TextBox 2.0: A Text Generation Library with Pre-trained Language Models

Tianyi Tang[1,5†], **Junyi Li**[1,3†], **Zhipeng Chen**[2†], **Yiwen Hu**[2], **Zhuohao Yu**[2], **Wenxun Dai**[4], **Wayne Xin Zhao**[1,5*], **Jian-Yun Nie**[3], and **Ji-Rong Wen**[1,2,5]

[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Information, Renmin University of China
[3]DIRO, Université de Montréal [4]Xidian University
[5]Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE
steventianyitang@outlook.com lijunyi@ruc.edu.cn batmanfly@gmail.com

Abstract

To facilitate research on text generation, this paper presents a comprehensive and unified library, **TextBox 2.0**, focusing on the use of pre-trained language models (PLMs). To be *comprehensive*, our library covers 13 common text generation tasks and their corresponding 83 datasets and further incorporates 45 PLMs covering general, translation, Chinese, dialogue, controllable, distilled, prompting, and lightweight PLMs. We also implement 4 efficient training strategies and provide 4 generation objectives for pre-training new PLMs from scratch. To be *unified*, we design the interfaces to support the entire research pipeline (from data loading to training and evaluation), ensuring that each step can be fulfilled in a unified way. Despite the rich functionality, it is easy to use our library, either through the friendly Python API or command line. To validate the effectiveness of our library, we conduct extensive experiments and exemplify four types of research scenarios. The project is released at the link: https://github.com/RUCAIBox/TextBox#2.0.

1 Introduction

Text generation, aiming to generate human-like texts on demand, has been a fundamental technique in many text applications, such as machine translation (Dabre et al., 2020), text summarization (El-Kassas et al., 2021), and dialogue system (Chen et al., 2017). Recently, pre-trained language models (PLMs) such as BART (Lewis et al., 2020) have been the mainstream approach to developing effective text generation models. With the great advances in text generation, it has become increasingly important to reproduce, develop, and compare various text generation models in a reliable, flexible, and unified way.

Considering the rapid progress of PLMs on text generation, in this paper, we present a significant extension of a previously released text generation library, *TextBox 1.0* (Li et al., 2021), called **TextBox 2.0**. Different from TextBox 1.0 and other text generation libraries (Miller et al., 2017; Klein et al., 2018; Zhu et al., 2018) (mostly including classical models based on recurrent neural networks or generative adversarial networks), this extension mainly focuses on building a comprehensive and unified framework for better supporting PLM-based text generation models. Although some libraries (*e.g.,* Fairseq (Ott et al., 2019) and Hugging Face (Wolf et al., 2020)) also include PLMs, they are designed for performing myriad NLP tasks (only considering a few text generation tasks). Moreover, they don't maintain a complete evaluation pipeline (*e.g.,* data loading, training, inference, and evaluation) specially designed for text generation. Thus, it is not fully suited for developing and evaluating text generation models in a unified way.

In order to better facilitate research on text generation, **TextBox 2.0** introduces a series of new features for supporting the use of PLMs, which can be summarized into three major aspects:

• *Generation Tasks*: Our library supports 13 commonly studied text generation tasks (*e.g.,* translation and story generation) and their corresponding 83 datasets, including most of the existing mainstream tasks and datasets for research. We reorganize these datasets so that they are framed in a unified text-to-text format. Users can simply set the dataset via the command line or configuration file without additional preprocessing efforts.

• *Generation Models*: As a key contribution, our library incorporates 45 PLMs, covering the categories of general, translation, Chinese, dialogue, controllable, distilled, prompting, and lightweight PLMs. We unify the interface to use existing PLMs and incorporate new PLMs, and it is convenient to run different PLMs for a specified task in our

[†]Equal contribution.
[*]Corresponding author

Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 435 - 444
December 7-11, 2022

Aspects	TextBox 1.0	TextBox 2.0
Tasks 6 *v.s.* 13	Summarization, translation, dialogue, unconditional generation, attribute-to-text generation, poem generation	Summarization, translation, dialogue, data-to-text, question generation, question answering, story generation, commonsense generation, Chinese generation, paraphrase, style transfer and simplification
Models 6 *v.s.* 45	**VAE**: LSTMVAE, CNNVAE, CVAE, HybridVAE **GAN**: SeqGAN, TextGAN, RankGAN, MaliGAN, LeakGAN, MaskGAN **PLM**: GPT-2, XLNet, BERT2BERT, T5, BART, ProphetNet **Seq2Seq**: RNN, Transformer, Attr2Seq, Context2Seq, HRED	**General**: GPT-2, BERT2BERT, BART, T5, ProphetNet, GPT, GPT-Neo, OPT, UniLM, MASS, PEGASUS, MVP, Bigbird, LED **Translation**: mBART, mT5, Marian, M2M 100, NLLB, XLM **Chinese**: CPM, CPT, Chinese-BART, Chinese-T5, Chinese-GPT2 **Dialogue**: Blenderbot and DialoGPT **Controllable**: CTRL and PPLM **Distilled**: DistilGPT2 and DistilBART **Prompting**: PTG and Context-Tuning **Lightweight**: Adapter, Prefix-tuning, Prompt tuning, LoRA, BitFit, P-Tuning v2
Training Strategies	Distributed data parallel	Distributed data parallel, efficient decoding, hyper-parameter optimization, repeated experiments, pre-training objectives

Table 1: Comparison of TextBox 1.0 and TextBox 2.0. We also present a comparison of the numbers of *tasks* and pre-trained *models* between the two versions.

library. We also provide a standard way to compare these models and analyze the generated results.

• *Training Strategies*: To support the optimization of PLMs, we provide four efficient and robust training strategies (*e.g.,* efficient decoding) and four pre-training objectives (*e.g.,* denoising auto-encoding) for text generation. These strategies make optimizing text generation models more efficient and reliable. Users can either pre-train a new model from scratch or fine-tune a pre-trained model for research purposes.

As another merit, TextBox 2.0 has been largely aligned with our previous survey on PLM-based text generation (Li et al., 2022b) in terms of task, model, and training. It will be meaningful for beginners to explore and learn text generation models with the survey and supporting libraries.

To summarize, TextBox 2.0 has contributed a significant addition to the previous version (see Table 1 for a detailed comparison) to better support the use of PLMs for text generation. It implements and maintains a unified way to conduct research on text generation with 45 included models, covering 13 tasks, and 83 datasets. We also perform extensive test experiments, and these results show that TextBox 2.0 can produce very competitive performance compared to the original implementations.

2 Library Design

In order to facilitate PLM-based text generation research, TextBox 2.0 has introduced various new features, mainly from three aspects: *generation tasks*, *generation models*, and *training strategies*.

2.1 Generation Tasks

Since there are various text generation applications, we include 13 widely studied tasks and collect the corresponding 83 datasets.

Tasks. These 13 tasks in TextBox 2.0 include text summarization, machine translation, open-ended dialogue system, data-to-text generation, question generation, question answering, story generation, task-oriented dialogue system, commonsense generation, paraphrase generation, text style transfer, and text simplification. Besides these English-centric tasks, we also include Chinese generation tasks. Existing PLM-based libraries such as Hugging Face (Wolf et al., 2020) are focused on performing extensive NLP tasks and only consider a few text generation tasks (mainly text summarization and machine translation), which are not comprehensive for text generation research.

Datasets. For each task, we collect widely-used datasets and reorganize them in a unified text-to-text format. In total, we include 83 datasets, and report their details on the page[1], including the dataset description, basic statistics, and training/validation/testing samples. In addition, we build a leaderboard for each dataset by collecting the automatic results and generated texts of the latest research. It is convenient for users to quickly learn about the baselines and their results. We also encourage community users to collaboratively maintain the leaderboard and submit their model results.

[1] `https://github.com/RUCAIBox/TextBox#dataset`

Metrics. To conduct evaluations with these tasks and datasets, TextBox 2.0 supports four categories of automatic metrics: (1) lexical metrics, such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), to measure the n-gram overlap between generated texts and golden texts; (2) semantic metrics, such as BERTScore (Zhang et al., 2020b) and style strength (Lai et al., 2021), to compare the texts at sentence level; (3) diversity metrics, such as Distinct (Li et al., 2016) and Self-BLEU (Zhu et al., 2018), to evaluate the lexical diversity of generated texts; (4) accuracy metrics, such as exact match (Rajpurkar et al., 2016) and inform (Budzianowski et al., 2018a), to calculate the precision of important phrases. In total, we include 12 general metrics and 5 task-specific metrics[2].

Besides the analysis using automatic metrics, TextBox 2.0 provides several visualization tools to explore and analyze the generated texts in various dimensions (Liu et al., 2021b; Tuckute et al., 2022). For instance, Figure 2 shows how it offers new insights to improve summarization tasks (details can be found in Section 4.3).

2.2 Generation Models

To support the rapid progress of PLMs on text generation, TextBox 2.0 incorporates 45 PLMs[3] and aims to build a unified and standardized framework based on PLMs. We list some included models as follows:

- **General PLMs**: GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2020);
- **Translation PLMs**: mBART (Liu et al., 2020) and XLM (CONNEAU and Lample, 2019);
- **Chinese PLMs**: CPM (Zhang et al., 2021) and CPT (Shao et al., 2021);
- **Dialogue PLMs**: DialoGPT (Zhang et al., 2020c) and Blenderbot (Roller et al., 2021);
- **Controllable PLMs**: CTRL (Keskar et al., 2019) and PPLM (Dathathri et al., 2020);
- **Distilled PLMs**: DistilGPT2 (Sanh et al., 2019) and DistilBART (Shleifer and Rush, 2020).
- **Prompting PLMs**: PTG (Li et al., 2022a) and Context-Tuning (Tang et al., 2022);
- **Lightweight modules**: Adapter (Houlsby et al., 2019), Prefix-tuning (Li and Liang, 2021).

The wide coverage of PLMs makes it possible to deal with different text generation tasks using TextBox 2.0. For example, to perform specific tasks such as dialogue system, users can adopt task-specific PLMs such as DialoGPT; to deal with Chinese generation tasks, users can adopt CPT. In resource-constrained situations, lightweight PLMs such as prefix-tuning can be a good choice.

2.3 Training Strategies

TextBox 2.0 provides four pre-training objectives to help users pre-train a model from scratch, including language modeling (Radford et al., 2019), masked sequence-to-sequence modeling (Song et al., 2019), denoising auto-encoding (Lewis et al., 2020), and masked span prediction (Raffel et al., 2020). These pre-training tasks can also be utilized for domain-adaptive pre-training and task-adaptive pre-training (Gururangan et al., 2020) to tailor existing PLM to the domain of a target task.

Also, TextBox 2.0 provides four useful training methods for improving the optimization of PLMs. It supports distributed data parallel to implement models on multiple GPUs and machines to improve the efficiency of pre-training and fine-tuning. We incorporate *Accelerate*[4] to support distributed training with a simple API. To further accelerate the decoding efficiency, we integrate FastSeq (Yan et al., 2021) to optimize the decoding process by attention cache optimization, repeated n-gram detection, and asynchronous parallel I/O.

Moreover, TextBox 2.0 enables users to adjust and select hyper-parameters automatically. Based on the library Hyperopt (Bergstra et al., 2013), users just need to set the parameter range and search methods, and then the optimal hyper-parameters and corresponding results will return. It is useful for PLMs to search for hyper-parameters such as batch size and learning rate. Our library also supports performing repeat experiments using different random seeds in one command line, which is especially useful to alleviate randomness especially under few-shot settings.

3 Library Usage

In this section, we introduce how to use our library in four different kinds of research scenarios by showing the example codes.

Reproducing existing models. TextBox 2.0 includes various PLMs and supports many text generation tasks and datasets. It is convenient for users

[2] https://github.com/RUCAIBox/TextBox#evaluation

[3] https://github.com/RUCAIBox/TextBox#model

[4] https://github.com/huggingface/accelerate

```
python run_textbox.py \
    --dataset=xsum --model=pegasus \
    --model_path=google/pegasus-large
```

(a) Example for reproducing existing models

```
# hyper.test
Learning_rate choice [1e-5, 3e-5]
train_batch_size choice [64, 256]

# command line instruction
python run_hyper.py \
    --space=hyper.test \
    --dataset=xsum --model=pegasus \
    --model_path=google/pegasus-large
```

(b) Example for hyper-parameter optimization

```
class new_model(AbstractModel):
    def __init__(self):
        …
        self.t5 = T5_MODEL
        self.gnn = GNN_MODEL
    def forward(self, input, label):
        …
        embeds = self.gnn(input)
        output = self.t5(embeds)
        return loss_func(output, label)
    def generate(self, input):
        …
        embeds = self.gnn(input)
        return self.t5.generate(embeds)
```

(c) Example for implementing a new model

```
python run_textbox.py \
    --dataset=wudao --model=bart \
    --pretrain_task=denoising
```

(d) Example for pre-training a Chinese BART

```
accelerate config
accelerate launch run_textbox.py \
    --dataset=wudao --model=bart \
    --pretrain_task=denoising
```

(e) Example for multi-GPU pre-training

```
python run_analysis.py \
    --dataset=cnndm \
    BART_output.txt T5_output.txt
```

(f) Example for analyzing

Figure 1: Example usage of our TextBox 2.0.

to quickly run existing PLMs and reproduce results for each dataset. In particular, users only need to specify the dataset and model by setting the configurations `dataset`, `model`, and `model_path`, within a simple command line.

Figure 1(a) presents an example to fine-tune PEGASUS (Zhang et al., 2020a) on XSum (Narayan et al., 2018) dataset. Moreover, TextBox 2.0 enables users to conduct hyper-parameter optimization by only providing a list of possible values. Figure 1(b) shows an example that automatically adjusts the hyper-parameters `learning_rate` and `batch_size` from the ranges $[1 \times 10^{-5}, 3 \times 10^{-5}]$ and $[64, 256]$, respectively.

Implementing a new model. Since TextBox 2.0 builds a unified pipeline for text generation research, users only need to define a new model class without considering other procedures to implement a new model. Specially, users should first inherit from our base model class `AbstractModel` before specifying three specific model functions: (1) `__init__()`: this function initializes the architectures and parameters of the model; (2) `forward()`: this function is used to calculate the loss for optimization during training; (3) `generate()`: this function generates texts based on input during inference.

Figure 1(c) presents an example of implementing a new model for the KG-to-text generation task . In this example, the model adopts a graph neural network (GNN) to encode KG and then uses T5 (Raffel et al., 2020) to generate texts. We first define the GNN and T5 models in the `__init__()` function. Then, we use GNN to encode KG to embeddings as the input of T5 and compute the loss according to target labels in the `forward()` function. Finally, we use a similar process to generate text in the `generate()` function.

Pre-training a new model. In TextBox 2.0, we provide several pre-training objectives for users to pre-train new models from scratch. Specifically, users just need to specify the pre-training task, pre-training corpus, and architecture by setting `pretrain_task`, `dataset`, and `model`. Figure 1(d) shows an example that pre-trains a Chinese BART on the WuDaoCorpora (Yuan et al., 2021) using the denoising pre-training objective.

To improve the pre-training efficiency, TextBox 2.0 supports distributed data parallel and efficient decoding (Section 2.3). Figure 1(e) shows an illustrative example of how users can use the `accelerate` command to set configurations of multiple devices and launch the training code.

Analyzing generated results. Besides simply obtaining the evaluation results, our library provides several visualization analysis mechanisms to perform deep analysis on the generated results of models. For example, we support the use of the statistical chart to analyze the mean and standard deviation scores for different sentence lengths. These methods can help users learn about the advantages and disadvantages of different models in a detailed comparison. Figure 1(f) shows an example of how to run the analysis using a simple command line and the results can be found in Figure 2. This example compares the generated texts of BART and T5 on the CNN/DailyMail dataset.

4 Experiments

In this section, we conduct extensive experiments to verify the generation abilities of TextBox 2.0.

4.1 Result Reproduction

As an open-source library, TextBox 2.0 should be able to reproduce the results of existing work effectively. To verify this, we select a number of

	Text Summarization			Text Simplification			Chinese Generation			Translation	
	R-1	R-2	R-L	B-4	ME	R-2	LCSTS	CSL	ADGEN	En→Ro	Ro→En
BART	44.16[a]	21.28	40.90	88.30[b]	55.60	86.10	40.60[c]	64.20	10.00	37.70[d]	37.80
BART (ours)	$44.47_{0.10}$	$21.50_{0.14}$	$41.35_{0.08}$	$90.81_{0.24}$	$57.58_{0.19}$	$83.36_{0.07}$	$42.96_{0.18}$	$64.34_{0.63}$	$10.20_{0.15}$	$37.20_{0.17}$	$37.48_{0.31}$
	Data-to-text Generation			**Commonsense Generation**			**Question Generation**			**QA**	
	B-4	ME	R-L	B-4	CIDEr	SPICE	B-4	ME	R-L	F1	EM
BART	64.55[e]	46.51	75.13	27.50[f]	14.12	30.00	22.00[g]	26.40	50.30	91.56[h]	84.23
BART (ours)	$67.33_{0.06}$	$47.78_{0.07}$	$76.83_{0.04}$	$28.18_{0.45}$	$12.98_{0.13}$	$33.00_{0.40}$	$25.08_{0.13}$	$26.73_{0.18}$	$52.55_{0.07}$	$93.04_{0.08}$	$86.44_{0.21}$
	Open-ended Dialogue System				**Task-oriented Dialogue System**				**Story Generation**		
	B-1	B-2	D-1	D-2	B-4	Success	Inform	Comb.	B-1	B-2	D-4
BART	49.90[g]	40.00	1.30	8.00	17.89[i]	74.91	84.88	97.78	30.70[j]	13.30	69.90
BART (ours)	$49.58_{1.12}$	$39.24_{0.90}$	$1.44_{0.09}$	$8.89_{0.57}$	$20.17_{0.63}$	$75.40_{1.22}$	$84.40_{1.15}$	$100.07_{0.53}$	$33.79_{0.13}$	$15.78_{0.21}$	$78.76_{2.15}$
	Paraphrase Generation					**Style Transfer (E&M)**			**Style Transfer (F&R)**		
	B-4	ME	R-1	R-2	R-L	B-4	Acc.	HM	B-4	Acc.	HM
BART	47.30[k]	49.70	73.30	54.10	75.10	76.50[l]	92.90	83.90	79.30	92.00	85.20
BART (Ours)	$48.35_{0.70}$	$50.60_{0.49}$	$74.16_{0.47}$	$55.25_{0.74}$	$75.84_{0.42}$	$76.93_{0.55}$	$94.37_{0.87}$	$84.74_{0.05}$	$80.11_{0.29}$	$92.29_{0.37}$	$85.77_{0.10}$

Table 2: The results of BART on thirteen tasks from the original papers and our TextBox 2.0. QA is short for question answering. B, R, D, ME, EM, HM, Acc., and Comb. denote BLEU, ROUGE, Distinct, METEOR, exact match, harmonic mean, accuracy, and combined score, respectively. LCSTS, CSL, ADGEN, and En↔Ro are evaluated using the R-L, R-L, B-4, and B-4 metrics, respectively. [a](Lewis et al., 2020) [b](Gehrmann et al., 2021) [c](Shao et al., 2021) [d](Liu et al., 2020) [e](Ke et al., 2021) [f](Lin et al., 2020a) [g](Liu et al., 2021a) [h](Xu et al., 2021) [i](Lin et al., 2020b) [j](Guan et al., 2021) [k](Sun et al., 2021) [l](Lai et al., 2021)

widely-used datasets for each task (introduced in Section 2.1) and compare the results conducted by TextBox 2.0 with those in the original papers. We totally evaluate 13 tasks using 14 datasets, including CNN/DailyMail (See et al., 2017), WikiAuto + Turk (Liu et al., 2021a), LCSTS (Hu et al., 2015), CSL[5], ADGEN (Shao et al., 2019), WMT 16 English-Romanian (En↔Ro) (Bojar et al., 2016), WebNLG 2.1 (Gardent et al., 2017), CommonGen (Lin et al., 2020a), SQuAD (Rajpurkar et al., 2016), PersonaChat (Zhang et al., 2018), MultiWOZ 2.0 (Budzianowski et al., 2018b), ROCStories (Mostafazadeh et al., 2016), GYAFC (E&M and F&R) (Rao and Tetreault, 2018), and Quora (Kumar et al., 2020).

Since BART is the prevalent PLM for text generation, we endeavor to reproduce existing works with $\text{BART}_{\text{LARGE}}$[6]. For all experiments, we employ the sequence-to-sequence cross-entropy loss with a label smoothing factor of 0.1 as the objective function. We optimize the model using AdamW (Loshchilov and Hutter, 2019) with a constant learning rate of 3×10^{-5}. The accumulated batch size is set to 192. During inference, we apply beam search with a beam size of 5 and no-repeat n-gram size of 3. To reduce randomness, we report the mean and standard deviation of our results based on three random seeds: 2020, 2021, and 2022. All codes are implemented in PyTorch 1.11.0 on Ubuntu SMP 20.04.1 (Linux 5.15.0-46) with one GPU (NVIDIA GeForce RTX 3090 24GB).

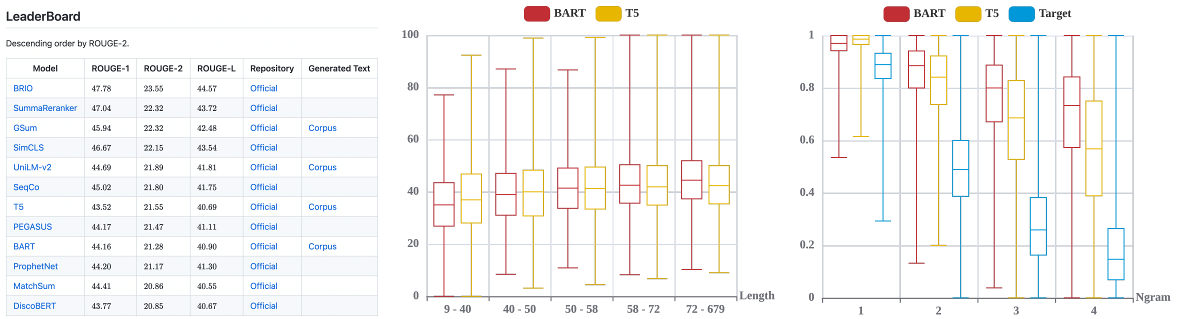
To conduct these experiments, we only need to run the script shown in Figure 1 (a) with different `dataset` names. As shown in Table 2, our TextBox 2.0 can faithfully reproduce the results reported in existing work. Remarkably, our library achieves better performances than original works on 37 of the 44 metrics evaluated. It might be because we adopt optimization strategies such as label smoothing and large batch sizes.

Library	**Preparation** (minutes)	**Training** (minutes)	**Generation** (minutes)
Fairseq	$2.93_{0.02}$	$410.05_{8.86}$	$79.24_{1.50}$
Hugging Face	$4.02_{0.12}$	$416.25_{4.47}$	$75.69_{2.53}$
TextBox 2.0	$3.81_{0.14}$	$393.99_{5.09}$	$27.05_{1.03}$

Table 3: Efficiency comparison of three libraries for $\text{BART}_{\text{LARGE}}$ fine-tuned on CNN/DailyMail. The preparation stage consists of configuration loading, text tokenization, and necessary initialization options. The training stage takes time for fine-tuning on the training set in one epoch. The generation stage takes time to generate on the test set with a beam size of 5.

[5] https://github.com/CLUEbenchmark/CLGE

[6] For translation tasks, we utilize mBART-CC25 (Liu et al., 2020). For Chinese generation tasks, we utilize Chinese $\text{BART}_{\text{LARGE}}$ (Shao et al., 2021).



LeaderBoard

Descending order by ROUGE-2.

Model	ROUGE-1	ROUGE-2	ROUGE-L	Repository	Generated Text
BRIO	47.78	23.55	44.57	Official	
SummaReranker	47.04	22.32	43.72	Official	
GSum	45.94	22.32	42.48	Official	Corpus
SimCLS	46.67	22.15	43.54	Official	
UniLM-v2	44.69	21.89	41.81	Official	Corpus
SeqCo	45.02	21.80	41.75	Official	
T5	43.52	21.55	40.69	Official	Corpus
PEGASUS	44.17	21.47	41.11	Official	
BART	44.16	21.28	40.90	Official	Corpus
ProphetNet	44.20	21.17	41.30	Official	
MatchSum	44.41	20.86	40.55	Official	
DiscoBERT	43.77	20.85	40.67	Official	

(a) Leaderboard of CNN/DailyMail

(b) ROUGE-L scores of BART and T5 for different input lengths

(c) N-gram overlap of target and generated texts with input document

Figure 2: The partial visualization analysis on CNN/DailyMail dataset. The whole one can be found at https://github.com/RUCAIBox/TextBox/blob/2.0.0/asset/example-analysis.html.

4.2 Efficiency Comparison

In addition to accurately reproducing results, we have optimized TextBox 2.0 for computational efficiency. We streamline the training process and support efficient decoding strategies. To compare the efficiency, we choose the well-known PLM libraries Fairseq[7] and Hugging Face[8], and then test the time consumption under identical settings described in Section 4.1.

From the results in Table 3, we can see that our TextBox 2.0 is more efficient than Fairseq and Hugging Face. During training, TextBox 2.0 simplifies the training process and reduces the time spent on non-essential functions such as trainer management and loss tracking. In the generation process, our library is significantly faster than the other two libraries due to the incorporation of efficient decoding strategies introduced in Section 2.3.

4.3 Visualization Analysis

Besides reproducing a model, it is also important to compare existing methods, analyze the generated texts, and explore directions for improvement. Our library sets a specific leaderboard for each dataset, including basic metric results, author repositories, and generated texts. Figure 2 (a) showcases the leaderboard for the CNN/DailyMail dataset.

Users can also utilize TextBox 2.0 to conduct visualization analysis for specified models. For example, our library can automatically plot the boxplot of the ROUGE-L score for different input lengths and the n-gram overlap of target and generated texts with the input document. From the results in Figure 2 (b), we can find that T5 excels at short document summarization while BART excels at long document summarization. It is useful to analyze and improve the deficiencies of text generation models or obtain better performance by combining their results. As another example, Figure 2 (c) illustrates that BART and T5 have a significantly higher n-gram overlap ratio than golden sentences, indicating that they tend to "copy" the input document rather than "summarize" it. From such analysis results, users can apply the methods proposed by Goyal et al. (2022) to alleviate it.

5 Conclusion

This paper presented **TextBox 2.0**, a comprehensive and unified library for conducting research on PLM-based text generation. Our library makes significant extensions in three major aspects, namely generation tasks (13 tasks and 83 datasets), generation models (45 PLMs), and training strategies (*e.g.*, distributed data parallel and efficient decoding). Results from extensive test experiments demonstrate that our library can accurately reproduce existing models. Besides, we also provide a series of utility tools to better analyze and explore the generated results. To summarize, our library can be very useful to facilitate text generation research, and our team will improve this library with regular updates.

Acknowledgement

This work was partially supported by Beijing Natural Science Foundation under Grant No. 4222027, and Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098. Xin Zhao is the corresponding author.

[7] We utilize the code from Fairseq 0.12.2.

[8] We utilize the code from Transformers 4.20.1.

References

James Bergstra, Daniel Yamins, and David Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 115–123.

Ondřej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Liane Guillou, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Pavel Pecina, Martin Popel, Philipp Koehn, Christof Monz, Matteo Negri, Matt Post, Lucia Specia, Karin Verspoor, Jörg Tiedemann, and Marco Turchi, editors. 2016. *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*. Association for Computational Linguistics, Berlin, Germany.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018a. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018b. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *SIGKDD Explor. Newsl.*, 19(2).

Alexis CONNEAU and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5).

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.

Tanya Goyal, Jiacheng Xu, Junyi Jessy Li, and Greg Durrett. 2022. Training dynamics for text summarization models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2061–2073, Dublin, Ireland. Association for Computational Linguistics.

Jian Guan, Xiaoxi Mao, Changjie Fan, Zitao Liu, Wenbiao Ding, and Minlie Huang. 2021. Long text generation by modeling sentence-level and discourse-level coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6379–6393, Online. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea

Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *ICML*.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A large scale Chinese short text summarization dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1967–1972, Lisbon, Portugal. Association for Computational Linguistics.

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. JointGT: Graph-text joint representation learning for text generation from knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2526–2538, Online. Association for Computational Linguistics.

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 Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander Rush. 2018. OpenNMT: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 177–184, Boston, MA. Association for Machine Translation in the Americas.

Ashutosh Kumar, Kabir Ahuja, Raghuram Vadapalli, and Partha Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8:329–345.

Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2021. Thank you BART! rewarding pre-trained models improves formality style transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 484–494, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Gaole He, Jinhao Jiang, Xiaoxuan Hu, Puzhao Xie, Zhipeng Chen, Zhuohao Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2021. TextBox: A unified, modularized, and extensible framework for text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 30–39, Online. Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Jian-Yun Nie, Ji-Rong Wen, and Xin Zhao. 2022a. Learning to transfer prompts for text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3506–3518, Seattle, United States. Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2022b. A survey of pretrained language models based text generation. *arXiv preprint arXiv:2201.05273*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020a. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

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.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020b. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405, Online. Association for Computational Linguistics.

Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, Pengcheng Wang, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, Ruofei Zhang, Winnie Wu, Ming Zhou, and Nan Duan. 2021a. GLGE: A new general language generation evaluation benchmark. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 408–420, Online. Association for Computational Linguistics.

Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaichen Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig. 2021b. ExplainaBoard: An explainable leaderboard for NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 280–289, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. ParlAI: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84, Copenhagen, Denmark. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

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*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Fei Yang, Li Zhe, Hujun Bao, and Xipeng Qiu. 2021. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *arXiv preprint arXiv:2109.05729*.

Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. Long and diverse text generation with planning-based hierarchical variational model. 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 3257–3268, Hong Kong, China. Association for Computational Linguistics.

Sam Shleifer and Alexander M Rush. 2020. Pre-trained summarization distillation. *arXiv preprint arXiv:2010.13002*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Jiao Sun, Xuezhe Ma, and Nanyun Peng. 2021. AESOP: Paraphrase generation with adaptive syntactic control. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5176–5189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Context-tuning: Learning contextualized prompts for natural language generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6340–6354, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Greta Tuckute, Aalok Sathe, Mingye Wang, Harley Yoder, Cory Shain, and Evelina Fedorenko. 2022. SentSpace: Large-scale benchmarking and evaluation of text using cognitively motivated lexical, syntactic, and semantic features. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, pages 99–113, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Peng Xu, Davis Liang, Zhiheng Huang, and Bing Xiang. 2021. Attention-guided generative models for extractive question answering. *arXiv preprint arXiv:2110.06393*.

Yu Yan, Fei Hu, Jiusheng Chen, Nikhil Bhendawade, Ting Ye, Yeyun Gong, Nan Duan, Desheng Cui, Bingyu Chi, and Ruofei Zhang. 2021. FastSeq: Make sequence generation faster. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 218–226, Online. Association for Computational Linguistics.

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 the 37th International Conference on Machine Learning*, pages 11328–11339.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020c. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, et al. 2021. Cpm: A large-scale generative chinese pre-trained language model. *AI Open*, 2:93–99.

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, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1097–1100. ACM.