Data-centric Learning is All LLMs Need

Anonymous ACL submission

Abstract

In the past year, a multitude of open-source large language models (LLMs) have emerged, demonstrating increasingly robust performance. However, numerous issues surrounding these LLMs remain unresolved. In this work, we focus on how data influences LLMs. We propose a new data processing pipeline to obtain high-quality data. Subsequently, we introduce an efficient pre-training strategy, continuing pre-training LLaMA2 to obtain the Ziya2 model. Ziya2 significantly outperforms LLaMA2 on all selected benchmarks and meets or exceeds the performance of representative open-source models of the same size. Upon analyzing the intermediate checkpoints, we find that different types of data vary in their efficiency in enhancing the performance of Ziya2. Thus, we define three data attributes and establish new data-centric scaling laws to illustrate how different data impacts LLMs. Finally, we arrive at two conclusions that improving the 'Coherence' and 'Readability' of the data provides a larger boost to the LLMs than improving the 'Similarity' to the downstream tasks.

1 Introduction

Large Language Models (LLMs) have achieved great success in the field of artificial intelligence (AI), especially natural language processing (NLP), over the past few years. Generally, LLMs are pre-trained on large amounts of text and show promising performance in a variety of NLP tasks without requiring intensive task-specific tuning on a large amount of labeled data (Devlin et al., 2019; Raffel et al., 2020; Joshi et al., 2020; Qin et al., 2021; Wang et al., 2022; Lu et al., 2022; Tian et al., 2023; Ping et al., 2023; Huang et al., 2023a). Representative LLMs include ChatGPT (OpenAI, 2022) and its successors GPT-4 (OpenAI, 2023) and PaLM-2 (Anil et al., 2023), demonstrating strong performance and adaptability.

Recently, in response to the difficulty of obtaining parameters and source code for the above closed-source LLMs, many researchers have turned to train open-source alternatives, such as LLaMA2 (Touvron et al., 2023b) and Falcon (Penedo et al., 2023). The open-source LLMs provide a foundation for continual pre-training and promote transparency and accountability in AI research.

However, in addition to the various benefits that open-source LLMs bring, the development of LLMs is currently facing the following three major challenges. The first one is that pre-training models from scratch necessitate long-term training on many GPUs, rendering the training of LLMs an extremely high-cost process. Continual pre-training presents a relatively cost-effective solution, yet it may be accompanied by issues such as catastrophic forgetting, which could be attributed to the disparities in data distribution. The second one is that open-source LLMs often do not come with open-source data. The effectiveness of data processing methods directly impacts the performance of LLMs. Currently, there is no standardized methodology or criteria for cleaning pre-training data. The third one is that many studies on LLMs prioritize expanding the capacity of model parameters and pre-training data to enhance model performance, often overlooking the impact of the quality of pre-training data on model performance. Specifically, to our best knowledge, there is no study investigating which attributes of pre-training data exert the most significant influence on the LLMs. Under constraints of computational budget and a certain number of parameters, it is also worth exploring which type of data should be prioritized.

In this work, we focus on the continual pre-training strategies and understanding the intricate relationship between data and model performance. First, we propose a new data processing pipeline consisting of five steps to derive high-quality pre-training data from a vast corpus. We also involve

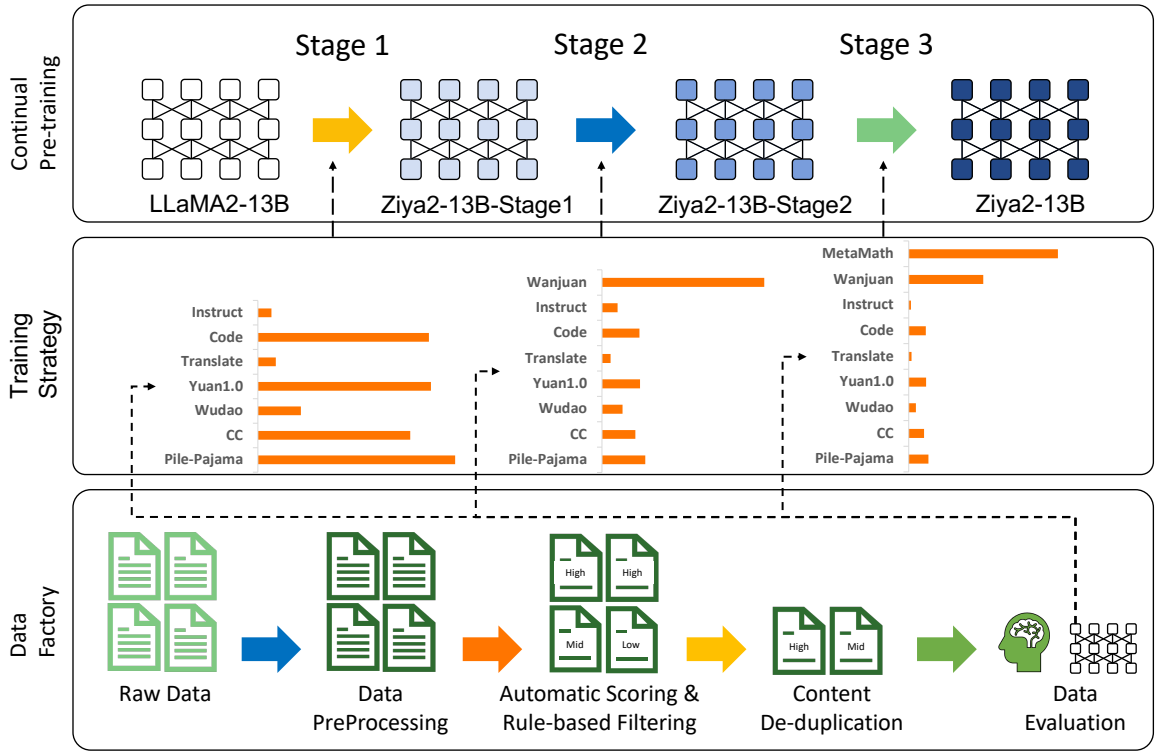

Figure 1: The overall data-centric process for training Ziya2, where the pipeline to obtain high-quality data, the training strategies, and the three-stage training process are presented in the bottom, middle, and top parts of the figure, respectively. The training strategies for different stages are illustrated by comparing the data distributions.

human evaluation to ensure the quality of the processed data. Then, we use LLaMA2-13B as the base model and propose a three-stage continual pre-training strategy. The first training stage utilizes lots of unsupervised data in English and Chinese. The second stage uses a relatively small amount of data but contains many supervised datasets. The third stage employs little augmented data that focuses on improving math abilities. Finally, we obtain Ziya2, and the whole data-centric training process is shown in Figure 1.

The evaluation of Ziya2 is performed on several representative benchmarks, and the results show that Ziya2 has a significant improvement over LLaMA2. Compared to other open-source models of the same size, Ziya2 also demonstrates superior performance. Our analysis of the intermediate checkpoints indicates that the impact of data from different stages on the model capabilities varies. Consequently, we conduct additional experiments and establish new data-centric scaling laws. We draw two conclusions suggesting that improving the 'Coherence' and 'Readability' of data is more effective in improving LLMs' capabilities than improving 'Similarity' to downstream tasks.

In summary, our contributions include: (1) We propose a new data processing pipeline and contribute over 600 billion tokens of high-quality data. (2) We propose an effective continual pre-training strategy and obtain the Ziya2 model. We also perform a detailed analysis of the impact of the three stages' data on intermediate checkpoints. (3) We define three data attributes and analyze their effect on the LLM's performance. We then establish the data-centric scaling laws and arrive at two conclusions for future research.

2 Data Factory

Data serves as the fundamental cornerstone for LLMs, with the scale and quality of the data determining the effectiveness of these models. In this work, we introduce a new data processing pipeline, as depicted in Figure 2. Ultimately, we have obtained high-quality data exceeding 600 billion tokens in scale.

2.1 Data Processing Pipeline

This pipeline encompasses a variety of functions, including data preprocessing (DP), automatic scoring (AS), rule-based filtering (RF), content deduplication (CD), and data evaluation (DE):

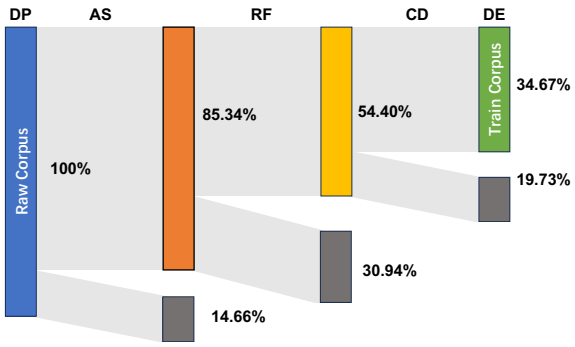

Figure 2: An overview of the proposed data processing pipeline, including five components: data preprocessing (DP), automatic scoring (AS), rule-based filtering (RF), content deduplication (CD), and data evaluation (DE). The gray and colored blocks indicate the proportion of data that is filtered out and retained relative to the original dataset, respectively.

Data Preprocessing (DP) Initially, we conduct a language detection on the collected corpus, selecting only English and Chinese data. Subsequently, we standardize the encoding of the corpus and convert all Traditional Chinese text into Simplified Chinese text. Following this, we eliminate meaningless tokens, such as invisible control characters, special symbols, emoticons, and so on.

Automatic Scoring (AS) For the preprocessed data, we perform an automatic quality control (filtering) by language model scoring. We employ KenLM (Heafield, 2011) to train two models seperately from the Chinese Wikipedia and the English Wikipedia and use them to evaluate the perplexity (PPL) of the data. We rank the PPL score from low to high. The top 30% of data in terms of PPL ranking is considered high-quality data, and data with PPL ranking between 30% and 60% is deemed medium-quality data. We only keep high-quality data and medium-quality data.

Rule-based Filtering (RF) Data mined from the internet may contain a substantial amount of explicit, terrorist, and discriminatory text, posing ethical challenges for LLMs (Zhuo et al., 2023; Touvron et al., 2023b). Therefore, we design more than 30 filtering rules at three levels: document, paragraph, and sentence. At the document level, rules are designed around content length and format. At the paragraph and sentence levels, the focus of the rules shifts towards the toxicity. Notably, in the initial stages of rule design, we also randomly sample some original text for human evaluation. We then update the rules based on the human feedback to ensure the effectiveness of the data filtering process.

Content Deduplication (CD) Duplicate text within the pre-training data can detrimentally impact the performance of LLMs (Lee et al., 2021; Tirumala et al., 2023; Penedo et al., 2023). Thus, we use Bloomfilter (Bloom, 1970) and Simhash (Charikar, 2002) to deduplicate the text. We also use caching and bucketing techniques to optimize this process. The detailed deduplication steps are as follows: First, we use Bloomfilter to deduplicate text based on URLs, which significantly reduces the computational load required for subsequent content deduplication. Second, we use Bloomfilter to perform precise deduplication on these web pages. Finally, we employ SimHash for the fuzzy deduplication of textual content. Although this step may exclude some high-quality data, the evaluation of the sampled deduplicated data indicates that this loss is acceptable in the pursuit of greater training efficiency.

Data Evaluation (DE) The last step of the pipeline is to evaluate the processed data. We evaluate the data quality using both automatic and human evaluations with specific metrics in Appendix F. For automatic evaluation, we randomly select 1% of the processed data to conduct evaluation. For human evaluation, we randomly sample 1,000 instances from the processed data and hire workers to evaluate them. Based on the automatic and human evaluations, we determine whether an example meets our data quality criteria. Then, we compute the rate of the unqualified examples over all evaluated instances. If the rate is lower than a threshold, we consider these data to meet our quality criteria and include them in the training datasets. If the rate is higher than the threshold, we consider the data does not meet our criteria. We will be more rigorous with processes like automatic scoring and rule-based filtering, and then conduct evaluations until the processed data meets the criteria.

2.2 High-quality Pre-training Data

We use the proposed data processing pipeline to clean multiple pre-training datasets: Pile-Pajama, CC, Wudao (Yuan et al., 2021), Yuan1.0 (Wu et al., 2021), Wanjuan (He et al., 2023), MetaMath (Yu et al., 2023) and pre-training data collected by ourselves, including code and book content. Please refer to Appendix A for more details.

Table 1 shows the statistics of the high-quality

Pre-training Stage	Dataset	Language	Size	Doc #	Token #
Stage 1	Pile-Pajama	en	400GB	47M	110B
	CC	en	384GB	52M	109B
	Wudao	zh	156GB	51M	48B
	Yuan1.0	zh	590GB	260M	193B
	Translate	multi	3GB	12M	1.5B
	Code	multi	480GB	124M	191B
	Instruct	multi	1.6GB	0.9M	0.8B
Stage 2	Wanjuan	zh	76GB	16M	29B
Stage 3	MetaMath	en	0.3GB	0.4M	0.1B

Table 1: The statistics of the obtained high-quality pre-training datasets. “en”, “zh”, and “multi” indicate that the main language of the dataset is English, Chinese, and multilingual, respectively.

pre-training dataset we constructed. We use different pre-training data in the three continual pre-training stages of LLMs. The data in the first stage contains solely unsupervised data. The second stage incorporates a substantial amount of supervised data. The data in the third stage is an augmentation of data specifically for mathematical tasks. We will elaborate on how to use them in the following section. Some pre-training data examples are presented in Appendix H.

3 Training Strategy

In this section, we briefly introduce the proposed training strategy. We continue pre-training LLaMA2 (Touvron et al., 2023b) on the constructed high-quality dataset using a novel three-stage training strategy.

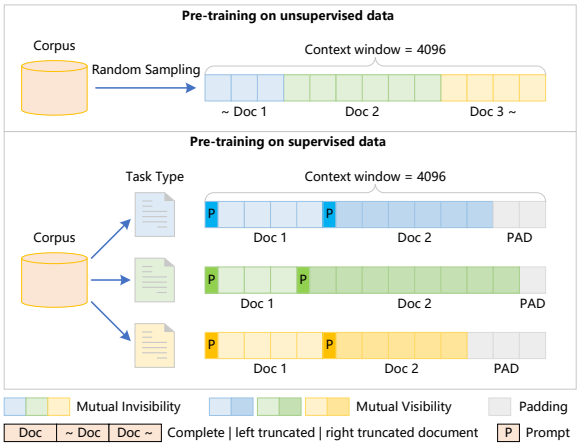

Figure 3: An illustration of how to process unsupervised and supervised data in the three training stages.

LLaMA2 performs excellently on general English benchmarks such as MMLU (Hendrycks et al., 2020) but performs poorly in solving math and code programming problems. In addition, the fact that the pre-training data of LLaMA2 only contains little Chinese corpus also causes it to perform poorly on the Chinese benchmarks. We want to enhance the Chinese, mathematical, and code programming capabilities of LLaMA2 while maintaining or improving its performance in English. To avoid the issue of catastrophic forgetting, we propose a three-stage training strategy.

In the first training stage, the training data consists of 650 billion tokens and is unsupervised. We randomly splice different documents together, using the special token ‘<eos>’ to denote the end of a document. Text exceeding a length of 4,096 is truncated, as depicted in the upper half of Figure 3.

In the second stage, we train the model using supervised task data with English and Chinese prompts. The supervised tasks include classification, reasoning, knowledge of single-choice & multiple-choice, and question-answering. We design specific prompts for each task and concatenate these prompts with task documents. We still use the next token prediction method for continual pre-training. However, in contrast to the first stage, we concatenate documents of the same task type. Furthermore, we ensure that the length of the concatenated text is close to but does not exceed 4,096, with the remaining part filled with the ‘<pad>’ token. This process is shown in the lower half of Figure 3. To maintain the English capability of the model, we randomly sample data of the same size from the first stage.

In the third stage, we incorporate the supervised dataset MetaMath (Yu et al., 2023) to improve the mathematical reasoning abilities of the model, using the same composition approach as in the second stage. We also randomly sample 0.1B tokens from the first and the second stage data to mix with Meta-

Source	Model	MMLU	CMMLU	C-Eval	GSM8K	MATH	HumanEval
Closed-source	GPT-4	83.93	70.33	68.40	89.99	40.20	69.51
	GPT-3.5-turbo	68.54	54.06	51.10	57.77	13.96	52.44
Open-source	ChatGLM2-6B	47.86	49.30	51.70	28.89	6.40	9.15
	Falcon-7B	26.03	25.66	24.23	5.46	1.68	-
	Vicuna-13B	52.00	36.28	32.80	28.13	4.36	16.46
	XVERSE-13B	55.21	58.44	53.70	18.20	2.18	15.85
	WeMix-13B	59.70	42.60	42.70	45.20	6.90	24.40
	Baichuan2-13B	59.17	**61.97**	58.10	52.77	10.08	17.07
	LLaMA2-13B	55.74	37.90	34.99	28.81	4.98	15.24
Ours	Ziya-13B	43.88	31.09	28.97	17.97	3.08	14.02
	Ziya2-13B	**61.36**	60.95	**58.84**	**68.46**	**10.98**	**28.87**

Table 2: Comparison of Ziya2 with other closed-source and open-source LLMs on six benchmark datasets for LLM evaluation, where the **boldface** indicates the best-performing result over all open-source LLMs.

Math, respectively. Finally, we obtain the Ziya2 model.

We also improve the model structure of LLaMA2 to improve the training efficiency. Due to the space limitation, please refer to Appendix B for the improvement of the model structure and Appendix C for the training details.

4 Results

4.1 Evaluation Method

Following previous research (Zeng et al., 2022; Touvron et al., 2023b; Yang et al., 2023), we evaluate LLMs on six representative benchmarks: MMLU (Hendrycks et al., 2020), CMMLU (Li et al., 2023), C-Eval (Huang et al., 2023b), GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), and HumanEval (Chen et al., 2021). MMLU tests the English general abilities of LLMs. CMMLU and C-Eval test the Chinese general abilities of LLMs. GSM8K and MATH test the math ability of LLMs. HumanEval tests the code programming ability of LLMs. Please refer to Appendix D for more evaluation details.

4.2 Benchmark Results

We choose multiple open-source models with similar sizes to Ziya2 as baselines: ChatGLM2 (Zeng et al., 2022), Falcon (Penedo et al., 2023), Vicuna (Chiang et al., 2023), Baichuan2 (Yang et al., 2023), XVERSE, WeMix, LLaMA2 (Touvron et al., 2023b) and Ziya (Zhang et al., 2022). We also choose closed-source models GPT-3.5-turbo and GPT-4 (OpenAI, 2023) as references. Please refer to Appendix E for more details about baselines. The results of different LLMs on the six benchmarks are shown in Table 2 with the following observations.

First, Ziya2 significantly outperforms LLaMA2 on all the benchmarks. Specifically, for general English tasks, Ziya2 outperforms LLaMA2 by 6 points on MMLU. For general Chinese tasks, Ziya2 surpasses LLaMA2 by 23 and 24 points on CMMLU and C-Eval, respectively. For specific downstream tasks, Ziya2 outperforms LLaMA2 by 40, 6, and 13 points on GSM8K, MATH, and HumanEval datasets, respectively.

The results highlight the effectiveness of our continual pre-training strategy. It not only enhances LLaMA2's English capabilities and mitigates catastrophic forgetting but also significantly improves its performance in Chinese, mathematical, and code programming tasks.

Second, compared to open-source baselines of comparable size, Ziya2 outperforms almost all LLMs on all benchmarks. Remarkably, Ziya2's Chinese proficiency is even superior than that of GPT-3.5-turbo, and Ziya2's mathematical ability is on par with GPT-3.5-turbo. Although the Baichuan2-13B model performs comparably to Ziya2 on general tasks, Ziya2 significantly outperforms Baichuan2-13B in mathematical and code programming abilities. The results indicate that the Ziya2 model is at the forefront on the model scale of 13 billion parameters.

In Appendix I, we give examples generated by Ziya2 across three major domains: English, Chinese, and code programming.

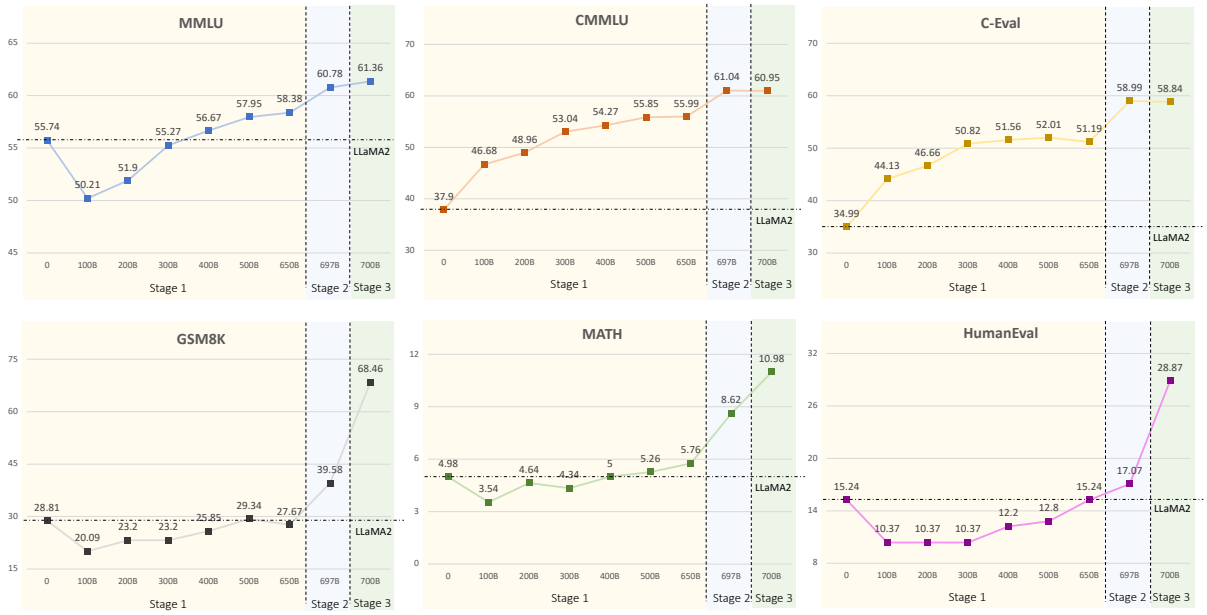

Figure 4: The performance of Ziya2 on the six benchmark datasets with respect to the three training stages. The training process in different stages is represented by the number of training tokens at a particular training step. The performance of LLaMA2 on the datasets is illustrated by dashed lines for reference.

4.3 Data Efficiency

In addition to the final results, we report the performance of Ziya2's intermediate checkpoints on six benchmarks in Figure 4. This section presents the observations from the three training stages, emphasizing the different effects of data on LLMs' performance.

In the first training stage of continual pre-training, Ziya2's performance on MMLU deteriorates due to the inclusion of a large amount of Chinese corpus in the training data that is different from the setting of LLaMA2. However, as the number of training steps increases, Ziya2 learns from a broader view of more data, which enhances its capabilities in both Chinese and English text processing. In particular, new data significantly improves Ziya2's performance on CMMLU and C-Eval benchmarks for Chinese tasks that LLaMA2 is not optimized for. A modest enhancement is also synchronously observed in Ziya2's mathematical and code programming abilities.

In the second stage, Ziya2 exhibits a more substantial enhancement on six benchmarks relative to the first stage, with notable advancements on CMMLU, C-Eval, GSM8K, and MATH. The results may underscore the greater contribution of supervised data over unsupervised data to continual pre-training. Therefore, employing supervised data for pre-training is able to reduce the number of training steps and economize costs.

In the third stage, we observe that training with the data-augmented dataset significantly improves the performance of Ziya2 on GSM8K, MATH, and HumanEval. On one hand, the experimental results prove that data augmentation specific to a particular dataset is able to significantly boost the model's performance on that dataset. On the other hand, such an "effortless" enhancement in model performance may not necessarily be beneficial, as the model might merely learn the format of the problems, rather than genuinely improving its mathematical capabilities. More details are given in the next section.

5 Data-centric Scaling Laws

We want to explore the effect of different data on the model's performance, keeping the size of the model constant. Previous research on the scaling laws simply represents data in terms of data volume (Kaplan et al., 2020; Hoffmann et al., 2022; Muennighoff et al., 2023). In this work, we employ two attributes, 'Coherence' and 'Readability' detailed in Appendix F, along with the 'Similarity' attribute to characterize the data. 'Coherence' and 'Readability' are also used to evaluate the quality of the pre-trained data in Section 2.1. We discard the 'Toxic' attribute due to the sparsity of its human evaluation results as showed in Table 3. 'Coherence' reflects the consistency and usefulness of the data content, 'Readability' assesses the

grammatical and formatting correctness of the data, and 'Similarity' measures the extent of similarity between the pre-training data and the test set of downstream tasks.

Hoffmann et al. (2022) propose Eq. 1: to minimize the loss (L) under the computational resource constraints (C, measured in FLOPs) through an optimal allocation of the model size (N) and the data volume (N).

$$\underset{\text{N,D}}{\text{argmin}} L(N, D) \quad \text{s.t. FLOPs}(N, D) = C \tag{1}$$

The loss (L) can be modeled as a parametric equation of the model size and the data volume, where {A, α, B, β, E} are learnable parameters:

$$L(N, D) = \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} + E \tag{2}$$

Our goal is to introduce a modified version of Eq. 2 to account for the effects of different attributes of the data on the LLM's performance. In the modified equation, the loss (L) is modeled as a parametric equation of 'Coherence (CH)', 'Readability (RA)', 'Similarity (SIM)', and the data volume (V), where $\{\alpha, A, B, C, E, F\}$ are learnable parameters:

$$L(D) = D^{-\alpha} + E \tag{3}$$

$$D = (A * \text{CH} + B * \text{RA} + \text{C} * \text{SIM}) * V + F \tag{4}$$

In contrast to Eq. 2 , we only conduct experiments on a fixed model size, so the term $\frac{A}{N^{\alpha}}$ can be omitted as a constant. V represents the data volume to continual pre-training. When V is 0, L denotes the loss of the initial pre-trained model on the test set. A, B, C are the coefficients of the data attributes. E represents the loss for an ideal generative process on the data distribution. F represents the effect of the pre-training data on the loss.

We perform human evaluations by taking 1,000 pieces of data from each of the three stages of continual pre-training data to obtain the values of 'Coherence' and 'Readability'. The results are shown in Table 3. We then extract 100,000 pieces of data from each of the three stages and calculate the 'Similarity' value of the data to the test set for each of the six downstream tasks, shown in Table 4. Different from the experimental setup in Section 4.3, we

Attibute	Stage1	Stage2	Stage3
Coherence	0.801	0.653	0.993
Readability	0.917	0.975	0.916
Toxic	0.990	1.000	1.000

Table 3: The values of 'Coherence', 'Readability', and 'Toxic' of data from the three stages. We perform human evaluations, and the criteria are shown in Figure 8.

Benchmark	Stage1	Stage2	Stage3
MMLU	0.6714	0.6677	**0.6827**
CMMLU	0.7057	**0.7417**	0.6567
C-EVAL	0.7024	**0.7374**	0.6634
GSM8K	0.6612	0.6570	**0.7524**
MATH	0.6552	0.6505	**0.7477**
HumanEval	0.5689	0.5563	**0.6211**

Table 4: The 'Similarity' value between data from the three stages and the test sets of the six tasks. We use the OpenAI's text embeddings api and cosine similarity to calculate the 'Similarity' value.

continue pre-training only on data from the three stages, respectively. We have not mixed the data with other stages to avoid interference. Then, we calculate the test loss for the downstream tasks at various checkpoints with different training data volumes. Finally, we have obtained 84 samples, each characterized by three unique data attributes, data volumes, and test loss values.

To estimate $\{\alpha, A, B, C, E, F\}$ in Eq. 3 and Eq. 4, we minimize the Huber loss (Huber, 1964) between the observed log loss $L_i^{'}$ and the predicted log loss, where LSE is the log-sum-exp operator:

$$\min_{\alpha,\text{A,B,C,e}} = \sum_{\text{Runs i}} \text{Huber}_{\delta} (\text{LSE}(1 - \alpha log D_i, e) - log L_i^{'}) \tag{5}$$

In Eq. 5, we set e = log(E) and $\delta = 10^{-3}$ for the Huber loss. We use the L-BFGS algorithm (Liu and Nocedal, 1989) to find the local minima of the objective above. More fitting details can be found in Appendix G. Empirically, we find after fitting Eq. 3 and Eq. 4 that

$$L(D) = D^{-0.06} + 0.31 \tag{6}$$

$$D = (1.66 * \text{CH} + 1.98 * \text{RA} + 0.70 * \text{SIM}) * V + 1.24 * 10^{10} \tag{7}$$

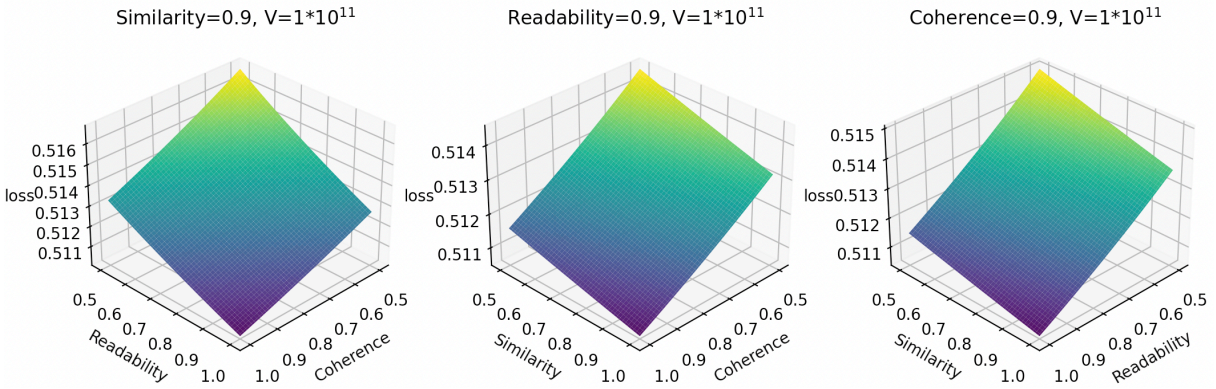

Figure 5: The different effects of three data attributes on the loss. The loss decreases at roughly the same rate as 'Readability' and 'Coherence' increase, but much faster than when 'Similarity' increases.

From Eq. 7, it is evident that the coefficients of 'Coherence' and 'Readability' are much larger than that of 'Similarity'. It suggests that to enhance the performance of LLMs, improving the semantic consistency and grammatical correctness of the pre-training data is more effective than increasing the similarity between the pre-training data and the downstream tasks. We visualize the effect of different data attributes on loss, as shown in Figure 5. Although previous work (Yu et al., 2023; Liu et al., 2023) augment data for specific tasks such as mathematics, resulting in a substantial improvement in the model's performance on these tasks, our experimental results indicate that if the model's performance on general tasks is taken into account, enhancing the similarity between the data and the downstream tasks does not significantly improve the model's capabilities. Finally, we arrive at two conclusions: **(1) Improving the semantic and grammatical quality of pre-training data is more effective in enhancing model performance than data augmentation. (2) If data augmentation methods are used to obtain pre-training data, it is better to mix in high-quality generic data and report the model's performance on general tasks concurrently.**

6 Related Work

In recent years, LLMs have made significant strides in artificial intelligence. GPT-3 (Brown et al., 2020), which uses only a decoder structure, represents an important milestone due to its enormous model size and impressive zero-shot and few-shot in-context learning performance. InstructGPT (Ouyang et al., 2022) further improves GPT-3 by supervised fine-tuning and reinforcement learning from human feedback (RLHF), where human-annotated data is used to train the LLM. Following InstructGPT, ChatGPT (OpenAI, 2022) has gained significant attention from the public due to its remarkable text-processing capabilities. Later, GPT-4 (OpenAI, 2023) presents huge improvements over its predecessors and can process multimodal data, including both images and text.

In order to utilize and study LLMs, researchers are committed to developing open-source models since the parameters of the above LLMs are not publicly available. LLaMA (Touvron et al., 2023a) and LLaMA2 (Touvron et al., 2023b) are two representative open-source LLMs used in many downstream tasks. LLaMA2 is mainly trained on English data and is trained on more tokens than LLaMA. In response to the subpar performance of open-source LLMs on Chinese tasks, many researchers propose new LLMs for Chinese text processing. For instance, Chinese-Vicuna (Fan et al., 2023) is a low-cost Chinese dialogue model based on LLaMA. ChatGLM2 (Zeng et al., 2022) supports dialogue in both Chinese and English, which is specifically optimized for Chinese dialogue tasks. Baichuan2 (Yang et al., 2023), QWEN (Bai et al., 2023), and Yi (01-ai, 2023) have achieved commendable results on multi-disciplinary benchmarks in both English and Chinese.

7 Conclusion

In this work, we focus on the data-centric continual pre-training strategies. We propose a new data processing pipeline and use the obtained high-quality data to efficiently continue pre-training LLaMA2, ultimately resulting in Ziya2. Based on the experimental results, we explore the impact of different data attributes on LLMs and propose data-centric scaling laws.

Limitations

Due to computational resource constraints, we only conduct experiments on the LLaMA2-13B model, and we are unable to give ablation studies for the data processing pipeline. We could only use the experimental results to show that our data processing methods and training strategies are efficient. The experiments in this work are all based on LLaMA2, so the obtained conclusions including data efficiency and scaling laws are only applicable for continual pre-training. In the future, with more training resources, we hope to continue pre-training with more high-quality data on a model of larger size.

References

01-ai. 2023. Yi-34b.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. PaLM 2 Technical Report. *arXiv preprint arXiv:2305.10403*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Burton H Bloom. 1970. Space/time Trade-offs in Hash Coding with Allowable Errors. *Communications of the ACM*, 13(7):422–426.

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

Moses S Charikar. 2002. Similarity Estimation Techniques from Rounding Algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An Open-source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. *https://vicuna.lmsys.org (accessed 14 April 2023)*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

Together Computer. 2023. RedPajama: An Open Dataset for Training Large Language Models. https://github.com/togethercomputer/RedPajama-Data.

OpenCompass Contributors. 2023. Opencompass: A Universal Evaluation Platform for Foundation Models.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and Memory-efficient Exact Attention with IO-Awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Zhenyi Fan, Chenghao Lu, and Jie Tian. 2023. Chinese-Vicuna: A Chinese Instruction-following LLaMA-based Model.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *arXiv preprint arXiv:2101.00027*.

Conghui He, Zhenjiang Jin, Chao Xu, Jiantao Qiu, Bin Wang, Wei Li, Hang Yan, JiaQi Wang, and Dahua Lin. 2023. Wanjuan: A Comprehensive Multimodal Dataset for Advancing English and Chinese Large Models. *arXiv preprint arXiv:2308.10755*.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring Massive Multitask Language Understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving with the MATH Dataset. *arXiv preprint arXiv:2103.03874*.

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

Yongfeng Huang, Yanyang Li, Yichong Xu, Lin Zhang, Ruyi Gan, Jiaxing Zhang, and Liwei Wang. 2023a. MVP-Tuning: Multi-View Knowledge Retrieval with Prompt Tuning for Commonsense Reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13417–13432, Toronto, Canada.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. 2023b. C-Eval: A Multi-level Multi-discipline Chinese Evaluation Suite for Foundation Models. *arXiv preprint arXiv:2305.08322*.

Peter J. Huber. 1964. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

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

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating Training Data Makes Language Models Better. *arXiv preprint arXiv:2107.06499*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023. CMMLU: Measuring Massive Multitask Language Understanding in Chinese. *arXiv preprint arXiv:2306.09212*.

Bingbin Liu, Sebastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. 2023. Tinygsm: achieving> 80% on gsm8k with small language models. *arXiv preprint arXiv:2312.09241*.

Dong C Liu and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101*.

Junyu Lu, Ping Yang, Ruyi Gan, Jing Yang, and Jiaxing Zhang. 2022. Unified BERT for Few-shot Natural Language Understanding. *arXiv preprint arXiv:2206.12094*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2017. Mixed Precision Training. *arXiv preprint arXiv:1710.03740*.

Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling data-constrained language models. *arXiv preprint arXiv:2305.16264*.

OpenAI. 2022. Introducing ChatGPT.

OpenAI. 2023. GPT-4 Technical Report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training Language Models to Follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The Refinedweb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. *arXiv preprint arXiv:2306.01116*.

Yang Ping, JunYu Lu, Ruyi Gan, Junjie Wang, Yuxiang Zhang, Pingjian Zhang, and Jiaxing Zhang. 2023. UniEX: An Effective and Efficient Framework for Unified Information Extraction via a Span-extractive Perspective. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16424–16440, Toronto, Canada.

Han Qin, Yuanhe Tian, and Yan Song. 2021. Relation Extraction with Word Graphs from N-grams. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2860–2868, Online and Punta Cana, Dominican Republic.

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

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory Optimizations toward Training Trillion Parameter Models. In *SC20:*

International Conference for High Performance Computing, Networking, Storage and Analysis, pages 1–16. IEEE.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural Machine Translation of Rare Words with Subword Units. *arXiv preprint arXiv:1508.07909*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training Multi-billion Parameter Language Models using Model Parallelism. *arXiv preprint arXiv:1909.08053*.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced Transformer with Rotary Position Embedding. *arXiv preprint arXiv:2104.09864*.

Yuanhe Tian, Weidong Chen, Bo Hu, Yan Song, and Fei Xia. 2023. End-to-end Aspect-based Sentiment Analysis with Combinatory Categorial Grammar. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13597–13609, Toronto, Canada.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S Morcos. 2023. D4: Improving Llm Pretraining Via Document De-duplication and Diversification. *arXiv preprint arXiv:2308.12284*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. LLaMA 2: Open Foundation and Fine-tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Junjie Wang, Yuxiang Zhang, Ping Yang, and Ruyi Gan. 2022. Towards No. 1 in CLUE Semantic Matching Challenge: Pre-trained Language Model Erlangshen with Propensity-Corrected Loss. *arXiv preprint arXiv:2208.02959*.

Shaohua Wu, Xudong Zhao, Tong Yu, Rongguo Zhang, Chong Shen, Hongli Liu, Feng Li, Hong Zhu, Jiangang Luo, Liang Xu, et al. 2021. Yuan 1.0: Large-scale Pre-trained Language Model in Zero-shot and Few-shot Learning. *arXiv preprint arXiv:2110.04725*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, et al. 2023. Baichuan 2: Open Large-scale Language Models. *arXiv preprint arXiv:2309.10305*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap Your Own Mathematical Questions for Large Language Models. *arXiv preprint arXiv:2309.12284*.

Sha Yuan, Hanyu Zhao, Zhengxiao Du, Ming Ding, Xiao Liu, Yukuo Cen, Xu Zou, Zhilin Yang, and Jie Tang. 2021. Wudaocorpora: A Super Large-scale Chinese Corpora for Pre-training Language Models. *AI Open*, 2:65–68.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. GLM-130B: An Open Bilingual Pre-trained Model. *arXiv preprint arXiv:2210.02414*.

Biao Zhang and Rico Sennrich. 2019. Root Mean Square Layer Normalization. *Advances in Neural Information Processing Systems*, 32.

Jiaxing Zhang, Ruyi Gan, Junjie Wang, Yuxiang Zhang, Lin Zhang, Ping Yang, Xinyu Gao, Ziwei Wu, Xiaoqun Dong, Junqing He, et al. 2022. Fengshenbang 1.0: Being the Foundation of Chinese Cognitive Intelligence. *arXiv preprint arXiv:2209.02970*.

Terry Yue Zhuo, Yujin Huang, Chunyang Chen, and Zhenchang Xing. 2023. Exploring ai ethics of chatgpt: A diagnostic analysis. *arXiv preprint arXiv:2301.12867*.

A Details of the Pre-training Datasets

The pre-training data comes from the following datasets:

- **Pile-Pajama** is a deduplicated fusion of Pile (Gao et al., 2020) and Redpajama (Computer, 2023) datasets after removing data from Common Crawl.
- **CC** is the Common Crawl [1] data from Pile and Redpajama.
- **Wudao** (Yuan et al., 2021) is a super-large Chinese corpus containing about 3TB of training data.
- **Yuan1.0** (Wu et al., 2021) is an open-source dataset provided by Inspur Technology with more than 1TB of training data.
- **Translate** is a multilingual translation dataset collected by ourselves.
- **Code** is a code dataset we collect from GitHub, which includes multiple programming languages such as C, C++, and Python. We add the program language type before the code and change it to a format that the Markdown syntax is able to recognize.
- **Instruct** is a single-turn instructive dialogue dataset collected by ourselves.
- **Wanjuan** (He et al., 2023) is a multilingual dataset from many different sources containing more than 1TB of training data.
- **MetaMath** (Yu et al., 2023) is a dataset that achieves data augmentation by rewriting mathematical problems from multiple perspectives.

We also add some Chinese and English prompts for Instruct, Wanjuan, and MetaMath datasets, such as "*Q-A*", "*question-answer*", "*problem-solution*", etc.

B Improvements to the Structure of LLaMA2

The Ziya2 architecture is built on top of LLaMA2, and aims to enhance the quality of input data processing, token and hidden representations.

The details of these improvements are described in the following sections.

B.1 Tokenizer

To improve the preservation of semantic meaning in text and provide better support for Chinese, we have adopted a BPE (Byte-Pair Encoding) tokenizer (Sennrich et al., 2015). The vocabulary of the tokenizer includes over 600 Chinese tokens originally used in LLaMA2, as well as an additional 7,400 commonly used simplified and traditional Chinese tokens. The reason for adding extra Chinese tokens is that the original LLaMA2 vocabulary with BPE tokenizer is not efficient for Chinese, primarily due to how computers handle character encoding. For instance, in most cases, one UTF-8 Chinese character is encoded into 2-4 tokens using BPE encoding. After adding Chinese tokens to the vocabulary, testing on a 10GB Chinese corpus shows an efficiency improvement of approximately 34% in Chinese encoding and decoding compared to the original LLaMA2 tokenizer.

B.2 Positional Embedding

During continual pre-training, we observe a divergence in the distribution of text lengths between our continued pre-training dataset and the LLaMA2 pre-training dataset. This necessitates an adaptation of the position embedding to different data distributions. Meanwhile, considering downstream tasks involving lengthy texts, the scalability of position embedding is of significant importance. LLaMA2 employs rotary position encoding (Su et al., 2021), which, through the mechanism of absolute position encoding, accomplishes relative position encoding. To avoid the overflow issues associated with mixed precision, we implement rotary position encoding using FP32 precision, thereby accommodating the variation in data length distribution in continual pre-training.

B.3 Layer Normalization and Attention

We find that the direct implementation of mixed precision training in layer normalization and attention leads to precision overflow, which results in training instability. In order to maintain the efficiency and stability of model training, we propose structural improvements to layer normalization and attention in LLaMA2. For layer normalization, we utilize an APEX [2] RMSNorm (Zhang and Sennrich, 2019) implementation that operates under FP32 precision training. For attention, we employ a fused operator to replace the original scaling, mask, and softmax operators within the attention module, thereby expediting the computation of attention. To

[1] https://commoncrawl.org/

[2] https://github.com/NVIDIA/apex

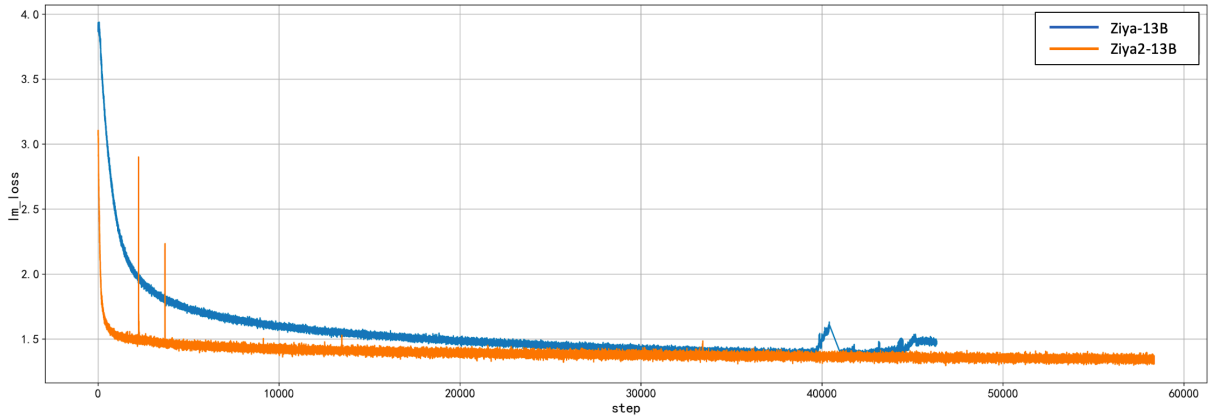

Figure 6: The pre-training loss of Ziya2-13B and Ziya-13B with respect to the number of training steps.

prevent overflow during mixed precision training in the softmax operator, it is trained using FP32 precision. These structural improvements enhance the training efficiency of the model and better adapt the training process to the instability brought about by changes in data distribution.

C Training Details

C.1 Initialization

As mentioned in Section B.1, we expand the vocabulary by 7,400 Chinese tokens. These tokens can be represented by 2-4 tokens of the vocabulary of LLaMA2. Thus, we calculate the weighted average embeddings of the corresponding tokens of the vocabulary of LLaMA2 as the initial embeddings of the new Chinese tokens.

C.2 Optimizer

The AdamW optimizer (Loshchilov and Hutter, 2017) with hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.95$ is used to train the model. Our findings indicate that incorporating additional Chinese and code data in our pre-training dataset compared to LLaMA2 results in a disparity in the overall data distribution. To address this issue, we propose a more extended warmup for continual pre-training. Specifically, we adopt a warmup ratio of 1% instead of the 0.4% warmup ratio utilized in LLaMA2. This is followed by a cosine learning rate decay schedule, which reaches a final learning rate of $1e^{-5}$. We also implement a weight decay of 0.1 and gradient clipping set at 1.0 to enhance the training efficiency of the model.

C.3 Training Frameworks

We have employed Megatron (Shoeybi et al., 2019) and DeepSpeed (Rasley et al., 2020) as foundational training frameworks. Megatron enables distributed training of large-scale models through data parallelism, tensor parallelism, and pipeline parallelism. We also utilize the ZeRO technology (Rajbhandari et al., 2020) from DeepSpeed to optimize memory savings. We trained the model with 80 NVIDIA A100-80G GPUs, setting the value of model parallelism to 1 and pipeline parallelism to 2. We have implemented multiple advanced techniques such as flash-attention (Dao et al., 2022) and fused-softmax to enhance training efficiency. As a result, each GPU achieves an industry-leading efficiency of 163.0 TFLOPS per second.

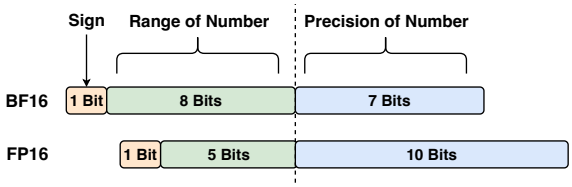

Figure 7: An illustration of the representations of floating numbers by BF16 and FP16.

C.4 Training Stability

We notice that when using the FP16 mixed precision (Micikevicius et al., 2017) to train the Ziya-13B model, a loss spike occurs in the later stages of training. We find that the limited numerical range of FP16 leads to overflow problems, especially in operations such as softmax. Compared to FP16, BF16 has a larger representation range, as shown in Figure 7. Hence, we opt for BF16 mixed-precision training for the continual pre-training of Ziya2-13B. We also fix some underlying bugs in DeepSpeed to address the loss spike problems. As a result, our model achieves convergence even when training continues for 700B tokens, as shown in Figure 6.

D Evaluation Details

We use the official evaluation scripts in OpenCompass (Contributors, 2023) to compute the accuracy and perform 5-shot evaluations. The details of the evaluation datasets are shown as follows:

- **MMLU** (Massive Multitask Language Understanding) (Hendrycks et al., 2020) provides a comprehensive evaluation of models in both zero-shot and few-shot settings, spanning across a wide range of 57 subjects. The unique aspect of MMLU is that it tests the models' world knowledge and problem-solving capabilities.
- **CMMLU** (Chinese Massive Multitask Language Understanding) (Li et al., 2023) is an extensive evaluation suite tailored to measure the advanced knowledge and reasoning skills of LLMs in the context of the Chinese language and culture. It contains a broad spectrum of 67 topics, ranging from basic to advanced professional levels.
- **C-Eval** (Chinese Evaluation) (Huang et al., 2023b) is a thorough evaluation suite specifically designed for foundational models in Chinese. It is composed of 13,948 multiple-choice questions that cover a wide array of 52 different disciplines and span across four levels of difficulty.
- **GSM8K** (Grade School Math 8K) (Cobbe et al., 2021) is a collection of 8,500 high-quality math word problems created by human writers. The objective of this dataset is to facilitate the task of question answering on fundamental problems that necessitate reasoning through multiple steps.
- **MATH** (Hendrycks et al., 2021) aggregates 12,500 intricate competition mathematics problems. A unique feature of this dataset is that each problem comes with a comprehensive step-by-step solution. These detailed solutions serve as a valuable resource for teaching models to generate derivation processes and explanations.
- **HumanEval** (Chen et al., 2021) is a meticulously constructed set of 164 programming challenges. This dataset serves as a benchmark for evaluating the ability of a system to generate functionally correct programs based on provided doc strings. The challenges encompass a range of topics, including language comprehension, algorithmic problems, and basic mathematics.

E Baseline Details

We choose the following representative LLMs as baselines:

- **ChatGLM2-6B** (Zeng et al., 2022) is developed by Tsinghua University and Zhipu AI. The model is the second generation of the bilingual dialogue model ChatGLM-6B.
- **Falcon-7B** (Penedo et al., 2023) is a causal decoder-only model developed by Technology Innovation Institute (TII). It's trained on 1,500 billion tokens of RefinedWeb.
- **Vicuna-13B** (Chiang et al., 2023) is an open-source LLM developed by the Language Model Systems (LMSYS) organization and is fine-tuned from LLaMA on user-shared conversations between humans and ChatGPT.
- **Baichuan2-13B** (Yang et al., 2023) is developed by Baichuan Intelligent Technology and is trained with 2.6 trillion tokens.
- **XVERSE-13B**[3] is developed by XVERSE Technology and is trained with 1.4 trillion tokens.
- **WeMix-13B**[4] is developed by Shanghai AI lab and is built on LLaMA2-Accessory.
- **LLaMA2-13B** (Touvron et al., 2023b) is Meta's open-source large language model. It is designed for dialogue scenarios and is freely available for research and commercial use.
- **Ziya-13B** (Zhang et al., 2022) is continue pre-trained on the LLaMA-13B model and performs well on many downstream tasks.
- **GPT-3.5-turbo** is a high-performance variant of GPT-3 and it is proficient in text completion.
- **GPT-4** (OpenAI, 2023) is the latest state-of-the-art LLM developed by OpenAI. It exhibits human-level performance on various professional and academic benchmark datasets.

F Evaluation Metrics

We use several metrics to evaluate data quality, as shown in Table 5. We also give the criteria for human annotations in Figure 8.

G Fitting Details

For 'Coherence' and 'Readability', we extract 1,000 pieces of data from each of the stages for human evaluation. The results are then normalized to a range of [0,1]. For 'Similarity', we randomly extract 100,000 pieces of data from each of the stages and compute their embeddings, respectively. We use text-embedding-ada-002 from OpenAI to

[3] https://github.com/xverse-ai/XVERSE-13B.
[4] https://github.com/Alpha-VLLM/WeMix-LLM.

Annotation criteria for pre-training data quality

Please annotate the given text according to the following three metric, and mark the problems found in the text in red.

1. **Coherence:** Does the given text contain meaningless information such as advertisements, URL navigation bar? You also need to determine if the multiple pieces of information contained in this text are not related to each other. If yes, mark it as 1, otherwise mark it as 0.

Example1: Established on March 1, 1985, Institute of Software, Chinese Academy of Sciences (ISCAS) is a comprehensive base-type institute dedicated to the research and development of computer science theory and software high technology. Due to the need of work, we are now recruiting a supervisor of cultivation and degree work. Your E-mail*: cannot be empty! Please enter a valid E-mail address!

Explanation: You should mark this example as 1. Text marked in red is not part of the body of the page.

Example2: Hello and welcome to 168Flora.com! Deciduous large trees, up to 30 m high, up to 1 m in diameter at breast height; bark gray-brown, scale-like; branchlets lenticellate, stout, extended.

Explanation: You should mark this example as 1. The first sentence is an advertisement.

Example3: “人间若有天堂，大马士革必在其中。天 [… 最近网上“佛系”一词很火， 各种佛系头像 [… 地狱未空，誓不成佛， 众生度尽，方证菩提 [… 一场接一场的秋雨， 令苏城的温度又降了些 […

Explanation: You should mark this example as 1. The four pieces of information in the content are irrelevant.

Example4: 前阵子在网路购物时看到【书宝二手书 T1 / 一般小说_IPL】赤壁之战_张云风的商品，发现品质还不错，很多口碑文章推荐，但书宝二手书店:【书宝二手书 T1 / 一般小说_IPL】赤壁之战_张云风哪里买比较划算呢？【书宝二手书 T1 / 一般小说_IPL】赤壁之战_张云风在 PTT 上面说这个网站便宜又快速，所以我都到这个网站购买【书宝二手书 T1 / 一般小说_IPL】赤壁之战_张云风了！

Explanation: You should mark this example as 1. The text contains a lot of URLs.

2. **Readability:** Whether the given text has grammatical errors or punctuation errors. If yes, mark it as 1, otherwise mark it as 0.

Example1: According to Nicola Arthur, head of the research team, "We tried to find a different way of treating cancer from the usual methods, and eventually found one that doesn't kill cancer cells but stops them from...

Explanation: You should mark this example as 1. The sentence is incomplete and should not use an ellipsis.

Example2: 巨人游戏载入后，点击“跳过”按钮的底部，并从主菜单中，单击“开始游戏”。有两种不同的游戏模式：街机与挑战。在街机模式中，你有一分钟，可以跳过你想要的，并尽可能多的机器人，你可以销毁。的游戏，用鼠标控制。拧紧鼠标左键，建立实力雄厚的跳。

Explanation: You should mark this example as a 1. The text contains numerous grammatical errors and punctuation errors.

3. **Toxic:** whether the given text contains terrorist, violence, or pornographic information. If yes, mark it as 1, otherwise mark it as 0.

Example1: 前，用力的一吸。啊。清醇的声音情不自禁的溢出。圣儿真是个诚实的孩子呢。龙焱寒毫不客气的表扬，但是圣儿的叫声似乎还不够哦。语毕，唇再一次的吸住了东城凤胸前的红点，湿润的舌头开始绕着红点挑逗，另一只手同时轻柔着捏着东城凤的另一边。吟嗯。急促的呼吸声夹着舒服的呻吟声音陆陆续续的溢出。吟舒舒服。纯真的人儿用最动听的声音在表达自己的意愿。

Explanation: You should mark this example as 1. The text contains pornographic information.

Figure 8: The criteria for human annotations.

Metric	Method	Level	Comments
Privacy Word Rate	Re Matching	1	Includes email addresses, phone numbers and ect.
Toxic Word Rate	Re Matching	1	Checking political, explicit, violent, or similar content.
Adv Word Rate	Re Matching	1	Checking for the presence of advertising keywords.
Webpage Funcword Rate	Re Matching	1	Detecting the presence of HTML tags.
Perplexity	LM	2	Whether the sentences in the article are coherent.
Informativeness	LM	2	Assessing the information content within the text.
Readability	Counts	2	Assessing text readability
Language	LM	3	Analyzing the distribution of languages.
Doc Length	Counts	3	Evaluating the distribution of text lengths.
Topic	Statistics	3	Detecting the topic distribution of the text.

Automatic Evaluation

Metric	Method	Level	Comments
Coherence	Human Check	1	Assessing consistency and usefulness.
Readability	Human Check	1	Assessing grammatical and formatting correctness.
Toxic	Human Check	1	Checking political, explicit, violent, or similar content.

Human Evaluation

Table 5: The metrics we utilize for evaluating data quality. The **Method** denotes our detection method, **Re Matching** signifies the method involving regular expressions for counting. **LM** represents the utilization of a language model for predicting relevant metrics. **Counts** indicates the use of statistical methods for directly countable metrics. **Human Check** indicates manual spot-checking conducted by humans. **Level** indicates the degree of stringency we apply to the respective metrics. We consider it compliant if it does not exceed one in a thousand.

calculate the embeddings. Subsequently, we calculate the embeddings for the test sets of six benchmarks in Section 4.1. The average cosine similarity between them is used as the value for 'Similarity', with a range of [0,1].

The L-BFGS algorithm starts on a grid initialization given by: $\alpha \in \{0, 0.67, 1.33, 2\}$, $A \in \{0, 0.29, ..., 1.71, 2\}$, $B \in \{0, 0.29, ..., 1.71, 2\}$, $C \in \{0, 0.29, ..., 1.71, 2\}$, $E \in \{0, 0.67, 1.33, 2\}$, $F \in \{1, 1.67, 2.33, 3\} * 10^{12}$.

H Pre-training Data

The following charts presents examples of unsupervised and supervised training data.

English Unsupervised Data

Text

ABC News' Good Morning America outstripped NBC News' Today by 761,000 viewers and 279,000 news demo viewers the week of April 7. It's GMA's seventh consecutive week on top of the morning infotainment show race in both metrics, and its largest demo margin in three months. GMA has ranked No. 1 in overall audience for 89 of the past 93 weeks, and No. 1 in the news demo for 25 of this season's 29 weeks to date. Today meanwhile, boasted it finished first with the younger, 18-49 year old age bracket, for the 42nd consecutive week. Today is on top of the ratings in the day part with men 25-54 this season, NBC noted — as well as adults, men and women 18-49. Today has posted seven consecutive months of ratings growth in total viewers, and both the 25-54 and 18-49 demos which NBC says is the show's biggest ratings uptick since '97. For the week, GMA clocked 5.617 million viewers — 2.212 million in the demo. Today logged 4.856 million viewers — 1.933 million in the demo. GMA bested CBS This Morning's 3.041 million viewers — 956,000 in the news demo.

Chinese Unsupervised Data

Text

你走了，带着我们还来不及开始的爱。你说我不够勇敢与果断、说我只是把喜欢寄放在你那边却从来没有想细心灌溉。但我要怎么能灌溉呢？你就像是一沃富饶的土地，大家总把他们的爱埋在你的心窝。圳水引来的滋润来不及洒落在我需要你的那些岁月，我向着有你在的阳光趋去，却如伊藤润二笔下的漩涡，只不过成了让人害怕的扭曲怪状。***{...}*** 所以谢谢我曾经在你心里的位置那么前面、那么靠近你心的地方。

You're gone, with the love we haven't had time to start yet. You say I am not brave and decisive enough, and you say I just place my love on your side but never want to irrigate it carefully. But how can I irrigate it? You are like a fertile land, where everyone always buries their love in your heart. The nourishment brought by the water in the canal did not fall on the years when I needed you. I moved towards the sunshine with you, but like a vortex in the pen of Runji Ito, it only became a frightening twisted and strange shape ***{...}*** *So thank you for being so in front of you and so close to your heart.*

English Supervised Data

Text

Problem:
*Human rights are moral principles or norms, which describe certain standards of human behaviour****{...}*** *Military aircraft can be either combat or non-combat: TurboJET is the brand name for the operations of the Hong Kong-headquartered Shun Tak-China Travel Ship Management Limited, which was established from the joint venture between Shun Tak Holdings Limited and China Travel International Investment Hong Kong Limited in July 1999. It operates hydrofoil ferry services in southern China. Question:use beechcraft starship*
A.military
B.general aviation
C.service
Answer:B.general aviation

Chinese Supervised Data

Prompt

问题:{问题和内容} 回答:{回答的内容}

Question:{the question} Answer:{the answer}

Text

问题:
请问下面描述属于哪一种事件类型？文章：昨晚，在号称“亚洲第一魔鬼主场”的天河体育场，国足在占据天时地利人和的情况下，半场就击垮对手，狂轰*6*球***{...}*** 这样的战绩和表现也在赛后引发了巨大的争议。
*A.*胜负
*B.*加息
*C.*融资
*D.*爆炸
回答：答案：*A.*胜负

Question:
May I ask which type of event does the following description belong to? Article: Last night, at the Tianhe Stadium, which is known as the "Number One Devil's Stadium in Asia," the Chinese football team occupied Under the favorable conditions of timing, location, and people, he defeated his opponent in half time and roared 6 goals ***{...}*** *Such achievements and performances have also sparked great controversy after the game.*
A.Success or Failure
B.Raise Interest Rates
C.Financing
D.Explosion
Answer: The answer is：A.Success or Failure

Code

Prompt

```` (sql / python / c / etc.) + {Code} ```

Text

```` *SQL*
***ALTER TABLE*** *kooste.tamperewfs_luontorastit* ***DROP CONSTRAINT*** *tamperewfs_luontorastit_pk;*
***ALTER TABLE*** *kooste.tamperewfs_luontorastit* ***DROP CONSTRAINT*** *tunnus_rasti_unique;*
***ALTER TABLE*** *kooste.tamperewfs_luontorastit* ***ADD COLUMN*** *kooste_id bigint* ***NOT NULL*** *GENERATED ALWAYS* ***AS IDENTITY****;...{Code~}*
```

## MetaMath

### Prompt

Question/Q:{the question} Answer/A:{the answer}

### Text

*Question:*
*The coffee shop sold 8 more than four times the number of lattes as it did teas. If 6 teas were sold how many lattes did the coffee shop sell?*
*Answer:*
*If the coffee shop sold 6 teas, then the number of lattes sold is 8 more than four times the number of teas. Four times the number of teas is 4 * 6 = 24. So, the number of lattes sold is 24 + 8 = 32. The answer is: 32*
```

# I Generation Cases

The following charts presents examples generated by Ziya2 with particular inputs.

## Chinese Generation

模型输入: 我国的三皇五帝分别指的是谁?

Model Input: *Who are the "Three Emperors and Five Sovereigns" of our country?*

### Ziya2 Generation

"三皇五帝"是指我国古代部落首领，其中"三皇"一般认为指的是燧人、伏羲、神农；而"五帝"一般指的是黄帝、颛顼、帝喾、尧、舜。

*The "Three Emperors and Five Sovereigns" refer to ancient tribal leaders in China, among which the "Three Emperors" are generally believed to refer to Suiren, Fuxi, and Shennong; The "Five Sovereigns" generally refer to Huang Di, Zhuanxu, Sovereigns Ku, Yao, and Shun.*

## English Generation

Model Input: *Friends is a series of sitcoms produced by NBC in the United States,*

### Ziya2 Generation

*and created by David Crane and Marta Kauffman. The series premiered on September 22, 1994, and ended on May 6, 2004, lasting ten seasons. With an ensemble cast starring Jennifer Aniston, Courteney Cox, Lisa Kudrow, Matt LeBlanc, Matthew Perry and David Schwimmer, the show revolves around six friends in their 20s and 30s who live in Manhattan, New York City. The series was produced by Bright Kids Productions, in association with Warner Bros. Television. The original executive producers were Kevin S. Bright, Marta Kauffman, and David Crane. Friends had received both critical acclaim and popular success throughout its original ten-season run on NBC; it received accolades such as winning the Emmy Award for Outstanding Comedy Series in 2002, and was ranked No. 25 on TV Guide's 50 Greatest TV Shows of All Time in 2002.*

## Code Generation

模型输入: 给定一个整数*n*，计算所有小于等于*n* 的非负整数中数字*1* 出现的个数。例如:
输入：*n = 13*
输出：*6*
分析如下：输入*n=13*，小于等于*13*非负整数里面包含*1*的数字为*1*，*10*，*11*，*12*，*13*，所以数字*1*共出现了*6*次。下面是具体的*python*代码：```*python*

Model Input: *Given an integer n, calculate the number of occurrences of the number 1 in all non negative integers less than or equal to n. For example:*
*Input: n=13*
*Output: 6*
*The analysis is as follows: input n=13, and the number containing 1 in a non negative integer less than or equal to 13 is 1, 10, 11, 12, 13, so the number 1 appears together Six times. The following is the specific Python code:* ``` *Python*

### Ziya2 Generation

```python
import re
    def countDigitOne(self, n: int) -> int:
        count = 0
        for i in range(1, n + 1):
            s = str(i)
            for x in s:
                if x == '1':
                    count += 1
    return count
```