SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2024, Vol. 67, Iss. 5, 152102:1–152102:13
https://doi.org/10.1007/s11432-021-3536-5

CPT: a pre-trained unbalanced transformer for both Chinese language understanding and generation

Yunfan SHAO[1,3], Zhichao GENG[1,3], Yitao LIU[1,3], Junqi DAI[1,3], Hang YAN[1,3], Fei YANG[2], Zhe LI[2], Hujun BAO[2] & Xipeng QIU[1,3*]

[1]*School of Computer Science, Fudan University, Shanghai* 200433*, China;*
[2]*Zhejiang Lab, Hangzhou* 311121*, China;*
[3]*Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai* 200433*, China*

Received 22 December 2021/Revised 1 April 2022/Accepted 28 April 2022/Published online 27 March 2024

Abstract In this paper, we take the advantage of previous pre-trained models (PTMs) and propose a novel Chinese pre-trained unbalanced transformer (CPT). Different from previous Chinese PTMs, CPT is designed to utilize the shared knowledge between natural language understanding (NLU) and natural language generation (NLG) to boost the performance. CPT consists of three parts: a shared encoder, an understanding decoder, and a generation decoder. Two specific decoders with a shared encoder are pre-trained with masked language modeling (MLM) and denoising auto-encoding (DAE) tasks, respectively. With the partially shared architecture and multi-task pre-training, CPT can (1) learn specific knowledge of both NLU or NLG tasks with two decoders and (2) be fine-tuned flexibly that fully exploits the potential of the model. Moreover, the unbalanced transformer saves the computational and storage cost, which makes CPT competitive and greatly accelerates the inference of text generation. Experimental results on a wide range of Chinese NLU and NLG tasks show the effectiveness of CPT.

Keywords pre-trained model, transformer, language model, generation, unified model

1 Introduction

Recently, large-scale pre-trained models (PTMs) have become backbone models for many natural language processing (NLP) tasks [1]. However, existing PTMs are usually trained with different architectures and pre-training tasks. When applying PTMs to a downstream task, we should choose a suitable one as the backbone model according to its pre-training nature. For example, we usually select bidirectional encoder representations from transformers (BERT) or RoBERTa [2,3] as the backbone model for natural language understanding (NLU) tasks, and bidirectional and auto-regressive transformer (BART) or generative pre-trained transformer (GPT) [4,5] for natural language generation (NLG) tasks. With the success of PTMs in English, many studies have been done to train the counterparts for Chinese [6–11]. However, these Chinese PTMs usually follow the settings of English PTMs, which makes these models focus on either language understanding or language generation, lacking the use of sharing knowledge between NLU and NLG tasks. Therefore, it is attractive to pre-train a joint model for both NLU and NLG tasks.

Few studies attempt to fuse NLU and NLG into a unified model. Unified pre-trained language models (UniLMs) [12,13] and general language model (GLM) [14] adopt a unified transformer encoder for both understanding and generation; however, their architectures restrict them to employ more flexible pre-training tasks, such as denoising auto-encoding (DAE) used in BART, a widely successful pre-training task for NLG. Pre-trained autoencoding and autoregressive language model (PALM) [15] adopts the standard transformer and adds an auxiliary masked language modeling (MLM) task to enhance the understanding ability; however, it still focuses on language generation tasks.

In this paper, we propose CPT, a novel Chinese pre-trained unbalanced transformer for both NLU and NLG tasks. The architecture of CPT is very concise (as shown in Figure 1), which divides a full

* Corresponding author (email: xpqiu@fudan.edu.cn)

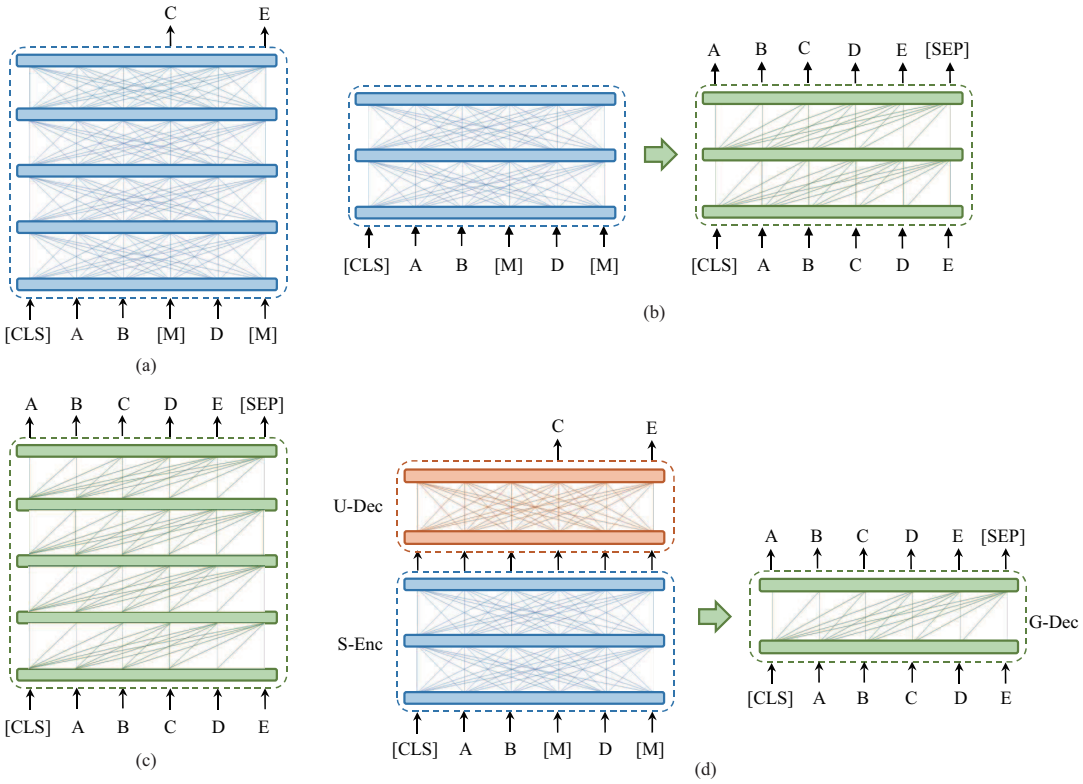

Figure 1 (Color online) Architecture of CPT and the counterpart PTMs. (a) BERT; (b) BART; (c) GPT; (d) CPT. Different from other PTMs, CPT consists of three parts: S-Enc, U-Dec, and G-Dec.

transformer encoder-decoder into three parts: (1) a shared encoder (S-Enc) to capture the common representation; (2) a decoder for understanding (U-Dec), which uses full self-attention and is pre-trained with MLM; (3) a decoder for generation (G-Dec), which adopts masked self-attention and is pre-trained with the DAE task. By multi-task pre-training, CPT is able to improve the performance on both language understanding and generation, respectively[1)].

The main properties of CPT are as follows.

(1) CPT can be regarded as two separated PTMs with an S-Enc. Two specific decoders are pre-trained with MLM and DAE tasks, respectively. Each decoder can learn the specific knowledge on either NLU or NLG tasks, while the S-Enc learns the common knowledge for universal language representation.

(2) Two separated decoders enable CPT to adapt to various downstream tasks flexibly. For example, CPT could be fine-tuned with at least five modes for classification tasks (as shown in Figure 2), which exploits the full potential of CPT. Thus, we could choose a suitable fine-tuning mode based on the attributes and characteristics of downstream tasks.

(3) The overall architecture of CPT is an unbalance transformer. To make the computational cost and the size of CPT comparable with popular PTMs, such as BERT and BART, we use a novel architecture consisting of a deeper S-Enc and two shallower decoders. Especially, the shallow G-Dec greatly accelerates the inference of text generation.

We conduct experiments on various language understanding and text generation tasks, including datasets for text classification, sequence labeling, machine reading comprehension (MRC), summarization, and data-to-text generation. Results show that CPT could achieve competitive results with state-of-the-art on these datasets.

1) Code is available at https://github.com/fastnlp/CPT.

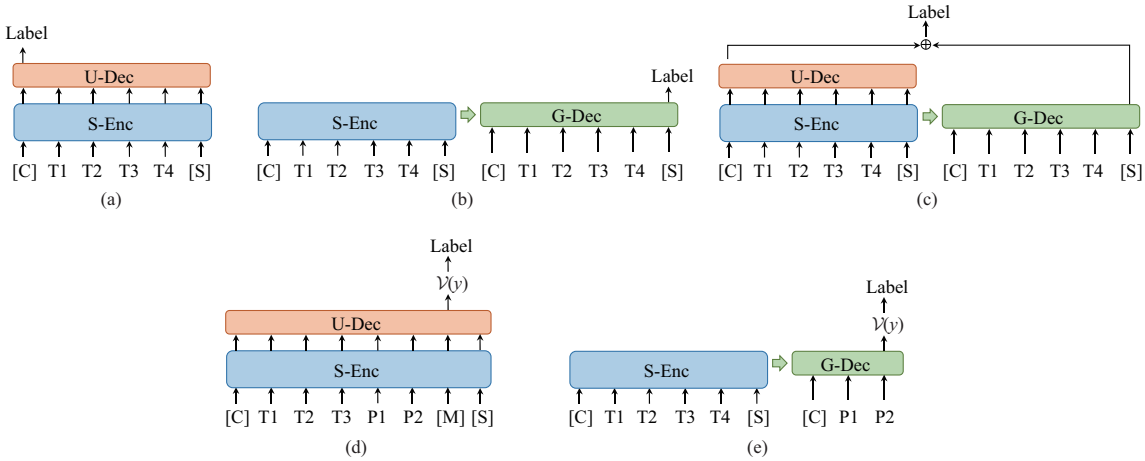

Figure 2 (Color online) Five ways to fine-tune CPT for text classification. (a) CPT_u; (b) CPT_g; (c) CPT_{ug}; (d) CPT_{u+p}; (e) CPT_{g+p}. "T1"–"T4" and "P1", "P2" refer to text input $\boldsymbol{x}$ and prompt tokens, respectively. $\mathcal{V}(y)$ is the mapping function that maps the language model predictions to the label. `[C]` and `[S]` are abbreviations for `[CLS]` and `[SEP]`, respectively.

2 Related work

2.1 PTMs towards both NLU and NLG

Recently, there are some efforts to combine language understanding and generation into a single PTM. UniLM [12] pre-trained with an ensemble of attention masks, which allows the model to be used for both generative and classification tasks. A difference is that all parameters of UniLM are shared between generation and discrimination, whereas CPT uses two separated decoders. Thus, CPT can utilize the DAE pre-training task which is proven to be effective for NLG tasks [4].

PALM [15] is a PTM focusing on conditional generation. To force the encoder to comprehend the meaning of the given context, MLM is added to pre-train the encoder. In contrast, CPT has an individual decoder for MLM which can avoid the negative effects brought by DAE. Therefore CPT also has good performance on NLU tasks.

More recently, ERNIE 3.0 [16] also uses a universal encoder and several task-specific decoders, but it adopts Transformer-XL as the backbone and its generative pre-training task is left-to-right LM with a special masked attention matrix. Different from ERNIE 3.0, CPT adopts the encoder-decoder architecture and is more suitable for sequence-to-sequence (Seq2Seq) tasks.

2.2 Chinese PTMs

Many attempts have been conducted to pre-train the Chinese counterparts of PTMs.

The first line of works follows BERT and uses MLM with whole word masking (WWM) strategy to pre-train transformer encoder, such as Chinese versions of BERT and RoBERTa [6], the neural contextualized representation for Chinese language understanding (NEZHA) [8], ZEN [17]. Some of them add special features of Chinese characters or words to further boost the performance of NLU tasks, such as ERNIE 1.0/2.0 [7,18], ChineseBERT [19]. However, these PTMs could not be adopted to text generation directly.

The second line of works follows GPT and uses the left-to-right LM task to pre-train a transformer decoder, such as the Chinese pre-trained language model (CPM) [10] and PanGu [11]. Although large-scale PTMs with tens of billions of parameters have been released recently, the huge computation and storage cost hinders their applications.

The third line of works aims to pre-train the full transformer encoder-decoder. CPM-2 [10] follows T5 [20] and adopts a Seq2Seq MLM pre-training task, which predicts the masked tokens in a Seq2Seq fashion. Although BART [4] has achieved wide success on conditional text generation tasks, such as text summarization [21, 22] and dialogue system [23], it still lacks corresponding Chinese versions[2)].

Different from the above Chinese PTMs, CPT is a pre-trained unbalanced transformer with MLM and DAE tasks, which is capable of achieving competitive results on both NLU and NLG tasks. Besides, CPT is parameter efficient compared to these large-scale models. Table 1 compares different Chinese PTMs.

2) Besides CPT, we also provide a Chinese BART as a byproduct.

Table 1 Summary of some representative Chinese PTMs[a)]

	BERT RoBERTa	ZEN NEZHA ERNIE-1.0/2.0	PanGu-α	CPM	CPM-2	BART	CPT
# Params	Base-110M Large-340M	≈ BERT	32Layers-2.6B 40Layers-13.1B 64Layers-207.0B	Small-110M Medium-340M Large-2.6B	Base-11B MOE-198B	Base-139M Large-406M	Base-121M Large-393M
Arch.	Transformer encoder	Transformer encoder variant	Transformer decoder	Transformer decoder	Full transformer	Full transformer	Unbalanced full transformer
PreTrain. Task	MLM	MLM	LM	LM	Seq2Seq MLM	DAE	MLM+DAE
Tok.	Char	Char	Word/char	Word/char	Word/char	Char	Char
Masking	Word	–	–	–	–	Word	Word
Prediction	Char	Char	Word/char	Word/char	Word/char	Char	Char
NLU	✓	✓	✗	✗	✗	✗	✓
NLG	✗	✗	✓	✓	✓	✓	✓

a) "# Params" refers to the number of parameters. "Arch." refers to the model architecture. "LM" refers to language modeling in auto-regression fashion, while "Seq2Seq MLM" refers to MLM in Seq2Seq fashion. "Tok.", "Masking" and "Prediction" refer to the tokenization, masking, and prediction granularity of the model, respectively. "✓" means "could be directly used to" while "✗" means "need to be adapted to".

2.3 Multi-task pre-training

Incorporating multi-task learning into pre-training has drawn increasing attention recently. Most recent advancements attempt to improve performance by leveraging multi-task learning beyond standard pre-training [20,24–26]. This line of works focuses on downstream task performance improvements by utilizing a collection of labeled datasets. However, our work is focusing on closing the gap between language understanding and text generation tasks by applying multi-task learning on large-scale unlabeled texts.

3 Model architecture

As shown in Figure 1, The architecture of CPT is a variant of the full transformer and consists of three parts:

(1) S-Enc. A transformer encoder with fully-connected self-attention, which is designed to capture the common semantic representation for both language understanding and generation.

(2) U-Dec. A shallow transformer encoder with fully-connected self-attention, which is designed for NLU tasks. The input of U-Dec is the output of S-Enc.

(3) G-Dec. A transformer decoder with masked self-attention, which is designed for generation tasks with auto-regressive fashion. G-Dec utilizes the output of S-Enc with cross-attention.

With the two specific decoders, CPT can be used flexibly. For example, CPT can be easily fine-tuned for NLU tasks using just S-Enc and U-Dec, and can be regarded as the standard transformer encoder; while for NLG tasks, CPT adopts S-Enc and G-dec, and forms a transformer encoder-decoder. With different combinations, CPT is able to be effectively applied on various downstream tasks, which fully exploits the pre-trained parameters and obtains competitive performance. More combinations and use cases will be discussed in Section 5.

Different from most PTMs with encoder-decoders, we exploit a deep-shallow framework for S-Enc and decoders. More specifically, we use a deeper encoder and two shallow decoders for CPT. We assume that a shallow decoder retains the performance on text generation and reduces decoding time, which has proven to be effective for neural machine translation [27] and spell checking [28].

The deep-shallow setup makes CPT more general for both understanding and generative tasks with minor parameter overheads. It also accelerates the inference of CPT for text generation as the G-Dec is a light decoder.

4 Pre-training

To make CPT good at both NLU and NLG tasks, we introduce two pre-training tasks.

(1) MLM. We pre-train the parameters of S-Enc and U-Dec with MLM [2, 6]. Given a sentence, we randomly replace some tokens with the `[MASK]` token and train S-Enc and U-Dec to predict the masked tokens. Following [6], we adopt WWM to replace the tokens. Compared with randomly token masking, WWM is more suitable for inducing semantic information carried by words and spans.

(2) DAE. We pre-train the parameters of S-Enc and G-Dec by reconstructing the original document based on the corrupted input. According to the studies of BART [4], we corrupted the input in two effective ways. (i) Token Infilling: a WWM strategy with single mask replacement. First, a number of words are sampled based on the segmentation. Then, each selected word is replaced with a single `[MASK]` token, regardless of how many tokens it consists. (ii) Sentence permutation: sentences are extracted from a document based on punctuation, and shuffled in a random order.

In practice, We first use a Chinese word segmentation (CWS) tool to split the sentences into words. Then, we select 15% of the words and mask the corresponding characters. For the masked characters, we follow the setup of BERT to (1) replace 80% of them with a special `[MASK]` token, (2) replace 10% of them with random tokens, (3) keep the rest 10% of them unchanged.

Finally, we train CPT with two pre-training tasks under a multi-task learning framework. Thus, CPT can learn for both understanding and generation, and can easily deal with downstream NLU or NLG tasks.

5 Fine-tuning

PTMs are usually fine-tuned in only a few ways for a given downstream task. For example, for sentence-level classification, we fine-tune BERT by taking the top-layer output of `[CLS]` token as the representation of the whole sentence, while fine-tune GPT by using the representation of the last token of the sequence.

Thanks to the separated understanding and G-Decs, CPT can be fine-tuned in multiple patterns. For a given downstream task, one could choose the most suitable way to fully stimulate the potential of CPT to achieve competitive results.

5.1 Fine-tuning for sentence-level classification

When incorporating external classifiers, CPT has three fine-tuning modes for sequence-level classification (As shown in Figures 2(a)–(c)).

(1) CPT_u. A BERT-style mode. The sentence representation is from the U-Dec module only, which is usually the first state of `[CLS]` token.

(2) CPT_g. A BART-style mode. The same input is fed into the S-Enc and G-Dec, and the representation from the final output token `[SEP]` from G-Dec is used.

(3) CPT_{ug}. The same input is fed into the S-Enc and G-Dec, and the final representation is the concatenation of the first output of U-Dec and the final output of G-Dec.

Recently, a powerful and attractive framework, prompt-based learning [29–31], is also able to boost the performance of PTMs. By defining prompting templates and reformulating the classification tasks in a generative fashion, the framework utilizes PTMs to generate words corresponding to task labels. The generative patterns are so close to the pre-training tasks of PTMs that they have the ability of few-shot or even zero-shot learning.

The prompt-based methods could also be applied to CPT with more flexible fashions since CPT has two decoders. As shown in Figures 2(d) and (e), we construct prompts and convert the task into a generation task with CPT by the following two modes.

(1) CPT_{u+p}: a MLM task. We manually construct an input template and assign a word to each task label. CPT is fine-tuned to predict the word at the masked positions, which will be mapped to the task labels. Since a word may be tokenized into multiple character tokens, the predicted distributions at masked positions are averaged to get the predicted distribution of labels.

(2) CPT_{g+p}: conditional text generation. We encode the input text with S-Enc and train CPT to generate prompt text initialized with corresponding labels by teacher forcing. For inference, we first construct the prompt text for each label. Then, the perplexity of each prompt text is calculated. Finally, the prediction is assigned to the label with the highest corresponding perplexity.

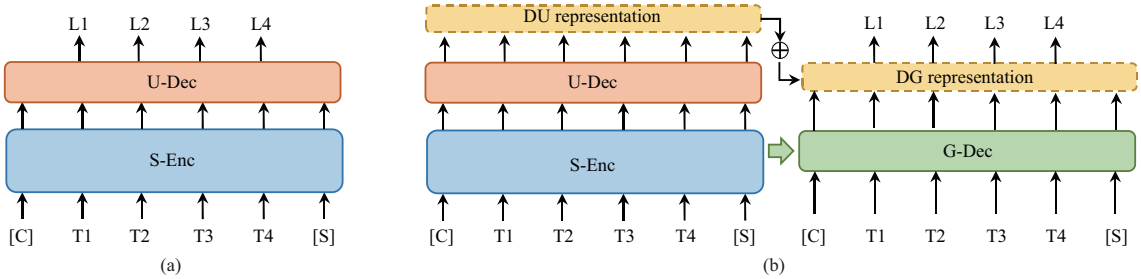

Figure 3 (Color online) Two examples of fine-tuning CPT for sequence labeling. (a) CPT$_u$; (b) CPT$_{ug}$. "T1"–"T4" and "L1"–"L4" refer to text input $\boldsymbol{x}$ and token labels, respectively.

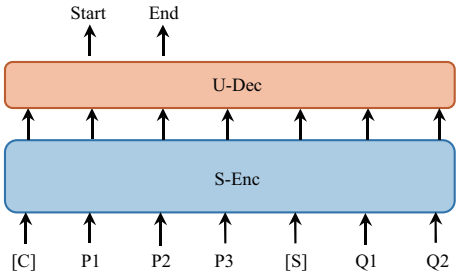

Figure 4 (Color online) Example of fine-tuning CPT$_u$ for machine reading comprehension. "P1"–"P3" and "Q1", "Q2" refer to passages and questions, respectively.

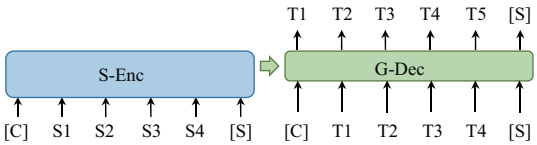

Figure 5 (Color online) Example of fine-tuning CPT$_g$ for conditional generation. "S1"–"S4" and "T1"–"T5" refer to input and target sequences, respectively.

5.2 Fine-tuning for sequence labeling

For sequence labeling, each token needs a representation for token-level classification. Similar to sequence-level classification, we leverage PTMs to obtain high quality token representations and then put the representations to a trainable classifier to assign labels for these tokens. Thus, similar to sentence-level classification, we can fine-tune CPT for sequence labeling as CPT$_u$, CPT$_g$, and CPT$_{ug}$, using (1) U-Dec only, (2) G-Dec only, or (3) both U-Dec and G-Dec. Figure 3 shows two examples of sequence labeling.

5.3 Fine-tuning for MRC

MRC requires the model to predict an answer span shown in the passage for a given question. A typical fine-tuning pattern is to train PTMs to predict the start and end positions of the span in the passage. The prediction is based on the tokens of the passage. Thus, CPT$_u$, CPT$_g$, and CPT$_{ug}$ can be fine-tuned, similar to sequence-labeling. Figure 4 shows the example of CPT$_u$.

5.4 Fine-tuning for conditional generation

Apart from NLU tasks, CPT can do text generation efficiently. As shown in Figure 5, we simply fine-tune CPT$_g$ with S-Enc and G-Dec modules on text generation tasks, similar to the usage of other auto-regressive PTMs [4].

6 Experiments

6.1 Pre-training setups

We implement two versions of CPT, namely, base and large, respectively consisting of 14/28 transformer layers with 10/20 layers for the S-Enc and 2/4 layers for each task specific decoder. And the hidden units and attention heads per layer for base and large versions are 768/1024 and 12/16, respectively. The total number of layers activated for a given task is always equal to 12/24, which makes our model compared with the base/large-size of BERT and its variants (RoBERTa, ERNIE 1.0/2.0, etc).

We train our models on the open source large-scale raw text, Chinese Wikipedia, and a part of WuDaoCorpus. The training data contains 200 GB cleaned text ranges from different domains. We use Jieba to

segment Chinese words for WWM and use WordPiece tokenizer inherited from BERT to split input text into tokens. We use Adam to train the models for 500k steps, with a batch size of 2048, a learning rate of 1e−4, $\beta_1 = 0.9$, $\beta_2 = 0.98$, weight decay of 0.01. We warmup the learning rate for the first 10000 steps then do linear decay. In addition, a Chinese BART is pre-trained with the same corpora, tokenization, and hyper-parameters as a baseline.

6.2 Evaluation tasks

To evaluate the effectiveness of our model, we conduct experiments on various NLP datasets across different understanding and generation tasks, with details illustrated below.

Classification. We evaluate the model on the Chinese language understanding evaluation benchmark (CLUE) [32], which contains text classification datasets, TouTiao text classification for news titles (TNEWS) and IFLYTEK, natural language inference (NLI) dataset, the original Chinese natural language inference (OCNLI), sentence pair matching (SPM) dataset, ant financial question matching corpus (AFQMC), and coreference resolution (CoRE) dataset, CLUEWSC 2020 (WSC) key word recognition (KwRE) dataset, Chinese scientific literature (CSL). We conduct data augmentation on CSL as [33] performed, and evaluate TNEWS on version 1.1 test set. Accuracy is used for these datasets.

Sequence labeling. We evaluate our model on Chinese word segmentation (CWS) and named entity recognition (NER), which are two representative sequence labeling tasks. We use two datasets from SIGHAN2005 [34] for CWS, which are Microsoft Research Corpus (MSR), Beijing University Corpus (PKU). And for NER, the Microsoft Research Asia NER dataset (MSRA) [35], OntoNotes[3)] are used. We use the same dataset preprocessing and split methods as in previous studies [36–38]. And F1 scores are reported.

MRC. Span based machine reading comprehension (MRC) dataset CMRC 2018 (CMRC) [39] and traditional Chinese MRC dataset DRCD [40] are used. We follow the data processing in [6, 41] and transform the text from DRCD is transformed to simplified Chinese. The exact match (EM) scores are reported.

Text generation. We use two abstractive summarization datasets, the large scale Chinese short text summarization dataset (LCSTS) [42] and CSL[4)], and a data-to-text generation dataset, ADGEN [43] to evaluate the text generation ability of our model. Among them, LCSTS is a large corpus of Chinese short text summarization dataset constructed from Sina Weibo, consisting of 2 million real Chinese short texts with short summaries. And CSL is an academic domain text summarization dataset, constructed from abstract and titles from publications in the computer science domain. And ADGEN is a data-to-text dataset that requires models to generate long text for the advertisement based on some keywords. And we evaluate PTMs on test sets of LCSTS and ADGEN and the development set of CSL. The character-level Rouge-L is used to evaluate the summarization results. For ADGEN, we follow [10] to use BLEU-4.

6.3 Compared PTMs

We compare CPT with a series of state-of-the-art PTMs for either NLU or text generation. The details are as follows.

PTMs for NLU. PTMs with the transformer encoder structure and pre-trained with MLM usually perform well in NLU tasks, such as the Chinese versions of BERT and RoBERTa [6], NEZHA [8], ERNIE 2.0 [18], MacBERT [41]. Unless otherwise specified, we use BERT and RoBERTa to refer to `BERT-wwm-ext` and `RoBERTa-wwm-ext`, respectively.

PTMs for NLG. For text generation, we compare CPT with generative transformers ranging from normal size to large scale, including BART [4], mBART [44], mT5 [45], CPM-2 [10], and models with pre-trained encoders. BART is a sequence-to-sequence model pre-trained with a DAE task. Due to the missing of the Chinese version, we train a Chinese BART as mentioned in Subsection 6.1. mBART is a multilingual variant of BART. And mT5 is a multilingual variant of T5 pre-trained on over 101 languages, including Chinese. CPM-2 is a large-scale encoder-decoder model with 11 billion parameters, pre-trained in multiple stages with large-scale Chinese and bilingual data. We also report generative models adopted from transformer encoders such as RoBERTa and ERNIE 2.0 that follow the generation style of UniLM [12], to further evaluate the effectiveness of generative pre-training.

3) https://catalog.ldc.upenn.edu/LDC2011T03.
4) https://github.com/CLUEbenchmark/CLGE.

Table 2 Accuracy results on the development sets of CLUE benchmark. We fine-tune CPT with five different ways as shown in Figure 2[a)]

Models	TNEWS	IFLYTEK	OCNLI	AFQMC	CSL	WSC	Average
BERT (B)	56.8	58.9	75.4	72.0	82.3	83.2	71.4
RoBERTa (B)	57.5	59.4	76.5	74.4	86.1	88.8	73.8
BART (B)	57.2	60.0	76.1	73.0	85.8	79.6	71.9
CPT_u (B)	**58.4**	60.5	76.4	**75.1**	86.1	**91.1**	**74.6**
CPT_g (B)	57.3	60.4	76.3	71.4	86.4	87.2	73.2
CPT_{ug} (B)	57.4	**61.9**	**76.8**	70.6	86.3	89.8	73.8
CPT_{g+p} (B)	54.9	25.4	76.6	73.7	**86.9**	79.9	66.2
CPT_{u+p} (B)	**58.4**	61.6	76.6	**75.1**	**86.9**	79.9	73.1
RoBERTa (L)	58.3	61.7	78.5	75.4	86.3	89.5	75.0
BART (L)	**59.2**	62.1	79.7	75.7	**87.3**	90.1	75.7
CPT_u (L)	58.8	61.8	79.5	**75.9**	86.5	92.1	75.8
CPT_g (L)	59.1	61.7	**79.9**	75.8	86.9	91.8	75.9
CPT_{ug} (L)	**59.2**	**62.4**	79.8	75.8	86.6	**93.4**	**76.2**
CPT_{g+p} (L)	54.5	29.2	79.8	75.4	87.1	89.5	69.2
CPT_{u+p} (L)	59.0	61.2	79.6	75.4	**87.3**	87.8	75.1

a) (B) and (L) refer to base-size and large-size of PTMs, respectively. The bold font indicates the highest value.

Table 3 Accuracy results on the test sets of CLUE benchmark[a)]

Models	TNEWS	IFLYTEK	OCNLI	AFQMC	CSL	WSC	Average
BERT (B)	58.6	59.4	73.2	74.1	84.2	74.5	70.7
RoBERTa (B)	**59.5**	60.3	**73.9**	74.0	84.7	76.9	71.5
BART (B)	58.5	**60.7**	72.1	74.0	85.4	67.6	69.7
CPT_u (B)	59.2	60.5	73.4	**74.4**	**85.5**	**81.4**	**72.4**
RoBERTa (L)	58.9	**63.0**	76.4	**76.6**	82.1	74.6	71.9
BART (L)	58.6	62.7	78.1	74.3	**86.7**	82.1	73.7
CPT_{ug} (L)	**59.2**	62.4	**78.4**	75.0	85.5	**86.2**	**74.5**

a) The bold font indicates the highest value.

6.4 Main results

To fully release the potential of our model, we fine-tune CPT for NLU tasks in different ways as mentioned in Section 5, denoted as CPT_u, CPT_g, and CPT_{ug}, CPT_{u+p}, and CPT_{g+p}, respectively. We use (B) and (L) to distinguish base and large versions of PTMs, respectively.

Classification. Table 2 shows the development set results of the CLUE benchmark of different fine-tuning modes. As a result, CPT_u (B) achieves a 74.6 on average, surpassing other baselines and fine-tuning patterns on the base version of CPT. Besides, CPT_{ug} (L) obtains an average accuracy of 76.2, which is better than RoBERTa (L) by a large margin. Therefore, we choose CPT_u (B) and CPT_{ug} (L) as the most suitable fine-tuning patterns to do the classification. We find that the best fine-tuning modes are different between base and large models. We believe the difference is brought by the scale of the parameters. For the base model, the G-Dec is too shallow to transfer for NLU tasks, which makes CPT_{ug} could not beat the CPT_u. And the G-Dec in the large version of CPT has more parameters and layers, which makes the decoder easy to transfer.

For prompt-based fine-tuning (Table 2), we find that directly fine-tuning without prompt works well on some datasets, with the small gaps between CPT_u, CPT_g, and CPT_{ug}. Moreover, CPT_{u+p} achieves good results on some datasets that even outperform methods without prompt tuning. However, the accuracy of prompt-base methods on other datasets drops a lot. As there are many factors that affect prompt tuning performance including prompt design and choices of words for labels. Manually designed prompts may be suboptimal. Besides, we find that CPT_{g+p} degenerates obviously on TNEWS and IFLYTEK. Both datasets have more than 3 classes, which contain 15 and 112 labels, respectively. Moreover, these labels are hard to represent by a single character. In practice, we assign words with up to 7 characters to a label. We presume that the large number of labels and the multi-token issue hinders CPT_{g+p} to generate correctly.

Table 3 reports the performance of CPT on classification tasks and the comparison with previous

Table 4 Results on sequence labeling datasets[a)]

	CWS		NER	
	MSR	PKU	MSRA	OntoNotes
BERT (B)	98.24	96.50	95.13	81.73
ERNIE 2.0* (B)	–	–	93.80	–
RoBERTa (B)	98.14	96.15	95.23	81.52
CPT_u (B)	**98.29**	**96.58**	**95.78**	**82.08**
ERNIE 2.0* (L)	–	–	95.00	–
RoBERTa (L)	98.42	96.37	95.20	81.78
CPT_u (L)	**98.51**	**96.70**	**96.20**	**83.08**

a) The F1 scores on test sets are reported. Models with * indicate the results are from [18]. The bold font indicates the highest value.

Table 5 Results on MRC datasets[a)]

	CMRC 2018	DRCD	
	Development set	Development set	Test set
RoBERTa (B)	67.9	85.9	85.2
MacBERT* (B)	68.2	**89.2**	88.7
ERNIE 2.0* (B)	**69.1**	88.5	88.0
NEZHA* (B)	67.8	–	–
CPT_u (B)	68.8	89.0	**89.0**
RoBERTa (L)	70.6	89.1	88.9
MacBERT* (L)	70.1	90.8	90.9
ERNIE 2.0* (L)	71.5	89.7	89.0
NEZHA* (L)	68.1	–	–
CPT_u (L)	**72.3**	**91.0**	**91.1**

a) EM scores are reported. Models with * indicate the results from the corresponding work. The bold font indicates the highest value.

representative Chinese PTMs. We report accuracy on the test sets of these datasets. Among the fine-tuned CPTs, we choose base version CPT_u and large version CPT_{ug} as they obtain the best results on development sets. Base size CPT consistently outperforms BERT, RoBERTa, and ERNIE. Moreover, large-size CPT achieves a 74.5 average score, outperforming RoBERTa (L) by a large margin. We find that generative PTMs, such as BART, also have the ability to handle discrimination tasks (see Tables 2 and 3). However, their performance is suboptimal compared with the CPT. As the uni-directional layers of generative models could hurt the performance of NLU tasks.

Sequence labeling. The CPT is fine-tuned as CPT_u, CPT_g, and CPT_{ug} and evaluated on development sets. We find that CPT_u constantly obtains the best development results. We conjecture that CWS and NER have more dependency on local syntax than complex semantics used for text generation. Thus, CPT_u is more suitable for CWS and NER with its bidirectional fully connected self-attention. As a result, we report the test set results of CPT_u to compare with other PTMs.

We compare our model with other state-of-the-art methods on sequence labeling datasets. As shown in Table 4, CPT_u (L) achieves the highest performance and exceeds the BERT (L), RoBERTa (L), and ERNIE (L) on all sequence labeling tasks, both CWS and NER. And CPT_u (B) obtains comparable results, surpassing base versions of BERT and RoBERTa.

Note that CPT_{ug} outperforms the CPT_u in the large size while surpassed by CPT_u in the base version. We believe that it is the large discrepancy between pre-training and fine-tuning tasks, which makes the G-Dec trained by the DAE task hard to be transferred to classification. G-Dec is harder to be fine-tuned than U-Dec, especially in the base model where G-Dec is very shallow. And it also explains that the performance gap between CPT_u and CPT_g in the base version is larger than the large size.

MRC. Table 5 shows the experimental results on MRC tasks, which also indicates the effectiveness of CPT. We report the EM score on CMRC development set, DRCD development and test sets. We try and evaluate CPT_u, CPT_u, and CPT_u on the development sets of these datasets and choose the pattern that acquires the best results to report. In conclusion, CPT_u obtains comparable or higher results compared to previous systems that are widely used, such as RoBERTa, MacBERT, ERNIE, and NEZHA. Moreover, CPT_u consistently outperforms other strong baselines by a large margin, with a 72.3 EM score on the CMRC development set and 91.1 EM on the DRCD test set.

Table 6 Results on text generation datasets[a)]

Models	LCSTS (Rouge-L)	CSL (Rouge-L)	ADGEN (BLEU-4)
mT5 (S)	33.5	56.7	**10.2**
BART (B)	37.8	62.1	9.9
CPT_g (B)	**38.2**	**63.0**	9.8
CPM-2†	35.9	–	10.6
mBART (L)	37.8	55.2	8.5
mT5 (B)	36.5	61.8	–
ERNIE 2.0* (L)	41.4	–	–
RoBERTa* (L)	41.0	–	–
BART (L)	40.6	**64.2**	10.0
CPT_g (L)	**42.0**	63.7	**10.7**

a) The small(base) version of mT5 has almost the same parameters as the base(large) version of other PTMs. CPM-2 has a much larger number of parameters than other large-size PTMs. Models with * and † indicate the results from [16] and [10], respectively. The bold font indicates the highest value.

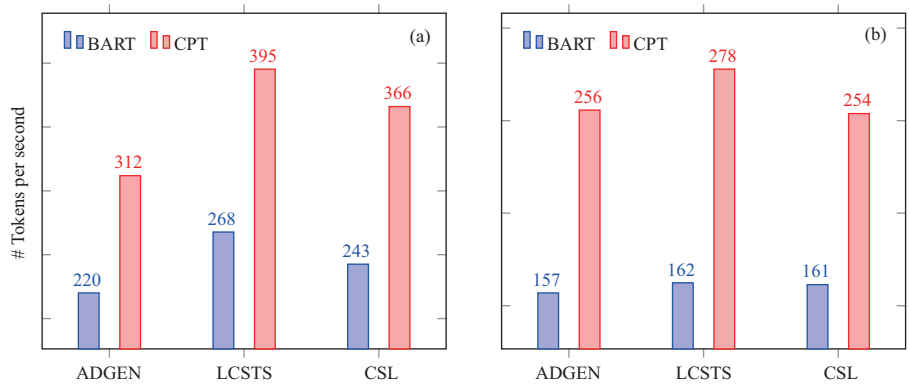

Figure 6 (Color online) Inference throughput for BART and CPT. It is measured on the same parts of datasets that the models are evaluated. The beam size is 4 and the batch size is 8. (a) Throughput of BART (B) and CPT (B); (b) throughput of BART (L) and CPT (L).

Text generation. Table 6 compares the performance of our model on generation datasets with other strong methods. The character-level Rouge-L is used to evaluate the summarization results. For ADGEN, we follow [10] to use BLEU-4.

In conclusion, CPT_g achieves competitive performance on text generation compared with other methods, such as mT5, CPM-2, and BART. In addition, compared with other pre-trained encoders (RoBERTa and ERNIE 2.0), CPT_g improves the generation score with the NLG enhanced pre-training. When compared with pre-trained mT5 and CPM-2, CPT_g acquires better results on both base and large versions. We assume the difference in pre-training tasks leads to the performance gaps. Both mT5 and CPM-2 exploit a T5 style masked span generation as their pre-training task, while CPT is pre-trained with DAE, which shows the effectiveness of DAE for text generation pre-training.

In addition, the shallow decoder of CPT_g may affect the performance of long text generation. However, the performance gaps are still small. And we believe the multi-task pre-training of CPT closes the gaps. Tables 7 and 8 illustrate some examples generated by BART (L) and CPT_g (L). With the help of pre-training for understanding, CPT_g is able to summarize text with more information captured in the input content.

Moreover, because of the shallow decoder, CPT could generate texts more efficiently (Figure 6), which could be faster than other depth symmetric encoder-decoder transformers with the same number of layers of the encoder and the decoder. As BART and CPT have a similar number of parameters in both base and large versions. On all generation dataset, the decoding speed of CPT surpass BART by a large margin. Our model achieves 1.4×–1.5× speedup compared with BART and still maintains comparable generation results in base size. And CPT (L) has up to 1.7× relative speedup compared to BART (L). As a conclusion, the shallow G-Dec is able to speed up the generation with minor performance loss.

Table 7 Summary examples generated by BART (L) and CPT (L) given input text on LCSTS

Input	今日, 刘胜义在 2013 腾讯智慧峰会上指出, 在移动化时代, 数字媒体、消费行为、数字营销都需要重新定义. 并且移动化媒体应具备三个特征: 从实时媒体发展成全天候媒体; 从大众媒体发展到智能媒体阶段; 从资讯媒体发展到生活类型的媒体. Today, in the 2013 Tencent Wisdom Summit, Shengyi Liu pointed out that in the mobile era, digital media, consumer behavior, and digital marketing all need to be redefined. And mobile media should have three characteristics: real-time media develope to 24-hour media; mass media develope to smart media; information and news media develope to life media.
Reference	腾讯刘胜义: 移动化引发媒体及营销体系变革 Shengyi Liu from Tencent: Mobile process leads to the changes in media and marketing systems.
BART (L)	刘胜义: 移动化时代数字媒体需重新定义 Shengyi Liu: Digital media need to be redefined in the mobile era.
CPT_g (L)	腾讯总裁刘胜义: 移动化时代数字媒体需重新定义 Tencent President Shengyi Liu: Digital media need to be redefined in the mobile era.
Input	近年来, 逢雨必涝、逢涝必瘫, 几成我国城市通病. 上周, 中国青年报对全国 31 个省 (区、市) 5375 人进行的调查显示, 91.6% 的人关注所在城市的排水问题; 84.7% 的受访者赞同, 城市现代化更表现在地面之下, 应加大地下民生工程建设投入. In recent years, flooding and paralysis in floods have become a common problem in Chinese cities. Last week, the China Youth Daily conducted a survey of 5375 people in 31 provinces (regions and cities) across the country. It shows that 91.6% of people are concerned about the drainage problems in their cities; 84.7% of the interviewees agree that urban modernization is shown under the ground, and the government should increase investment in the construction of underground livelihood projects.
Reference	84.7% 受访者期待国家加大地下民生工程投入 84.7% of respondents expect the country to increase investment in underground livelihood projects.
BART (L)	84.7% 受访者赞同加大地下民生工程建设投入 84.7% of respondents agree to increase investment in the construction of underground livelihood projects.
CPT_g (L)	超八成受访者赞同加大地下民生工程投入 Over 80% of respondents agree to increase investment in the underground livelihood projects.

Table 8 Text examples generated by BART (L) and CPT (L) given keywords on ADGEN

Input	[类型, 上衣], [版型, 宽松], [颜色, 蓝色], [风格, 简约], [风格, 清新], [衣样式, 衬衫], [衣领型, 翻领], [衣长, 中长款], [衣门襟, 单排扣] [Type, Top], [Fit, Loose], [Color, Blue], [Style, Simple], [Style, Fresh], [Clothing Style, Shirt], [Collar Type, Lapel], [Cloth Length , Mid-length], [Clothes placket, Single-Breasted]
Reference	很适合学生穿的一件**衬衫**, **蓝色**调是属于比较小**清新**的色调, 而且还能衬托出女性的完美好气色哦, 让你穿上之后瞬间有个好心情; **简约**的小**翻领**设计加以精致的**单排扣**点缀, 颇有一番正式的感觉, 结合上**宽松**以及**中长款**的设计, 从而打造出女性的率性 bf 风范. A **shirt** that is very suitable for students. The **blue** tone is a relatively **fresh** hue, which can bring out the perfect look of women, making you have a good mood instantly when wearing it; the **simple** small design of **lapel** is combined with **single-breasted** embellishment, which shows quite a formal feel. And when combined with **loose** and **mid-length** design, it creates a feminine and casual bf style.
BART (L)	这款**衬衫**采用**清新**的**蓝色**调, 展现出女性内心的纯洁与美好. **简约**的小**翻领**设计, 修饰出精致小巧的脸型. 精致的**单排扣**门襟, 方便穿脱. **宽松**的**中长款**版型, 对身材的包容性很好. This **shirt** uses a **fresh blue** tone to show the purity and beauty of women's hearts. The **simple** design of small **lapel** makes the face delicate and small. The exquisite **single-breasted** placket makes it easy to put on and take off. The **loose mid-length** clothing design is easily fit the body.
CPT_g (L)	这款**衬衫**采用**中长款**的版型设计, 搭配**宽松**的直筒版型, 给身体更多的活动量, 行动更为方便, 穿着更为舒适. 加上**清新**的**蓝色**调, 更为衬托肤色白皙. 结合**简约**的**单排扣**门襟, 方便穿脱的同时, 也丰富了整体的层次感. 而精致的小**翻领**, 则为整体增添了几分干练的气质. This **shirt** adopts a **mid-length** clothing design with a **loose** straight type, giving the body more activity, making it more convenient to move and to wear. Coupled with a **fresh blue** tone, it whitens the skin. Combined with a **simple single-breasted** placket, it is convenient to put on and take off, which also enriches the overall sense of layers. The exquisite and small **lapel** adds a capable and experienced style to the whole.
Input	[类型, 上衣], [版型, 显瘦], [颜色, 黑色], [衣样式, 毛衣], [衣领型, v 领] [Type, Top], [Fit, Skinny], [Color, Black], [Clothing Style, Sweater], [Collar Type, V-Neck]
Reference	金丝的加入为低调内敛的**黑色**添加闪闪亮色, 胸前的 **v 领**加排扣设计, 摈弃**毛衣**臃肿松散的固有印象, 轻薄**修身**, 保留温度的同时又不至于在猝不及防的倒春寒中失去风度. The golden silk adds sparkling color to the low-key and restrained **black**. The **v-neck** on the chest and the breasted design abandons the inherent impression of bloated and loose **sweaters**, which is light and **slim**. It keeps the temperature without losing its grace in the unexpected cold spring.
BART (L)	这款**黑色**的**修身**长款**毛衣**, 最大的设计亮点在于衣身 **v 领**设计, 这样的款式设计使得整件毛衣看起来与众不同, 既个性又修身舒适哦. The biggest highlighted design of this **black slim** long **sweater** is the **v-neck** design of the clothing. This design makes the whole sweater unique, personal and comfortable.
CPT_g (L)	这款**毛衣**采用了经典的**黑色**作为底色设计, 搭配上**修身**的版型剪裁, 穿起来更加的**显瘦**. 领口处采用 **v 领**的剪裁方式, 可以起到修饰脸型的作用, 更显脸小精致. 衣摆处的开叉处理, 更是增添了几分随性的感觉. This **sweater** uses a classic **black** background with a **slim** fit cut, which makes you look thin. The neckline adopts a **v-neck** tailoring method, which can frame the face and make the face small and delicate. The split treatment at the hem adds a casual feel.

7 Conclusion

In this paper, we propose CPT, a novel Chinese PTM for both language understanding and generation. With the flexible design, CPT can be assembled and disassembled in various fashions, which could fully exploit the potential of CPT. Experimental results on a wide range of Chinese NLU and NLG tasks show the effectiveness of CPT.

In future work, we will introduce more specific designs according to Chinese properties, such as better tokenization, pre-training tasks, and model architectures.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2020AAA0108702) and National Natural Science Foundation of China (Grant No. 62022027).

References

1 Qiu X P, Sun T X, Xu Y G, et al. Pre-trained models for natural language processing: a survey. Sci China Tech Sci, 2020, 63: 1872–1897
2 Devlin J, Chang M, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019. 4171–4186
3 Liu Y, Ott M, Goyal N, et al. RoBERTa: a robustly optimized BERT pretraining approach. 2019. ArXiv:1907.11692
4 Lewis M, Liu Y, Goyal N, 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, 2020. 7871–7880
5 Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training. 2018. https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf
6 Cui Y M, Che W X, Liu T, et al. Pre-training with whole word masking for Chinese BERT. 2019. ArXiv:1906.08101
7 Sun Y, Wang S H, Li Y K, et al. ERNIE: enhanced representation through knowledge integration. 2019. ArXiv:1904.09223
8 Wei J Q, Ren X Z, Li X G, et al. NEZHA: neural contextualized representation for Chinese language understanding. 2019. ArXiv:1909.00204
9 Zhang Z, Han X, Zhou H, et al. CPM: a large-scale generative Chinese pre-trained language model. AI Open, 2021, 2: 93–99
10 Zhang Z, Gu Y, Han X, et al. CPM-2: large-scale cost-effective pre-trained language models. AI Open, 2021, 2: 216–224
11 Zeng W, Ren X Z, Su T, et al. Pangu-α: large-scale autoregressive pretrained Chinese language models with auto-parallel computation. 2021. ArXiv:2104.12369
12 Dong L, Yang N, Wang W H, et al. Unified language model pre-training for natural language understanding and generation. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019. 13063–13075
13 Bao H B, Dong L, Wei F R, et al. UniLMv2: pseudo-masked language models for unified language model pre-training. In: Proceedings of the 37th International Conference on Machine Learning, 2020. 642–652
14 Du Z X, Qian Y J, Liu X, et al. All NLP tasks are generation tasks: a general pretraining framework. 2021. ArXiv:2103.10360
15 Bi B, Li C L, Wu C, et al. PALM: pre-training an autoencoding&autoregressive language model for context-conditioned generation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2020. 8681–8691
16 Sun Y, Wang S H, Feng S K, et al. ERNIE 3.0: large-scale knowledge enhanced pre-training for language understanding and generation. 2021. ArXiv:2107.02137
17 Diao S Z, Bai J X, Song Y, et al. ZEN: pre-training Chinese text encoder enhanced by n-gram representations. In: Proceedings of the Findings of the Association for Computational Linguistics, 2020. 4729–4740
18 Sun Y, Wang S H, Li Y K, et al. ERNIE 2.0: a continual pre-training framework for language understanding. In: Proceedings of the AAAI Technical Track: Natural Language Processing, 2020. 8968–8975
19 Sun Z J, Li X Y, Sun X F, et al. Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. 2021. ArXiv:2106.16038
20 Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res, 2020, 21: 5485–5551
21 Dou Z, Liu P F, Hayashi H, et al. GSum: a general framework for guided neural abstractive summarization. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021. 4830–4842
22 Liu Y X, Liu P F. SimCLS: A simple framework for contrastive learning of abstractive summarization. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, 2021. 1065–1072
23 Lin Z J, Madotto A, Winata G I, et al. MinTL: minimalist transfer learning for task-oriented dialogue systems. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020. 3391–3405
24 Liu X, He P, Chen W, et al. Multi-task deep neural networks for natural language understanding. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, 2019. 4487–4496
25 Aghajanyan A, Gupta A, Shrivastava A, et al. Muppet: massive multi-task representations with pre-finetuning. 2021. ArXiv:2101.11038
26 Wei J, Bosma M, Zhao V Y, et al. Finetuned language models are zero-shot learners. 2021. ArXiv:2109.01652
27 Kasai J, Pappas N, Peng H, et al. Deep encoder, shallow decoder: reevaluating non-autoregressive machine translation. In: Proceedings of International Conference on Learning Representations, 2021
28 Sun X, Ge T, Wei F R, et al. Instantaneous grammatical error correction with shallow aggressive decoding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, 2021. 5937–5947
29 Schick T, Schütze H. It's not just size that matters: small language models are also few-shot learners. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021. 2339–2352
30 Gao T Y, Fisch A, Chen D Q. Making pre-trained language models better few-shot learners. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, 2021. 3816–3830
31 Liu P F, Yuan W Z, Fu J L, et al. Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. 2021. ArXiv:2107.13586

32 Xu L, Hu H, Zhang X W, et al. CLUE: a Chinese language understanding evaluation benchmark. In: Proceedings of the 28th International Conference on Computational Linguistics, 2020. 4762–4772
33 Zhang X S, Li P S, Li H. AMBERT: a pre-trained language model with multi-grained tokenization. 2020. ArXiv:2008.11869
34 Emerson T. The second international Chinese word segmentation bakeoff. In: Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing, 2005
35 Levow G. The third international Chinese language processing bakeoff: word segmentation and named entity recognition. In: Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing, 2006. 108–117
36 Li X N, Shao Y F, Sun T X, et al. Accelerating BERT inference for sequence labeling via early-exit. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics, 2021. 189–199
37 Li X B, Yan H, Qiu X P, et al. FLAT: Chinese NER using flat-lattice transformer. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020. 6836–6842
38 Qiu X P, Pei H Z, Yan H, et al. A concise model for multi-criteria Chinese word segmentation with transformer encoder. In: Proceedings of the Findings of the Association for Computational Linguistics, 2020. 2887–2897
39 Cui Y M, Liu T, Che W X, et al. A span-extraction dataset for Chinese machine reading comprehension. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2019. 5882–5888
40 Shao C C, Liu T, Lai Y T, et al. DRCD: a Chinese machine reading comprehension dataset. 2018. ArXiv:1806.00920
41 Cui Y M, Che W X, Liu T, et al. Revisiting pre-trained models for Chinese natural language processing. In: Proceedings of the Findings of the Association for Computational Linguistics, 2020. 657–668
42 Hu B T, Chen Q C, Zhu F Z. LCSTS: a large scale Chinese short text summarization dataset. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2015. 1967–1972
43 Shao Z H, Huang M L, Wen J T, et al. Long and diverse text generation with planning-based hierarchical variational model. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2019. 3255–3266
44 Liu Y, Gu J, Goyal N, et al. Multilingual denoising pre-training for neural machine translation. Trans Assoc Comput Linguist, 2020, 8: 726–742
45 Xue L T, Constant N, Roberts A, et al. mT5: a massively multilingual pre-trained text-to-text transformer. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021. 483–498