# From Continuous Pretraining to Domain-Adaptive Reranking via Task Vector Adaptation

Sanghyun Cho*
shcho.nlp@etri.re.kr
Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea

Myeongjin Lee*
myeongjin216@etri.re.kr
Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea

Jong-hun Shin
jhshin82@etri.re.kr
Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea

Jeong Heo
jeonghur@etri.re.kr
Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea

Kiyoung Lee
leeky@etri.re.kr
Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea

Soojong Lim†
isj@etri.re.kr
Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea

## Abstract

Large language model (LLM)-based rerankers have demonstrated strong performance in information retrieval tasks, but adapting them to specialized domains remains challenging due to the substantial cost and effort required to construct high-quality domain-specific training datasets. To address this limitation, we leverage task vectors derived from continuous pretraining as a mechanism for transferring domain knowledge. However, existing task vector integration methods are highly sensitive to scaling factors and can lead to unstable performance across domains and model scales. In this work, we propose a fine-grained task vector adaptation method that learns parameter-wise scaling coefficients for the task vector. These coefficients are optimized using a language modeling objective while keeping all model parameters fixed, enabling effective integration of domain-specific knowledge without degrading reranking capabilities. Experiments on a general-domain benchmark and five specialized domains across two model scales demonstrate that our method provides stable improvements across most domains and avoids the degradation observed with fixed task vector scaling.

## CCS Concepts

• **Computing methodologies** → *Learning settings*; • **Information systems** → **Language models**; **Learning to rank**.

## Keywords

task vector arithmetic; domain-adaptive reranking; adaptive merging

**ACM Reference Format:**
Sanghyun Cho, Myeongjin Lee, Jong-hun Shin, Jeong Heo, Kiyoung Lee, and Soojong Lim. 2026. From Continuous Pretraining to Domain-Adaptive Reranking via Task Vector Adaptation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3805712.3809933

*Both authors contributed equally to this research.
†Corresponding author.


ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3809933

## 1 Introduction

Recent advances in Large Language Models (LLMs) have enabled the development of LLM-based rerankers, which demonstrate state-of-the-art performance across diverse retrieval benchmarks. However, despite strong performance in general-domain settings, publicly available rerankers exhibit significant performance degradation under domain shift[5, 15]. This limitation arises because most models are trained on large-scale general-domain corpora; when applied to specialized domains, distributional discrepancies lead to reduced effectiveness. A straightforward solution is supervised domain adaptation through fine-tuning on domain-specific relevance annotations. Nevertheless, the construction of such datasets is resource-intensive and often impractical, particularly in expert-driven domains[3, 19].

A practical alternative is to use unlabeled domain corpora, which are easier to obtain than annotated reranking data. Continuous pretraining (CP) on domain text can incorporate domain knowledge without manual labeling. Its objective—next-token prediction—does not always align with relevance estimation, and excessive CP may lead to forgetting of general-domain knowledge.

To mitigate the instability associated with direct continual pretraining, prior work[4] proposed task vectors, defined as the parameter differences between a domain-adapted model and its base model. This approach has been applied to information retrieval and reranking tasks, where domain-specific task vectors are scaled by a global coefficient and added to pretrained models, thereby enabling zero-shot domain adaptation without additional fine-tuning. However, scalar-based integration is highly sensitive to the choice of the global scaling factor, which may result in unstable behavior or negative transfer.

Motivated by these limitations, we propose a fine-grained task vector adaptation framework for domain-specific LLM-based rerankers. Instead of relying on a manually tuned global coefficient, we introduce a vector of trainable scaling factors that are applied element-wise to the task vector. Specifically, a learnable scaling coefficient is assigned to each parameter of the task vector and optimized using a language modeling objective. This parameter-wise modulation enables fine-grained control over how domain knowledge is incorporated, allowing the model to selectively amplify beneficial components while suppressing harmful ones. The main contributions of this work are summarized as follows:

- A fine-grained task vector adaptation framework is proposed, replacing a single global scaling factor with element-wise trainable coefficients, thereby enabling parameter-wise control of domain knowledge integration using only unlabeled domain corpora.
- The proposed approach achieves competitive or improved performance across multiple domains compared to existing task vector integration methods.

# 2 Related Works

## 2.1 Task Vector Arithmetic

Task vector arithmetic has emerged as an effective approach for adapting pretrained language models to diverse downstream tasks. By representing task-specific knowledge as parameter differences between a base model and a task-adapted model, prior studies have shown that task vectors can be linearly combined or scaled to enable flexible task composition and adaptation [4, 7, 8, 13]. Beyond simple addition or scaling, more principled methods have been proposed to combine task vectors while mitigating interference between task-specific parameters [21, 24].

Chat vector[8] represents instruction-following and alignment capabilities as parameter differences between a base model and a chat-aligned model. By incorporating such vectors into continuously pretrained models, prior work demonstrates that instruction-following behavior can be transferred without additional supervised fine-tuning. Similarly, Braga et al.[4] showed that task vectors derived from domain-specific fine-tuned models can be integrated into information retrieval systems, enabling improved zero-shot retrieval performance without further training. These findings highlight the effectiveness of task vectors as a mechanism for injecting new capabilities or domain knowledge into pretrained models. However, prior work[4, 8] has shown that performance is highly sensitive to the choice of the scaling factor used for task vector integration.

To address this limitation, several approaches have been proposed to replace a single global scaling factor with more structured weighting schemes. AdaMerging[23] formulates task vector merging as an optimization problem, learning task-specific coefficients via entropy minimization on unlabeled multi-task data. Instead of directly learning scaling coefficients, LATA[6] estimates the relative dominance of instruction-following and task-specific behavior at each layer by measuring layer-wise cosine similarity between instruction vectors. Based on this estimate, layer-dependent weights are assigned to control the strength of task vector arithmetic at each layer. In contrast to layer- or block-level approaches, we assign trainable scaling coefficients to each parameter of the task vector and optimize them using a language modeling objective for fine-grained adaptation.

## 2.2 LLM-based Reranker

Recent studies have demonstrated that LLMs achieve strong performance in retrieval and reranking tasks by leveraging advanced reasoning and language understanding capabilities [1, 11, 14, 25]. In particular, LLM-based rerankers have been widely explored, where a model evaluates the relevance between a query and a document by jointly encoding them within a single context.

In reranking, existing approaches can be broadly categorized into prompt-based zero-shot methods[18] and supervised fine-tuning (SFT)-based methods. The former assess relevance by conditioning the LLM on a query–document pair and comparing the generation probabilities of predefined tokens (e.g., *yes*/*no*), while the latter improves performance by training on labeled data.

Qwen3-Reranker[25] adopts a point-wise reranking formulation, in which the query, document, and instruction prompt are concatenated into a single input context, and relevance is estimated by comparing the generation probabilities of binary tokens. In this work, we use Qwen3-Reranker for our experiments, as both the reranker and its corresponding pretrained base model[22] are publicly available.

# 3 Methodology

## 3.1 Preliminary

Following prior work[4], we construct a domain-specific reranker by leveraging task vectors derived from continuous pretraining. Specifically, we apply continuous pretraining to a pretrained base model using unlabeled raw corpora from the target domain, enabling the model to acquire domain-specific knowledge. We then compute a task vector as the parameter difference between the continuously pretrained model and the original base model, and integrate it into the reranker to obtain a domain-adapted reranking model.

Let $\boldsymbol{\theta}_{\text{pre}}$ denote the parameters of the pretrained base model, and $\boldsymbol{\theta}^{(i)}_{\text{dom}}$ denote those of the model continuously pretrained on the $i$-th domain corpus. The parameters of the reranker model are denoted as $\boldsymbol{\theta}_{\text{rerank}}$.

The task vector corresponding to the $i$-th domain is defined as the parameter difference between the domain-adapted model and the base model:

$$\boldsymbol{v}^{(i)} = \boldsymbol{\theta}^{(i)}_{\text{dom}} - \boldsymbol{\theta}_{\text{pre}}. \tag{1}$$

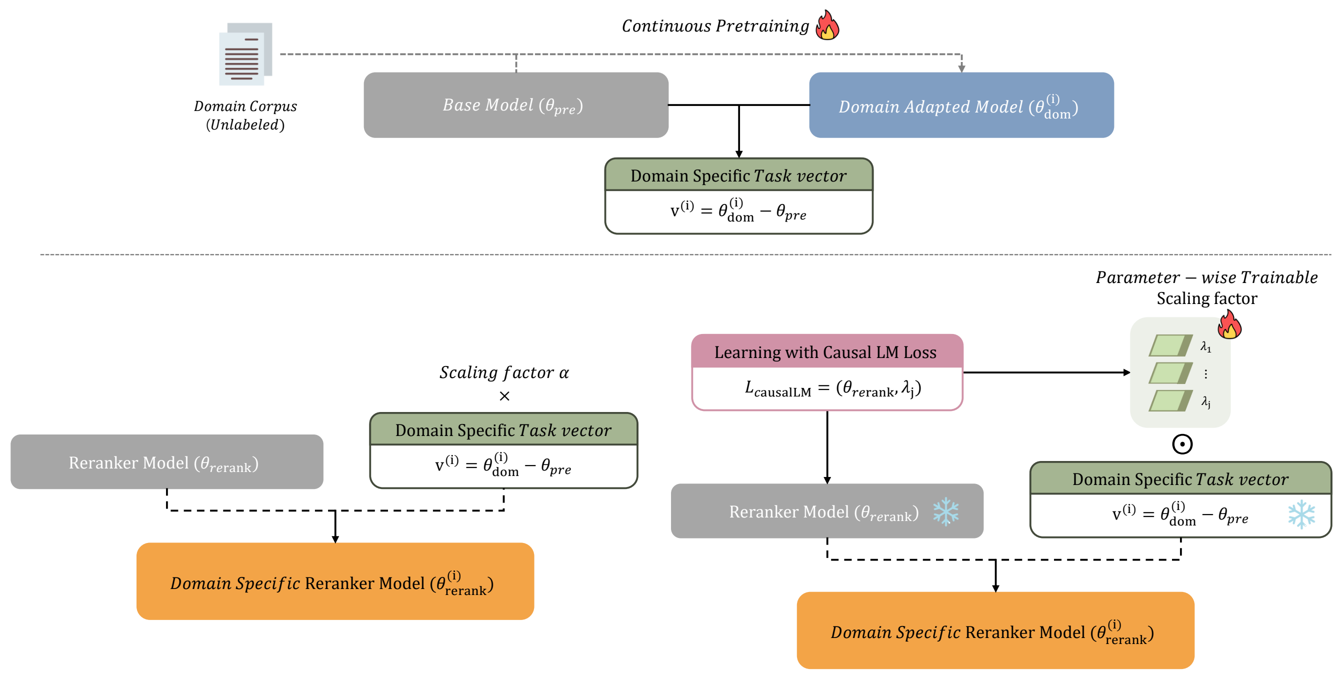
The domain-specific reranker is obtained by combining the reranker parameters with the task vector as follows:

$$\boldsymbol{\theta}^{(i)}_{\text{rerank}} = \boldsymbol{\theta}_{\text{rerank}} + \alpha \boldsymbol{v}^{(i)}, \tag{2}$$

where $\alpha$ is a scalar coefficient controlling the contribution of the task vector.

## 3.2 Fine-grained Task Vector Adaptation

Previous studies have shown that the performance of task vector-based adaptation is highly sensitive to the scalar coefficient that governs the contribution of the task vector [4, 8]. Depending on the value of this coefficient, incorporating a task vector can lead to substantial performance gains, but may also result in performance degradation in certain cases. This observation highlights the importance of integrating task vectors in a way that effectively leverages




(A) Task vector adaptation with fixed scalar scaling

(B) Fine-grained task vector adaptation with parameter-wise trainable scaling

**Figure 1: Method overview comparing prior task vector integration with our proposed fine-grained adaptation.**

domain-specific knowledge while avoiding interference with the base model's original capabilities.

To address this issue, we assign a randomly initialized trainable scalar coefficient to each parameter of the task vector and optimize these coefficients using the language modeling loss. By learning parameter-wise scaling factors, our method enables fine-grained control over how domain-specific knowledge is incorporated into the reranker. Specifically, we introduce a vector of trainable scaling coefficients $\boldsymbol{\lambda} \in \mathbb{R}^{|v^{(i)}|}$, where each element corresponds to a weight in the task vector. The domain-specific reranker parameters are then computed as:

$$\boldsymbol{\theta}^{(i)}_{\text{rerank}} = \boldsymbol{\theta}_{\text{rerank}} + \boldsymbol{\lambda} \odot \boldsymbol{v}^{(i)}, \tag{3}$$

where $\odot$ denotes element-wise multiplication.

To ensure that the task vector effectively contributes to the reranking task, we further optimize the scaling coefficients using a language modeling loss on general-domain reranker training data. Specifically, we use the TriviaQA training set as the general-domain question answering dataset. For each question, the positive passage is defined as the Wikipedia passage that contains the ground-truth answer, while negative passages are sampled from passages that do not contain the answer.

Unlike prior approaches that apply a shared scaling coefficient at the layer or block level, our method assigns independent trainable scalars to individual parameters of the task vector. Although this fine-grained formulation increases adaptation flexibility, it can also distort the original task vector, potentially weakening the domain-specific knowledge acquired through continuous pretraining. To mitigate this issue, we incorporate a portion of the unlabeled domain corpus used for continuous pretraining into the training data for coefficient optimization, together with the general-domain QA dataset. This mixed training strategy encourages the learned scaling coefficients to preserve domain-specific knowledge while maintaining compatibility with the reranking task.

During this stage, all parameters of the reranker model and the task vector are frozen, and only the scaling coefficients are updated. In our experiments, we set the mixing ratio between general-domain QA data and unlabeled domain corpora to 9:1 based on empirical validation.

## 4 Experimental Setup

To evaluate the proposed method, we conduct experiments on both a general-domain corpus and five specialized domains: patent, legal, medical, financial, and biomedical. For each domain, we perform continuous pretraining using domain-relevant unlabeled corpora and evaluate reranking performance on corresponding QA datasets. For the general-domain setting, we construct reranking evaluation datasets based on open-domain QA benchmarks, namely Natural Questions (NQ) [10] and WebQuestions (WebQ) [2]. We use these datasets as they provide explicit annotations linking questions to source Wikipedia documents, enabling reliable construction of positive and negative passages. Passages containing the ground-truth

**Table 1: Comparison of reranking performance across domains using the 0.6B model with different task vector integration methods. The upper table reports Top-1 Accuracy and MRR, while the lower table is reserved for nDCG@3 and nDCG@5.**

| Method | Finance | | Bio | | Law | | Medical | | Patent | | NQ | | WebQ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | MRR | Acc | MRR | Acc | MRR | Acc | MRR | Acc | MRR | Acc | MRR | Acc | MRR |
| Qwen3-Reranker | .604 | .774 | .864 | .920 | .578 | .751 | .580 | .748 | .562 | .743 | .463 | .653 | .282 | .452 |
| Qwen3-Reranker + CP | .685 | .819 | .868 | .922 | .406 | .632 | .394 | .626 | .569 | .746 | .477 | .659 | .296 | .467 |
| Task Vector (Fixed) | .765 | .866 | .820 | .895 | .610 | .771 | .564 | .737 | **.663** | **.805** | .484 | .664 | .307 | .470 |
| Task Vector (TIES) | **.858** | **.920** | .830 | .901 | .588 | .758 | .566 | .738 | .634 | .788 | .460 | .652 | .307 | .470 |
| Task Vector (Ours) | .811 | .895 | **.883** | **.930** | **.619** | **.778** | **.605** | **.767** | .487 | .699 | **.506** | **.679** | **.321** | **.486** |
| Method | Finance | | Bio | | Law | | Medical | | Patent | | NQ | | WebQ | |
| | n@3 | n@5 | n@3 | n@5 | n@3 | n@5 | n@3 | n@5 | n@3 | n@5 | n@3 | n@5 | n@3 | n@5 |
| Qwen3-Reranker | .816 | .832 | .930 | .940 | .784 | .814 | .779 | .812 | .781 | .808 | .666 | .706 | .417 | .475 |
| Qwen3-Reranker + CP | .831 | .851 | .932 | .940 | .657 | .724 | .656 | .720 | .780 | .810 | .674 | .713 | .440 | .492 |
| Task Vector (Fixed) | .889 | .900 | .908 | .921 | .801 | .829 | .766 | .803 | **.835** | **.855** | .681 | .713 | .435 | .496 |
| Task Vector (TIES) | **.936** | **.940** | .913 | .925 | .789 | .819 | .768 | .804 | **.835** | **.855** | .670 | .706 | .437 | .497 |
| Task Vector (Ours) | .915 | .922 | **.941** | **.948** | **.812** | **.834** | **.799** | **.826** | .781 | .807 | **.691** | **.729** | **.454** | **.513** |

**Table 2: Comparison of reranking performance across domains using the 4B model with different task vector integration methods. The upper table reports Top-1 Accuracy and MRR, while the lower table is reserved for nDCG@3 and nDCG@5.**

| Method | Finance | | Bio | | Law | | Medical | | Patent | | NQ | | WebQ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | MRR | Acc | MRR | Acc | MRR | Acc | MRR | Acc | MRR | Acc | MRR | Acc | MRR |
| Qwen3-Reranker | .945 | .970 | .947 | .970 | **.940** | **.967** | .929 | .960 | .895 | .943 | .546 | .706 | .311 | .477 |
| Qwen3-Reranker + CP | .888 | .939 | .935 | .962 | .910 | .950 | .912 | .952 | .868 | .928 | .539 | .701 | .338 | .494 |
| Task Vector (Fixed) | .915 | .953 | .929 | .959 | .926 | .959 | .903 | .947 | **.917** | **.954** | .540 | .703 | .346 | **.497** |
| Task Vector (TIES) | .935 | .964 | .935 | .963 | .935 | .964 | .905 | .948 | .916 | .953 | .539 | .703 | .346 | **.497** |
| Task Vector (Ours) | **.967** | **.982** | **.968** | **.982** | .929 | .961 | **.932** | **.964** | .911 | .952 | **.547** | **.707** | **.350** | **.497** |
| Method | Finance | | Bio | | Law | | Medical | | Patent | | NQ | | WebQ | |
| | n@3 | n@5 | n@3 | n@5 | n@3 | n@5 | n@3 | n@5 | n@3 | n@5 | n@3 | n@5 | n@3 | n@5 |
| Qwen3-Reranker | .977 | .978 | **.976** | .977 | **.974** | **.976** | .969 | .971 | .955 | .957 | **.720** | **.754** | .451 | .498 |
| Qwen3-Reranker + CP | .954 | .955 | .968 | .968 | .960 | .963 | .962 | .964 | .942 | .946 | .711 | .750 | .462 | .518 |
| Task Vector (Fixed) | .963 | .965 | .965 | .969 | .968 | .970 | .957 | .960 | .962 | **.965** | .717 | .749 | .458 | .523 |
| Task Vector (TIES) | .971 | .973 | .968 | .972 | .972 | .973 | .958 | .961 | .962 | **.965** | .717 | .749 | .458 | .520 |
| Task Vector (Ours) | **.986** | **.986** | .968 | **.985** | .969 | .971 | **.972** | **.973** | **.963** | .964 | **.720** | **.754** | **.476** | **.523** |

answer within the annotated Wikipedia documents are used as positive passages. When multiple such passages are available for a given question, we retain only one positive passage per question by sampling. Other passages from the same documents that do not contain the answer are treated as negative passages. We use the Wikipedia corpus for continuous pretraining, while the QA data with their corresponding positive and negative passages are used for reranking evaluation.

For specialized domains, we evaluate reranking performance on datasets closely aligned with the corpora used for continuous pretraining. We first perform continuous pretraining on publicly available domain-specific corpora, and then synthesize QA pairs with positive and negative passages based on these corpora using a large-scale commercial language model. For each question, we generate one positive passage and five to seven negative passages. This procedure enables the construction of reranking evaluation datasets that reflect the domain knowledge learned during continuous pretraining.

Specifically, we use the BigPatent dataset[17] for the patent domain, the English legislation split of MultiLegalPile[12] for the legal domain, Medical Abstracts[16] for the medical domain, S&P 500 Earnings Transcripts[9] for the financial domain, and CORD-19[20] for the biomedical domain. For datasets containing more than 50K documents, we randomly sample 50K documents. For smaller datasets, we use the full corpus for both continuous pretraining and evaluation data synthesis.

We conduct all experiments using Qwen3-Base[22] and Qwen3-Reranker[25] models with 0.6B and 4B. As a baseline, we adopt a task vector integration approach that adds a task vector scaled by a fixed scalar coefficient to the reranker parameters, following prior work[4]. We further include models constructed using TIES-merging[21] to evaluate alternative task vector combination strategies beyond simple addition.

**Table 3: Comparison of data mixture ratios for scaling coefficient optimization on the 0.6B reranker across seven domains.**

| Ratio | Finance | | Bio | | Law | | Medical | | Patent | | NQ | | WebQ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | n@5 | Acc | n@5 | Acc | n@5 | Acc | n@5 | Acc | n@5 | Acc | n@5 | Acc | n@5 | Acc |
| Proposed ratio | **.922** | **.811** | **.948** | **.883** | .834 | .619 | .826 | .605 | .807 | .487 | **.729** | **.506** | **.513** | **.321** |
| Domain raw only | .900 | .764 | .907 | .803 | .849 | .654 | .793 | .542 | **.843** | **.639** | .726 | .503 | .507 | .305 |
| General QA only | .893 | .745 | .898 | .778 | **.852** | **.661** | **.831** | **.615** | .776 | .493 | .718 | .488 | .492 | .296 |
| Half ratio | .909 | .778 | .923 | .826 | .831 | .613 | .817 | .588 | .784 | .510 | .716 | .481 | .476 | .278 |

## 5 Results and Discussion

Tables 1 and 2 present reranking performance across domains using the 0.6B and 4B models, respectively. Qwen3-Reranker refers to the publicly available reranker evaluated without additional training, while Qwen3-Reranker + CP corresponds to the reranker directly adapted via continuous pretraining on domain-specific corpora. Among task vector-based methods, Fixed uses a task vector scaled by a fixed coefficient, TIES denotes models constructed via TIES-merging, and Ours represents our proposed parameter-wise task vector adaptation optimized with a language modeling loss.

Directly adapting the reranker via continuous pretraining often results in degraded performance compared to the base reranker, particularly in the Law and Medical domains. This suggests that domain-specific pretraining can lead to forgetting of the model's original reranking capabilities, rather than effectively incorporating domain knowledge. In contrast, incorporating task vectors derived from domain-specific continuous pretraining generally yields better reranking performance than directly fine-tuned rerankers across most domains.

For the 0.6B model, our method achieves the best performance in most domains, including Bio, Law, Medical, NQ, and WebQ, while remaining competitive in Finance and showing weaker performance in the Patent domain. While TIES-merging performs strongly in certain domains such as Finance, our method exhibits more consistent improvements across domains. For the larger 4B model, the baseline reranker already achieves strong performance across domains, leaving limited room for improvement. While some task vector integration methods lead to performance drops in certain domains, our method still improves performance in Finance, Bio, Medical, NQ, and WebQ, although slight degradations remain in Law and Patent.

Table 3 compares the performance of different data mixture ratios for scaling coefficient optimization on the 0.6B reranker across seven domains. Although the optimal data mixture differs between domains, the proposed mixture generally delivers more consistent performance across a wide range of settings. In particular, general QA data helps align the task vector coefficients with the reranking objective, while incorporating a domain corpus helps preserve domain-specific knowledge acquired during continuous pretraining. The mixed strategy offers a balance between these two factors, leading to more stable overall performance. These results suggest that coefficient optimization benefits from balancing task-specific supervision with domain-relevant unlabeled data, rather than relying exclusively on either source.

## 6 Conclusion

In this paper, we study task vector-based adaptation for domain-specific reranking with large language models. While task vectors derived from continuous pretraining provide a promising mechanism for transferring domain knowledge without additional task-specific supervision, our results show that simple integration strategies are sensitive to scaling choices and can lead to performance degradation in certain domains.

To address this limitation, we propose a fine-grained task vector adaptation method that assigns parameter-wise trainable scaling coefficients to the task vector. By optimizing these coefficients using a language modeling objective while keeping all original model parameters fixed, our approach enables finer control over the integration of domain-specific knowledge into pretrained rerankers.

Experimental results across general-domain and five specialized domains demonstrate that our method achieves competitive performance in most domains for the 0.6B model, outperforming other task vector-based approaches in several cases. For the larger 4B model, where baseline performance is already strong, our approach still improves performance in most domains, with only minor degradations observed in Law and Patent. Overall, compared to alternative integration strategies, our method demonstrates more consistent behavior across domains.

These findings highlight the importance of fine-grained control over task vector contributions as a practical and data-efficient mechanism for domain adaptation in LLM-based reranking systems.

## Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) (No. RS-2023-00216011, Development of Artificial Complex Intelligence for Conceptually Understanding and Inferring like Human) and IITP grant funded by the Korea Government (MSIT) (No. RS-2024-00338140, Development of learning and utilization technology to reflect sustainability of generative language models and up-to-dateness over time).

## References

[1] Yauhen Babakhin, Radek Osmulski, Ronay Ak, Gabriel De Souza Pereira Moreira, Mengyao Xu, Benedikt Schifferer, Bo Liu, and Even Oldridge. 2025. Llama-Embed-Nemotron-8B: A Universal Text Embedding Model for Multilingual and Cross-Lingual Tasks. *ArXiv* abs/2511.07025 (2025).

[2] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1533–1544.

[3] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144* (2022).

[4] Marco Braga, Pranav Kasela, Alessandro Raganato, and Gabriella Pasi. 2025. Investigating Task Arithmetic for Zero-Shot Information Retrieval. *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2025).

[5] Ramraj Chandradevan, Kaustubh D. Dhole, and Eugene Agichtein. 2024. DUQGen: Effective Unsupervised Domain Adaptation of Neural Rankers by Diversifying Synthetic Query Generation. In *North American Chapter of the Association for Computational Linguistics*.

[6] Yan-Lun Chen, Yi-Ru Wei, Chia-Yi Hsu, Chia-Mu Yu, Chun-Ying Huang, Ying-Dar Lin, Yu-Sung Wu, and Wei-Bin Lee. 2025. Layer-Aware Task Arithmetic: Disentangling Task-Specific and Instruction-Following Knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2025*. 12033–12054.

[7] Constanza Fierro and Fabien Roger. 2025. Steering Language Models with Weight Arithmetic. *ArXiv* abs/2511.05408 (2025).

[8] Shih-Cheng Huang, Pin-Zu Li, Yu-chi Hsu, Kuang-Ming Chen, Yu Tung Lin, Shih-Kai Hsiao, Richard Tsai, and Hung-yi Lee. 2024. Chat Vector: A Simple Approach to Equip LLMs with Instruction Following and Model Alignment in New Languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[9] Kurry. 2025. *S&P 500 Earnings Transcripts Dataset*. https://huggingface.co/datasets/kurry/sp500_earnings_transcripts

[10] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (2019), 452–466.

[11] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-Tuning LLaMA for Multi-Stage Text Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2421–2425.

[12] Joel Niklaus, Veton Matoshi, Matthias Stürmer, Ilias Chalkidis, and Daniel Ho. 2024. MultiLegalPile: A 689GB Multilingual Legal Corpus. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 15077–15094.

[13] Abraham Toluwase Owodunni and Sachin Kumar. 2025. Continually Adding New Languages to Multilingual Language Models. *ArXiv* abs/2509.11414 (2025).

[14] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy J. Lin. 2023. RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze! *ArXiv* abs/2312.02724 (2023).

[15] Jon Saad-Falcon, O. Khattab, Keshav Santhanam, Radu Florian, Martin Franz, Salim Roukos, Avirup Sil, Md Arafat Sultan, and Christopher Potts. 2023. UDAPDR: Unsupervised Domain Adaptation via LLM Prompting and Distillation of Rerankers. *ArXiv* abs/2303.00807 (2023).

[16] Tim Schopf, Daniel Braun, and Florian Matthes. 2022. Evaluating Unsupervised Text Classification: Zero-shot and Similarity-based Approaches. *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval* (2022).

[17] Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization. *ArXiv* abs/1906.03741 (2019). https://api.semanticscholar.org/CorpusID:182953211

[18] Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *ArXiv* abs/2304.09542 (2023).

[19] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663* (2021).

[20] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Nancy Xin Ru Wang, Christopher Wilhelm, Boya Xie, Douglas M. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. CORD-19: The COVID-19 Open Research Dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*.

[21] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. TIES-Merging: Resolving Interference When Merging Models. In *Neural Information Processing Systems*.

[22] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Jingren Zhou, Junyan Lin, Kai Dang, Keqin Bao, Ke-Pei Yang, Le Yu, Li-Chun Deng, Mei Li, Min Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shi-Qiang Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 Technical Report. *ArXiv* abs/2505.09388 (2025).

[23] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2023. AdaMerging: Adaptive Model Merging for Multi-Task Learning. *ArXiv* abs/2310.02575 (2023).

[24] 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 *(ICML'24)*.

[25] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *ArXiv* abs/2506.05176 (2025).