Custom Project: Multi-agent Fine-tuning for Specialization and Cooperation

Kevin Liu rliu79@jh.edu

Tom Wang xwang397@jh.edu

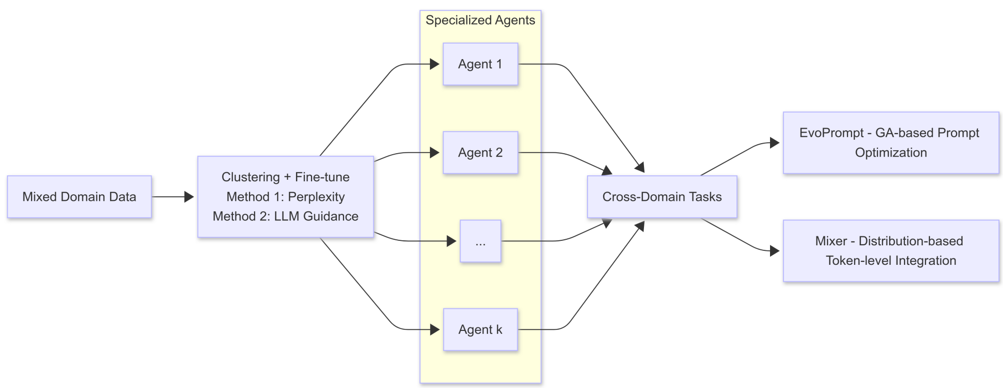
Matt Wang hwang302@jh.edu

Lena Wang lwang216@jh.edu

1 Introduction

In the rapidly evolving landscape of LLM finetuning, effective management and usage of large, diversified corpus always plays a significant role in developing successful and robust models. However, few work has addressed the fact that most datasets show **multi-domain** characteristics, where parts of the corpus each has consistent semantic phenomena and patterns within oneself, but conisderable gaps exist among different data point clusters. Traditional fine-tuning approaches often treat available data as a monolithic entity, overlooking the inherent multi-domain nature of many real-world corpora. Our method addresses this gap by introducing a **Multi-agent System (MaS)** finetuning workflow that trains multiple agents from a dataset by clustering data points and reinforcing colaboration.

The proposed methodology (Figure 1) begins with a data clustering phase, where the extensive corpus is segmented into k distinct groups, each representing a coherent domain-specific subcorpus. Each corpus yields a finetuned, specialized agent, thereby enhancing its ability to process and generate domain-relevant content with higher precision. The second step integrates these specialized agents into a cooperative system using various techniques. Extensive experiments would be conducted to systemmatically evaluate the performance of the proposed workflow against state-of-the-art baselines. Through the development of a robust multi-agent system, we anticipate significant improvements in domain-specific performance and cross-domain generalization, paving the way for more effective and scalable NLP solutions.



2 Related Work

Current Collaborative Studies for Multi-agent Systems

Merging, ensemble, and cooperation are the three main collaborative strategies for MaS. Merging involves combining models through, for example, a weighted average. Ensemble methods combine outputs from multiple LLMs, either before, during, or after inference. Cooperation strategies allow models to dynamically interact. Out of the three, cooperation was speculated to be the most flexible and promising, enabling agents to specialize and support each other under complex tasks [3].

Role-Based Agent Architectures for Knowledge Integration

LEGO, a multi-agent collaborative framework that assigns distinct roles to five LLM agents, was developed to generate human-like reasoning processes. The system includes a module that helps agents gather relevant background knowledge by reasoning from both sides (cause and effect) and another module that improves explanations by letting other agents give feedback and revise based on that feedback. LEGO significantly outperforms single-agent baselines on datasets like WIKIWHY and e-CARE, showing improvements through the collaboration of agents [2].

3 Methods

3.1 Datasets

Overall, we have the following dataset:

- **Mixed Dataset**: This is a noisy dataset consisting of multi-domain documents.
- **Cross-domain Dataset**: This is a dataset used to evaluate our model's ability to retain and structurally use information across domains to tackle hard and comprehensive questions. This data is split into a held-out set for GA prompt optimization and a test set for final performance evaluation.

3.2 Expert Training

In the first phase, our goal is to split the mixed corpus into specialized k piles of mostly disjoint subsets. Each subset will be the fine-tuning data for a specific expert agent.

3.2.1 Method 1: Clustering as Monte Carlo EM (perplexity)

We are inspired by VAE (Monte Carlo EM) [4] and view this clustering task as uncovering the latent structure of the mixed dataset D. In this setting, we are doing model finetuning and data clustering at the same time.

Here is the algorithm's detail:

First, instantiate k copies of a general pretrained model as the starting state of k experts, $e_1, e_2, \ldots, e_k$. Then let D be the mixed dataset. For a training epoch, we continue to fetch a new document $d \in D$, and query the current experts on their perplexity on this document. Then, we choose an expert e^* with probability inverse to the relative perplexity of that document under that expert, and then do a finetuning of this particular document on e^*.

The idea was to start with a uniform selection when the experts begin similar. After training on the first few sequences, we hope that randomness give each expert a unique finetune set that bias towards a particular area. And so it might make that expert more acquaint with ideas and terminologies in that field, decreasing the perplexity of related documents in the future training, giving model a high chance to see related documents again, enforcing the bias towards a particular expert area.

3.2.2 Method 2: Clustering via LLM guidance.

Inspired by the union-find algorithm, we propose another way to cluster the mixed corpus into piles, with guidance utilizing LLM's ability to identify related documents.

With $|D| = N$, we start by instantiating N clusters, each containing one unique document in D. Then for each cluster, we query the LLM with a summary of the documents in the cluster and ask it to find another document outside the cluster.

Overtime, the clusters will contain more and more documents, and we will perform merge operations between two clusters that mostly overlaps. We will keep merging until we have are left with only k clusters. After we form the clusters, we then perform finetuning of one expert on each pile.

3.3 Expert Cooperatioon

To enable agents to adapt to their roles and achieve effective collaboration, we propose two methods:

3.3.1 EvoPrompt: Genetic Prompt Optimization

Each agent begins by automatically generating a simple prompt based on its fine-tuned training data, analyzing patterns to identify core areas of expertise, such as *mathematical proofs* or *scientific analysis*, and drafting an initial description to define its role.

These prompts are then refined through the **EvoPrompt** framework, which uses a Genetic Algorithm (GA) to optimize both the agent's **description** and the global **instruction**. For each role, a diverse population of prompt variants is maintained and evolved using standard genetic operations [1][5]. Mutation tweaks phrasing, crossover merges effective segments from top-performing prompts, and selection favors variants that achieve high task accuracy and inter-agent agreement. The GA also refines global instructions, such as *Collaborate by sharing relevant insights and working in sequence*, to strengthen coordination.

A lightweight evaluator continuously monitors prompt performance based on logical consistency, task coverage, and efficiency. Underperforming variants are discarded, while high-performing ones are iteratively improved. This dual-track optimization drives each agent toward a clearer self-concept and more effective collaboration, resulting in stronger specialization and seamless teamwork across the system.

3.3.2 Mixer: Token-level Cooperation

To make cooperation more fine-grained, we are inspired by MoE architecture and hope to foster cooperation on a token by token level. We treat cooperation among agents as sampling under a convex combination of each agents' next token distribution. Our goal is then to train a mixer, that assign weights to experts at each prefix position, indicating how confident we should be for each agent on the next token.

Training of the mixer:
The mixer $M : H \to R$ is a probe on an expert language model that reads the last hidden state of the model and tries to give a scalar value, representing how confident a particular should be for this token. With clusters we get in the Expert Training phase, we have a map from document to particular expert model, i.e. $\hat{p} : D \to Z_n$. So for all token in a document d, we can train M to predict 1 if the hidden state belongs to the expert that are assigned d in its pile, and 0 otherwise.

4 Milestone and Plan

At the halfway point, our goal is to implement one of the data clustering–based expert training methods that enables the creation of specialized agents. Also, we plan to develop an initial form of cooperation training to facilitate basic inter-agent collaboration. The performance of this approach will be compared against directly fine-tuning a model on K-means clustered mixed domain data and cross-domain data to evaluate its advantages in specialization and coordination.

References

[1] Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *The Twelfth International Conference on Learning Representations*, 2024.

[2] Zhitao He, Pengfei Cao, Yubo Chen, Kang Liu, Ruopeng Li, Mengshu Sun, and Jun Zhao. Lego: A multi-agent collaborative framework with role-playing and iterative feedback for causality explanation generation. pages 9142–9163, 01 2023.

[3] Jinliang Lu, Ziliang Pang, Min Xiao, Yaochen Zhu, Rui Xia, and Jiajun Zhang. Merge, ensemble, and cooperate! a survey on collaborative strategies in the era of large language models, 07 2024.

[4] Achille Thin, Nikita Kotelevskii, Arnaud Doucet, Alain Durmus, Eric Moulines, and Maxim Panov. Monte carlo variational auto-encoders. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10247–10257. PMLR, 18–24 Jul 2021.

[5] Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Wang Yanggang, Haiyu Li, and Zhilin Yang. GPS: Genetic prompt search for efficient few-shot learning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8162–8171, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.