🎓 OpenBA: An Open-sourced 15B Bilingual Asymmetric seq2seq Model Pre-trained from Scratch

Juntao Li[*], **Zecheng Tang**[†], **Yuyang Ding**[†], **Pinzheng Wang**[†]

Pei Guo, **Wangjie You**, **Dan Qiao**, **Wenliang Chen**, **Guohong Fu**, **Qiaoming Zhu**,

Guodong Zhou[‡], **Min Zhang**[‡]

Soochow University

Abstract

Large language models (LLMs) with billions of parameters have demonstrated outstanding performance on various natural language processing tasks. This report presents OpenBA, an open-sourced 15B bilingual asymmetric seq2seq model, to contribute an LLM variant to the Chinese-oriented open-source model community. We enhance OpenBA with effective and efficient techniques as well as adopt a three-stage training strategy to train the model from scratch. Our solution can also achieve very competitive performance with only 380B tokens, which is better than LLaMA-70B on the BELEBELE benchmark, BLOOM-176B on the MMLU benchmark, GLM-130B on the C-Eval (hard) benchmark. This report provides the main details to pre-train an analogous model, including pre-training data processing, Bilingual Flan data collection, the empirical observations that inspire our model architecture design, training objectives of different stages, and other enhancement techniques. We have refactored our code to follow the design principles of the Huggingface Transformers Library, making it more convenient for developers to use, and released checkpoints of different training stages at `https://huggingface.co/openBA`. More details of our project are available at `https://github.com/OpenNLG/openBA.git`.

1 Introduction

The scaling law (Kaplan et al., 2020; Clark et al., 2022; Hoffmann et al., 2022; Touvron et al., 2023a) of language models has brought unprecedented success. These large language models pre-trained on massive textual data demonstrate enormous superiority over previous paradigms for various fields and even have obtained newly emergent capabilities. Though very powerful and developing rapidly, these models at scale are still far from perfect or satisfactory for most of the real-world usages. To advance the development of LLMs, the open-source community has made great efforts to provide strong and publicly accessible LLMs, covering different data sources, architectures, language modeling objectives, training pipelines, model scales, and language of expertise, e.g., BLOOM (Scao et al., 2022), LLaMA (Touvron et al., 2023a,b), FlanT5 (Chung et al., 2022), AlexaTM (Soltan et al., 2022).

As for Chinese, the open-source community has also released many large language models either by pre-training from scratch, e.g., GLM (Zeng et al., 2022), Baichuan (Inc., 2023) or conducting

---

[*] Project Leader. `ljt@suda.edu.cn`

[†] Equal Contribution. `{zctang,yyding23,pzwang1}@stu.suda.edu.cn`

[‡] Corresponding Author: `{gdzhou,minzhang}@suda.edu.cn`

Technical Report

further fine-tuning on existing open-sourced multilingual models, e.g., Huatuo (Wang et al., 2023), Luotuo (Ziang Leng & Li, 2023), Phoenix (Chen et al., 2023), Chinese-LLaMA (Cui et al., 2023), MOSS (Sun et al., 2023). These publicly available LLMs provide researchers and developers with strong general language models (i.e., the framework used by GLM (Du et al., 2022)) and different decoder-only variants, but leaving the Encoder-Decoder framework (e.g., Flan-T5 (Chung et al., 2022)) under-explored, which has been proven universally effective for different prompt settings (zero-shot, few-shot, and chain-of-thought) (Longpre et al., 2023) and various tasks (e.g., language understanding, commonsense reasoning, question answering, information retrieval, and multi-turn chit-chat conversation) (Tay et al., 2022; Zheng et al., 2023).

To fill this blank, we contribute an open-sourced 15B bilingual asymmetric seq2seq model (OpenBA) pre-trained from scratch, providing not only the model checkpoints but also the data collection and processing details to construct pre-training data and bilingual Flan collection from openly accessible data resources (e.g., Common Crawl, the Pile corpus, C-Book), the motivations and empirical observations for the model architecture design, and the key details of other enhancement strategies. Concretely, our collected pre-training data consists of balanced English and Chinese tokens to make the Chinese language modeling benefit from high-quality English data. Since it is difficult to construct a Flan-like Chinese collection that covers extensive tasks and settings from open resources, we incorporate more English data sampled from the Flan collection in our Bilingual-Flan corpus. Unlike the vanilla Flan-T5 (Chung et al., 2022) of a balanced encoder-decoder structure and the asymmetric deep-encoder shallow-decoder in AlexaTM (Soltan et al., 2022), we utilize another asymmetric model structure, i.e., shallow-encoder deep decoder to enhance the generation capability, which is motivated by our empirical study in Sec. 5.1. Our training process comprises three different stages, including the UL2 pre-training, length-adaptation, and Flan training. We also incorporate enhancement strategies in model architecture and training to improve model capability, training stability, and efficiency.

Evaluations across different benchmarks (MMLU, CMMLU, C-Eval, SuperGLUE, BELEBELE, BBH) and tasks (e.g., understanding, reasoning, and generation) have demonstrated the effectiveness of our model in different settings (zero-shot, few-shot, held-in, and held-out). Though merely trained on 380B tokens, our model can surpass many representative models, e.g., LLaMA-70B on BELEBELE, BLOOM-176B on MMLU, ChatGLM-6B on CMMLU and C-Eval. Throughout the whole training process, OpenBA-15B produces around $6.5\ tCO_{2eq}\ in\ total$, which is much less than the LLaMA-7B model that costs $14\ tCO_{2eq}$. All the implementation details are open-accessible, not limited to data collection and processing, codes, model checkpoints, and evaluations. As we are still working on a few directions to improve and apply the OpenBA model, we welcome any comments and suggestions and look forward to further cooperation with the open-source community.

2 Related Work

2.1 Large Language Models

Benefiting from scaling law (Kaplan et al., 2020; Clark et al., 2022; Hoffmann et al., 2022) and the growth of computational resources (Schaller, 1997), the recent years have witnessed the remarkable evolution in the field of LLMs, which pushes the boundaries of various NLP tasks (Radford et al., 2018; Brown et al., 2020; Zeng et al., 2021; Sun et al., 2021; Zhang & Li, 2021; Zhang et al., 2021, 2022; Touvron et al., 2023a). The introduction of transformer model (Vaswani et al., 2017) is a notable turning point in the NLP field, as models based on such an architecture like GPT-2 (Radford et al., 2019) and T5 (Raffel et al., 2020) have demonstrated outstanding performance across a wide range of NLP tasks by unifying the formulation of different tasks and scaling up the model size. This trend has continued with the advent of GPT-3 (Brown et al., 2020), which brings about groundbreaking advancements in scaling with its 175-billion-parameter model. Consequently, the research community has gradually recognized the benefits of LLMs, leading to a series of subsequent models following in rapid succession, such as Gopher (Rae et al., 2021), Megatron-Turing (Smith et al., 2022), Chinchilla (Hoffmann et al., 2022), BLOOM (Scao et al., 2022), LLaMA (Touvron et al., 2023a,b), ChatGPT (Ouyang et al., 2022; Bubeck et al., 2023), Falcon (Penedo et al., 2023a), etc. These models have consistently advanced the frontiers in the NLP domain, promoting ongoing development and progress. However, in this process, two serious issues have gradually emerged. The first issue is the open-sourcing of LLMs. Due to concerns such as data sources and privacy (Pan et al., 2020), many LLMs are not available to the public or can only be accessed via limited or commercial

APIs, e.g., ChatGPT (Ouyang et al., 2022) and PaLM (Chowdhery et al., 2022), while the open-source alternatives have relative lower performance compared to their closed-source counterparts (Hendrycks et al., 2020; Li et al., 2023a). Another issue is that, following the success of decoder-only models like GPT-3 and ChatGPT, the current focus of research mainly revolves around decoder-only architecture, while the investigation on large-scale encoder-decoder models, such as T5 (Raffel et al., 2020) and AlexaTM (Soltan et al., 2022), presents a relatively "under-explored area" in this field. Additionally, there is no clear consensus on whether encoder-decoder or decoder-only models hold an advantage over the others in terms of architectural superiority (Tay et al., 2022; Fu et al., 2023). In an effort to contribute to the open-source community and supplement the existing encoder-decoder models, we developed OpenBA, featuring an asymmetric encoder-decoder architecture. We collect and filter the pre-training data from open-accessible corpora. Additionally, we manually construct the Chinese Flan-like data from various publicly available annotated datasets and combine them with the English Flan corpus to obtain the Bilingual Flan collection. We employ a stage-wise training strategy to optimize the model performance by utilizing these datasets. Our model achieves outstanding performance on multiple widely-used benchmarks, such as SuperGLUE (Wang et al., 2019) and C-Eval (Huang et al., 2023).

2.2 Instruction Tuning

Instruction tuning, which involves the method of fine-tuning LLMs on an instruction dataset in a supervised manner, has played a crucial role in the significant advancements of LLMs in recent years (Zhang et al., 2023a). Starting from the T5 model (Raffel et al., 2020), which pioneers the concept of consolidating diverse NLP tasks as generative tasks. By employing task-specific prompts to guide the model, this method streamlines the process of applying LLMs to an extensive array of applications, laying the foundation for subsequent instruction tuning models such as FLAN (Wei et al., 2021; Chung et al., 2022) and T0 (Sanh et al., 2021), which further improve performance across diverse tasks by incorporating more task-specific instructions during the pre-training phase. An approach related to instruction tuning is chain-of-thought (CoT) prompting (Nye et al., 2021; Wei et al., 2022), which enhances instructions with descriptions of intermediate reasoning steps, thereby boosting LLM performance (Wang et al., 2022; Zelikman et al., 2022; Wu et al., 2023b; Xu et al., 2023). At present, the open-source community offers a multitude of instruction datasets, such as Alpaca (Taori et al., 2023) and Dolly (Conover et al., 2023). These instructions aim to enhance specific professional abilities of LLMs, such as code generation ability (Chaudhary, 2023), or the general capabilities like commonsense reasoning skills (Zhang et al., 2023b). However, the wide variety and inconsistent quality of these datasets pose challenges, with each dataset typically comprising a relatively small amount of data and focusing on a single language. In this work, we construct the BiFlan dataset, the first Bilingual Flan dataset built upon the cleansed Flan data (Longpre et al., 2023), containing various instruction types and tasks in English and Chinese language. Experimental results indicate that training on the BiFlan dataset can significantly enhance model performance on various strong benchmarks, such as MMLU (Hendrycks et al., 2020) and CMMLU (Li et al., 2023a).

3 Methodology

This section presents the details of our OpenBA model, including our considerations and implementations in pre-training data collection and processing, Bilingual Flan data construction, model architecture, training objectives and pipelines, as well as the model implementation and techniques.

3.1 Dataset Collection

This part elaborates on the data collection and filtering process for each training stage: pre-training data for stage I and II (Sec. 3.1.1), and Bilingual Flan (BiFlan) data for stage III (Sec. 3.1.2).

3.1.1 Pre-training Data Collection and Filtration

Data Sources Considering that our primary target is to construct an open-source model, we collect our pre-training data from publicly accessible resources consisting of English, Chinese, and code data. Concretely, the English and code data are sampled from the Pile corpus (Gao et al., 2020)[4],

---

[4] https://pile.eleuther.ai/

which is a collection of 22 diverse high-quality subsets. The Chinese data is mainly collected from the Internet (i.e., a cleaned subset from Common Crawl[5]), Chinese books (i.e., C-Book[6]), News (i.e., news2016zh[7]), Encyclopedias (i.e., baidu_baike[8] and wiki2019zh_corpus[7]), Comments (i.e., comments2019zh_corpus[7]) and Laws (i.e., CAIL2018[9]).

Filtration Before mixing these different data components with a certain proportion, we filter them with both personal privacy protection and quality check strategies[10], following (Sun et al., 2021). Our designed filtration strategies includes:

- **Privacy Filtering**: We removed all phone numbers, email addresses, and web links from the corpus to prevent potential privacy breaches.
- **Deduplication**: Basically, we collect our data from different open-sourced datasets with disparate sources. Thus, we mainly conduct deduplication at document, character, and paragraph levels orderly. We treat each sample as a document and use a hash algorithm to remove redundant documents, i.e., retaining only unique documents. Similarly, we also leverage a hash algorithm with an extra sentence segmenter at the paragraph level to identify and remove repeated sentences or paragraphs (we treat consecutive 1-99 sentences as a paragraph). At the character level, we delete redundant characters and reduce the sequences of repeated characters to a single instance.
- **Language Filtering**: We use polyglot[11] to detect the language of the text and only keep the texts with high confidence in either Chinese or English. We find it useful to filter out gibberish, especially for texts extracted from PDFs via OCR algorithms.
- **Internet Data Cleaning**: Data collected from the Internet often suffers from incompletions, unrecognizable characters, and web page tags. Thus, we filter out sentences with less than 10 words and remove unusual characters and HTML tags.

Mixing and Statistics Following (Zeng et al., 2022), our pre-training data consists of the same proportion of Chinese and English tokens, in which we sample 190B English tokens[12] from the Pile corpus and 190B tokens from our filtrated Chinese data. We also sample 20B code tokens from the Pile corpus to make the overall percentages (5 %) of code tokens resemble LLaMA (Touvron et al., 2023a) (4.5 % code tokens). In total, our pre-training dataset is a mix of 190B English tokens, 190B Chinese tokens, and 20B code tokens. Following our strategies, one can construct a pre-training dataset with trillion tokens. Nevertheless, we only collect 400B tokens for pre-training due to our limited computation resources. Fig. 1(a) shows the detailed composition of the pre-training dataset.

3.1.2 Bilingual Flan Data Collection

English Flan Data Collection The Flan Collection (Chung et al., 2022; Longpre et al., 2023) serves as a foundational dataset for effective instruction tuning, encompassing more than 1800 tasks. We follow the official guidelines to collect and process the English Flan collection with two steps, i.e., downloading five sub-mixtures from the Flan Collection and then combing them according to the specified mixture rates[13].

Chinese Flan Data Collection Many open-source Chinese instruction datasets are derived from English translations or generated by ChatGPT using various prompts (Ouyang et al., 2022; Bubeck et al., 2023), which can lead to translation inaccuracies and incorrect responses. Thus, the quality and

---

[5]https://commoncrawl.org/

[6]https://github.com/FudanNLPLAB/CBook-150K

[7]https://github.com/CLUEbenchmark/CLUE

[8]https://github.com/BIT-ENGD/baidu_baike

[9]https://github.com/thunlp/CAIL

[10]Since the Pile is a cleaned high-quality corpus, we directly sample English and code data from the Pile corpus without further filtration. Our filtration strategies are applied to the Chinese data.

[11]https://github.com/aboSamoor/polyglot

[12]These English tokens exclude code data but inevitably contain a small percentage of non-language tokens (e.g., 1.24 % of math data) since we randomly select samples based on the original proportion of the Pile corpus except for code data.

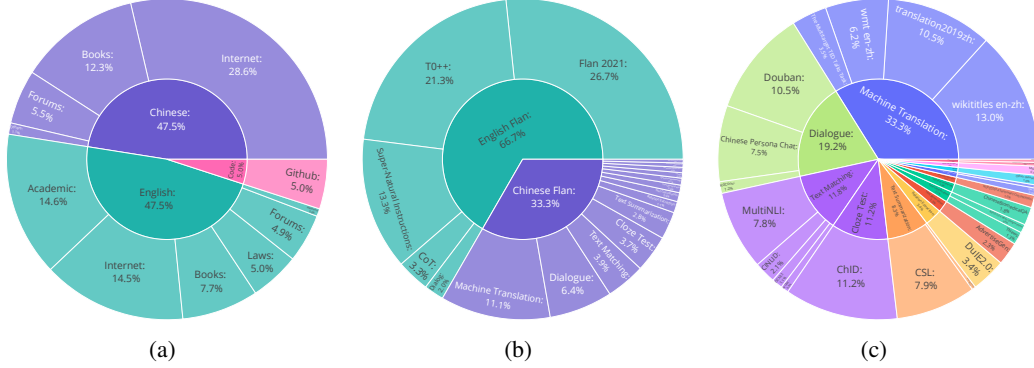[13]https://github.com/google-research/FLAN/tree/main/flan/v2

Figure 1: The composition of Data collection. Figure (a) represents the composition ratio of the pre-training dataset. Figure (b) represents the composition of the Bilingual Flan dataset. Figure (c) represents the finer-grained composition of the Chinese Flan dataset.

quantity of existing Chinese instruction corpora are often inadequate. To tackle these challenges, we build our own Chinese instruction corpus. More concretely, we collect 44 different Chinese tasks with a total of 50 million data entries, in which the data sources include various competitions, academic papers, and open-source projects. The distribution of the Chinese Flan dataset is shown in Fig. 1(c). , and we list all the data sources in Tab. 14 (Appendix B). For each task, we manually design the Chinese instructions.

Bilingual Data Combination Due to the smaller number of Chinese data compared to English data, we perform sampling within the English Flan datasets to ensure a balanced proportion between Chinese and English data. As illustrated in Fig. 1(b), the Bilingual Flan (BiFlan) dataset consists of 66.7% English Flan data and 33.3% Chinese Flan data. We also filter out samples whose lengths exceed the encoder's maximum length, ensuring the critical parts of instructions are not truncated.

3.2 Model Architecture

Generally, the OpenBA model follows the standard encoder-decoder architecture like T5 (Raffel et al., 2020). However, it is worth noting that the encoder and decoder serve different roles, where the encoder provides strong comprehension capability, while the decoder offers generative ability (Vaswani et al., 2017), and existing works indicate that an encoder-decoder model with more encoder layers can achieve powerful performance (Soltan et al., 2022). To enhance generative capability and fill the gap of deeper decoder-based LLM, we also design another asymmetric structure, where the detailed model settings are given in Tab. 1.

Encoder	Decoder	Attn Heads	d_{model}	d_{ff}	#Param.(B)
12	36	40	4,096	16,384	14.6*

Table 1: Model settings for OpenBA, where #Param. denotes the number of model parameters. We share the embedding weights between the encoder and decoder, which are not repeatedly counted.

Apart from leveraging the asymmetric shallow-encoder deep-decoder, we also incorporate the following improvement strategies into the model architecture:

- **Sandwich Layer Normalization** To stabilize the training process, we adopt the sandwich layer normalization (Ding et al., 2021) by normalizing both the input of each transformer block and the output of each attention layer. We use the RMSNorm (Zhang & Sennrich, 2019) for normalization.

- **Rotary Embedding** We apply the rotary embedding (Su et al., 2021) that encodes the absolute position with a rotation matrix and meanwhile incorporates the explicit relative position dependency in self-attention instead of the relative positional embedding in T5.
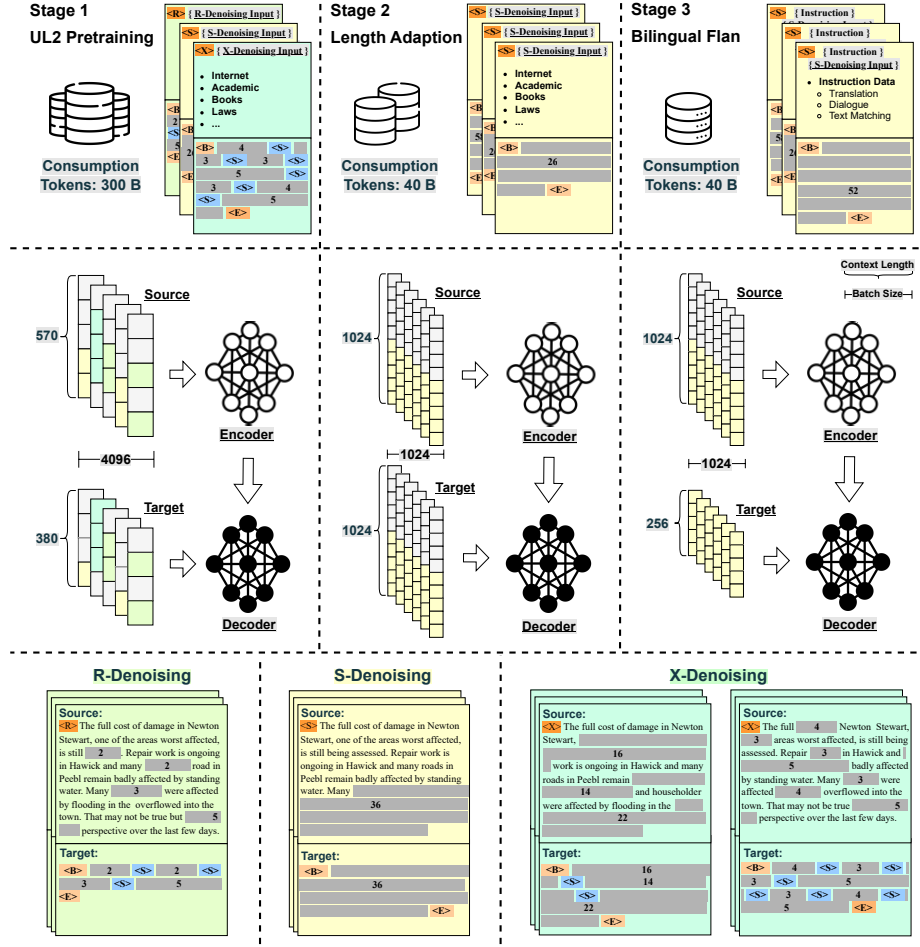
Figure 2: Overview of training process.

- **SwiGLU Activation Function** Inspired by LLaMA (Touvron et al., 2023a), we replace the original ReLU activation with the SwiGLU activation function (Shazeer, 2020), and set the dimension as $\frac{2}{3}4d$.

- **mT5 Tokenizer** For the bilingual setting, we use mT5 (Xue et al., 2020) tokenizer as it not only covers Chinese and English but also provides possibilities for further training and applications in other languages.

3.3 Training Process and Language Modeling Tasks

As shown in Fig. 2, we adopt the stage-wise training strategy (Barshan & Fieguth, 2015) that consists of UL2 (Tay et al., 2022) pre-training, length-adaptation, and Flan training stages (Wei et al., 2021), and set different context length and batch size for different stages (Tab. 2). For all the stages, we adopt the span-denoising language modeling task as proposed in T5 (Raffel et al., 2020) and BART (Lewis et al., 2019). More concretely, given a sequence $\boldsymbol{x} = \{x_i\}_{i=1}^{n}$ containing n tokens, we corrupt it with mask sentinel $m_j = \{x_i\}_{i=p}^{k}$, where $p < k, 1 \le p, k \le n$. Then, the training objective is to recover the masked span, which can be formally written as:

$$\mathcal{L}(\boldsymbol{x}) = \log P(\boldsymbol{m} \mid \boldsymbol{x}_{\setminus \boldsymbol{m}}, \theta) \tag{1}$$

where $\boldsymbol{m} = \{m_j\}_{j=1}^{K}$ is the set of masked spans, $\boldsymbol{x}_{\setminus \boldsymbol{m}}$ denotes the corrupted sequence, and θ represents the model parameters. For OpenBA model, $\boldsymbol{x}_{\setminus \boldsymbol{m}}$ is fed to the encoder, which can retain

Stage	Strategy	Encoder Context	Decoder Context	Batch Size	#Tokens (B)
I	UL2 Pre-training	570	380	4,096	300
II	Length-Adaptation	1,024	1,024	1,024	40
III	Bilingual Flan Training	1,024	256	1,024	40

Table 2: Configurations for each training stage, where #Tokens represents the number of consumption tokens in each stage.

Type	Span Length (μ)	Corruption Ratio (%)	#Num	Sentinel
R-Denoising	{3, 8}	15.0	K	<R>
S-Denoising	-	25.0	1	<S>
X-Denoising	{3, 8, 64} / {64}	50.0 / 15.0	K	<X>

Table 3: Settings of three denoising strategies for the UL2 pre-training stage, where μ is the mean of the normal distribution, #Num represents the number of masked spans, and K is determined by the sequence length, span length, and corruption ratio.

a bidirectional receptive field, and m is predicted by the decoder. Next, we will introduce the aforementioned different stages more concretely.

Stage I: UL2 Pre-training   Starting with the UL2 training strategy, we adopt a mixture of denoisers training strategy proposed by Tay et al. (2022), exposing OpenBA to a diverse set of problems during the first pre-training stage. In this stage, the OpenBA model is fed with 300B tokens sampled from the pre-training corpus, and we employ three different training objectives which are listed below:

- **R-Denoising** Regular denoising is the standard span corruption that sets a range of 2 to 5 tokens as the masked span length and masks ratio about 15% of the input tokens. This denoising task is relatively simple since the span is short and efficient for the model to acquire knowledge embedded in the text.

- **S-Denoising** Sequence denoising aims to endow the model with generation capability, where the input text is split into two sub-sequences, and the model should predict the latter sequence conditioned on the first sequence. In the S-Denoising setting, the model can acquire the generation ability.

- **X-Denoising** To bridge the gap between the R-Denoising and S-Denoising, X-Denoising can be viewed as an extreme version of denoising, where approximately 50% of the input sequence is masked by increasing either the masked span length or the corruption rate. Such a denoising strategy simulates the situation where a model needs to generate long targets from a memory with relatively limited information.

We list the settings of these three denoising strategies in Tab. 3. It is worth noting that we conduct these denoising strategies from the instance level and prepend three special tokens before each corrupted sequence to prompt the current denoising task for OpenBA (Tay et al., 2022). We uniformly sample a value based on μ as the masked span length for the R-denoising and X-denoising. For S-denoising, we limit each masked span to end at the end of the input text and allow only one masked span. Besides, we set encoder-decoder context length as 570/380 in this stage for sampling efficiency.

Stage II: Length-Adaptation   Considering the context length for the first pre-training stage is short, which may not support the long input and output formats of some tasks, such as in-context learning (Min et al., 2021) and long text generation (Guan et al., 2021), we extend the encoder-decoder context length to 1,024/1,024 during the length-adaptation stage. During this stage, we utilize 40B tokens sampled from the pre-training corpus and ensure that there is no overlap between these data and the data from the previous stage. Additionally, we simply apply the S-Denoising training objective and adjust the corruption ratio to 50%. We keep the special sentinel <S> before each corrupted text and decrease the batch size for training stability in this stage.
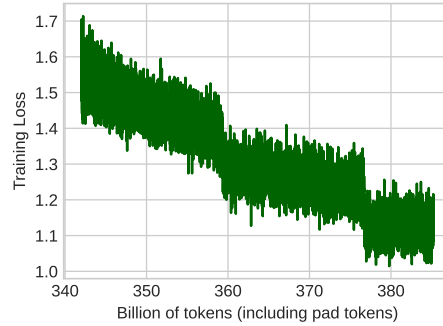
Stage III: Bilingual Flan Training   Inspired by the previous work (Chung et al., 2022), we apply Flan instruction training on the length-adapted OpenBA checkpoint. We still prepend the special
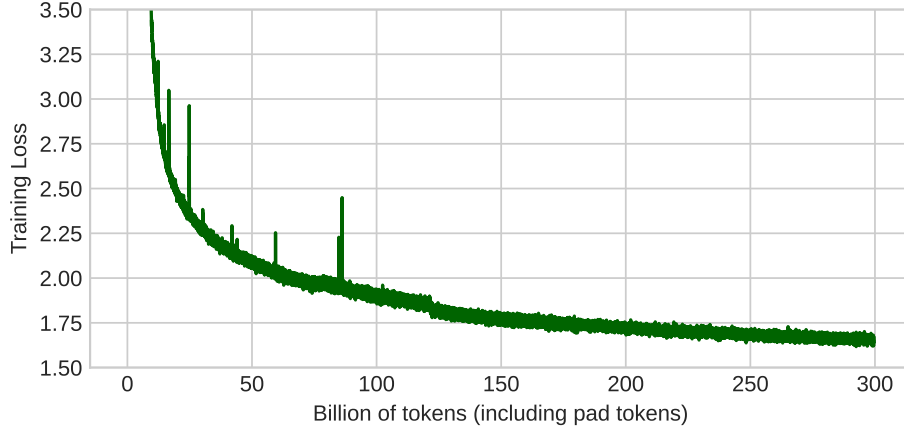
(a) Training loss of Stage I
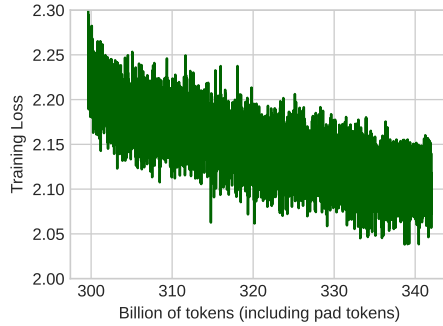
(b) Training loss of Stage II

(c) Training loss of Stage III

Figure 3: Loss curves for each training stage.

token `<S>` before each text for the generation task and apply the constructed BiFlan dataset in this stage. In addition, we set the encoder-decoder sequence length as 1,024/256 in this stage for sampling efficiency since we observe that most outputs of Flan datasets are short, i.e., less than 256 tokens.

3.4 Model Implementation and Techniques

We train OpenBA on a cluster with 4 nodes (8 × NVIDIA A100-SXM4-80GB GPUs), which are linked with the InfiniBand network (Grun, 2010) and interconnected through the NVLink system. The model has consumed nearly 400B bilingual tokens and achieved 1.2×10^{22} FLOPs (floating point of operations) in total. We implement our model based on the NVIDIA-Megatron framework[14] and make several optimizations for training stabilization and inference efficiency. We plot the training loss for the aforementioned three stages in Fig. 5, and list the techniques we have used below:

• **3D Parallelism** 3D parallelism (Shoeybi et al., 2019) aims to scale and accelerate the training process of LLMs, which harnesses three core parallelism techniques, i.e., data parallelism, model parallelism (mp), and pipeline parallelism (pp). Considering the model size, the number of GPUs and the communication speed among GPUs, we settle on an optimal setting of *mp_size*=4 and *pp_size*=1, reaching 120 TFLOP/s per GPU.

• **Checkpoint Activation** Checkpoint activation is a technique designed to optimize memory usage during training. Instead of storing all intermediate layer activations, only certain ones are

---

[14]`https://github.com/NVIDIA/Megatron-LM/`

Model	#Param.	Tokens	GPU/TPU type	GPU hours	Total Power Consumption	Carbon emitted (tCO$_2$eq)
OPT Zhang et al. (2022)	175B	180B	A100-80GB	809,472	356 MWh	137
BLOOM Scao et al. (2022)	176B	366B	A100-80GB	1,082,880	475 MWh	183
GLM Zeng et al. (2022)	130B	400B	A100-40GB	1,105,920	442 MWh	257
ChatGLM Zeng et al. (2022)	6B	1.0T	-	-	-	-
Falcon Penedo et al. (2023b)	40B	1.0T	A100-40GB	-	-	-
Flan-T5-XL Chung et al. (2022)	3B	>1.0T	TPU-v3/v4	-	-	-
LLaMA Touvron et al. (2023a)	7B	1.0T	A100-80GB	82,432	36 MWh	14
LLaMA Touvron et al. (2023a)	13B	1.0T	A100-80GB	135,168	59 MWh	23
LLaMA Touvron et al. (2023a)	65B	1.4T	A100-80GB	1,022,362	449 MWh	173
LLaMA-2-Chat Touvron et al. (2023b)	70B	>2.0T	A100-80GB	1,720,320	-	291
Baichuan Inc. (2023)	7B	1.2T	A800	-	-	-
BatGPT Li et al. (2023b)	15B	1.0T	-	-	-	-
MOSS Sun et al. (2023)	16B	>700B	-	-	-	-
OpenBA	15B	380B	A100-80GB	38,214	17 MWh	6.5

Table 4: The number of parameters, consumed tokens, and training cost for the LLMs mentioned in the paper, where #Param. denotes the model parameters. We report the carbon emission according to the official statement, and calculate the carbon emission of OpenBA according to Wu et al. (2022).

preserved. During back-propagation, the missing activations are recalculated, trading off additional computational efforts for memory savings. This strategy allows for the training of larger models even on GPUs with limited memory capacity. In fact, training a 15B model on 80GB GPUs becomes manageable in terms of memory. We specifically apply the checkpoint activation to the attention computation, which is relatively cost-effective to recompute. In practical deployment, we observe a significant improvement in GPU memory utilization, enhancing the overall system performance.

• **Distributed Optimizer**   The distributed optimization approach offers an alternative for saving GPU memory, enabling the utilization of an increased batch size, albeit at the expense of communication burden among GPUs. By adopting the ZeRO method proposed by Rajbhandari et al. (2020) and implementing the distributed optimization technique (Shoeybi et al., 2019), we can increase the batch size, thereby enhancing the training speed.

• **Attention Weights Computation in FP32 Precision**   During the softmax computation, particularly when handling large values, there exists a possibility of numerical overflow. Conducting this computation with FP32 precision mitigates this risk compared to using FP16 precision. The previous works (Nijkamp et al., 2022) indicate that such an issue can easily take place when computing attention weights in FP16 precision. In the early training stage of OpenBA, we adopt half-precision calculation for all the model modules and often observe the phenomenon of loss collapsing. However, such an issue has been greatly alleviated when converting the attention weight calculations to full precision (FP32). Thus, we can empirically conclude that attention weight computation in FP32 precision can significantly enhance the stability of the training process.

• **Inference Efficiency**   To accelerate the inference speed, we adopt the KV-cache technique and decrease the computation by pre-computing the rotary embeddings for all the positions.

4   Results

4.1   Evaluation Settings

We evaluate OpenBA from three aspects: natural language understanding, natural language generation, and commonsense reasoning. Specifically, we evaluate the natural language understanding capability on the SuperGLUE (Wang et al., 2019) and BELEBELE (Bandarkar et al., 2023) benchmark, natural language generation ability with five downstream tasks (summarization, machine translation, text simplification, paraphrase, and story generation), and commonsense reasoning ability on five authoritative benchmarks, including MMLU (Hendrycks et al., 2020), CMMLU (Li et al., 2023a), BBH (Suzgun et al., 2022), and C-Eval (Huang et al., 2023). Following the previous works (Brown et al., 2020; Touvron et al., 2023a), we consider both the zero-shot and few-shot settings and strictly distinguish the domain distribution of training and testing data. The illustration and the corresponding implementation of each setting are as follows:

Model Metrics	#Param.	Avg.	BoolQ Acc.	CB Acc.	RTE Acc.	ReCoRD F1	ReCoRD EM	WSC Acc.
BERT-Large	340M	69.0	77.4	83.6	71.6	<u>72.0</u>	71.3	64.3
BERT-Large++	340M	71.5	79.0	<u>90.4</u>	79.0	<u>72.0</u>	<u>73.0</u>	64.3
Flan-T5-XL	3B	**79.3**	**89.3**	**91.2**	**90.4**	57.2	56.6	**84.9**
GPT3	175B	71.8	76.4	75.6	69.0	**91.1**	**90.0**	<u>80.1</u>
OpenBA	15B	<u>73.1</u>	<u>82.6</u>	85.6	<u>83.9</u>	69.4	68.8	76.0

Model Metrics	#Param.	WiC Acc.	CoPA Acc.	MultiRC F1	MultiRC EM	AX_b MCC	AX_g GPS	AX_g Acc
BERT-Large	340M	**69.5**	70.6	70.0	24.0	23.0	<u>97.8</u>	51.7
BERT-Large++	340M	**69.5**	73.8	70.4	24.5	38.0	**99.4**	51.4
Flan-T5-XL	3B	65.7	**97.6**	**87.0**	**57.9**	**50.1**	97.2	**91.9**
GPT3	175B	49.4	<u>92.0</u>	75.4	30.5	21.1	90.4	55.3
OpenBA	15B	57.2	85.8	<u>77.1</u>	<u>38.9</u>	<u>40.8</u>	94.4	<u>70.2</u>

Table 5: Zero-shot results on SuperGLUE benchmark, where #Param. denotes the model parameters, and Avg. denotes average accuracy.

- **Zero-Shot** We provide a textual description of the task for each testing sample, and the model will respond in an open-ended manner. Templates for all tasks are listed in Appendix A.

- **Few-Shot** We evaluate each example in the testing set by randomly selecting ℓ examples from the training set of each task as conditioning. In this paper, we set $\ell = 5$ as default if not specified.

- **Domain Held-in / Held-out** We differentiate between the held-in and held-out settings based on whether the training data includes the domain of the testing set. If the model has been trained on the training data corresponding to the testing task, it is viewed as held-in; otherwise, it is held-out (Longpre et al., 2023).

We also apply the CoT technique for some tasks, and the corresponding templates are also shown in Appendix A. It is worth noting that we will specifically elaborate on the basic settings for each evaluation task and compare them to the models under the same settings. Additionally, we will evaluate the results using the officially recommended evaluation metrics and platforms whenever possible and utilize the **bold font** to indicate **the best performance** and adopt <u>underline</u> to denote <u>the second-best</u> performance in all the experiments.

4.2 Training Cost Analysis

All the models we compare are listed in Tab. 4, where we report their parameters, consumption tokens, training cost, and the corresponding carbon emissions, respectively. To calculate carbon emissions, we follow Wu et al. (2022) and Touvron et al. (2023a) by taking a PUE of 1.1 and a carbon intensity factor set at the national US average of 0.385 kg CO2e per KWh, and the formula is:

$$tCO_{2eq} = MWh \times 0.385 \qquad (2)$$

It is worth noting that, ***the training process of OpenBA is highly efficient and environmentally friendly.*** Taking ***LLaMA-13B*** as an example, it consumes around 1TB tokens with a total 59MWh GPU power and emits around ***23 tCO_{2eq} carbon***. However, our model has consumed only 6.5 tCO_{2eq} carbon for 380B tokens, i.e., around ***28.26 % of the total carbon emission of the LLaMA-13B model***. More training details and model implementation can be found in Sec. 3.4.

4.3 Natural Language Understanding

We evaluate the natural language understanding performance of OpenBA model on the SuperGLUE benchmark, which contains 13 sub-tasks. Since the BiFlan dataset contains partial training data of some testing tasks in SuperGLUE, we mainly compare OpenBA with models in the held-in setting (except GPT-3 (Brown et al., 2020)), i.e., these models have also been trained on the training data of some testing tasks in SuperGLUE. As we can observe in Tab. 5, the performance of OpenBA

Model	#Param.	eng_Latn	zho_Hans	zho_Hant	Avg.
Falcon[†]	40B	77.2	66.0	62.2	68.5
LLaMA[†]	70B	<u>82.5</u>	64.6	57.7	68.2
InfoXLM[‡]	550M	79.3	74.6	72.4	75.4
XLM-V[‡]	1.2B	76.2	71.0	67.1	71.4
LLaMA-2-Chat[*]	70B	78.8	62.4	59.3	66.8
GPT3.5-Turbo[*]	-	**87.7**	**77.6**	**76.3**	**80.5**
OpenBA[*]	15B	78.6	<u>75.2</u>	<u>73.7</u>	<u>75.8</u>

Table 6: Model performance on BELEBELE benchmark, where † denotes 5-shot setting, ‡ denotes full fine-tuning in English and * denotes the zero-shot setting for instructed models. We report the accuracy score for all the models.

Model	#Param.	Rouge-1	Rouge-2	Rouge-L
ChatGLM	6B	<u>27.3</u>	**17.2**	<u>26.7</u>
Baichuan	7B	19.9	<u>14.4</u>	20.0
BatGPT	15B	25.6	12.2	25.0
OpenBA	15B	**30.2**	13.9	**28.6**

Table 7: Model performance on CLTS subset containing 100 sentences sampled from CLTS test set. We report Rouge-1, Rouge-2 and Rouge-L score.

Model	#Param.	Zh⇒En	En⇒Zh
ChatGLM	6B	17.2	32.5
Alpaca	7B	15.1	9.8
PARROT	7B	19.6	24.8
BatGPT	15B	<u>23.1</u>	**38.7**
MOSS	16B	17.2	32.5
OpenBA	15B	**23.3**	<u>37.4</u>

Table 8: Model performance on Flores subset containing 50 sentences sampled from Flores.

surpasses that of the BERT model (Devlin et al., 2018) fine-tuned on the SuperGLUE training set and GPT-3, but is slightly behind that of the Flan-T5-XL (Chung et al., 2022) model.

We evaluate the reading comprehension ability of OpenBA with BELEBELE benchmark (Bandarkar et al., 2023) and select the Chinese (Simplified), Chinese (Traditional), and English subsets for evaluation. We follow the official settings and compare with both LLMs and fine-tuned down-stream models, including Falcon (Penedo et al., 2023a), LLaMA (Touvron et al., 2023a,b), XLM-V (Liang et al., 2023a), InfoXLM (Chi et al., 2020) and ChatGPT (Ouyang et al., 2022). We provide all the instructions we use for zero-shot setting in Appendix A. As we can observe from Tab. 6, OpenBA can achieve outstanding results in the Chinese reading comprehension tasks, ranking just behind ChatGPT. For English reading comprehension tasks, the performance of OpenBA is comparable to that of the Falcon-40B model, which is trained with around 1TB tokens of multilingual data. It is also worth noting that OpenBA achieves better performance among multiple current open-source LLMs, including two strong LLaMA models and the Falcon-40B model, under the bilingual setting.

## 4.4 Natural Language Generation

We evaluate the natural language generation ability of our model on five tasks, including machine translation on the Flores (Goyal et al., 2022) benchmark, text summarization on the CLTS benchmark (Liu et al., 2020), paraphrase task on the QQP dataset[15], text simplification on the WIKI-AUTO (Coster & Kauchak, 2011) dataset, and story generation on the ROC (Mostafazadeh et al., 2016) dataset.

Summarization To evaluate the summarization task under the held-out setting, we select a subset containing 100 sentences sampled from CLTS benchmark (Liu et al., 2020), which is excluded from the BiFlan dataset. Specifically, we prepend the task instruction before each test sentence (the task instruction is listed in Appendix A) and allow models to conduct zero-shot inference. We evaluate the generated results with Rouge-n metric (Lin, 2004) and report the results in Tab. 7. We observe that OpenBA can achieve the best performance on the Rouge-1 and Rouge-L scores, indicating that the content generated from OpenBA is faithful to the original text in the summarization task.
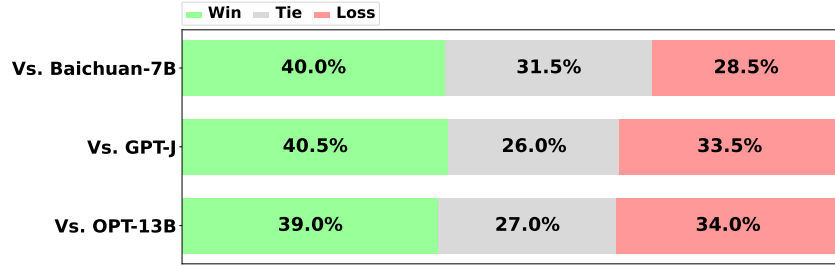
---

[15] https://www.kaggle.com/c/quora-question-pairs

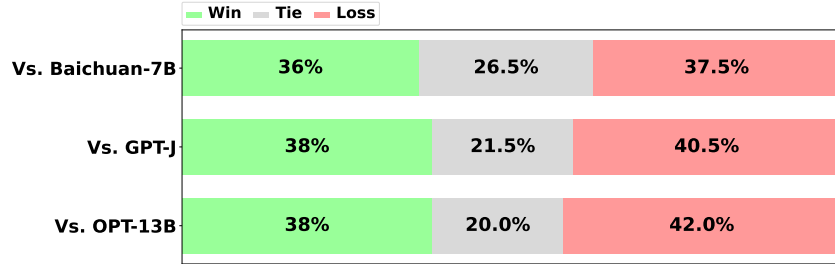| Model	#Param.	WIKI AUTO					QQP				
		B-2 (↑)	D-2 (↑)	LR-2 (↓)	Mav (↑)	SIM (↑)	B-2 (↑)	D-2 (↑)	LR-2 (↓)	Mav (↑)	SIM (↑)
BatGPT	15B	25.5	1.5	89.9	96.5	5.5	19.4	1.6	67.6	58.3	7.6
chatGLM	6B	**29.2**	1.4	90.7	97.7	4.0	**25.0**	2.0	63.9	93.6	5.6
MOSS	16B	27.8	1.5	82.9	96.8	5.4	19.3	1.4	72.7	37.2	7.8
OpenBA	15B	27.9	**1.9**	**75.6**	**99.1**	**6.6**	22.7	**2.0**	**48.0**	**94.4**	**7.9**

Table 9: Model performance on WIKU AUTO and QQP datasets.
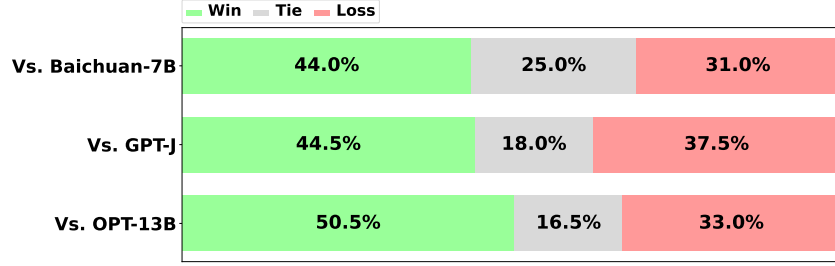
(a) Coherence

(b) Consistency

(c) Correctness

Figure 4: Human evaluation results on the ROC dataset.

Machine Translation   We compare the model performance on the bilingual machine translation tasks, including Chinese-to-English and English-to-Chinese translation, on the Flores (Goyal et al., 2022) machine translation benchmark. We strictly follow the official settings by selecting 50 testing samples provided for each translation task. It is worth noting that all the models are under the held-out zero-shot setting. We report the BLUE-4 (Post, 2018) scores in Tab. 8 and can observe that OpenBA can achieve the best performance on the Chinese-to-English translation task and obtain comparable results with the SOTA achieved by BatGPT on the English-to-Chinese translation task.

Text Simplification and Paraphrase   We evaluate the text simplification and paraphrase ability of OpenBA on the WIKI AUTO and QQP datasets. We evaluate the model performance with BLUE, Distinct-n (D-n) metrics (Li et al., 2015), Lexical Repetition (Rep-n, 4-gram repetition for n-times) (Shao et al., 2019b), Mauve (Pillutla et al., 2021) and Semantic Similarity (SIM, semantic

Model	#Param.	Humanities	STEM	Social Sciences	Other	Average
LLaMA[†]	7B	34.0	30.5	38.3	38.1	35.1
LLaMA[†]	13B	**45.0**	<u>35.8</u>	**53.8**	**53.3**	**46.9**
BLOOM[†]	176B	34.1	**36.8**	41.5	46.5	39.1
ChatGLM[†]	6B	35.4	31.3	41.0	40.5	36.9
Baichuan[†]	7B	38.4	35.6	<u>48.9</u>	<u>48.1</u>	<u>42.3</u>
BatGPT[†]	15B	35.4	33.5	36.3	37.0	36.7
MOSS[†]	16B	30.5	29.3	33.8	34.4	31.9
OpenBA[†]	15B	34.6	29.8	40.1	40.0	36.0
OpenBA[‡]	15B	<u>38.7</u>	33.8	45.0	43.6	40.2
OpenBA[*]	15B	36.7	31.4	42.8	42.3	38.2

Table 10: Model performance on MMLU benchmark, where #Param. denotes the model parameters, † denotes 5-shot, ‡ denotes 0-shot, and ∗ represents the chain-of-thought.

Model	#Param.	Humanities	STEM	Social Science	Other	China-specific	Average
Falcon	40B	<u>43.5</u> / 41.3	33.3 / 31.1	44.3 / 40.9	<u>44.8</u> / 40.6	<u>39.5</u> / 36.1	<u>41.5</u> / 38.5
LLaMA	65B	40.2 / 34.5	34.5 / 31.1	41.6 / 36.1	42.9 / 37.9	37.0 / 32.9	39.8 / 34.9
ChatGLM	6B	39.2 / 42.9	32.4 / 32.2	39.7 / 44.8	38.6 / 42.6	37.7 / 41.9	37.5 / 40.8
Baichuan	7B	**48.1** / 44.4	**35.3** / 32.8	**47.9** / 46.8	**46.6** / 44.8	**44.1** / 43.1	**44.4** / 42.3
BatGPT	15B	35.5 / 36.5	<u>35.0</u> / 33.7	36.3 / 38.1	42.1 / 46.9	37.9 / 38.3	37.2 / 38.5
OpenBA	15B	40.9 / 40.9	33.5 / 33.8	<u>45.2</u> / 44.7	44.5 / 43.6	39.1 / 38.6	<u>41.5</u> / 41.2
OpenBA[*]	15B	30.0	37.6	40.6	39.2	36.4	37.0

Table 11: Performance on CMMLU benchmark, where #Param. denotes the model parameters, and ∗ denotes chain-of-thought. We report the 5-shot and 0-shot performance with diagonal bar division.

similarity between generations and corresponding prompts) (Guan et al., 2021) metrics, and report the model performance in Tab. 9. Based on the observation that OpenBA can attain the best results on the Mav and SIM metrics, which evaluate semantic relevance with gold text and input text respectively, we can conclude that our model excels at capturing the overall semantic information of the input content and generating relevant content accordingly.

Story Generation We evaluate the open-domain generation capability of OpenBA on the ROC dataset, where the model should continue generating based on the existing context and the story plot. More concretely, we feed the model with the prompt directly and compare OpenBA with two other models: GPT-J (Wang & Komatsuzaki, 2021) and OPT-13B (Zhang et al., 2022), which are also trained on the Pile corpus. We randomly sample 100 generated cases and invite annotators to score the text from three aspects, including **coherence** between the generated text and the prompt, **consistency** and **correctness** of the generated text. The annotators are allowed to choose "Tie" if it is hard to distinguish two generation cases. As shown in Fig. 4, we can observe our model can obtain strong performance on the coherence and consistency aspect and attain comparable performance with two other models on the correctness aspect.

Model	#Param.	BBH
LLaMA	13B	**37.1**
ChatGLM	6B	31.3
Baichuan	7B	31.9
BatGPT	15B	<u>34.1</u>
MOSS	16B	29.3
OpenBA	15B	<u>34.1</u>

Table 13: Model performance on the BBH benchmark. We report the accuracy score for all the models.

4.5 Common Sense Reasoning

We evaluate the common sense reasoning ability of OpenBA on four benchmarks, including MMLU, CMMLU, BBH, and C-Eval. To ensure a fair comparison, we conduct all the evaluations under the held-out setting, follow the recommended setting of each benchmark, and compare with other strong LLMs under the same settings. For the MMLU (Tab. 10) and C-Eval (Tab. 12) benchmarks, we report the zero-shot, 5-shot, and 5-shot CoT results. For CMMLU (Tab. 11), we report the zero-shot, 5-shot,

Model	#Param.	STEM	Social Science	Humanities	Others	Avg.	Avg.(Hard)
LLaMA	65B	<u>37.8</u>	45.6	36.1	37.1	38.8	<u>31.7</u>
ChatGLM	6B	33.3	48.3	41.3	38.0	38.9	29.2
Baichuan	7B	**38.2**	<u>52.0</u>	<u>46.2</u>	39.3	<u>42.8</u>	**31.5**
MOSS-moon-sft	16B	31.6	37.0	33.4	32.1	33.1	28.4
GLM-130B	130B	36.7	**55.8**	**47.7**	**43.0**	**44.0**	30.7
OpenBA	15B	34.8	46.6	41.1	<u>41.5</u>	39.8	31.1
OpenBA*	15B	30.7	43.7	40.9	35.2	36.3	27.0

Table 12: Model performance on C-Eval benchmark, where * denotes chain-of-thought and Avg. represents average accuracy. We report the 5-shot and 0-shot performance with diagonal bar division.

and zero-shot CoT results. We report the zero-shot results for BBH in Tab. 13. It is worth noting that the first block for each table is multilingual- or English-oriented models, the second block is Chinese-oriented models, and we rank the models in each block by model size. We can observe that, on all the benchmarks, OpenBA can achieve better performance than two strong Chinese-oriented models, i.e., ChatGLM (Du et al., 2022) and BatGPT[16] (Li et al., 2023b), and obtain comparable results with Baichuan-7B model (Inc., 2023), which is trained on datasets much larger than ours, i.e., 1.2TB tokens. Furthermore, our model surpasses English-oriented models on most benchmarks and even outperforms some tasks where English-oriented models have over 100 billion parameters, e.g., BLOOM-176B, on the MMLU benchmark. Additionally, OpenBA can achieve comparable scores under both zero-shot and few-shot settings and even performs slightly better under the zero-shot setting, indicating that the OpenBA model has a strong instruction-following capability.

5 Analysis

5.1 Model Architecture Selection

Our asymmetric shallow-encoder deep-decoder model architecture stems from the following motivations and considerations:

- **Enhanced Generative Capabilities.** For the three tasks in UL2, namely R-Denoising, S-Denoising, and X-Denoising, a deeper decoder setup is particularly effective for the S-Denoising task, which reflects the model's language modeling ability.

- **Potential Acceleration in Dialogue Inference.** Decoder-only architectures similar to GPT have already achieved excellent results in multi-turn dialogue tasks. However, for encoder-decoder models, how to store dialogue history presents a significant challenge. A common approach is to embed the dialogue history into the encoder's input. However, continuously altering this history results in increased computational costs in the encoder, and it's not amenable to acceleration via KV-caching. To address this challenge, we can place the dialogue history into the decoder. This shift imposes a greater demand on the decoder's capabilities. Thus, we explore training a deeper decoder to endow it with enhanced capabilities.

We conduct experiments to explore the influence of the model architecture, where we train the model with the UL2 training objective. Specifically, we set the batch size as 128 and the sequence length as 570/380. We validate the model performance after 15k training steps.

Model Configuration We mainly explore three model structures: (1) a shallow encoder with a deep decoder, (2) a deep encoder with a shallow decoder, and (3) the encoder and decoder with equal depth. We assess their performance metrics across the R-Denoising, S-Denoising, and X-Denoising tasks to learn their respective merits. To maintain consistent parameter counts across different configurations, we adopt these layer structures: (1) EncoderLayer=18, DecoderLayer=6, (2) EncoderLayer=6, DecoderLayer=18, and (3) EncoderLayer=DecoderLayer=12.

---

[16]BatGPT achieves the best performance on the official C-Eval leaderboard, but we do not obtain the reported results using the open-source version of BatGPT.
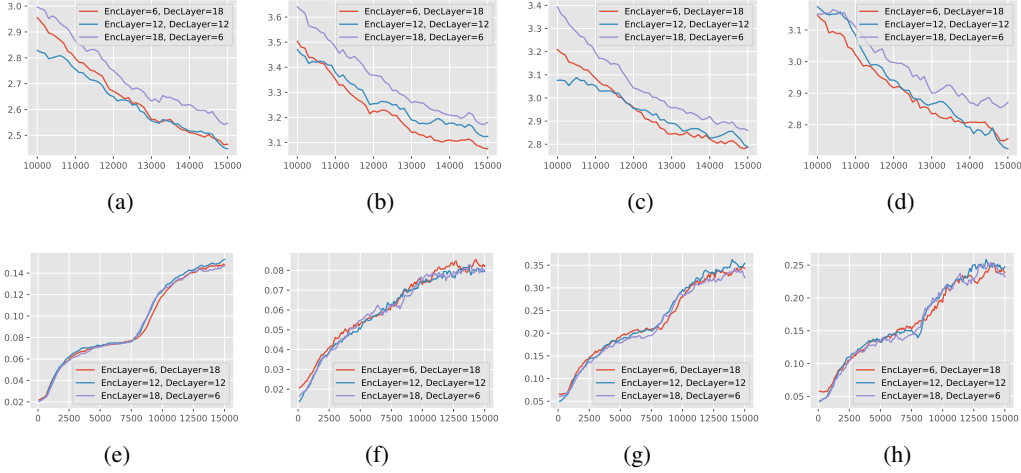
Figure 5: The performance in terms of loss and accuracy of the three model configurations across four denoising tasks. The first row of figures illustrates the loss performance, while the second row depicts the accuracy. The four columns respectively represent the four tasks: R-Denoising, S-Denoising, X-Denoising, and a combination of the three.

Evaluation Metric   To get a direct view of the model performance pre-trained from scratch, we choose Loss and Acc. as convenient metrics. Specifically, we construct validation sets for R-Denoising, S-Denoising, X-Denoising, and a combination of the three, respectively, and test the model's performance throughout the training process. Acc. indicates the model's predictive accuracy for the next word:

$$\text{Acc.} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(\text{argmax}_{w \in V} P(\boldsymbol{x}_i = w | \boldsymbol{x}_{<i}, \theta) = \boldsymbol{x}_i), \tag{3}$$

where n denotes the sequence length, V denotes the vocabulary size and $\mathbb{I}$ is an indicator function.

Analysis   Fig. 5 shows our results. We can conclude that:

- As a measurement of the model's generation ability, the S-Denoising task is generally more challenging to learn. This is evident as, regardless of the model configuration, the S-Denoising task consistently has a higher loss and a lower accuracy.

- The model with a shallow encoder and deep decoder configuration performs better on the S-denoising task (from Fig. 5(b) and 5(f)), though it doesn't outperform the balanced setup across all three tasks (from Fig. 5(b) and 5(f)).

5.2   Evolution of Performance During Training

In this section, we evaluate the performance of OpenBA at various stages of the overall training. We employ three benchmarks for evaluation, including MMLU for English common sense reasoning, CMMLU for Chinese common sense reasoning, and BELEBELE for reading comprehension. As shown in Fig. 6(a), Fig. 6(b), and Fig. 6(c), the performance on most tasks increases with the number of training steps during the UL2 Pre-training stage, experiences slight fluctuations during the Length-Adaptation stage, and exhibits a significant improvement during the Bilingual Flan Training stage. The emergence curves of Chinese and English are similar, indicating that our Bilingual Flan dataset effectively enhances multi-language task performance on held-out tasks.

Moreover, we measure the performance on MMLU when given different extra paradigm tokens, i.e., {<R>, <S>, <X>}. We find that the performance with different extra paradigm tokens shows differences during the UL2 pre-training stage, while these differences gradually diminish in the subsequent stages. This might be attributed to the fact that we utilize these extra paradigm tokens to guide the mode-switching only in the first stage for different UL2 tasks. Specifically, the performance

(a) MMLU

(b) CMMLU

(c) BELEBELE

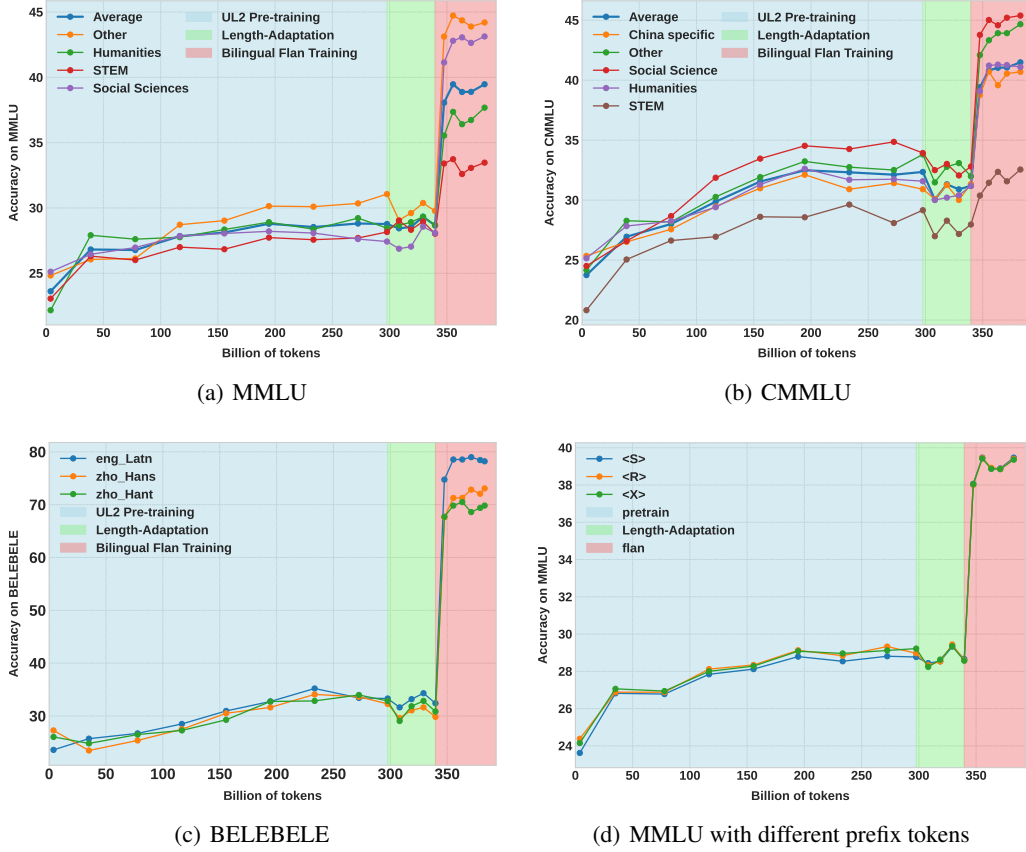(d) MMLU with different prefix tokens

Figure 6: Evolution of model performance during training.

for the S-denoising task in continuous writing is slightly inferior compared to the X-denoising and R-denoising tasks for masked span recovery.

6 Conclusion and Future Work

In this report, we present OpenBA, an Open-sourced 15B Bilingual Asymmetric seq2seq model pre-trained from scratch. We provide all the necessary details to pre-train an asymmetric seq2seq model from scratch, including 1) how to construct and process the pre-training data, 2) how to construct the Bilingual Flan data collection, 3) the implementation details of model architectures, configurations, objectives, and training pipelines. We also release our codes to supplement the descriptions of this report. On a variety of benchmarks, though fed with 380B tokens, OpenBA obtains remarkable performance, e.g., CMMLU and BELEBELE, and even surpasses the models consuming significantly more data.

Work in Progress We are currently working on the following directions about our model:
• We are conducting further evaluation to comprehensively calibrate the generation capability of OpenBA, especially for various tasks of controlled text generation (Tang et al., 2023a), and open-ended long text generation (Liang et al., 2023b).
• OpenBA faces ethical challenges and is prone to biases and toxicity since we have not yet performed any alignment operations (Ouyang et al., 2022). After the alignment stage, we would like to test a few effective detoxification strategies on our model, e.g., detox-chain (Tang et al., 2023b).
• The model's conversational capabilities need to be optimized for dialogue use cases (Yan et al., 2022), such as the generation correctness (Tang et al., 2021b; Bryant et al., 2022).
• The ability to invoke tools, as we have tried to use sentinel tokens at the UL2 pre-training stage to activate different tool usage, i.e., multi-modal generation invoked by tools Wu et al. (2023a).

• OpenBA needs to be further extended in terms of input and output length to adapt to a wider range of tasks, such as dialogue generation.

Acknowledgments

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856, 2019.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. *arXiv preprint arXiv:2308.16884*, 2023.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation*, volume 2, pp. 1–61. Association for Computational Linguistics, 2019.

Elnaz Barshan and Paul Fieguth. Stage-wise training: An improved feature learning strategy for deep models. In *Feature extraction: Modern questions and challenges*, pp. 49–59. PMLR, 2015.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, pp. 1–59, 2022.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Sahil Chaudhary. Code alpaca: An instruction-following llama model for code generation. `https://github.com/sahil280114/codealpaca`, 2023.

Zhihong Chen, Feng Jiang, Junying Chen, Tiannan Wang, Fei Yu, Guiming Chen, Hongbo Zhang, Juhao Liang, Chen Zhang, Zhiyi Zhang, et al. Phoenix: Democratizing chatgpt across languages. *arXiv preprint arXiv:2304.10453*, 2023.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834*, 2020.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Aidan Clark, Diego De Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. In *International Conference on Machine Learning*, pp. 4057–4086. PMLR, 2022.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018.

Mike Conover, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, et al. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023.

William Coster and David Kauchak. Simple english wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 665–669, 2011.

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. A span-extraction dataset for Chinese machine reading comprehension. 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)*, pp. 5886–5891. Association for Computational Linguistics, 2019.

Yiming Cui, Ziqing Yang, and Xin Yao. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 320–335, 2022.

Nan Duan. Overview of the nlpcc-iccpol 2016 shared task: Open domain chinese question answering. In *NLPCC/ICCPOL*, 2016.

Kevin Duh. The multitarget ted talks task. `http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/`, 2018.

Claire Cardie Faisal Ladhak, Esin Durmus and Kathleen McKeown. Wikilingua: A new benchmark dataset for multilingual abstractive summarization. In *Findings of EMNLP, 2020*, 2020.

Zihao Fu, Wai Lam, Qian Yu, Anthony Man-Cho So, Shengding Hu, Zhiyuan Liu, and Nigel Collier. Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder. *arXiv preprint arXiv:2304.04052*, 2023.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538, 2022.

Paul Grun. Introduction to infiniband for end users. *White paper, InfiniBand Trade Association*, 55, 2010.

Jian Guan, Xiaoxi Mao, Changjie Fan, Zitao Liu, Wenbiao Ding, and Minlie Huang. Long text generation by modeling sentence-level and discourse-level coherence. *arXiv preprint arXiv:2105.08963*, 2021.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence Moss. Ocnli: Original chinese natural language inference. In *Findings of the Association for Computational Linguistics*, pp. 3512–3526. Association for Computational Linguistics, 2020.

Xuming Hu, Zhijiang Guo, GuanYu Wu, Aiwei Liu, Lijie Wen, and Philip Yu. Chef: A pilot chinese dataset for evidence-based fact-checking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3362–3376. Association for Computational Linguistics, 2022.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*, 2023.

Baichuan Inc. Baichuan-7b. https://github.com/baichuan-inc/Baichuan-7B, 2023.

Zhiyuan Liu Jiahua Liu, Yankai Lin and Maosong Sun. Xqa: A cross-lingual open-domain question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2358–2368, 2019.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*, 2023a.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

Shuangjie Li, Wei He, Yabing Shi, Wenbin Jiang, Haijin Liang, Ye jiang, Yang Zhang, Yajuan Lyu, and Yong Zhu. Duie: A large-scale chinese dataset for information extraction. In *Tang, J., Kan, MY., Zhao, D., Li, S., Zan, H. (eds) Natural Language Processing and Chinese Computing*, volume 11839. Springer, Cham, 2019.

Yudong Li, Yuqing Zhang, Zhe Zhao, Linlin Shen, Weijie Liu, Weiquan Mao, and Hui Zhang. Csl: A large-scale chinese scientific literature dataset. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 3917–3923. International Committee on Computational Linguistics, 2022.

Zuchao Li, Shitou Zhang, Hai Zhao, Yifei Yang, and Dongjie Yang. Batgpt: A bidirectional autoregressive talker from generative pre-trained transformer. *arXiv preprint arXiv:2307.00360*, 2023b.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. Xlm-v: Overcoming the vocabulary bottleneck in multilingual masked language models. *arXiv preprint arXiv:2301.10472*, 2023a.

Xiaobo Liang, Zecheng Tang, Juntao Li, and Min Zhang. Open-ended long text generation via masked language modeling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 223–241, 2023b.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Xiaojun Liu, Chuang Zhang, Xiaojun Chen, Yanan Cao, and Jinpeng Li. Clts: a new chinese long text summarization dataset. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pp. 531–542. Springer, 2020.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.

Qi Lv, Ziqiang Cao, Lei Geng, Chunhui Ai, Xu Yan, and Guohong Fu. General and domain adaptive chinese spelling check with error consistent pretraining. In *ACM Trans. Asian Low-Resour. Lang. Inf. Process*. Association for Computing Machinery, 2022.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*, 2021.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 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*, pp. 839–849, 2016.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*, 2022.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1314–1331. IEEE, 2020.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023a.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023b. URL https://arxiv.org/abs/2306.01116.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6319.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

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

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

Robert R Schaller. Moore's law: past, present and future. *IEEE spectrum*, 34(6):52–59, 1997.

Chih Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. Drcd: a chinese machine reading comprehension dataset. *arXiv preprint arXiv:1806.00920*, 2018.

Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 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*, pp. 3257–3268. Association for Computational Linguistics, 2019a.

Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. Long and diverse text generation with planning-based hierarchical variational model. *arXiv preprint arXiv:1908.06605*, 2019b.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Yaozong Shen, Lijie Wang, Ying Chen, Xinyan Xiao, Jing Liu, and Hua Wu. An interpretability evaluation benchmark for pre-trained language model. *arXiv preprint arXiv:2207.13948*, 2022.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*, 2022.

Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, et al. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model. *arXiv preprint arXiv:2208.01448*, 2022.

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

Tianxiang Sun, Xiaotian Zhang, Zhengfu He, Peng Li, Qinyuan Cheng, Hang Yan, Xiangyang Liu, Yunfan Shao, Qiong Tang, Xingjian Zhao, Ke Chen, Yining Zheng, Zhejian Zhou, Ruixiao Li, Jun Zhan, Yunhua Zhou, Linyang Li, Xiaogui Yang, Lingling Wu, Zhangyue Yin, Xuanjing Huang, and Xipeng Qiu. Moss: Training conversational language models from synthetic data. 2023.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*, 2021.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Hongxuan Tang, Hongyu Li, Jing Liu, Yu Hong, Hua Wu, and Haifeng Wang. Dureader_robust: A chinese dataset towards evaluating robustness and generalization of machine reading comprehension in real-world applications. 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, pp. 955–963, 2021a.

Zecheng Tang, Yixin Ji, Yibo Zhao, and Junhui Li. Chinese grammatical error correction enhanced by data augmentation from word and character levels. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics, Hohhot, China*, pp. 13–15, 2021b.

Zecheng Tang, Pinzheng Wang, Keyan Zhou, Juntao Li, Ziqiang Cao, and Min Zhang. Can diffusion model achieve better performance in text generation? bridging the gap between training and inference! *arXiv preprint arXiv:2305.04465*, 2023a.

Zecheng Tang, Keyan Zhou, Pinzheng Wang, Yuyang Ding, Juntao Li, et al. Detoxify language model step-by-step. *arXiv preprint arXiv:2308.08295*, 2023b.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. Ul2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, May 2021.

Haochun Wang, Chi Liu, Nuwa Xi, Zewen Qiang, Sendong Zhao, Bing Qin, and Ting Liu. Huatuo: Tuning llama model with chinese medical knowledge. *arXiv preprint arXiv:2304.06975*, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yan Wang, Xiaojiang Liu, and Shuming Shi. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 845–854. Association for Computational Linguistics, 2017.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 4: 795–813, 2022.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023a.

Yang Wu, Yanyan Zhao, Zhongyang Li, Bing Qin, and Kai Xiong. Improving cross-task generalization with step-by-step instructions. *arXiv preprint arXiv:2305.04429*, 2023b.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. Sequential matching network: A new archtechture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pp. 496–505. Association for Computational Linguistics, 2017.

Bright Xu. Nlp chinese corpus: Large scale chinese corpus for nlp, September 2019. URL `https://doi.org/10.5281/zenodo.3402023`.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. Clue: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4762–4772. International Committee on Computational Linguistics, 2020.

Liang Xu, Xiaojing Lu, Chenyang Yuan, Xuanwei Zhang, Huilin Xu, Hu Yuan, Guoao Wei, Xiang Pan, Xin Tian, Libo Qin, and Hu Hai. Fewclue: A chinese few-shot learning evaluation benchmark. *arXiv preprint arXiv:2107.07498*, 2021.

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

Rui Yan, Juntao Li, Zhou Yu, et al. Deep learning for dialogue systems: Chit-chat and beyond. *Foundations and Trends® in Information Retrieval*, 15(5):417–589, 2022.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, et al. Pangu-$alpha$: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *arXiv preprint arXiv:2104.12369*, 2021.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

Min Zhang and Juntao Li. A commentary of gpt-3 in mit technology review 2021. *Fundamental Research*, 1(6):831–833, 2021.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023a.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

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

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*, 2023b.

Chujie Zheng, Minlie Huang, and Aixin Sun. ChID: A large-scale Chinese IDiom dataset for cloze test. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 778–787. Association for Computational Linguistics, 2019.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

Hao Zhou, Chujie Zheng, Kaili Huang, Minlie Huang, and Xiaoyan Zhu. Kdconv: A chinese multi-domain dialogue dataset towards multi-turn knowledge-driven conversation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7098–7108. Association for Computational Linguistics, 2020.

Qiyuan Chen Ziang Leng and Cheng Li. Luotuo: An instruction-following chinese language model, lora tuning on llama. `https://github.com/LC1332/Chinese-alpaca-lora`, 2023.

A   Instruction Template

The task instruction prompts for evaluation are provided here:

Test prompt example for MMLU:

Context:
(Examplar)
Question: Which of the following occurred first during the separation of the elements of Pangaea through continental drift? Options: A. Gondwana and Laurasia were formed. B. Africa separated from South America. C. India collided with Eurasia to form the Himalayan mountain chain. D. Australia separated from the rest of the continental landmasses. Answer:A
...*(Other examplars, if any)*

(Test case)
Question: Experiments on song development in birds have shown that when a young male reared in isolation hears only the song of a different bird species, he will develop an adult song repertoire that lacks certain characteristics typical of his own species. This result shows that the song of his species is most likely Options: A. entirely learned during development B. entirely instinctive C. both instinctive and learned D. dependent upon hormones for proper development Answer:

Response: A

Test prompt example for CMMLU:

Context:
(Instruction)
以下是关于(大学教育学)的单项选择题，请直接给出正确答案的选项。

(Examplar)
题目：在古代文献记载中，我国西周时期设在王都的小学和大学，总称为()
A. 都学 B. 乡学 C. 官学 D. 国学
答案是：D

...*(Other examplars, if any)*

(Test case)
以下是关于(大学教育学)的单项选择题，请直接给出正确答案的选项。
题目：教育的本质特征是 ()
A. 系统性 B. 知识性 C. 科学性 D. 育人性
答案是：

Response: D

Test prompt example for C-Eval:

Context:
(Instruction)
以下是关于(中国语言文学)的单项选择题，请直接给出正确答案的选项。

(Examplar)
题目：元朝政府曾经实行残酷的民族政策，把全国人民分为____四个等级。
A. 色目人、蒙古人、汉人、南人 B. 蒙古人、汉人、南人、色目人 C. 蒙古人、南人、色目人、汉人 D. 蒙古人、色目人、汉人、南人
答案是：D

...(Other examplars, if any)

(Test case)
以下是关于(中国语言文学)的单项选择题，请直接给出正确答案的选项。
题目：《国语》和____，都是国别史。
A. 《左传》 B. 《战国策》 C. 《史记》 D. 《汉书》
答案是：

Response: D

Test prompt example for BBH:

Context:
(Examplar)
not (True) and (True) is
Answer: False

...(Other examplars, if any)

False or not not not False and True is

Response: True

Test prompt example for $En \Rightarrow Zh$ Machine Translation:

Context:
将以下中文翻译成英文，并输出英文翻译：
Local authorities are warning residents in the vicinity of the plant to stay indoors, turn off air-conditioners and not to drink tap water.

Response:
当地政府警告核电站附近的居民，要待在室内，关掉空调，不要喝自来水。

Test prompt example for $Zh \Rightarrow En$ Machine Translation:

Context:
将以下英文翻译成中文，并输出中文翻译：
当地政府警告核电站附近的居民，要待在室内，关掉空调，不要喝自来水。

Response:
Local government warns residents near nuclear power plant to stay indoors, turn off air conditioning, and do not drink bottled water.

Test prompt example for BoolQ:

> **Context:**
> Parity (mathematics) – In mathematics, parity is the property of an integer's inclusion in one of two categories: even or odd. An integer is even if it is evenly divisible by two and odd if it is not even. For example, 6 is even because there is no remainder when dividing it by 2. By contrast, 3, 5, 7, 21 leave a remainder of 1 when divided by 2. Examples of even numbers include -4, 0, 82 and 178. In particular, zero is an even number. Some examples of odd numbers are -5, 3, 29, and 73.
> question: can an odd number be divided by an even number?
> answer:

> **Response:** yes

Test prompt example for RTE:

> **Context:**
> Yet, we now are discovering that antibiotics are losing their effectiveness against illness. Disease-causing bacteria are mutating faster than we can come up with new antibiotics to fight the new variations.
> Can we say the following?
> Bacteria is winning the war against antibiotics.
> OPTIONS: - yes - no

> **Response:** yes

Test prompt example for ReCoRD:

> **Context:**
> Tracy Morgan hasn't appeared on stage since the devastating New Jersey crash that nearly ended his life last summer, but all that will change this fall when he returns to host Saturday Night Live. NBC announced on Twitter Monday that Morgan, an SNL alum with seven seasons as a cast member under his belt, will headline the third episode of Season 41 airing October 17. For Morgan, 46, it will be a second time hosting the long-running variety show, the first since the June 2014 pileup on the New Jersey Turnpike that killed his friend and mentor James 'Jimmy Mack' McNair.
> @highlight
> Morgan, 46, will host third episode of season 41 of SNL airing October 17
> @highlight
> He tweeted to his fans: 'Stoked to be going home...#SNL'
> @highlight
> For the SNL alum who had spent seven years as cast member, it will be a second time hosting the show
> @highlight
> Morgan has been sidelined by severe head trauma suffered in deadly June 2014 crash on New Jersey Turnpike that killed his friend
> @highlight
> First episode of new SNL season will be hosted by Miley Cyrus, followed by Amy Schumer

> **Response:**
> On October 10, acclaimed comedian and star of the summer box office hit Trainwreck Amy Schumer will make her SNL debut, followed by Morgan a week later.

Test prompt example for WSC:

> *Context:*
> Bernard , who had not told the government official that he was less than 21 when he filed for a homestead claim, did not consider that he had done anything dishonest. Still, anyone who knew that he was 19 years old could take his claim away from him . "him" refer to what?

> *Response:* anyone

Test prompt example for WiC:

> *Context:*
> An emerging professional class.
> Apologizing for losing your temper, even though you were badly provoked, showed real class.
> The word 'class' used in the same way in the two sentences above, True or False?
> answer:

> *Response:* False

Test prompt example for CoPA:

> *Context:*
> The man turned on the faucet, so __
> A. The toilet filled with water. B. Water flowed from the spout.
> answer:

> *Response:* B

Test prompt example for MultiRC:

> *Context:*
> Please read the following article and judge whether the answer to the question is correct:
> What causes a change in motion? The application of a force. Any time an object changes motion, a force has been applied. In what ways can this happen? Force can cause an object at rest to start moving. Forces can cause objects to speed up or slow down. Forces can cause a moving object to stop. Forces can also cause a change in direction. In short, forces cause changes in motion. The moving object may change its speed, its direction, or both. We know that changes in motion require a force. We know that the size of the force determines the change in motion. How much an objects motion changes when a force is applied depends on two things. It depends on the strength of the force. It also depends on the objects mass. Think about some simple tasks you may regularly do. You may pick up a baseball. This requires only a very small force.
> questions: Would the mass of a baseball affect how much force you have to use to pick it up?
> answer: No.
> Is this answer True or False?

> *Response:* False

Test prompt example for AX_b:

> *Context:*
> Read the sentence below and answer the question: The cat sat on the mat.
> Question: The cat did not sit on the mat. True or False?
> Answer:

> *Response:* False

Test prompt example for AX_g

> *Context:*
> Read the sentence below and answer the question: The taxpayer met with the accountant to get help filing his taxes.
> Question: The accountant sought help filing taxes. True or False?
> Answer:

> *Response:* False

Test prompt example for BELEBELE:

> *Context:*
> Please read the following article and answer the questions:
> Make sure your hand is as relaxed as possible while still hitting all the notes correctly - also try not to make much extraneous motion with your fingers. This way, you will tire yourself out as little as possible. Remember there's no need to hit the keys with a lot of force for extra volume like on the piano. On the accordion, to get extra volume, you use the bellows with more pressure or speed.
> Please answer the following multiple-choice questions:
> According to the passage, what would not be considered an accurate tip for successfully playing the accordion? Options: (A) For additional volume, increase the force with which you hit the keys (B) Keep unnecessary movement to a minimum in order to preserve your stamina (C) Be mindful of hitting the notes while maintaining a relaxed hand (D) Increase the speed with which you operate the bellows to achieve extra volume Answer: (

> *Response:* B

Test prompt example for CLTS:

> *Context:*
> 简要概括以下文本：
> 温布尔登网球锦标赛官方微博发布消息称，全英草地网球俱乐部今天非常遗憾地决定，由于新冠状病毒疫情持续蔓延产生的公共卫生安全问题，将取消2020年温布尔登网球锦标赛，第134届锦标赛将于2021年6月28日至7月11日举行。（原题为《2020年温网锦标赛取消 将于2021年6月至7月举行》）(本文来自澎湃新闻，更多原创资讯请下载"澎湃新闻"APP)
> 概括为：

> *Response:*
> 由于新冠状病毒疫情持续蔓延产生的公共卫生安全问题，将取消2020年温布尔登网球锦标赛，第134届锦标赛将于2021年6月28日至7月11日举行。

B   Chinese Flan Collection

As shown in Fig. 7, the unified format of each data includes "instruction," "input," "output," "domain" and "task", where "instruction" denotes the description of the task that provides LLM with a clear purpose. "input" and "output" are the question and answer respectively. "domain" is the topic of the task, such as medicine, news and so on. The last one "task" indicates the type categorized into one of eighteen task types. Table 14 shows all the tasks and the source of the instruction datasets in each task.

Example of Unified Json Format:
```
{
"instruction": "请将输入的英语句子翻译成中文",
"input": "Yet not nearly enough has been invested in this effort.",
"output": "但目前这方面的投入还远远不够。",
"domain": "多领域",
"task": "机器翻译"
}
```

Figure 7: An example of the unified format of Chinese Flan Dataset.

Task	Source	Dataset
Question Answering	Jiahua Liu & Sun (2019)	XQA
	Duan (2016)	ChineseDBQA
	Artetxe et al. (2019)	Xquad
	`https://www.luge.ai/#/luge/dataDetail?id=40`	ChineseBiomedicalQA
Text Classification	Conneau et al. (2018)	XNLI
	`http://tcci.ccf.org.cn/conference/2014/dldoc/evtestdata6.zip`	Chinese News Categorization
	`https://tianchi.aliyun.com/dataset/133838?spm=a2c22.28136470.0.0.6e5a6a23SPZMrX&from=search-list`	TNEWS
	`https://huggingface.co/datasets/dirtycomputer/ChnSentiCorp_htl_all`	ChnSentiCorp_htl_all
	`https://storage.googleapis.com/cluebenchmark/tasks/iflytek_public.zip`	iflytek
Sentiment Classification	Xu et al. (2021)	FewCLUE EPRSTMT
	`https://www.luge.ai/#/luge/dataDetail?id=25`	ChnSentiCorp
	`https://www.heywhale.com/mw/dataset/5e09a9eb2823a10036b126c0/file`	BDCI 2019
	`https://www.luge.ai/#/luge/dataDetail?id=20`	NLPCC14-SC
Named Entity Recognition	`https://huggingface.co/datasets/msra_ner`	MSRA_NER
	`https://storage.googleapis.com/cluebenchmark/tasks/cluener_public.zip`	CLUE Fine-Grain NER
Text Matching	Xu et al. (2020)	CLUE WSC 2020
	Xu et al. (2020)	CMNLI
	Hu et al. (2020)	OCNLI
	`https://www.luge.ai/#/luge/dataDetail?id=39`	CINLID
	`https://tianchi.aliyun.com/dataset/106411`	AFQMC
Text Summarization	Li et al. (2022)	CSL
	Faisal Ladhak & McKeown (2020)	WikiLingua Chinese
	`http://tcci.ccf.org.cn/conferen`	Weibo Oriented Chinese News Summarization NLPCC2015
Reading Comprehension	Xu et al. (2020)	C3
	Cui et al. (2019)	CMRC2018
	Shao et al. (2018)	DRCD
Question Generation	Tang et al. (2021a)	DuReader_QG
	`https://tianchi.aliyun.com/dataset/dataDetail?dataId=86895`	TCM Literature Question Generation
Dialogue	Wu et al. (2017)	douban
	Zhou et al. (2020)	kdconv
	`https://www.luge.ai/#/luge/dataDetail?id=38`	Chinese Persona Chat
Machine Translation	Duh (2018)	mttt
	Xu (2019)	translation 2019 zh
	Barrault et al. (2019)	WMT19 en-zh
	`https://www.kaggle.com/datasets/garyongguanjie/wikititles-zhen`	wikititles_en-zh
Cloze Test	Zheng et al. (2019)	ChiD
Text Generation	Shao et al. (2019a)	AdvertiseGen
Semantic Analysis	Wang et al. (2017)	Math23K
Relation Extraction	Li et al. (2019)	DuIE2.0
Grammatical Error Correction	Lv et al. (2022)	MD-SCS
Fact-checking	Hu et al. (2022)	CHEF
Interpretable Evaluation	Shen et al. (2022)	DuExplain
Event Extraction	`https://tianchi.aliyun.com/dataset/dataDetail?dataId=110904`	tianchi_event_doclevel_attr

Table 14: All types of tasks and the source of the instruction datasets in each task.