Prompting to Distill: Boosting Data-Free Knowledge Distillation via Reinforced Prompt

Xinyin Ma[1] , **Xinchao Wang**[2] , **Gongfan Fang**[1] , **Yongliang Shen**[1] and **Weiming Lu**[1*]
[1]College of Computer Science and Technology, Zhejiang University
[2]Department of Electrical and Computer Engineering, National University of Singapore
{maxinyin,fgf,syl,luwm}@zju.edu.cn, xinchao@nus.edu.sg

Abstract

Data-free knowledge distillation (DFKD) conducts knowledge distillation via eliminating the dependence of original training data, and has recently achieved impressive results in accelerating pre-trained language models. At the heart of DFKD is to reconstruct a synthetic dataset by inverting the parameters of the uncompressed model. Prior DFKD approaches, however, have largely relied on hand-crafted priors of the target data distribution for the reconstruction, which can be inevitably biased and often incompetent to capture the intrinsic distributions. To address this problem, we propose a prompt-based method, termed as PromptDFD, that allows us to take advantage of learned language priors, which effectively harmonizes the synthetic sentences to be semantically and grammatically correct. Specifically, PromptDFD leverages a pre-trained generative model to provide language priors and introduces a reinforced topic prompter to control data synthesis, making the generated samples thematically relevant and semantically plausible, and thus friendly to downstream tasks. As shown in our experiments, the proposed method substantially improves the synthesis quality and achieves considerable improvements on distillation performance. In some cases, PromptDFD even gives rise to results on par with those from the data-driven knowledge distillation with access to the original training data.

1 Introduction

Knowledge distillation (KD) has recently emerged as a popular technique for model lightening, finding its applications in a wide spectrum of domains. In some scenarios, however, the training data per se is unavailable, which calls for a variant of KD, known as *data-free knowledge distillation* (DFKD). Despite the more challenging problem setting, DFKD has also received increasing attention from the community due to its practical nature [Micaelli and Storkey, 2019; Yin *et al.*, 2020; Ma *et al.*, 2020; Fang *et al.*, 2021].

Method	Synthesized Sample
AS-DFD	... indonesia (logctricdae beings eantarize whilellus civilian indiana influencesidium landed gaddafi ...
PromptDFD	***[Topic Prompt]*** If the obama administration is ***[Content]*** planning to release all of Obama's "red-book" travel documents, but it's unclear if they include information on Obama's first foreign trip. The documents, as leaked yesterday by ABC News ...

Table 1: Examples of pseudo samples generated for the category "World" in AG News. We use nearest neighbor search to approximate the tokens for synthesized embeddings generated by AS-DFD to make the synthesized embeddings readable.

In the literature, [Lopes *et al.*, 2017] initially introduces the idea of data-free knowledge distillation, and follows a learning-by-generating paradigm to craft synthetic data that approximates the original distribution. The key step towards DFKD is to generate high-quality and diverse pseudo samples from a distribution similar to that of the original training set. Once the pseudo samples have been generated, conventional KD algorithms [Sun *et al.*, 2019; Jiao *et al.*, 2020; Hou *et al.*, 2020] can be easily deployed to transfer knowledge from the teacher to the student network.

Prior DFKD algorithms in natural language processing [Ma *et al.*, 2020; Rashid *et al.*, 2021] focused on synthesizing pseudo samples from the teacher's parameters through model inversion [Mahendran and Vedaldi, 2015], where a batch of synthetic utterances or a sentence generator is optimized under the restrictions of some human-crafted distributional priors. The confidence-based prior is the most widely used human-crafted prior for sentence synthesizing. For example, AS-DFD [Ma *et al.*, 2020] aims to find some pseudo samples that can produce high-confidence predictions when fed to the teacher. As shown in Table 1, despite that AS-DFD indeed generates some task-related keywords or phrases that are related to the task, these utterances are still unnatural and of low-quality without correct semantic and syntax.

The above issue is in part caused by the insufficiency of the confidence-based prior in regularizing the synthesized samples. For classification problems, the teacher's predictions mainly contain high-level information like the category of sentences, from which the low-level information (corre-

[*] Corresponding author

lation between words, syntax, etc.) is removed. Therefore, the confidence-based prior alone cannot provide robust regularization for the low-level features to make the whole sentences plausible, and leads to some out-of-distribution samples known as "rubbish samples" [Goodfellow *et al.*, 2015]. These samples are far from the underlying distribution of the original training set and are hardly helpful for distillation. Existing methods address this problem by introducing more regularizations into data synthesis. For instance, [Yin *et al.*, 2020] adds extra restrictions using the stored mean and variance statistics in Batch Normalization to improve the quality of synthesized samples. However, these methods are usually inapplicable for language models as they heavily rely on some special architectures like Batch Normalization that are not used in transformer-based models. Therefore, how to introduce more priors to regularize the synthesized sentence is still a challenging problem for DFKD.

To address these problems, we resort to learning natural priors in pre-trained generative language models, and develop a prompting-to-distill framework for DFKD which we term as PromptDFD. Generative language models pre-trained on a large scale of unlabeled corpora grasp the generic prior for languages, which can be incorporated into the synthesis to alleviate the quality issue caused by confidence-based prior. Specifically, the generation of synthesized samples is divided into two steps: 1) a prompt step that explores the language prior in the pre-trained language model in a controllable manner by a task-related topic prompt, and 2) a completion step to expand the prompt into a complete utterance that contains more informative details. To enforce the prompt to be tightly related to the target task when no original data is available, we propose an adversarial reinforced prompt that takes the feedback from both the teacher and the student model into account, simultaneously satisfying the topic and the difficulty requirement. As shown in the experiment, PromptDFD significantly improves the synthesis quality and boosts the student's performance. Extensive experiments demonstrated that the synthesized samples generated by PromptDFD not only contain task-related words, but are meaningful and follow grammatical and linguistic conventions.

In summary, the contribution of this paper can be summarized as follows:

- We propose a novel prompting-to-distill framework for data-free knowledge distillation, through introducing controllable language prior into the synthesized sample.
- We propose a reinforced topic prompter with an adversarial reward. With no access to the original training set, the reinforced prompter can dynamically adjust the topic and the difficulty of the synthesized sample to satisfy the topic constraint of the target tasks
- PromptDFD achieves superior performance on four text classification datasets with different network settings. In some datasets, PromptDFD even yields results on par with those of data-driven KD methods.

2 Related Work

Data-driven knowledge distillation. Knowledge distillation (KD), proposed by [Hinton *et al.*, 2015], uses one or more large and cumbersome models (called the teacher model) to train a lightweight model (called the student model), where the student is required to mimic the softened response of the teacher. DistillBERT [Sanh *et al.*, 2019] was the first work to adopt KD on BERT, after which PatientKD [Sun *et al.*, 2019], Tinybert [Jiao *et al.*, 2020], Universal-KD [Wu *et al.*, 2021] and other algorithms have developed other feature alignment criteria, such as intermediate representations, attention maps, extracted relations, etc. Some works [Hou *et al.*, 2020; Chen *et al.*, 2020] focuses on how to dynamically select the most valuable structure of student models during knowledge distillation. Those KD algorithms can be applied as alternative ways to distill knowledge when synthesized samples are obtained.

Data-free knowledge distillation. Existing DFKD algorithms in computer vision adopt the idea of model inversion [Mahendran and Vedaldi, 2015] to reconstruct the training samples from the parameters of the model. Under the constraint of the probabilistic output [Nayak *et al.*, 2019] or historical statistics stored in Batch Normalization [Yin *et al.*, 2020], the training samples [Lopes *et al.*, 2017] or pseudo-sample generators [Chen *et al.*, 2019] are optimized to explore the underlying distribution of the training data stored in the teacher network. [Micaelli and Storkey, 2019] quantifies the degree of belief matching between the teacher and student. [Fang *et al.*, 2021] proposes to solve the problem of inter-sample pattern collapse using contrast learning. As for methods designed for natural language processing, [Ma *et al.*, 2020] generates pseudo-embeddings to address the discrete problem of tokens, and [Rashid *et al.*, 2021] adopts Gumbel-Softmax to pass the gradient to the generator. However, the only constraint that is accessible in BERT is the class prior, which is too weak to reconstruct the underlying distribution.

3 Problem Settings

The goal of data-free knowledge distillation is to reduce redundant parameters in the network without relying on the original dataset. We first start from the data-driven scenario.

Data-driven knowledge distillation. Model compression can be achieved using knowledge distillation, in which the original model is called the teacher network $\mathcal{T}$ and the compressed model is called the student network $\mathcal{S}$. Given a sample $x = [t_1, t_2, \dots, t_l] \in \mathcal{X}$ and its pre-defined label y, where t_i is the i-th token in the sample and l is the length of the sentence, the objective of KD can be defined as follows:

$$\mathcal{L}_{KD} = \alpha \cdot \mathcal{L}_{CE}(\mathcal{S}(x), y) + (1-\alpha)\mathcal{L}_{KL}(\mathcal{T}(x), \mathcal{S}(x)) \quad (1)$$

where $\mathcal{T}(x)$ and $\mathcal{S}(x)$ are the predicted class probabilities on input x. α balances between two criteria, i.e., Cross-Entropy loss $\mathcal{L}_{CE}$ and Kullback-Leibler divergence loss $\mathcal{L}_{KL}$.

Data-free knowledge distillation. Data-free knowledge distillation is severely hampered by the lack of access to the initial distribution of data $\mathcal{X}$, and as a consequence, supervisory signals and predefined labels are not available. Thus, constructing the synthesized sample $\hat{x} \in \hat{\mathcal{X}}$ that can reflect the distribution of $\mathcal{X}$ is the key to the data-free knowledge distillation. After the transfer set $\hat{\mathcal{X}}$ is rebuilt, knowledge distillation techniques can be applied to minimize $\mathcal{L}_{KD}$.

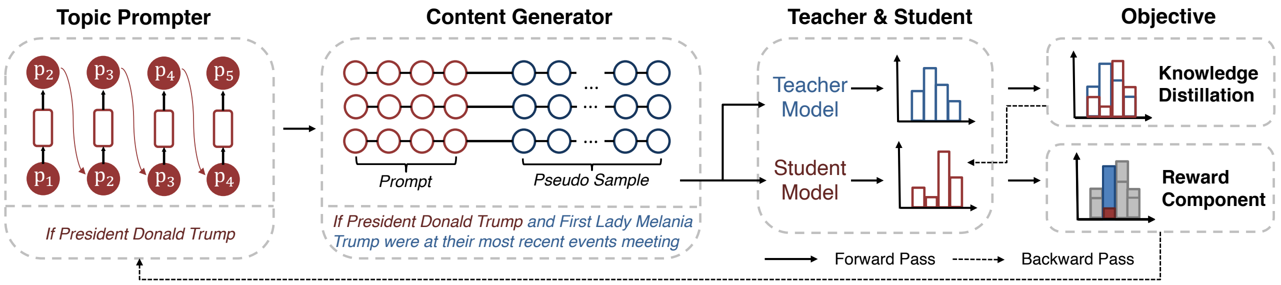


Figure 1: The illustration of the proposed PromptDFD. Synthesizing a sample comprises two steps: generating a topic prompt and expanding the prompt into a full sentence. The synthesized utterance is then used to train the student model with a conventional KD objective and adjust the generation of the topic prompter with an adversarial reward component.

4 Method

4.1 Prompting to Distill

Model inversion with the one-hot class prior fails to constrain the low-level correlation in context, resulting in the semantic confusion of the utterance. The auto-regressive pre-trained language models, such as GPT2, are trained on large-scale corpora to capture these informative co-occurrences. Inspired by [Shin *et al.*, 2020] and [Li and Liang, 2021], our intuition is to query a pre-trained generative language model by prompt engineering and explore the language prior of the language model in a controllable manner.

Step 1: Prompt step. A prefix prompt is used to restrict the topic of the entire synthesized sentence, ensuring that the synthesized sentences are assigned some topics that are relevant to the original training set. Specifically, the topic prompt $P_{1:n} = (p_1, p_2, \ldots, p_n)$, where n is the length of the prompts, contains a shortlist of introductory words and can be considered as the controller for the pre-trained generative LM in step 2. We will discuss the design of topic prompts in section 4.2.

Step 2: Completion step. Based on the generated prompt, we adopt a pre-trained auto-regressive generator, named the content generator $\mathcal{G}_c$, to extend the prompt and explicitly display the knowledge embedded in the pre-trained LM. To avoid the knowledge in the pre-trained LM being disturbed when tuning, we freeze the parameter $\theta_\mathcal{G}$ of $\mathcal{G}_c$. The goal of $\mathcal{G}_c$ is to extend the topic prompt $P_{1:n}$ and generate the content $\hat{x}$, which behaves as the synthesized sample in knowledge distillation. Given a half-finished sequence $(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_i)$ generated by $\mathcal{G}_c$, the auto-regressive LM computes the hidden states based on the past tokens and the prompt:

$$\begin{aligned} h_i &= \mathcal{G}_c([p_1, \ldots, p_n, \hat{x}_1, \ldots, \hat{x}_i]; \theta_\mathcal{G}) \\ \Pr(\hat{x}_{i+1} | P, \hat{x}_{\leq i}) &= \mathrm{softmax}(W h_i) \end{aligned} \tag{2}$$

where h_i is the hidden state for the temporary token and W projects the hidden state into the distribution space of the tokens. For $\mathcal{G}_c$, we use pre-trained GPT2 [Radford *et al.*, 2019] for content filling. When decoding, a top-k nucleus sampling [Holtzman *et al.*, 2020] is used to encourage the diversity of pseudo samples.

4.2 Prefix Prompt for Topic Controlling

Hand-crafted prompt. We start with a straightforward way of designing a topic prompt by manually selecting some keywords that are pertinent to the task. Table 2 contains a few templates that are designed manually. While hand-crafted prompts are capable of guiding the generation of pseudo-samples within a given subject area, there are two major inherent flaws: it is difficult to determine the optimal way to design prompts, and the number of hand-crafted prompts is limited. Though decoding with sampling can provide some variability and randomness in the generation process, the content of the synthesized sample is restricted to limited information.

Datasets	Hand-crafted Prompt
AG News	A latest [Category] news
DBPedia	A document about [Category]
IMDb	A [Category] movie review
SST-2	[Category] sentence:

Table 2: Manually designed prompts for prefix prompting. [Category] represents the name of the category, such as positive or negative for IMDB, world, sports, business or science for AG News.

Reinforced prompt. To solve the problem encountered when designing prompts, we propose to generate prompts automatically. We train a topic prompter $\mathcal{G}_p$, which can be tuned to generate topic prompts that are closely related to the domain of the original dataset.

The generation of prompt $P_{1:n}$ can also be defined as a left-to-right sequence generation problem. Suppose that $P_{1:t} = (p_1, \ldots, p_t)$ has been generated, it's desired to infer the probability distribution of the next token $\Pr(p_{t+1} | P_{1:t})$. Unlike the training objective in the sequence generation problem in general, in DFKD there is no immediate and ground-truth label for $\Pr(p_{t+1} | P_{1:t})$. Furthermore, the teacher model may take a different tokenization method with the content generator $\mathcal{G}_c$, leading to the mismatch in the subwords. Thus, previous gradient-based methods that synthesize embeddings or leverage the Gumbel-Softmax cannot be applied.

We propose a reinforced prompter, which tunes itself through the feedback from both the teacher and the student model. Suppose that the state s_t for the current moment t is defined as the generated sequence $P_{1:t}$ and the action a_t as the selection of the next token p_{t+1}. The selecting policy π for the next action is inferred by the prompt generator $\mathcal{G}_p$with the input being the generated sequence and the output being the probabilities of each token as follows:

$$a_t \sim \pi\left(s_t, \phi_\mathcal{G}\right) = \mathcal{G}_p(p_{t+1} | P_{1:t}, \phi_\mathcal{G}) \tag{3}$$

$\phi_\mathcal{G}$ represents the parameter in $\mathcal{G}_p$. It is assumed that the transition is deterministic and that no intermediate reward is given before the prompt is generated. With the discounted factor set to 1, the action-value function $Q_{\phi_\mathcal{G}}(s_t, a_t)$ equals to the state-value function of the next state $V_{\phi_\mathcal{G}}(s_{t+1})$. To maximize $J(\phi_\mathcal{G}) = V_{\phi_\mathcal{G}}\left(s_1\right)$ when $Q_{\phi_\mathcal{G}}(s_t, a_t) = V_{\phi_\mathcal{G}}(s_{t+1})$, following [Yu *et al.*, 2017; Sutton *et al.*, 1999], the gradient of $J(\phi_\mathcal{G})$ can be derived as:

$$\begin{aligned}\nabla_{\phi_\mathcal{G}} J(\phi_\mathcal{G}) &= \sum_{t=1}^{n-1} \mathbb{E}_{P_{1:t} \sim \mathcal{G}_p}\left[\sum_{a_t} \nabla_{\phi_\mathcal{G}} \mathcal{G}_p(a_t \mid s_t) Q(s_t, a_t)\right] \\ &\simeq \sum_{t=1}^{n-1} \mathbb{E}_{a_t \sim \pi\left(s_t, \phi_\mathcal{G}\right)}\left[Q(s_t, a_t) \nabla_{\phi_\mathcal{G}} \log \mathcal{G}_p(a_t \mid s_t)\right]\end{aligned} \tag{4}$$

Since an initial token p_1 is required for the auto-regressive generator, we choose nine common initial words for p_1: The, It, To, There, What, This, All, If and We. In Eq.(4), $Q(s_t, a_t)$ is undefined, and we will discuss how to model $Q(s_t, a_t)$ to make the topic prompt task-relevant.

Adversarial reward. The teacher and the student model provide delayed estimation for the generated prompts once the synthesized utterance is generated. For state $s_t = P_{1:t}$ and its corresponding action $a_t = p_{t+1}$, the content that expanded based on the temporary topic prompt is as follows:

$$\hat{x} \sim \mathcal{G}_c(\cdot \mid P_{1:t+1}, \theta_G) \tag{5}$$

where $\hat{x}$ is a complete sentence on which the teacher and the student can estimate its probability for each category. Considering the probability as the likelihood that the synthesized sample belongs to a specific category, the teacher chooses the highest probability among all classes as the returned reward:

$$Q(s = P_{1:t}, a = p_{t+1}) = \max_{\mathcal{C}}\left(\mathcal{T}\left(\hat{x}\right)\right) \tag{6}$$

where $\mathcal{C}$ is the label space. However, unitary feedback from the teacher model makes the prompt generator susceptible to overfitting to a specific prompt, which sometimes corresponds to an out-of-distribution sample (see Figure 2). Thus, an adversarial reward is proposed, where the teacher reward is conjunct with the student reward:

$$\begin{aligned}&c' = \operatorname{argmax}_{c \in \mathcal{C}}\left(\mathcal{T}\left(\hat{x}\right)\right) \\ &Q(s = P_{1:t}, a = p_{t+1}) = \mathcal{T}_{c'}\left(\hat{x}\right) - \mathcal{S}_{c'}\left(\hat{x}\right)\end{aligned} \tag{7}$$

where c' represents the category with the largest response in the teacher network. Eq.(7) can be considered as an estimation of the discrepancy between the teacher and the student. The discrepancy is narrowed when optimizing the student, while the topic prompter is designed to enlarge this discrepancy. Once the student network has mastered an utterance guided by a certain prompt, the probability of the occurrence for this prompt should be reduced and a lower reward should be assigned. This adversarial reward allows for continuous adjustments of the prompts during the training process, significantly increasing the diversity of the pseudo-samples.

Algorithm 1: The training procedure of PromptDFD

```
Input: The teacher model $\mathcal{T}$ and its parameter $\theta_\mathcal{T}$, the
       content generator $\mathcal{G}_c$ and its parameter $\theta_\mathcal{G}$
Output: The student model $\mathcal{S}$ and its parameter $\theta_\mathcal{S}$,
        the topic prompter $\mathcal{G}_p$ and its parameter $\phi_\mathcal{G}$
Initial $\theta_\mathcal{S}$ with $\theta_\mathcal{T}$
for $i \leftarrow 1$ to $N$ do
    Fix $\theta_\mathcal{T}$, $\theta_\mathcal{S}$, $\theta_\mathcal{G}$
    Generate $P_{1:n} = (p_1, p_2, \ldots, p_n) \sim \mathcal{G}_p(\cdot; \phi_\mathcal{G})$
    for $m \leftarrow 1$ to $n-1$ do
        Generate $\hat{x}$ using $P_{1:m+1}$ by Eq.(5)
        Compute $Q(s_m, a_m)$ by Eq.(7)
    end
    Fix $\theta_\mathcal{T}$ and update $\theta_\mathcal{S}$, $\phi_\mathcal{G}$
    Update $\theta_\mathcal{S}$ by Eq.(1)
    Update $\phi_\mathcal{G}$ by Eq.(4) and Eq.(8)
end
```

Repeat penalty. To avoid the reduction of valid information contained in the prompts due to the repetition of tokens, diversity loss is introduced to maximize the difference between the probability distributions between different tokens:

$$L_{repeat} = -\sum_{i=1}^{n}\sum_{j=1}^{i-1} \mathrm{KL}\left(\Pr(p_i | p_{<i}, \phi_G), \Pr(p_j | p_{<j}, \phi_G)\right) \tag{8}$$

Algorithm 1 describes the training procedure of PromptDFD.

5 Experiments

5.1 Setup

Datasets. We conduct experiments on four text classification datasets to validate the efficacy of PromptDFD: AG News [Zhang *et al.*, 2015], DBPedia [Auer *et al.*, 2007], IMDb [Maas *et al.*, 2011] and SST-2 [Socher *et al.*, 2013].

Model structure. BERT-base is selected as the structure for the teacher model due to its widespread use. For consistency with previous methods, we use three kinds of structures for the student: BERT$_6$, BERT$_4$, and BERT$_{mini}$. As for $\mathcal{G}_c$, the pre-trained GPT2 [Radford *et al.*, 2019] is utilized to incorporate informative knowledge into synthesized samples, while a lightweight version of GPT2, DistilGPT2[1], is selected as $\mathcal{G}_p$ to generate prompts that are relatively short in length.

Comparison methods. Considering that the content generator $\mathcal{G}_c$ pre-trains on a large scale of unlabeled corpora, which can serve as an additional resource in the data-free setting, we include some baselines incorporating the out-of-domain data for fair comparison. Baselines are separated into three groups: data-driven knowledge distillation (Vanilla-KD), data-free knowledge distillation without any external resources (Random Text, Modified-ZSKT, Modified-ZSKD, AS-DFD [Ma *et al.*, 2020]) and data-free knowledge distillation with out-of-domain data (Unlabel-KD and [Rashid *et al.*, 2021]). The introduction of each baseline is given in Appendix .

[1] https://huggingface.co/distilgpt2

	AG News		DBPedia		IMDb	
	Acc.	Agree.	Acc.	Agree.	Acc.	Agree.
BERT-base	94.75	-	99.36	-	88.52	-
BERT$_6$						
Vanilla-KD	94.1	99.89	99.27	99.91	87.05	98.40
Random Text	85.4	90.66	93.9	94.50	77.1	87.10
Modified-ZSKT	88.4	93.84	FAIL	FAIL	78.1	88.23
Modified-ZSKD	88.6	94.06	97.1	97.73	78.2	88.34
AS-DFD	90.4	95.97	98.2	98.83	79.8	90.15
Unlabel-KD	92.49	97.61	99.21	99.91	86.15	97.32
PromptDFD-Manual	93.29	98.46	99.10	99.74	85.83	96.96
PromptDFD-RL	**93.74**	**98.93**	**99.30**	**99.94**	**86.94**	**98.22**
BERT$_4$						
Vanilla-KD	93.8	99.57	99.26	99.90	85.90	97.04
Random Text	78.5	83.33	77.3	77.80	67.6	76.37
Modified-ZSKT	81.1	96.09	FAIL	FAIL	70.4	79.53
Modified-ZSKD	83.8	88.96	83.0	83.53	70.7	79.87
AS-DFD	88.2	93.63	94.1	94.71	77.2	87.21
Unlabel-KD	91.54	96.61	98.97	99.61	83.86	94.74
PromptDFD-Manual	91.75	96.83	98.89	99.53	83.48	94.31
PromptDFD-RL	**92.61**	**97.74**	**99.17**	**99.81**	**85.41**	**96.49**

Table 3: Results over AG News, DBPedia and IMDB in terms of Accuracy(%) and Agreement(%). Since our teacher models differ from the baselines in terms of accuracy, we add a metric 'Agree.' to reflect the agreement between the teacher and the student models.

Implementation. For BERT$_n$, the student uses the first n layers of the teacher network as an initialization scheme. For BERT$_{mini}$, we follow [Rashid *et al.*, 2021] to take the pre-trained BERT-mini to initialize the student. The maximum token length is set to 128. A grid search on parameters is performed in our experiment, where the learning rate of the student is selected from {1e-5, 2e-5, 5e-5, 1e-4}, the learning rate of the $\mathcal{G}_p$is selected from {5e-6, 1e-5, 2e-5}, the batch size is selected from {128, 256}, the epoch for training is set to 10, the temperature of KD is selected from {1, 5, 10}, and α is selected from {0.5, 0.9}. Adam is used to optimize the student network and the topic prompter. The top 50 tokens with the highest probability are selected for decoding and the threshold for top-p is set to 0.95. More implementation details for PromptDFD and Unlabel-KD are given in Appendix.

5.2 Experimental results

We evaluate our proposed methods with two types of prompts: hand-crafted prompts (**PromptDFD-Manual**) and reinforced prompts (**PromptDFD-RL**). Table 3 and Table 4 compare the accuracy and the teacher-student agreement of different data-free distillation algorithms. PromptDFD-RL outperforms all other algorithms in four datasets. A significant improvement is observed compared with AS-DFD, largely narrowing the gap between data-driven distillation and data-free distillation. Under the setting that external resources are used, PromptDFD also shows superior performance over [Rashid *et al.*, 2021] and Unlabel-KD. Surprisingly, in SST-2 and DBPedia with BERT$_6$ as student, PromptDFD-RL achieves slightly better performance compared with Vanilla-KD. This indicates that the synthesized set provides sufficient knowledge for students compared with the original training set.

	BERT$_6$		BERT$_{mini}$	
	Acc.	Agree.	Acc.	Agree.
BERT-base	93.00	-	93.00	-
Vanilla-KD	91.63	98.53	88.30	94.95
Unlabel-KD$^\diamond$	-	-	84.9	91.29
Unlabel-KD	90.94	97.78	86.24	92.73
[Rashid *et al.*, 2021]	-	-	85.9	92.37
PromptDFD-Manual	90.48	97.29	86.35	92.85
PromptDFD-RL	**92.09**	**99.02**	**87.73**	**94.33**

Table 4: Performance on the dev set of SST-2. Unlabel-KD$^\diamond$ is reported in [Rashid *et al.*, 2021], and we re-implement Unlabel-KD with the whole set of wikitext-103.

	3	4	5	6	8	10
BERT$_6$	93.33	93.57	93.70	**93.74**	93.57	93.53
BERT$_4$	92.26	92.55	**92.61**	92.54	92.42	92.41

Table 5: Accuracy comparison of the PromptDFD on AG News dataset when varying the length of the prompt.

5.3 Quantitative Analysis

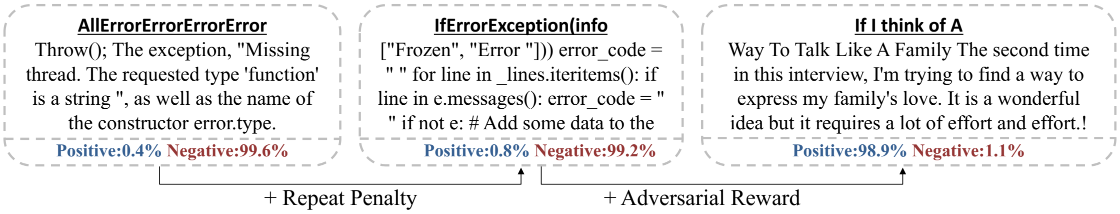
Impact of prompt length. Table 5 shows how the length of the prompt influences the distillation performance. The case where the prompt length is extended until it reaches the length of the pseudo sample can be characterized as a direct fine-tuning of the synthesized sentence. According to the table, the optimal length for prompts is approximately 5–6 words. If the prompt is too short, there will be insufficient prefix constraints in the prompt, while lengthening the prompt results in a decrease in performance due to the increasing complexity of training $\mathcal{G}_p$ with a huge action space.

Ablation study. Ablation results are reported in Table 6. Compared with PromptDFD-Manual (designed with templates in table 2), the reinforced prompts can steadily improve the performance, indicating the hand-crafted prompts are not the best choice for pseudo-sample construction. The repeat penalty reduces the redundancy between tokens, showing a 0.12%-0.74% decrease on the performance. Adversarial reward shows a little effect on AG News and IMDb but observes a significant impact on SST-2. The main reason for this is that the AG News and IMDb datasets are relatively simpler than SST-2, which can be demonstrated by the performance of PromptDFD-Manual.

An example of how each module affects the generation of prompts is shown in Figure 2. The original prompt is quite repetitive, which is mitigated by introducing a diversity loss. Nevertheless, we find that the generated prompts are far from the domain of the dataset. The teachers give a high likelihood to OOD samples, but they are not helpful in distilling, which is reflected in the performance of models that are distilled using these samples. While the adversarial reward does not prevent the model from generating such OOD samples, a near-zero reward is given after the student has learned these samples. The topic prompter is biased not to generate those

	AG News		IMDb		SST-2	
	BERT$_6$	BERT$_4$	BERT$_6$	BERT$_4$	BERT$_6$	BERT$_{mini}$
PromptDFD-Manual	93.29	91.75	85.82	83.48	89.11	78.55
PromptDFD-RL	93.74	92.61	86.94	85.41	92.09	87.73
w/o Adv Reward	93.57(0.17↓)	92.47(0.14↓)	86.75(0.19↓)	84.87(0.54↓)	88.42(3.67↓)	81.08(6.65↓)
w/o Repeat Penalty	93.01(0.73↓)	92.45(0.16↓)	86.69(0.25↓)	84.67(0.74↓)	91.97(0.12↓)	86.70(1.03↓)
w/o Adv Reward + Repeat Penalty	92.62(1.12↓)	92.36(0.25↓)	86.58(0.36↓)	84.90(0.51↓)	88.30(3.79↓)	79.01(8.72↓)

Table 6: Ablation results for PromptDFD. "w/o Adv Reward" denotes the adversarial feedback is replaced by Eq.(6). "w/o Repeat Penalty" indicates the objective Eq.(8) is not included in the training objective.



AllErrorErrorErrorError
Throw(); The exception, "Missing thread. The requested type 'function' is a string ", as well as the name of the constructor error.type.
Positive:0.4% Negative:99.6%

IfErrorException(info
["Frozen", "Error "]) error_code = " " for line in _lines.iteritems(): if line in e.messages(): error_code = " " if not e: # Add some data to the
Positive:0.8% Negative:99.2%

If I think of A
Way To Talk Like A Family The second time in this interview, I'm trying to find a way to express my family's love. It is a wonderful idea but it requires a lot of effort and effort.!
Positive:98.9% Negative:1.1%

Figure 2: Generated prompt with different modules in PromptDFD. This prompt is optimized under the SST-2 dataset.

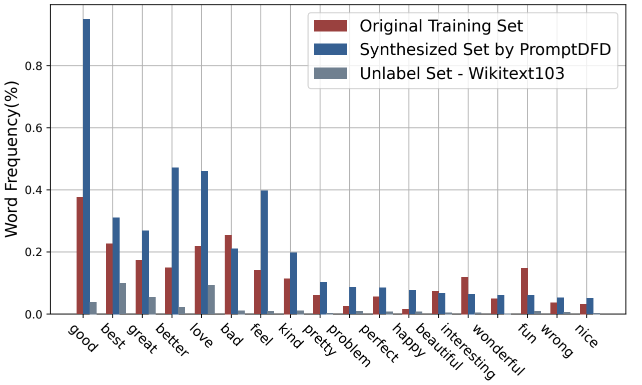


Figure 3: Word frequency of some positive/negative words in the training set of SST-2, synthesized samples generated by PromptDFD and wikitext103.

prompts, avoiding the prompt generators from getting stuck in out-of-distribution samples.

The quality of the synthesized samples are evaluated by two criteria: the presence of task-related terms and the reasonableness and meaningfulness of the utterance.

Topic relevance of synthesized samples. To visually demonstrate the relationship between the synthesized samples and the original training data, we analyze the frequency of task-related keywords in Figure 3. On the sentiment classification dataset SST-2, PromptDFD tends to generate more emotional words than the unlabeled corpus like wikitext103. These words frequently appear in the original training set, demonstrating that the synthesized set generated by PromptDFD mimics the original data to some extent.

Importance of synthesizing meaningful samples To destroy the semantics in the synthesized samples generated by PromptDFD-RL, we shuffle the words, resulting in a synthesized set that shares the same set of words with broken meanings. Taking these disordered sentences as the distillation set, a noticeable performance drop is observed in Table 7, showing that plausible synthetic utterances are indispensable in data-free knowledge distillation.

	IMDb		SST-2	
	BERT$_6$	BERT$_4$	BERT$_6$	BERT$_{mini}$
PromptDFD	86.94	85.41	92.09	87.16
w/o semantic	84.99	83.17	90.25	85.44

Table 7: Accuracy when the semantics of the synthesized utterances are broken down by shuffling the words.

6 Conclusion

In this work, we propose a novel data-free knowledge distillation framework PromptDFD that incorporates controllable language prior into the synthesized samples by reinforced prompts. The reinforced prompt is guided by the adversarial feedback to adjust the topic dynamically and increase the diversity. PromptDFD largely improves the quality of the synthesized samples and the distillation performance, and even reaches the performance of data-driven knowledge distillation with the original training data.

Acknowledgments

This work is supported by the Key Research and Development Program of Zhejiang Province, China (No. 2021C01013), the National Key Research and Development Project of China (No. 2018AAA0101900), CKCEST, and MOE Engineering Research Center of Digital Library. This work is also supported by NUS Faculty Research Committee Grant (WBS: A-0009440-00-00) and Advanced Research and Technology Innovation Centre (Project Reference: ECT-RP2).

References

[Auer *et al.*, 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In *Proc. of ISWC*, 2007.

[Chen *et al.*, 2019] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proc. of ICCV*, 2019.

[Chen *et al.*, 2020] Daoyuan Chen, Yaliang Li, Minghui Qiu, Zhen Wang, Bofang Li, Bolin Ding, Hongbo Deng, Jun Huang, Wei Lin, and Jingren Zhou. Adabert: Task-adaptive BERT compression with differentiable neural architecture search. In *Proc. of IJCAI*, 2020.

[Fang *et al.*, 2021] Gongfan Fang, Jie Song, Xinchao Wang, Chengchao Shen, Xingen Wang, and Mingli Song. Contrastive model invertion for data-free knolwedge distillation. In *Proc. of IJCAI*, 2021.

[Goodfellow *et al.*, 2015] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proc. of ICLR*, 2015.

[Hinton *et al.*, 2015] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, 2015.

[Holtzman *et al.*, 2020] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *Proc. of ICLR*, 2020.

[Hou *et al.*, 2020] Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. Dynabert: Dynamic BERT with adaptive width and depth. In *Proc. of NeuIPS*, 2020.

[Jiao *et al.*, 2020] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling BERT for natural language understanding. In *Proc. of ACL*, 2020.

[Li and Liang, 2021] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proc. of ACL*, 2021.

[Lopes *et al.*, 2017] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017.

[Ma *et al.*, 2020] Xinyin Ma, Yongliang Shen, Gongfan Fang, Chen Chen, Chenghao Jia, and Weiming Lu. Adversarial self-supervised data-free distillation for text classification. In *Proc. of EMNLP*, 2020.

[Maas *et al.*, 2011] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proc. of ACL*, 2011.

[Mahendran and Vedaldi, 2015] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015.

[Micaelli and Storkey, 2019] Paul Micaelli and Amos J Storkey. Zero-shot knowledge transfer via adversarial belief matching. In *Proc. of NeuIPS*, 2019.

[Nayak *et al.*, 2019] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *Proc. of ICML*, 2019.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

[Rashid *et al.*, 2021] Ahmad Rashid, Vasileios Lioutas, Abbas Ghaddar, and Mehdi Rezagholizadeh. Towards zero-shot knowledge distillation for natural language processing. In *Proc. of EMNLP*, 2021.

[Sanh *et al.*, 2019] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, 2019.

[Shin *et al.*, 2020] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proc. of EMNLP*, 2020.

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*, 2013.

[Sun *et al.*, 2019] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for BERT model compression. In *Proc. of EMNLP*, 2019.

[Sutton *et al.*, 1999] Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proc. of NeuIPS*, 1999.

[Wu *et al.*, 2021] Yimeng Wu, Mehdi Rezagholizadeh, Abbas Ghaddar, Md Akmal Haidar, and Ali Ghodsi. Universal-KD: Attention-based output-grounded intermediate layer knowledge distillation. In *Proc. of EMNLP*, 2021.

[Yin *et al.*, 2020] Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proc. of CVPR*, 2020.

[Yu *et al.*, 2017] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proc. of AAAI*, 2017.

[Zhang *et al.*, 2015] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proc. of NeuIPS*, 2015.