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 tasks. It shows that the small model can outperform or perform comparable to over 20x bigger LLMs. Analysis reveals that BRD effectively influences the probability distribution of the small model, and has orthogonality to either knowledge distillation or task distillation.

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 LLMs into smaller language models. 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, while task distillation (Chen et al., 2020) usually trains the student models to imitate explicit behaviors of the teacher models.

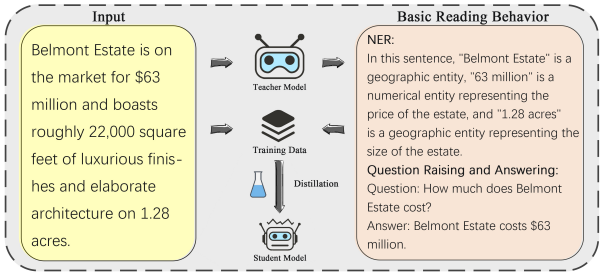

Figure 1: The illustration of BRD process.

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. Such basic reading education on every sentence is important before application on downstream tasks, while is always neglected in both knowledge distillation and task distillation.

The benefits of BRD are two-fold: Firstly, beyond only using texts for training next token prediction, BRD educates the student model to deeply understand the texts via interactions. All available data such as web mined corpora can be extended to be magnitudes larger by BRD, breaking the data scale and diversity limitation criticized by Gudibande et al. (2023) in the task distillation. Secondly, BRD also avoids the implicit nature of knowledge distillation which imitates latent features such as logits (Hinton et al., 2015), hidden lay-

ers (Jiao et al., 2020), and attention maps (Li et al., 2020; Wang et al., 2021b). Such implicit nature leads to the deficiency of learning interpretability, while BRD demonstrates explicit reading behaviors that are easy to interpret.

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 Google Bigbench tasks, show that although the student model is trained on the general data that is irrelevant to the downstream tasks, it can inherit teacher model abilities, leading to excellent downstream performances better than or comparable to those of larger models. Furthermore, after this basic education of the student model on general sentences, we fine-tune the student model for downstream tasks, and find that the basic reading education leads to further improvement 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 compute the cross entropy between the student model and the teacher model, and find that the student model distribution approaches closer to the teacher model distribution after BRD, leading to better performances than non-educated ones. In summary, the main contributions are:

- We propose BRD that educates the student model to imitate basic reading behaviors of the teacher model.
- 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 analysis reveals that BRD can drive the student model distribution closer to the teacher model distribution, resulting in significant performance improvements.

2 Related Works

There are mainly two streams of distillation approaches: knowledge distillation and task distillation. Knowledge distillation focuses on teaching implicit features inside the teacher model, while task distillation focuses on teaching explicit behaviors of the teacher model on downstream tasks. In addition, we introduce intrinsic task pre-training that focuses on intrinsic task data derived from the training plain texts.

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 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., 2023a). 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 the 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 downstream tasks, while our BRD mainly focuses on general sentences unrelated to any specific downstream tasks, and the basic reading behaviors in BRD are basic education resource

not aiming at any specific applications.

2.3 Intrinsic Task Pre-training

Different to task distillation that utilizes downstream task data, intrinsic task pre-training uses general training set to synthesize task data. PICL is a framework focusing on intrinsic tasks (Gu et al., 2023b). It posits that many paragraphs in the training set documents contain intrinsic tasks such as sentiment analysis, and retrieves paragraphs of the same intrinsic task to compose in-context learning examples, but its retriever is trained on 37 downstream tasks, which are opposite to the "intrinsic task" nature, and limit the scale and diversity of the composed task data. In comparison, our BRD does not refer to any downstream tasks, and focuses on the contents of the training set texts, thus keeping more freedom in curating the task data. In addition, PICL aims to train the in-context learning ability, while BRD is for the model distillation. The intrinsic task data in PICL may not have task labels since the original paragraphs do not necessarily have both task queries and answers simultaneously, e.g., a sentiment expression paragraph may not explicitly states its positive or negative label for the sentiment analysis task. In comparison, BRD always gets education queries and responses.

Zhang et al. (2023) propose a similar intrinsic task pre-training approach that transforms fragmented sentences from babyLM training set into a cohesive paragraph (Warstadt et al., 2023). Their task is quite challenging to accomplish since the sentences in the training set are sampled from diverse resources, and lack strong semantic ties with each other, resulting in the hardness of composing a cohesive paragraph. Such fiction data generation are different to our BRD approach, which generates solid basic education data on reading activities.

3 Approach

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, question is constrained to be about the content, structure, or attitude of the input sentence. The question 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> 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 to be consistent with the usual language model pre-training that utilizes 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 rais-

Perform named entity recognition on a given sentence without recognizing personal pronouns in the input sentence as human names. **Enter a sentence:** Barack Obama was the 44th President of the United States. **Output:** 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:** 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:** In this sentence, "Eiffel Tower" is a landmark name entity, and "Paris" and "France" are geopolitical entities. **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:**
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 the bottom.

ing 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

For predicting the answers of the downstream tasks when testing the student model, 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 the input to the student model, y denotes the candidate answer for x_{prompt}, and n is the total number of words in y. We select y with the maximal $\bar{P}$ as the final answer for x_{prompt}. This average computation is to cover tasks such as Google BIG-bench [1] (bench authors, 2023), whose candidate answers are phrases/sentences rather than single words.

[1] https://github.com/google/BIG-bench

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 for comparison. In BRD, we use one million passages from CC-100 corpus to collect the basic reading data generated by Vicuna-13B.

4.1 Baselines

We consider three baselines in our experiments:

- Knowledge distillation (KD): We use two KD models released in Gu et al. (2023a)[4] for the comparison. One is the standard KD (SKD) that uses teacher distribution to supervise the student model. The other is MiniLLM that uses reverse Kullback-Leibler divergence for KD.
- Task distillation (Wang et al., 2021a; Iliopoulos et al., 2022): The teacher model generates the answers given the downstream task inputs,

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

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

[4] https://github.com/microsoft/LMOps/tree/main/minillm

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:".
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.
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.
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 the bottom.

and these generated pseudo answers are used to supervise the student model.

- Supervised Fine-tuning(SFT): Directly fine-tunes the student model on the downstream tasks supervised by the gold labels.

4.2 Evaluation

We adopt a spectrum of downstream tasks for the evaluation, 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 Google BIG-bench(bench authors, 2023). In Google BIG-bench tasks, we only consider multiple choice QA tasks which have the fixed answers easy for the evaluation, resulting in a total of 73 tasks. The results are evaluated by the accuracy of the predicted answers. The prompts for the downstream tasks are presented in the appendix A.2.

4.3 Results

The main results are grouped into three parts as shown in Table 3. The top part presents the accuracies of the original models, including the teacher model Vicuna-13B, the student model XGLM-564M, the large model XGLM-7.5B which has same origin to the student model, plus an extension model XGLM-564M-FURTHER, which further trains the student model on the original one million passages from CC-100 corpus. The number of the further training steps is set 18,000.

The middle part and the bottom part list the accuracies of various distillation or fine-tuning approaches under two scenarios: without downstream task supervision and with downstream task supervision, respectively. The difference between the two scenarios is the availability of the downstream task gold answers.

Results Without Downstream Task Supervision. In this scenario, the downstream task gold answers are not available. It is further divided into two conditions. One is the blind test setting, in which any task training set data is NOT accessible. It is for applications of the student model on fairly new tasks. The other is the relaxed test setting, in which only the training set input data (without gold answers) are accessible. It is for applications on tasks that manual labeling for the training set input data is not available.

- In the blind test setting, we compare our XGLM-BRD with the two released KD works: SKD and MiniLLM. In the multiple student models of the two KD works, we select their GPT-2 760M version student models for the comparison due to the similar model size. The results show that XGLM-BRD performs significantly better than SKD and MiniLLM in most tasks, demonstrating that XGLM-BRD has better generalization ability to various unseen tasks. We also combine BRD with the two KD works, and the results are listed in the orthogonal analysis section 5.1 and Table 4.

  Regarding the comparison between XGLM-BRD and XGLM-564M, BRD significantly improves the performance of the small model, indicating that basic reading education does enhance the ability of the small

Model	Task							Average
	XNLI	RTE	CB	PAWS-X	BOOLQ	SST-2	BIG-bench-Avg	
Vicuna-13B	59.1	78.3	71.4	62.9	84.3	81.5	35.6	67.6
XGLM-7.5B	36.6	50.9	60.7	56.8	57.2	69.5	34.3	52.34
XGLM-564M	35.5	46.2	53.6	51.3	51.2	63.9	34.0	48.0
XGLM-564M-FURTHER	34.9	46.6	51.8	51.6	51.5	59.4	34.0	47.1
Without Downstream Task Supervision								
SKD	33.7	53.8	51.8	43.0	49.1	60.7	34.2	46.6
MiniLLM	34.2	**58.1**	**73.2**	44.1	55.9	62.4	34.6	51.8
XGLM-BRD	**36.2**	53.8	58.9	**56.7**	**61.0**	**78.1**	**34.8**	**54.2**
TaskDistillation	57.1	58.1	60.7	64.8	74.8	77.2	41.6†	62.0
XGLM-BRD^2	**59.2**	**62.5**	**82.1**	64.8	**75.0**	**81.9**	**44.1**†	**67.1**
With Downstream Task Supervision								
SFT	81.4	67.1	83.9	**92.4**	77.5	91.5	68.3†	80.3
XGLM-BRD^2-SFT	**81.6**	**69.3**	**91.1**	91.5	**77.8**	**92.2**	**69.1**†	**81.8**

Table 3: Main results of the teacher models, student models, and various distillation and fine-tuning approaches. Unless otherwise specified, the student models are all initialized by XGLM-564M. BIG-bench-Avg is the accuracy averaged over the 73 bench tasks(† denotes the averaged accuracy on the reduced set of BIG-bench tasks), and detailed accuracies are reported in the appendix A.3.

model. Moreover, XGLM-564M-FURTHER performs much worse 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. XGLM-BRD also approaches or even surpasses XGLM-7.5B, which is 15x bigger, on the downstream tasks. There is still a gap between XGLM-BRD and the teacher model Vicuna-13B, but this gap is significantly reduced or disappeared when we conduct relaxed test.

- In the relaxed test setting, we compare our BRD with the task distillation approach, which uses the teacher model to generate pseudo answers on the task training set for supervising the student model XGLM-564M. Because BIG-bench tasks do not divide training, tuning, and test sets, we only consider tasks each of whom has more than 2K instances in the relaxed test, and finally select tasks that rank top-5 according to the number of instances as the reduced set of BIG-bench tasks(denoted by † in Table 3). For each task, we save ten percent of instances as test set, ten percent of instances as tuning set, and other instances as training set. 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 continual training the new student model, denoted as XGLM-BRD^2. The results show that the task distillation approach establishes a strong baseline that significantly outperforms both XGLM-564M and KD models, demonstrating 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 by XGLM-BRD^2.

  When comparing XGLM-BRD^2 with the teacher model Vicuna-13B, it shows that XGLM-BRD^2 outperforms Vicuna-13B in some tasks, and in the other tasks, the performance gap is significantly reduced. This comparison proves the effectiveness of BRD, that leads to comparable or superior performance to the 26x bigger teacher model.

Results With Downstream Task Supervision. In this scenario, the downstream task gold answers are available. We compare BRD with SFT, which fine-tunes the student model XGLM-564M based on the task supervision data. Table 3 shows that with the gold supervision, SFT significantly improves the ability of the student model, and beats the 26x bigger model Vicuna-13B with a large margin in certain tasks. In comparison to this strong baseline, we conduct BRD on the SFT data to get the basic reading data of the tasks, then continue training XGLM-BRD on this basic reading data. The trained model is denoted as XGLM-BRD^2-SFT. The results show that XGLM-BRD^2-SFT surpasses SFT in most tasks, demonstrating the effectiveness of the basic reading education for the student model when the downstream task supervision is available.

Model	Approach	Task						
		XNLI	RTE	CB	PAWS-X	BOOLQ	SST-2	Average
GPT-2 120M	SKD	35.9	44.4	57.1	52.1	47.6	68.8	51.0
	+BRD	**37.0**	**54.5**	**66.1**	**56.8**	**62.2**	**76.9**	**58.9**
	MiniLLM	**35.9**	48.0	67.9	**57.0**	53.1	53.8	52.6
	+BRD	35.1	**51.6**	**71.4**	48.5	**60.9**	**79.1**	**57.8**
GPT-2 340M	SKD	33.6	46.9	51.8	**54.2**	48.8	63.3	49.8
	+BRD	**34.2**	**55.2**	**67.9**	53.8	**64.6**	**78.1**	**59.0**
	MiniLLM	**32.8**	46.6	50.0	**57.0**	56.9	55.3	49.0
	+BRD	32.7	**56.3**	**67.9**	53.4	**64.1**	**74.4**	**58.1**
GPT-2 760M	SKD	33.7	**53.8**	51.8	43.0	49.1	60.7	48.7
	+BRD	**34.7**	52.3	**64.3**	**56.4**	**62.1**	**68.5**	**56.7**
	MiniLLM	34.2	**58.1**	73.2	44.1	55.9	62.4	54.7
	+BRD	**35.2**	52.0	**76.8**	**50.8**	**60.5**	**67.3**	**57.1**
XGLM-564M	TaskDistillation	57.1	58.1	60.7	**64.8**	**74.8**	77.2	65.5
	+BRD	**58.1**	**61.0**	**71.4**	63.1	74.4	**81.1**	**68.2**

Table 4: Results of combining BRD with various distillation approaches. The models for initializing the student models are listed in the model column.

	XNLI	RTE	CB	PAWS-X	BOOLQ	SST-2	BIG-bench-Avg
XGLM-564M	461.7	**64.8**	13.2	1065.6	1098.7	55.6	145.1
XGLM-BRD	**407.5**	66.2	**12.8**	**892.4**	**1001.1**	**37.0**	**112.2**

Table 5: Cross entropy between the distributions of the teacher model and small models. The lower the better for measuring the consistency.

5 Analysis

5.1 Orthogonality of BRD to Knowledge Distillation and Task Distillation

Since BRD focuses on basic reading education for the student model without referring to any implicit model features or downstream tasks, it is orthogonal to either knowledge distillation or task distillation. So, we combine BRD with knowledge distillation by further training the student model of knowledge distillation on our general basic reading data, or combine BRD with task distillation by further training the student model of task distillation on the basic reading data of the downstream tasks. Table 4 lists the combination results.

It shows that combining BRD in most cases significantly improves the performances of the two distillation approaches, which proves the orthogonality of BRD to either knowledge distillation or task distillation.

5.2 Effectiveness Verification Based on Cross Entropy Evaluation

BRD educates the student model via explicit basic reading behaviors. We study if such education can effectively influence the probability distribution of the student model. We compute the cross entropy, which is often used to measure the consistency between the teacher distribution and the student distribution, for the teacher model Vicuna-13B and the student model XGLM-BRD:

$$-\sum_{i=1}^{N} p(y)\log q(y')$$

where p is the teacher model probability, q is the student model probability, y and y' are subword sequences of the same text according to the teacher model and the student model, respectively. N is the number of the texts. Since y and y' have different lengths, we set p and q as sequence-level probabilities averaged over y and y', respectively. We use the texts from the downstream tasks for computing the cross entropy. For the considered 73 tasks in BIG-bench, we randomly choose 1K instances from each task for the computation, and report the cross entropy averaged over the tasks. We include the original XGLM-564M to compute q for comparison.

Table 5 shows the comparison result in the blind test. Lower cross entropy means better consistency between the teacher model and the student model. It shows that on most downstream tasks, XGLM-BRD approaches more closer to the teacher model than the original XGLM-564M does, demonstrating significant advantage in shaping the student model probability distribution towards that of the teacher model.

	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
PassageLevel	**36.2**	53.8	**58.9**	**56.7**	**61.0**	**78.1**	**57.5**
SentenceLevel	34.1	**55.6**	55.4	53.6	58.9	76.0	55.6

Table 6: The comparison between the passage level training and the sentence level training evaluated by the blind test.

	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 relaxed test.

5.3 Comparison to Sentence Level Training

In building the BRD training data presented in section 3.2, we divide a passage into sentences, then annotate each sentence with basic reading behaviors by using the teacher model, and finally compose all sentences and their annotations into a passage according to the original sentence order. To check whether this passage level training has the positive effect, we abandon the last composing step and leave the sentences and their annotations unordered. Then we conduct the sentence level training on this dataset to compare with the passage level training. Table 6 presents the comparison result.

It shows that the sentence level training generally performs worse than the passage level training. Since the downstream tasks are mostly the tasks with multiple sentences as input, the passage level training is more suitable for the downstream tasks than the sentence level training due to its multiple sentence training nature.

5.4 Ablation 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 relaxed test. It shows that deleting the QRA data impacts the performance more significantly than deleting the NER data in most tasks. QRA focuses on the sentence understanding, thus contributing more in the basic reading education.

The coordination between NER and QRA is related to the multi-task learning (Chen et al., 2024) that boosts the model ability through training on multiple tasks with potential generalization to other tasks. Different to the multi-task learning that predefines a fixed set of tasks, BRD focuses only on the basic reading education that has flexible contents changing from sentence to sentence. This flexibility empowers the distilled model to perform well on various downstream tasks.

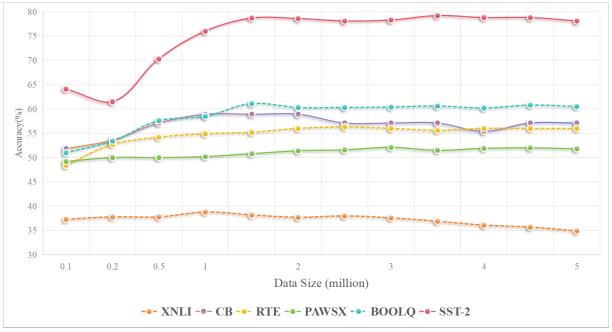

Figure 2: The performance curve along with different BRD data sizes (in million passages).

5.5 The Impact of BRD Data Size

We investigate how performance varies along with different BRD data sizes in the blind test. Figure 2 shows the curve. Most tasks exhibit a steady improvement as BRD data gets bigger, and the performance plateaued when BRD data size arrives at more than one 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 on general texts, the small models are well educated to perform better on the downstream tasks. Experiments on various tasks including language inference benchmarks and Google Big-Bench tasks show that the small models after such distillation can surpass or perform comparable to LLMs that are 20x bigger. Verification by the cross entropy shows that such basic reading education can drive small model distribution closer to its teacher model distribution, leading to better performances than non-educated ones. Analysis also reveals that BRD has orthogonality to either knowledge distillation or task distillation.

Limitations

In the distillation approach, we acknowledge certain limitations in the coverage of language models. We only use Vicuna-13B as our teacher model due to its high efficiency in generating large volume of texts. Calling the recent proprietary LLMs through API or using larger released LLMs incurs high cost in time and deployment for the massive distillation. It represents an area for potential future exploration to provide a more comprehensive understanding of using larger LLMs as the teacher model for the distillation.

Ethics Statement

We honor the Code of Ethics. We do not use any private data or non-public information in this work. The 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. The distillation process does not involve any personally sensitive information.

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. *Preprint*, arXiv:2306.13649.

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. *Preprint*, arXiv:2211.08264.

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

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

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. *Preprint*, arXiv:2307.03109.

Shijie Chen, Yu Zhang, and Qiang Yang. 2024. Multi-task learning in natural language processing: An overview. *ACM Comput. Surv.*, 56(12).

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*.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023b. Pre-training to learn in context. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4849–4870, Toronto, Canada. Association for Computational Linguistics.

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. *Preprint*, arXiv:2305.15717.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

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, and 2 others. 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. *Preprint*, arXiv:2306.02707.

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.

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.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023. Call for papers – the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2301.11796.

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.

Zheyu Zhang, Han Yang, Bolei Ma, David Rügamer, and Ercong Nie. 2023. Baby's CoThought: Leveraging large language models for enhanced reasoning in compact models. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 158–170, Singapore. 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, and 3 others. 2023. A survey of large language models. *Preprint*, arXiv:2303.18223.

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 Training Configuration

The student model is trained 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. Four A100 GPUs are used in both the data synthesis and the distillation training.

A.2 Prompts for The Downstream Tasks

The prompt templates for the downstream tasks are listed in table 8. For the 73 tasks in BIG-bench, we follow the general instruct with the task prefix and input as the prompt.

A.3 Detailed Accuracies on BIG-bench

Figure 3 presents the accuracy comparison between the distilled model XGLM-BRD and the baseline model XGLM-564M on the 73 tasks in BIG-bench in the blind test. The results of ten tied tasks are not listed in the figure. It shows that XGLM-BRD improves the performances on about 2/3 tasks, demonstrating better ability generalized to a wide range of tasks than the baseline.

In the relaxed test and the setting with the downstream task supervision, we select tasks that rank top-5 according to the number of instances for the sufficiency consideration of dividing training, tuning, and test sets on these tasks. The five tasks are movie dialogue, formal fallacies and syllogisms with negation, Shakespeare dialogue, VitaminC, and WinoWhy. Table 9 presents the results on this reduced BIG-bench set. It shows that BRD models perform superior to the corresponding baselines no matter the supervisions are available or not.

A.4 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 4 presents the results of probing XGLM-564M and XGLM-BRD in the blind test. 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.

A.5 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

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 8: The prompt templates for the downstream tasks.

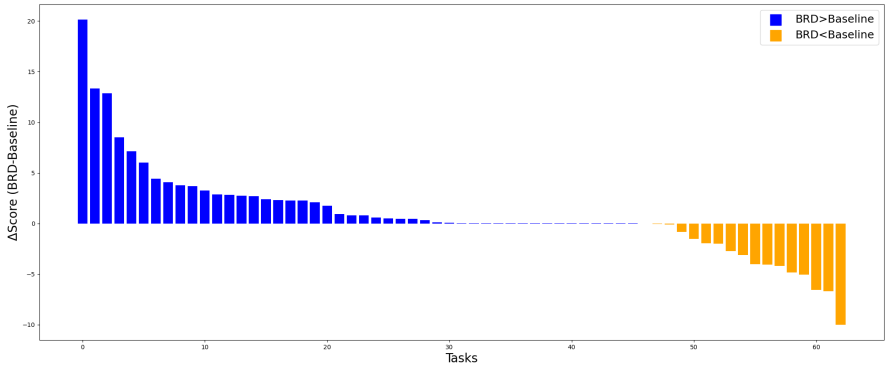

Figure 3: The accuracy comparison between BRD and the baseline on BIG-bench tasks.

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 10 shows the result in the blind test. Excluding the sentiment-related data does influence SST-2 performance significantly, resulting in a decrease of 4 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 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.

Model	Task					Average
	MovieDialog	FormalFallacies	ShakespeareDialogue	VitaminC	WinoWhy	
Relaxed Test						
TaskDistillation	46.7	50.0	42.2	13.6	55.2	41.6
XGLM-BRD^2	**50.1**	50.0	**49.8**	13.8	**56.9**	**44.1**
With Downstream Task Supervision						
SFT	69.2	**69.9**	69.3	55.9	76.9	68.3
XGLM-BRD^2-SFT	**70.6**	69.3	**70.2**	**56.0**	**79.1**	**69.1**

Table 9: Results on the reduced BIG-bench set.

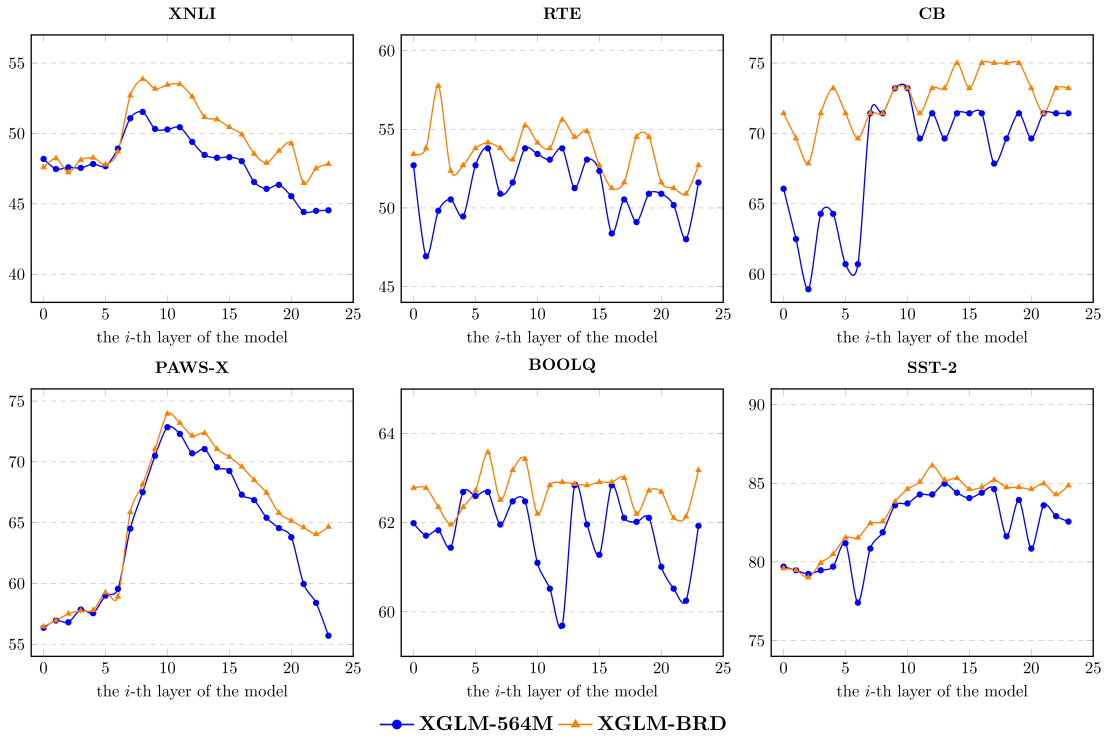

Figure 4: The results of probing XGLM-564M and XGLM-BRD layer-wisely on the downstream tasks in the blind test. The horizontal axis represents the specific layer in the model, and the vertical axis is the prediction accuracy (%) for each task.

	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	36.2	53.8	**58.9**	**56.7**	**61.0**	**78.1**	57.5
−SentData	**39.2**	**54.5**	57.1	51.5	59.1	74.2	55.9

Table 10: The result of training XGLM-BRD based on the data excluding the sentiment-related questions and answers, denoted by −SentData, in the blind test.