Enhancing LLMs via High-Knowledge Data Selection

Feiyu Duan[1,5*], **Xuemiao Zhang**[2,5*], **Sirui Wang**[3,5†], **Haoran Que**[1], **Yuqi Liu**[5], **Wenge Rong**[4†], **Xunliang Cai**[5]

[1]Sino-French Engineer School, Beihang University, Beijing, China
[2]Peking University, Beijing, China
[3]Department of Automation, Tsinghua University, Beijing, China
[4]School of Computer Science and Engineering, Beihang University, Beijing, China
[5]Meituan, Beijing, China
{duanfeiyu, 2224124, w.rong}@buaa.edu.com, zhangxuemiao@pku.edu.cn, {liuyuqi, wangsirui, caixunliang}@meituan.com

Abstract

The performance of Large Language Models (LLMs) is intrinsically linked to the quality of its training data. Although several studies have proposed methods for high-quality data selection, they do not consider the importance of knowledge richness in text corpora. In this paper, we propose a novel and gradient-free **H**igh-**K**nowledge **S**corer (HKS) to select high-quality data from the dimension of knowledge, to alleviate the problem of knowledge scarcity in the pre-trained corpus. We propose a comprehensive multi-domain knowledge element pool and introduce knowledge density and coverage as metrics to assess the knowledge content of the text. Based on this, we propose a comprehensive knowledge scorer to select data with intensive knowledge, which can also be utilized for domain-specific high-knowledge data selection by restricting knowledge elements to the specific domain. We train models on a high-knowledge bilingual dataset, and experimental results demonstrate that our scorer improves the model's performance in knowledge-intensive and general comprehension tasks, and is effective in enhancing both the generic and domain-specific capabilities of the model.

1 Introduction

The impressive performance of large language models (LLMs) has been demonstrated in various natural language processing (NLP) tasks (Touvron et al. 2023; OpenAI 2023), yet it is significantly influenced by the quality of the training data (Li et al. 2023; Xie et al. 2024a). Typically, the training data is sourced from extensive text collections such as internet crawls (Patel 2020). However, the quality of such data is frequently inconsistent (Kreutzer et al. 2022). Therefore, the identification and selection of high-quality data from these vast resources is a critical consideration for achieving optimal model training.

To effectively address this issue, a sub-problem that needs to be solved first is how to define high-quality data. Current methodologies typically adopt one of two approaches: the first involves devising a metric to assess the quality of the text, focusing on attributes such as textual fluency (Marion et al. 2023; Muennighoff et al. 2024); the second approach involves manually curating a subset from the available corpus, which is considered high-quality based on human experience, to serve as a standard reference, with Wikipedia articles often being chosen for this purpose (Xie et al. 2024b; Engstrom, Feldmann, and Madry 2024). Techniques derived from the first approach include strategies to eliminate redundant data (Lee et al. 2022) or employing existing models to compute metrics like perplexity (PPL) (Marion et al. 2023) or self-influence scores (Thakkar et al. 2023) for the dataset. On the other hand, the latter includes the development of models to assign quality scores (Brown et al. 2020) to data or the determination of significance weights (Xie et al. 2024b) to guide the sampling process.

In practice, the mentioned selection criteria often favor fluent texts that may lack knowledgeable information, as shown in the Appendix Figure 9. This observation inspires us to propose a novel approach: **H**igh **K**nowledge **S**corer (HKS), which assesses text quality by detecting the knowledge content of each text sample. We begin by quantifying the knowledge encapsulated within the texts of training data. Unlike the structured knowledge definitions in Allen-Zhu and Li (2024), we simplify knowledge into *knowledge elements* to facilitate quicker knowledge annotation. Following this, we create a multi-domain knowledge element pool consists of 5M knowledge elements from various sources. We employ a multiple pattern matching algorithm to identify all the knowledge elements contained in each text sample and introduce two metrics, *knowledge density* and *knowledge coverage*, to facilitate quantitative analysis. Leveraging these metrics, we propose a comprehensive knowledge scorer to select high-knowledge texts, which are positively correlated with these two metrics. Benefiting from the categorization of each knowledge element during the creation of the multi-domain knowledge element pool, our approach can select high-knowledge data aligned to the desired domain.

We train a 1.1B model from scratch on a 20B bilingual dataset. We find that: (1) In comparison to baseline methods, our method excels both in knowledge-intensive tasks and general understanding tasks, with an average improvement of 2.37 pp over random selection. We further validate

[*]These authors contributed equally to this work and should be considered as co-first authors.
[†]Corresponding authors.

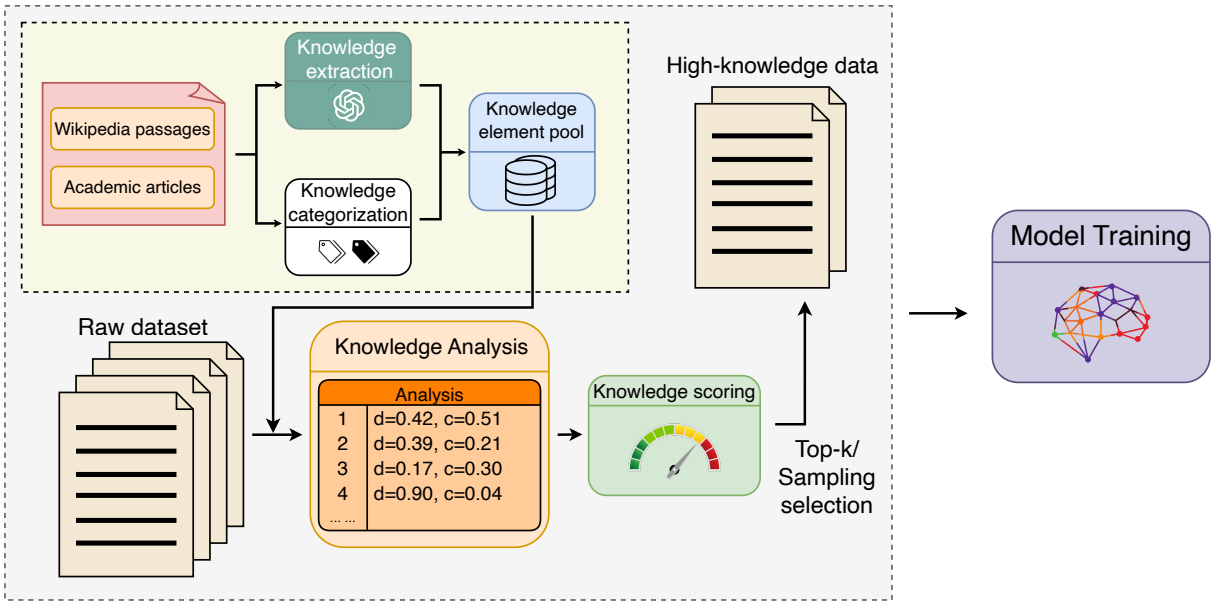

Figure 1: The overall framework of HKS. Our methodology begins with sourcing knowledge from Wikipedia articles and academic literature. We extract knowledge elements, which are categorized into several domains. Each text sample from the raw dataset is then characterized by its knowledge density and coverage, resulting in a knowledge score for each text. Texts with higher scores are identified as high-knowledge data, selected via top-k selection or weighted sampling.

this conclusion through continual pretraining experiments on Llama-3-8B, observing an average increase of 2.4 pp. (2) When applied to specific domains, our method can enhance the model's performance in the desired domain, with an absolute improvement of up to 2.5 pp compared to baseline.

Our contributions can be summarized as follows:

- We simplify the definition of knowledge and introduce *knowledge element* to facilitate knowledge parsing of texts, which guides us to establish a multi-domain knowledge element pool. Moreover, we propose knowledge density and coverage as metrics to quantify the knowledge in texts.
- We propose a novel and gradient-free knowledge scorer for selecting high-knowledge data, which is comprehensive and proportional to knowledge density and coverage. A series of experiments consistently showcase the superior performance of our knowledge scorer.
- We propose a domain-aware high-knowledge data selection method for domain-specific augmentation. Experimental results demonstrate the effectiveness of our approach.

2 Methodology

2.1 Overview of Our Approach

Figure 1 illustrates the pipeline of our method. Initially, we used Wikipedia and academic literature as knowledge sources. Knowledge elements are extracted and classified using GPT4 (Achiam et al. 2023) and a BERT-based model. For each text in the pre-training dataset, we enumerate all the included factual knowledge elements, which are then used to determine the two knowledge metrics: density and coverage. A knowledge scorer, proportional to these two metrics, assigns comprehensive scores to each text, which is further considered as data selection criteria.

2.2 Knowledge Element Pool Construction

Knowledge is usually represented as structured (name, relation, value) triplets in knowledge graphs (Zhu and Li 2023; Pan et al. 2024), but parsing triplets across the entire corpus is extremely time-consuming and challenging. To simplify this process, we reduce knowledge triplet to *knowledge element*:

Definition 1 *A n-gram noun, represented by several tokens* $(t_1, t_2, \ldots, t_n)$*, can be considered as a* ***knowledge element*** *if it encapsulates or is closely associated with a specific concept, fact, theory, principle, definition, and so forth.*[1]

Based on the above definition, we build a knowledge element pool that covers multiple domains. The knowledge elements are derived from two sources: 1) *Wikipedia documents* 2) *Academic article datasets*. We first add Wikipedia entries and keywords of academic articles to the knowledge element pool, where the academic dataset we use is the OAG dataset (Zhang et al. 2022). The knowledge elements obtained amount to a total of 20M. While these elements are highly specialized, they lack flexibility. Therefore, we also choose to write instructions to guide GPT4 in extracting knowledge elements contained in the Wikipedia documents[2]. The prompt is listed in Appendix C.

We categorize the knowledge elements into five domains: *science, society, culture, art, and life*. Given that individual

[1]We only use the names rather than their contents.

[2]GPT-4 extracted knowledge elements comprise 13.9% of the entire pool.

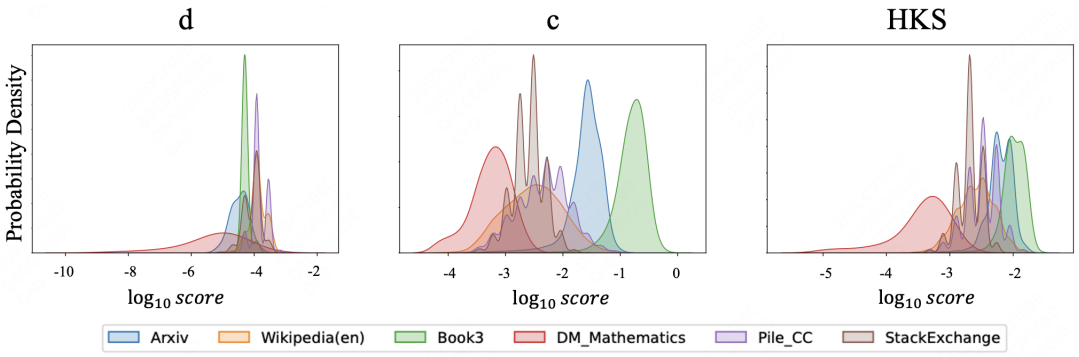

Figure 2: Score density in Pile subsets. There is a noticeable difference in the distribution of knowledge density (d) and coverage (c) within Pile subsets. d tends to favor samples from Wikipedia, while c tends to favor samples from books and ArXiv.

knowledge elements are difficult to label directly, we match related documents for each knowledge element as auxiliary materials for judgment. Then, we manually annotate 10K knowledge elements and take 20% of the annotated data as the test set. We train a BERT-based (Devlin et al. 2019) labeling model with a labeling accuracy of 96.2%.

After knowledge categorization, we write an additional instruction and use GPT4 to review all the knowledge elements, filtering out those that do not align with the annotated categories or are of poor quality. Additionally, we remove knowledge elements with a string length of less than 2. After deduplication, we finally built a high-quality knowledge element pool containing 5M terms. More detailed construction process and statistics can be found in Appendix E.

2.3 Knowledge Content Evaluation within the Text

For each text in the training data, we label all the knowledge elements that appear in it[3]. For quantitative analysis of the pre-training corpus, we establish the following two knowledge metrics.

Definition 2 *Given a text sample x, n_k is the total number of knowledge elements in the sample, and n_p is the text token length,* ***Knowledge Density (d)*** *is defined as $d(x) = n_k/n_p$.*

Definition 3 *Given a text sample x, $\widetilde{n}_k$ is the total number of non-duplicated knowledge elements in the sample, and N_k is the total number in the full knowledge pool.* ***Knowledge Coverage (c)*** *is defined as $c(x) = \widetilde{n}_k/N_k$.*

Knowledge density is used to quantify the amount of knowledge contained in a text sample. We do not use the total token count of the knowledge elements as n_k to avoid any bias arising from longer knowledge elements. Knowledge coverage serves as an indicator of the diversity of knowledge within a text sample.

We compute two metrics for the texts in the Pile dataset, and present cases in Appendix Figure 9. After separately selecting the top 1M samples from the Pile dataset based on density and coverage, we observe that articles with high knowledge density tend to have an average length of 20,950 tokens, whereas those with high knowledge coverage are typically much shorter, averaging only 811 tokens. This suggests that d and c are two significantly distinct metrics. To examine the observations, we analyze the distribution of d and c across various subsets within the Pile. We select several subsets and draw 10K random samples from each subset. The samples are then divided into buckets based on the minimum and maximum values of the metrics, and we count the number of samples falling into each bucket. Figure 2 shows the results, revealing distinct preferences of d and c across different subsets. Notably, d exhibits a preference for subsets such as FreeLaw and Wikipedia, whereas c prefers Arxiv.

[3]We use the Aho-Corasick automaton algorithm (Pao, Lin, and Liu 2010) to accelerate our labeling process.

2.4 Knowledge-Based Data Selection

Generic high-knowledge scorer According to the aforementioned results, we decide to combine these two orthogonal metrics to create a comprehensive knowledge scorer. Specifically, for a text sample x, we materialize the scorer as a scoring function:

$$score(x) = \phi(d(x), c(x)) \tag{1}$$

As we have seen in the extraction results (Section 2.3), these two metrics lead to two completely different extraction results with less entanglement, so we assume that d and c are two variables independent of each other, and we can simplify the function ϕ to a product form:

$$score(x) = f(d(x)) \cdot g(c(x)) \tag{2}$$

For f and g, we give some empirical assumptions:

1. f and g should be incremental functions, as we suggest that texts abundant in knowledge information generally yield high scores on knowledge density and coverage.
2. When the values of d and c are high, their incremental contributions to the overall effect are not expected to be as significant as when their values are low. This implies that the functions f and g do not exhibit upward concavity:

$$\frac{\partial^2 f}{\partial d^2(x)}(d(x)) \leq 0, \frac{\partial^2 g}{\partial c^2(x)}(c(x)) \leq 0 \tag{3}$$

We have experimented with various combinations of functions, and the details can be found in Appendix D. The scoring formula that we ultimately chose is as follows:

$$score(x) = d(x) \cdot ln(c(x) + 1) \tag{4}$$

The selection cases through our knowledge scorer are displayed in Appendix Figure 9 and 10. Besides, we also analyze the score distribution in Pile. The results in Figure 2 show that samples from Book3, Arxiv, and FreeLaw achieve higher HKS scores compared to those from DM Mathematics.

Domain-specific knowledge scorer Given that our knowledge elements are categorized, we are able to perform domain-specific knowledge scoring and select high-knowledge data belong to that domain, thereby achieving domain-specific enhancement. Specifically, we constrain our knowledge elements into the target domain m, therefore obtaining the knowledge density d_m and coverage c_m for specific domain[4]. Similar to the generic knowledge scorer, we can evaluate each text using a domain-specific scoring function for domain m:

$$score_m(x) = d_m(x) \cdot ln(c_m(x) + 1) \tag{5}$$

Filtering strategies After scoring each text in the pre-training dataset, we adopt two methods for high-knowledge data selection.

- *Top-k*: We select the top-k text samples based on our defined scores, with some selection cases displayed in Appendix Figure 9 and 10. It is evident that texts with high scores are more knowledgeable and of higher quality, while samples with low scores contain less knowledge content.
- *Sampling*: In addition to the top-k selection technique, various studies have highlighted the efficacy of sampling methods (Sachdeva et al. 2024; Wettig et al. 2024). In our research, we employ softmax-based sampling strategies: We treat the normalized score as an importance weight, and apply the softmax function to each sample x_i in the pre-training dataset to calculate sampling probability:

$$P(x_i) = \frac{exp(\frac{score_i}{\tau})}{\sum_j exp(\frac{score_j}{\tau})} \tag{6}$$

  τ is the temperature term. We perform sampling without replacement and utilize the Gumbel top-k trick (Kool, Van Hoof, and Welling 2019) to facilitate the sampling process. Here we choose $\tau = 2$.

3 Experiments

3.1 Setups

Dataset We utilize the Pile (Gao et al. 2020) and Wudao (Yuan et al. 2021) datasets as our pre-training dataset for training a bilingual language model. Pile is an extensive English text corpus that includes 22 diverse subsets, while Wudao consists of Chinese passages collected from web sources. We extract 10B tokens from each, resulting in a 20B token bilingual dataset, which aligns with the compute-optimal amount from previous studies (Hoffmann et al. 2022).

[4]See Appendix E for detailed definition.

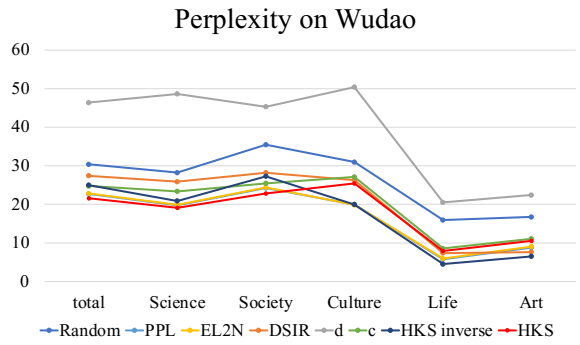

Figure 3: Perplexity evaluation on Wudao validation dataset.

Model and training We train a model of 1.1B parameters, which has the same architecture of Bloom (Le Scao et al. 2023). We train our model in one epoch, with a cosine learning rate scheduler. We use a global batch size of 2048 with gradient accumulation and a max context window length of 2048. We use Megatron framework to train our model in 16 A100 GPUs, with fp16 setting, which needs 21 hours to finish our training. More details can be found in Appendix F.

Baselines We compare our method with the following baselines: (1) *Random*: Data is selected randomly from each source dataset with uniform probability. (2) *density d and coverage c*: We utilize density d and coverage c as the criteria for data selection, respectively. (3) *HKS inverse*: In our ablation studies, we conduct experiments where we select the texts with the lowest scores, as determined by the HKS, to train the model. (4) *Perplexity (PPL)*: In line with the methodology outlined in Marion et al. (2023), we employ Bloom 1.1B model (Le Scao et al. 2023) to calculate the perplexity (PPL) for each data sample. We then retain the top-k samples exhibiting the lowest PPL values. (5) *Error L2-Norm (EL2N)*: Addition to perplexity, we also calculate the error l2-norm for each data sample (Marion et al. 2023). We then retain the top-k samples exhibiting the lowest EL2N values. (6) *Data Selection via Importance Resampling (DSIR)*: Following the methodology outlined in Xie et al. (2024b), we use bigram representation to compute hash features for each sample. We utilize documents from Wikipedia as the target domain to compute the importance weight.

Benchmarks and metrics We train all the models three times using different random seeds and report the average results. We conduct a holistic assessment of all the methods:

- We measure perplexity on both Pile and Wudao datasets. For the Pile dataset, we extract 10K samples from each subset to serve as a validation dataset, ensuring these samples are not encountered during the training process. Since the Wudao dataset does not have a predefined subset split, we divide it according to categories of the included knowledge elements. We then apply the same validation process as with the Pile dataset, extracting samples for evaluation.
- We assess downstream task performance using in-context learning (Dong et al. 2022). For knowledge-intensive tasks, we conduct evaluations on ARC-C (Bhakthavatsalam et al.

Method		English Tasks		Chinese Tasks		AVG.
		Knowledge intensive	General understanding	Knowledge intensive	General understanding	
Random		$23.77_{0.06\ +0.00}$	$47.56_{0.19\ +0.00}$	$25.15_{0.28\ +0.00}$	$28.66_{0.14\ +0.00}$	$32.49_{0.10\ +0.00}$
PPL		$24.44_{0.07\ +0.67}$	$45.92_{0.28\ -1.64}$	$24.46_{0.20\ -0.69}$	$22.99_{0.06\ -5.67}$	$29.41_{0.25\ -3.08}$
EL2N		$24.52_{0.18\ +0.75}$	$48.23_{0.22\ +0.67}$	$26.11_{0.10\ +0.96}$	$24.34_{0.10\ -4.32}$	$30.84_{0.23\ -1.65}$
DSIR		$19.26_{0.20\ -4.51}$	$48.12_{0.03\ +0.56}$	$25.61_{0.07\ +0.46}$	$23.97_{0.23\ -4.69}$	$29.75_{0.02\ -2.74}$
d	Sampling	$24.34_{0.09\ +0.57}$	$48.04_{0.27\ +0.48}$	$25.46_{0.27\ +0.31}$	$26.34_{0.15\ -2.32}$	$31.64_{0.14\ -0.85}$
	Top-k	$21.72_{0.21\ -2.05}$	$43.75_{0.20\ -3.81}$	$24.79_{0.01\ -0.36}$	$24.39_{0.23\ -4.27}$	$29.11_{0.11\ -3.38}$
c	Sampling	$23.70_{0.29\ -0.07}$	$43.08_{0.29\ -4.48}$	$26.75_{0.14\ +1.60}$	$26.71_{0.29\ -1.95}$	$30.55_{0.18\ -1.94}$
	Top-k	$25.27_{0.29\ +1.50}$	$43.05_{0.05\ -4.51}$	$26.21_{0.01\ +1.06}$	$27.45_{0.27\ -1.21}$	$31.08_{0.20\ -1.41}$
HKS	Inverse	$20.29_{0.00\ -3.48}$	$46.12_{0.11\ -1.44}$	$26.55_{0.25\ +1.40}$	$25.21_{0.21\ -3.45}$	$30.08_{0.28\ -2.41}$
	Sampling	$24.89_{0.11\ +1.12}$	$43.61_{0.22\ -3.95}$	$26.74_{0.19\ +1.59}$	$26.98_{0.24\ -1.68}$	$31.00_{0.02\ -1.49}$
	Top-k	$\mathbf{26.32}_{0.11\ +2.55}$	$\mathbf{48.93}_{0.24\ +1.37}$	$\mathbf{27.37}_{0.16\ +2.22}$	$\mathbf{32.00}_{0.26\ +3.34}$	$\mathbf{35.07}_{0.26\ +2.58}$

Table 1: Few-shot results of downstream tasks. Bold indicates the best result in each column. All results are averaged over 3 seeds, with standard deviations indicated in subscripts. Signed numbers indicate the difference in scores from the random baseline.

2021), OpenBookQA (Mihaylov et al. 2018), MMLU (Hendrycks et al. 2020), CMMLU (Li et al. 2024), and C-Eval (Huang et al. 2023). Additionally, we evaluate the model's general understanding capabilities on a range of tests, including RTE (Wang et al. 2018), BBH (Suzgun et al. 2023), WiC (Pilehvar and Camacho-Collados 2019), COPA (Roemmele, Bejan, and Gordon 2011), BoolQ (Clark et al. 2019) and sub-tasks derived from CLUE (Xu et al. 2020) and FewCLUE (Xu et al. 2021). These test sets encompass both English and Chinese languages. We summarize our test results on *knowledge intensive* and *general understanding* tasks, more details can be found in Appendix G.

3.2 Main Results

Table 1 details the results of our main tests. (1) Firstly, we can find that our HKS outperforms the baseline models in most tasks, demonstrating that high-knowledge data can improve the performance of LLMs. Notably, HKS exhibits superior performance relative to PPL, EL2N, and random sampling in terms of average score, with 2.37 pp improvement compared to random sampling, which shows the efficacy of our knowledge scorer. Furthermore, HKS outperforms DSIR in knowledge-intensive tasks, achieving a 0.81 pp improvement. While DSIR uses Wikipedia passages as its target domain, which are rich in knowledge, HKS demonstrates greater efficacy in selecting high-knowledge data. (2) In addition, the performance of the d and c baselines is inferior to that of HKS, which underscores the importance of integrating density and coverage into a comprehensive knowledge scorer. On the other hand, inverse selection of HKS yields lower results than a random baseline, further affirming the efficacy of our approach from a different perspective.

We report the results of the two filtering methods, top-k and sampling. The results show that except for d, top-k is better than sampling, which indicates that in the dimension of knowledge, the top-ranked data do have higher quality than lower-ranked, indirectly reflecting that our knowledge scorer can accurately identify the high-quality data in the dataset.

Method	Tokens	MMLU	CMMLU	CEVAL
Llama-3-8B	/	65.8	51.5	50.8
+ Random	100B	65.9	53.7	52.5
+ HKS	100B	**66.4**	**57.5**	**55.2**

Table 2: Comparison of continual pretrained models.

We also present the performance of our model on perplexity in Figures 3 and 4. On the Pile dataset, our perplexity is marginally higher than that of the c baseline; however, HKS outperforms the c baseline regarding downstream task scores, indicating that perplexity may not be strictly positively correlated with downstream task performance. In the context of specific subsets, our model records a lower perplexity on the Wikipedia and book corpora, which are generally regarded as high-quality sources abundant in knowledge.

3.3 Analysis

Extending to Larger Scale To conduct larger-scale verification, we select 100B tokens from the Pile and Wudao datasets to continue training on Llama-3-8B (Dubey et al. 2024). The results are reported in Table 2. Compared to random selection, our approach results in improvements of 0.9 pp, 3.8 pp, and 2.7 pp on MMLU, CMMLU, and C-Eval, respectively. Furthermore, in comparison to the original Llama-3-8B, our model, post-continual training, exhibits a significant enhancement in knowledge-intensive tasks. The results demonstrate that HKS can be effectively applied to larger datasets and models.

Higher Knowledge Richness in Data Benefits Model Performance To further investigate the effects of knowledge richness in data on our final model performance, we sort the texts in Pile and Wudao from high to low according to their HKS scores, and then employ a top-k strategy to select the

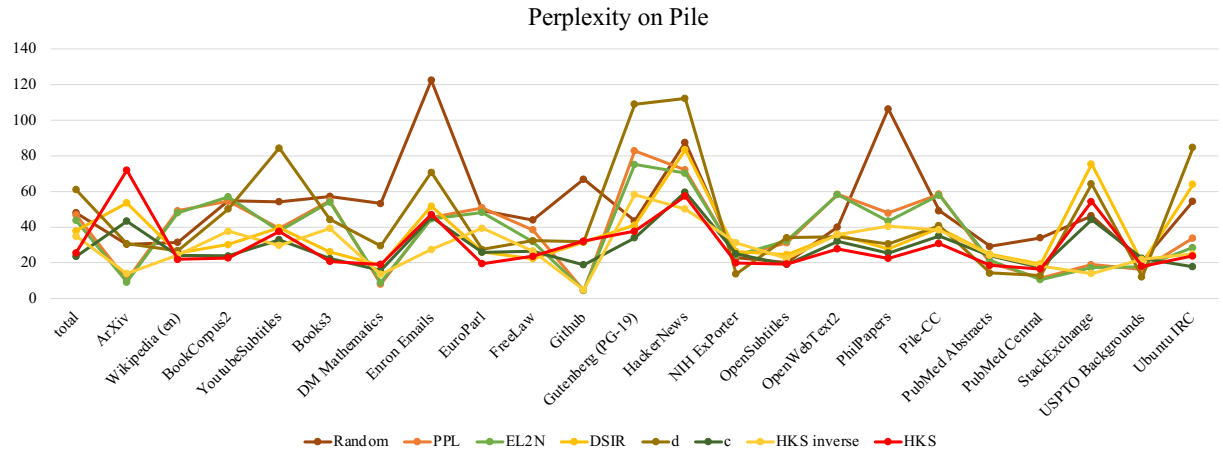

Figure 4: Perplexity evaluation on Pile validation dataset.

α	MMLU	CMMLU	C-Eval	BBH	AVG.
1.00	**27.91**	27.85	**26.89**	**29.66**	**28.08**
0.75	27.07	**27.90**	26.60	26.38	26.67
0.50	25.78	26.63	25.47	26.98	26.53
0.25	26.67	26.11	25.50	26.62	26.23
0.00	25.90	25.25	24.85	27.17	25.79

Table 3: Impact of data quality on model performance. We train models on the merged datasets, varying the proportion of high-knowledge data α.

highest-scoring portion of the data so that the total length of their tokens is 10B, respectively. The score of the lowest-scoring sample in the selected data is determined to be the threshold for the division of high-knowledge data and low-knowledge data. Then we define $\alpha = \frac{N_h}{N_h+N_l}$, where N_h and N_l denote the number of tokens of high and low-knowledge data, respectively. According to the α, we perform uniform sampling from both the high and low knowledge portions. The sampled subsets are then merged to form a 20B token dataset, which is utilized for training our model.

The results outlined in Table 3 demonstrate a trend of diminishing average performance of both knowledge-intensive (MMLU, CMMLU, C-Eval) and reasoning (BBH) tasks, as we move from a dataset consisting entirely of high-knowledge data ($\alpha = 1.00$) to the one that is solely comprised of low-knowledge data ($\alpha = 0.00$). This indicates that the knowledge content of the data not only affects the model's ability to memorize knowledge but is also potentially linked to the model's reasoning abilities. In addition, each benchmark responds differently to changes in the knowledge content of data. For instance, the results of CMMLU show an increase in performance when the dataset includes a mixture of high and low-knowledge data ($\alpha = 0.75$), whereas the results of MMLU, C-Eval, and BBH tend to perform better with higher proportions of high-knowledge data.

Domain-Specific Enhancement Results We explore the application of the HKS model to specific domains, taking the enhancement of the science domain as an illustrative example. We use the Equation 5 to cherry-pick data rich in scientific knowledge. Similar to Section 3.3, we distinguish between high and low-scientific knowledge data by a score threshold at 10B. Subsequently, we define $\beta = \frac{N_{hs}}{N_{hs}+N_{ls}}$ where N_{hs} and N_{ls} denote the token count from high and low-scientific knowledge data, respectively. We follow β to perform uniform sampling across these two distinct parts and finally mixed into 20B token training data. We also categorize the questions in MMLU and CMMLU into scientifically relevant and irrelevant sections[5], and evaluate the model's performance in these different partitions.

β	MMLU		CMMLU	
	Science	Others	Science	Others
1.00	**27.29**	24.16	**28.88**	24.52
0.75	26.63	24.18	28.65	25.36
0.50	25.56	24.97	27.26	25.10
0.25	25.62	23.81	26.38	25.15
0.00	22.69	23.53	26.22	25.10
Random	26.52	24.78	26.39	25.12

Table 4: We carry out experiments focusing on scientific knowledge enhancement, where β signifies the proportion of high scientific knowledge data within the entire dataset.

The results are summarized in Table 4. We can find that: (1) Our domain-specific HKS is effective in selecting high-knowledge data in the desired domain. Within the *Science* category of the MMLU and CMMLU, the optimal performance is attained when the value of β is set to 1.00, with improvements of 0.77 pp and 2.49 pp compared to the random baseline, respectively. The results strongly imply that there is a direct correlation between the amount of HKS-selected domain-specific data and the final result within that domain, indirectly underscoring the effectiveness of domain-specific

[5]Questions that fall under the STEM category are considered as *science-related*.

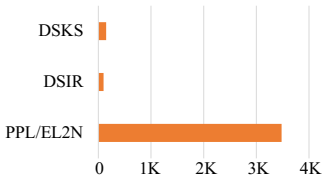

Figure 5: We compare the costs of the various approaches based on cloud server rental fees.

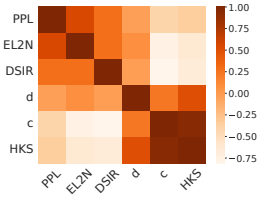

Figure 6: We use Spearman's rank correlation to assess the relation between various methods.

enhancement through our knowledge scorer. (2) Conversely, the most suboptimal performance across all categories is observed when β is set to 0.00. This situation is marked by the absence of high-scientific knowledge data in the training dataset, indicating that such data significantly contribute to the overall performance of the model.

HKS Achieves Superior Cost Efficiency Our methodology employs a gradient-free knowledge scorer, which enables our scoring program to run efficiently on a CPU machine. This offers significant cost and time advantages compared to methods such as Perplexity (Marion et al. 2023) or model-based scoring (Li et al. 2023). To facilitate a more equitable comparison, we consult the rental rates for CPU and GPU servers on Azure[6] and present the costs of the different methods in Figure 5[7]. Our method incurs considerably lower expenses than the PPL/EL2N, and although it is marginally more costly than DSIR, it delivers superior results.

Score Correlation Analysis To investigate the correlation between our method and the baseline methods, we extract a portion of the training data for analysis. We randomly sample 500K texts from the Pile dataset and label this subset with perplexity, error L2-norm, DSIR, d, c, and HKS scores. Then we calculate Spearman's rank correlation coefficient (Gauthier 2001) between these scoring methods pairwise.

From the results depicted in Figure 6, we can find out that: (1) The correlations between the HKS and baseline methods are remarkably low, trending towards a negative correlation. Perplexity is frequently employed as a metric for evaluating linguistic fluency, suggesting that high-knowledge and fluency do not necessarily correlate strongly. Nevertheless, the HKS still delivers superior performance, indicating that compared to text fluency, high-knowledge data are more beneficial for model training. (2) There is a low correlation between d and c, indicating that these dimensions are relatively orthogonal to each other, which is consistent with our observation in Section 2.3. (3) HKS scores show a strong correlation with both d and c, indicating that our scoring function effectively synthesizes metrics along these two dimensions.

[6]https://azure.microsoft.com/en-us/pricing/details/machine-learning/

[7]The cost is determined by multiplying the number of CPUs/GPUs, the execution time, and the hourly rate per device.

4 Related Works

High-quality data selection Research on high-quality training data selection falls into two approaches: metric-based selection and reference data-based selection. The former typically employs a manually defined metric to assess data quality. Rule-based methods (Rae et al. 2021; Dodge et al. 2021; Cai et al. 2024) and deduplication (Lee et al. 2022; Touvron et al. 2023) are two widely used approaches, which are straightforward but may not comprehensively evaluate data quality. Model-based methods leverage language models to derive data attributes, such as perplexity (Marion et al. 2023), self-influence (Thakkar et al. 2023), or density (Sachdeva et al. 2024). Several researchers have also explored directly querying LLMs to give data quality scores (Sachdeva et al. 2024; Wettig et al. 2024). While these methods may enhance the model's performance, they have not taken into account the data from the perspective of knowledge content.

Reference data-based selection involves pre-defining a set of high-quality data to guide the filtering process. Commonly, a classifier is employed to distinguish data between high and low quality (Gururangan et al. 2022; Chowdhery et al. 2023; Li et al. 2023), which is dependent on the classifier's precision. Xie et al. (2024b) involve importance resampling to extract a subset from the raw dataset that closely aligns with the target domain distribution, while Engstrom, Feldmann, and Madry (2024) propose using datamodels (Ilyas et al. 2022) to assess data quality and optimize the model's performance. We argue that these selection methods may be influenced by the bias towards "high-quality data". Additionally, they could potentially lack diversity in knowledge, a deficiency that can be mitigated by incorporating our knowledge coverage metric.

Knowledge analysis Kandpal et al. (2023) uses salient entities within test questions and answers to assess the distribution of relevant knowledge in the training set, while Lucy et al. (2024) use the 'AboutMe' page to identify the main knowledge of a webpage. Lu et al. (2023) involve annotating the tag fragments contained in each text of the instruction-tuning training set, but these tags are often viewed as topics rather than knowledge pieces. In contrast to these studies, we have developed a comprehensive, categorized collection of knowledge elements to tag and filter training data.

5 Conclusion

In this work, we propose a novel method for selecting high-quality data from a knowledge perspective. We create a comprehensive knowledge element pool covering various fields, and develop an effective method free from parameter updates for selecting high-knowledge data for both general and specific domains. Our extensive experimental results prove that our method outperforms all the baselines in knowledge-intensive and general understanding tasks, which suggests that the high-knowledge data we selected is of superior quality. In addition, our experiments also confirm that we can effectively improve the data quality in specific domains. Nevertheless, our work does have some limitations, which we will discuss in the Appendix A.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62477001.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Allen-Zhu, Z.; and Li, Y. 2024. Physics of language models: Part 3.3, knowledge capacity scaling laws. *arXiv preprint arXiv:2404.05405*.

Bhakthavatsalam, S.; Khashabi, D.; Khot, T.; Dalvi Mishra, B.; Richardson, K.; Sabharwal, A.; Schoenick, C.; Tafjord, O.; and Clark, P. 2021. Think you have Solved Direct-Answer Question Answering? Try ARC-DA, the Direct-Answer AI2 Reasoning Challenge. *arXiv e-prints*, arXiv–2102.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Cai, Z.; Cao, M.; Chen, H.; Chen, K.; Chen, K.; Chen, X.; Chen, X.; Chen, Z.; Chen, Z.; Chu, P.; et al. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113.

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2924–2936.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dodge, J.; Sap, M.; Marasović, A.; Agnew, W.; Ilharco, G.; Groeneveld, D.; Mitchell, M.; and Gardner, M. 2021. Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 1286–1305.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; and Sui, Z. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Engstrom, L.; Feldmann, A.; and Madry, A. 2024. DsDm: Model-Aware Dataset Selection with Datamodels. *arXiv preprint arXiv:2401.12926*.

Gao, L.; Biderman, S.; Black, S.; Golding, L.; Hoppe, T.; Foster, C.; Phang, J.; He, H.; Thite, A.; Nabeshima, N.; et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Gauthier, T. D. 2001. Detecting trends using Spearman's rank correlation coefficient. *Environmental forensics*, 2(4): 359–362.

Gururangan, S.; Card, D.; Dreier, S.; Gade, E.; Wang, L.; Wang, Z.; Zettlemoyer, L.; and Smith, N. A. 2022. Whose Language Counts as High Quality? Measuring Language Ideologies in Text Data Selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2562–2580.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; Casas, D. d. L.; Hendricks, L. A.; Welbl, J.; Clark, A.; et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Huang, Y.; Bai, Y.; Zhu, Z.; Zhang, J.; Zhang, J.; Su, T.; Liu, J.; Lv, C.; Zhang, Y.; Lei, J.; Fu, Y.; Sun, M.; and He, J. 2023. C-Eval: A Multi-Level Multi-Discipline Chinese Evaluation Suite for Foundation Models. arXiv:2305.08322.

Ilyas, A.; Park, S.; Engstrom, L.; Leclerc, G.; and Madry, A. 2022. Datamodels: Predicting Predictions from Training Data, Baltimore. In *Proceedings of the 39 th International Conference on Machine Learning*, volume 162.

Kandpal, N.; Deng, H.; Roberts, A.; Wallace, E.; and Raffel, C. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, 15696–15707. PMLR.

Kool, W.; Van Hoof, H.; and Welling, M. 2019. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*, 3499–3508. PMLR.

Kreutzer, J.; Caswell, I.; Wang, L.; Wahab, A.; van Esch, D.; Ulzii-Orshikh, N.; Tapo, A.; Subramani, N.; Sokolov, A.; Sikasote, C.; et al. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10: 50–72.

Le Scao, T.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; Gallé, M.; et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Lee, K.; Ippolito, D.; Nystrom, A.; Zhang, C.; Eck, D.; Callison-Burch, C.; and Carlini, N. 2022. Deduplicating Training Data Makes Language Models Better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8424–8445.

Li, H.; Zhang, Y.; Koto, F.; Yang, Y.; Zhao, H.; Gong, Y.; Duan, N.; and Baldwin, T. 2024. CMMLU: Measuring massive multitask language understanding in Chinese. arXiv:2306.09212.

Li, Y.; Bubeck, S.; Eldan, R.; Del Giorno, A.; Gunasekar, S.; and Lee, Y. T. 2023. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Lu, K.; Yuan, H.; Yuan, Z.; Lin, R.; Lin, J.; Tan, C.; Zhou, C.; and Zhou, J. 2023. # InsTag: Instruction Tagging for Analyzing Supervised Fine-tuning of Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Lucy, L.; Gururangan, S.; Soldaini, L.; Strubell, E.; Bamman, D.; Klein, L.; and Dodge, J. 2024. AboutMe: Using Self-Descriptions in Webpages to Document the Effects of English Pretraining Data Filters. *arXiv preprint arXiv:2401.06408*.

Marion, M.; Üstün, A.; Pozzobon, L.; Wang, A.; Fadaee, M.; and Hooker, S. 2023. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*.

Mihaylov, T.; Clark, P.; Khot, T.; and Sabharwal, A. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Conference on Empirical Methods in Natural Language Processing*.

Muennighoff, N.; Rush, A.; Barak, B.; Le Scao, T.; Tazi, N.; Piktus, A.; Pyysalo, S.; Wolf, T.; and Raffel, C. A. 2024. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36.

OpenAI, R. 2023. GPT-4 technical report. *ArXiv*, 2303.

Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; and Wu, X. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Pao, D.; Lin, W.; and Liu, B. 2010. A memory-efficient pipelined implementation of the aho-corasick string-matching algorithm. *ACM Transactions on Architecture and Code Optimization (TACO)*, 7(2): 1–27.

Patel, J. M. 2020. *Introduction to Common Crawl Datasets*, 277–324. Berkeley, CA: Apress. ISBN 978-1-4842-6576-5.

Pilehvar, M. T.; and Camacho-Collados, J. 2019. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 1267–1273.

Rae, J. W.; Borgeaud, S.; Cai, T.; Millican, K.; Hoffmann, J.; Song, F.; Aslanides, J.; Henderson, S.; Ring, R.; Young, S.; et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Roemmele, M.; Bejan, C. A.; and Gordon, A. S. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Sachdeva, N.; Coleman, B.; Kang, W.-C.; Ni, J.; Hong, L.; Chi, E. H.; Caverlee, J.; McAuley, J.; and Cheng, D. Z. 2024. How to Train Data-Efficient LLMs. *arXiv preprint arXiv:2402.09668*.

Suzgun, M.; Scales, N.; Schärli, N.; Gehrmann, S.; Tay, Y.; Chung, H. W.; Chowdhery, A.; Le, Q.; Chi, E.; Zhou, D.; et al. 2023. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. In *Findings of the Association for Computational Linguistics: ACL 2023*, 13003–13051.

Thakkar, M.; Bolukbasi, T.; Ganapathy, S.; Vashishth, S.; Chandar, S.; and Talukdar, P. 2023. Self-Influence Guided Data Reweighting for Language Model Pre-training. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2033–2045.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 353–355.

Wettig, A.; Gupta, A.; Malik, S.; and Chen, D. 2024. QuRating: Selecting High-Quality Data for Training Language Models. *arXiv preprint arXiv:2402.09739*.

Xie, S. M.; Pham, H.; Dong, X.; Du, N.; Liu, H.; Lu, Y.; Liang, P. S.; Le, Q. V.; Ma, T.; and Yu, A. W. 2024a. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36.

Xie, S. M.; Santurkar, S.; Ma, T.; and Liang, P. S. 2024b. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36.

Xu, L.; Hu, H.; Zhang, X.; Li, L.; Cao, C.; Li, Y.; Xu, Y.; Sun, K.; Yu, D.; Yu, C.; et al. 2020. CLUE: A Chinese Language Understanding Evaluation Benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, 4762–4772.

Xu, L.; Lu, X.; Yuan, C.; Zhang, X.; Xu, H.; Yuan, H.; Wei, G.; Pan, X.; Tian, X.; Qin, L.; et al. 2021. Fewclue: A chinese few-shot learning evaluation benchmark. *arXiv preprint arXiv:2107.07498*.

Yuan, S.; Zhao, H.; Du, Z.; Ding, M.; Liu, X.; Cen, Y.; Zou, X.; Yang, Z.; and Tang, J. 2021. Wudaocorpora: A super large-scale chinese corpora for pre-training language models. *AI Open*, 2: 65–68.

Zhang, F.; Liu, X.; Tang, J.; Dong, Y.; Yao, P.; Zhang, J.; Gu, X.; Wang, Y.; Kharlamov, E.; Shao, B.; et al. 2022. Oag: Linking entities across large-scale heterogeneous knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*.

Zhu, Z. A.; and Li, Y. 2023. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*.