2020 Jul 5–10; online. 2020. p. 4656–67.
[38] Wang X, Gao T, Zhu Z, Zhang Z, Liu Z, Li J, et al. KEPLER: a unified model for knowledge embedding and pre-trained language representation. Trans Assoc Comput Linguist 2021;9:176–94.
[39] Sun Y, Wang S, Feng S, Ding S, Pang C, Shang J, et al. ERNIE 3.0: large-scale knowledge enhanced pre-training for language understanding and generation. 2021. arXiv:2107.02137.
[40] Wang R, Tang D, Duan N, Wei Z, Huang X, Ji J, et al. K-Adapter: infusing knowledge into pre-trained models with adapters. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021); 2021 Aug 1–6; online. 2021. p. 1405–18.
[41] Wu S, Dredze M. Beto, Bentz, Becas: the surprising cross-lingual effectiveness of BERT. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019 Nov 3–7; Hong Kong, China. 2019. p. 833–44.
[42] Conneau A, Lample G. Cross-lingual language model pretraining. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019); 2019 Dec 8–14; Vancouver, BC, Canada. 2019. p. 7057–67.
[43] Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, et al. Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020); 2020 Jul 5–10; online. 2020. p. 8440–51.
[44] Chi Z, Dong L, Wei F, Yang N, Singhal S, Wang W, et al. InfoXLM: an information-theoretic framework for cross-lingual language model pre-training. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2021 Jun 6–11; online. 2021. p. 3576–88.
[45] Ouyang X, Wang S, Pang C, Sun Y, Tian H, Wu H, et al. ERNIE-M: enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2021 Nov 7–11; online. 2021. p. 27–38.
[46] Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, et al. DeCAF: a deep convolutional activation feature for generic visual recognition. In: Proceedings of the 31st International Conference on Machine Learning (ICML 2014); 2014 Jun 21–26; Beijing, China. 2014. p. 647–55.
[47] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2014 Jun 23–28; Columbus, OH, USA. 2014. p. 580–7.
[48] Sun C, Shrivastava A, Singh S, Gupta A. Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22–29; Venice, Italy. 2017. p. 843–52.
[49] Schneider S, Baevski A, Collobert R, Auli M. Wav2vec: unsupervised pre-training for speech recognition. In: Proceedings of the 20th Annual Conference of the International Speech Communication Association (InterSpeech 2019); 2019 Sep 15–19; Graz, Austria. 2019. p. 3465–9.
[50] Deng J, Dong W, Socher R, Li LJ, Li K, Li FF. ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2009 Jun 20–25; Miami, FL, USA. 2009. p. 248–55.
[51] Mahajan D, Girshick R, Ramanathan V, He K, Paluri M, Li Y, et al. Exploring the limits of weakly supervised pretraining. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018 Sep 8–14; Munich, Germany. 2018. p. 181–96.
[52] Zhai X, Kolesnikov A, Houlsby N, Beyer L. Scaling vision transformers. 2021. arXiv:2106.04560.
[53] Doersch C, Gupta A, Efros AA. Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2015 Dec 7–13; Santiago, Chile. 2015. p. 1422–30.
[54] Noroozi M, Favaro P. Unsupervised learning of visual representations by solving jigsaw puzzles. In: Proceedings of the European Conference on Computer Vision (ECCV); 2016 Oct 8–16; Amsterdam, The Netherlands. 2016. p. 69–84.
[55] Misra I, van der Maaten L. Self-supervised learning of pretext-invariant representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 14–19; online. 2020. p. 6707–17.
[56] Gidaris S, Singh P, Komodakis N. Unsupervised representation learning by predicting image rotations. In: Proceedings of the 6th International Conference on Learning Representations (ICLR 2018); 2018 Apr 30–May 3; Vancouver, BC, Canada; 2018.
[57] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16 × 16 words: transformers for image recognition at scale. In: Proceedings of the 9th International Conference on Learning Representations (ICLR 2021); 2021 May 3–7; Vienna, Austria; 2021.
[58] Van den Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding. 2018. arXiv:1807.03748.
[59] He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 14–19; online. 2020. p. 9729–38.
[60] Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning (ICML 2020); 2020 Jul 12–18; online. 2020. p. 1597–607.
[61] Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning (ICML 2021); 2021 Jul 18–24; online. 2021. p. 8748–63.
[62] Jia C, Yang Y, Xia Y, Chen YT, Parekh Z, Pham H, et al. Scaling up visual and vision–language representation learning with noisy text supervision. In: Proceedings of the 38th International Conference on Machine Learning (ICML 2021); 2021 Jul 18–24; online. 2021. p. 4904–16.
[63] Lu J, Batra D, Parikh D, Lee S. ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019); 2019 Dec 9–14; Vancouver, BC, Canada. 2019. p. 13–23.
[64] Tan H, Bansal M. LXMERT: learning cross-modality encoder representations from transformers. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019 Nov 3–7; Hong Kong, China; 2019.
[65] Li LH, Yatskar M, Yin D, Hsieh CJ, Chang KW. VisualBERT: a simple and performant baseline for vision and language. 2019. arXiv:1908.03557.
[66] Sun C, Myers A, Vondrick C, Murphy K, Schmid C. VideoBERT: a joint model for video and language representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. 2019. p. 7464–73.
[67] Sun C, Baradel F, Murphy K, Schmid C. Learning video representations using contrastive bidirectional transformer. 2019. arXiv:1906.05743.
[68] Chuang YS, Liu CL, Lee H, Lee L. SpeechBERT: an audio-and-text jointly learned language model for end-to-end spoken question answering. In: Proceedings of the 21st Annual Conference of the International Speech Communication Association (Interspeech 2020); 2020 Oct 25–29; Shanghai, China. 2020. p. 4168–72.
[69] Ramesh A, Pavlov M, Goh G, Gray S, Voss C, Radford A, et al. Zero-shot text-to-image generation. In: Proceedings of the 38th International Conference on Machine Learning (ICML 2021); 2021 Jul 18–24; online. 2021. p. 8821–31.
[70] Yu F, Tang J, Yin W, Sun Y, Tian H, Wu H, et al. ERNIE-ViL: knowledge enhanced vision–language representations through scene graphs. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence; 2021 Feb 2–9; online. Palo Alto: AAAI Press; 2021. p. 3208–16.
[71] Gan Z, Chen YC, Li L, Zhu C, Cheng Y, Liu J. Large-scale adversarial training for vision-and-language representation learning. In: Proceedings of the 34th

Conference on Neural Information Processing Systems (NeurIPS 2020); 2020 Dec 7–12; online. 2020. p. 6616-28.
[72] Cho J, Lei J, Tan H, Bansal M. Unifying vision-and-language tasks via text generation. In: Proceedings of the 38th International Conference on Machine Learning (ICML 2021); 2021 Jul 18–24; online. 2021. p. 1931–42.
[73] Kalyan KS, Rajasekharan A, Sangeetha S. AMMUS: a survey of transformer-based pretrained models in natural language processing. 2021. arXiv:2108.05542.
[74] Kaliyar RK. A multi-layer bidirectional transformer encoder for pre-trained word embedding: a survey of BERT. In: Proceedings of 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence); 2020 Jan 29–31; Noida, India. 2020. p. 336–40.
[75] Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G. Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. 2021. arXiv:2107.13586.
[76] Min B, Ross H, Sulem E, Veyseh APB, Nguyen TH, Sainz O, et al. Recent advances in natural language processing via large pre-trained language models: a survey. 2021. arXiv:2111.01243.
[77] Li J, Tang T, Zhao WX, Wen JR. Pretrained language models for text generation: a survey. 2021. arXiv:2105.10311.
[78] Zaib M, Sheng QZ, Zhang W. A short survey of pre-trained language models for conversational AI—a new age in NLP. In: Proceedings of the Australasian Computer Science Week Multiconference (ACSW'20); 2020 Feb 3–7; Melbourne, VIC, Australia. 2020.
[79] Ramponi A, Plank B. Neural unsupervised domain adaptation in NLP—a survey. In: Proceedings of the 28th International Conference on Computational Linguistics; 2020 Dec 8–13; onine. 2020. p. 6838–55.
[80] Qiu XP, Sun TX, Xu YG, Shao YF, Dai N, Huang XJ. Pre-trained models for natural language processing: a survey. Sci China Technol Sci 2020;63 (10):1872–97.
[81] Bommasani R, Hudson DA, Adeli E, Altman R, Arora S, von Arx S, et al. On the opportunities and risks of foundation models. 2021. arXiv:2108.07258.
[82] Han X, Zhang Z, Ding N, Gu Y, Liu X, Huo Y, et al. Pre-trained models: past, present and future. AI Open 2021;2:225–50.
[83] Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. San Francisco: OpenAI; 2019.
[84] Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV. XLNet: generalized autoregressive pretraining for language understanding. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019); 2019 Dec 9–14; Vancouver, BC, Canada. 2019. p. 5754–64.
[85] Taylor WL. "Cloze procedure": a new tool for measuring readability. J Mass Commun Q 1953;30(4):415–33.
[86] Du Z, Qian Y, Liu X, Ding M, Qiu J, Yang Z, et al. GLM: general language model pretraining with autoregressive blank infilling. 2021. arXiv:2103.10360.
[87] Joshi M, Chen D, Liu Y, Weld DS, Zettlemoyer L, Levy O. SpanBERT: improving pre-training by representing and predicting spans. Trans Assoc Comput Linguist 2020;8:64–77.
[88] Song K, Tan X, Qin T, Lu J, Liu TY. MASS: masked sequence to sequence pre-training for language generation. In: Proceedings of the 36th International Conference on Machine Learning (ICML 2019); 2019 Jun 9–15; Long Beach, CA, USA. 2019. p. 5926–36.
[89] Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, et al. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020); 2020 Jul 5–10; online. 2020. p. 7871–80.
[90] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: a robustly optimized BERT pretraining approach. 2019. arXiv:1907.11692.
[91] Dong L, Yang N, Wang W, Wei F, Liu X, Wang Y, et al. Unified language model pre-training for natural language understanding and generation. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019); 2019 Dec 9–14; Vancouver, BC, Canada. 2019. p. 13042–54.
[92] Bao H, Dong L, Wei F, Wang W, Yang N, Liu X, et al. UniLMv2: pseudo-masked language models for unified language model pre-training. In: Proceedings of the 37th International Conference on Machine Learning (ICML 2020); 2020 Jul 12–18; online. 2020. p. 642–52.
[93] Xiao D, Zhang H, Li Y, Sun Y, Tian H, Wu H, et al. ERNIE-GEN: an enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI); 2021 Jan 7–15; Yokohama, Japan. 2021. p. 3997–4003.
[94] Zhang J, Zhao Y, Saleh M, Liu P. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In: Proceedings of the 37th International Conference on Machine Learning (ICML 2020); 2020 Jul 12–18; online. 2020. p. 11328–39.
[95] Rosset C. Turing-NLG: a 17-billion-parameter language model by Microsoft [Internet]. Redmond: Microsoft; 2020 Feb 13 [cited 2021 Nov 4]. Available from: https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/.
[96] Kim B, Kim HS, Lee SW, Lee G, Kwak D, Hyeon JD, et al. What changes can large-scale language models bring? Intensive study on HyperCLOVA: billions-scale Korean generative pretrained transformers. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2021 Nov 7–11; online. 2021. p. 3405–24.
[97] Xue L, Constant N, Roberts A, Kale M, Al-Rfou R, Siddhant A, et al. mT5: a massively multilingual pre-trained text-to-text transformer. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2021 Jun 6–11; online. 2021. p. 483–98.
[98] Zhang Z, Gu Y, Han X, Chen S, Xiao C, Sun Z, et al. CPM-2: large-scale cost-effective pre-trained language models. 2021. arXiv:2106.10715.
[99] Fedus W, Zoph B, Shazeer N. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. 2021. arXiv: 2101.03961.
[100] Wu S, Zhao X, Yu T, Zhang R, Shen C, Liu H, et al. Yuan 1.0: large-scale pre-trained language model in zero-shot and few-shot learning. 2021. arXiv: 2110.04725.
[101] Du N, Huang Y, Dai AM, Tong S. Lepikhin D, Xu Y, et al. GLaM: efficient scaling of language models with mixture-of-experts. 2021. arXiv: 2112.06905.
[102] Rae JW, Borgeaud S, Cai T, Millican K, Hoffmann J, Song F, et al. Scaling language models: methods, analysis & insights from training gopher. 2021. arXiv: 2112.11446.
[103] Ding M, Yan Z , Hong W, Zheng W, Zhou C, Yin D, et al. CogView: mastering text-to-image generation via transformers. 2021. arXiv: 2105.13290.
[104] Lin J, Men R, Yang A, Zhou C, Ding M, Zhang Y, et al. M6: a Chinese multimodal pretrainer. 2021. arXiv:2103.00823.
[105] Li W, Gao C, Niu G, Xiao X, Liu H, Liu J, et al. UNIMO: towards unified-modal understanding and generation via cross-modal contrastive learning. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021); 2021 Aug 1–6; online. 2021. p. 2592–607.
[106] Huo Y, Zhang M, Liu G, Lu H, Gao Y, Yang G, et al. WenLan: bridging vision and language by large-scale multi-modal pre-training. 2021. arXiv:2103.06561.
[107] Zhang H, Yin W, Fang Y, Li L, Duan B, Wu Z, et al. ERNIE-ViLG: unified generative pre-training for bidirectional vision-language generation. 2021. arXiv:2112.15283.
[108] Huang Y, Cheng Y, Bapna A, Firat O, Chen D, Chen M, et al. GPipe: efficient training of giant neural networks using pipeline parallelism. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019); 2019 Dec 9–14; Vancouver, BC, Canada. 2019. p. 103–12.
[109] Shoeybi M, Patwary M, Puri R, LeGresley P, Casper J, Catanzaro B. Megatron-LM: training multi-billion parameter language models using model parallelism. 2019. arXiv:1909.08053.
[110] Narayanan D, Shoeybi M, Casper J, LeGresley P, Patwary M, Korthikanti V, et al. Efficient large-scale language model training on GPU clusters using megatron-LM. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC 21); 2021 Nov 14–19; St. Louis, MO, USA; 2021.
[111] Bian Z, Liu H, Wang B, Huang H, Li Y, Wang C, et al. Colossal-AI: a unified deep learning system for large-scale parallel training. 2021. arXiv:2110.14883.
[112] Shazeer N, Mirhoseini A, Maziarz K, Davis A, Le Q, Hinton G, et al. Outrageously large neural networks: the sparsely-gated mixture-of-experts layer. In: Proceedings of the 5th International Conference on Learning Representations (ICLR 2017); 2017 Apr 24–26; Toulon, France; 2017.
[113] Narang S, Diamos G, Elsen E, Micikevicius P, Alben J, Garcia D, et al. Mixed precision training. In: Proceedings of the 6th International Conference on Learning Representations (ICLR 2018); 2018 Apr 30–May 3; Vancouver, BC, Canada; 2018.
[114] Rajbhandari S, Rasley J, Ruwase O, He Y. ZeRO: memory optimizations toward training trillion parameter models. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC 20); 2020 Nov 9–19; Atlanta, GA, USA; 2020.
[115] Kim Y. Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29; Doha, Qatar. 2014. p. 1746–51.
[116] Hu H, Richardson K, Xu L, Li L, Kübler S, Moss L. OCNLI: original Chinese natural language inference. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020 Nov 16–20; online. 2020. p. 3512–26.
[117] Shao CC, Liu T, Lai Y, Tseng Y, Tsai S. DRCD: a Chinese machine reading comprehension dataset. 2018. arXiv:1806.00920..
[118] Schick T, Schütze H. Exploiting cloze-questions for few-shot text classification and natural language inference. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume; 2021 Apr 19–23; online. 2021. p. 255–69.
[119] Gray S, Radford A, Kingma DP. GPU kernels for block-sparse weights. 2017. arXiv:1711.09224.
[120] Lin H, Lu Y, Han X, Sun L. Sequence-to-nuggets: nested entity mention detection via anchor-region networks. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019); 2019 Jul 28–Aug 2; Florence, Italy. 2019. p. 5182–92.
[121] Lin Y, Meng Y, Sun X, Han Q, Kuang K, Li J, et al. BertGCN: transductive text classification by combining GCN and BERT. 2021. arXiv: 2105.05727.
[122] Zhang R, Tetreault J. This email could save your life: introducing the task of email subject line generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019); 2019 Jul 28–Aug 2; Florence, Italy. 2019. p. 446–56.
[123] Zhou H, Zheng C, Huang K, Huang M, Zhu X. KdConv: a Chinese multi-domain dialogue dataset towards multi-turn knowledge-driven conversation. In:

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020); 2020 Jul 5–10; online. 2020. p. 7098–108.
[124] Cho J, Seo M, Hajishirzi H, et al. Mixture content selection for diverse sequence generation. 2019. arXiv:1909.01953.
[125] Ribeiro LFR, Zhang Y, Gardent C, Gurevych I. Modeling global and local node contexts for text generation from knowledge graphs. Trans Assoc Comput Linguist 2020;8:589–604.
[126] Zhang Y, Sun S, Galley M, Chen YC, Brockett C, Gao X, et al. DialoGPT: large-scale generative pre-training for conversational response generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL 2020); 2020 Jul 5–10; online. 2020. p. 270–8.
[127] Adiwardana D, Luong MT, So DR, Hall J, Fiedel N, Thoppilan R, et al. Towards a human-like open-domain chatbot. 2020. arXiv:2001.09977.
[128] Roller S, Dinan E, Goyal N, Ju D, Williamson M, Liu Y, et al. Recipes for building an open-domain chatbot. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume; 2021 Apr 19–23; online. 2021. p. 300–25.
[129] DuerOS [Internet]. Beijing: Baidu; c2017 [cited 2021 Nov 4]. Available from: https://dueros.baidu.com/en/index.html.
[130] Bao S, He H, Wang F, Wu H, Wang H, Wu W, et al. PLATO-2: towards building an open-domain chatbot via curriculum learning. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021); 2021 Aug 1–6; online. 2021. p. 2513–25.
[131] Gunasekara C, Kim S, D'Haro LF, Rastogi A, Chen YN, Eric M, et al. Overview of the ninth dialog system technology challenge: DSTC9. 2020. arXiv:2011.06486.
[132] Bao S, He H, Wang F, Wu H, Wang H, Wu W, et al. PLATO-XL: exploring the large-scale pre-training of dialogue generation. 2021. arXiv:2109.09519.
[133] Wang Y, Ke P, Zheng Y, Huang K, Jiang Y, Zhu X, et al. A large-scale Chinese short-text conversation dataset. In: Proceedings of the 9th CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC 2020); 2020 Oct 14–18; Zhengzhou, China. 2020. p. 91–103.
[134] Qi W, Gong Y, Yan Y, Xu C, Yao B, Zhou B, et al. ProphetNet-X: large-scale pre-training models for English, Chinese, multi-lingual, dialog, and code generation. 2021. arXiv:2104.08006.
[135] Zhou H, Ke P, Zhang Z, Gu Y, Zheng Y, Zheng C, et al. EVA: an open-domain Chinese dialogue system with large-scale generative pre-training. 2021. arXiv:2108.01547.
[136] Vinyals O, Le Q. A neural conversational model. 2015. arXiv:1506.05869.
[137] Serban I, Sordoni A, Bengio Y, Courville A, Pineau J. Building end-to-end dialogue systems using generative hierarchical neural network models. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence; 2016 Feb 12–17; Phoenix, AZ, USA. Palo Alto: AAAI Press; 2016. p. 3776–83.
[138] Worswick S. "Mitsuku wins loebner prize 2018!" [Internet]. Medium; 2018 Sep 13 [cited 2021 Nov 4]. Available from: https://medium.com/pandorabots-blog/mitsuku-wins-loebner-prize-2018-3e8d98c5f2a7.
[139] Zhou L, Gao J, Li D, Shum HY. The design and implementation of XiaoIce, an empathetic social chatbot. Comput Linguist 2020;46(1):53–93.
[140] Xin J, Tang R, Lee J, Yu Y, Lin J. DeeBERT: dynamic early exiting for accelerating BERT inference. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020); 2020 Jul 5–10; online. 2020. p. 2246–51.
[141] Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, Laroussilhe QD, Gesmundo A, et al. Parameter-efficient transfer learning for NLP. In: Proceedings of the 36th International Conference on Machine Learning (ICML 2019); 2019 Jun 9–15; Long Beach, CA, USA. 2019. p. 2790–9.
[142] Li XL, Liang P. Prefix-tuning: optimizing continuous prompts for generation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021); 2021 Aug 1–6; online. 2021. p. 4582–97.
[143] Gao T, Fisch A, Chen D. Making pre-trained language models better few-shot learners. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021); 2021 Aug 1–6; online. 2021. p. 3816–30.
[144] Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2021 Nov 7–11; online. 2021. p. 3045–59.
[145] Liu X, Zheng Y, Du Z, Ding M, Qian Y, Yang Z, et al. GPT understands, too. 2021. arXiv:2103.10385.
[146] Han X, Zhao W, Ding N, Liu Z, Sun M. PTR: prompt tuning with rules for text classification. 2021. arXiv:2105.11259.
[147] Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning. 2017. arXiv:1702.08608.
[148] Wallace E, Feng S, Kandpal N, Gardner M, Singh S. Universal adversarial triggers for attacking and analyzing NLP. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019 Nov 3–7; Hong Kong, China. 2019. p. 2153–62.
[149] Fort K, Couillault A. Yes, we care! Results of the ethics and natural language processing surveys. In: Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016); 2016 May 23–28; Portorož, Slovenia. 2016. p. 1593–600.
[150] Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: visualising image classification models and saliency maps. In: Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014); 2014 Apr 14–16; Banff, AB, Canada; 2014.
[151] Hewitt J, Manning CD. A structural probe for finding syntax in word representations. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2019 Jun 2–7; Minneapolis, MN, USA. 2019. p. 4129–38.
[152] Jawahar G, Sagot B, Seddah D. What does BERT learn about the structure of language? In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019); 2019 Jul 28–Aug 2; Florence, Italy. 2019. p. 3651–7.
[153] Linzen T, Dupoux E, Goldberg Y. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. Trans Assoc Comput Linguist 2016;4:521–35.
[154] Ribeiro MT, Singh S, Guestrin C. "Why should I trust you?" explaining the predictions of any classifier. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2016 Jun 12–17; San Diego, CA, USA. 2016. p. 1135–44.
[155] Davison J, Feldman J, Rush AM. Commonsense knowledge mining from pretrained models. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019 Nov 3–7; Hong Kong, China. 2019. p. 1173–8.
[156] Petroni F, Rocktäschel T, Riedel S, Lewis P, Bakhtin A, Wu Y, et al. Language models as knowledge bases? In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019 Nov 3–7; Hong Kong, China. 2019. p. 2463–73.
[157] Talmor A, Elazar Y, Goldberg Y, Berant J. oLMpics-on what language model pre-training captures. Trans Assoc Comput Linguist 2020;8:743–58.
[158] Morris JX, Lifland E, Yoo JY, Grigsby J, Jin D, Qi Y. TextAttack: a framework for adversarial attacks, data augmentation, and adversarial training in NLP. 2020. arXiv:2005.05909.
[159] Jia J, Liu Y, Gong NZ. BadEncoder: backdoor attacks to pre-trained encoders in self-supervised learning. 2021. arXiv:2108.00352.
[160] Devlin J. Google-research/bert [Internet]. GitHub; 2018 Oct 11 [cited 2021 Nov 4]. Available from: https://github.com/google-research/bert.
[161] Baidu Ernie Team. Paddlepaddle/ernie [Internet]. GitHub; 2019 Apr 19 [cited 2021 Nov 4]. Available from: https://github.com/PaddlePaddle/ERNIE.
[162] Huawei. Pcl-platform.intelligence/pangu-alpha [Internet]. San Francisco: OpenAI; 2021 Apr 26 [cited 2021 Nov 4]. Available from: https://git.openi.org.cn/PCL-Platform.Intelligence/PanGu-Alpha.
[163] Ding S, Shang J, Wang S, Sun Y, Tian H, Wu H, et al. ERNIE-Doc: a retrospective long-document modeling transformer. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021); 2021 Aug 1–6; online. 2021. p. 2914–27.
[164] Huggingface [Internet]. Hugging Face; 2020 Apr 26 [cited 2021 Nov 4]. Available from: https://huggingface.co.
[165] Ott M, Edunov S, Baevski A, Fan A, Gross S, Ng N, et al. FAIRSEQ: a fast, extensible toolkit for sequence modeling. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Demonstrations); 2019 Jun 2–7; Minneapolis, MN, USA. 2019. p. 48–53.
[166] Baidu PaddlePaddle Team. Paddlepaddle/paddlenlp [Internet]. GitHub; 2020 Nov 16 [cited 2021 Nov 4]. Available from: https://github.com/PaddlePaddle/PaddleNLP.
[167] Wenxin ernie [Internet]. Beijing: Baidu; c2021 [cited 2021 Nov 4]. Available from: https://wenxin.baidu.com.
[168] Alibaba Damo Academy. AliceMind [Internet]. Aliyuncs; c2021 [cited 2021 Nov 4]. Available from: https://alicemind.aliyuncs.com.
[169] Openai API [Internet]. San Francisco: OpenAI; c2021 [cited 2021 Nov 4]. Available from: https://openai.com/api.
[170] Garg Y, Chatterjee N. Sentiment analysis of twitter feeds. In: Proceedings of the 3rd International Conference on Big Data Analytics (BDA 2014); 2014 Dec 20–23; New Delhi, India. 2014. p. 33–52.
[171] AlQahtani ASM. Product sentiment analysis for amazon reviews. Int J Comput Sci Inf Technol 2021;13(3):15–30.
[172] Singh M, Jakhar AK, Pandey S. Sentiment analysis on the impact of coronavirus in social life using the BERT model. Soc Netw Anal Min 2021;11:33.
[173] Chen Z, Sokolova M. Sentiment analysis of the COVID-related r/Depression posts. 2021. arXiv:2108.06215.
[174] Liu Y, Liu J, Chen L, Lu Y, Feng S, Feng Z, et al. ERNIE-SPARSE: learning hierarchical efficient transformer through regularized self-attention. 2022. arXiv:2203.12276.
[175] Jwa H, Oh D, Park K, Kang JM, Lim H. exBAKE: automatic fake news detection model based on bidirectional encoder representations from transformers (BERT). Appl Sci 2019;9(19):4062.
[176] Soares LB, FitzGerald N, Ling J, Kwiatkowski T. Matching the blanks: distributional similarity for relation learning. 2019. arXiv:1906.03158.
[177] Wang Z, Xu Y, Cui L, Shang J, Wei F. LayoutReader: pre-training of text and layout for reading order detection. In: Proceedings of the 2021 Conference on

Empirical Methods in Natural Language Processing (EMNLP); 2021 Nov 7–11; online. 2021. p. 4735–44.
[178] gpt-2-for-the-advertising-industry [Internet]. San Francisco: OpenAI; 2017 Aug 1 [cited 2021 Nov 4]. Available from: https://www.narrativa.com/gpt-2-for-the-advertising-industry.
[179] Agarwal R, Kann K. Acrostic poem generation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020 Nov 16–20; online. 2020. p. 1230–40.
[180] Lee DH, Hu Z, Lee RKW. Improving text auto-completion with next phrase prediction. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2021 Nov 7–11; online. 2021. p. 4434–8.
[181] Mukherjee S, Mukherjee S, Hasegawa M, Awadallah AH, White R. Smart to-do: automatic generation of to-do items from emails. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020); 2020 Jul 5–10; online. 2020. p. 8680–9.
[182] What are Alexa Built-in Devices? [Internet]. Seattle: Amazon; c2010–2023 [cited 2021 Nov 4]. Available from: https://developer.amazon.com/alexa-voice-service.
[183] Mari A. Voice commerce: understanding shopping-related voice assistants and their effect on brands. In: Proceedings of the International Media Management Academic Association Annual Conference; 2019 Oct 4–6; Doha, Qatar; 2019.
[184] Google assistant update speech recognition name pronunciation BERT smart speakers [Internet]. NDTV; 2021 Apr 29 [cited 2021 Nov 4]. Available from: https://gadgets.ndtv.com/apps/news/google-assistant-update-speech-recognition-name-pronunciation-bert-smart-speak.
[185] Vincent J. The future of AI is a conversation with a computer [Internet]. New York City: The Verge; 2021 Nov 1 [cited 2021 Nov 4]. Available from: https://www.theverge.com/22734662/ai-language-artificial-intelligence-future-models-gpt-3-limitations-bias/.
[186] Meet the AI powering today's smartest smartphones [Internet]. San Francisco: Wired; 2017 Aug 1 [cited 2021 Nov 4]. Available from: https://www.wired.com/sponsored/story/meet-the-ai-powering-todays-smartest-smartphones.
[187] Nayak P. Understanding searches better than ever before [Internet]. Google; [cited 2021 Nov 4]. Available from: https://blog.google/products/search/search-language-understanding-bert/.
[188] Sun Y, Wang S, Li Y, Feng S, Tian H, Wu H, et al. ERNIE 2.0: a continual pre-training framework for language understanding. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence; 2020 Feb 7–12; New York City, NY, USA. Palo Alto: AAAI Press; 2020. p. 8968–75.
[189] Liu Y, Lu W, Cheng S, Shi D, Wang S, Cheng Z, et al. Pre-trained language model for web-scale retrieval in Baidu Search. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 21); 2021 Aug 14–18; online. 2021. p. 3365–75.
[190] Huang JT, Sharma A, Sun S, Xia L, Zhang D, Pronin P, et al. Embedding-based retrieval in Facebook Search. In: Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 20); 2020 Jul 6–10; online. 2020. p. 2553–61.
[191] Yu P, Fei H, Li P. Cross-lingual language model pretraining for retrieval. In: Proceedings of the Web Conference; 2021 Apr 19–23; online. 2021. p.1029–39.
[192] Ni M, Huang H, Su L, Cui E, Bharti T, Wang L, et al. M3P: learning universal representations via multitask multilingual multimodal pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 19–25; online. 2021. p. 3977–86.
[193] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2019. arXiv:1910.01108.
[194] Gordon MA, Duh K, Andrews N. Compressing BERT: studying the effects of weight pruning on transfer learning. In: Proceedings of the 5th Workshop on Representation Learning for NLP; 2020 Jul 9; Seattle, WA, USA. 2020. p. 143–55.
[195] Kim S, Gholami A, Yao Z, Mahoney MW, Keutzer K. I-BERT: integer-only BERT quantization. In: Proceedings of the 38th International Conference on Machine Learning (ICML 2021); 2021 Jul 18–24; online. 2021. p. 5506–18.