CHATBOT IN PYTHON

Innovation:

Exploring advanced techniques like using pre trained language models to enhance the quality of responses.

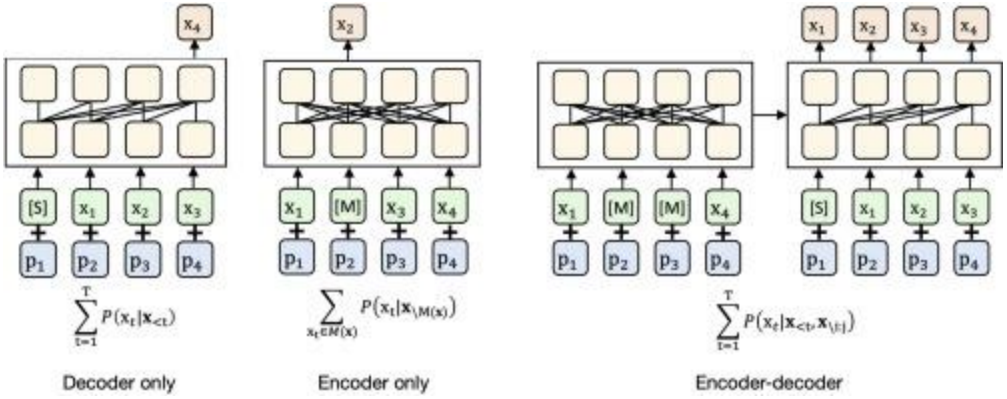
Pre trained transformer:

Generative Pre-trained Transformer 3 (GPT-3) is a new language model created by openAI that is able to generate written text of such quality that is often difficult to differentiate from text written by a human.

Pre-Trained Language Models and Their Applications

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.

Graphical Representation



- Previous article in issue
- Next article in issue

Keywords

Pre-trained models
Natural language processing

1. A brief history of pre-trained models

The concept of pre-training is related to transfer learning . 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 annotated data has become the dominant transfer learning approach. The difference is that pre-training methods use annotated 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 annotated texts . The first milestone of neural language modeling appears in Ref. 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 . Among the RNN family, long short-term memory (LSTM) 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 considerable efforts have been devoted to building stronger and more efficient language models based on the transformer architecture. In neural language modeling, distributed word

representations named “word embedding’s” that are learned with models such as Word2Vec and Glove have become common initializations for the word vectors of deep learning models, significantly improving the performance of downstream tasks such as named-entity recognition part-of-speech tagging , and question answering .

Although methods that leverage static word embedding’s 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 introduced context–aware language modeling, which uses annotated data to improve sequencelearning 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 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.

In another pioneering study on pre-trained models (PTMs), embedding’s 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 . 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) 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 , 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) , text-to-text transfer transformers (T5) GPT-3 large-scale generative Chinese pre-trained language model (CPM) PanGu-α and ERNIE 3.0 Titan . 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 . 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.

encourages models to align the representations of two languages together. Researchers have also released more multilingual language models, such as XLM-RoBERTa (XLM-R) , InfoXLM , and ERNIE-M 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.

Recently, contrastive learning has been successfully utilized for visual self-supervised pre-training. Contrastive predictive coding has achieved strong results in various scenarios, including speech, image, and text. These methods 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, 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 , videos and speech enabling a broad application scope of PTMs. Such methods 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 and CLIP-guided generation 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. propose knowledge-enhanced scene graph prediction to capture the alignments of more detailed semantics. Gan et al. incorporate adversarial training into pre-training and achieves higher performance. Cho et al. 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 , BERT-based training techniques prompted-based learning , data augmentation , text generation, and conversational agent design Another line provides a panoramic perspective of the whole progress of PTMs. For example, Ramponi and Plank provide an overview from early traditional non-neural methods to PTMs in NLP. Qiu et al. systematically categorize existing PTMs from four different perspectives and outlines some potential directions of PTMs for future research. Bommasani et al. 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. 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.

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 $x = x_1, x_2, x_3, \ldots, x_T$ (where x is the original sentence, x_t ($t = 1, 2, \ldots, T$) is the tth token, and T is the sequence length), an autoregressive model factorizes the likelihood of the input text sequence as $p(x) = \prod_{t=1}^{T} p(x_t \mid x_{<t})$, 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.

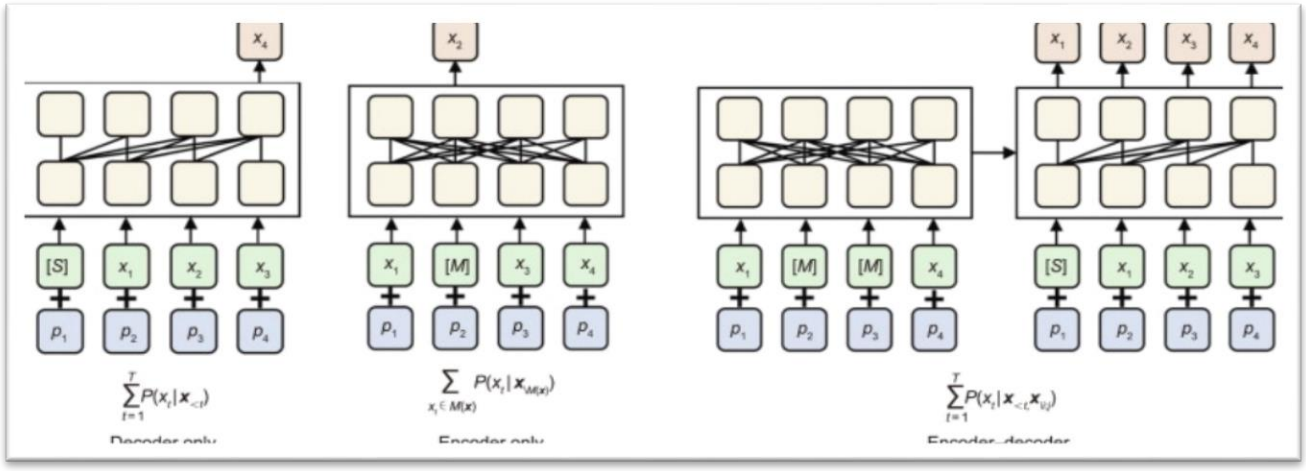


Fig. 1. An illustration of the existing prevalent pre-training frameworks, where x is the original sentence, x_t ($t = 1, 2, \ldots, T$) is the $t$he token, T is the sequence length, and $M(x)$ is the set of masked tokens in x. S denotes the start token embedding of a sequence. p_1, p_2, p_3, and p_4 denote the position embedding's 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.

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. GPTwas 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 Books Corpus 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 the potential of large-scale PTMs. GPT-2 follows the unidirectional framework with a transformer decoder that was trained with a larger corpus, namely, Web Text, 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 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. 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. 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.

ThankYou