Effective Structured Prompting by Meta-Learning and Representative Verbalizer

Weisen Jiang[◇†] Yu Zhang[◇] James T. Kwok[†]

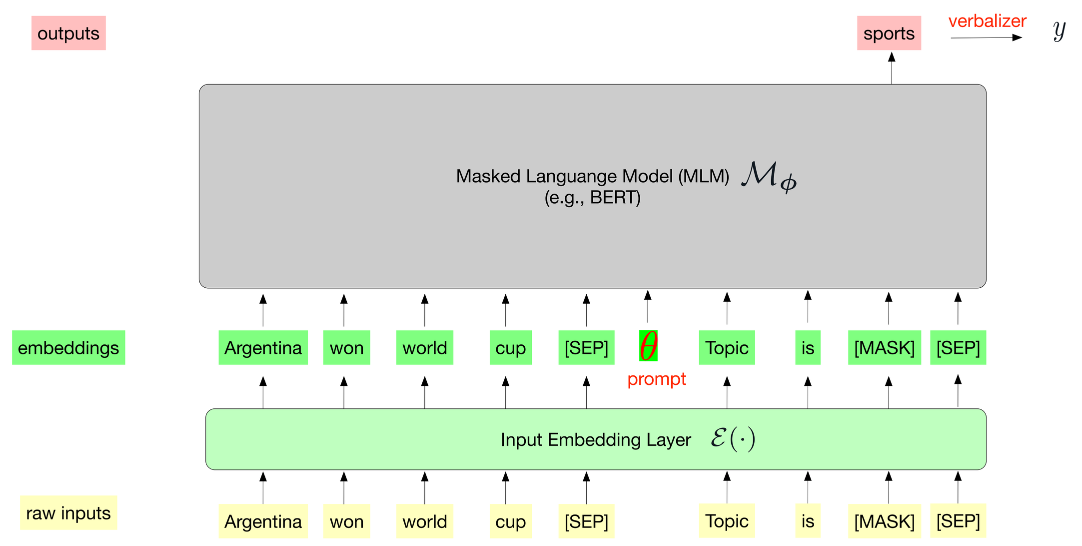
[◇]Southern University of Science and Technology [†]Hong Kong University of Science and Technology

PAPER CODE

Background

- Masked language model (MLM) can fill the masked tokens in a sentence, thus, can be used for classification tasks:
  1. input sentence is wrapped by a **template**: "sentence, Topic is [MASK]";
  2. MLM **predicts a token** at the [MASK] position;
  3. a **verbalizer** map the predicted token to label.
- Prompt tuning introduces **a learnable prompt** θ in the template:

$$\tilde{\mathbf{x}} \equiv \mathbb{T}(\mathbf{x}; \boldsymbol{\theta}) = (\mathcal{E}(\mathbf{x}), \boldsymbol{\theta}, \mathcal{E}(\texttt{Topic}), \mathcal{E}(\texttt{is}), \mathcal{E}(\texttt{[MASK]})).$$



- The initialization of θ plays an important role in prompt tuning.
- Recently, MetaPrompting (COLING 2022) proposes to meta-learn a shared prompt initialization for all task-specific prompts with a hand-crafting verbalizer.
- **Challenges:**
  1. a **single** meta-initialized prompt is insufficient for adaptation to complex tasks;
  2. **not parameter-efficient** as the whole MLM needs tuning;
  3. hand-crafting verbalizer is **labor-intensive**.
- *Notations*: task τ with support set $\mathcal{S}_\tau$, query set $\mathcal{Q}_\tau$, and label set $\mathcal{Y}_\tau$.

Representative Verbalizer (RepVerb)

We **propose a novel verbalizer RepVerb**, which is simple and effective:

- For input $\mathbf{x}$, compute its feature embeding $\mathbf{h}_{\text{[MASK]}}(\tilde{\mathbf{x}})$.
- For class y, compute label embedding $\mathbf{v}_y = \frac{1}{|\mathcal{S}_{\tau,y}|}\sum_{(\mathbf{x},y)\in\mathcal{S}_{\tau,y}} \mathbf{h}_{\texttt{[MASK]}}(\tilde{\mathbf{x}})$.
- Prediction: compute cosine similarity between $\mathbf{h}_{\text{[MASK]}}(\tilde{\mathbf{x}})$ and $\{\mathbf{v}_y : y \in \mathcal{Y}_\tau\}$.

procedure ComputeLabelEmbedding($\mathcal{S}_\tau$):
  compute $\mathbf{h}_{\texttt{[MASK]}}(\tilde{\mathbf{x}})$ for $(\mathbf{x}, \cdot) \in \mathcal{S}_\tau$;
  compute $\mathbf{v}_y$ for each $y \in \mathcal{Y}_\tau$;
end procedure

procedure Predict($\mathbf{x}; \mathbf{v}_y : y \in \mathcal{Y}_\tau$)
  compute $\mathbf{h}_{\texttt{[MASK]}}(\tilde{\mathbf{x}})$ for $\mathbf{x}$;
  $\tilde{\mathbb{P}}(y|\mathbf{x};\phi,\boldsymbol{\theta}) = \frac{\exp(\rho\cos(\mathbf{v}_y, \mathbf{h}_{\texttt{[MASK]}}(\tilde{\mathbf{x}})))}{\sum_{y'\in\mathcal{Y}_\tau}\exp(\rho\cos(\mathbf{v}_{y'}, \mathbf{h}_{\texttt{[MASK]}}(\tilde{\mathbf{x}})))}$
end procedure

MetaPrompter

- Use **a prompt pool to extract more task knowledge** for constructing instance-dependent prompt:
  1. **The prompt pool** has K learnable prompts $\{(\mathbf{k}_i, \boldsymbol{\theta}_i) : i = 1, \dots, K\}$, with key $\mathbf{k}_i$ and value $\boldsymbol{\theta}_i$;
  2. **Instance-dependent prompt** is constructed by a weighted combination of all values ($\boldsymbol{\theta}_i$'s): $\boldsymbol{\theta}_{\mathbf{x}}(\mathbf{K}, \boldsymbol{\Theta}) = \sum_{i=1}^K a_i \boldsymbol{\theta}_i$, where attention weight $\mathbf{a}$ is computed between input $\mathbf{x}$ and the K prompts.
- Prediction (**hand-crafting verbalizer** + **RepVerb**):

$$\mathbb{P}(y|\mathbf{x};\boldsymbol{\theta}_{\mathbf{x}}) = (1-\lambda) \times \hat{\mathbb{P}}(y|\mathbf{x};\boldsymbol{\theta}_{\mathbf{x}}) + \lambda \times \tilde{\mathbb{P}}(y|\mathbf{x};\boldsymbol{\theta}_{\mathbf{x}})$$

where $\hat{\mathbb{P}}(y|\mathbf{x};\boldsymbol{\theta}) = \frac{1}{|\mathcal{V}_y|}\sum_{\mathtt{w}\in\mathcal{V}_y}\mathbb{P}_{\mathcal{M}}(\texttt{[MASK]} = \mathtt{w}|\mathbb{T}(\mathbf{x};\boldsymbol{\theta}))$ ($\mathcal{V}_y$ is a set of label tokens).

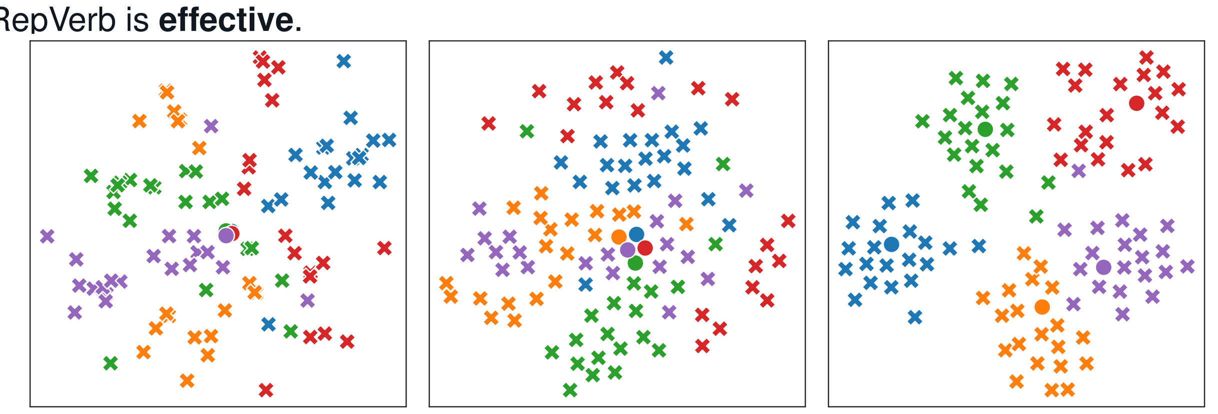
- We **propose a novel algorithm MetaPrompter** to learn the prompt pool by meta-learning (e.g., MAML):
  - **base learner**: (1) build instance-dependent prompts from $(\mathbf{K}, \boldsymbol{\Theta})$ (2) compute loss on support set $\mathcal{L}(\mathcal{S}_\tau; \mathbf{K}, \boldsymbol{\Theta})$ (3) update task-specific prompt pool $(\mathbf{K}^{(\tau)}, \boldsymbol{\Theta}^{(\tau)}) = (\mathbf{K}, \boldsymbol{\Theta}) - \alpha\nabla_{(\mathbf{K},\boldsymbol{\Theta})}\mathcal{L}(\mathcal{S}_\tau; \mathbf{K}, \boldsymbol{\Theta})$.
  - **meta-learner**: (1) compute loss on query set $\mathcal{L}(\mathcal{Q}_\tau; \mathbf{K}^{(\tau)}, \boldsymbol{\Theta}^{(\tau)})$ (2) update meta prompt pool $(\mathbf{K}, \boldsymbol{\Theta}) \leftarrow (\mathbf{K}, \boldsymbol{\Theta}) - \eta\nabla_{(\mathbf{K},\boldsymbol{\Theta})}\mathcal{L}(\mathcal{Q}_\tau; \mathbf{K}^{(\tau)}, \boldsymbol{\Theta}^{(\tau)})$.
- Compared with MetaPrompting, MetaPrompter is
  1. **Parameter-efficient**: only the prompt pool is tuned;
  2. **More flexible**: Instance-dependent prompt allows better adaptation to complex tasks.

Evaluation on RepVerb

Table 1: Meta-testing accuracy of 5-way few-shot classification.

		20News	*Amazon*	*HuffPost*	*Reuters*	*HWU64*	*Liu54*
5-shot	WARP	61.43	59.53	46.31	68.67	68.60	73.11
	ProtoVerb	71.33	71.74	57.93	80.93	73.43	76.19
	RepVerb	**78.81**	**77.56**	**61.90**	**88.33**	**78.37**	**82.14**
1-shot	WARP	49.87	48.94	38.21	52.88	53.20	58.68
	ProtoVerb	54.13	55.07	41.40	57.27	55.17	60.16
	RepVerb	**59.86**	**59.18**	**44.65**	**63.63**	**59.83**	**66.17**

- Baselines: soft verbalizers WARP (ACL 2021) and ProtoVerb (ACL 2022) learned by supervised and contrastive learning, respectively.
- RepVerb **outperforms** WARP and ProtoVerb on both the 1-shot and 5-shot settings.
- Hence, RepVerb is **effective**.



(a): WARP. (b): ProtoVerb (c): RepVerb.

Figure 1: t-SNE visualization of feature embeddings (crosses) and label embeddings (circles) for a 5-way 5-shot task from *Reuters*.

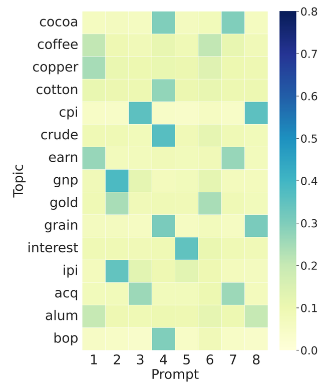
- RepVerb has **more discriminative and compact** embeddings than WARP and ProtoVerb.
- By design, RepVerb's label embedding is **consistent** with samples' embeddings, but those of WARP and ProtoVerb are not.

Evaluation on MetaPrompter

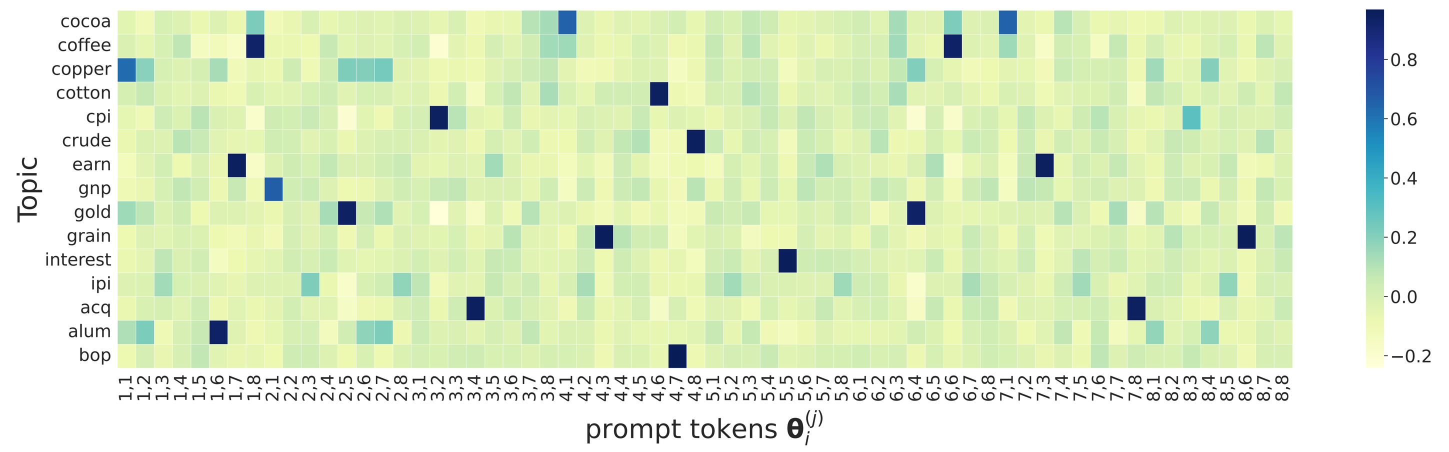
Table 2: 5-way 5-shot classification meta-testing accuracy.

	#param $(\times 10^6)$	*20News*	*Amazon*	*HuffPost*	*Reuters*	*HWU64*	*Liu54*
HATT	0.07	55.00	66.00	56.30	56.20	–	–
DS	1.73	68.30	81.10	63.50	96.00	–	–
MLADA	0.73	77.80	86.00	64.90	96.70	–	–
ConstrastNet	109.52	71.74	85.17	65.32	95.33	92.57	93.72
MetaPrompting	109.52	85.67	84.19	72.85	95.89	93.86	94.01
MetaPrompting+WARP	109.52	85.81	85.54	71.71	97.28	93.99	94.33
MetaPrompting+ProtoVerb	109.52	86.18	84.91	73.11	97.24	93.81	94.38
MetaPrompting+RepVerb	109.52	86.89	85.98	74.62	97.32	94.23	94.45
MetaPrompter	0.06	**88.57**	**86.36**	**74.89**	**97.63**	**95.30**	**95.47**

- Baselines: (i) state-of-the-art prompt-based methods (MetaPromping and its variants); (ii) non-prompt-based methods (HATT (AAAI 2019), DS (ICLR 2020), MLADA (ACL 2021), ConstrastNet (AAAI 2022)).
- MetaPrompter is **better than** both prompt-based and non-prompt-based baselines.
- MetaPrompter **outperforms** MetaPrompting+RepVerb, showing effectiveness of the prompt pool.
- MetaPrompter is **much more parameter-efficient** than MetaPrompting ($1800\times$ fewer).
- RepVerb is **beneficial** to MetaPrompting.



(a): Distribution of attention weights on 5-way 5-shot classification of *Reuters* (15 topics).



(b): Cosine similarities between learned prompt tokens and topic embeddings on 5-way 5-shot classification of *Reuters* (recall that K and L_p are set to 8). In the x-axis, (i, j) stands for the jth row of $\boldsymbol{\theta}_i$ (i.e., $\boldsymbol{\theta}_i^{(j)}$).

- Samples from topic *cocoa* prefer the 4th and 7th prompts (left), as *cocoa*'s embedding is similar to $\boldsymbol{\theta}_4^{(1)}$ and $\boldsymbol{\theta}_7^{(1)}$ (right).

Summary

- Problem: improve the effectiveness and parameter-efficiency of prompt tuning.
- Propose a novel algorithm MetaPrompter, consisting of:
  1. **a novel verbalizer RepVerb**: simple and effective.
  2. structured prompting: **a meta-learned prompt pool** to construct instance-dependent prompts.
- MetaPrompter is **parameter-efficient** as only the prompt pool is tuned.
- Experimental results demonstrate:
  1. RepVerb achieves higher accuracy than other soft verbalizers (WARP and ProtoVerb).
  2. MetaPrompter performs better than MetaPrompting.
  3. MetaPrompter is much more parameter-efficient than MetaPrompting ($1800\times$).