AI Open 2 (2021) 93–99

Contents lists available at ScienceDirect

AI Open

journal homepage: www.sciencedirect.com/journal/ai-open

CPM: A large-scale generative Chinese Pre-trained language model

Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, Fanchao Qi, Xiaozhi Wang, Yanan Zheng, Guoyang Zeng, Huanqi Cao, Shengqi Chen, Daixuan Li, Zhenbo Sun, Zhiyuan Liu *, Minlie Huang **, Wentao Han, Jie Tang, Juanzi Li, Xiaoyan Zhu, Maosong Sun

Department of Computer Science and Technology, Tsinghua University & BAAI, China

ARTICLE INFO

Keywords:
Pre-trained language model
Zero-shot learning

ABSTRACT

Pre-trained Language Models (PLMs) have proven to be beneficial for various downstream NLP tasks. Recently, GPT-3, with 175 billion parameters and 570 GB training data, drew a lot of attention due to the capacity of few-shot (even zero-shot) learning. However, applying GPT-3 to address Chinese NLP tasks is still challenging, as the training corpus of GPT-3 is primarily English, and the parameters are not publicly available. In this technical report, we release the Chinese Pre-trained Language Model (CPM) with generative pre-training on large-scale Chinese training data. To the best of our knowledge, CPM, with 2.6 billion parameters and 100 GB Chinese training data, is the largest Chinese pre-trained language model, which could facilitate several downstream Chinese NLP tasks, such as conversation, essay generation, cloze test, and language understanding. Extensive experiments demonstrate that CPM achieves strong performance on many NLP tasks in the settings of few-shot (even zero-shot) learning. The code and parameters are available at https://github.com/TsinghuaAI/CPM.

1. Introduction

Pre-trained Language Models (PLMs) (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019; Brown et al., 2020) have been developed for a variety of tasks in Natural Language Processing (NLP), as they can learn rich language knowledge from large-scale corpora, which is beneficial for downstream tasks. ELMo (Peters et al., 2018) first introduces bidirectional language models to learn contextual word vectors via large-scale pre-training. GPT (Radford et al., 2018) applies generative pre-training to a Transformer-based language model (Vaswani et al., 2017), which improves natural language understanding on a wide range of benchmarks. BERT (Devlin et al., 2019) is proposed to pre-train deep bidirectional representations on unlabeled texts by jointly conditioning on both left and right contexts. RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2020) enhance BERT (Devlin et al., 2019) by dynamic masking, parameter sharing and modifying pre-training tasks. ERNIE (Zhang et al., 2019), KEPLER (Wang et al., 2019) and SentiLARE (Ke et al., 2020) introduce external knowledge to language representation learning by auxiliary pre-training tasks.

Among these PLMs, GPT-3 (Brown et al., 2020), with 175 billion parameters and 570 GB training data, has been the center of attention and proven to be effective in various few-shot (even zero-shot) NLP tasks. The powerful text generation capability of GPT-3 makes it available to diverse applications, such as question answering, summarization, conversation, computing basic arithmetic, and generating kinds of text, including essay, fiction, code, spreadsheets, etc. However, incorporating GPT-3 to address Chinese NLP tasks is still challenging, as the training corpus of GPT-3 is primarily English (93% by word counting as reported by Brown et al. (Brown et al. (2020)), and the parameters are not publicly available. Although there are some previous works providing powerful Chinese pre-trained language models (Cui et al., 2019a, 2020; Xu et al., 2020; Wei et al., 2019; Sun et al., 2019), their capabilities are limited due to the model size. Hence, how to pre-train a large-scale Chinese language model needs more exploration, such as the construction of Chinese vocabulary and the design of the training strategy.

In this technical report, we release the **C**hinese **P**re-trained Language **M**odel (**CPM**) with generative pre-training on large-scale Chinese corpora. CPM is a Transformer-based autoregressive language model, with 2.6 billion parameters and 100 GB Chinese training data. To the best of our knowledge, CPM is the largest Chinese pre-trained language

* Corresponding author.
** Corresponding author.
E-mail addresses: liuzy@tsinghua.edu.cn (Z. Liu), aihuang@tsinghua.edu.cn (M. Huang).

https://doi.org/10.1016/j.aiopen.2021.07.001
Received 25 March 2021; Accepted 9 July 2021
Available online 15 July 2021

model, which could facilitate downstream Chinese NLP tasks, such as conversation, essay generation, cloze test, and language understanding. Experiments on various Chinese NLP tasks demonstrate that CPM achieves strong performance on many NLP tasks in the few-shot (even zero-shot) settings. With the increase of parameters, CPM performs better on most datasets, indicating that larger models are more proficient at language generation and language understanding.

The main contributions of this technical report can be summarized as follows:

- We release a Chinese autoregressive language model with generative pre-training, called CPM, which has 2.6 billion parameters.
- We construct a new sub-word vocabulary based on the word segmented corpus to adapt for Chinese corpora and increase the batch size to 3, 072 for more stable model training.
- Extensive experiments demonstrate that CPM achieves strong performance on many NLP tasks in the few-shot (even zero-shot) settings.

2. Our approach

2.1. Chinese PLM

Our current model is a left-to-right Transformer decoder, which is similar to the model architecture of GPT (Radford et al., 2019). We pre-train three models with different sizes, as shown in Table 1. In order to adapt CPM to Chinese corpora, we build a new sub-word vocabulary and adjust the training batch size.

Vocabulary Construction: Previous works on Chinese pre-trained models usually adopt the sub-word vocabulary of BERT-Chinese (Devlin et al., 2019), which would split the input text to a character-level sequence. However, Chinese words usually contain several characters, and some important semantic meanings of words would be lost in the character-level sequence. To solve this problem, we construct a new sub-word vocabulary, containing both words and characters. For example, some common words would be added to the vocabulary.

Training Strategy: Since the sparseness of word distributions of Chinese is more serious than that of English, we adopt a large batch size to make the model training more stable. Compared to the batch size (1 million tokens) used in GPT-3 2.7B (Brown et al., 2020), our batch size (3 million tokens) is two times larger. For the largest model, which cannot be stored in a single GPU during training, we partition the model across GPUs along the width dimension to make the large-scale training available and reduce data-transfer among nodes.

2.2. Data processing

Specifically, we construct a new sub-word vocabulary based on the word segmented corpus using *unigram language model* (Kudo et al., 2018). Meanwhile, considering that the word segmentation introduces extra splitters between words, we set a special token as the splitter to make the sub-word process reversible. If not, the splitters will be missed in both encoding and decoding. In contrast, the tokenizer of BERT-Chinese is irreversible because it will insert extra spaces between Chinese characters and treat the extra spaces as the same as the original spaces in the text.

We collect different kinds of texts in our pre-training, including encyclopedia, news, novels, and Q&A. The details of our training data are shown in Table 2. Since the input sequence length is usually larger than that of a single document, we concatenate different documents together by adding "end of document" token after each document to make full use of the input length.

2.3. Pre-training details

Based on the hyper-parameter searching on the learning rate and batch size, we set the learning rate as 1.5×10^{-4} and the batch size as 3, 072, which makes the model training more stable. In the first version, we still adopt the dense attention and the max sequence length is 1, 024. We will implement sparse attention in the future. We pre-train our model for 20, 000 steps, and the first 5, 000 steps are for warm-up. The optimizer is Adam (Kingma et al., 2015). It takes two weeks to train our largest model using 64 NVIDIA V100.

3. Experiments

3.1. Text classification

Dataset: We use TouTiao News Titles Classification (TNEWS), IFLYTEK app description classification (IFLYTEK), and Original Chinese NLI (OCNLI) as our benchmark datasets for text classification (Xu et al., 2020; Hu et al., 2020). Since we aim to evaluate the zero-shot ability of CPM on text classification tasks, we directly use the validation sets of these three datasets without any training instance. The amount of the validation set of TNEWS/IFLYTEK/OCNLI is 10K/2.6K/3K. Note that, we exclude the instances with the label "-" in OCNLI.

Implementation Details: We calculate the perplexity of each candidate sentence-label pair and treat the pair having the lowest perplexity as the prediction. The templates of these three tasks are formulated by.

$$
\begin{aligned}
\text{TNEWS:}\ & \text{这是关于}\underline{L}\text{的文章：}\ \underline{P} \\
& (\text{This passage is about } \underline{L}\text{: } \underline{P}), \\
\text{IFLYTEK:}\ & \text{这是关于}\underline{L}\text{的程序：}\ \underline{P} \\
& (\text{This program is about } \underline{L}\text{: } \underline{P}), \\
\text{OCNLI:}\ & \underline{S_1}\text{？对，}\ \underline{S_2}\ (\underline{S_1}\text{? Yes, } \underline{S_2}), \\
& \underline{S_1}\text{？错，}\ \underline{S_2}\ (\underline{S_1}\text{? No, } \underline{S_2}), \\
& \underline{S_1}\text{？也许，}\ \underline{S_2}\ (\underline{S_1}\text{? Maybe, } \underline{S_2}),
\end{aligned}
$$

Where L is the label name, P is the input text, S_1 and S_2 are the premise and hypothesis. These templates are designed by ourselves but they are not optimal.

Since TNEWS and IFLYTEK have more than 10 kinds of labels, we adopt a simpler validation setting, which randomly samples 3 false labels for each instance and performing 4-class classification for better efficiency. To make it more stable, we repeat it three times and report

Table 1
Model sizes. n_{param} is the number of parameters. n_{layers} is the number of layers. d_{model} is the dimension of hidden states, which is consistent in each layer. n_{heads} is the number of attention heads in each layer. d_{head} is the dimension of each attention head.

	n_{param}	n_{layers}	d_{model}	n_{heads}	d_{head}
CPM-Small	109M	12	768	12	64
CPM-Medium	334M	24	1024	16	64
CPM-Large	2.6B	32	2560	32	80

Table 2
Details of training data.

Data Source	Encyclopedia	Webpage	Story	News	Dialogue
Size	~ 40 GB	~ 39 GB	~ 10 GB	~ 10 GB	~ 1 GB

the averaged results. For OCNLI, which only has 3 kinds of labels, we hold the original validation set. However, the validation set of OCNLI is unbalanced, where the amount of "entailment"/"neutral"/"contradiction" is 947/1103/900. If the model only predicts the label "neutral", the accuracy is about 0.374.

Table 3
Zero-shot performance on text classification tasks (accuracy). Random prediction would have 0.25 on TNEWS and IFLYTEK, 0.33 on OCNLI.

	TNEWS	IFLYTEK	OCNLI
CPM-Small	0.677	0.718	0.378
CPM-Medium	0.673	0.724	0.379
CPM-Large	**0.688**	**0.735**	**0.442**

Results: As shown in Table 3, CPM-large achieves promising results on these classification datasets without any training samples. Compared to random prediction, the knowledge learned from pre-training significantly improves the performance. Although the medium model is three times as large as the small model, the performances on TNEWS and OCNLI are very close. However, CPM-Large significantly outperforms these two smaller models on all three datasets. It indicates that the magic of the model size is not linear and would happen when the model size exceeds a specific boundary. Besides, the results of CPM-small and CPM-medium on OCNLI are close to that of the strategy only predicting the label "neutral". It suggests that natural language inference is harder than other downstream tasks in the setting of zero-shot learning, which is consistent with the observation in Brown et al. (2020).

3.2. Chinese idiom cloze

Dataset: We use the Chinese IDiom cloze test dataset (ChID) (Zheng et al., 2019) as our benchmark dataset. Each passage in the dataset may contain multiple blanks. For each blank, there are 10 candidate idioms with 1 golden truth. Some of the false candidates are similar to the answer in meanings. The amount of training/validation/test set is 520K/20K/20K.

Implementation Details: For the supervised setting, we use a template to convert the passage and the candidates to a natural language question. Given the passage P and 10 candidate idioms $I_1, I_2, \ldots, I_{10}$, the template can be formulated as.

$$\text{选项1: } \underline{I_1} \text{ ... 选项10: } \underline{I_{10}}\ \underline{P}\ \text{答案是:}\underline{L}$$
$$\text{(Option 1: } \underline{I_1} \text{ ... Option 10: } \underline{I_{10}}\ \underline{P}\ \text{Answer:}\underline{L}\text{).}$$

Then, we train the model to predict the answer L. Note that if there exists more than one idiom in a passage, we predict each one independently. Specifically, When we are predicting one idiom, we leave its blank in the passage and remove the blanks of other idioms from the passage.

For the unsupervised setting, we fill the candidate idioms into the blank to form a group of complete passages. We also consider each idiom blank individually if there are multiple blanks in a passage. For each blank, we can get 10 passages corresponding to the 10 candidate idioms. Then we calculate the perplexity of each passage and treat the one with the lowest perplexity as the prediction.

Table 4
Results on ChID dataset in the supervised and unsupervised settings. The random prediction would have 0.10 in the unsupervised setting.

	Supervised	Unsupervised
CPM-Small	0.657	0.433
CPM-Medium	0.695	0.524
CPM-Large	**0.804**	**0.685**

Table 5
Results on STC dataset in the few-shot and supervised settings.

	Average	Extrema	Greedy	Dist-1	Dist-2
Few-shot (Unsupervised)					
CDial-GPT	0.899	0.797	0.810	1963/**0.011**	20,814/0.126
CPM-Large	**0.928**	**0.805**	**0.815**	**3229**/0.007	**68,008/0.154**
Supervised					
CDial-GPT	0.933	**0.814**	**0.826**	2468/0.008	35,634/0.127
CPM-Large	**0.934**	0.810	0.819	**3352/0.011**	**67,310/0.233**

Results: The results are shown in Table 4. We report the accuracy on the test set of each model. For the fully supervised setting, we can see that CPM can be fine-tuned for the specific input template, solving multiple-choice tasks by uni-direction auto-regressive language modeling. In our experiments, we didn't take much time to design the input template for this task, and thus there might exist better templates that can help the model to show its full ability. We will leave this part as future work. For the unsupervised setting, we can see that CPM produces promising results. The unsupervised result of CPM-Large even outperforms the result of CPM-Small and is comparable to CPM-Medium in the supervised setting, reflecting the strong power of CPM in Chinese language modeling.

3.3. Dialogue generation

Dataset: We use Short-Text Conversation (STC) (Shang et al., 2015) as our benchmark dataset for dialogue generation, which consists of post-response pairs from Weibo. We adopt the same data split as the existing work (Wang et al., 2020). The amount of training/validation/test set is 4.4M/20K/20K, respectively. The average length of posts/responses is 20.6/15.4.

Baseline: We choose CDial-GPT (Wang et al., 2020) as our baseline, which is the state-of-the-art pre-trained model for Chinese dialogue generation. We directly use the codes and the pre-trained model released by the original paper.

Implementation Details: In the supervised experiment, we utilize a similar hyper-parameter setting as pre-training and fine-tune CPM on the training set of STC. In the few-shot experiment which doesn't include the fine-tuning process, we follow the existing work (Radford et al., 2019; Brown et al., 2020) to condition the language model on a context of 4 examples pairs of the format *Context: sentence Response: sentence*. Note that the examples are randomly sampled. After a final prompt *Context: sentence Response:*, we acquire the generation results with Top-p sampling (Holtzman et al., 2020), where p is set to 0.9. The temperature of sampling is 0.9 in both few-shot and supervised experiments.

Metrics: Since BLEU is not a proper metric for dialogue generation, we use embedding-based metrics (including greedy matching, embedding average, and vector extrema) to evaluate the similarity between generated responses and references (Liu et al., 2016). For diversity, we choose the number and proportion of distinct n-grams (Li et al., 2016; Xing et al., 2017; Ke et al., 2018) as our metric.

Table 6
Examples of generated responses on STC.

Post:	现在高中数学竞赛这么凶残了！ How crazy the math competition of high school is!	每天嘻嘻哈哈的笑着，只有自己才知道活的有多累。 I laugh every day, but only I know how tired I am.
CDial-GPT: (Few-shot)	你还真信? Do you really believe it?	我也是 Me, too.
CDial-GPT: (Supervised)	数学让我们学会了很多，好好学，明天回家 Math lets us learn a lot. Study hard and go home tomorrow.	只有自己才知道活的有多累。 Only I know how tired I am.
CPM-Large: (Few-shot)	现在都是微积分,大学的东西比高中简单多了,还有什么呢? Now it's all calculus. The things in college are much simpler than those of high school. What else?	当你想要哭泣的时候,就去那些熟悉的地方吧。 When you want to cry, go to those familiar places.
CPM-Large: (Supervised)	这是要逼死我吗?每道题都不会做啊 Is this going to kill me? I can't answer any question.	真的不知道,生活怎么会这么累 I really don't know how my life could be so tiring.

Table 7
Results of CPM with different amounts of parameters on STC dataset in the few-shot setting.

	Average	Dist-1	Dist-2
CPM-Small	**0.928**	2201/0.004	22,754/0.046
CPM-Medium	0.910	2842/0.005	31,934/0.058
CPM-Large	**0.928**	**3229/0.007**	**68,008/0.154**

Table 8
Zero-shot (zs) and one-shot (os) results on Question Answering (QA) datasets, including DuReader (Zhidao and Search) and CMRC 2018, we did experiments on models with three different sizes: small (s), medium (m) and large(l).

	Zhidao		Search		CMRC2018	
	F1	EM	F1	EM	F1	EM
s + zs	4.01	0.18	4.15	0.65	6.03	0.20
s + os	4.75	0.34	4.45	0.59	6.14	0.22
m + zs	5.29	0.29	5.03	0.61	8.60	0.53
m + os	5.76	0.47	5.14	0.55	9.00	0.75
l + zs	5.18	0.27	5.08	0.59	13.37	1.31
l + os	**6.08**	**0.56**	**5.19**	**0.68**	**16.56**	**3.73**

Results: We present the main results in the few-shot and supervised settings in Table 5. We can see that CPM outperforms CDial-GPT with a large margin in the few-shot experiment, showing the generalization ability of our model. As for the supervised experiment, our model still performs better, especially on the diversity metrics. Since fine-tuning large pre-trained models on the supervised downstream tasks is often challenging (Dodge et al., 2020; Mosbach et al., 2020; Lee et al., 2020), we leave how to further improve the performance in the supervised setting as future work. Some cases are provided in Table 6 to intuitively show the effectiveness of our model.

We also conduct experiments to show the few-shot performance of CPM with different parameter sizes in Table 7. As the number of parameters grows, CPM can generate more diverse responses with reasonable values on the embedding-based metrics.

3.4. Question answering

Dataset: We adopt CMRC 2018 (Cui et al., 2019b) and DuReader (He et al., 2018) as our benchmark for Question Answering (QA). CMRC2018 requires the model to extract an answer span from a Wikipedia passage for the given question, which is similar to SQuAD (Rajpurkar et al., 2016). DuReader consists of questions from real-world user logs from Baidu Search and Baidu Zhidao. The answers in DuReader are manifold, such as an entity or a description. We treat DuReader as an extractive QA task and thus ignore those instances with yes-or-no answers during evaluation.

Implementation Details: We evaluate CPM on zero-shot (zs) and one-shot (os) setting and report F1 score (F1) and Exact Match (EM) for both CMRC2018 and DuReader. For the zero-shot setting, we concatenate the passage and question as input to CPM, and CPM is then required to generate an answer according to the observed (passage, question) pair. For the one-shot setting, we randomly select a ground truth triple (passage, question, answer) in the training set and insert it to the front of the instance to be treated as a hint for CPM to generate the answer.

Results: As shown in Table 8, we perform the experiments on three datasets and compare models with different sizes: small (s), medium (m) and large (l). From the table, we can see that with the size growing, CPM is performing better. Among all the models, large is always the best. And, the results in the one-shot setting are better than those in the zero-shot setting. We guess CPM is able to imitate the format in previous sequences and organize the language accordingly. We also analyze the generated answer and find that CPM prefers to generate long and repetitive sentences instead of a short and precise one, which results in low scores. We believe it is worth exploring how to make CPM generate brief and proper answers in the future. In general, CPM does not achieve very high scores in either benchmark. We guess it's related to the format of the pre-training data.

Table 9
BLEU-1 results of CPM with different amounts of parameters on XLORE dataset in the few-shot setting.

	$N = 2$			$N = 4$		
CPM	Small	Medium	Large	Small	Medium	Large
主要工艺 (Main Process)	0.500	0.500	**0.700**	0.400	0.200	**0.400**
释义 (Explanation)	0.000	0.000	**0.071**	0.000	0.000	**0.075**
商品品牌 (Brand)	0.098	0.033	**0.483**	0.183	0.050	**0.450**
学科 (Subject)	0.000	0.025	**0.124**	0.059	0.053	**0.108**
全名 (Full Name)	0.035	0.010	**0.108**	0.000	0.014	**0.122**
涉及领域 (Related Field)	0.042	0.065	**0.104**	0.063	0.037	**0.125**
主要作物 (Main Crop)	0.000	**0.150**	0.050	0.100	**0.150**	0.100
所在国家 (In Country)	0.033	0.033	0.033	**0.050**	0.000	**0.050**
病原类型 (Pathogen Type)	0.250	0.220	**0.370**	0.200	0.300	**0.340**
首任总统 (The First President)	0.000	0.000	0.000	**0.016**	0.009	0.014

Table 10
Examples of generated entities on XLORE with CPM-Large.

Relation:	首都 (Capital)
Prompt:	美国 首都 华盛顿 America Capital Washington 中国 首都 北京 China Capital Beijing 日本 首都 Japan Capital
CPM:	东京 Tokyo
Relation:	主要工艺 (Main Process)
Prompt:	酱焖辣椒 主要工艺 焖 (Sauce Braised Chili) (Main Process) Stew 当归鸭肉煲 主要工艺 煲 (Duck with Angelica) (Main Process) Boil 韭菜煎蛋饼 主要工艺 (Leek Omelette) (Main Process)
CPM:	煎 Fried
Relation:	学科 (Subject)
Prompt:	恒星级黑洞 学科 宇宙论 (Stellar Black Hole) Subject Cosmology 品类需求强度 学科 品牌经济学 (Category Demand Intensity) Subject Economics 大地构造学 学科 (Tectonic Geology) Subject
CPM:	地质学 Geology

3.5. Entity generation

Dataset: We use XLORE, which includes 446,236 relations and 16,284,901 entities, as our benchmark dataset for entity generation. These relations and entities are from Wikipedia and Baidu Baike.

Implementation Details: We evaluate CPM on the few-shot setting with different amounts of parameters and report BLEU-1 results because there are several possible answers and they are similar to each other. In detail, we randomly select triples (head entity, relation, tail entity) by the same relations from XLORE and combine N triples and an incomplete triple (head entity, relation) into a prompt. Then, given the prompt, the models need to predict the corresponding tail entity. Each complete triple is ended with a special token so when the model generates the special token we will stop the generation for the incomplete triple.

Results: We present the results in Table 9. As we can see from the table, CPM-large achieves the best performance among these three models. Surprisingly, given a prompt with two triples, CPM can achieve comparable results to that with four triples. It indicates that CPM can imitate the format and probe factual knowledge to generate a proper tail entity in the extreme few-shot scenarios. We also provide some cases in Table 10 to demonstrate the ability of CPM.

4. Conclusion and Future Work

In this paper, we explore to train a large-scale generative language model on Chinese corpora and release CPM, which is a pre-trained model with 2.6 billion parameters. Experimental results show that CPM excel in several downstream Chinese NLP tasks, including conversation, essay generation, and language understanding.

In the future, we will further explore the power of large-scale pre-trained models on Chinese by adding more training data and increasing the model size. Due to the extremely expensive cost of pre-training, we will try to optimize the training framework, such as the data-transfer scheme between different nodes, to accelerate the process. There are some previous works including LAMB (You et al., 2020) and DeepSpeed (Rasley et al., 2020). Besides, it is important to reduce the model size by model compression (Sanh et al., 2019; Jiao et al., 2019; Zhang et al., 2020).

Meanwhile, we will also include diverse data to enhance model performance. For text data, we will add a multi-lingual corpus to train a large-scale Chinese-centered multi-lingual language model. For structured data such as knowledge graphs, which is important for PLMs (Peters et al., 2019; Xiong et al., 2020; Su et al., 2020), we will explore new learning algorithms to train a joint model, which can learn from both texts and knowledge graphs for better general intelligence.

5. Disclaimer of warranties

The text generated by CPM is automatically generated by a neural network model trained on a large number of texts, which does not represent our official attitudes and preferences. The text generated by CPM is only used for technical and scientific purposes. If it infringes on your rights and interests or violates social morality, please do not propagate it, but contact us and we will deal with it promptly.

Contributions

Zhengyan Zhang, Xu Han, and Hao Zhou implemented the large-scale models and model-parallel strategies.

Huanqi Cao, Shengqi Chen, Daixuan Li, and Zhenbo Sun built the training infrastructure.

Pei Ke, Deming Ye, Jian Guan, Fanchao Qi, and Xiaozhi Wang collected, filtered, deduplicated the training data.

Zhengyan Zhang, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, and Haozhe Ji implemented the downstream tasks and the software framework for supporting them.

Hao Zhou, Guoyang Zeng, Xu Han, and Yanan Zheng implemented the demos of language generation and knowledge retrieval using our CPM.

Guoyang Zeng conducted the human evaluations of the model.

Hao Zhou, Zhengyan Zhang, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, and Yusheng Su wrote the paper.

Zhiyuan Liu, Minlie Huang, and Wentao Han designed and led the research.

Jie Tang, Juanzi Li, Xiaoyan Zhu, Maosong Sun provided valuable advices to the research.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2020AAA0106501). Thanks to the Beijing Academy of Artiicial Intelligence (BAAI) for providing the computing resources and web services of this work. In addition, we would like to thank NetEase Inc., zhihu.com, and aminer.cn for the support in collecting the Chinese corpus.

References

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language Models Are Few-Shot Learners, p. 14165 arXiv preprint arXiv:2005.

Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z., Wang, S., Hu, G., 2019a. Pre-training with Whole Word Masking for Chinese Bert arXiv preprint arXiv:1906.08101.

Cui, Y., Liu, T., Che, W., Xiao, L., Chen, Z., Ma, W., Wang, S., Hu, G., 2019b. A span-extraction dataset for Chinese machine reading comprehension. In: Inui, K., Jiang, J., Ng, V., Wan, X. (Eds.), Proceedings of EMNLP.

Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., Hu, G., 2020. Revisiting pre-trained models for Chinese natural language processing. Findings of EMNLP.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT.

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., Smith, N., 2020. Fine-tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping arXiv preprint arXiv:2002.06305.

He, W., Liu, K., Liu, J., Lyu, Y., Zhao, S., Xiao, X., Liu, Y., Wang, Y., Wu, H., She, Q., Liu, X., Wu, T., Wang, H., 2018. DuReader: a Chinese machine reading comprehension dataset from real-world applications. In: Choi, E., Seo, M., Chen, D., Jia, R., Berant, J. (Eds.), Proceedings of ACL Workshop.

Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y., 2020. The curious case of neural text degeneration. Proceedings of ICLR.

Hu, H., Richardson, K., Xu, L., Li, L., Kuebler, S., Moss, L.S., 2020. Ocnli: Original Chinese Natural Language Inference arXiv preprint arXiv:2010.05444.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., Liu, Q., 2019. TinyBERT: Distilling Bert for Natural Language Understanding arXiv preprint arXiv: 1909.10351.

Ke, P., Guan, J., Huang, M., Zhu, X., 2018. Generating informative responses with controlled sentence function. Proceedings of ACL 1499–1508.

Ke, P., Ji, H., Liu, S., Zhu, X., Huang, M., 2020. SentiLARE: sentiment-aware language representation learning with linguistic knowledge. In: Proceedings of EMNLP.

Kingma, D.P., Ba, J., Adam, 2015. A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations.

Kudo, T., Richardson, J., Sentencepiece, 2018. A simple and language independent subword tokenizer and detokenizer for neural text processing. Proceedings of EMNLP 66–71.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R., 2020. ALBERT: a lite bert for self-supervised learning of language representations. In: Proceedings of ICLR.

Lee, C., Cho, K., Kang, W., 2020. Mixout: effective regularization to finetune large-scale pretrained language models. In: Proceedings of ICLR.

Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B., 2016. A diversity-promoting objective function for neural conversation models. In: Proceedings of NAACL-HLT.

Liu, C.-W., Lowe, R., Serban, I.V., Noseworthy, M., Charlin, L., Pineau, J., 2016. How not to evaluate your dialogue system: an empirical study of unsupervised evaluation metrics for dialogue response generation. Proceedings of EMNLP 2122–2132.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach, p. 11692 arXiv preprint arXiv:1907.

Mosbach, M., Andriushchenko, M., Klakow, D., 2020. On the Stability of Fine-Tuning Bert: Misconceptions, Explanations, and Strong Baselines arXiv preprint arXiv: 2006.04884.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. In: Proceedings of NAACL-HLT.

Peters, M.E., Neumann, M., IV, R.L.L., Schwartz, R., Joshi, V., Singh, S., Smith, N.A., 2019. Knowledge enhanced contextual word representations. In: Inui, K., Jiang, J., Ng, V., Wan, X. (Eds.), Proceedings of EMNLP, pp. 43–54.
Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., 2018. Improving language understanding by generative pre-training. In: Proceedings of OpenAI Technical Report.
Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., 2019. Language models are unsupervised multitask learners. In: Proceedings of OpenAI Technical Report.
Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P., 2016. SQuAD: 100, 000+ questions for machine comprehension of text. In: Su, J., Carreras, X., Duh, K. (Eds.), Proceedings of EMNLP.
Rasley, J., Rajbhandari, S., Ruwase, O., He, Y., 2020. Deepspeed: system optimizations enable training deep learning models with over 100 billion parameters. Proceedings of KDD 3505–3506.
Sanh, V., Debut, L., Chaumond, J., Wolf, T., 2019. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter arXiv preprint arXiv:1910.01108.
Shang, L., Lu, Z., Li, H., 2015. Neural responding machine for short-text conversation. In: Proceedings of ACL-IJCNLP.
Su, Y., Han, X., Zhang, Z., Li, P., Liu, Z., Lin, Y., Zhou, J., Sun, M., 2020. Contextual Knowledge Selection and Embedding towards Enhanced Pre-trained Language Models arXiv preprint arXiv:2009.13964.
Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., Wu, H., 2019. Ernie: Enhanced Representation through Knowledge Integration arXiv preprint arXiv:1904.09223.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Proceedings of NIPS.
Wang, X., Gao, T., Zhu, Z., Liu, Z., Li, J., Tang, J., 2019. Kepler: A Unified Model for Knowledge Embedding and Pre-trained Language Representation arXiv preprint arXiv:1911.06136.
Wang, Y., Ke, P., Zheng, Y., Huang, K., Jiang, Y., Zhu, X., Huang, M., 2020. A large-scale Chinese short-text conversation dataset. In: Proceedings of NLPCC.
Wei, J., Ren, X., Li, X., Huang, W., Liao, Y., Wang, Y., Lin, J., Jiang, X., Chen, X., Liu, Q., 2019. Nezha: Neural Contextualized Representation for Chinese Language Understanding arXiv preprint arXiv:1909.00204.
Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., Ma, W.-Y., 2017. Topic aware neural response generation. Proceedings of AAAI 3351–3357.
Xiong, W., Du, J., Wang, W.Y., Stoyanov, V., 2020. Pretrained encyclopedia: weakly supervised knowledge-pretrained language model. Proceedings of ICLR.
Xu, L., Zhang, X., Li, L., Hu, H., Cao, C., Liu, W., Li, J., Li, Y., Sun, K., Xu, Y., et al., 2020. Clue: A Chinese Language Understanding Evaluation Benchmark arXiv preprint arXiv:2004.05986.
You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., Hsieh, C.-J., 2020. Large batch optimization for deep learning: training bert in 76 minutes. In: Proceedings of ICLR.
Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q., 2019. ERNIE: enhanced language representation with informative entities. In: Proceedings of ACL.
Zhang, Z., Qi, F., Liu, Z., Liu, Q., Sun, M., 2020. Know what You Don't Need: Single-Shot Meta-Pruning for Attention Heads arXiv preprint arXiv:2011.03770.
Zheng, C., Huang, M., Sun, A., 2019. ChID: a large-scale Chinese IDiom dataset for cloze test. In: Proceedings of ACL.