Generating Synthetic Dataset for Few-shot Prompt Tuning

Xu Guo | Zilin Du | Boyang Li | Chunyan Miao | Nanyang Technological University, Singapore

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Background

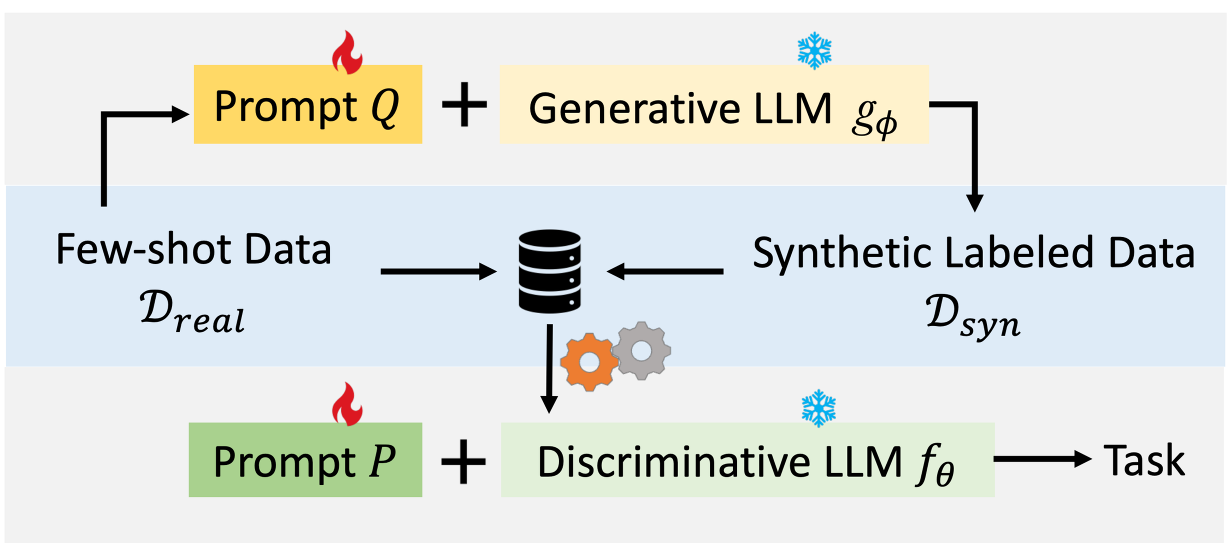
Prompts are essential for adapting LLMs to downstream tasks, but manually crafting them for each task is challenging. Prompt tuning, which learns "soft" prompts from a labeled dataset, can outperform manual prompts and closes the gap with model tunning. However, it requires a sufficiently large labeled dataset to be effective. In few-shot learning scenarios, it significantly underperforms model tuning.

Motivation

To boost Prompt Tuning in few-shot tasks, previous research mainly leverages transfer learning to pretrain soft prompts on large real-world corpora and then fine-tune them on the task. However, finding such datasets is challenging due to domain discrepancies and limited availability, especially in low-resource areas. Recently, there has been a growing interest in generating training data with LLMs. This paper builds on them to boost prompt tuning in few-shot learning settings, bypassing the need for large-scale labeled data.

Method

A schematic overview of the framework:



Step 1: Algorithm 1

Step 2: Training Data generation

Step 3: Algorithm 2

Step 1 - Adapting g_ϕ to the task domain in a parameter-efficient manner (Q). Meanwhile, encouraging $g_{\phi,Q}$ to generate label-relevant text data through a trainable weight net $\Phi_W : \mathbb{R}^d \mapsto \mathbb{R}$ under bi-level optimization.

Algorithm 1: Generator Tuning.

Data: Few-shot real dataset $\mathcal{D}_{real}$.
Initialize: Prompts Q, Pre-trained LLM g_ϕ.
initialize $t = 0$;
while $t < T + 1$ **do**
  $t+ = 1$;
  $\mathcal{B} \leftarrow$ Sample a batch from $\mathcal{D}_{real}$;
  $Q^t(W^t) \leftarrow$ Take a gradient descent step on $\mathcal{B}$ with $\mathcal{L}_{\text{DawGen}}(Q^t; W^t)$;
  $W^{t+1} \leftarrow$ Take a gradient descent step on $\mathcal{B}$ with $\mathcal{L}_{\text{disc}}(Q^t(W^t))$;
  $Q^{t+1} \leftarrow$ Take a gradient descent step on $\mathcal{B}$ with $\mathcal{L}_{\text{DawGen}}(Q^t; W^{t+1})$;
end
Output: Generator $g_{\phi,Q}$

Prioritizing label-discriminative tokens:

$$\mathcal{L}_{\text{wGen}}(Q_l) = -\mathbb{E}_{X \in \mathcal{D}_{real}, Y=Y_l} \mathbb{E}_{x_j \in X} W_j \cdot \log \Pr_\phi(x_j | x_{<j}; Q_l).$$

Learning to generate discriminative tokens:

$$\mathcal{L}_{\text{disc}}(W) = -\mathbb{E}_{x_j \in X} \frac{\Pr_\phi(x_j | x_{<j}; Q_l(W))}{\sum_{l'} \Pr_\phi(x_j |, x_{<j}; Q_{l'}(W))}$$

Semantic regularization: $Z_{i,l} = g_{\phi,Q}(X_i)$.

$$\mathcal{L}_{\text{dist}}(Q) = \mathbb{E}_{(X,y) \in \mathcal{D}_{real}} max(0, 1 - D(W \cdot Z_{i,l}, W \cdot Z_{j,l}) + D(W \cdot Z_{i,l}, W \cdot Z_{j,l'})).$$

$$\mathcal{L}_{\text{DawGen}}(Q) = \mathbb{E}_{l \in [1,L]} \mathcal{L}_{\text{wGen}}(Q_l) + \mathcal{L}_{\text{dist}}(Q).$$

Step 2 – Label-conditioned training data generation: $\mathcal{Y} = \{Y_l\}_{l=1}^L$

$$X = \max \prod_{j=1}^{K} \Pr_\phi(x_j | x_{<j}, Y_l; Q_l, W)$$

Step 3 – Train soft prompt (P) on the combined few-shot and synthetic data, and resolve conflicts with gradient surgery.

Standard cross entropy loss:

$$\mathcal{L}_{\text{ce}}(P) = -\mathbb{E}_{(X,y) \in \mathcal{D}} \log \Pr_\theta([\text{MASK}] = y | [P; E])$$

Gradient projection:

$$\text{Proj}_{\delta_{real}}(\delta_{syn}) = \frac{\delta_{syn} \cdot \delta_{real}}{\| \delta_{real} \|} \cdot \frac{\delta_{real}}{\| \delta_{real} \|}$$

Algorithm 2: Prompt Tuning.

Data: Few-shot $\mathcal{D}_{real}$ and synthetic $\mathcal{D}_{syn}$.
Initialize: Prompts P, Pre-trained LLM f_θ.
initialize $t = 0$;
while $t < T + 1$ **do**
  $t+ = 1$;
  $\mathcal{B}_{real}, \mathcal{B}_{syn} \leftarrow$ Sample a batch from $\mathcal{D}_{real}, \mathcal{D}_{syn}$;
  Compute gradients $\delta_{real} = \frac{\partial \mathcal{L}_{\text{ce}}(P)}{\partial P}$ on $\mathcal{B}_{real}$;
  Compute gradients $\delta_{syn} = \frac{\partial \mathcal{L}_{\text{ce}}(P)}{\partial P}$ on $\mathcal{B}_{syn}$;
  if $\delta_{syn} \cdot \delta_{real} < 0$ **then**
    $\delta'_{\text{syn}} = \delta_{syn} - \text{Proj}_{\delta_{real}}(\delta_{syn})$;
  end
  $\delta = \delta_{real} + \epsilon \cdot \delta'_{syn}$;
  $P \leftarrow P - \eta \cdot \delta$;
end
Output: Soft prompt P

gradient surgery

Comparisons with model tuning and transfer learning

Augmenting few-shot prompt tuning with synthetic data improves performance by ~18% on average and outperforms full-model tuning by 3.8% on T5-large. The results are competitive with transfer learning on QQP, MRPC, and SICK.

Method	#Trainable Params	QQP	MRPC	MNLI	SNLI	QNLI	RTE	SICK	AVG
				T5-large					
Prompting	0	42.63	33.80	33.20	33.31	49.46	52.35	14.51	37.04
In-Context		59.55	33.52	34.49	33.80	49.72	48.52	40.90	42.93
FT	770M	**72.50**	61.72	42.82	48.90	50.11	55.81	**77.90**	58.54
Prompt-based FT		60.15	59.66	42.94	**54.16**	51.75	57.18	69.98	56.55
PFT + soft prompt		60.22	56.18	43.86	48.45	57.38	55.60	76.23	56.85
PPT	410K	46.11	52.37	34.05	35.28	52.86	48.59	45.64	44.99
Prompt Tuning	102K	47.28	58.94	33.29	33.21	52.68	51.70	27.80	43.49
Ours	102K	66.77	**69.67**	**53.20**	46.81	**69.84**	**57.40**	72.73	**62.35**
SPOT†	102K	64.5	68.7	74.3	78.8	-	-	72.9	-
OPTIMA†		69.1	71.2	78.4	82.1	-	-	73.3	-
				Flan-T5-large					
Prompt Tuning	102K	70.40	72.82	59.89	63.26	83.73	26.78	60.61	62.49
Ours		82.14	**78.40**	71.84	82.43	88.80	56.82	79.88	77.19

Prompting: Zero-shot. In-Context: Few-shot. FT: Full-model Tuning without prompts. Prompt-based FT (PFT): wrap samples with prompts (e.g., LM-BFF). PFT+soft prompt: adds a soft prompt (e.g., PET). PPT: pre-trains soft prompt on large corpora. SPOT / OPTIMA transfer learning with source-domain datasets.

Investigating data combination strategies

How synthetic and real data interact during learning is crucial. A naive combination can lead to poor results, and label smoothing offers limited benefits. Starting with real data doesn't always enhance learning, but pairing it with synthetic data yields better results.

Method	Generator	QQP	MRPC	MNLI	SNLI	QNLI	RTE	SICK	AVG
				T5-Large					
Real+Syn	FewGen	52.64	**70.53**	38.15	33.96	57.08	52.64	48.01	50.43
Real+Syn+LS		53.09	67.94	38.58	34.11	56.99	56.03	58.05	52.11
Real → Syn		59.51	70.04	47.46	41.99	**65.52**	**57.91**	65.81	58.32
Syn → Real		63.78	68.92	36.97	35.17	63.24	53.86	52.90	53.55
(Real, Syn)		**66.44**	68.12	**48.03**	**44.81**	64.15	56.54	**68.46**	**59.51**
Real+Syn	DawGen	**62.86**	**70.38**	43.97	35.62	60.11	53.29	51.31	53.93
Real → Syn		62.60	69.39	47.94	46.14	66.33	**58.12**	61.48	58.85
Syn → Real		62.69	69.28	42.45	38.01	60.35	55.31	56.20	54.89
(Real, Syn)		61.77	69.99	**48.76**	**45.10**	**66.37**	57.20	**70.80**	**59.99**
				Flan-T5-large					
Real+Syn	DawGen	83.60	76.81	71.85	72.48	84.11	53.72	69.17	73.10
Real → Syn		80.50	75.64	66.42	69.41	86.52	**54.22**	73.33	72.29
Syn → Real		**83.26**	**78.55**	**72.15**	77.29	**87.51**	50.76	72.17	74.53
(Real, Syn)		81.83	76.96	70.18	**79.69**	87.38	51.63	**76.97**	**74.94**

Real+Syn: combines synthetic and few-shot data directly. Real+Syn+LS: applies label smoothing regularization. Real→Syn: trains soft prompt on $\mathcal{D}_{real}$ first and then on $\mathcal{D}_{syn}$ in every training epoch. Syn→Real: the opposite approach. (Real, Syn): pairing them in every batch.

Ablation study

Distribution-aligned regularization improves synthetic data quality. Using few-shot data to enhance learning from synthetic data is helpful (Real ✓). Gradient surgery (GS) further reconciles learning from both real and synthetic data.

Generator	Real	GS	QQP	MRPC	MNLI	SNLI	QNLI	RTE	SICK	AVG
					T5-Large					
FewGen			56.70	69.25	42.18	34.63	56.91	53.87	34.11	49.66
	✓		66.44	68.12	48.03	44.81	64.15	56.54	68.46	59.51
	✓	✓	67.85	70.05	49.52	46.39	66.96	55.96	72.08	61.25
DawGen			58.62	69.11	44.30	36.63	61.97	55.16	50.39	53.74
	✓		61.77	69.99	48.76	45.10	66.37	57.20	70.80	59.99
	✓	✓	66.77	69.67	53.20	46.81	69.84	57.40	72.73	62.35

Conclusion

This paper introduces a Distribution-Aligned Weighted GENerator tuning method to generate label-relevant training data. We further apply gradient surgery to better utilize the synthetic data. Experiments on seven datasets and two LLMs show the method largely boosts prompt tuning in few-shot settings, bypassing full-model tuning and offering an alternative to real data in some tasks.

First Conference on Language Modeling (COLM 2024)

COLM