Exploring the Benefits of Training Expert Language Models over Instruction Tuning

Main Idea: this paper looks to explore the author's discovery that training an expert LM fine-tuned on a single task can outperform a multi-task (MT) LM trained on hundreds of tasks (more specifically regarding multi-task performance). This goes against other findings that claim that scaling the number of tasks in MT-LMs is key to making them have stronger performance. Referring to LMs fine-tuned on a single task means a system of multiple Expert Language Models (ELMs), each fine-tuned on a single task, not a single LM trained on a single task.

OBS: Instruction-tuning -> fine-tuning LMs with instructions (prompts).

ELM Framework

- Training experts – two types of experts are trained:
    - Prompt Experts
        - Trained via PEFT through an adapter (an adapter layer is trained on top of the pre-trained LLM, with the pre-trained LLM's weights kept frozen).
        - Trained to perform well on a single prompt specific to the task.
    - Dataset Experts
        - Trained via regular fine-tuning of the pre-trained LLM's weights on a single task (all weights are updated).
        - Idea is to train an expert that will perform well to different prompts, so it can merge with other experts.

- Routing mechanism – Retrieval-of-Experts
    - Consists of constructing an Expert Library and training a dense retriever.
        - Each row in the Expert Library corresponds to an expert and consists of keys of S random training instances of that expert and a corresponding expert id.
            - S used was 100.
        - The dense retriever is a Sentence Transformer, and it also assumes that Q examples of the target task are available. It takes the embeddings of the input task and chooses the most relevant expert(s) based on each expert's similarity to this input task (based on the training instances stored for each expert and the target task instances).
            - Q used was 32.
- Merging of experts
    - The merging of Dataset Experts is also explored, retrieving more than one expert for an unseen task.
        - Merging does not make sense with Prompt Experts, since they were trained to perform well on a single prompt, therefore they would not be performant at this setting (at merging).
    - The merged LM ends up being created at the parameter level. It is a weighted-average (parameter-average) of the selected experts.
        - Since the parameters are merged, the inference cost will be the same as the inference cost of the single MT-LM trained on hundreds of tasks.

Experimental Setup

- 296 Prompt Experts, 36 Dataset Experts (on around 8 prompts each) trained.
- 50,000 samples used for training each classification task. 10,000 for each generative task.
    - On top of the pre-trained T5 model.
- 5 epochs used for training with a constant learning rate of 1e-4.
- Rouge-L score used for evaluating generative tasks.

Results – Prompt Experts

1. A single Prompt Expert significantly outperforms its dense MT-LM baseline (trained on hundreds of tasks).
    a. The single Prompt Expert that achieved this was trained on CosmosQA.
    b. Perhaps this means that the dataset being diverse is more important than the number of tasks trained?
2. The Retrieval-of-Experts (ROE) method with an oracle gate significantly outperforms all other models, including T0-11B (the base LLM used for the adapters was T5-3B) and GPT-3.
    a. This shows that improving the retrieval method is a promising area of future research.
3. A simple ROE approach outperforms T0-3B (the MT-LM baseline) on classification tasks, but not on generative tasks.
    a. A better ROE method reverses this.

b. Using more diverse data (in quantity) seems to help seems to help generative tasks (perhaps due to the higher complexity in text generation compared to classification?).

Results – Dataset Experts

1. There was negative task transfer when merging the adapter experts (Prompt Experts).
    a. Merging Prompt Experts results in worse performance – does not work.
2. Merging the fully fine-tuned experts (Dataset Experts) resulted in positive task transfer.
    a. Merging resulted in improved performance (merged capabilities > individual capabilities).

- The 3 datasets that show the best performance on unseen tasks (when training on a single task) are all commonsense reasoning datasets (for both merging and not merging).
    - Points to models trained on commonsense reasoning having higher generalization abilities to unseen tasks – commonsense reasoning data is higher quality data.
- Retrieval of the correct expert(s) seems important as the best expert on unseen generative tasks performed poorly on unseen classification tasks.

Benefits of Expert LMs over MT-LMs

- ELMs are less susceptible to negative task transfer on seen tasks (the tasks used for training).

- ELMs have continual learning abilities on new tasks without needing access to all the data at the same time.
- ELMs allow for merging experts on compositional instructions (merging of task prompts).

Limitations of ELMs over MT-LMs

- The method explored assumes a batch of the target task is available for RoE, which is not always a realistic assumption.
- MT-LMs bigger than 11B parameters, which might not suffer from negative task transfer due to increased capacity, were not analyzed.
- For some tasks, merging experts on compositional instructions may not be so simple.

My takeaways:

- A system of ELMs outperforming a single LM in a multi-task setting seems to show that the benefits of specialization outweigh the benefits of shared knowledge between tasks.
    - An ELM system also allows for choosing an expert trained on a task that resembles the target data – ensemble of closely-related experts sounds, in theory, better than a single LM fine-tuned on multiple tasks (that could be both relevant and irrelevant to the target task).
- More exploration is needed in the Retrieval-of-Experts (routing mechanism used) to alleviate the constraint of having training and target instances stored, as well as to

appropriate it to scenarios where we do not have examples of the target task available since this task would be unknown.