CPM-3: A Controllable and Versatile Generative Pre-trained Language Model

Fanchao Qi[1,2*], **Xu Han**[1,3*†], **Zhi Zheng**[3,4], **Chuancheng Lv**[1,2], **Guoyang Zeng**[3,4], **Fengyu Wang**[1], **Gang Chen**[2], **Zhenhuan Huang**[3], **Weilin Zhao**[1], **Nianning Liang**[1], **Xing Wang**[5‡], **Ziqing Qiao**[1], **Wenhao Li**[1], **Xinhao Zhao**[1], **Lei Zhang**[2], **Yuqi Luo**[1], **Yining Ye**[1], **Ruoyu Qin**[1], **Jiarui Yao**[1], **Yusi Wu**[6‡], **Zhiyuan Liu**[1,7], **Maosong Sun**[1,7]

[1]Tsinghua University [2]DeepLang AI [3]OpenBMB [4]ModelBest

[5]University of Electronic Science and Technology of China

[6]Communication University of China [7]Beijing Academy of Artificial Intelligence

Abstract

Recently large pre-trained language models (PLMs) have shown powerful text generation ability. However, two big challenges in text generation are not well resolved. The first is the controllability. Existing PLMs have limited ability to control what to generate, thus cannot generate text people want consistently. The second is the versatility. Most PLMs use language modeling as the only pre-training objective, and cannot perform well in transferring to some downstream tasks (e.g., paraphrasing) without adequate labelled data. To tackle the two challenges, we propose a controllable and versatile generative PLM named CPM-3. We pre-train CPM-3 with various types of control signals in multiple generation modes, teaching the model to generate text according to given signals and modes. We construct a large Chinese corpus to pre-train CPM-3, and conduct extensive evaluations to measure its text generation performance. The results of both automatic and human evaluations show that CPM-3 achieves the overall best performance in different kinds of text generations tasks compared with other large PLMs.

1 Introduction

In recent years, pre-trained language models (PLMs) (Kenton and Toutanova, 2019; Radford et al., 2019) have been widely explored in the NLP field (Han et al., 2021). Owing to applying self-supervised learning (Liu et al., 2020b), PLMs can capture rich knowledge from a tremendous amount of unlabeled data and serve as the backbone for handling complex NLP tasks. Boosted by distributed systems and massive data, we can increase the parameter scale of PLMs (Brown et al., 2020; Chowdhery et al., 2022). The increase in parameter scale brings further improvements for PLMs, and these large-scale PLMs exhibit impressive text generation capabilities, which is difficult to achieve with conventional NLP models.

Although large-scale PLMs have achieved promising results, especially on various natural language generation (NLG) tasks, applying these PLMs to practical scenarios still faces challenges: (1) **The controllability challenge**. Given a specific context, a PLM can follow several feasible directions to generate subsequent tokens but has no idea which result is the most desirable. Therefore, taking control over PLMs is necessary to ensure what we expect can be generated. Some preliminary studies have attempted to make PLMs controllable (Keskar et al., 2019; Sun et al., 2021), yet the controllability of PLMs remains an open problem. (2) **The versatility challenge**. The objectives of language modeling enable PLMs to generate fluent free text, yet applying PLMs to some NLG tasks still requires tuning PLMs on sufficient task-specific data, especially for those tasks inconsistent with pre-training objectives, such as summarization (Lewis et al., 2019) and paraphrasing (Witteveen and Andrews, 2019). It is not easy to annotate enough data for these special NLG tasks.

For these challenges, we propose a controllable and versatile generative PLM named CPM-3. For the controllability challenge, CPM-3 can generate text according to control signals. During pre-training CPM-3, we let the model grasp some common signals, including lexical constraints, text styles, knowledge triplets, and text-related events. For the versatility challenge, we introduce multiple generation modes as pre-training objectives. As shown in Figure 1, we use text infilling (linking broken paragraphs), text transformation (paraphrasing paragraphs), text compression (making paragraphs shorter without changing core meanings), and text expansion (making paragraphs longer without changing core meanings) as pre-training objectives. The above modes can cover most NLG tasks,

*Equal contribution

†Corresponding author

‡Work done during internship at Tsinghua University

Figure 1: Example of pre-training objectives for CPM-3, including infilling, transformation, compression, expansion, and using control signals for generation. For each input-output pair, we give its corresponding English interpretation.

such as summarization, paraphrasing, story generation, and style transfer (Deng et al., 2021).

To make CPM-3 better learn diverse generation modes and control signals, instead of applying the vanilla encoder-decoder architecture of Transformer (Vaswani et al., 2017), we introduce a unified modeling architecture to simultaneously encode contexts and generate tokens by modifying attention masks to switch generation modes. Based on the unified architecture, we adopt a prompt-based multi-segment mechanism to further enhance the controllability and versatility of CPM-3. In this mechanism, we pre-train soft prompts for each mode so that we can stimulate mode-specific knowledge of CPM-3 to obtain better generation quality. Moreover, mode prompts, input contexts, control signals are placed into different segments, and each segment has its own relative position encoding mechanism to perform segment-specific text encoding. Concatenating prompts and segments can work well for different NLG scenarios.

We use a massive Chinese corpus to pre-train CPM-3 and incorporate data augmentation rules to heuristically annotate large amounts of pseudo-labeled data for different generation modes. By introducing a variety of distributed computing strategies, the pre-training of CPM-3 (7 billion parameters) takes a total of 28 days on 176 NVIDIA V100 GPU cards. We conduct extensive evaluations on six typical Chinese text generation datasets covering four classic NLG tasks. All these evaluations are conducted in three settings, including full-data fine-tuning, few-shot fine-tuning, and zero-shot inference. The experimental results show that CPM-3 can outperform existing typical Chinese PLMs on these NLG datasets, beating GLM (Du et al., 2022) and CPM-2 (Zhang et al., 2021a) (over 10 billion parameters), and even ERNIE-3 Zeus (Sun et al., 2021) (100 billion parameters) as well as Yuan (Wu et al., 2021) (245 billion parameters). In addition to the generated text quality, we also evaluate the controllability, finding CPM-3 also performs best.

To conclude, this paper has three major contributions to the generative PLMs: (1) we propose to introduce controllable generation during the pre-training phase; (2) we propose to conduct pre-training with multiple objectives that cover almost all NLG tasks; and (3) we train a controllable

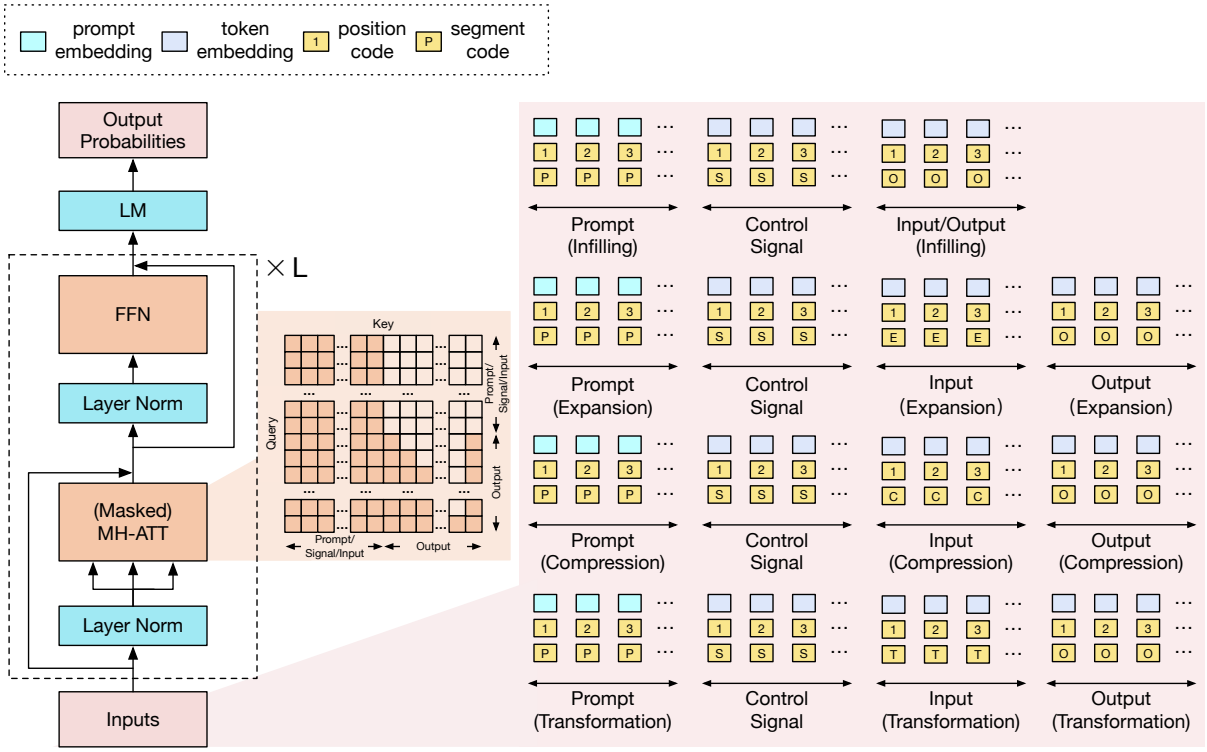

Figure 2: The overall framework and pre-training objectives of CPM-3.

and versatile generative Chinese PLM named CPM-3 that outperforms other Chinese PLMs in terms of either generation quality or controllability.

2 Model Architecture

As shown in Figure 2, CPM-3 introduces several improvements over conventional PLM architectures: (1) applying a multi-segment mechanism to manipulate complex combinations of generation modes and control signals, (2) pre-training mode-specific soft prompts for better controllability, and (3) applying a unified modeling architecture for better versatility.

2.1 Multi-segment Mechanism

In the pre-training stage, we use text infilling [1], text expansion, text compression, and text transformation as our pre-training objectives, and we argue that these four modes can cover most downstream NLG tasks. Besides making CPM-3 capable of as many NLG tasks as possible, we also enable CPM-3 to generate contents according to preset control signals. Considering the complexity of the combination of generation modes and control signals, we introduce a multi-segment mechanism to organize these generation scenarios.

Specifically, we divide the input of CPM-3 into four parts to carry soft prompts, control signals, input tokens, and output tokens, respectively. Since text infilling aims to complete the input tokens and continue to generate subsequent tokens, rather than generating new contents based on the input, we thus use the same segment to carry both the input and output of text infilling. Formally, the input of CPM-3 can be denoted as

$$\begin{aligned}&\{\{\mathbf{e}_1, p_1, s_1\}, \ldots, \{\mathbf{e}_n, p_n, s_n\},\\ &\{\mathbf{e}_{n+1}, p_{n+1}, s_{n+1}\}, \ldots, \{\mathbf{e}_{n+m}, p_{n+m}, s_{n+m}\}\},\end{aligned} \tag{1}$$

where $\mathbf{e}_i$, p_i, s_i are the embedding, position code, and segment code of the i-th token, respectively.

The first n tokens are soft prompt tokens, details of which are described in Section 2.2. The last m tokens are text tokens, including control signals, input tokens, and output tokens. As shown in Figure 2, all tokens in a segment are assigned the same segment code, and their position codes also indicate relative positional information within the segment. Segment codes and position codes are used to provide multi-segment relative position bias, details of which are shown in Section 2.3.

[1] Both autoencoding language modeling and autoregressive language modeling can be formalized as text infilling. Therefore, in this paper, we will not detailly show how to pre-train CPM-3 on these basic language modeling tasks.

2.2 Pre-trained Soft Prompts

In recent years, inspired by Lester et al. (2021), prompt tuning has been widely explored to effectively and efficiently adapt PLMs to downstream tasks. Prompt tuning has two advantages: (1) Prompt tuning formalizes all downstream tasks as cloze-style or generation objectives to reduce the gap between pre-training tasks and downstream tasks, which can improve the effectiveness of knowledge transfer; (2) Prompts can serve downstream tasks by activating specific parameters in PLMs to stimulate task-specific knowledge. Given an input instance $\mathbf{x}$ and its output $\mathbf{y}$, we train models by optimizing the function

$$\arg\max_{\theta} \sum_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x};\theta), \tag{2}$$

where θ is the model parameters. As shown in Figure 2, learnable soft prompt embeddings $\mathbf{P}$ are concatenated to the beginning of the input sequence in prompt tuning. Instead of optimizing Eq. (2), models are then trained by optimizing the function

$$\arg\max_{\mathbf{P},\theta} \sum_{\mathbf{x}} \log p\big(\mathbf{y} \mid [\mathbf{P};\mathbf{x}];\mathbf{P},\theta\big), \tag{3}$$

where $[\cdot;\cdot]$ is the concatenation function.

Lester et al. (2021) show that only tuning $\mathbf{P}$ and freezing θ can achieve comparable results to tuning all parameters of PLMs, due to the power of large-scale PLMs. However, Gu et al. (2022) show that learning effective soft prompts is not easy, which may cause prompt tuning to perform poorly in various few-shot scenarios. To obtain effective soft prompts, Gu et al. (2022) generalize downstream NLP tasks into several modes and pre-train soft prompts for each mode. In CPM-3, we follow the settings of Gu et al. (2022) and pre-train soft prompts for all generation modes. By introducing pre-trained soft prompts, on the one hand, we can better support the adaptation of CPM-3 to downstream NLG tasks. On the other hand, the soft prompts of each generation mode can also stimulate mode-specific knowledge in CPM-3 to obtain better generation quality. More details on prompt tuning can be found in the papers (Lester et al., 2021; Gu et al., 2022).

2.3 Unified Modeling Architecture

To make CPM-3 better learn diverse language modeling modes and control signals, we reform the model architecture based on CPM-2 (Zhang et al., 2021a) to accommodate various combinations between language modeling modes and control signals. Instead of applying the encoder-decoder architecture of Transformer (Vaswani et al., 2017), we use a unified modeling architecure to simultaneously encode contexts and generate tokens, by modifying attention masks to control the generation process.

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Softmax}\big(\mathbf{M} \odot (\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}})\big)\mathbf{V}, \tag{4}$$

where $\mathbf{M}$ is the attention mask and $\odot$ is the Hadamard product. Similar unified encoder architectures have demonstrated their effectiveness and simplicity in prelimary works (Dong et al., 2019; Du et al., 2022).

It is well known that the position information of tokens is important for encoding semantics, since different token orders may indicate completely different semantics. The original Transformer structure encodes absolute positions. More specifically, each absolute position is assigned a learnable embedding, and position embeddings and token embeddings are added together as the input to the Transformer structure. Recently, Shaw et al. (2018) show that relative distances between tokens cannot be captured by applying a strategy that encodes only absolute positions. To this end, various recent efforts have begun to explore strategies for encoding relative distances between tokens. Relative position bias is a typical method for relative position encoding, which is applied in T5 (Raffel et al., 2020) and CPM-2 (Zhang et al., 2021a):

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Softmax}\big(\mathbf{M} \odot (\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{B})\big)\mathbf{V}, \tag{5}$$

where $[\mathbf{B}]_{i,j} = f(p_i - p_j)$ is the mapping of the distance between the i-th token and the j-th token, and the mapping function $f(\cdot)$ is learnable.

As we mentioned before, to handle complex combinations of generation modes and control signals, we adopt a multi-segment mechanism to organize the data for CPM-3. Considering that different modes and signals have their own position encoding requirements, we implement multi-segment relative position bias by replacing $\mathbf{B}$ in Eq. (5) with

$$[\mathbf{B}]_{i,j} = \begin{cases} f_{s_i,s_j}(p_i - p_j), & s_i = s_j, \\ b_{s_i,s_j}, & s_i \neq s_j, \end{cases} \tag{6}$$

Intuitively, compared with the vanilla relative position bias in Eq. (5), the multi-segment version can

Data Type	Data Source	Data Size
Web Pages	Chinese web pages from WudaoCorpora and CommonCrawl.	72TB
E-book	Electronic books in different genres.	6TB
Encyclopedia	Chinese Wikipedia, Baidu Baike and other encyclopedias.	32GB
News	News from mainstream media.	28GB
Specific contents	Academic papers, financial reports, patents, government documents, etc.	2.4TB

Table 1: Detail of the raw data for pre-training.

fully consider the segment correlation to encode relative distances. In this paper, for simplicity, if two tokens do not belong to the same segment, no matter what their relative position distance is, we will assign a unified bias value b_{s_i,s_j}.

3 Data Augmentation

To pre-train CPM-3, we build a large and clean corpus that comprises text from various sources, and apply a series of data augmentation methods for automatically labeling pseudo data. In this section, we describe the details of building pre-training data and performing data augmentation for CPM-3.

3.1 Data Source

Our corpus is roughly derived from five parts: (1) web pages from WuDaoCorpora (Yuan et al., 2021)[2] and CommonCrawl;[3] (2) e-book, which embraces Chinese electronic books in different genres; (3) encyclopedia, which is composed of text from online encyclopedias, such as Wikipedia, Baibu Baike; (4) news, including text from some mainstream newspapers and news websites; (5) specific contents, which mainly comprise academic papers, financial reports, patents, government documents. The total size of the raw data exceeds 80TB. Table 1 shows the details of the raw data.

3.2 Data Processing

To obtain a clean and high-quality corpus for pre-training CPM-3, we design a sophisticated data processing workflow. In the workflow, we adjust the data filtering criteria to control the amount of retained data, especially for those web pages that contain a lot of noise. Following Hoffmann et al. (2022), we first determine the appropriate amount of pre-training data for CPM-3, which is about 130B tokens (about 1.1 TB text), and then determine the specific filtering criteria, rather than the other way around. The whole data processing workflow can be divided into three steps, namely standardization, cleaning, and filtering.

Standardization This step aims to transform the raw data into a standard format, especially for those complex web pages. Then, we use the JSON file format to store data. A datum usually corresponds to an article, a web page, or a chapter of a book. The main keys in a JSON item of a datum include title, content, and some metadata (abstract, keywords, authors, etc.)

Cleaning This step aims to normalize the data and remove low-quality parts of the data. Specifically, this step includes character replacement and normalization (mainly for some character variants), character deletion (mainly for some whitespace characters), punctuation normalization, and traditional-simplified Chinese character conversion. Finally, we use some hashing algorithms to deduplicate the previously processed data.

Filtering This step aims to filter the data that is not suitable for pre-training a model. We use three filtering strategies, including keyword-based filtering, rule-based filtering, and model-based filtering. For the keyword-based filtering strategy, we build a filtering vocabulary based on existing sensitive/advertising vocabularies and then use the filtering vocabulary to delete the harmful or sensitive contents in the data. For the rule-based filtering, we use some criteria including length of contents, proportion of Chinese characters, frequency of certain characters, and density of named entities. For the model-based filtering, we train a small language model with some quality-guaranteed text (ebooks, news, papers, etc.). By computing the perplexity of the data using the small model, we can filter high-perplexity parts of the data.

3.3 Pseudo Data Generation

After finishing the data processing, to incorporate CPM-3 with various generation abilities, we gen-

[2] https://resource.wudaoai.cn/home
[3] https://commoncrawl.org/

erate pseudo data that more closely resembles different generation tasks. Our pseudo data can be classified into four categories and we describe each category as follows:

Text Infilling and Language Modeling We generate pseudo data to teach CPM-3 the capabilities of language modeling and text infilling. In each training instance, one or multiple text spans are masked out and the model is trained to causally predict the masked spans. For the single-span masking, the text span is 50% at the end of the instance and 50% at an arbitrary position, and the masking rate is drawn from $[0.5, 1.0]$. For the multi-span masking, we draw a masking rate from a uniform distribution $\mathcal{U}(0, 1)$ and then randomly mask tokens according to the masking rate.

Text Transformation We employ some back-translation techniques to generate the paraphrasing data. Specifically, we divide each document into sentences, translate these sentences into English and then back into Chinese. The back-translated sentences are eventually assembled into a document which is used as the source and the original document is treated as the target.

Text Compression For the text compression task, we generate two different types of pseudo data, corresponding to sentence compression and paragraph summarization, respectively. For sentence compression, our approach is based on syntactic parsing. In particular, we use DDParser (Zhang et al., 2020b) to obtain the syntactic tree for a sentence and design several heuristic rules to remove unnecessary components such as adjectives. For paragraph summarization, we refer to Su (2021) and Zhang et al. (2020a) and adopt Gap Sentence Generation (GSG). Specifically, given a document with n sentences, we pick $n/4$ sentences such that the common subsequence of these picked sentences and the remaining sentences is the longest. We take the remaining $3n/4$ sentences as the source and regard the picked $n/4$ sentences as the target.

Text Expansion To some extent, text expansion is the inverse task of text compression. Therefore, for text expansion, we also use GSG and exchange the source and target of the generated data. However, GSG is not completely suitable for the text expansion task, as it may lead the model to generate sentences irrelevant to the input. We attribute it to the fact that most remaining sentences are not closely related to the summary for some types of corpora, such as novels. Therefore, we take the whole document (n sentences) as the target and the summary ($n/4$ sentences) as the source.

3.4 Control Signals

We provide control signals in the pre-training stage to enable CPM-3 to generate text according to the given information. We extract three kinds of control signals from target texts, namely keywords, relations and events. Extraction methods are described as follows:

Keyword Extraction We utilize THULAC (Sun et al., 2016) for tokenization and KeyBERT (Grootendorst, 2020) to obtain keywords. We filter keywords with low quality, and remove duplicated keywords. Besides, keywords are kept in the order of occurrence in the target span, the number of which depends on the length of the target span.

Knowledge Graph Triplet Extraction We first use NLTK to extract named entities and then extract relations with a multilingual relation extraction model trained on the FewRel dataset covering 80 common relations (Han et al., 2018; Gao et al., 2019). Similar to keywords extraction, we filter those relations with low quality.

Event Extraction. Semantic role labeling (SRL) is used for event extraction, which labels some phrases in sentences as the argument (semantic roles) of a given predicate, such as agent, recipient, time and place, etc. We implement it with LTP (Che et al., 2020) and sample no more than four events for each training instance.

4 Training Setup

We employ the BMTrain[4] toolkit to train CPM-3 for 72.5K steps on 176 32GB NVIDIA V100 GPUs, which takes a total of 21 days. We set the total batch size to 1584 (9 per GPU) and the maximum sequence length to 2048. We initialize the weight of our model using a normal distribution with zero mean, and the standard deviation is set to 0.02 and 0.1 for embedding layer and other layers, respectively, which is shown to speed up the convergence in our pilot experiments. We use AdamW optimizer (Loshchilov and Hutter, 2019) with weight decay of 0.01 and set the β_1 and β_2 to 0.9 and 0.999, respectively. We set the initial learning rate to 0.1 and warmup steps to 2K, and use

[4] https://github.com/OpenBMB/BMTrain

Model	CPM-2	GLM	Wenzhong	Mengzi	ERNIE 3.0	Yuan 1.0	Benetnasch
Type	Open	Open	Open	Open	Close	Close	Close
Architecture	En-De	De	De	En-De	En-De	De	De
Size	11B	10B	3.5B	220M	100B	245B	13B
Evaluation Setting	All	All	All	All	0-shot	0-shot	0-shot

Table 2: The details of compared Chinese generative PLMs. "Open" and "Close" represent open-source and close-source, respectively. "En" and "De" denote Encoder and Decoder, respectively.

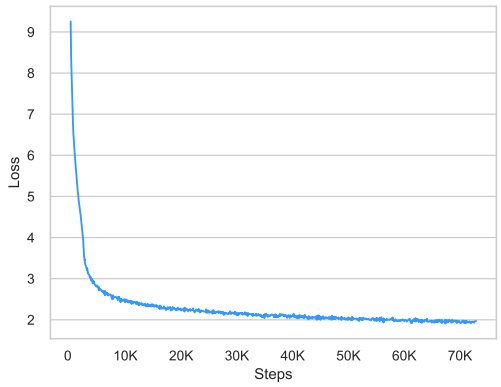

Figure 3: Training loss of CPM-3.

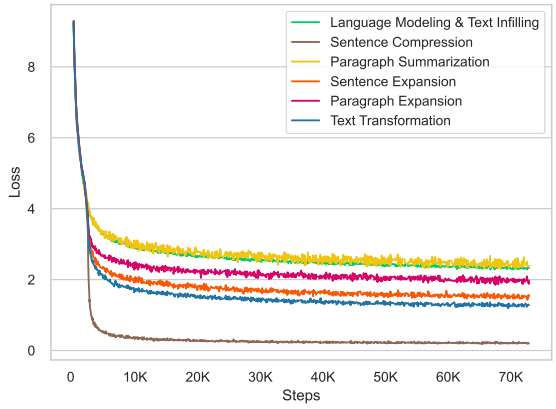

Figure 4: Losses of different pre-training tasks of CPM-3.

the Noam learning rate scheduler (Vaswani et al., 2017). The training loss curve is shown in Figure 3. We also show losses of different pre-training tasks in Figure 4.

5 Automatic Evaluation

5.1 Compared Models

We try to compare CPM-3 with as many Chinese generative PLMs as possible. We first select some large open-source Chinese PLMs, including CPM-2 (Zhang et al., 2021a), GLM (GLM-XXLarge-Chinese) (Du et al., 2022), Wenzhong (Wenzhong-GPT2-3.5B) (IDEA-CCNL, 2021) and Mengzi (Mengzi-T5-base) (Zhang et al., 2021b). We evaluate them in all three settings (full-data fine-tuning, few-shot fine-tuning and zero-shot inference).

In addition to the above open-source models, we also make a comparison between CPM-3 and three close-source large-scale Chinese language models, including ERNIE 3.0 Zeus[5], Yuan 1.0 (Wu et al., 2021) and Benetnasch[6]. We conduct experiments with the above three models by using their APIs in the zero-shot setting.

The details of these compared models are shown in Table 2.

5.2 Datasets and Metrics

Datasets

We use seven datasets to evaluate the language capabilities of CPM-3.

Language Modeling To conduct a fair comparison of the language modeling capabilities between CPM-3 and baselines, we construct a new dataset named RMRB, which contains news from People's Daily since May 1st, 2022.[7] The length of each news article is between 50 and 256 characters, and the first 20 characters are used as input.

Text Expansion We use the outline-conditioned generation task (Rashkin et al., 2020) to evaluate the text expansion ability of PLMs, which is to expand the outline of into a semantically coherent and fluent paragraph. We conduct experiments on the OutGen dataset from LOT (Guan et al., 2021), a benchmark for evaluating Chinese long text generation. The OutGen dataset is a outline-conditioned story generation dataset, which uses the title of the story and an out-of-order set of phrases extracted

[5] https://wenxin.baidu.com/wenxin/modelbasedetail/ernie3_zeus

[6] https://openapi.singularity-ai.com/index.html#/documentIndex

[7] All PLMs do not use these news text data for pre-training since they are up-to-date.

Dataset	Train	Dev	Test	Article Len.	Summary Len.
LCSTS	2.4M	8.7K	0.7K	103.7	17.9
CEPSUM	434.0K	5.0K	4.9K	325.5	79.2
CLTS	148.3K	20.3K	16.7K	1363.7	58.1

Table 3: Details of three text summarization datasets.

automatically from the story as the outline to guide the model to generate the full story. Here we treat phrases as keywords as control signals of CPM-3 to generate the story.

Summarization For a comprehensive analysis of the summarization ability of CPM-3, we conduct experiments on three datasets of different lengths, i.e., LCSTS (Hu et al., 2015), CEPSUM (Yuan et al., 2020) and CLTS (Liu et al., 2020c), the details of which are shown in Table 3. The average lengths of articles and summaries are listed in the "Article Len." and "Summary Len." columns, respectively.

Paraphrase To our best knowledge, there is no widely used dataset for generative paraphrase. Existing datasets, such as LCQMC (Liu et al., 2018) and BQ Corpus (Chen et al., 2018), are more suitable for the semantic similarity task. After comparing the diversity and semantic similarity of existing datasets, we choose to use the OPPO Xiaobu text semantic matching dataset[8] to evaluate the paraphrase ability of PLMs. We only keep the sentence pairs with label 1, which means that they have the same meaning, and then filter out those that are too similar morphologically.

Metrics

As shown in Table 4, we use different evaluation metrics for different datasets, following previous work. Next, we briefly describe these metrics.

BLEU BLEU-n (Papineni et al., 2002) evaluates the quality of the output by measuring the overlap between the n-grams of generated text and reference text. A higher BLEU score indicates the generated text is more consistent with the reference text and the quality of the generated result is better.

ROUGE-L ROUGE-L F1 score (Lin, 2004) is used to measure the quality of the generated summary in summarization task. As a typical summarization metric, a higher ROUGE-L F1 score means higher similarity between output and reference and better generation performance.

Coverage and Order Following Guan et al. (2021), we utilize Coverage and Order to evaluate the outline inclusion condition of the generated text in text expansion task. The Coverage score is the average Rouge-L Recall score between the output and each outline phrases. The higher Coverage score is, the more outline phrases the generated text covers and the stronger the controllability of the model is. The Order score measures the gap between the positional orders of outline phrases in the generated texts. A higher Order score means the model is better able to sort out-of-order outline phrases and expand them into smooth text.

Self-BLEU Based on BLEU score, the Self-BLEU score(Zhu et al., 2018) is obtained by calculating the average BLEU-Score between texts and is used to measure the diversity of the generation results of the model. A low Self-BLEU score indicates less similarity between the generated results and therefore more diversity.

BERTScore Paraphrase task requires rewriting the texts while keeping the semantics basically unchanged, so the BERTScore(Zhang et al., 2019a) is utilized to measure the semantic similarity between the output and the original text that needs to be rewritten. A higher BERTScore means that texts are more semantically similar to each other. We calculate the average BERTScore between the original text and target text of the dataset as threshold and only when the result of the model is above this threshold can it be considered as a valid result.

5.3 Experimental Results

For CPM-3 and baselines, we employ three different decoding strategies, namely, beam search, nucleus sampling (Holtzman et al., 2020), and contrastive search (Su et al., 2022) for generation and choose the best result from them.

Comparison with Open-source PLMs

As shown in Table 2, we compare CPM-3 with strong open-source baseline PLMs in three settings, including full-data fine-tuning, few-shot fine-tuning and zero-shot inference, on datasets mentioned in Section 5.2.

Table 4 shows the evaluation results. We can see that CPM-3 achieves overall best performance in all three settings and all datasets. Although in

[8] https://www.luge.ai/#/luge/dataDetail?id=28

Setting	Model	RMRB		OutGen				OPPO		LCSTS	CEPSUM	CLTS
		BLEU-1	BLEU-2	BLEU-1	BLEU-2	Coverage	Order	Self-BLEU-4	BERTScore	ROUGE-L	ROUGE-L	ROUGE-L
Full	Mengzi	8.72	4.09	27.39	19.76	62.03	57.86	46.81	89.37	32.51	25.61	44.57
	Wenzhong	18.18	8.73	31.65	20.85	69.57	57.46	24.35	82.21	36.87	25.16	47.65
	GLM	14.70	7.77	37.43	27.12	85.28	**68.54**	24.98	83.14	47.40	24.54	50.58
	CPM-2	16.44	8.54	41.73	27.32	73.75	61.53	28.55	83.47	44.04	26.92	50.76
	CPM-3	**18.63**	**9.89**	**41.96**	**29.43**	**85.77**	65.72	**23.95**	82.21	**47.99**	**27.34**	**51.30**
0-shot	Mengzi	0.00	0.00	0.00	0.00	0.00	0.00	0.28	15.92	11.06	16.37	25.08
	Wenzhong	10.50	5.19	9.75	5.37	36.49	42.40	6.33	58.26	0.25	2.76	2.84
	GLM	4.94	0.23	1.01	0.39	10.73	29.24	3.02	58.69	22.72	18.20	23.40
	CPM-2	0.31	0.11	0.00	0.00	1.19	22.75	0.00	48.18	0.00	0.00	0.00
	CPM-3	**12.51**	**5.69**	**28.89**	**19.59**	**86.23**	**60.84**	**26.03**	81.00	**23.18**	**18.97**	**26.62**
1-shot	Mengzi	0.08	0.04	2.15	1.38	25.87	38.59	0.61	29.43	18.82	15.12	18.71
	Wenzhong	11.36	4.98	22.47	11.92	47.66	46.54	6.42	60.13	0.32	8.21	12.21
	GLM	6.48	3.18	3.72	1.75	13.15	32.14	8.70	59.15	24.51	16.76	41.07
	CPM-2	0.30	0.13	13.80	6.95	44.78	46.78	0.00	47.41	15.61	15.71	16.82
	CPM-3	**14.29**	**7.11**	**31.33**	**21.01**	**81.96**	**59.93**	**30.36**	81.37	**28.91**	**17.81**	**43.61**
4-shot	Mengzi	1.55	0.53	6.41	4.15	35.31	45.42	4.76	60.27	24.22	16.95	29.89
	Wenzhong	12.46	5.30	26.13	15.08	55.31	49.65	13.32	65.04	21.60	9.45	20.19
	GLM	7.83	3.89	10.89	5.43	32.74	39.56	5.50	58.96	26.34	16.50	41.84
	CPM-2	0.24	0.14	19.12	10.99	66.78	57.94	0.03	49.81	24.75	19.19	23.38
	CPM-3	**13.62**	**7.02**	**34.65**	**23.65**	**78.19**	**59.21**	**24.47**	80.77	**30.93**	19.19	**43.06**
16-shot	Mengzi	1.35	0.40	8.70	5.74	39.61	48.51	31.49	80.22	26.71	16.94	29.41
	Wenzhong	11.25	4.81	31.11	19.13	65.51	53.92	32.61	76.75	24.07	16.11	23.89
	GLM	7.85	3.89	17.94	10.08	52.28	48.50	7.77	60.99	31.02	18.21	**44.68**
	CPM-2	0.69	0.43	20.87	13.01	74.40	59.45	2.31	55.67	30.86	21.73	33.32
	CPM-3	**12.15**	**6.26**	**34.84**	**24.10**	**77.74**	**59.88**	**28.69**	83.11	**31.44**	**21.92**	42.03

Table 4: Automatic evaluation results of CPM-3 and open-source baseline PLMs on different datasets.

Model	RMRB		OutGen				OPPO		CEPSUM
	BLEU-1	BLEU-2	BLEU-1	BLEU-2	Coverage	Order	Self-BLEU-4	BERTScore	ROUGE-L
ERNIE 3.0 Zeus	5.32	2.59	27.36	14.97	48.41	46.77	1.25	56.67	17.90
Yuan 1.0	10.04	4.76	17.45	9.43	49.68	46.55	3.53	61.29	18.98
Benetnasch	9.80	4.38	17.27	9.07	47.41	45.62	2.87	61.10	18.47
CPM-3	**12.43**	**5.64**	**28.97**	**19.63**	**86.29**	**60.89**	**26.28**	**80.98**	18.98

Table 5: Automatic evaluation results of CPM-3 and other close-source PLMs in the 0-shot setting.

rare cases, such as Order in full-data fine-tuning setting on OutGen and ROUGE-L in 16-shot fine-tuning setting on CLTS, the performance of CPM-3 is exceeded by GLM, it should be noted that the gap between the two results is not large, and GLM has far more parameters than CPM-3.

In addition, greatly benefited from our proposed pre-training with multiple objectives covering various generation tasks, CPM-3 has a stronger advantage in the few-shot fine-tuning and especially zero-shot inference scenarios. And the experimental results show that in these two settings CPM-3 achieves more significantly better results compared with baselines than that in full-data fine-tuning setting, which demonstrates the power of our proposed pre-training strategy.

As mentioned in Section 5.2, we set the threshold for BERTScore to filter out the results that do not meet the semantic similarity requirements for the OPPO dataset. In Table 4, the results with red background color are those that do not meet the standard. It can be seen that almost none of the baseline models could meet the standard in few-shot and zero-shot scenarios, while CPM-3 achieves excellent results.

Comparison with Close-source PLMs

To avoid the impact of the sensitive word filtering mechanism of APIs, we remove data from each dataset that might be identified as sensitive by the open APIs and compare CPM-3 with ERNIE 3.0 Zeus, Yuan 1.0 and Benetnasch in the zero-shot setting. Due to the limit of API calls of some PLMs, we only use one summarization dataset for evluation, namely the medium-length dataset CEPSUM.

For OutGen, OPPO and CEPSUM, we need to formulate the data into a format as prompts for baselines to generate better results. And the templates we use to formulate the data as prompts are in Table 11. As shown in Table 5, CPM-3 achieves better performance in all datasets than the three strong baseline models, which demonstrates that the pre-training strategy of CPM-3 is quite effective, especially in the 0-shot setting.

Setting	Model	RMRB			OutGen			OPPO				CEPSUM			AVERAGE		
		CO	FL	LO	CO	FL	LO	PL	CO	FL	LO	KI	FL	LO	CO	FL	LO
Full	Mengzi	1.43	0.79	1.18	1.08	0.66	1.14	0.37	0.67	0.72	0.72	1.00	0.94	1.13	1.06	0.78	1.04
	Wenzhong	1.85	1.97	1.90	1.44	1.84	1.64	0.40	0.66	0.77	0.73	1.33	1.79	1.74	1.32	1.59	1.50
	GLM	1.85	**1.98**	1.93	**1.46**	1.80	1.59	0.48	0.90	0.93	0.92	1.44	**1.90**	**1.88**	1.40	1.65	1.58
	CPM-2	1.84	1.82	1.82	1.24	1.78	1.62	0.31	0.53	0.59	0.60	1.23	1.46	1.55	1.20	1.41	1.40
	CPM-3	1.85	1.96	1.93	1.40	**1.87**	**1.73**	**0.66**	**1.09**	**1.25**	**1.30**	**1.65**	1.84	1.85	**1.45**	**1.73**	**1.70**
0-shot	ERNIE 3.0 Zeus	**1.79**	**1.91**	1.86	1.34	**1.72**	**1.64**	0.08	0.06	0.08	0.07	1.11	**1.50**	1.45	1.06	1.30	1.25
	Yuan 1.0	1.69	1.69	1.73	1.36	1.23	1.35	0.10	0.14	0.14	0.13	**1.45**	1.31	1.03	1.06	1.09	1.06
	Benetnasch	1.74	1.77	1.80	1.35	1.26	1.46	0.06	0.06	0.06	0.06	1.27	1.23	1.14	1.05	1.09	1.13
	CPM-3	1.72	1.90	**1.87**	**1.40**	1.64	1.52	**0.23**	**0.38**	**0.42**	**0.44**	1.44	1.45	**1.50**	**1.17**	**1.36**	**1.33**

Table 6: Human evaluation results of CPM-3 and baseline PLMs. "CO" means "Coherence", "FL" means "Fluency", "LO" means "Logicality", "PL" means the proportion of "Paraphrase-Like", and "KI" means the result contains the "Key Information" of the original text. Columns in AVERAGE are average results of corresponding evaluation dimensions of the four datasets.

6 Human Evaluation

In this section, we conduct two human evaluations to measure the text generation quality and controllability of different PLMs.

6.1 Text Generation Quality

We first manually evaluate the quality of generated text by PLMs.

Datasets and Baselines For the four capabilities of language modeling, text expansion, summarization and paraphrase, we randomly sampled 100 instances from RMRB, OutGen, CEPSUM and OPPO respectively for evaluation. As for models, in the full parameter fine-tuning setting, we compare the CPM-3 with CPM-2, GLM, Wenzhong, and Mengzi; and in the zero-shot setting, we compare CPM-3 with ERNIE 3.0 Zeus, Yuan 1.0, and Benetnasch.

Evaluation Dimensions We evaluate **Coherence** between the generated text and the conditioned text, and **Fluency** and **Logicality** of the generated text. Besides, for summarization, we want the generated text to contain the key information instead of containing all information of the original text. So we evaluate the model's ability to retain **Key Information** instead of evaluating the Coherence. And for paraphrase, in order to punish cases that the generated text are essentially identical to the original text, we need to evaluate whether the generated text has changed the sentence form to a large extent while keeping the semantics basically unchanged, that is, whether the generated text is **Paraphrase-like**. We ask the annotators to evaluate whether the sample meets the requirements of paraphrase and then set the other three metrics of the unqualified samples to 0. Each instance of text generated by each model is rated by three annotators and annotators are asked to evaluate the above metrics on [0,1,2], with 0 being "very bad", 1 being "medium" and 2 being "very good".

Evaluation Results As shown in Table 6. For RMRB, OutGen and CEPSUM, CPM-3 achieves comparable performance with larger scale models. In terms of OPPO, CPM-3 has reached the standard of paraphrase in more samples and gets better results. And in average, CPM-3 achieves better performance in zero-shot/full parameters fine-tuning setting.

6.2 Controllability

As described in Section 3.4, we integrate three types of control signals in the pre-training stage, i.e., keywords, relations (knowledge graph triplets) and events. We evaluate the controllability of CPM-3 in the zero-shot setting by human.

Datasets We use the DuIE (Li et al., 2019) and DuEE (Li et al., 2020) datasets for controllability evaluation. DuIE is a relation extraction dataset that provides several relations for each article and we use the head and tail entities as keywords control. However, since the relation and event types provided by DuIE and DuEE do not match ours, we extract relations and events from these two datasets as control signals, using the methods described in Section 3.4.

Evaluation Settings For CPM-3, we employ the *language modeling* mode and formulate the control signals as control codes. For other models, we provide the model control signals using prompts, which are shown in Table 10. For each instance, we use the first five characters in the original text as

Model	% of Inclusion	Fluency	Logicality
Yuan 1.0	0.50	1.15	0.65
ERNIE	0.48	1.75	1.60
Benetnasch	0.61	1.23	1.10
CPM-3	0.86	1.53	0.85

Table 7: Controllability of keywords.

Model	% of Inclusion	Fluency	Logicality
Yuan 1.0	0.66	1.03	1.23
ERNIE	0.54	1.68	1.73
Benetnasch	0.53	1.15	1.28
CPM-3	0.88	1.93	1.68

Table 8: Controllability of events.

Model	% of Inclusion	Fluency	Logicality
Yuan 1.0	0.60	0.73	0.75
ERNIE	0.23	1.53	1.30
Benetnasch	0.38	1.03	0.73
CPM-3	0.40	1.45	1.10

Table 9: Controllability of relations.

the context. In the experiments, we sample 100 instances for each type of control signal. We conduct human evaluation from three perspectives, namely, inclusion ratio of control signals, fluency and logicality.

Evaluation Results Table 7-9 show the evaluation results of different types of controls. It can be seen from Table 7 that the keyword inclusion ratio of CPM-3 surpasses all baselines by a large margin. Though the fluency and logicality scores of CPM-3 are lower than some baselines, we emphasize that it can be easy to generate high quality text without considering control signals. From Table 8, we can see that CPM-3 outperforms all baselines in terms of event inclusion ratio and fluency score, and performs comparably to ERNIE in terms of logicality. In terms of relations, we can see from Table 9 that CPM-3 outperforms Benetnasch consistently, and achieves a good balance between controllability and generation quality, compared with Yuan 1.0 and ERNIE. In summary, the controllable generation ability of CPM-3 is quite good in the zero-shot setting, demonstrating the effectiveness of our pre-training strategy.

7 Related Work

Owing to the extensive exploration of self-supervised learning (Liu et al., 2020b) and Transformer (Vaswani et al., 2017), we can pre-train various large-scale PLMs to automatically capture rich knowledge from large amounts of unlabeled data. By tuning the parameters of PLMs on task-specific data, these pre-trained models can work well as backbones for handling complex NLP tasks. It has gradually become a consensus in the NLP field that tuning PLMs outperforms learning task-specific models from scratch (Qiu et al., 2020; Han et al., 2021).

Existing efforts mainly follow four dircetions to build PLMs, taking different model architectures and pre-training tasks: (1) taking the encoder architecture to perform autoencoding language modeling, such as BERT (Kenton and Toutanova, 2019) and RoBERTa (Liu et al., 2020d), which can well handle language understanding tasks; (2) taking the decoder architecture to perform autoregressive language modeling, such as GPT1 (Radford and Narasimhan, 2018) and GPT2 (Radford et al., 2019), which can well handle language generation tasks; (3) taking the encoder-decoder architecture to perform sequence-to-sequence language modeling, such as T5 (Raffel et al., 2020) and BART (Lewis et al., 2020), which can formalize NLP tasks into a unified sequence-to-sequence form for processing; (4) taking the unified decoder instead of the complicated encoder-decoder architecture to perform hybrid language modeling, such as UniLM (Dong et al., 2019) and GLM (Du et al., 2022).

Besides the above efforts, various improvements over Transformer-based PLMs are also advancing, including compressing parameters (Lan et al., 2019; Sanh et al., 2019), encoding relative positions (Yang et al., 2019; Su et al., 2021), encoding structure knowledge (Zhang et al., 2019b; Peters et al., 2019; Liu et al., 2020a), encoding multi-lingual knowledge (Xue et al., 2020; Lample and Conneau, 2019), encoding multi-modal knowledge (Radford et al., 2021; Ramesh et al., 2021, 2022), cncoding domain knowledge (Lee et al., 2020; Beltagy et al., 2019).

8 Conclusion and Future Work

In this paper, we introduce CPM-3, a controllable and versatile generative pre-trained language model. We make the first attempt to conduct controllable and multi-mode generative pre-training, for which we revise the architecture of the Transformer decoder and generate different kinds of

pseudo data. Extensive experimental results show that CPM-3 can achieve overall better text generation performance than existing PLMs, even much larger ones. In the future, we will try to extend CPM-3 to more languages, and propose corresponding multilingual pre-training task. In addition, we will increase the size of CPM-3 to observe whether it can achieve even better text generation performance. Finally, we will explore more control signals (e.g., sentiment) and try to improve the controllability of the model.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Proceedings of NeurIPS*.

Wanxiang Che, Yunlong Feng, Libo Qin, and Ting Liu. 2020. N-ltp: A open-source neural chinese language technology platform with pretrained models. *arXiv preprint arXiv:2009.11616*.

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 2018. The BQ corpus: A large-scale domain-specific chinese corpus for sentence semantic equivalence identification. In *Proceedings of EMNLP*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Mingkai Deng, Bowen Tan, Zhengzhong Liu, Eric Xing, and Zhiting Hu. 2021. Compression, transduction, and creation: A unified framework for evaluating natural language generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7580–7605.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of ACL*.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6251–6256.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. Ppt: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423.

Jian Guan, Zhuoer Feng, Yamei Chen, Ruilin He, Xiaoxi Mao, Changjie Fan, and Minlie Huang. 2021. LOT: A benchmark for evaluating chinese long text understanding and generation. *arXiv preprint arXiv:2108.12960*.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. 2021. Pre-trained models: Past, present and future. *AI Open*, 2:225–250.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *Proceedings of ICLR*.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A large scale chinese short text summarization dataset. In *Proceedings of EMNLP*.

IDEA-CCNL. 2021. Fengshenbang-lm. https://github.com/IDEA-CCNL/Fengshenbang-LM.

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

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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

Shuangjie Li, Wei He, Yabing Shi, Wenbin Jiang, Haijin Liang, Ye Jiang, Yang Zhang, Yajuan Lyu, and Yong Zhu. 2019. Duie: A large-scale chinese dataset for information extraction. In *Proceedings of NLPCC*.

Xinyu Li, Fayuan Li, Lu Pan, Yuguang Chen, Weihua Peng, Quan Wang, Yajuan Lyu, and Yong Zhu. 2020. Duee: A large-scale dataset for chinese event extraction in real-world scenarios. In *Proceedings of NLPCC*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020a. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

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

Xiaojun Liu, Chuang Zhang, Xiaojun Chen, Yanan Cao, and Jinpeng Li. 2020c. CLTS: A new chinese long text summarization dataset. In *Proceedings of NLPCC*.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. LCQMC: A large-scale chinese question matching corpus. In *Proceedings of COLING*.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of ICLR*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

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

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. Plotmachines: Outline-conditioned generation with dynamic plot state tracking. *arXiv preprint arXiv:2004.14967*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468.

Jianlin Su. 2021. T5 pegasus - zhuiyiai. Technical report.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *arXiv preprint arXiv:2202.06417*.

Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, and Zhiyuan Liu. 2016. THULAC: An efficient lexical analyzer for chinese.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

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

Sam Witteveen and Martin Andrews. 2019. Paraphrasing with large language models. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 215–220.

Shaohua Wu, Xudong Zhao, Tong Yu, Rongguo Zhang, Chong Shen, Hongli Liu, Feng Li, Hong Zhu, Jiangang Luo, Liang Xu, and Xuanwei Zhang. 2021. Yuan 1.0: Large-scale pre-trained language model in zero-shot and few-shot learning. *arXiv preprint arXiv:2110.04725*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Peng Yuan, Haoran Li, Song Xu, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. On the faithfulness for e-commerce product summarization. In *Proceedings of COLING*.

Sha Yuan, Hanyu Zhao, Zhengxiao Du, Ming Ding, Xiao Liu, Yukuo Cen, Xu Zou, Zhilin Yang, and Jie Tang. 2021. Wudaocorpora: A super large-scale chinese corpora for pre-training language models. *AI Open*, 2:65–68.

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

Shuai Zhang, Lijie Wang, Ke Sun, and Xinyan Xiao. 2020b. A practical chinese dependency parser based on a large-scale dataset. *arXiv preprint arXiv:2009.00901*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019a. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Zhengyan Zhang, Yuxian Gu, Xu Han, Shengqi Chen, Chaojun Xiao, Zhenbo Sun, Yuan Yao, Fanchao Qi, Jian Guan, Pei Ke, et al. 2021a. Cpm-2: Large-scale cost-effective pre-trained language models. *AI Open*, 2:216–224.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019b. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451.

Zhuosheng Zhang, Hanqing Zhang, Keming Chen, Yuhang Guo, Jingyun Hua, Yulong Wang, and Ming Zhou. 2021b. Mengzi: Towards lightweight yet ingenious pre-trained models for chinese. *arXiv preprint arXiv:2110.06696*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

Control Signal	Input	Template	Prompt
Keywords	[keywords] = 法拉墨，迪耐瑟二世 [context] = 在魔戒圣战	关键词：[keywords] 全文：[context]	关键词：法拉墨，迪耐瑟二世 全文：在魔戒圣战
Relation	[relations] = 雪山飞狐，登场角色，胡一刀 楚留香，原著作者，古龙 [context] = 金庸作品《	关系：[relations] 全文：[context]	关系：雪山飞狐，登场角色，胡一刀 楚留香，原著作者，古龙 全文：金庸作品《
Events	[events] = 谓词：开战，主体：勇士队和猛龙队的总决赛第5场竞赛，状语：如期 谓词：伤愈，地点：在今天的竞赛当中，主体：杜兰特	事件：[events] 全文：	事件：谓词：开战，主体：勇士队和猛龙队的总决赛第5场竞赛，状语：如期 谓词：伤愈，地点：在今天的竞赛当中，主体：杜兰特 全文：

Table 10: Prompt templates for controllability.

Dataset	Input	Template	Prompt
Outgen	[title]：懒纺妇 [keywords]：地上缠成一大团；女人提起纺纱；纱线放进锅；妻子懒得；不想干活；不绕成团；逃避纺纱；赶紧起身	标题：[title] 关键词：[keywords] 全文：	标题：懒纺妇 关键词：地上缠成一大团；女人提起纺纱；纱线放进锅；妻子懒得；不想干活；不绕成团；逃避纺纱；赶紧起身 全文：
OPPO	[source]：换句话说就是	[source]换句话说就是：	你上班地方在哪我想问问换句话说就是：
CEPSUM	[source]：新秀丽时尚拉杆箱登机箱行李箱旅行箱密码箱男女静音万向轮11浅蓝色银色28英寸送原装箱套，年轻拒绝平庸，轻松装下出行所需，智慧空间分隔，TSA密码锁，镜面外观，静音万向轮，浅蓝/银，弹性设计，顶侧设置，舒适便利，醒目的专属RED设计，凸显气质，分类明确，合理划分内部空间，双层结构设计，侧面，正面，背面	产品描述：[source] 产品摘要：	产品描述：新秀丽时尚拉杆箱登机箱行李箱旅行箱密码箱男女静音万向轮11浅蓝色银色28英寸送原装箱套，年轻拒绝平庸，轻松装下出行所需，智慧空间分隔，TSA密码锁，镜面外观，静音万向轮，浅蓝/银，弹性设计，顶侧设置，舒适便利，醒目的专属RED设计，凸显气质，分类明确，合理划分内部空间，双层结构设计，侧面，正面，背面 产品摘要：

Table 11: Prompt templates for baseline models.

Mode	Input	Output	Target
Language Modeling	推广美育也是家庭和社会的义务，家庭的熏陶 (The promotion of aesthetic education is also the obligation of the family and society, and the influence of the family)	，社会的文明。美育是培养人的素质，提高人的审美观的教育，在家庭中开展美育，对孩子的健康成长有着不可估量的作用。 (, the civilization of society. Aesthetic education is to cultivate people's quality, improve people's aesthetic education, and carry out aesthetic education in the family, which has an immeasurable effect on the healthy growth of children.)	与化育，公共空间的艺术呈现，都能让美进入日常生活，使美的种子生根发芽。 (and chemical education, the artistic presentation of public spaces, can make beauty into daily life, so that the seeds of beauty take root and sprout.)
Text Expansion	地方搭建巢穴, 这幢建筑, 纷纷离开, 旧窠, 寻找新, 麻雀归来, 面目一新, 堵起来 (The place where the nest was built, the building, left one after another, the old, looking for the new, the sparrow returned, the face was new, blocked)	麻雀是我的朋友。它住在我家附近的地方搭建巢穴，这幢建筑是我和小黄结婚时盖的。我的朋友在这里安了家，我们都搬走了，它还没有回来。有一天，它回来了，我很高兴，因为它是我的朋友呀！麻雀纷纷离开旧窠，寻找新家，可是它没有找到家，也找不到回家的路。一天，一只麻雀飞到我家的窗子上，看见我站在窗前望着它，它以为我死了，伤心地哭了。第二天，这只麻雀又飞来找我，说："朋友，别伤心了，我在这里等你。"我把门打开，让它进屋休息。过了些日子，我再去看它时，它已经长大了，不再叫我的名字了。它的羽毛是灰色的，眼睛黑黑的像鹰爪。我想：麻雀归来，面目一新，一定很快乐。于是我在门前贴了"喜鹊登梅"的条幅，还买了一盆花插在门上。这几天，麻雀把家堵起来了，再也不肯出来了。 (Sparrows are my friends. It built its nest near my house, a building I built when I got married to Xiao Huang. My friend made a home here, we all moved away and it hasn't come back yet. One day, it came back and I was so happy because it was my friend! The sparrow left the old house in search of a new home, but it did not find a home, nor a way home. One day a sparrow flew up to the window of my house and saw me standing in front of the window looking at it, and it thought I was dead and wept bitterly. The next day, the sparrow flew to me again and said, "Friend, don't be sad, I'm waiting for you here." "I opened the door and let it into the house to rest. After some time, when I went to see it again, it had grown up and no longer called my name. Its feathers were grey, and its eyes were black like eagle claws. I thought: The sparrow has returned, with a new face, and must be very happy. So I put up a banner in front of the door that reads "Magpie Dengmei" and bought a pot of flowers to put on the door. These days, the sparrow has blocked the house and refused to come out again.)	一座曾经给麻雀们提供无数巢穴的老教堂进行了修缮。当麻雀归来之时，教堂已面目一新。它们寻找自己的旧窠，却发现都被堵起来了。麻雀们嚷嚷着这幢建筑对它们来说已经没有用了。于是，纷纷离开了教堂，去寻找新的地方搭建巢穴了。 (An old church that once provided countless nests for sparrows was restored. When the sparrow returned, the church had a new look. They looked for their old kilns, but found that they were all blocked. The sparrows shouted that the building was no longer useful to them. So they left the church and went to find a new place to build a nest.)
Summarization	迪桑娜女包新款真皮斜挎单肩包头层牛皮斜挎小包包抽绳马鞍包拼色，精细车缝线，采用高质感半植物，细节展示，人性化肩带，多角度展示，商品信息，整齐精细的车缝线体，效果增加别样层次感，节都保持严谨心态，弹性。自然通透光泽，摔，自然有型，有分量感，使整体视觉效果增加别样层次感和精致。可调节肩带设计，根，颗粒有大小属于正常。现工匠精神，每个细，由调节长度，随心搭，配，成为时尚达人。分量感，使整体视觉，侧面五金与肩带衔接，据身高和使用习惯自，容量：若干化妆品、银包、手机，结合，利用皮料代替，容量：手提，度，适合都市时毗白领，日常外出皆可。和精致。结构：无，货号：81，吊牌价：子处于自然随性状态，如同当代社会人士的消费态，宽130mm，798元 (Desanna women's bag new leather cross-body shoulder bag first layer cowhide cross-body small bag drawstring saddle bag color, fine car stitching, the use of high-quality semi-plant, detail display, humanized shoulder straps, multi-angle display, product information, neat and fine car seam body, the effect increases the sense of layering, sections are maintained rigorous mentality, elasticity. Natural translucent luster, fall, natural shape, there is a sense of weight, so that the overall visual effect increases a different sense of layering and refinement. Adjustable shoulder strap design, roots, particles have sizes that belong to normal. Now the craftsman spirit, each fine, by adjusting the length, take it as you like, match, become a fashionista. The sense of weight, so that the overall vision, side hardware and shoulder straps, according to height and habits of use, capacity: a number of cosmetics, silver bags, mobile phones, combined, the use of leather instead, capacity: hand, degree, suitable for the city when the white-collar workers, daily outings can be. and exquisite. Structure: none, goods number: 81, tag price: sub is in a natural and casual state, just like the consumption state of contemporary social people, width130mm, 798 yuan)	迪桑娜女包新款真皮斜挎单肩包头层牛皮斜挎小包包抽绳马鞍包拼色，精细车缝线，采用高质感半植物，细节展示，人性化肩带，多角度展示，商品信息，整齐精细的车缝线体，效果增加别样层次感和精致。 (Desanna women's bag new leather cross-body shoulder bag first layer cowhide cross-body small bag drawstring saddle bag color, fine stitching, using high-texture semi-plant, detail display, humanized shoulder straps, multi-angle display, product information, neat and fine car seam body, the effect increases the sense of layering and refinement.)	抽绳的开口设计，方便进行调节的同时也扩充了包身的容量，轻松满足出行所需，小巧的包身设计，时髦精致不失韵味，拼色的设计，强化了包身的层次感，提升搭配的时尚品味。 (The opening design of the drawstring is convenient for adjustment at the same time, but also expands the capacity of the body, easily meet the needs of travel, small body design, fashionable and exquisite without losing charm, color design, strengthen the layering of the body, enhance the fashion taste of collocation.)
Paraphrase	如何增大内存 (How to increase memory)	内存怎么扩容 (How to expand the memory)	如何把内存变大 (How to make the memory bigger)

Table 12: Examples of evaluation data for CPM-3 in different modes with corresponding English text of input, output and target text.

Context	Control Signal	Output	Target
2006年	Keywords: 蔡畅,葛健豪,李特特 (Chang Cai, Jianhao Ge, Tete Li)	12月，蔡畅、葛健豪在《快乐男声》中演唱《月亮代表我的心》。2007年1月，李特特在《星光下的约定》中担任领唱。 (In December, Cai Chang and Ge Jianhao sang "The Moon Represents My Heart" in "Happy Male Voice". In January 2007, Li Tete served as the lead singer in "The Covenant Under the Stars".)	5月，中央电视台《走遍中国》栏目组为拍摄《女杰之乡——双峰》，特请来了葛健豪的外孙女、蔡畅的女儿李特特老人 (In May, in order to shoot "The Land of Female Masters - Twin Peaks", the CCTV "Walking All Over China" column group specially invited Ge Jianhao's granddaughter and Cai Chang's daughter Li Tete to shoot the elderly)
人物简介叶	Relations: 青田-所在行政领土实体-浙江 (Qingtian-Territorial-Zhejiang)	志平叶志平浙江青田人[1]生于1962年12月,1985年毕业于浙江大学生物系，获学士学位,1988年至1989年在浙大生命科学与技术系攻读硕士研究生。现为中国科学技术大学教授、博士生导 (Born in December 1962, Zhiping Ye Zhiping graduated from the Department of Biology of Zhejiang University with a bachelor's degree in 1985 and studied for a master's degree in the Department of Life Sciences and Technology of Zhejiang University from 1988 to 1989. He is currently a professor and doctoral supervisor at the University of Science and Technology of China)	人物简介叶震宇，男，祖籍浙江青田，1948年出生于温州，1973年移民荷兰 (Character Profile Ye Zhenyu, male, ancestral home of Qingtian, Zhejiang, born in Wenzhou in 1948, immigrated to the Netherlands in 1973)
-	Events: 谓词: 发生 受事人: 重庆城口县 主体: 山体滑坡 (Verb: Occur Object: Chengkou County, Chongqing Subject: Landslide)	重庆城口县发生山体滑坡致2人死亡 (A landslide occurred in Chengkou County, Chongqing, killed 2 people)	重庆城口县发生山体滑坡 1人死亡6人失联 (A landslide occurred in Chengkou County, Chongqing, 1 person died and 6 people were missing)

Table 13: Examples of evaluation data by CPM-3 with different control signals.