Basic Reading Distillation

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated remarkable abilities in various natural language processing areas, but they demand high computation resources which limits their deployment in real-world. Distillation is one technique to solve this problem through either knowledge distillation or task distillation. Both distillation approaches train small models to imitate specific features of LLMs, but they all neglect basic reading education for small models on generic texts that are unrelated to downstream tasks. In this paper, we propose basic reading distillation (BRD) which educates a small model to imitate LLMs basic reading behaviors, such as named entity recognition, question raising and answering, on each sentence. After such basic education, we apply the small model on various tasks including language inference benchmarks and BIG-Bench-Hard tasks. It shows that the small model can outperform or perform comparable to over 20x bigger LLMs. Probing analysis reveals that our small model gains strengthened abilities layer-wisely, leading to better performances across various tasks.

1 Introduction

Large language models (LLMs) exhibit consistent performance gains across various areas (Zhao et al., 2023; Huang and Chang, 2023; Chang et al., 2023). Nevertheless, their formidable size and high computational requirements impede their real-world applications. Distillation is one widespread approach to tackle this issue by distilling smaller language models from LLMs. It is divided into mainly two categories: knowledge distillation and task distillation. Both distillation approaches adopt the teacher-student framework, in which the smaller language models act as the student models, and are trained to imitate specific features of LLMs, which act as the teacher models. Specifically, knowledge distillation (Hinton et al., 2015) usually trains the student models to imitate implicit features inside the teacher models, such as hidden layers (Jiao et al., 2020), attention maps (Li et al., 2020; Wang et al., 2021b), and output logits (Liu et al., 2020). Task distillation usually trains the student models to imitate explicit task behaviors of LLMs, such as label prediction (Chen et al., 2020; Wang et al., 2021a; Iliopoulos et al., 2022; Agrawal et al., 2023) and rationale generation (Hsieh et al., 2023; Wang et al., 2023; Ho et al., 2023), on various downstream tasks.

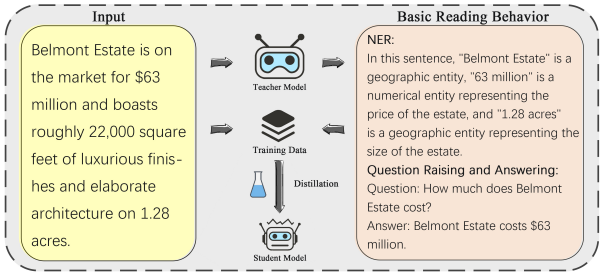

Figure 1: The illustration of BRD process.

Different to both distillation approaches, we propose basic reading distillation (BRD) that teaches a student model basic reading abilities such as named entity recognition, question raising, and question answering, on general sentences. It simulates human reading education via interactions including raising questions about parts of a sentence, answering the questions, extracting important information such as named entities. We believe that such basic reading education on every sentence is important before application on downstream tasks. It can avoid the hurry of task distillation which rushes into downstream applications directly without basic education. BRD also avoids the implicit nature of knowledge distillation which leads to the deficiency of learning interpretability. BRD demonstrates explicit reading behaviors that are easy to interpret. Furthermore, BRD can perform on un-

limited data resource, breaking the data scale and diversity limitation criticized by Gudibande et al. (2023).

Figure 1 illustrates the process of BRD. It starts by prompting LLMs to generate basic reading behaviors on general sentences, then proceeds with training the student model to imitate these behaviors. Experiments on various NLP tasks, including language inference benchmarks and Big-Bench-Hard tasks, show that although the student model is trained on the general imitation data that is irrelevant to the downstream tasks, it can inherit LLMs abilities, leading to excellent zero-shot downstream performances better than or comparable to those of LLMs. Furthermore, after this basic education of the student model on general sentences, we finetune the student model for downstream tasks, and find that the basic reading eduction is necessary before the application on downstream tasks, achieving on par or better performances when compared to the over 20x bigger teacher model. To analyze the effect of BRD, we insert probes to layers of the student model. The probing result shows that those distilled layers exhibit better abilities to comprehend sentences well, leading to better performances across various tasks. In summary, the main contributions are:

- We propose BRD that educates the student model to imitate basic reading behaviors of the teacher model on general sentences.
- Experiments show that the student model exhibits excellent abilities distilled from the teacher model on various downstream tasks, achieving on par or even better performances against the teacher model.
- The probing analysis reveals that BRD well trains layers of the student model to be ready for downstream applications.

2 Related Work

There are mainly two streams of distilling approaches: knowledge distillation and task distillation. Knowledge distillation focuses on teaching the student model to imitate the implicit features inside the teacher model, while task distillation focuses on teaching the student model to imitate explicit behaviors of the teacher model on downstream tasks.

2.1 Knowledge Distillation

The field is pioneered by Bucila et al. (2006); Hinton et al. (2015), followed by works using various types of internal information from the teacher model, including attention maps (Li et al., 2020; Wang et al., 2021b), output logits (Liu et al., 2020), hidden layers (Jiao et al., 2020). In the era of LLMs, GKD uses advanced memory optimization methods to address the memory constraint problem in distilling from LLMs (Tan et al., 2023), MiniLLM uses reverse KL divergence to prevent the student model from overestimating the void regions of the teacher distribution (Gu et al., 2023). Agarwal et al. (2024) use on-policy distillation that trains the student model on its self-generated mistakes. In the case that internal information of LLMs is not accessible and only decisions of LLMs are available, Zhou et al. (2023) estimate logits from the decision distributions to train the student model.

2.2 Task Distillation

The task predictions or reasoning rationales made by the teacher model are used to train the student model in task distillation. Despite the noisy predictions of the teacher model, the student model achieves good imitation effects in performing the tasks (Chen et al., 2020; Wang et al., 2021a; Iliopoulos et al., 2022; Agrawal et al., 2023). Besides the task predictions, rationales for the answers generated by the teacher model show efficiency in training the student model with less data (Hsieh et al., 2023; Wang et al., 2023; Ho et al., 2023; Magister et al., 2023). Task distillation is closely related to model imitation researches (Orekondy et al., 2019; Wallace et al., 2020), which collect API outputs of a a proprietary LM for some tasks, then use the outputs to fine-tune an open-source LM. Gudibande et al. (2023) criticize the data scale and limited diversity in model imitation. Mukherjee et al. (2023) address this criticism by using explanation tuning, more task data, and instructions. In comparison, BRD can perform on every sentence, leading to unlimited data resource that breaks the limitation on data scale and diversity.

In summary, task distillation focuses on the data of specific tasks, while our BRD mainly focuses on general sentences unrelated to the specific tasks, and the basic reading behaviors in BRD are basic education resource not aiming at the specific applications.

Task Instruction	Perform named entity recognition on a given sentence without recognizing personal pronouns in the input sentence as human names.
Examples	**Enter a sentence:** Barack Obama was the 44th President of the United States. **Output result:** In this sentence, "Barack Obama" is a person name entity, and "United States" is a geopolitical entity. **Enter a sentence:** I just bought a new MacBook Pro from Apple. **Output result:** In this sentence, "Apple" is an organization name entity, and "MacBook Pro" is a product name entity. **Enter a sentence:** The Eiffel Tower is a famous landmark in Paris, France. **Output result:** In this sentence, "Eiffel Tower" is a landmark name entity, and "Paris" and "France" are geopolitical entities.
Model Input	**Enter a sentence:** Belmont Estate is on the market for $63 million and boasts roughly 22,000 square feet of luxurious finishes and elaborate architecture on 1.28 acres. **Output result:**
Model Output	In this sentence, "Belmont Estate" is a geographic entity, "63 million" is a numerical entity representing the price of the estate, and "1.28 acres" is a geographic entity representing the size of the estate.

Table 1: The prompt for the teacher model to extract named entity information from an input sentence. Each example consists of a sentence and its named entity information. The response from the teacher model is listed in model output.

3 Approach

In BRD, we use a subset of CommonCrawl (CC-100) corpus, which is usually included in LLMs pre-training, as the education resource to conduct the basic reading education. The whole education process contains two stages. In the first stage, for each sentence in the corpus, the teacher model is prompted to perform basic reading. In the second stage, we collect all basic reading behavior data to train the student model, and finally test the student model ability on various tasks.

3.1 Basic Reading Behaviors of the Teacher Model

We utilize the in-context learning ability of the teacher model to elicit its basic reading behaviors including named entity recognition, question raising and answering. Given the corpus, we set up a prompt template consisting of task description, task examples, and input sentence from the corpus.

Table 1 lists the named entity recognition prompt and the response from the teacher model. We can see that, given the few-shot examples including entities and their types, the teacher model responses with more detailed contents of the entities, such as the price or size of the entities, which are beneficial for educating the student model to grasp the important information contained in the input sentence. Table 2 lists the question raising and answering prompt and the response from the teacher model. In the task instruction of the prompt, question is constrained to be about the content, structure, or attitude of the input sentence. Its raising and answering embody the teacher model's reading ability, which is targeted to be transferred to the student model.

3.2 Training the Student Model

The student model is initialized by a released smaller pre-trained language model. We continue training the student model based on the basic reading behavior data generated by the teacher model. To stabilize the training process, we mix the basic reading behavior data with the original sentences of the corpus to avoid the catastrophic forgetting of the pre-trained model.

Suppose we have a passage consisting of three sentences s_1, s_2, and s_3, we constitute the named entity recognition passage: s_1 <sep> NER(s_1) <sep> s_2 <sep> NER(s_2) <sep> s_3 <sep> NER(s_3), where NER denotes the named entity recognition result of the teacher model for each sentence, and <sep> is the delimiter. Similarly, we constitute the question raising and answering passage: s_1 <sep> QRA(s_1) <sep> s_2 <sep> QRA(s_2) <sep> s_3 <sep>

Task Instruction	Ask a question to the input sentence, you can ask questions about the content, structure or attitude of the sentence, and then find the answer to the corresponding question in the original sentence. Output in the format "Question: Answer:".
Example	**The sentence:** In order to graduate with honors, he needed to maintain a high GPA throughout college. **Question:** What did he need to do in order to graduate with honors? **Answer:** Maintain a high GPA throughout college.
Model Input	**The sentence:** Belmont Estate is on the market for $63 million and boasts roughly 22,000 square feet of luxurious finishes and elaborate architecture on 1.28 acres.
Model Output	**Question:** How much does Belmont Estate cost? **Answer:** Belmont Estate costs $63 million.

Table 2: The prompt for the teacher model to perform question raising and answering on an input sentence. Question is limited to be about the input sentence. The response from the teacher model is listed in model output.

QRA(s_3), where QRA denotes the question raising and answering result of the teacher model for each sentence. The original passage is formatted as s_1 <sep> s_2 <sep> s_3. We use passage instead of sentence like the usual language model pre-training conducts to utilize long contexts.

In this way, we build all original passages, denoted as D_{ORI}, all named entity recognition passages, denoted as D_{NER}, and all question raising and answering passages, denoted as D_{QRA}. We mix them together to build the training set D_{TRAIN}, on which we train the student model to minimize the loss in an autoregressive manner:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\log P(y_t|y_{<t})$$

where y is the passage with length T, and N is the number of passages in D_{TRAIN}.

3.3 Testing the Student Model

We test the ability of the student model in two manual-label-free settings: zero-shot test and unsupervised distillation. In zero-shot test, the student model is directly tested on various downstream tasks. In unsupervised distillation, we use sentences of the downstream tasks for further BRD, but the labeled answers for these sentences are not used in this setting to avoid manual labor.

Zero-shot Test. We evaluate the zero-shot capability of the student model on a spectrum of downstream tasks, including natural language inference (XNLI(Conneau et al., 2018), CB(de Marneffe et al., 2019), RTE(Wang et al., 2018)) , paraphrasing (PAWS-X(Zhang et al., 2019)) , Boolean QA (BOOLQ(Clark et al., 2019)) , sentiment analysis (SST-2(Socher et al., 2013)), and Big-Bench-Hard(Suzgun et al., 2022). The prompt templates for these tasks are listed in table 3. For the multiple tasks in BIG-Bench-Hard, we utilize the prompt format provided in Gao et al. (2021)[1].

For predicting the answers of the tasks, we use the average of per-token log-probabilities of candidate answers as the scoring function for all downstream tasks:

$$\bar{P} = \frac{1}{n}\sum_{i=1}^{n}\log P_i(y_i|x_{\text{prompt}})$$

where x_{prompt} denotes an input to the student model, y denotes a candidate answer for x_{prompt}, and n is the total number of words in y. This average computation is to cover BIG-Bench-Hard tasks, whose candidate answers are phrases/sentences rather than single words.

Unsupervised Distillation. In this setting, we use the downstream task data (excluding answers) to further distill the teacher model into the student model. It is a common practice that the teacher model generates the answers given the downstream task inputs in the training set, and these generated pseudo answers are used to supervise the student model. Although it is simple, Wang et al. (2021a); Iliopoulos et al. (2022) have proved its efficiency in distilling the downstream task abilities of the teacher model into the student model.

We add BRD into this distillation process to include basic reading education for the downstream

[1] https://github.com/EleutherAI/lm-evaluation-harness

Task	Template	Candidate Answers
XNLI	{premise} Question: Does this imply that "{hypothesis}"? Yes, no or maybe? Answer:	Yes | No | Maybe
RTE CB	Question: Can we infer that "{hypothesis}" ? {premise} Answer:	Yes | No | Maybe
PAWS-X	Sentence 1: {sentence1} Sentence 2: {sentence2} Question: Do Sentence 1 and Sentence 2 express the same meaning? Answer:	Yes | No
BOOLQ	{passage} Question: {question} Answer:	Yes | No
SST-2	Question: Does the following sentence have a positive or negative sentiment? Sentence: {sentence} Answer:	positive | negative

Table 3: The prompt templates for the different tasks in the zero-shot test.

tasks. With the strengthened basic reading ability, we expect the student model performs better in the downstream tasks. We choose the downstream tasks same to the zero-shot test setting except BIG-Bench-Hard task, which lacks downstream task training data, and is thus not suitable for the distillation.

In particular, for each sentence in each downstream task training set, we prompt the teacher model to generate NER result and question raising and answering result, then we concatenate the original sentence with the two results respectively to build the basic reading training data. During distillation, on this basic reading data, the student model is further trained auto-regressively, but on the original data, the student model is further trained to just predict answers. During testing, the student model uses the prompts same to those used in the zero-shot test setting.

4 Experiment

We use the well-known LLM Vicuna-13B [2] (Chiang et al., 2023) as our teacher model due to its high efficiency in generating large volume of texts for teaching. We use XGLM-564M (Lin et al., 2022) [3], which is the smaller language model of the same decoder-only family, to initialize our student model. To compare the student model with larger model pre-trained on the same data origin, we also include XGLM-7.5B in comparison. In BRD, we use 5 million passages from CC-100 corpus to collect the basic reading data generated by Vicuna-13B. We train the student model with learning rate = 0.0003, batch size = 8, and max input length = 2048, for a maximum of 40000 steps. We save the model every 1000 steps. We test the student model in the two settings specified in section 3.3: zero-shot test and unsupervised distillation. We present the detailed experimental setup in A.1.

4.1 Zero-shot Test Results

We denote the student model as XGLM-BRD. Since XGLM-BRD is initialized by XGLM-564M, and is further trained on both the original passages and the correspondingly generated basic reading data, we also further trained XGLM-564M only on the original passages to check the effect of the basic reading data. Such further trained model is denoted as XGLM-564M-FURTHER. To ensure fair comparison, the number of further training steps of either XGLM-BRD or XGLM-564M-FURTHER is set 18,000. Table 4 lists the zero-shot test results.

BRD effectively enhances the zero-shot performance of the smaller model. In comparison to XGLM-564M, XGLM-BRD shows significant improvement in zero-shot testing across the downstream tasks. The most notable increases are in RTE, CB, BOOLQ, and SST-2, with the relative improvements of 20.35%, 8.77%, 17.38%, and 23.94%, respectively. Moreover, XGLM-564M-FURTHER performs much worse

[2] https://github.com/lm-sys/FastChat

[3] https://github.com/facebookresearch/fairseq/tree/main/examples/xglm

Model	Tasks							
	XNLI	RTE	CB	PAWS-X	BOOLQ	SST-2	BBH-Avg	Avg
Vicuna-13B	59.1	78.3	71.4	62.9	84.3	81.5	35.9	67.6
XGLM-7.5B	36.6	50.9	60.7	56.8	57.2	69.5	33.9	52.2
XGLM-564M	35.5	46.2	53.6	51.3	51.2	63.9	31.9	47.7
XGLM-564M-FURTHER	34.9	46.6	51.8	51.6	51.5	59.4	32.2	46.9
XGLM-BRD	**36.6**	**55.6**	**58.3**	**51.7**	**60.1**	**79.2**	**33.4**	**53.6**

Table 4: The results of the zero-shot testing. The top part lists the LLMs results, and the bottom part lists the results of the smaller models that have 564M parameters each.

than XGLM-BRD, revealing that only using the original passages for further training does not yield enhancements and may even leads to decreases in some tasks. It is the basic reading data for further training that advance the student model via basic reading education. Note that the basic reading data are not related to the downstream tasks. They come from the general domain CC-100 corpus. The basic reading abilities of NER, question raising and answering, which are acquired via BRD, DO help the student model to perform well in unseen tasks.

BRD narrows the performance gap between the smaller model and LLMs. The zero-shot performance of XGLM-BRD approaches or even surpasses that of XGLM-7.5B, which is 15x bigger, in the downstream tasks. On the XNLI task, XGLM-BRD performs comparably to XGLM-7.5B. In RTE, BOOLQ, and SST-2 tasks, XGLM-BRD achieves relative improvements of 9.23%, 5.07%, and 13.96% respectively. Although in CB and PAWS-X tasks, XGLM-BRD does not reach the anticipated performance, the gap has been narrowed. There is still a gap between the student model XGLM-BRD and the teacher model Vicuna-13B, but this gap is significantly reduced or disappeared when we conduct unsupervised distillation for XGLM-BRD on the downstream tasks as shown in the experiment section 4.2.

BRD is effective on the Big-Bench-Hard tasks. Table 4 only lists the average performance on Big-Bench-Hard subtasks. The full results on Big-Bench-Hard all subtasks are listed in the appendix Table 10 due to space limit. It is evident that BRD significantly improves the overall performance of the smaller model. In particular, in the tasks of "Geometric Shapes" and "Reasoning About Colored Objects", XGLM-BRD achieves substantial increases of 83.25% and 59.64%, respectively, over XGLM-564M. In many tasks, XGLM-BRD approaches or surpasses LLMs. This finding underscores the potential of BRD in enhancing model performance, especially in complex tasks. However, in certain tasks such as "Date Understanding", Vicuna-13B still maintains a significant lead. This indicates that the student model still needs to improve its time concepts in training.

Model	Tasks						
	XNLI	RTE	CB	PAWS-X	BOOLQ	SST-2	Avg
Vicuna-13B	59.1	78.3	71.4	62.9	84.3	81.5	72.9
XGLM-7.5B	36.6	50.9	60.7	56.8	57.2	69.5	55.3
XGLM-564M	35.5	46.2	53.6	51.3	51.2	63.9	50.3
XGLM-UTD	57.1	58.1	60.7	**64.8**	74.8	77.2	65.5
XGLM-BRD	36.6	55.6	58.3	51.7	60.1	79.2	56.9
XGLM-BRD^2	**59.2**	**62.5**	**82.1**	**64.8**	**75.0**	**81.9**	**70.9**

Table 5: The results of the Unsupervised Distillation. The top part lists the LLMs results, and the bottom part lists the results of the models whose parameter size is 564M.

4.2 Unsupervised Distillation Results

In the unsupervised distillation setting, the baseline is the task distillation approach, which uses the pseudo answers generated by the teacher model on the downstream tasks to supervise the student model. We denote the student model in this baseline XGLM-UTD, which is initialized by XGLM-564M. Our approach in this setting uses BRD twice, that is, on the general data we conduct BRD to obtain the student model XGLM-BRD, then on the downstream task data, we conduct BRD again to obtain the new student model, denoted as XGLM-BRD^2. Table 5 lists the results on the downstream tasks.

BRD outperforms the task distillation approach XGLM-UTD. XGLM-UTD establishes a strong baseline that significantly outperforms XGLM-564M. This demonstrates that even pseudo answers can supervise the student model to perform well on the downstream tasks. When BRD is introduced into this process, the improvement is even more pronounced. In comparison to the strong

	XNLI	RTE	CB	PAWS-X	BOOLQ	SST-2
XGLM-564M-FBRD	58.1	61.0	71.4	63.1	74.4	81.1
XGLM-BRD^2	**59.2**	**62.5**	**82.1**	**64.8**	**75.0**	**81.9**

Table 6: The comparison between basing on XGLM-BRD and basing on XGLM-564M for further BRD on the downstream tasks.

	Tasks						
	XNLI	RTE	CB	PAWS-X	BOOLQ	SST-2	Avg
XGLM-BRD^2	59.2	62.5	82.1	64.8	75.0	81.9	70.9
−NER	58.0	61.4	71.4	64.1	74.3	81.4	68.4
−QRA	58.3	61.0	67.9	63.9	74.9	80.5	67.8

Table 7: The effects of deleting different basic reading behaviors for XGLM-BRD^2 in the unsupervised distillation test.

	Tasks						
	XNLI	RTE	CB	PAWS-X	BOOLQ	SST-2	Avg
XGLM-564M	35.5	46.2	53.6	51.3	51.2	63.9	50.3
XGLM-BRD	36.6	**55.6**	**58.3**	**51.7**	**60.1**	**79.2**	**56.9**
−SentData	**39.2**	54.5	57.1	51.5	59.1	74.2	55.9

Table 8: The result of training XGLM-BRD based on the data excluding the sentiment-related questions and answers, denoted by −SentData, in the zero-shot test.

XGLM-UTD, our XGLM-BRD^2 achieves further improvements on XNLI, RTE, CB, BOOLQ, and SST-2, with the relative increases of 3.68%, 7.57%, 35.26%, 0.27%, and 6.09%, respectively.

BRD enhances the smaller model to perform better than or comparable to the teacher model. When compare the student model XGLM-BRD^2 with the teacher model Vicuna-13B, XGLM-BRD^2 outperforms in four tasks: XNLI, CB, PAWS-X, and SST-2. In RTE and BoolQ, the performance gap is significantly reduced. This comparison shows that BRD can fully develop the potential of the student model via strengthening its basic reading abilities, leading to comparable or superior performance to the 26x bigger teacher model.

Basing on XGLM-BRD is better than basing on XGLM-564M. In the above results, further BRD in training XGLM-BRD^2 on the downstream tasks is based on XGLM-BRD. We also test further BRD based on XGLM-564M, which is denoted as XGLM-564M-FBRD. Table 6 lists the comparison result. It shows that XGLM-BRD^2 generally outperforms XGLM-564M-FBRD across various downstream tasks, highlighting that basing on XGLM-BRD is more effective. These results emphasize the importance of BRD as a prerequisite step in improving the adaptability and efficacy of models in downstream applications.

5 Analysis

5.1 Layer-wise Probing

Inserting probes can reveal the interpretable aspects hidden in the neural networks (Belinkov, 2022). We insert probes layer-wisely to check the efficacy of the distilled student model. In particular, for each downstream task, we extract the representation by averaging vectors per layer for each sentence in the training set, and train the probing classifier per layer based on the representation. The training loss is the regularized cross-entropy loss of the task prediction against the true label of the sentence. Through inserting probes layer-wisely, we can check how well each layer prepares for the downstream tasks.

Figure 2 presents the results of probing XGLM-564M and XGLM-BRD in the zero-shot test setting. It is clear that XGLM-BRD outperforms XGLM-564M on almost all layers for all downstream tasks. Although XGLM-BRD is trained on the general corpus that is not related to the downstream tasks, basic reading education influences deep layers of the model, empowering each layer with enhanced downstream task prediction ability.

5.2 Ablation Study

The impact of different basic reading behaviors. We test the contribution of the different basic reading behaviors by deleting either NER or QRA data of the downstream tasks in training XGLM-BRD^2. Table 7 lists the ablation results in the unsupervised distillation test. It shows that deleting the QRA data impacts the performance more significantly than deleting the NER data in most tasks.

The impact of sentiment-related questions and answers. Since our QRA data include questions and answers about the attitude of a sentence, which are related to the SST-2 task, we exclude such data for training XGLM-BRD by deleting the questions about the attitude or the answers containing words of positive/negative/neutral. The objective is to check whether the performance improvement is due to the presence of such data.

Table 8 shows the result in the zero-shot test setting. Excluding the sentiment-related data

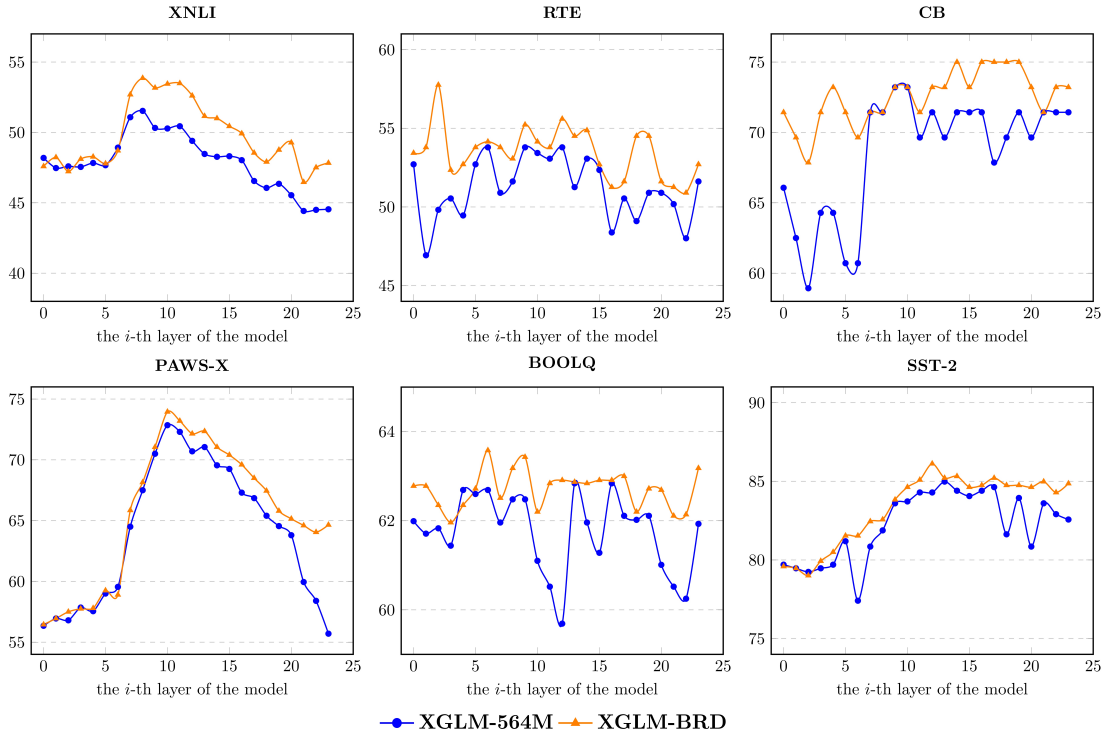

Figure 2: The results of probing XGLM-564M and XGLM-BRD layer-wisely on the downstream tasks in the zero-shot test setting. The horizontal axis represents the specific layer in the model, and the vertical axis is the prediction accuracy (%) for each task.

does influence SST-2 performance significantly, resulting in a decrease of 5 points compared to training XGLM-BRD on full data. Thanks to the remaining data for training XGLM-BRD, it still performs significantly better than XGLM-564M by a large margin on SST-2 task. On the other tasks unrelated to the sentiment analysis, the influence is not so significant, indicating that the remaining data is also effective for BRD across the tasks. On XNLI task, excluding the sentiment-related data obtains a significant improvement over XGLM-BRD trained on full data. This indicates that the sentiment-related data is not fit for the language inference task.

The impact of BRD data size. We investigate how performance varies along with different BRD data sizes in the zero-shot test. Figure 3 shows the curve. Most tasks exhibit a steady improvement as BRD data gets bigger, but the improvement gets saturated when BRD data size arrives at more than one million passages.

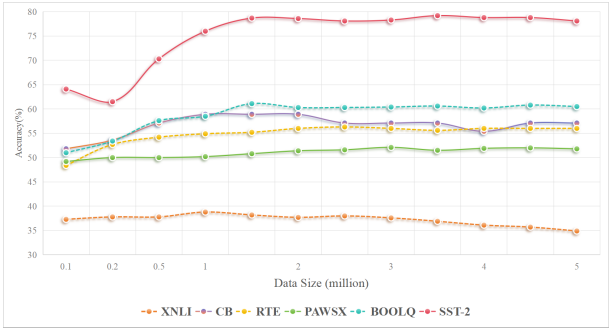

Figure 3: The performance curve along with different BRD data sizes (in million passages).

6 Conclusion

In this paper, we propose to distill the basic reading abilities of LLMs into small models. In particular, we collect basic reading behaviors of LLMs such as NER or question raising and answering about parts of an input text at first, then we train small models based on the collected behaviors. Through such basic education for the small models using general texts, the small models are well educated to perform better on the downstream tasks. Experiments on various tasks including language inference benchmarks and Big-Bench-Hard tasks show that the small models after such distillation can surpass or perform comparable to LLMs that are 20x bigger.

Limitations

There is a limitation on the coverage of language models and languages. LLMs such as GPT4 are not included as the teacher model due to the speed of calling API, and some smaller language models such as BLOOM-560M are not included to initialize the student model for the study. The basic reading behavior data and downstream task data are in English only.

Ethics Statement

The code and language models used in this paper are freely downloadable from web. The corpus for generating basic reading behaviors by the teacher model is commonly used in most LLMs pretraining, and is freely released. The downstream task data are also freely downloadable from web.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes.

Priyanka Agrawal, Chris Alberti, Fantine Huot, Joshua Maynez, Ji Ma, Sebastian Ruder, Kuzman Ganchev, Dipanjan Das, and Mirella Lapata. 2023. Qameleon: Multilingual qa with only 5 examples.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Knowledge Discovery and Data Mining*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. A survey on evaluation of large language models.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. 2020. Big self-supervised models are strong semi-supervised learners. In *Advances in neural information processing systems, 33:22243–22255.*

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 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)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *Proceedings of Sinnund Bedeutung 23*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. *ArXiv*, abs/2306.08543.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada. Association for Computational Linguistics.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Fotis Iliopoulos, Vasilis Kontonis, Cenk Baykal, Gaurav Menghani, Khoa Trinh, and Erik Vee. 2022. Weighted distillation with unlabeled examples. In *Advances in neural information processing systems*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Jianquan Li, Xiaokang Liu, Honghong Zhao, Ruifeng Xu, Min Yang, and Yaohong Jin. 2020. BERT-EMD: Many-to-many layer mapping for BERT compression with earth mover's distance. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3009–3018, Online. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. FastBERT: a self-distilling BERT with adaptive inference time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6035–6044, Online. Association for Computational Linguistics.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781, Toronto, Canada. Association for Computational Linguistics.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff nets: Stealing functionality of black-box models. In *CVPR*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

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

Shicheng Tan, Weng Lam Tam, Yuanchun Wang, Wenwen Gong, Shu Zhao, Peng Zhang, and Jie Tang. 2023. GKD: A general knowledge distillation framework for large-scale pre-trained language model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 134–148, Toronto, Canada. Association for Computational Linguistics.

Eric Wallace, Mitchell Stern, and Dawn Song. 2020. Imitation attacks and defenses for black-box machine translation systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5531–5546, Online. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 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*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Peifeng Wang, Aaron Chan, Filip Ilievski, Muhao Chen, and Xiang Ren. 2023. Pinto: Faithful language reasoning using prompt-generated rationales. In *Eleventh International Conference on Learning Representations*.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021a. Want to reduce labeling cost? GPT-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021b. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Qinhong Zhou, Zonghan Yang, Peng Li, and Yang Liu. 2023. Bridging the gap between decision and logits in decision-based knowledge distillation for pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13234–13248, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Model Configuration

Our models are implemented based on Transformers[4] Library. For zero-shot testing, we report results on the development set of all the tasks to be consistent with the work of (Lin et al., 2022).

For probing analysis, we separately add a 4096×4096 linear layer after the output of each layer of the model, and then train it on the data of various tasks. We train the linear layer with learning rate = 0.0003,and max input length = 512. We save the linear layer every 1000 steps.

For unsupervised distillation, the experimental setup is shown in Table 9. In the training process, we set various hyperparameters to balance the loss on multiple types of data. We select the best combination of hyperparameters based on the accuracy on the validation set.

Tasks	Batch_Size	Eval_Steps	Patience	Maximum_Steps
CB	8	50	20	10000
BOOLQ	64	200	10	10000
RTE	32	100	10	10000
SST-2	64	500	5	20000
PAWS-X	128	500	5	20000
XNLI	128	500	5	20000

Table 9: Experimental setup for unsupervised distillation.

A.2 The Results in Big-Bench-Hard

The zero-shot test results of all subtasks in Big-Bench-Hard are reported in Table 10.

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

Tasks	XGLM-564M	XGLM-BRD	XGLM-7.5B	Vicuna-13B
Causal Judgement	51.1	**52.1**	47.9	53.2
Date Understanding	30.1	**32.3**	38.5	64.2
Disambiguation QA	33.0	**35.3**	37.2	33.0
Dyck Languages	19.3	**23.0**	16.7	11.4
Formal Fallacies	**50.4**	50.3	50.2	50.2
Geometric Shapes	10.0	**18.4**	19.5	11.1
Hyperbaton	50.0	50.0	49.9	57.6
Logical Deduction (5 objects)	20.6	20.6	21.6	22.8
Logical Deduction (7 objects)	14.7	**17.0**	15.6	17.4
Logical Deduction (3 objects)	35.3	**37.7**	38.0	39.7
Movie Recommendation	**34.8**	32.2	37.2	27.8
Navigate	50.0	50.0	49.7	51.9
Reasoning About Colored Objects	16.6	**26.5**	25.9	46.2
Ruin Names	29.91	**31.0**	27.2	30.4
Salient Translation Error Detection	**25.7**	25.0	18.9	27.7
Snarks	**51.9**	50.8	51.4	55.8
Sports Understanding	49.1	**50.5**	50.3	49.7
Temporal Sequences	**29.4**	26.6	26.3	29.1
Tracking Shuffled Objects (5 objects)	18.8	**19.8**	19.1	21.0
Tracking Shuffled Objects (7 objects)	13.7	**14.3**	14.4	14.7
Tracking Shuffled Objects (3 objects)	35.3	**37.7**	38.0	39.7
Average	31.9	**33.4**	33.9	35.9

Table 10: The zero-shot results on the subtasks in Big-Bench-Hard.