Learning to Generate Task-Specific Adapters from Task Description

Qinyuan Ye, Xiang Ren University of Southern California {qinyuany, xiangren}@usc.edu **https://github.com/INK-USC/hypter**

1. Problem Setting

We study the "learning from task descriptions" problem [1]

Input: A task description
e.g., "What accessibility services does this national park offer?"
Output: A model that is ready to solve that task
e.g., "Mammoth Cave National Park is pleased to offer sign interpreter services ..." → "sign interpreter services"

2. Prior Work

Solution 1. Using task descriptions as "prompt"
i.e., Train a text-to-text transformer with (description + source, target) pairs. At test time, directly use the novel task description.
e.g., "What accessibility services does this national park offer? Mammoth Cave National Park is pleased to offer sign interpreter services ..." → "sign interpreter services"
✔ **Straight-forward**
✘ **Reducing the "learning from task descriptions" problem back to the "learning from examples" problem**

Solution 2. Using hypernetwork approaches [2] that generate parameters from task descriptions
i.e., Encode the task description into hidden representations, then decode all parameters for the final model.
✔ **Indeed "learning from task descriptions"**
✘ **Text-to-text transformers have millions of parameters, they are too large to generate.**

3. Hypter: Using a Hypernetwork to Generate Task-Specific Adapters

Stage 1. Fine-tune a pre-trained text-to-text transformer with "prompted examples"
This step is the same as the solution 1 in prior work. We fine-tune a pre-trained BART model with all (description + source, target) examples in train tasks. We refer to the resulting model as the "main network".

Stage 2. Train the hypernetwork to generate task-specific adapters
Use a hypernetwork to generate task-specific adapters from the task description. Optimize the hypernetwork so that when the generated adapters are plugged into the main model, better performance is achieved on examples within that task.

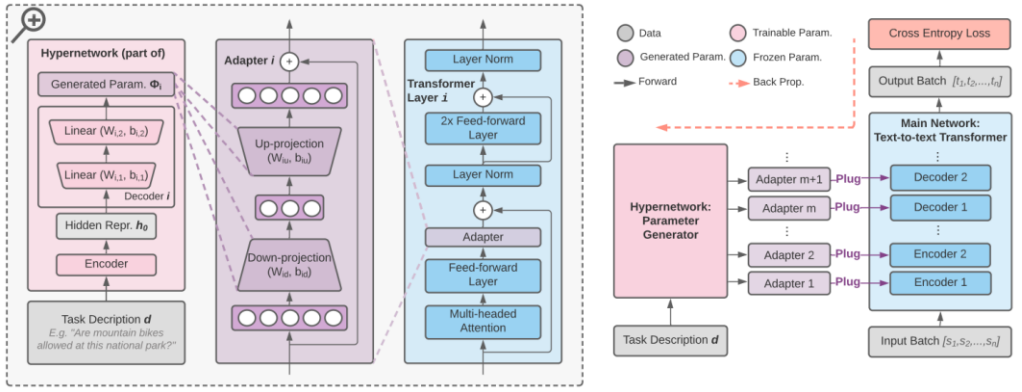

Illustration of Hypter Framework. Left: A hypernetwork generates parameters for a task-specific adapter that is plugged to a transformer layer in the text-to-text model. **Right:** The "adapted" main network is evaluated on the task examples. The final cross entropy loss is back-propagated to update the hypernetwork.

4. Performance and Analysis

ZEST Dataset [1] (Official Test Set)

	C@75	C@90
Bart-Large	10.91	3.98
+ Hypter	11.35 (+4%)	4.43 (+11%)

Repurposed SQuAD Dataset [3] (Test Set)

	SQuAD	NQ	NewsQA
Bart-Base	74.79±0.91	49.78±0.95	56.37±0.90
+ Hypter	75.53±0.68*	50.39±1.01*	56.41±0.85

- **Better generalization to novel tasks**
- **Better generalization to out-of-distribution inputs**
- **Sufficient #tasks and sufficient #examples per task are both necessary for Hypter training.**

[1] Weller, Orion, et al. "Learning from Task Descriptions." *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020. **[2]** Jin, Tian, et al. "Language to Network: Conditional Parameter Adaptation with Natural Language Descriptions." *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020. **[3]** Rajpurkar, Pranav, et al. "SQuAD: 100,000+ Questions for Machine Comprehension of Text." *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016.