TransPrompt: Towards an Automatic Transferable Prompting Framework for Few-shot Text Classification

Chengyu Wang*[1], Jianing Wang*[2], Minghui Qiu [1], Jun Huang [1], Ming Gao [2]

[1] Alibaba Group [2] East China Normal University

Introduction (1)

✓Few-shot learning for pre-trained language models (PLMs)

- Using prompt-based approaches to fine-tune PLMs
- A few approaches (e.g., P-tuning) employ continuous prompts to ease the process of human engineering
- The performance may be limited by the small training sets

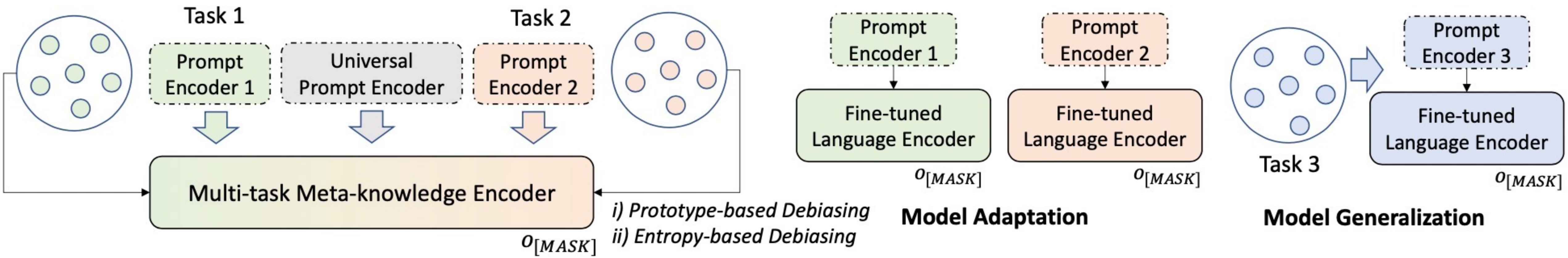
✓Transferable Few-shot Learning

- Prompt-based approaches can capture the knowledge across similar NLP tasks

Introduction (2)

✓Our idea: the TransPrompt framework

- Multi-task Meta-knowledge Acquisition: learning the transferable representations of prompt encoders and PLMs jointly across similar NLP tasks
- Task-aware Model Specification:
  - Model Adaptation: adaptive to specific existing tasks
  - Model Generalization: generalized to new tasks



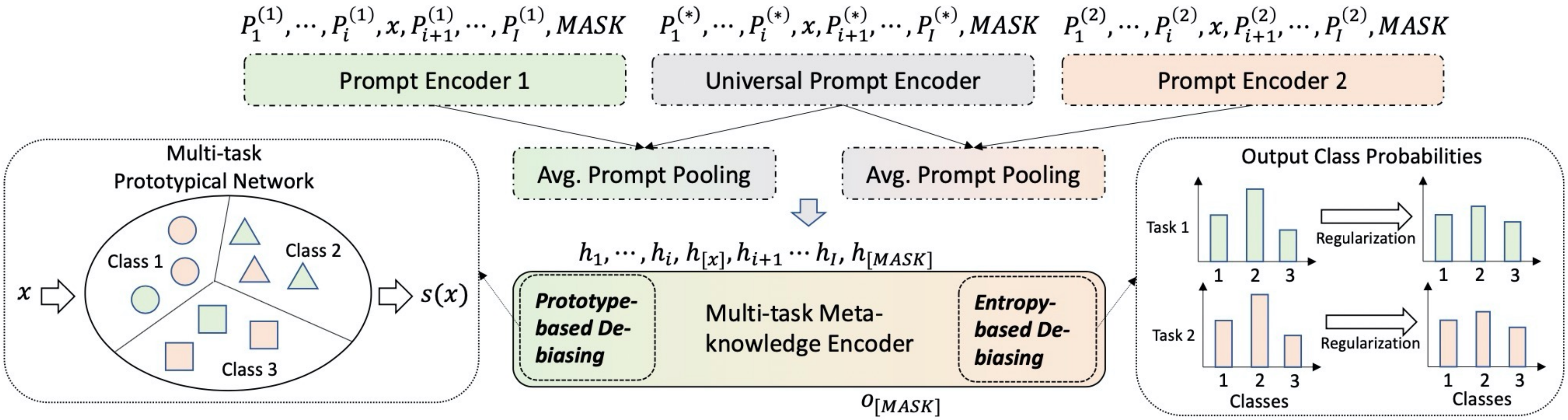
a) Multi-task Meta-knowledge Acquisition

b) Task-aware Model Specification

Multi-task Meta-knowledge Acquisition

✓New techniques for capturing transferable knowledge

- Universal Prompt Encoder
- Two Debiasing Techniques



Multi-task Meta-knowledge Acquisition

✓Protype-based Debiasing: learning instance-level transferable knowledge

- Compute prototype scores to select transferable instances across tasks

$$s(x) = \zeta \cdot \frac{\text{sim}(\mathcal{E}(x), \mathbf{c}_m(y))}{\sum_{\tilde{y}\in\mathcal{Y}} \text{sim}(\mathcal{E}(x), \mathbf{c}_m(\tilde{y}))} + \frac{1-\zeta}{M-1} \sum_{\tilde{m}=1 (m\neq\tilde{m})}^{M} \frac{\text{sim}(\mathcal{E}(x), \mathbf{c}_{\tilde{m}}(y))}{\sum_{\tilde{y}\in\mathcal{Y}} \text{sim}(\mathcal{E}(x), \mathbf{c}_{\tilde{m}}(\tilde{y}))}$$

$$\mathcal{L}(\Theta) = \sum_{m=1}^{M} \sum_{(x,y)\in\mathcal{D}_m} s(x) l(x, y; \Theta) + \lambda_1 \|\Theta\|$$

✓Entropy-based Debiasing: learning task-level transferable knowledge

- Add an entropy-based loss to make the PLM more task-agnostic

$$\mathcal{H}(\mathcal{D}_m) = -\frac{1}{|\mathcal{D}_m|} \sum_{(x,y)\in\mathcal{D}_m} \sum_{\hat{y}\in\mathcal{Y}} \hat{y}(x) \log \hat{y}(x),$$

Task-aware Model Specification

✓Model Adaptation: fine-tuning the corresponding prompt encoder and the PLM

$$\mathcal{L}^{(m)}(\Theta) = \sum_{(x,y)\in\mathcal{D}_m} l(x, y; \Theta) + \lambda_1 \|\Theta\|$$

✓Model Generalization: using the parameters of the universal prompt encoder to initialize its own prompt encoder

$$\tilde{\mathcal{L}}(\Theta) = \sum_{(x,y)\in\tilde{\mathcal{D}}} l(x, y; \Theta) + \lambda_1 \|\Theta\|$$

Experiments (1)

✓Few-shot experiments of TransPrompt

- Model: Roberta-large
- Training data: 16 shots

Method	Task: Sentiment Analysis			Task: NLI		Task: Paraphrase		Avg.
	SST-2	**MR**	**CR**	**MNLI**	**SNLI**	**MRPC**	**QQP**	
Single-task Baselines								
Fine-tuning (Devlin et al., 2019)	81.42	76.15	84.50	54.17	44.45	73.28	59.64	67.66
LM-BFF (man) (Gao et al., 2020)	90.75	86.60	90.50	63.62	70.77	74.05	60.27	76.65
LM-BFF (auto) (Gao et al., 2020)	91.62	87.25	91.80	64.25	71.21	74.23	60.59	77.28
P-tuning (Liu et al., 2021)	91.85	86.60	91.75	62.41	70.28	66.42	60.57	75.70
Cross-task Baselines								
Fine-tuning (mtl) (Sun et al., 2019)	83.37	79.30	84.75	41.32	48.14	53.12	59.31	64.19
Meta Fine-tuing (Wang et al., 2020a)	86.32	83.85	88.42	48.52	58.20	71.56	67.12	72.00
LM-BFF (mtl) (Gao et al., 2020)*	91.97	87.45	90.70	69.09	75.90	50.00	67.40	76.07
P-tuning (mtl) (Liu et al., 2021)*	93.12	87.75	91.35	68.83	74.24	70.83	69.99	79.44
***TransPrompt* (Proposed Approach)**	**93.58**	**88.80**	**92.00**	**71.90**	**76.99**	**75.98**	**75.80**	**82.15**

Experiments (2)

✓Full-data experiments of TransPrompt

- Model: Roberta-base
- Training data: full training sets

Method	Task: Sentiment Analysis			Task: NLI		Task: Paraphrase		Avg.
	SST-2	MR	CR	MNLI	SNLI	MRPC	QQP	
Single-task Baselines								
Fine-tuning (Devlin et al., 2019)	93.00	90.15	90.90	82.87	87.87	72.28	89.53	86.65
LM-BFF (man) (Gao et al., 2020)	93.65	88.50	90.98	87.23	91.10	88.75	85.12	89.33
LM-BFF (auto) (Gao et al., 2020)	93.81	88.75	91.25	87.01	91.51	88.97	83.12	89.20
P-tuning (Liu et al., 2021)	93.69	90.10	90.25	87.17	91.67	88.97	90.87	90.38
Cross-task Baselines								
Fine-tuning (mtl) (Sun et al., 2019)	94.72	90.65	91.05	87.10	91.80	69.85	90.20	87.91
Meta Fine-tuing (Wang et al., 2020a)	95.70	91.25	91.42	83.67	89.48	78.92	89.72	88.59
LM-BFF (mtl) (Gao et al., 2020)*	95.41	90.45	91.50	86.76	88.25	69.36	90.32	87.43
P-tuning (mtl) (Liu et al., 2021)*	95.30	90.40	90.08	86.97	91.48	68.87	90.59	87.67
***TransPrompt* (Proposed Approach)**	**96.05**	**91.78**	**91.59**	**88.70**	**91.88**	**86.87**	**91.27**	**91.16**

Conclusion

- ✓We present the TransPrompt framework for few-shot learning across similar NLP tasks.
- ✓Experiments confirm the effectiveness of TransPrompt over various NLP tasks.
- ✓Future work includes:
  - ✓Using TransPrompt in other application scenarios and other NLP tasks
  - ✓Exploring how TransPrompt can be applied to other PLMs apart from BERT-style models

THANKS

--------- Q&A Section --------