Selecting and Merging: Towards Adaptable and Scalable Named Entity Recognition with Large Language Models

Zhuojun Ding[1], **Wei Wei**[*1] **and Chenghao Fan**[1]

[1] School of Computer Science & Technology, Huazhong University of Science and Technology

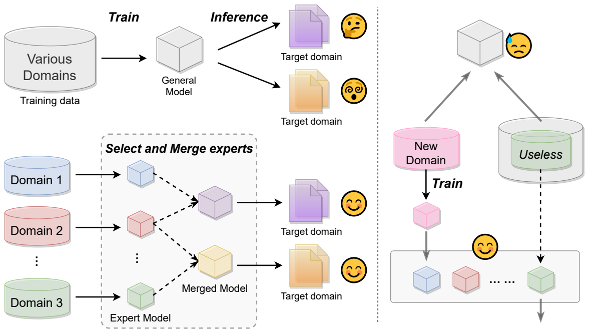
{dingzj, weiw}@hust.edu.cn, facicofan@gmail.com

Abstract

Supervised fine-tuning (SFT) is widely used to align large language models (LLMs) with information extraction (IE) tasks, such as named entity recognition (NER). However, annotating such fine-grained labels and training domain-specific models is costly. Existing works typically train a unified model across multiple domains, but such approaches lack adaptation and scalability since not all training data benefits target domains and scaling trained models remains challenging. We propose the SaM framework, which dynamically **S**elects **a**nd **M**erges expert models at inference time. Specifically, for a target domain, we select domain-specific experts pre-trained on existing domains based on (i) domain similarity to the target domain and (ii) performance on sampled instances, respectively. The experts are then merged to create task-specific models optimized for the target domain. By dynamically merging experts beneficial to target domains, we improve generalization across various domains without extra training. Additionally, experts can be added or removed conveniently, leading to great scalability. Extensive experiments on multiple benchmarks demonstrate our framework's effectiveness, which outperforms the unified model by an average of 10%. We further provide insights into potential improvements, practical experience, and extensions of our framework.[1]

1 Introduction

Large language models (LLMs) demonstrate remarkable performance across a wide range of tasks (Achiam et al., 2023; Yang et al., 2024; Guo et al., 2025a), but still struggle with information extraction (IE) tasks (Xu et al., 2024; Ding et al., 2024b; Fan et al., 2024b), such as Named Entity Recognition (NER). The inherent gap between task formulations and LLM training objectives is a critical factor underlying this limitation. To mitigate this, supervised fine-tuning (SFT) has become a widely used strategy, demonstrating significant improvements (Wang et al., 2023a; Zhou et al., 2024b; Fan et al., 2025).



(a) Comparision of adaptability (b) Comparision of scalability

Figure 1: (a) Existing methods train a general unified model across multiple domains, while we dynamically select and merge expert models at inference time. (b) A trained system struggles to accommodate changes in training data, while we flexibly add or remove expert models, ensuring great scalability.

However, annotating data and training domain-specific models each time is costly, particularly for fine-grained IE tasks. Most existing approaches collect large-scale training data from multiple domains to train a unified model (Wang et al., 2023a; Sainz et al., 2024; Yang et al., 2025). Although such models exhibit cross-domain generalization capabilities, they frequently exhibit suboptimal performance in both in-domain and out-of-domain test scenarios. This limitation arises primarily because (1) not all training samples universally enhance performance on a given target domain (Liu et al., 2024; Zhou et al., 2024a), and (2) inherent conflicts may emerge across heterogeneous domains during joint training, leading to compromised optimization efficacy (Sainz et al., 2024; Yang et al., 2025; Fan et al., 2024a). Additionally, even when data from the target domain is available, effectively integrating it into a trained model without compromising performance remains challenging.

[*] Corresponding author

[1] https://github.com/Ding-ZJ/SaM

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9869–9886
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

To address these issues, we adopt a model merging strategy (Ilharco et al., 2023) to dynamically select domain-specific models for different target domains and fuse their parameters to obtain task-specific models. Specifically, we first train multiple expert models in different domains with available data. Then, we design the **SaM** framework to derive task-specific models by **S**electing **a**nd **M**erging experts from two perspectives: (i) Domain similarity. We assess the domain similarity between the target domain and each expert model, and select the most relevant experts for parameter fusion to create a task-specific model tailored to the target domain. (ii) Sampling evaluation. We randomly sample k data instances from the target domain and integrate the predictions of all experts as pseudo-labels (no ground truth labels required). Based on these labels, we assess the performance of each expert on the sampled data and select the best-performing experts to merge into another task-specific model. The two perspectives synergistically complement each other: domain similarity provides a high-level, coarse-grained assessment that establishes theoretical priors, while sampling evaluation delivers a fine-grained, empirical quantification of expert performance to yield actionable practical guidance. By integrating the outputs of both task-specific models, we achieve more comprehensive results.

Compared to previous methods (Figure 1), we allow for task-specific customization across diverse target domains, providing improved generalization ability without extra training. It also provides great scalability, as experts can be easily added or removed based on practical needs. Notably, our approach is orthogonal to previous studies. By leveraging their practical insights, we can train more effective expert models, thereby enhancing the performance of our framework. In terms of resource requirements, our approach does not incur additional training costs. By leveraging parameter-efficient fine-tuning methods (Hu et al., 2022), we only introduce minimal storage overhead. Additionally, by employing either strategy individually or further integrating the target models derived from both strategies, we can achieve comparable performance without incurring additional inference costs.

In summary, our contributions are as follows: **(1)** We introduce a model-merging paradigm for LLM-based Named Entity Recognition, enhancing adaptability and scalability. **(2)** We propose a model selection strategy based on domain similarity and sampling evaluation, which effectively selects expert models beneficial to the target domain for merging. **(3)** Experimental results demonstrate the effectiveness of our framework, which outperforms the unified model by an average of 10% and by up to 20% in certain domains. Further experiments analyze potential improvements, practical experience, and framework generalizability, providing deeper practical insights.

2 Related Works

LLMs for Information Extraction Current LLMs-based IE mainly fall into two paradigms.

One paradigm uses larger models. Training them needs significant computational resources, and fine-tuning them specifically for IE tasks may be not cost-effective. However, these models excel in instruction-following and reasoning. Therefore, such methods focus on optimizing task instructions, reasoning strategies, or in-context learning (ICL) demonstrations. Li et al. (2023) show that code-style prompts enhance IE tasks. Pang et al. (2023) and Tong et al. (2025) prompt LLMs with more comprehensive information to improve task understanding. Xie et al. (2023) and Wan et al. (2023) introduce reasoning techniques such as Chain-of-Thought (CoT) to guide the model in step-by-step task completion. Xie et al. (2024) employ self-consistency to generate reliable ICL examples.

Another paradigm uses smaller models. While these models have weaker instruction-following capabilities, they require much fewer training resources. Such methods enhance LLMs through supervised fine-tuning (Wang et al., 2023a). Many studies design optimization strategies on the data side. Yang et al. (2025) resolves conflicts and redundancy in training data. Zhou et al. (2024b) distills more diverse data from ChatGPT. Li et al. (2024) formats training data in code style. Sainz et al. (2024) enriches instructions with detailed task descriptions. Ding et al. (2024a) emphasizes negative samples. In addition to instruction tuning, Qi et al. (2024) further employs alignment training (Rafailov et al., 2024), and Guo et al. (2025b) incorporates contrastive learning objectives. Instead of training one universal model, we train several domain experts and design a merging method to improve adaptability and scalability.

Additionally, the backbone model is also critical. Recently, code-based LLMs have gained popularity, as they may better suit IE tasks than natural language-based LLMs (Li et al., 2023).

Model Merging Model merging integrates multiple task-specific models at the parameter level to create a unified model, which could handle multiple tasks simultaneously and exhibit better out-of-domain generalization. Unlike multi-task learning, model merging reuses existing models, reducing computational and data demands since only model parameters are needed. Beyond simple parameter averaging, Matena and Raffel (2022) assigns different importance to model parameters. Ilharco et al. (2023) applies arithmetic operations for finer control over model behavior. Jin et al. (2023) enforces output consistency between the merged model and its constituent models. Yu et al. (2024) and Yadav et al. (2023) mitigate inter-model interference by addressing parameter redundancy or sign inconsistency, and weight sparsity, respectively. Lu et al. (2024) decomposes model parameters into shared and task-specific components. In this paper, we introduce model merging to improve adaptability and scalability across different target domains.

3 Methodology

In this section, we first provide an overview of the training process of domain expert models. Then we explain our SaM framework for selecting and merging experts, as shown in Figure 2.

3.1 Training Domain Experts

Data Collection. We first collect more than 20 commonly used NER datasets and classify them into six domains based on their sources: News, Social media, Biomedical, STEM (Science, Technology, Engineering, and Mathematics), Legal, and Traffic. We remove 90% of the NA data that contains no entities. Through sampling or redundancy, we limit the total samples per domain to between 10,000 and 50,000. The number of sampled instances from each dataset was proportional to the number of entity types it contained. Detailed information is provided in Appendix A.

Training Data Construction. Refering to practice of prior studies (Wang et al., 2023a; Qi et al., 2024), we format the raw data into task instructions, inputs, and outputs for training.

The task instructions consist of: **(1)** Data source description: A brief overview of the dataset source. **(2)** Entity type description: Concise definition of entity types. **(3)** In-context learning demonstrations: $1 \sim 5$ randomly selected input-output pairs from the training data. **(4)** Label drop: Excluding the requirement for recognizing certain entity types. **(5)** Label masking: Replacing entity labels with abstract placeholders such as "Type1". Empirically, we apply *(1)*, *(2)*, and *(3)* to 70% of the data; *(4)* to 30%; and *(5)* to 5%. These modifications are applied independently, except *(5)*, which should co-occur with *(2)*. For output parts, we adopt three formats: JSON (e.g., "{entity span: entity type}"), enumeration (e.g., "Type: span1, span2, ..."), and natural language descriptions.

These strategies help enhance model robustness to some extent. However, we claim this is not an optimal configuration, as our focus is not on training a best-performing model.

Model Training. Following prior work, we train the model using instruction tuning. Given a dataset $D_A = \{(I, X, Y)\}$ from domain A, where I is the task instruction, X is the input sequence, and $Y = \{y_i\}_{i=1}^{L}$ is the output sequence (i.e., entity predictions), the training loss of the domain-specific expert model $\mathcal{M}_A$ is defined as:

$$\mathcal{L}_{\theta_A} = -\sum_{D_A}\sum_{t=1}^{L} \log P_{\theta_A}(y_t \mid I, X, y_{<t}) \qquad (1)$$

where θ_A denotes the parameters of $\mathcal{M}_A$.

3.2 Selecting and Merging Experts

When handling a specific target domain, we select a subset of expert models and fuse their parameters to obtain task-specific models. As shown in Figure 2, this process is conducted from two perspectives.

Selecting with Domain Similarity. Given a domain with raw data $D_A = \{x_i\}$, we obtain the corresponding data embeddings $H_A = \{h_i\}$ through a text encoder. The domain embedding is then defined as the centroid of these data embeddings:

$$h_A = \frac{1}{|H_A|}\sum_{h_i \in H_A} h_i \qquad (2)$$

We compute domain embeddings $\{h_{e_i}\}$ for domains of all expert models and h_t for the target domain. Then we compute the similarity between the expert domains and the target domain with cosine distances. Finally, the top-m similar expert models are selected for model merging.

The domain embedding inherently captures the data distribution in the embedding space. Thus, in theory, the selected expert models exhibit a certain degree of similarity and are expected to perform well in the target domain due to the resemblance in these data distributions.

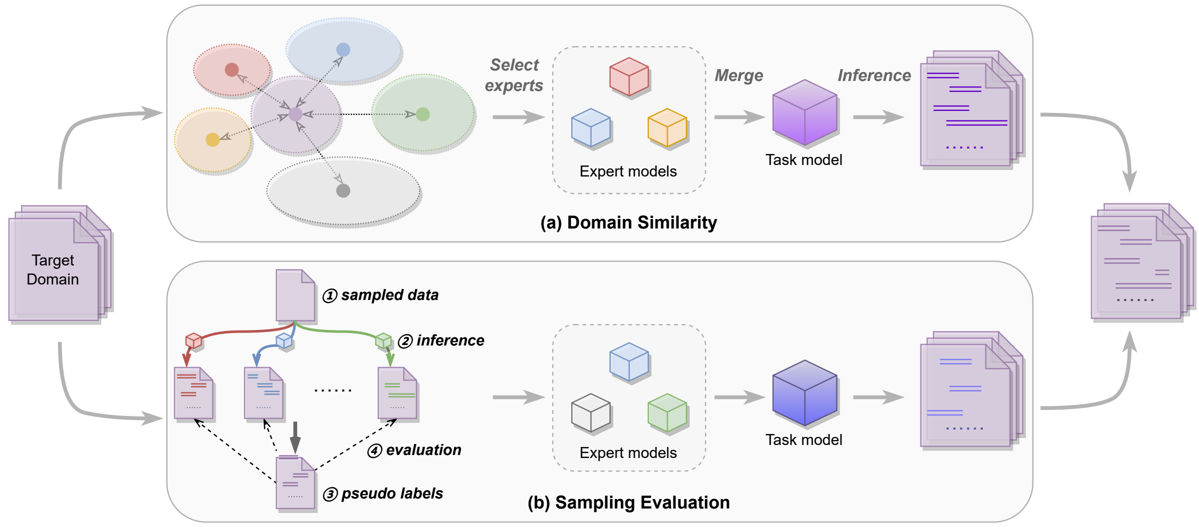


Figure 2: Framework overview. Given a target domain, we select expert models from two perspectives: **(a) Domain Similarity**, which selects experts from domains most similar to the target domain. We compute the centroid of all data embeddings as the domain embedding and measure similarities by cosine distance. **(b) Sampling Evaluation**, which selects experts with better performances on sampled instances from the target domain. To reduce reliance on ground-truth labels, we ensemble predictions from all experts as (pseudo) labels. We merge models within each expert subset to obtain two task-specific models. The final result integrates the outputs of the two task models.

Selecting with Sampling Evaluation. The selection of experts based on domain similarity is theoretically sound. However, in practice, we observe that the model with the highest domain similarity to the target domain does not always yield the best performance. Thus, we propose another selection strategy driven by model performance.

Specifically, we first randomly sample k data instances from the target domain. Then each expert model generates predictions for them. To reduce dependence on ground-truth labels, we aggregate predictions via majority voting to construct pseudo-labels, which are subsequently used to assess expert performance. The top m experts with the highest performance are selected for model merging.

Unlike domain similarity-based selection, this approach prioritizes practical effectiveness. As a result, the selected experts generally perform better individually in the target domain. However, we aim to obtain a superior task model through model merging, where individual performance is not the sole determining factor.

Merging Experts. Given a base model $\mathcal{M}_{base}$ and the supervised fine-tuned model $\mathcal{M}_{sft}$, we denote their parameters as θ_{base} and θ_{sft}, respectively. The delta parameter $\delta_{sft} = \theta_{sft} - \theta_{base}$ serves as a parametric representation of the model's learned capabilities and is also referred to as the task vector. Given multiple task-specific models $\{\mathcal{M}_{sft_i}\}$, we can merge them into a unified model $\mathcal{M}_{merge}$ with diverse capabilities (Matena and Raffel, 2022; Ilharco et al., 2023):

$$\theta_{merge} = \theta_{base} + \mathrm{Merge}(\delta_{sft_1}, \delta_{sft_2}, \cdots) \quad (3)$$

where $\mathrm{Merge}(\cdot)$ denotes the model merging technique, such as simple averaging and task arithmetic (Ilharco et al., 2023). We employ the Ties-Merging (Yadav et al., 2023) method, which addresses parameter redundancy and sign inconsistency to mitigate inter-model interference when merging multiple models. We selected two sets of expert models based on Domain Similarity (**DS**) and Sampling Evaluation (**SE**), respectively. These experts are subsequently merged to obtain two task-specific models, $\mathcal{M}_{DS}$ and $\mathcal{M}_{SE}$.

3.3 Inference

For a target domain, we obtain two task-specific models, $\mathcal{M}_{DS}$ and $\mathcal{M}_{SE}$, following the methodology described in Section 3.2. Each model independently generates predictions, producing two output sets, Y_{DS} and Y_{SE}. Taking the intersection of two sets of predictions typically enhances reliability. However, our two task-specific models are already tailored for the target domain and capture different and complementary perspectives. Therefore, we adopt their union as the final result.

		CrossNER					MIT		Average
		AI	Literature	Music	Politics	Science	Movie	Restaurant	
Recent Studies	InstructUIE	48.40	48.80	54.40	49.90	49.40	63.00	20.99	47.84
	UniNER	**62.90**	$\underline{64.90}$	70.60	66.90	$\underline{70.80}$	61.20	35.20	61.79
	GoLLIE	59.10	62.70	67.80	57.20	55.50	63.00	43.40	58.39
	KnowCoder	60.30	61.10	70.00	72.20	59.10	50.00	48.20	60.13
	GLiNER	57.20	64.40	69.60	$\underline{72.60}$	62.60	57.20	42.90	60.90
	B2NER	59.00	63.70	68.60	67.80	**72.00**	67.60	**53.30**	$\underline{64.57}$
Fully-trained	Llama	54.11	63.44	$\underline{72.53}$	62.92	59.09	64.81	49.68	60.94
	Qwen	51.38	53.46	61.12	54.99	59.39	66.66	52.69	57.10
SaM (Ours)	Llama	$\underline{60.98}_{12.7\%}$	$\mathbf{66.93}_{5.50\%}$	$\mathbf{73.53}_{1.4\%}$	$\mathbf{74.47}_{18.4\%}$	$62.60_{5.9\%}$	$\mathbf{72.17}_{11.4\%}$	$\underline{52.99}_{6.7\%}$	$\mathbf{66.24}_{8.70\%}$
	Qwen	$60.01_{15.8\%}$	$61.99_{16.0\%}$	$65.93_{7.9\%}$	$67.05_{21.9\%}$	$62.41_{5.1\%}$	$\underline{71.65}_{7.50\%}$	$52.90_{0.4\%}$	$63.13_{10.6\%}$

Table 1: Experimental results. We compare our method with recent studies and our fully-trained model (e.g., a unified model trained on all data we used). The best results are highlighted in bold, while suboptimal results are underlined. The right subscript denotes the percentage improvement compared to the fully trained model.

4 Experiments

4.1 Setup

Benchmarks, Baselines, and Metrics We evaluate our framework on two widely used benchmarks CrossNER (Liu et al., 2021) and MIT (Ushio and Camacho-Collados, 2021), which contain datasets from seven domains (AI, Literature, Music, Politics, Science, Movie, and Restaurant) in total.Our experiments are under zero-shot settings (i.e., no labeled target domain data), follow prior work. The source data for training and the target data for evaluation have different distributions.

We introduce two types of baselines for comparison. The first is the **Fully-trained** model, a single unified model trained on data from all domains using the same training configuration as us. This serves as the primary baseline to assess the effectiveness of our framework. The second includes recent studies that also train unified models but incorporate other advanced training optimizations, including **InstructUIE** (Wang et al., 2023a), **UniNER** (Zhou et al., 2024b), **GoLLIE** (Sainz et al., 2024), **KnowCoder** (Li et al., 2024), **GLiNER** (Zaratiana et al., 2024), and **B2NER** (Yang et al., 2025). Most of these models use LLMs as the foundation, except for GLiNER, which contains only 300 million parameters.

Following prior studies (Wang et al., 2023a), we use the entity-level micro-F1 score as the evaluation metric, where both the entity boundary and entity type should be correctly predicted.

Implementations We employ models from the Qwen and Llama series as base models for our experiments. Specifically, we adopt the base version of Qwen2.5-7B and Llama3.1-8B as foundations and train expert models using LoRA (Hu et al., 2022). We employ the all-MiniLM-L6-v2[2] text encoder to produce text embeddings. We set m (the number of selected models for merging) to 3. We set k (the number of sampled data instances) to 10. More details are reported in Appendix A.

4.2 Main Results

As shown in Table 1, our approach significantly outperforms the fully trained model across all target domains, achieving an average improvement of approximately 10%, with gains of up to 20% in specific domains. This demonstrates the effectiveness and superior domain adaptability of our approach. To ensure practical comparability, we also compare our results with recent studies. These methods employ various training optimization strategies. For example, B2NER mitigates redundant and conflicting information in the training data. These techniques are orthogonal to ours. Notably, comparing these methods with our fully-trained model suggests that refining our training configuration could enhance our expert models and further improve the performance of our approach. Our approach may incur slight computational and storage overhead, which is acceptable, as discussed in Appendix C.

4.3 Ablation Studies

We conduct ablation studies to validate the effectiveness of our design, as shown in Table 2: (1) **w/o Merging**, which directly uses the best expert model. (2) **w/o Domain Similarity**, which selects experts solely based on Sampling Evaluation. (3) **w/o Sampling Evaluation**, which selects experts

[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

		AI	Literature	Music	Politics	Science	Movie	Restaurant	Average
Llama	w/o Merging	53.72	59.95	71.61	71.88	59.09	66.43	50.62	61.90
	w/o Domain Similarity	58.43	59.87	72.89	74.20	59.92	68.91	51.69	63.70
	w/o Sampling Evaluation	56.42	63.92	72.25	72.99	61.71	71.08	52.48	64.41
	w/o Selection	60.14	63.74	**74.83**	73.00	**63.68**	67.30	47.64	64.33
	SaM (Ours)	**60.98**	**66.93**	73.53	**74.47**	62.60	**72.17**	**52.99**	**66.24**
Qwen	w/o Merging	57.90	59.01	65.33	65.73	60.21	65.45	51.11	60.68
	w/o Domain Similarity	58.57	61.61	65.63	**68.51**	59.67	70.66	52.69	62.48
	w/o Sampling Evaluation	58.23	60.39	64.11	64.57	61.00	71.04	50.11	61.35
	w/o Selection	57.47	60.93	64.66	60.28	61.08	69.43	47.78	60.23
	SaM (Ours)	**60.01**	**61.99**	**65.93**	67.05	**62.41**	**71.65**	**52.90**	**63.13**

Table 2: Ablation studies. Removing certain components typically results in performance degradation, confirming their significance. The best results are in bold, and the suboptimal ones are underlined.

	Mode1	Mode2	Mode3	SaM (Ours)
AI	60.36	58.19	**61.31**	60.01
Literature	56.86	**62.68**	58.90	61.99
Music	63.78	66.50	**66.85**	65.93
Politics	66.05	**68.27**	61.30	67.05
Science	58.37	61.73	**63.12**	62.41
Movie	70.66	70.66	70.50	**71.65**
Restaurant	52.82	52.85	52.23	**52.90**
Average	61.27	62.98	62.03	**63.13**

Table 3: Merging into a single task model (based on Qwen). "Mode*i*" denotes the method used to further extract a final set from two expert sets for model merging.

based on Domain Similarity. (4) **w/o Selection**, which merges all experts without selection. Results demonstrate that the merged model consistently outperforms the best individual expert, even when the selected models are not necessarily optimal. Overall, both expert selection strategies are effective and complementary, yielding the best results when combined. However, in some cases, using a single selection strategy or none at all yields better results, likely due to expert redundancy or insufficiency, as we fix the number of selected experts to $k = 3$ in our experiments. Further analysis of k are presented in Section 4.5

4.4 Merging into a Single Task Model

Since our approach employs two task-specific models, the inference cost is doubled. To mitigate this, we derive a single set from the two selected expert sets, reducing the number of task models to one. We propose and evaluate three strategies, with results in Table 3: (1) **Mode1** leverages the intersection of the two selected expert sets. (2) **Mode2** normalizes the evaluation metrics across both selection strategies (e.g., the domain similarity scores and F1-scores on sampled data points) to a common scale and selects the top three experts. (3) **Mode3** takes the union of the two expert sets while limiting the total number of selected experts to three. Experimental results show that Mode2 and Mode3 achieve comparable performance to us, making them effective alternatives without increasing inference costs. We refer to these as the economic versions of our framework (SaM$_{eco}$).

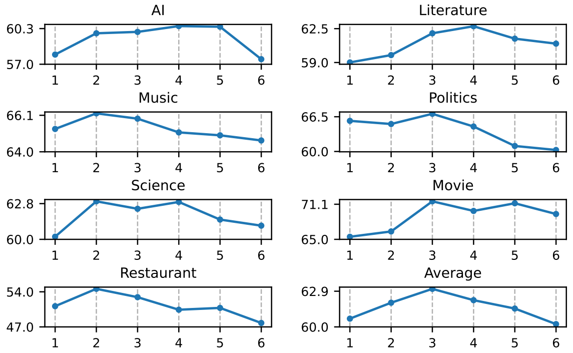


Figure 3: Performance changes with the number of experts. The horizontal axis is the number of experts for merging, and the vertical denotes the F1 scores.

4.5 Numbers of Experts for Merging

This section analyzes the impact of the number of merged models, k, with results shown in Figure 3. Here, $k = 1$ denotes the performance of the best individual expert, while $k = 6$ corresponds to merging all expert models. The optimal k varies across target domains, typically ranging from 2 to 4. We set $k = 3$ as it yields the best average performance. Notably, this corresponds to the merging algorithm as well. We adopt the Ties-Merging technique, where selecting $2 \sim 4$ models for merging is a commonly used configuration.

Source Domains	AI	Literature	Music	Politics	Science	Movie	Restaurant	Average
Three	56.42	63.01	71.69	72.24	61.17	69.91	50.62	63.58
Six	60.98	66.93	73.53	74.47	62.60	72.17	52.99	66.24
Nine	59.90	66.93	73.53	73.61	62.78	72.17	53.84	66.11

Table 4: Performance under different number of source domains. Using six source domains yields better results than three, but increasing to nine provides no further gains.

	AI	Literature	Music	Politics	Science	Movie	Restaurant	Average
Fully-trained	70.40	71.81	75.44	79.41	50.71	58.78	69.33	67.98
Sam(Ours)	83.45	71.81	77.11	81.56	77.14	78.01	73.33	77.49

Table 5: Performance under extreme scenarios with only one test sample (average of five trials). Our method still functions properly and adapts better than the fully trained single-model approach.

4.6 Number of Source Domains

We set six source domains in our experiments. This section presents a preliminary analysis of how the number of source domains affects performance, as shown in Table 4. While setting more source domains increases the diversity of candidate models during selection, it does not always lead to better results. The performance relies on the similarity between source and target domains, including conceptual relevance, data distribution, and overlap in entity types. Therefore, it's important to balance the number of source domains, their similarity to target domains, and the overall complexity.

4.7 Limited Target Resources Scenarios

The model selection process leverages target-domain raw texts for domain similarity calculation and sampling evaluation, typically a few hundred samples for the former and 10 for the latter. To test our method under more constrained settings, we simulate an extreme case with only one target-domain instance. As shown in Table 5, our method remains effective. However, very small sample pools can undermine the stability of expert selection, especially for the strategy using domain similarity. In such cases, we recommend using data augmentation to expand the sample pool.

4.8 Weighting Experts for Merging

We employ two metrics, *domain similarity* and *sampling evaluation*, to select expert models. These metrics reflect the importance of experts. Our framework does not consider the importance of experts but instead assigns equal weight to all expert models. To investigate the impact, we propose two simple weighting strategies, as shown in Table 6: (1) **Mode1** empirically assigns weights $(1.5, 1.0, 0.5)$ to the top three selected experts. (2) **Mode2** uses the middle-ranked metric value as a normalization factor to scale the three experts' metrics for weighting. Experimental results suggest that weighting has the potential to improve performance (Mode1), aligning with intuitive expectations. However, excessive reliance on heuristics may not always be justified. For example, while Mode2 theoretically provides a more precise weighting based on expert importance, it underperforms compared to Mode1 and Ours.

	Mode1	Mode2	SaM (Ours)
AI	**60.07**	59.33	60.01
Literature	**62.00**	62.03	61.99
Music	**66.35**	65.17	65.93
Politics	65.54	66.45	**67.05**
Science	62.27	62.15	**62.41**
Movie	71.20	71.48	**71.65**
Restaurant	**53.84**	52.84	52.90
Average	63.04	62.78	**63.13**

Table 6: Weighting experts for merging. "Mode*i*" denotes the method used to weight model parameters.

4.9 Finer Adaptation for Target Domains

Beyond the weighted merging strategy in Section 4.8, another potential approach for improvements is adopting finer adaptation. Specifically, we apply clustering to divide the target domain into multiple splits, selecting and merging expert models separately for each. However, as shown in Figure 4, this finer adaptation does not enhance performance. Instead, it results in an overall decline.

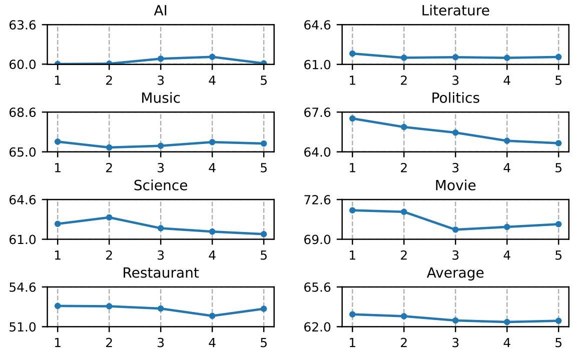


Figure 4: Performance changes with the number of data splits. The horizontal axis is the number of splits, and the vertical denotes the entity-level F1 scores.

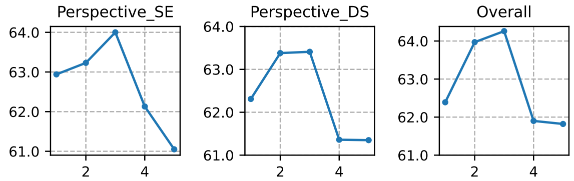


Figure 5: Performance changes with the number of data splits. We aggregate data from all domains into a unified dataset and subsequently conduct the split analysis.

To investigate this further, we aggregate data from all seven domains into a single dataset and conduct experiments. As shown in Figure 5, clustering improves performance across various strategies, including Sampling Evaluation (SE), Domain Similarity (DS), and their combination. Notably, despite the dataset spanning seven domains, the best performance is achieved when clustering divides it into only two or three groups. This is intuitively reasonable, as these seven domains originate from two broader datasets (CrossNER and MIT). The above analysis indicates that while finer adaptation to the target domain may bring improvements, excessive refinement without constraints may be counterproductive. For example, treating each data point as a distinct domain might seem optimal in theory but leads to poor performance in practice.

4.10 Analysis of Merging Technique

In addition to the Ties-Merging algorithm (**ties**) we employed, this section analyzes alternative merging strategies, including **linear**, **dare** (Yu et al., 2024), and their combinations. As shown in Table 7, dare and ties are effective. We do not present the performance of linear strategy, as this strategy leads to significant degradation. Specifically, models merged via the linear approach still produce some meaningful content but almost lose the ability to generate structured outputs, which is crucial for NER and other IE tasks. Consequently, their performance is bad, though minor improvements can be achieved through extensive post-processing on model outputs. However, combining dare with linear yields improved results. Both dare and ties address the issue of parameter redundancy for merging. These findings suggest that handling parameter redundancy is crucial for NER and similar structured output tasks. Additionally, the relatively weaker performance of the dare-ties combination may stem from excessive redundancy reduction, which could compromise useful capabilities of models.

	dare-linear	**ties**	**dare-ties**
AI	54.67	**60.01**	52.57
Literature	56.22	**61.99**	55.81
Music	**66.31**	65.93	66.22
Politics	**69.11**	67.05	66.42
Science	62.01	**62.41**	61.07
Movie	68.31	**71.65**	65.12
Restaurant	52.12	**52.90**	51.40
Average	61.25	**63.13**	59.80

Table 7: Comparison of different merging techniques. We compare linear, dare, ties, and some combinations.

	Llama			**Qwen**		
	T-SC	E-SC	Ours	T-SC	E-SC	Ours
AI	54.20	60.31	60.98	51.47	57.36	60.01
Literature	64.13	58.98	66.93	54.61	56.08	61.99
Music	70.01	71.06	73.53	61.58	68.01	65.93
Politics	63.58	66.27	74.47	55.76	65.99	67.05
Science	59.01	60.03	62.60	58.73	60.99	62.41
Movie	64.72	67.78	72.17	66.58	67.86	71.65
Restaurant	50.21	50.31	52.99	52.50	50.43	52.90
Average	60.84	62.11	**66.24**	57.32	60.96	**63.13**

Table 8: Comparison with self-consistency (SC) methods. T-SC employs a fully trained model to generate multiple outputs by adjusting the Temperature hyperparameter to the ensemble, while E-SC ensemble outputs from different Expert models.

4.11 Comparing with Self-Consistency

Considering that we trained multiple expert models, an intuitive approach is self-consistency (SC) (Wang et al., 2023b), which ensembles multiple outputs through voting. Results are shown in Table 8, where **T-SC** employs a full-trained model to generate multiple outputs by adjusting the **T**emperature hyperparameter to ensemble, while

E-SC ensemble outputs from different **E**xperts (though this slightly deviates from the strict definition of "self"-consistency, we refer to "SC" for simplicity). For our NER task, traditional SC (T-SC) shows limited improvement, while E-SC offers more significant gains due to greater output diversity. However, both SC strategies perform worse than our method and require higher inference costs.

4.12 Framework Generalization

Non-strict Domain In our experiments, each expert corresponds to a domain with real-world significance, such as news or law. To explore a more flexible scenario, we further test on non-strict domains. Specifically, we cluster each domain dataset into five subsets, and then sample one subset from each domain to create five new datasets, which are used to train five new expert models. The results in Table 9 indicate that our framework remains effective. Notably, the experts in this setting are no longer tied to specific domains but function as general-domain models while retaining diverse capabilities. This suggests that the key requirement is a set of experts with complementary strengths, which can enhance overall performance through mutual reinforcement.

	Expert Model					Ours
	E1	E2	E3	E4	E5	
AI	56.98	57.98	56.26	48.51	55.46	**60.88**
Literature	65.38	65.45	55.30	56.27	62.20	**67.03**
Music	62.38	66.00	63.63	63.88	65.17	**67.42**
Politics	60.94	**66.49**	63.52	57.54	58.50	66.09
Science	62.56	63.31	61.38	58.23	65.72	**66.75**
Movie	**68.75**	61.97	66.58	64.68	65.02	67.98
Restaurant	44.06	34.94	49.57	50.05	43.35	**52.66**
Average	60.15	59.45	59.46	57.02	59.35	**64.12**

Table 9: Analysis of experts for non-strict domains. We employ clustering to build five source domains and train expert models.

Multilingual Scenarios We extend our framework to multilingual scenarios and conduct preliminary experiments on six languages from the WikiANN dataset (Pan et al., 2017), including German (de), English (en), Spanish (es), Dutch (nl), Russian (ru), and Chinese (zh). We train a model for each language. When evaluating a target language, the model trained on that language is not used. The results in Table 10 provide several initial evidence that our framework has the potential to extend to multilingual scenarios.

	Expert Model						Ours
	de	en	es	nl	ru	zh	
de	–	80.69	80.75	81.00	81.74	74.99	**82.42**
en	72.29	–	77.10	**77.51**	73.08	69.18	76.84
es	86.09	87.28	–	**91.30**	88.46	78.85	89.67
nl	84.31	86.75	85.02	–	84.79	81.41	**87.59**
ru	75.91	75.68	74.44	76.93	–	61.29	**78.16**
zh	49.07	49.18	46.56	48.21	49.93	–	**51.37**
Avg	73.53	75.92	72.77	74.99	75.60	73.14	**77.68**

Table 10: Analysis of experts for different languages. When evaluating a target language, the model trained on that language is not used.

5 Future Work

Unified IE and Other Tasks We conduct experimental analyses with NER as a case study. Some prior studies train a unified model for multiple IE tasks, including NER, relation extraction (RE), event extraction (EE), etc. Our framework can be naturally extended to a broader IE setting by incorporating additional IE data to train IE experts. Additionally, our method extends beyond these tasks to a wide range of applications.

Detailed and Complete Design Our extended experiments investigated several optimization strategies. Further improvements could be realized by: (1) Incorporating more task-specific designs. For instance, alongside domain-level similarity, we could also leverage entity-type similarity when selecting source models. (2) Dynamically determining the number of merged models, k. Model merging may be unnecessary for certain target domains. As shown in Appendix B, when the target and source domains coincide, the single corresponding model already delivers optimal results. Section 4.5 demonstrates that the best choice of k varies across different target domains. Additionally, our framework is agnostic to the model architecture and readily extends to other model types other than the Llama and Qwen LLM families.

6 Conclusion

We propose the Select and Merging (SaM) framework for NER, which dynamically selects valuable domain expert models for the target domain and employs model merging to obtain the task-specific model. Compared to prior studies, we possess superior adaptability and scalability. Experimental results demonstrate the effectiveness of our framework. Extensive analysis further provides insights into potential improvements, practical experience, and broader extensions of our approach.

Limitations

We acknowledge the following limitations of our work: (1) Maintaining multiple expert models introduces some additional storage overhead, despite the use of LoRA. (2) For domain similarity calculation and clustering analysis, we simply employed a widely used encoder model from the HuggingFace repository to obtain text embeddings. Further optimization is possible. (3) Our analysis is limited to Named Entity Recognition (NER). Further experiments are needed for other IE tasks.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No.62276110, No.62172039. The authors would also like to thank anonymous reviewers for their comments on improving the quality of this paper.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ting Wai Terence Au, Vasileios Lampos, and Ingemar Cox. 2022. E-ner—an annotated named entity recognition corpus of legal text. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 246–255.

Pei Chen, Haotian Xu, Cheng Zhang, and Ruihong Huang. 2022. Crossroads, buildings and neighborhoods: A dataset for fine-grained location recognition. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 3329–3339.

Nigel Collier, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 73–78.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179.

Leon Derczynski, Eric Nichols, Marieke Van Erp, and Nut Limsopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147.

Yuyang Ding, Juntao Li, Pinzheng Wang, Zecheng Tang, Yan Bowen, and Min Zhang. 2024a. Rethinking negative instances for generative named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3461–3475.

Zhuojun Ding, Wei Wei, Xiaoye Qu, and Dangyang Chen. 2024b. Improving pseudo labels with global-local denoising framework for cross-lingual named entity recognition. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 6252–6260.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Chenghao Fan, Zhenyi Lu, Sichen Liu, Xiaoye Qu, Wei Wei, Chengfeng Gu, and Yu Cheng. 2025. Make lora great again: Boosting lora with adaptive singular values and mixture-of-experts optimization alignment.

Chenghao Fan, Zhenyi Lu, Wei Wei, Jie Tian, Xiaoye Qu, Dangyang Chen, and Yu Cheng. 2024a. On giant's shoulders: Effortless weak to strong by dynamic logits fusion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Chenghao Fan, Wei Wei, Xiaoye Qu, Zhenyi Lu, Wenfeng Xie, Yu Cheng, and Dangyang Chen. 2024b. Enhancing low-resource relation representations through multi-view decoupling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38:17968–17976.

Runwei Guan, Ka Lok Man, Feifan Chen, Shanliang Yao, Rongsheng Hu, Xiaohui Zhu, Jeremy Smith, Eng Gee Lim, and Yutao Yue. 2024. Findvehicle and vehiclefinder: a ner dataset for natural language-based vehicle retrieval and a keyword-based cross-modal vehicle retrieval system. *Multimedia Tools and Applications*, 83(8):24841–24874.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Quanjiang Guo, Yihong Dong, Ling Tian, Zhao Kang, Yu Zhang, and Sijie Wang. 2025b. Baner: Boundary-aware llms for few-shot named entity recognition. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10375–10389.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7:1–17.

Aman Kumar and Binil Starly. 2022. "fabner": information extraction from manufacturing process science domain literature using named entity recognition. *Journal of Intelligent Manufacturing*, 33(8):2393–2407.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuan-Jing Huang, and Xipeng Qiu. 2023. Codeie: Large code generation models are better few-shot information extractors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Lixiang Lixiang, Zhilei Hu, Long Bai, Wei Li, Yidan Liu, Pan Yang, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2024. KnowCoder: Coding structured knowledge into llms for universal information extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8758–8779.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021. Crossner: Evaluating cross-domain named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13452–13460.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. 2024. Twin-merging: Dynamic integration of modular expertise in model merging. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Michael S Matena and Colin Raffel. 2022. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*.

Alexis Mitchell, Stephanie Strassel, Shudong Huang, and Ramez Zakhary. 2005. Ace 2004 multilingual training corpus. *LDC corpora*.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1946–1958.

Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. 2023. Guideline learning for in-context information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15372–15389.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Sampo Pyysalo and Sophia Ananiadou. 2014. Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6):868–875.

Yunjia Qi, Hao Peng, Xiaozhi Wang, Bin Xu, Lei Hou, and Juanzi Li. 2024. Adelie: Aligning large language models on information extraction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations*.

Larry Smith, Lorraine K Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. 2008. Overview of biocreative ii gene mention recognition. *Genome biology*, 9:1–19.

Zeliang Tong, Zhuojun Ding, and Wei Wei. 2025. Evoprompt: Evolving prompts for enhanced zero-shot named entity recognition with large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5136–5153.

Asahi Ushio, Francesco Barbieri, Vitor Sousa, Leonardo Neves, and Jose Camacho-Collados. 2022. Named entity recognition in twitter: A dataset and analysis on short-term temporal shifts. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 309–319.

Asahi Ushio and Jose Camacho-Collados. 2021. T-ner: An all-round python library for transformer-based named entity recognition. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 53–62.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *LDC corpora*.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3534–3547.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. 2023a. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. Crossweigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5154–5163.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023. Empirical study of zero-shot ner with chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7935–7956.

Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2024. Self-improving for zero-shot named entity recognition with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 583–593.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Annual Conference on Neural Information Processing Systems*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yuming Yang, Wantong Zhao, Caishuang Huang, Junjie Ye, Xiao Wang, Huiyuan Zheng, Yang Nan, Yuran Wang, Xueying Xu, Kaixin Huang, Yunke Zhang, Tao Gui, Qi Zhang, and Xuanjing Huang. 2025. Beyond boundaries: Learning a universal entity taxonomy across datasets and languages for open named entity recognition. In *Proceedings of the 31st International Conference on Computational Linguistics*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. Gliner: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024a. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024b. Universalner: Targeted distillation from large language models for open named entity recognition. In *The Twelfth International Conference on Learning Representations*.

A Experimental Details

Training Data We collected 17 widely used NER datasets and categorized them into six domains based on their data sources. Table 11 presents detailed statistics for each dataset. As described in Section 3.1, we sample these datasets based on the proportion of their entity types while ensuring that the total data volume per domain remains between 10,000 and 50,000. The distribution of the training data across domains is illustrated in Figure 6, and the exact number of sampled instances per dataset is listed in the "#Sampled" column of Table 11. Figure 7 and 8 show examples of the formatted training data we constructed.

Implementation Details We adopt the Qwen2.5-7B [3] and Llama3.1-8B [4] as foundations and train expert models using LoRA (Hu et al., 2022). The LoRA rank is set to 32, with three training epochs, a batch size of 16, a learning rate of $2e-5$, and a warmup ratio of 0.05. During LoRA training, all linear layers are activated. We set the temperature to 0 for LLMs when inference. All experiments are conducted on one NVIDIA 4090 GPU.

B Data Merging or Model Merging

We use **"Data Merging"** to denote the prior approaches of training a unified model by integrating data from multiple domains and **"Model Merging"** to denote synthesizing a new model by merging the parameters of expert models. This section presents a preliminary analysis that motivates the adoption of the model merging strategy. Specifically, we compare three types of models: (1) Experts, which are trained on single-domain data. (2) Data Merging, which is trained on a mixture of **all** domain data. (3) Model Merging, which is obtained by merging the parameters of **all** expert models.

We conduct evaluations on both in-domain and out-of-domain settings, with the results presented in Tables 12 and 13. Our key observations are as follows: **(1)** Data Merging consistently yields suboptimal performance in both settings. **(2)** In in-domain tasks, Model Merging also performs suboptimally and is inferior to Data Merging. **(3)** In out-of-domain tasks, Model Merging generally achieves the best performance.

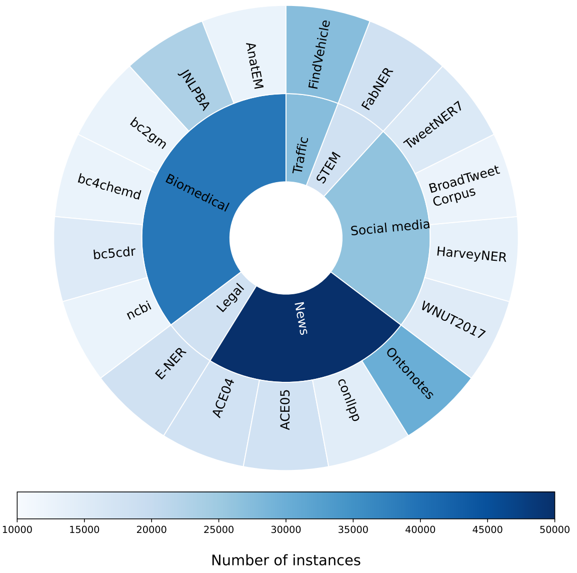


Figure 6: Distribution of the training data.

The aforementioned observation is our primary motivation for introducing model merging. For in-domain tasks, directly utilizing the corresponding expert model is sufficient. For out-of-domain tasks, merging expert models improves generalization. However, experimental results indicate that in some cases, a single expert model performs better, especially for in-domain tasks where merging may be unnecessary. Combined with Section 4.5, the number of selected experts, k, could be dynamically determined, with $k = 1$ being a valid consideration. This is a promising direction for further enhancing the adaptability of our framework.

C Analysis of Cost

Parameter Count and Storage Cost We assume H denotes the model dimension, r the rank of LoRA adapters, L the number of layers, n the number of domain-specific experts, and V the vocabulary size. For computational simplicity, we adopt a simplified Transformer architecture (e.g., omitting grouped-query attention mechanisms) in our base models (Llama3.1 and Qwen2.5). Since our LoRA implementation applies to all linear layers, each domain-specific expert requires $18HrL$ additional storage parameters. Consequently, storing n experts incurs a total overhead of $n \times 18HrL$. The base model itself requires approximately $(12H^2 + 13H)L + VH$ parameters. Given that $nr \ll H$, the additional storage overhead remains negligible. During inference, we merge LoRA adapters into one or two task-specific models, achieving state-of-the-art performance with only equivalent or doubled storage costs compared to the base model.

[3] https://huggingface.co/Qwen/Qwen2.5-7B

[4] https://huggingface.co/meta-llama/Llama-3.1-8B

Domain	Dataset	#Type	#Train	#Dev	#Test	#Sampled
Biomedical	AnatEM (Pyysalo and Ananiadou, 2014)	1	5,861	2,118	3,830	3,297
	JNLPBA (Collier et al., 2004)	5	16,691	1,855	3,856	16,487
	bc2gm (Smith et al., 2008)	1	12,500	2,500	5,000	3,297
	bc4chemd (Krallinger et al., 2015)	1	30,682	30,639	26,364	3,297
	bc5cdr (Li et al., 2016)	2	4,560	4,581	4,797	6,594
	ncbi (Doğan et al., 2014)	1	5,432	923	940	3,297
Law	E-NER (Au et al., 2022)	7	9,313	1,164	1,165	10,000
News	ACE04 (Mitchell et al., 2005)	7	6,202	745	812	9,722
	ACE05 (Walker et al., 2006)	7	7,299	971	1,060	9,722
	conllpp (Wang et al., 2019)	4	14,041	3,250	3,452	5,555
	OntoNotes (Pradhan et al., 2013)	18	59,924	8,528	8,262	25,000
Social media	WNUT2017 (Derczynski et al., 2017)	6	3,394	1,009	1,287	6,094
	HarveyNER (Chen et al., 2022)	4	3,967	1,301	1,303	4,063
	BroadTweetCorpus (Derczynski et al., 2016)	3	5,334	2,001	2,000	3,047
	TweetNER7 (Ushio et al., 2022)	7	7,111	886	576	7,110
STEM	FabNER (Kumar and Starly, 2022)	11	9,435	2,182	2,064	10,000
Traffic	FindVehicle (Guan et al., 2024)	8	21,565	20,777	20,777	21,565

Table 11: Statistics of raw training data and the number of sampled instances for training.

		Biomedical	Legal	News	Socia media	STEM	Traffic
Experts	Biomedical	**82.57**	40.02	36.61	48.09	27.27	21.05
	Legal	40.40	**84.79**	42.36	46.90	17.92	32.48
	News	53.83	48.40	**85.94**	48.28	23.53	41.84
	Social media	53.11	41.31	42.06	**66.57**	24.17	22.96
	STEM	28.22	16.26	22.81	23.53	**76.99**	24.96
	Traffic	42.14	28.23	32.67	42.01	20.23	**99.96**
Data Merging		80.53	81.87	84.97	60.27	77.40	98.91
Model Merging		67.57	64.13	55.47	56.25	31.80	45.29

Table 12: In-domain performance of expert models, full data trained model (Data Merging), and model obtained from merging all expert models (Model Merging).

		AI	Literature	Music	Politics	Science	Movie	Restaurant
Experts	Biomedical	56.90	59.01	63.98	**65.73**	60.21	57.96	39.41
	Legal	39.85	56.88	64.33	62.72	55.92	63.26	47.31
	News	41.77	41.03	54.74	40.36	49.66	62.13	39.28
	Social media	53.84	55.83	61.14	60.50	57.24	59.29	42.63
	STEM	41.01	41.35	43.35	45.41	37.38	46.61	26.78
	Traffic	49.97	50.82	62.20	64.09	53.25	65.45	52.11
Data Merging		51.38	53.46	61.12	54.99	59.39	66.66	**52.69**
Model Merging		**57.47**	**61.93**	**64.66**	60.28	**61.08**	**69.43**	47.78

Table 13: Our-of-domain performance of expert models, full data trained model (Data Merging), and model obtained from merging all expert models (Model Merging).

Computation FLOPs Analysis Compared to multi-task full fine-tuning (FFT), our approach utilizes the same amount of training data to produce multiple LoRA models, resulting in identical computational costs during training. During inference, however, our method employs a single task-specific

	Unified Models					SaM	SaM$_{eco}$
	InstructUIE	UniNER	GoLLIE	B2NER	Ours		
Training Instances	215.9K	45.9K	165.2K	51.9K	148.1K	148.1K	148.1K
Inference Times	1×	1×	1×	1×	1×	2×	1×
Storage (Normalized)	1	1	1	1	1	1+0.02n	1+0.02n
Performance (Average)	47.84	61.79	58.39	64.57	60.94	66.24	65.49

Table 14: Comparison of Resource Requirements. We compare unified models trained across multiple domains with our merging-based approach. SaM$_{eco}$ (economic) refers to integrating two task-specific models into a single one (details in Section 4.4). For storage, we normalize the value by setting the model size to 1. Here, n represents the number of experts. We achieve superior results with minimal additional overhead, particularly with our SaM$_{eco}$.

	Llama				Qwen			
	Mode1	**Mode2**	**Mode3**	**SaM(Ours)**	**Mode1**	**Mode2**	**Mode3**	**SaM(Ours)**
AI	60.36	58.19	**61.31**	60.01	54.73	55.62	57.90	**60.98**
Literature	56.86	**62.68**	58.90	61.99	60.16	59.87	63.42	**66.93**
Music	63.78	66.50	**66.85**	65.93	68.98	72.89	72.42	**73.53**
Politics	66.05	**68.27**	61.30	67.05	71.16	**75.42**	73.61	74.47
Science	58.37	61.73	**63.12**	62.41	62.19	59.58	**62.78**	62.60
Movie	70.66	70.66	70.50	**71.65**	67.07	68.91	**73.44**	72.17
Restaurant	52.82	52.85	52.23	**52.90**	51.69	51.69	**54.84**	52.99
Average	61.27	62.98	62.03	**63.13**	62.28	63.43	65.49	**66.24**

Table 15: Merging into a single task model. Complete experimental results of Section 4.4.

model—either derived from a single strategy or by integrating models from both strategies—which requires only one LoRA adapter. This achieves comparable inference costs to multi-task FFT while delivering superior performance. When leveraging models from both strategies simultaneously, our approach incurs twice the inference cost but further enhances performance, offering a flexible trade-off between efficiency and effectiveness. Table 14 compares our approach with several recent works that train a unified model across multiple domains. We compare (1) the amount of training data (with instance-level data size provided for reference), (2) the number of inference rounds required for model prediction, and (3) storage space requirements (with model size as the reference unit). We also report the average performance.

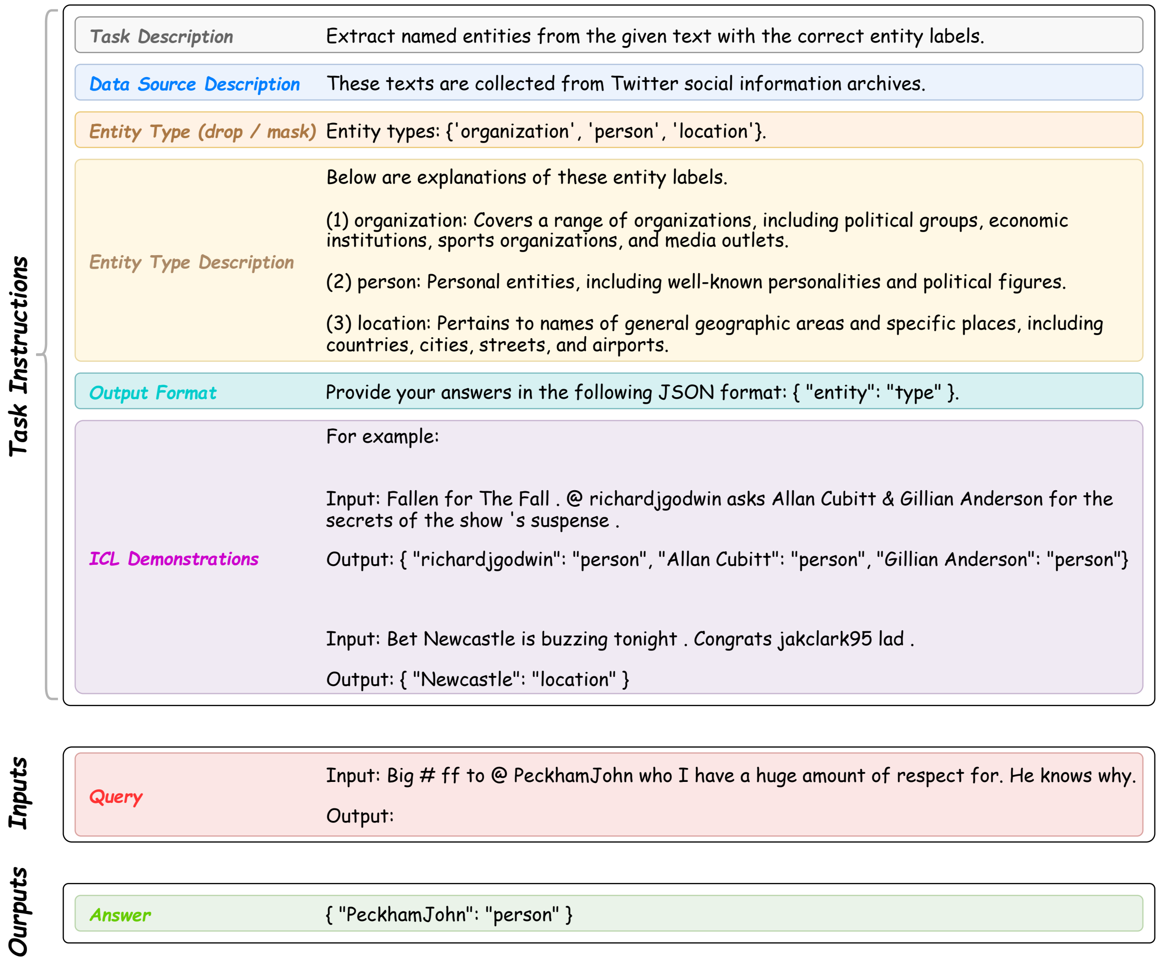
D Experimental Supplements

Section 4.4 extracts a single expert set for merging and presents the results based on Qwen. Here, we supplement the results of Llama in Table 15.

Our framework is based on two perspectives: Domain Similarity (DS) and Sampling Evaluation (SE). The experimental section reports the overall performance of the framework. Here, we provide several additional experimental results regarding these two strategies, respectively.

Section 4.5 explores the relationship between model performance and the number of merged experts. Here, we present results for each strategy, as shown in Figures 9 and 10. It can be seen that the two strategies exhibit similar overall trends, but with distinct differences. This further indicates that both strategies are important, highlighting the importance and complementarity of each.

Section 4.9 discusses fine-grained adaptation. Due to the compact nature of the target domain, this does not bring improvements and may even reduce the performance. Here, we analyze the performance of each strategy individually. As shown in Figure 11, the Domain Similarity strategy has a minimal impact on the subdivision (with a performance difference of less than 0.2 points on average), supporting the hypothesis that the target domain is already too homogeneous to generate distinct splits. In contrast, Figure 12 shows significant changes when using Sampling Evaluation, with a general downward trend, which accounts for the overall performance degradation.



Task Instructions

Task Description
Extract named entities from the given text with the correct entity labels.

Data Source Description
These texts are collected from Twitter social information archives.

Entity Type (drop / mask)
Entity types: {'organization', 'person', 'location'}.

Entity Type Description
Below are explanations of these entity labels.

(1) organization: Covers a range of organizations, including political groups, economic institutions, sports organizations, and media outlets.

(2) person: Personal entities, including well-known personalities and political figures.

(3) location: Pertains to names of general geographic areas and specific places, including countries, cities, streets, and airports.

Output Format
Provide your answers in the following JSON format: { "entity": "type" }.

ICL Demonstrations
For example:

Input: Fallen for The Fall . @ richardjgodwin asks Allan Cubitt & Gillian Anderson for the secrets of the show 's suspense .

Output: { "richardjgodwin": "person", "Allan Cubitt": "person", "Gillian Anderson": "person"}

Input: Bet Newcastle is buzzing tonight . Congrats jakclark95 lad .

Output: { "Newcastle": "location" }

Inputs

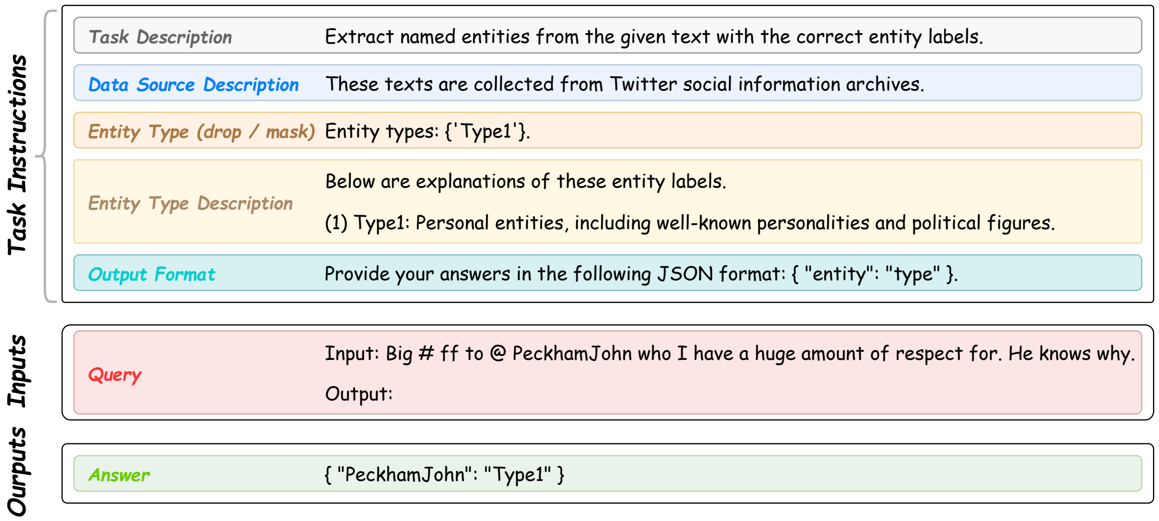
Query
Input: Big # ff to @ PeckhamJohn who I have a huge amount of respect for. He knows why.

Output:

Ourputs

Answer
{ "PeckhamJohn": "person" }

Figure 7: Formatted training data example. The example consists of task instructions, inputs, and outputs for training. For the task instructions, the Data Source Description, Entity Type drop/mask, and ICL Demonstrations are optional, with details in section 3.1. We adopt three output formats: JSON, enumeration, and natural language descriptions. The output format of ICL Demonstrations and Answers should be consistent with the specified.



Task Instructions

Task Description
Extract named entities from the given text with the correct entity labels.

Data Source Description
These texts are collected from Twitter social information archives.

Entity Type (drop / mask)
Entity types: {'Type1'}.

Entity Type Description
Below are explanations of these entity labels.

(1) Type1: Personal entities, including well-known personalities and political figures.

Output Format
Provide your answers in the following JSON format: { "entity": "type" }.

Inputs

Query
Input: Big # ff to @ PeckhamJohn who I have a huge amount of respect for. He knows why.

Output:

Ourputs

Answer
{ "PeckhamJohn": "Type1" }

Figure 8: Another example of our formatted training data. The instance here is the same as that of Figure 7, adopting the entity type drop and mask processing methods.

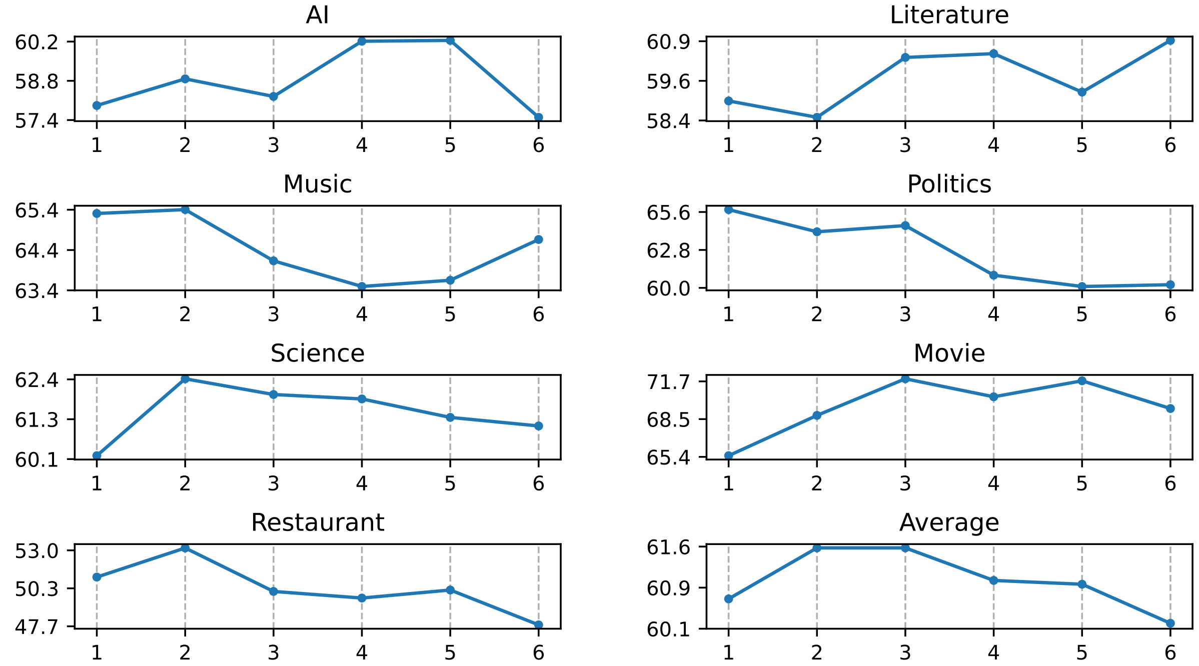


Figure 9: Performance changes with the number of expert models (denotes as k). The horizontal axis is the number of experts for merging, and the vertical denotes the entity-level F1 scores. **Only using Domain Similarity for expert selection**. It can be seen that the optimal k varies across target domains, typically ranging from 2 to 4.

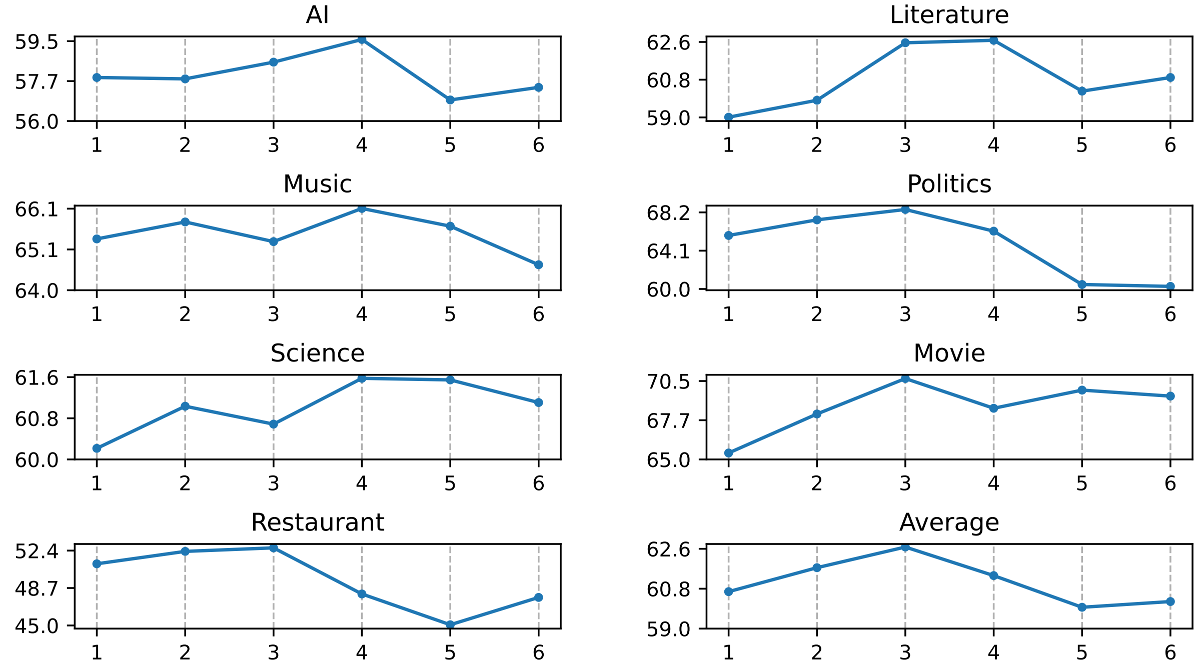


Figure 10: Performance changes with the number of expert models. The horizontal axis is the number of experts for merging, and the vertical denotes the entity-level F1 scores. **Only using Sampling Evaluation for expert selection.** It exhibits similar overall trends to Figure 9, but with distinct differences, indicating that both selecting strategies are important and complementary.

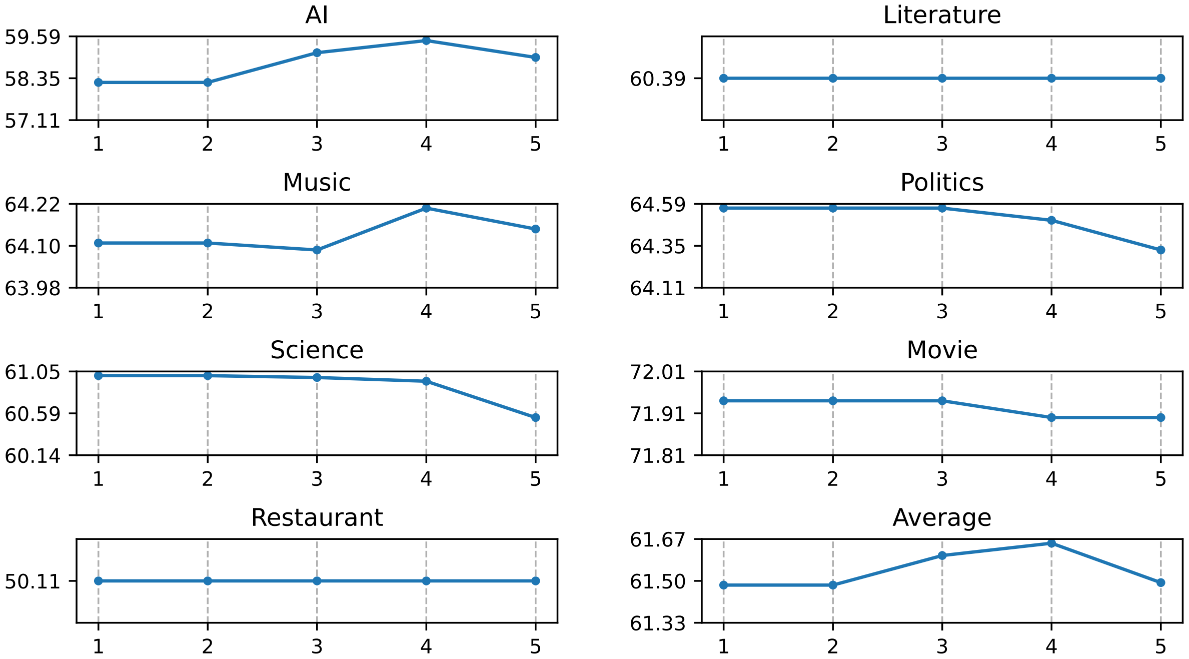


Figure 11: Performance changes with the number of data splits. The horizontal axis is the number of splits, and the vertical denotes the entity-level F1 scores. **Only using Domain Similarity for expert selection.** It can be seen that the Domain Similarity strategy has a minimal impact, since data from the target domain may be too homogeneous.

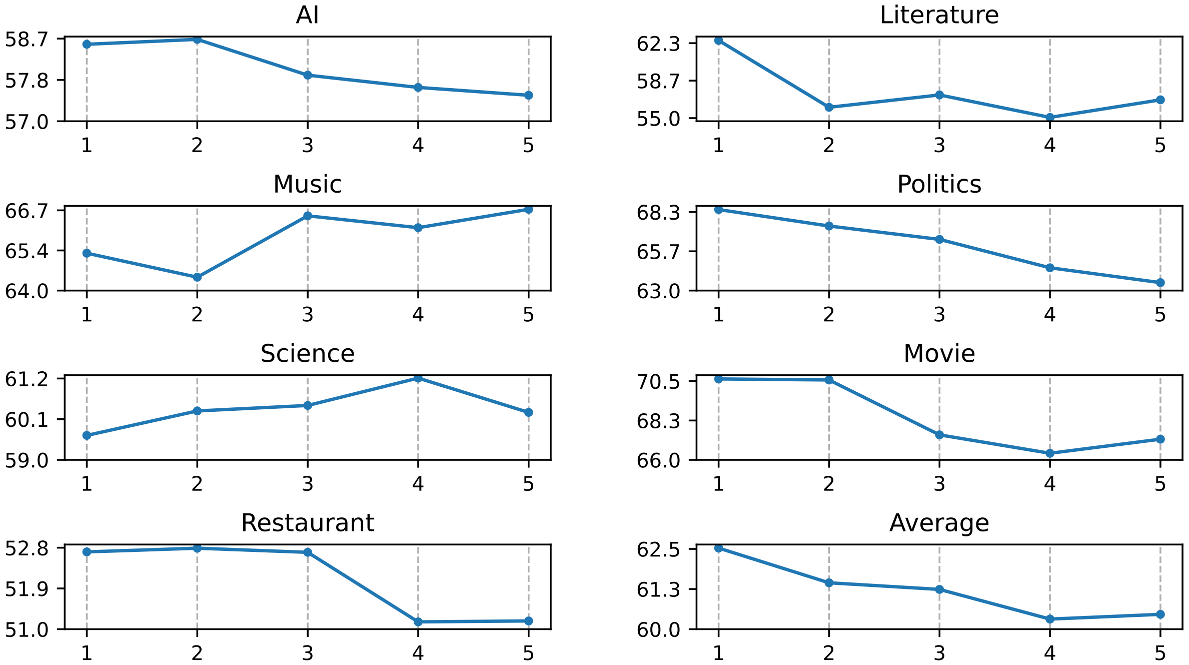


Figure 12: Performance changes with the number of data splits. The horizontal axis is the number of splits, and the vertical denotes the entity-level F1 scores. **Only using Sampling Evaluation for expert selection.** It shows significant changes when using Sampling Evaluation, with a general downward trend.