

PELMo: Prompt-based Ensemble Expert Language Models with Multi-label Routing

Jiansheng Wang¹, Jian Zhang^{*2}, Yuzhi Mu², Wei Han², Xuefan Xu², Junyu Shen¹, Tianming Ma¹

¹Shanghai Normal University, Shanghai China

²DataGrand Inc, Shanghai China

zhangjian@datagrand.com

Abstract. The large language models (LLMs) utilize few-shot and zero-shot prompting to support tasks across multiple domains better. Despite LLM's strong performance on a wide range of natural language tasks, a single LLM is often difficult to generalize to multiple domains that require different knowledge and abilities. To overcome this problem, we introduce PELMo, an ensemble framework designed to attain consistently superior performance by leveraging the diverse strengths of multiple language expert models. Our method combines multiple expert models by training an additional routing model. First, by optimizing prompts with instruction for different tasks, we obtain expert models with different task capabilities based on the same backbone. Afterwards, a multi-label routing model is trained to select k top-ranked expert models for each question strategically. Finally, the outputs of the selected expert models at the final layer are through weighted averaging to generate the ultimate answer. Our results demonstrate that PELMo outperforms the expert models within the target domain and achieves robust capabilities in the whole scope of tasks. Overall, these results demonstrate the benefits of ensembling k top-ranked expert models during language modeling.

Keywords: large language models, expert models, prompt, ensemble.

1 Introduction

With the rapid development of large language models, artificial intelligence has made significant progress. Through paradigms such as pre-training, supervised fine-tuning, and RLHF, LLMs have acquired powerful text generation capabilities. Currently, the performance of LLMs is very close to human-level abilities, and they have been used as key building blocks in numerous applications such as information retrieval, dialogue systems, and autonomous AI agents, yielding promising results. However, when faced with specific downstream tasks, the knowledge and skills learned during pre-training by large models are often insufficient to cover the diverse knowledge and skills required across all task domains. Therefore, in these applications, LLMs often adapt quickly to specific downstream tasks through prompting learning methods such as few-shot, enabling these systems to intelligently handle user needs and questions more effectively.

Although performance has been enhanced in specific applications, in real-world deployment scenarios, downstream tasks may involve multiple domains or topics, and the

* Corresponding author.

advantage of a single LLM in a specific domain may not fully translate into efficient processing capabilities for other domains. This leads to unsatisfactory performance of single language model in tasks within specific domains. To address this issue, researchers have begun to utilize Mixture-of-Experts (MoE) models to integrate the capabilities of experts from different domains to enhance the model's overall performance. This method primarily consists of a sparse gate-controlled deep learning model composed of expert models and gating models. MoE allocates tasks or training data among different expert models through gate networks, allowing each model to focus on handling its most proficient tasks, thereby improving the model's adaptability and performance in multi-domain tasks. Existing MoE methods typically route different tokens to expert parameters [1-3]. However, this method has not been widely adopted due to the high communication cost of routing each token in each sparse layer [4], the difficulty of enabling expert models to specialize in handling specific tokens [5], and the necessity of additional mechanisms to ensure load balancing among expert models [6].

There is also a merging strategy that makes each expert model generate responses independently. Approaches such as [7-9] use a given input to obtain the responses of multiple language models and the confidence scores of each output. Finally, the best model response is obtained through these confidence scores or by fusing multiple candidate outputs. These model merging methods can compensate for the limitations of individual models and improve the overall accuracy and robustness of predictions. However, these methods require each expert to perform inference independently, leading to significant computational costs and time consumption.

In this work, our goal is to design an effective framework to integrate expertise from different domains into a comprehensive and widely applicable mixture-of-experts model. At the same time, we aim for the new framework to be simpler, more practical, and have lower computational costs.

With these considerations, we propose prompt-based ensemble expert language models with multi-label routing (PELMo) to preserve each expert model's strengths while reducing computational costs and communication overhead. Specifically, we use prompt learning to implement domain expert models for each specific task. Then, we train a routing model based on the task domains to automatically assign expert models. During inference, we employ an offline routing approach instead of online load balancing to assign the given input to the top-k-ranked expert models, thus reducing computational costs by sparsely activating a subset of expert models through the routing model. Finally, we generate answers by weighted averaging of logits from the selected expert models.

We leverage the latest prompt learning methods to obtain expert models for different task domains. Prompt learning can effectively stimulate the potential of LLM by providing them with additional information and improving the knowledge and skills required for models to learn specific tasks. Some current prompt learning methods, such as the Thought of Chain [10] method and the retrieval-augmented prompting method [11], design appropriate prompts to enable the LLMs to follow the user's intent and generate appropriate responses.

Based on the theory of prompt learning, we design specific prompts for specific domains, enabling the LLM to become a domain expert on these datasets. According to

our evaluations of four representative benchmarks, our ensemble model performs strongly in general domain tasks, outperforming all baselines in target domains while maintaining strong performance and mostly surpassing single expert models. Overall, the experimental results show that the PELMo is a simple and effective method that can make contributions to decoder-based large language models in language generation tasks.

2 Methodology

Problem Statement. Assume there are N expert models E_1, E_2, \dots, E_N in different domains. We aim to combine these expert models to obtain an ensemble model that achieves strong performance across all known domains.

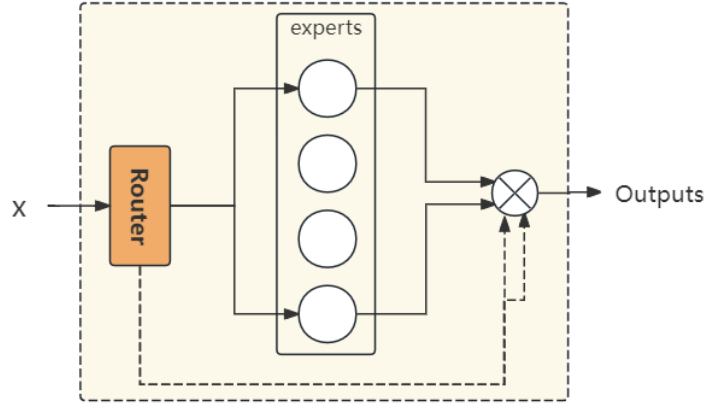


Fig. 1. Illustration of PELMo framework. A routing model routes each input x to one of the k experts among the four experts. The dashed lines in the figure represent the weights W_E . The output is the weighted sum of the probability distributions of the outputs selected by the k -chosen experts.

2.1 Expert Models

The primary step in our model is to acquire a diverse set of expert models tailored to the specific task domain. Based on this ensemble of expert models and utilizing dynamic routing model, we can effectively select the appropriate experts to fully leverage their strengths when faced with any specific question. Currently, there are many ways to build expert models, such as fine-tuning the model for specific tasks or incremental learning. However, these methods require considerable computational resources. Therefore, we design expert models by prompt learning to elicit the hidden knowledge and potential of the models. Prompt learning reduces the time and cost of model training compared to fine-tuning techniques. Especially in vertical domains lacking training data, prompt learning demonstrates outstanding performance. Furthermore, prompt

strategies can help understand how the model generates outputs, thereby improving the model's interpretability.

Based on the method proposed by [8] to construct different prompts for different reasoning task types to improve the performance of the model, we design special prompt according to the question characteristics of specific task domains to make the base model professional. Specifically, we use the collected QA datasets and design corresponding prompt by analyzing data characteristics to obtain a domain expert model. The specific steps are:

- **Natural Questions:** Natural question dataset [12]. We use the method of retrieval-augmented prompting [11]. For each test question, use Contriever [13] to retrieve the ten most relevant paragraphs from Wikipedia and combine them appended to the prompt.
- **HotpotQA:** Multi-hop reasoning question-and-answer dataset [14]. This dataset includes many multi-hop reasoning questions. Therefore, we use the chain of thought (CoT) prompt [10] method to design corresponding prompts for this task. We added manually written rationales after each demonstration question to elicit the multi-hop reasoning process for the test questions.
- **GSM8K:** Primary school mathematics question dataset [15]. The mathematics questions contained in this dataset require multi-hop reasoning to obtain answers. Therefore, we use the chain of thought (CoT) [10] method to design a prompt to make in the process of the model outputting the answer, the step-by-step reasoning method is used to make the calculated answer more accurate. Specifically, we add accompanying explanations provided in GSM8K after each demonstration question in the prompt.
- **CommonsenseQA(CSQA):** Commonsense question-and-answer dataset[16]. This dataset includes various commonsense questions and is used to test the knowledge of the model. We use the answer prompting [17] to use the same base LLM for each question to generate ten answers for each question, and append these answers to the prompt as additional knowledge.

In addition, we use 16 randomly sampled training examples as demonstration examples for each expert. Specifically, we use examples in NQ as demonstration examples for NQ domain experts, examples in HotpotQA as demonstration examples for HotpotQA domain experts, examples in GSM8K as examples for GSM8K domain experts, and examples in CSQA as CSQA Examples of domain experts. These demonstration examples are formatted using the corresponding dedicated, prompt strategies described above.

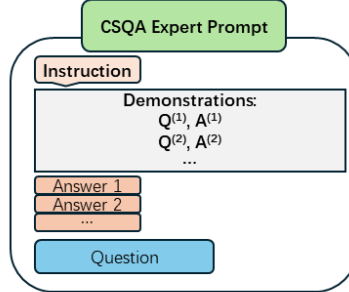


Fig. 2. The illustration of CSQA expert prompt. Other expert prompts are similar.

2.2 Routing Model

Regarding the expert selection problem in the MoE, most previous researchers have focused on the token routing level. For example, [2] learns the weight function g at the token level and assigns top-k experts to each token [1]. However, token-level routing models incur high communication costs for routing each token and require mechanisms to ensure load balancing of expert models. Therefore, our routing model performs expert selection at the sequence level; that is, it will route the input question to the top-k experts who are most likely to answer the question correctly.

Expert selection routing model. We trained our routing model based on the Roberta-base model [18] using examples from the same origin as the test set. The routing model in PELMo is similar to the gate structure in [2]. Based on the collected data, we construct data on whether each expert model can answer correctly and use a multi-label classification method to train the routing model. We trained a multi-label classifier as our routing model, using the correctness of answers from all expert models as labels (see **Fig. 3**).

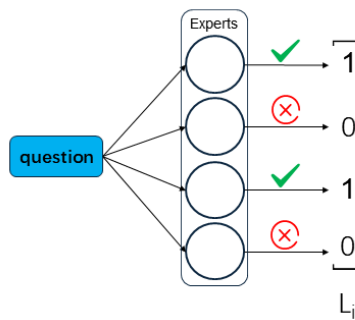


Fig. 3. The diagram illustrates how to obtain training labels for the routing model. L_i is the label vector, $|L_i| = N$. For each sample processed by each expert model, we record the correctness of the response. If an expert model correctly answers a question associated with a particular sample, the corresponding position in the label vector for that sample is set to 1; otherwise, it is set to 0. Through this approach, we construct a label matrix $L_{M \times N}$.

Training. During the actual training, we found that there were many cases where the multiple labels were 0; that is, there were cases where no expert could answer the question correctly. If this result is directly used for expert selection, the performance of the model will be greatly reduced. To solve this problem, we introduce an all-zero label handling strategy. We embed N answers provided by expert models and gold answer provided by expert models into vectors. The answers vectors a_0, \dots, a_N , and the gold answer vector a_g are used to calculate the similarity between them:

$$s_n = \frac{a_n \cdot a_g}{|a_n| \cdot |a_g|} \quad (1)$$

where s_n denote the cosine similarity between answer a_n and the gold answer a_g . The threshold for similarity is denoted by θ .¹ If there exists a_n such that $s_n > \theta$, the corresponding answer a_n is considered correct. In the event where all answers have similarity measures less than the threshold θ , the answer a_n with the maximum similarity measure is chosen as the correct answer.

The routing model trained through the above method can select the experts most capable of handling the input question. And this approach makes the multi-label classification model more robust and generalizable by introducing additional information (similarity calculation and threshold strategy) and can effectively deal with the situation of missing labels, improving the effect and applicability of the routing model.

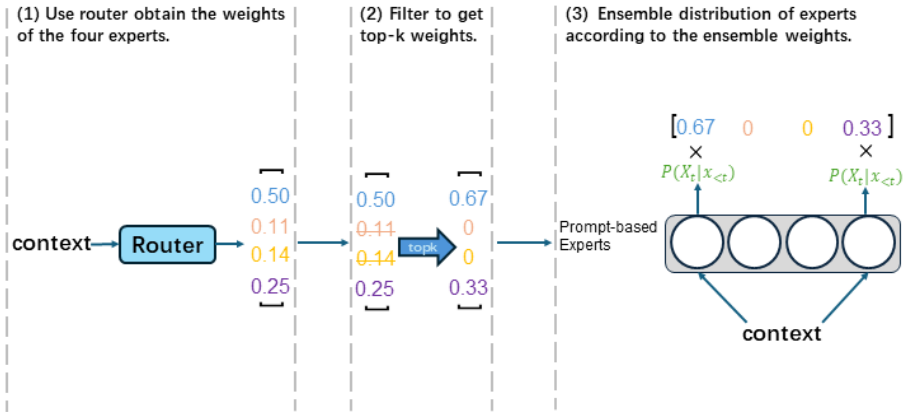


Fig. 4. PELMo inference process. At inference time, we first input each incoming context into the routing model to obtain the weights for each expert model. We then use the weights of the top-k experts to weight the outputs of the expert models.

2.3 Ensemble of Expert Models

Based on the expert models obtained above, we use the expert selection routing model to select the top-k expert models that are most capable of answering the current question

¹ According to different embedding models, the threshold for similarity θ takes different values. The threshold θ generally takes a relatively high value.

(see **Fig. 4**). Given W_E ensemble weights with representations $\{w_1, w_2, \dots, w_N\}$. W_E has one non-negative for each expert, most of which are zeros meaning the question is not dispatched to that expert. Specifically, during the inference, for each test question, we use the routing model to obtain the ensemble weights of each expert:

$$\text{logits} = \text{router}(x_s) \quad (2)$$

$$W_E = \text{topk}(\text{softmax}(\text{logits})) \quad (3)$$

Where *router* is the routing model, x_s is the incoming test contexts, and *logits* is the output of the final layer of the routing model. The *topk* function filters for the top-k probabilities and renormalizes their distributions so that they sum to 1. The routing model is trained according to the method developed in Section 2.2. Consider the probabilistic view of language modeling, where we estimate $P(X_t|x_{<t})$. Then the next-step conditional distribution of the ensemble model on the history $x_{<t}$ is:

$$P(X_t|x_{<t}) = \sum_{i=0}^N w_i \cdot P(X_t|x_{<t}, E_i) \quad (4)$$

The ensemble weight w_i is not updated at each time step of the current incoming token. Compared with the integration weight in [19], which is updated for each incoming token, our approach can reduce a certain amount of model calculation burden and improve reasoning efficiency. Furthermore, at inference time, since our expert model is obtained through prompt learning, only a single model parameter needs to be loaded, which effectively reduces the computational and communication costs.

3 Experimental Setup

This section describes the experimental setup, including the evaluation method, evaluation dataset, and the selection and design of the baselines. These fundamental elements are essential to ensure the scientific validity of the experiment and the trustworthiness of the results. We first introduce datasets used for evaluation and its characteristics, as well as our choice of the base model. Subsequently, we discuss the evaluation methods (evaluation metrics) we adopted. Finally, we introduce our compared models and provide details of their implementation.

3.1 Evaluation Dataset and Base Model

Our evaluation dataset also use the four QA datasets introduced in Section 2.1. These datasets cover different formats, domains, and reasoning skills. At the same time, we also extracted a smaller sample from the training set of each dataset as the training set of our routing model². we choose Llama-2-7B-chat [20] as the base model, which is superior to most open-source conversation models in terms of usefulness and robustness.

² We also tried to use more training sets to train our routing model, but no performance was observed to improve.

3.2 Metrics

Due to the use of multiple prompt learning methods, the answer format of the ensemble model is uncertain. Therefore, it is difficult to use keyword matching or simple pattern recognition to extract answers. To evaluate the performance, we adopt an evaluation strategy based on GPT-3.5, which is similar to the method of using GPT-3.5 for judgment in [21]. Except for the dataset GSM8K³, we use this strategy for other datasets to determine whether our model answers are correct. GPT-3.5 is a highly intelligent language model that is trained on large-scale text data through deep learning technology and can simulate human language understanding and generation capabilities. Traditional evaluation methods usually rely on keyword matching or simple pattern recognition to extract answers, while GPT can deeply understand the context of the answer and the meaning of the answer so that it can more accurately evaluate the relevance and accuracy of the answer.

3.3 Compared Models

We compare PELMo with several other baselines:

- **Specific few-shot:** 16 question-answer pairs were randomly sampled from the training set of each dataset and spliced with test questions as prompts for evaluation without using any specialized prompt learning method.
- **Single-label router:** The single-label routing model compares and analyzes the effectiveness of routing models. During training, the label used is the domain where the question is located. Let D denote the set of domains. If we index the experts by D and $d \in D$ is the domain label for the current training instance, then

$$L_{ij} = \begin{cases} 1 & \text{if } j = d \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where L_{ij} represents the j -th element of the label vector. The labels obtained through the above method are used to train a single-label classification model as a routing model.

- **Oracle Ensemble:** Calculate the upper bound of the model by taking the optimal answer for each question. If an expert model can output the correct answer for each question, the accuracy is 1.0.

4 Experimental Results

To objectively evaluate the effect of PELMo, we test the proposed ensemble model and baselines on four datasets and comparatively analyze the effectiveness of the proposed model. First, we will evaluate the performance of different expert models on four datasets, highlighting the significant improvements of each expert in the corresponding

³ GSM8K dataset can be used directly to extract numerical answers using pattern recognition.

tasks compared to the baseline. We then dive into the inadequacies of expert models' performance on data types outside their domain of expertise, highlighting the limitations of a single expert model. Finally, we analyzed the performance of PELMo on related tasks and the impact of the number of experts on the PELMo model.

4.1 Expert Models Performance

Four expert models were evaluated on four datasets, and through the experimental results (see **Table 1**), it can be observed that the expert models significantly improved on the corresponding tasks compared to the specific few-shot baseline. For example, the NQ expert model outperforms the specific few-shot baseline accuracy on NQ from 30.5% to 38.5%, and the GSM8K expert model improves accuracy on GSM8K from 20.0% to 25.58%. The only exception is that the best performing model on the dataset HotpotQA is the NQ model expert, the same result obtained by [8]. This is because HotpotQA is essentially a knowledge-intensive dataset, and retrieval-augmented can improve the effect of the model more than the thought of chain.

However, expert models mostly perform worse than specific few-shot on out of domain. For example, the NQ expert model performs worse than the specific few-shot baseline on GSM8K and CSQA datasets. Similarly, the GSM8K expert model performs worse than the baseline on all other datasets. This implies that a single expert model is unable to handle questions across various domains, which motivates us to propose our ensemble approach to combine the areas of expertise of different experts to perform well on questions in any domain.

Table 1. Per-dataset accuracy obtained through the evaluation method introduced in Section 3.2. The best results on each dataset are highlighted in the table. Expert models perform well on corresponding types of reasoning but lose generalization capabilities in other areas. In our ensemble model, the top-2 usually performs better than the top-1. PELMo top-2 achieves the best macro average on all datasets, and its performance exceeds that of all expert models.

	NQ	HOTPOTQA	GSM8K	CSQA	Macro-Average
Specific Few-shot	30.50	17.00	20.00	19.08	21.65
NQ Expert	38.50	26.25	11.25	5.60	20.40
HotpotQA Expert	31.00	19.25	23.25	13.74	21.81
GSM8K Expert	25.50	10.25	25.00	15.01	18.94
CSQA Expert	19.50	13.00	20.50	19.60	18.15
PELMo top-1	38.13	27.91	22.08	23.79	27.98
PELMo top-2	41.88	28.00	25.41	24.31	29.90
Oracle	55.50	41.00	44.75	30.79	43.01

4.2 Ensemble Models Performance

We compared and validated the routing selection mechanisms in our ensemble method in **Table 2** and analyzed the impact of the number of experts on the model's effectiveness. Among them, the core method we proposed is the multi-label router, which selects the top-2 expert models using a multi-label routing model (PELMo top-2).

Multi-Label Router

The performance of PELMo top-1 on both HOTPOTQA and CSQA datasets surpassed the expert model in this domain, improving by 1.66% and 4.19%, respectively (see **Table 1**). At the same time, PELMo top-1 ensures the performance level of other unrelated tasks. PELMo top-2 achieves the best performance on four datasets by ensembling top-2 expert models in different domains, greatly improving the overall capabilities of the model, surpassing expert models in all domains, and showing outstanding performance. Compared with specific few-shot, the performance of PELMo top-2 on NQ, HOTPOTQA, GSM8K, and CSQA datasets has improved by 11.38%, 11%, 5.41%, and 5.23%, respectively. Although PELMo achieves the best macro average on all datasets, it is still lower than oracle ensemble, but this also shows that the model ensemble method still has much room for improvement.

Single-Label Router

To verify and compare the effectiveness of our multi-label router, we trained a domain-based single-label routing model. The experimental results are shown in **Table 2**.

Table 2. Performance comparison of single-label routing model and multi-label routing model on four datasets.

	NQ	HOTPOTQA	GSM8K	CSQA	Macro-Average
Multi-label Router; top-1	38.13	27.91	22.08	23.79	27.98
Multi-label Router; top-2	41.88	28.00	25.41	24.31	29.90
Single-label Router; top-1	32.00	13.00	24.25	28.24	24.37
Single-label Router; top-2	39.25	15.75	23.25	29.01	26.82

Experimental results show that the performance of single-label routing models generally lags behind multi-label routing models on most datasets. This may be due to the relatively obvious data characteristics of each dataset. Multi-label routing models generally perform better on these datasets because they can better capture multiple features in the data and consider the correlation between them into account when routing model predictions. However, it is worth noting that the single-label routing model may still have unique advantages in some specific situations, especially when the data characteristics are relatively simple and consistent.

These results show that our ensemble method can effectively improve the versatility and effectiveness of the model by integrating existing expert language models.

Effects of Expert Quantity

We tested whether the choice of the number of experts in PELMo affects the final effect of the model and whether the experts in each domain could solve problems in the domain. Finally, we quantitatively tested the impact of the number of experts on the performance of our ensemble method.

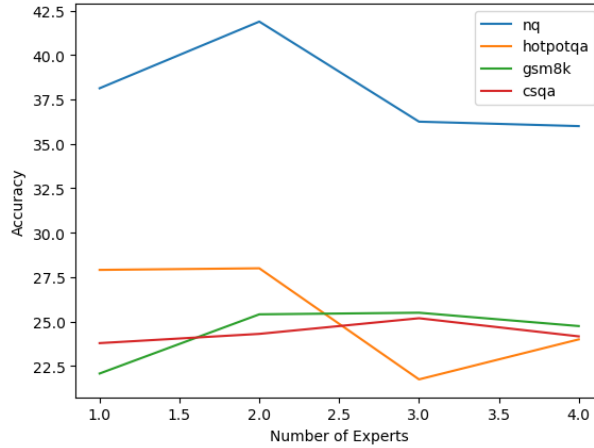


Fig. 5. This line chart shows how the accuracy of the ensemble model on the four datasets changes as the number of experts increases. It can be observed from the figure that when the number of experts is 2, the accuracy is the highest.

The results of **Fig. 5** show that PELMo has the best performance in all datasets when the number of expert models is 2. And as the number of experts continues to increase, the accuracy decreases or stabilizes at 3 and 4 experts. These results suggest that having a greater number of experts in our system does not always positively impact system performance; instead, an increased number of experts not only incurs additional computational resource consumption but may also adversely affect our methodology.

5 Related Work

Mixture-of-Experts. Ensemble learning is a popular technique for improving the capabilities of deep learning models by leveraging multiple weaker models. Among them, a classic algorithm is Mixture-of-Experts (MoE) [22], which proposes to use multiple expert models to fit a subset of multi-task data and then use a gating network to determine the distribution of data to reduce the interference of data from different tasks on the learning process, and improves the learning efficiency and generalization performance of the model. Subsequently, such as Sparsely-Gated MoE [23], GShard [1], Switch-Transformer [2], BASE Layer [6] and DEMIX Layer [24] and other methods have proposed constructive improvements to the MoE model architecture or training methods, and accelerated the development and application of MoE. Currently, the latest work is a MoE model named Mixtral [25], which utilizes a sparse representation-based gating mechanism to select expert models and achieves excellent performance on multiple NLP tasks. However, the above-mentioned MoE methods all require joint training of multiple expert models, which will bring high computational costs and basically rely on token-based routing mechanisms. For the routing selection mechanism of the expert models, our approach performs routing at the sequence level and generates predictions

for the entire question. There are no shared parameters between experts, which effectively eliminates all complexities associated with balancing expert utilization.

Merging strategies [7-9] aim to derive the responses of multiple language models and the confidence of each output. Ultimately, the best output is obtained through these confidences or by fusing multiple candidate responses. The key to this combined approach is to allow each expert model to leverage its unique strengths to provide diverse perspectives and solutions to the problem. However, the disadvantage of this ensemble approach is also apparent, as each expert requires independent inference, resulting in significant computational costs and time consumption. The experts of our ensemble model are sparsely activated. For any input context, only the top-2 expert models need to be selected for inference, which significantly reduces the calculation and communication costs of the model.

Another related direction is model merging. This method aims to merge multiple task-specific models into a single model with diverse capabilities [26-31]. The advantage of model merging over multi-task learning [9, 66] is that model parameter merging usually does not need to pay attention to the original training data without retraining but only needs to pay attention to the combination of model parameters [27, 29], and even this method does not significantly increase the computational cost. However, the performance of the model merging makes it difficult to surpass a single expert model in a specific task.

Router. As a hub component in the MoE model architecture, the router plays a vital role in the final performance of the model. Ensuring the load balancing of expert models and achieving proper allocation of inputs to experts through routers have always been focal points of attention in MoE models. [32, 33] employed hierarchical clustering to identify domains of expertise in specialized research domains. C-BTM [19] obtained expert models by clustering the training data and separately constructing domain language models for each cluster. Subsequently, input vectors were vectorized to compute distances from cluster centroids to select the top-2 experts. The above method is based on the cluster router. Cluster-based router methods mainly rely on data similarity. In contrast, our router uses multi-labels during the training, which is more consistent with the selection of multiple experts during the inference. This method can learn more features and patterns, thereby better handling the complex relationship between the input and the expert models. Our routing approach is most directly related to [8], which scores expert model answers by training a random forest routing model. Our routing model is inspired by the above methods, which provides a strong foundation for our research.

6 Conclusion

In this work, we propose the PELMo framework, an ensemble method to improve performance on target tasks without decreasing accuracy on other unrelated tasks. Experimental results demonstrate that by selecting top-2 expert models from different tasks, PELMo top-2 consistently outperforms baselines and individual expert models in most

cases, significantly enhancing the model's versatility and overall performance. This validates the effectiveness of PELMo and provides a feasible approach for facing cross-domain challenges.

In summary, PELMo provides empirical support for tackling multi-domain problems. However, there are still promising directions for future research. Firstly, we can consider introducing more domain experts, which can be heterogeneous or obtained through fine-tuning, to enrich further the knowledge base and capabilities of the ensemble method. By introducing more experts, we hope to extend the professional knowledge of the model to out-of-domains, thereby improving the model's performance in novel domains. Secondly, the routing model is also a potential optimization direction. The experimental process found that the routing model plays a key role in the ensemble model, and the final effect of the ensemble model is greatly affected by the routing model. Therefore, future research can focus on further optimizing the design of the routing model, and through careful design of the structure of the routing model, it is expected to improve the performance of the ensemble method further.

7 References

1. Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., Chen, Z.: Gshard: Scaling giant models with conditional computation and automatic sharding. arXiv preprint arXiv:2006.16668 (2020)
2. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research* 23, 1-39 (2022)
3. Clark, A., de Las Casas, D., Guy, A., Mensch, A., Paganini, M., Hoffmann, J., Damoc, B., Hechtman, B., Cai, T., Borgeaud, S.: Unified scaling laws for routed language models. In: *International conference on machine learning*, pp. 4057-4086. PMLR, (Year)
4. Artetxe, M., Bhosale, S., Goyal, N., Mihaylov, T., Ott, M., Shleifer, S., Lin, X.V., Du, J., Iyer, S., Pasunuru, R.: Efficient large scale language modeling with mixtures of experts. arXiv preprint arXiv:2112.10684 (2021)
5. Zhou, Y., Lei, T., Liu, H., Du, N., Huang, Y., Zhao, V., Dai, A., Chen, Z., Le, Q., Laudon, J.: Mixture-of-experts with expert choice routing, 2022. URL <https://arxiv.org/abs/2202.09368>
6. Lewis, M., Bhosale, S., Dettmers, T., Goyal, N., Zettlemoyer, L.: Base layers: Simplifying training of large, sparse models. In: *International Conference on Machine Learning*, pp. 6265-6274. PMLR, (Year)
7. Puerto, H., Şahin, G.G., Gurevych, I.: Metaqa: Combining expert agents for multi-skill question answering. arXiv preprint arXiv:2112.01922 (2021)
8. Si, C., Shi, W., Zhao, C., Zettlemoyer, L., Boyd-Graber, J.: Getting more out of mixture of language model reasoning experts. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8234-8249. (Year)
9. Jiang, D., Ren, X., Lin, B.Y.: Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. arXiv preprint arXiv:2306.02561 (2023)
10. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35, 24824-24837 (2022)

- 11.Si, C., Gan, Z., Yang, Z., Wang, S., Wang, J., Boyd-Graber, J., Wang, L.: Prompting gpt-3 to be reliable. arXiv preprint arXiv:2210.09150 (2022)
- 12.Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K.: Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7, 453-466 (2019)
- 13.Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., Grave, E.: Unsupervised dense information retrieval with contrastive learning. arXiv preprint arXiv:2112.09118 (2021)
- 14.Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W.W., Salakhutdinov, R., Manning, C.D.: HotpotQA: A dataset for diverse, explainable multi-hop question answering. arXiv preprint arXiv:1809.09600 (2018)
- 15.Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R.: Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168 (2021)
- 16.Talmor, A., Herzig, J., Lourie, N., Berant, J.: Commonsenseqa: A question answering challenge targeting commonsense knowledge. arXiv preprint arXiv:1811.00937 (2018)
- 17.Liu, J., Liu, A., Lu, X., Welleck, S., West, P., Bras, R.L., Choi, Y., Hajishirzi, H.: Generated knowledge prompting for commonsense reasoning. arXiv preprint arXiv:2110.08387 (2021)
- 18.Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
- 19.Gururangan, S., Li, M., Lewis, M., Shi, W., Althoff, T., Smith, N.A., Zettlemoyer, L.: Scaling expert language models with unsupervised domain discovery. arXiv preprint arXiv:2303.14177 (2023)
- 20.Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
- 21.<https://lmsys.org/blog/2023-06-29-longchat>
- 22.Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. *Neural computation* 3, 79-87 (1991)
- 23.Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538 (2017)
- 24.Gururangan, S., Lewis, M., Holtzman, A., Smith, N.A., Zettlemoyer, L.: Demix layers: Disentangling domains for modular language modeling. arXiv preprint arXiv:2108.05036 (2021)
- 25.Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D.S., Casas, D.d.l., Hanna, E.B., Bressand, F.: Mixtral of experts. arXiv preprint arXiv:2401.04088 (2024)
- 26.Wortsman, M., Ilharco, G., Gadre, S.Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A.S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S.: Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In: *International conference on machine learning*, pp. 23965-23998. PMLR, (Year)
- 27.Matena, M.S., Raffel, C.A.: Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems* 35, 17703-17716 (2022)

28. Ilharco, G., Ribeiro, M.T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., Farhadi, A.: Editing models with task arithmetic. arXiv preprint arXiv:2212.04089 (2022)
29. Jin, X., Ren, X., Preotiuc-Pietro, D., Cheng, P.: Dataless knowledge fusion by merging weights of language models. arXiv preprint arXiv:2212.09849 (2022)
30. Yadav, P., Tam, D., Choshen, L., Raffel, C., Bansal, M.: Resolving interference when merging models. arXiv preprint arXiv:2306.01708 (2023)
31. Zhang, J., Liu, J., He, J.: Composing Parameter-Efficient Modules with Arithmetic Operation. *Advances in Neural Information Processing Systems* 36, (2024)
32. Chronopoulou, A., Peters, M.E., Dodge, J.: Efficient hierarchical domain adaptation for pre-trained language models. arXiv preprint arXiv:2112.08786 (2021)
33. Chronopoulou, A., Peters, M.E., Fraser, A., Dodge, J.: AdapterSoup: Weight Averaging to Improve Generalization of Pretrained Language Models, March 2023. URL <http://arxiv.org/abs/2302.07027>