Is Micro Domain-Adaptive Pre-Training Effective for Real-World Operations? Multi-Step Evaluation Reveals Potential and Bottlenecks

Masaya Tsunokake **Yuta Koreeda** **Terufumi Morishita**
Koichi Nagatsuka **Hikaru Tomonari** **Yasuhiro Sogawa**
Research & Development Group, Hitachi, Ltd
Tokyo, Japan
Correspondence: masaya.tsunokake.qu@hitachi.com

Abstract

When applying LLMs to real-world enterprise operations, LLMs need to handle proprietary knowledge in small domains of specific operations (**micro domains**). A previous study (Xue et al., 2025) shows micro domain-adaptive pre-training (**mDAPT**) with fewer documents is effective, similarly to DAPT in larger domains. However, it evaluates mDAPT only on multiple-choice questions; thus, its effectiveness for generative tasks in real-world operations remains unknown. We aim to reveal the potential and bottlenecks of mDAPT for generative tasks. To this end, we disentangle the answering process into three subtasks and evaluate the performance of each subtask: (1) **eliciting** facts relevant to questions from an LLM's own knowledge, (2) **reasoning** over the facts to obtain conclusions, and (3) **composing** long-form answers based on the conclusions. We verified mDAPT on proprietary IT product knowledge for real-world questions in IT technical support operations. As a result, mDAPT resolved the elicitation task that the base model struggled with but did not resolve other subtasks. This clarifies mDAPT's effectiveness in the knowledge aspect and its bottlenecks in other aspects. Further analysis empirically shows that resolving the elicitation and reasoning tasks ensures sufficient performance (over 90%), emphasizing the need to enhance reasoning capability.

1 Introduction

Driven by the remarkable advances of large language models (LLMs) (Brown et al., 2020; OpenAI, 2023), many companies are increasingly utilizing LLMs for their internal operations. When applying LLMs to real-world operations, LLMs need to handle proprietary knowledge in each company or operation (Ling et al., 2023; Zhao et al., 2024). However, LLMs cannot generate content grounded in knowledge outside their training data.

Domain-adaptive pre-training (DAPT) (Gururangan et al., 2020) is one approach for enabling LLMs to handle unseen knowledge. Previous studies show DAPT improves LLMs on many tasks in medical (Singhal et al., 2023), financial (Wu et al., 2023), legal (Colombo et al., 2024), and code (Gunasekar et al., 2023) domains. Meanwhile, real-world operations often demand knowledge within **small** and **proprietary** domains of specific operations (**micro domains**), and micro domains have far fewer documents than larger domains.

Xue et al. (2025) investigated the effectiveness of DAPT in a micro domain (**mDAPT**: micro Domain-Adaptive Pre-Training). However, their evaluation was limited to multiple-choice questions (MCQs). Compared to MCQs where LLMs can select from limited choices, complicated real-world questions require LLMs not to select but generate long-form answers from scratch by utilizing their trained knowledge. To advance enterprise use of LLMs, it is important to reveal whether mDAPT is effective for generative tasks in real-world operations, and if not, what bottlenecks there are.

This paper aims to reveal the potential and bottlenecks of mDAPT for generative tasks. To clarify what aspects of generative tasks are difficult for mDAPT models, we disentangle the answering process into three subtasks: (1) **eliciting** facts relevant to questions from an LLM's own knowledge, (2) **reasoning** over the facts to obtain conclusions, and (3) **composing** long-form answers based on the conclusions (Figure 1(a)). We identify bottleneck subtasks by observing LLM's overall performance changes after inserting an ideal result of each subtask (**oracle result**) into prompts. Inserting a subtask's oracle result enables LLMs to solve subsequent subtasks based on it. Thus, if inserting an oracle result of a previous subtask improves overall performance, it indicates that the LLM was not able to adequately solve the subtask by itself, and the subtask is a bottleneck.

We trained mDAPT models from Qwen2.5-72B-Instruct (Qwen et al., 2025) with proprietary IT

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 5: Industry Track, pages 304–316

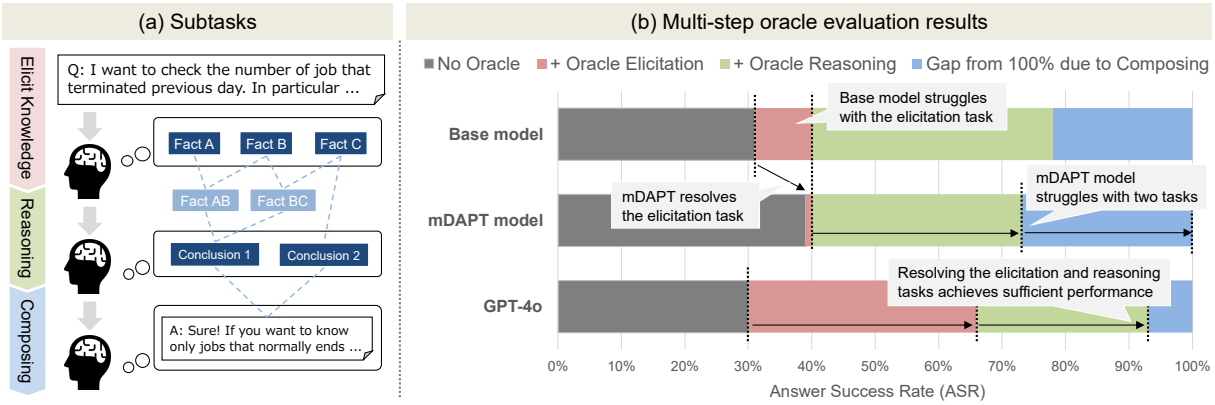

Figure 1: (a) Subtasks in an answering process (b) Results by our evaluation framework. To identify bottleneck tasks, our framework observes performance changes after inserting ideal results of each task (**oracle result**) into prompts. Although the base model struggles with the elicitation task, mDAPT resolves this difficulty, showing mDAPT's effectiveness. However, the mDAPT model still struggles with the reasoning and composing tasks.

product knowledge and evaluated them on real-world questions in IT technical support operations. We found that mDAPT resolves the elicitation task that the base model struggles with (Figure 1(b)), showing mDAPT's effectiveness in acquiring and eliciting knowledge. However, the mDAPT model exhibits insufficient performance (39%). The improvements by inserting oracle results show that the mDAPT model struggles with the reasoning and composing tasks, revealing mDAPT's bottlenecks.

For comparison, Figure 1(b) also shows the performance of GPT-4o (OpenAI et al., 2024). This indicates that sufficient performance (over 90%) could be achieved by resolving the elicitation and reasoning tasks if a stronger proprietary model were the base model. Given mDAPT resolves the elicitation task, our results empirically highlight that enhancing reasoning capability to resolve the reasoning task is a high-priority and promising approach for realizing a sufficiently usable model for real-world operations.

2 Background

2.1 Micro Domain

A previous study focused on micro domain knowledge of an IT product for evaluating mDAPT (Xue et al., 2025). Since many companies introduce IT products for their operations, including proprietary ones, question answering for IT products is a representative use case in real-world operations. Therefore, we adopt this use case to study mDAPT.

We use JP1 (Hitachi, 2025a), a typical IT product that constitutes proprietary domain knowledge, for mDAPT as in the previous study (Xue et al., 2025). JP1 consists of several software (e.g., job scheduler, database) with proprietary terminology and specification sets. Figure 2 shows JP1's proprietary terms. According to the upper example, the term "job" has a definition that differs from its general meaning. The lower example describes "waiting condition," which is a proprietary specification. Reasoning over underlined facts yields the new conclusion that "a wait condition can control the execution sequence of **jobs**." As like this, various facts mutually interact in this JP1 domain.

3.1 Hierarchical structure of the job network
In JP1/AJS3, the elements in an application to be automated are known as *units*. Each of the individual processes involved in an application is defined as a unit called a *job*. A job is the smallest unit. You then arrange the defined jobs in order of execution, creating a network of jobs grouped together to form a unified application. This collection of jobs is called a *jobnet*.

2.2.5 Using wait conditions to control the order of unit execution
You can use a *wait condition* to control the execution sequence of units in different jobnets. A unit assigned a wait condition does not start to execute until a specified unit ...

Figure 2: Facts written in JP1 manuals (Hitachi, 2025b)

2.2 Micro Domain-Adaptive Pre-Training

Our mDAPT consists of continual pre-training (CPT) and supervised fine-tuning (SFT), which is a standard DAPT procedure. CPT is performed by a next token prediction task on raw corpora. SFT is performed by the same task, using only a subset of tokens for loss computation. We use QA pairs for SFT; thus, the loss is calculated only on answer tokens. We synthesize the QA pairs from raw corpora used for CPT to enhance LLMs' capability to

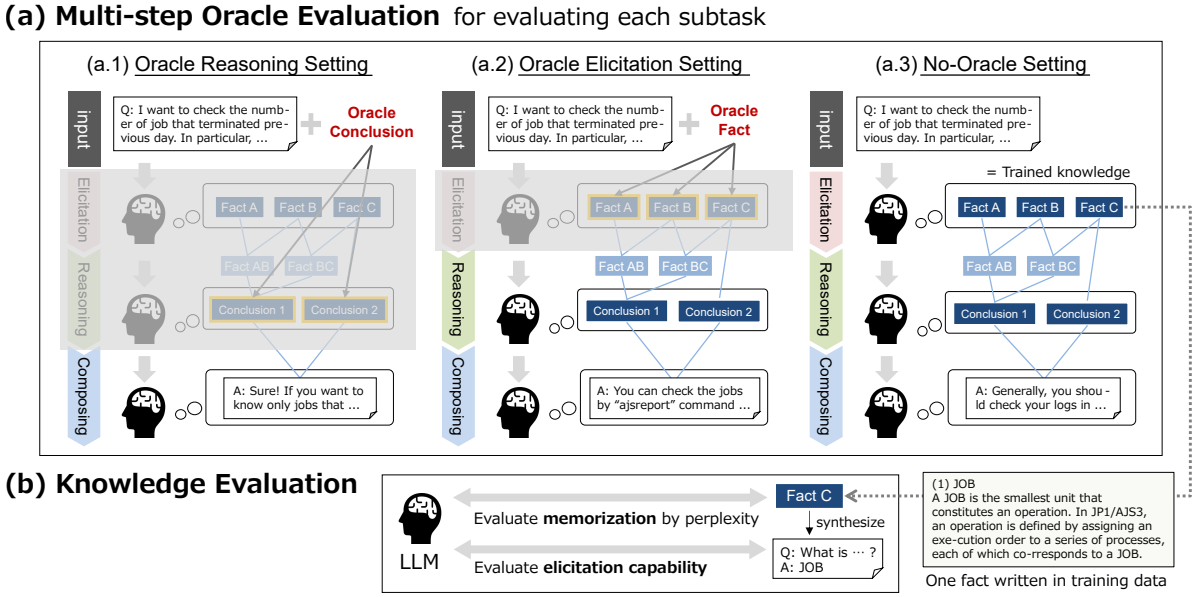

Figure 3: Our evaluation framework consists of multi-step oracle evaluation and knowledge evaluation. In multi-step oracle evaluation, we observe LLM's overall performance changes after inserting oracle results for subtasks. If the performance is improved, it means that an LLM could not solve the corresponding task by itself. In knowledge evaluation, we evaluate LLM's memorization and elicitation capability by using trained texts relevant to questions.

utilize knowledge (Cheng et al., 2024a; Jiang et al., 2024; Cheng et al., 2024b; Ziegler et al., 2024).

3 Evaluation Framework

3.1 Overview

LLMs generally need to generate answers using their knowledge. Thus, LLMs should be capable of **eliciting** knowledge. However, LLMs may receive questions asking about unseen facts not directly written in trained knowledge. In such scenarios, LLMs must **reason** over known facts to obtain new conclusions needed for answers, a well-known nature in multi-hop question answering (Yang et al., 2018; Trivedi et al., 2022). In enterprise use, LLMs need to handle multi-hop questions grounded in proprietary knowledge (Zhao et al., 2024). Therefore, we claim that mDAPT models need to address three subtasks in an answering process: (1) **eliciting** relevant facts from their trained knowledge, (2) **reasoning** over the facts to obtain conclusions, and (3) **composing** long-form answers based on the conclusions, as shown in Figure 3(a.3).

We aim to clarify whether each subtask is a bottleneck in real-world questions. To this end, we introduce a simple but explainable method that observes LLM's overall performance changes after inserting an ideal result of each subtask (**oracle result**) into prompts. For example, inserting an oracle result of the reasoning task (**oracle conclusions**) allows us to observe an LLM's overall performance when the reasoning task is perfectly solved. If inserting oracle conclusions improves the overall performance, it indicates that the LLM was not able to adequately solve the reasoning task by itself, and the reasoning task is a bottleneck. We call this evaluation **Multi-step Oracle Evaluation**.

The elicitation task assumes that LLMs have memorized the knowledge and can access them on demand. To verify whether these requirements are bottlenecks, we additionally evaluate LLMs' memorization and how well LLMs can access knowledge (**elicitation capability**). We call this evaluation **Knowledge Evaluation**.

3.2 Multi-step Oracle Evaluation

3.2.1 Multiple Oracle Settings

We define the following three prompt settings to realize the multi-step oracle evaluation (Figure 3(a)).

Oracle Reasoning Setting This inserts **oracle conclusions**, information directly leading to correct answers, into prompts. This allows LLMs to compose answers based on ideal results of the reasoning task (Figure 3(a.1)). We can determine whether the composing task is a bottleneck by comparing this performance with the upper bound. Oracle conclusions include domain-specific facts mentioned in correct answers and guidance information that specifies how LLMs should construct answers. Ta-

Usage	Data name	Example
Multi-step Oracle Evaluation	Condition for LLM-as-a-judge	The given answer describes a solution that uses the "ajsreport" command to output the previous day's performance report and aggregates "JOB_EXEC_END_N_NUM," "NEST_EXEC_END_N_NUM," and "RJ_EXEC_END_N_NUM."
	Oracle conclusion	By executing the "ajsreport" command to output the previous day's performance report and checking "JOB_EXEC_END_N_NUM," "NEST_EXEC_END_N_NUM," and "RJ_EXEC_END_N_NUM" items, you can confirm the number of jobs and jobnets that terminated normally previous day with a single command.
	Oracle fact 1	*ajsreport* The ajsreport command outputs JP1/AJS3 performance reports.
	Oracle fact 2	NEST_EXEC_END_N_NUM: Number of nested jobnets or nested remote jobnets that terminated normally. This value is incremented according to the number of jobnets that are placed in the Ended normally status.
Knowledge Evaluation	Closed-book QA synthesized from oracle fact 2	Q: Which parameter indicates the number of nested jobnets or nested remote jobnets that have terminated normally, where the value increases according to the number of jobnets in the "Ended normally" status? A: NEST_EXEC_END_N_NUM

Table 1: Example data used in our evaluation framework. We show examples made by authors that replicate the characteristics of our real data due to confidentiality issues. The condition for LLM-as-a-judge specifies information that LLMs' answers should cover. The oracle conclusion is information that directly leads to the correct answer. The oracle facts are extracted training data (Hitachi, 2025c) and support the oracle conclusion.

Name	Usage	Description	# Documents	Size (MB)		# Tokens (M)	
JP1 Documents	CPT	JP1 manuals (Hitachi, 2025b), release notes, and other references	193	72.1		18.3	
llm-jp-corpus-v2	CPT	Subset of llm-jp-corpus-v2, which contains various Japanese texts	-	651.4		-	
JP1-QA	SFT	JP1/AJS-related QA pairs synthesized from JP1 manuals	12,305	42.9	(9.5)	10.6	(2.1)

Table 2: Training data of mDAPT. The JP1 documents are obtained by converting original PDF files to texts. For JP1-QA, the statistics of the answer part, actually used for the SFT loss computation, are shown within parentheses.

ble 1 shows an example of an oracle conclusion.

Oracle Elicitation Setting This inserts oracle results of the elicitation task, relevant texts from training data (**oracle facts**), into prompts. Thus, LLMs can address only the reasoning and composing tasks based on oracle facts (Figure 3(a.2)). We can determine whether the reasoning task is a bottleneck by comparing performances of this and the oracle reasoning settings. Table 1 shows examples of oracle facts.

No-Oracle Setting This does not insert oracle results into prompts; thus, LLMs must address all tasks. We can determine whether the elicitation task is a bottleneck by comparing performances of this and the oracle elicitation settings.

3.2.2 LLM-as-a-judge for Each Setting

To efficiently evaluate each setting, we introduce an LLM-as-a-judge (Zheng et al., 2023) method that determines whether each generated answer covers all required information. Given multiple **checklists**[1] each of which contains **conditions** for one question to be deemed correct, an evaluator LLM judges whether a generated answer satisfies each condition. If the generated answer satisfies all conditions in any checklist, the answer is regarded as correct. Table 1 shows an example of a condition prompted into an evaluator LLM.

[1]Each question may be associated with more than one checklist as there can be multiple approaches to answering.

3.3 Knowledge Evaluation

Based on previous studies (Jiang et al., 2024; Chang et al., 2025), we evaluate **memorization** by computing perplexity of oracle facts and evaluate **elicitation capability** by computing accuracy on closed-book QAs synthesized from the oracle facts. Figure 3(b) shows the procedure. A QA pair is synthesized by paraphrasing a given oracle fact so that one noun phrase in the fact serves as the answer. Table 1 shows an example of a QA pair.

4 Experiment

4.1 Experimental Setting

4.1.1 Training Data

Table 2 shows our training data. As for CPT, We used (i) Japanese JP1 documents and (ii) a randomly sampled subset of llm-jp-corpus-v2 (LLM-jp, 2024), a Japanese corpus that covers diverse sources. Our SFT data is QAs synthesized from sampled JP1 manuals by *gpt-4-1106-preview*. Figure 6 shows the prompt for synthesis.

4.1.2 Evaluation Setting

Real-world QA Data We evaluate mDAPT on ten real-world technical support questions. They

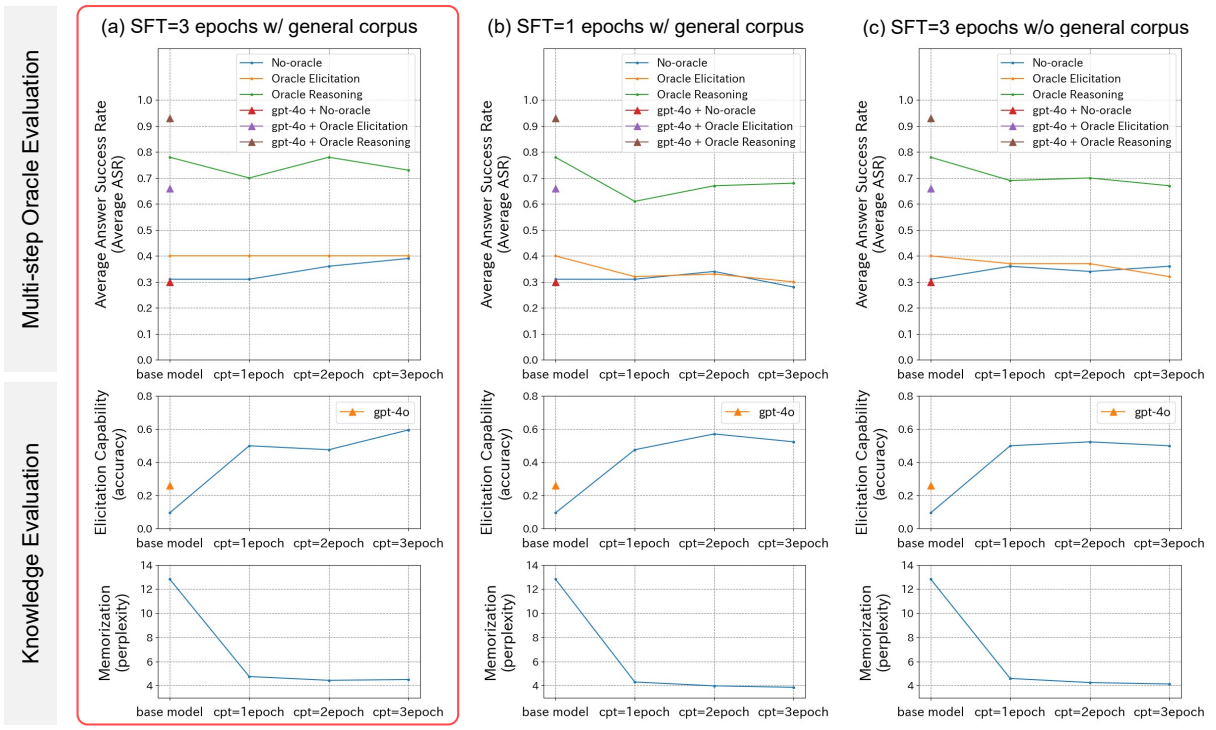

Figure 4: Evaluation results. The main results on the far-left show that memorization and elicitation capabilities improve as CPT epochs increase. At the third epoch, the ASR of the mDAPT model on the no-oracle setting reach the base model's ASR on the oracle elicitation setting, indicating that mDAPT resolves the elicitation task.

contain JP1 customers' questions which were too difficult for service desk personnel to handle and have to be escalated to JP1 experts.

Multi-step Oracle Evaluation Data The checklists for LLM-as-a-judge were created through interviews with a JP1 expert. Oracle conclusions were created by writing JP1-related facts based on the checklists. Oracle facts were created by extracting texts that support the oracle conclusions from JP1 manuals. We show more details in Appendix B.

LLM-as-a-judge Setting For each prompt setting, we generate multiple answers with ten different random seeds at a temperature of 0.7. After that, we compute the rate of answers judged as correct by our LLM-as-a-judge (**Answer Success Rate**; **ASR**). We used Qwen2.5-72B-Instruct (Qwen et al., 2025) as the evaluator. In a preliminary evaluation, there were only three cases where our LLM-as-a-judge contradicted a JP1 expert's evaluation among 100 generated answers.

Knowledge Evaluation We synthesized QA pairs with GPT-4o (OpenAI et al., 2024) from the oracle facts. To evaluate the elicitation capability, we generate answers by greedy decoding and compute accuracy based on exact matches.

4.1.3 Models

We performed mDAPT on Qwen2.5-72B-Instruct, which has high Japanese capability. For comparison, we evaluated GPT-4o, which is expected to have higher composing and reasoning capabilities.

4.1.4 mDAPT setting

We implemented CPT and SFT with Hugging Face[2] and TRL[3] libraries. On both CPT and SFT, learning rates were 1.0e-5, and global batch sizes were 120. Table 5 presents other training hyper-parameters. Optimizer was AdamW (Loshchilov and Hutter, 2019). We used DeepSpeed ZeRO-3 (Rajbhandari et al., 2020) and H100 GPU $\times$ 24 for training.

4.2 Main Result

Figure 4(a) shows the main result. As the number of CPT epochs increases, both memorization and elicitation capabilities improve. This shows that mDAPT encourages the LLM to memorize necessary knowledge and to access it when it is directly asked. In the multi-step oracle evaluation, ASR on the no-oracle setting improves as the number of CPT epochs increases. At the third epoch when the elicitation capability peaked, the no-oracle setting's

[2]https://pypi.org/project/transformers/
[3]https://pypi.org/project/trl/

ASR matched that of the base model on the oracle elicitation setting. This indicates that mDAPT resolves the base model's bottleneck caused by the elicitation task.

The mDAPT model's ASRs significantly improved on the oracle reasoning setting compared to the no-oracle setting. However, there is still a gap to 100%. This means that the mDAPT model struggles with both the reasoning and composing tasks. Thus, mDAPT cannot contribute to the reasoning and composing tasks although it is effective in the elicitation task. Such bottlenecks limit mDAPT's effectiveness in real-world operations.

Model	# QAs with each score			
	S1[1]	S2[2]	S3[3]	S4[4]
(a) Evaluation on 20 QAs				
Qwen2.5-72B-Instruct	13	6	1	0
+ RAG	13	5	2	0
GPT-4o	12	6	2	0
mDAPT model	10	9	1	0
(b) Evaluation on 10 QAs used in Figure 4's experiment				
mDAPT model	4	6	0	0
+ Oracle reasoning	1	5	1	3

[1] A given answer is **not useful**.
[2] A given answer contains misinformation but is **useful**.
[3] A given answer is **very useful** but needs some modifications.
[4] A given answer **can be adopted** for the actual answer as is.

Table 3: Human Evaluation by an expert

4.3 Discussion

A Key condition for achieving sufficient performance is to resolve the elicitation and reasoning tasks: As shown in Figure 4, GPT-4o's ASR improves with the oracle elicitation and reasoning, revealing bottlenecks in the elicitation and reasoning tasks. Moreover, the ASR on the oracle reasoning setting achieves sufficient performance (over 90%). This empirically reveals that resolving both tasks is a necessary and sufficient condition for achieving a usable LLM in real-world operations if the base model is a stronger model. Given that mDAPT can resolve the elicitation task, exploring how to enhance reasoning capability to resolve the reasoning task is a promising research direction.

SFT and using general-domain corpora are essential for preserving composing capability: To investigate how SFT affects performance, we changed SFT epochs from 3 (Figure 4(a)) to 1 (Figure 4(b)). The perplexity for oracle facts in Figure 4(b) is better than that in Figure 4(a) across all CPT epochs. Meanwhile, the highest elicitation capability in Figure 4(b) is slightly worse than that in Figure 4(a). This suggests that SFT encourages models to generalize from memorizing knowledge to leveraging it, consistent with a previous study (Jiang et al., 2024). The ASRs for the oracle reasoning results in Figure 4(b) are lower than those in Figure 4(a) across all CPT epochs. This indicates that SFT is beneficial for preserving the base model's composing capability required for the composing task.

Figure 4(c) shows evaluation results obtained without using llm-jp-corpus-v2 during training. The perplexity for oracle facts in Figure 4(c) is slightly better than that in Figure 4(a) across 2nd or 3rd CPT epochs. This indicates that incorporating general-domain corpora into the training data slightly hinders memorization of JP1 facts. On the other hand, the ASRs in Figure 4(c) on the oracle reasoning setting are lower than those in Figure 4(a) over all CPT epochs. Thus, using general-domain mitigates catastrophic forgetting of composing capability, which ensures performance over an entire answering process, and this positive effect surpasses the negative effect on memorization degradation.

Human Evaluation We also asked JP1 experts to evaluate LLMs. Since human evaluation is time-consuming, we selected the mDAPT model with 3 CPT epochs (Figure 4(a)), which achieved the highest ASR on the no-oracle setting. We used twenty technical support questions, including the ones described in Section 4.1.2. We asked experts to categorize the LLMs' answers into four scores based on the usefulness for writing actual answers.

Table 3(a) shows the result. The mDAPT model outperformed RAG (Lewis et al., 2020; Gao et al., 2024) and GPT-4o on the number of useful answers (scores of S2 or higher). This indicates that mDAPT is useful in real-world operations to some extent. We further focused on the ten QAs used in the multi-step oracle evaluation and asked experts to evaluate the answers generated in the no-oracle and oracle reasoning settings. As shown in Table 3(b), scores in the oracle reasoning setting are distributed at a higher position than those in the no-oracle setting. This emphasizes that the reasoning task is a bottleneck, consistent with Section 4.2.

5 Conclusion

We evaluated mDAPT in real-world operations that require micro domain knowledge. mDAPT can re-

solve the elicitation task, but cannot resolve other tasks, which limits mDAPT's effectiveness. Further analysis revealed that sufficient performance can be achieved by additionally resolving the reasoning task. Applying mDAPT to pretrained reasoning models while somehow avoiding catastrophic forgetting is one direction to addressing this challenge, yet such methods are not sufficiently explored.

Limitations

In this paper, we clarified both the potential and the bottlenecks of mDAPT through careful multi-perspective evaluation using real-world data from professional operations. While our evaluation framework is language-independent and can be applied to a wide range of tasks, we evaluated LLMs only on technical support QA data. However, we believe that conducting and sharing an in-depth evaluation on proprietary real-world data has substantial value even when the results come from a single domain. In proprietary business domains, it is often difficult to release data publicly, which makes it challenging to accumulate quantitative findings regarding the effectiveness of LLMs in real-world operations. Therefore, we consider it important to accumulate quantitative findings even from a single domain, and our paper offers valuable and meaningful results in this respect. By sharing both our findings and the evaluation framework, we hope to enable other organizations to conduct similar analyses in different business domains, thereby accelerating the accumulation of findings and insights about challenges related to LLM deployment across industry and academia.

The main goal of this study is to establish a framework for clarifying bottlenecks of current mDAPT, and many DAPT attempts use non-reasoning models. Therefore, we adopted Qwen2.5-72B-Instruct, which is a non-reasoning model with top-level Japanese capability, as a base model. However, evaluating mDAPT on more recent non-reasoning models or reasoning models would be an interesting direction for future work. In the latter direction, we should avoid catastrophic forgetting when applying mDAPT to reasoning models. However, reasoning models are generally constructed through specialized training to enhance reasoning capability (DeepSeek-AI et al., 2025; Muennighoff et al., 2025; Liu et al., 2025a), and mDAPT on such reasoning models keeping reasoning capability is still under-explored.

Moreover, many approaches for training reasoning models focus on developing logical thinking for mathematical problems, rather than on reasoning capability over proprietary knowledge. Thus, whether such models' reasoning capabilities resolve our defined reasoning sub-task is not trivial. Training methods for the latter purpose are still under development in large proprietary domains (Huang et al., 2025; Wu et al., 2025; Liu et al., 2025b). If well-established methods become available, it would be desirable to apply such training method instead of mDAPT to available LLMs and evaluate the trained models using our framework.

To evaluate standard mDAPT approach, we fine-tuned all parameters of LLMs by CPT and SFT procedures. This requires large machine resources, H100 GPU $\times$ 24 for both CPT and SFT in our experiments, and makes practical adoption difficult. Thus, we further would like to verify light-weight training methods such as LoRA (Hu et al., 2022) in future work.

Acknowledgments

We would like to thank anonymous reviewers and Kana Ozaki for their valuable feedback. We would also like to thank Dr. Masaaki Shimizu for the maintenance and management of large computational resources. Additionally, we would also like to thank Hitachi Solutions, Ltd. for their support of this research.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. 2025. How do large language models acquire factual knowledge during pretraining? In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS 2024)*, pages 60626–60668.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024a. Instruction pre-training: Language models are supervised multitask learners. In *Proceedings of the 2024 Conference on*

Empirical Methods in Natural Language Processing, pages 2529–2550.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2024b. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. 2024. SaulLM-7B: A pioneering large language model for law. *Computing Research Repository*, arXiv:2403.03883.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *Computing Research Repository*, arXiv:2501.12948.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Computing Research Repository*, arXiv:2312.10997.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. *Computing Research Repository*, arXiv:2306.11644.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Hitachi. 2025a. Integrated operations management JP1. Accessed at 2025/07/30.

Hitachi. 2025b. Manuals - Middleware - JP1. Accessed at 2025/07/30.

Hitachi. 2025c. Manuals - Middleware - JP1 - Job scheduler : JP1/Automatic Job Management System 3. Accessed at 2025/10/19.

Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*.

Xiaoke Huang, Juncheng Wu, Hui Liu, Xianfeng Tang, and Yuyin Zhou. 2025. m1: Unleash the potential of test-time scaling for medical reasoning with large language models. *Computing Research Repository*, arXiv:2504.00869.

Zhengbao Jiang, Zhiqing Sun, Weijia Shi, Pedro Rodriguez, Chunting Zhou, Graham Neubig, Xi Lin, Wen-tau Yih, and Srini Iyer. 2024. Instruction-tuned language models are better knowledge learners. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5421–5434.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, Tianjiao Zhao, Amit Panalkar, Wei Cheng, Haoyu Wang, Yanchi Liu, Zhengzhang Chen, Haifeng Chen, Chris White, Quanquan Gu, and 2 others. 2023. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *Computing Research Repository*, arXiv:2305.18703.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025a. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. *Computing Research Repository*, arXiv:2505.24864.

Zhaowei Liu, Xin Guo, Fangqi Lou, Lingfeng Zeng, Jinyi Niu, Zixuan Wang, Jiajie Xu, Weige Cai, Ziwei Yang, Xueqian Zhao, Chao Li, Sheng Xu, Dezhi Chen, Yun Chen, Zuo Bai, and Liwen Zhang. 2025b. Fin-R1: A large language model for financial reasoning through reinforcement learning. *Computing Research Repository*, arXiv:2503.16252.

LLM-jp. 2024. LLM-jp corpus v2. Accessed at 2025/11/13.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Computing Research Repository*, arXiv:2501.19393.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. GPT-4o system card. *Computing Research Repository*, arXiv:2410.21276.

OpenAI. 2023. GPT-4 technical report. *Computing Research Repository*, arXiv:2303.08774.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Computing Research Repository*, arXiv:2412.15115.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. ZeRO: Memory optimizations toward training trillion parameter models. *Computing Research Repository*, arXiv:1910.02054.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, and 12 others. 2023. Towards expert-level medical question answering with large language models. *Computing Research Repository*, arXiv:2305.09617.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. ♫ MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Juncheng Wu, Wenlong Deng, Xingxuan Li, Sheng Liu, Taomian Mi, Yifan Peng, Ziyang Xu, Yi Liu, Hyunjin Cho, Chang-In Choi, Yihan Cao, Hui Ren, Xiang Li, Xiaoxiao Li, and Yuyin Zhou. 2025. MedReason: Eliciting factual medical reasoning steps in llms via knowledge graphs. *Computing Research Repository*, arXiv:2504.00993.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A large language model for finance. *Computing Research Repository*, arXiv:2303.17564.

Yawen Xue, Masaya Tsunokake, Yuta Koreeda, Ekant Muljibhai Amin, Takashi Sumiyoshi, and Yasuhiro Sogawa. 2025. Agent fine-tuning through distillation for domain-specific LLMs in microdomains. *Computing Research Repository*, arXiv:2510.00482.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K. Qiu, and Lili Qiu. 2024. Retrieval augmented generation (RAG) and beyond: A comprehensive survey on how to make your LLMs use external data more wisely. *Computing Research Repository*, arXiv:2409.14924.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks.*, pages 46595–46623.

Ingo Ziegler, Abdullatif Köksal, Desmond Elliott, and Hinrich Schütze. 2024. CRAFT your dataset: Task-specific synthetic dataset generation through corpus retrieval and augmentation. *Computing Research Repository*, arXiv:2409.02098.

A Detailed Training Settings

Table 4 provides a detailed breakdown of the training data used for CPT. JP1 documents, our micro domain-specific corpora, consist of JP1 manuals (Hitachi, 2025b), JP1 release notes, and other JP1 references in Japanese. All JP1 documents were created by converting original PDF files to texts by PDFminer[4].

As for SFT data, Figure 6 shows a prompt used for synthesizing JP1-QA, our SFT data, from JP1 manuals.

We sequentially performed CPT and SFT in mDAPT. Table 5 shows detailed training hyperparameters. We used H100 GPU $\times$ 24 for both CPT and SFT. The mDAPT model on CPT three epochs shown in the left side of Figure 4, which has the highest ASR on the no-oracle setting, was trained by 80 hours.

B Evaluation Data

Real-world QA Data QA pairs used in our experiments are from real-world technical support operations. However, personal information has been manually anonymized, and e-mail histories leading up to questions have been manually removed.

Multi-step Oracle Evaluation Data The checklists for LLM-as-a-judge were constructed through interviews with JP1 experts, who actually work for technical support operations. The average number of checklists per QA pair is 1.8. The average number of conditions per checklist is 2.6.

Oracle conclusions were created by writing JP1-related facts based on the checklists' conditions. Some of them constitute ground-truth answers, thus, LLMs can generate correct answers given oracle conclusions if the LLMs have sufficient reading and generation capabilities. In a pilot study, *gpt-4-0613* generated correct answers for seven out of ten QA pairs by inserting the oracle conclusions into the prompts. We found that the remaining three QA pairs need not only factual knowledge, but also domain-specific thinking patterns based on factual knowledge to generate correct answers. Thus, we appended guidance information (how to make plausible explanation, where to emphasize, etc.), described in Section 3.2, to the oracle conclusions.

Oracle facts were created by searching rationale texts for oracle conclusions from JP1 manuals and extracting them. Texts extracted from the same section were unified to one oracle fact by concatenating the texts. After that, we appended section titles to a part of oracle facts. There are 4.6 oracle facts per QA pair. A JP1 expert evaluated all created oracle facts as being relevant to questions and 74% of oracle facts as being mandatory knowledge for writing answers.

Knowledge Evaluation For evaluating memorization, we computed perplexity for each paragraph in the oracle facts and report the average values. For evaluating elicitation capability, we synthesized closed-book QA pairs from the oracle facts by GPT-4o. Figure 5 shows a prompt used for synthesizing. The example of "### Document" in Figure 5 is cited from https://itpfdoc.hitachi.co.jp/manuals/3021/30213L4210e/AJSF0060.HTM. After synthesizing by this prompt, we manually corrected or deleted synthesized questions that have multiple possible answers. Consequently, 42 QAs were created, and we used them in experiments.

C Evaluation Details

When generating answers for technical support questions, we simply asked LLMs to answer given questions in prompts. On the oracle reasoning setting, we insert "## Background Knowledge" and itemized oracle conclusions into the prompts before the questions. If the oracle conclusions include guidance information that specifies LLMs' reasoning direction, we additionally insert texts shown in Figure 7 after the questions. On the oracle elicitation setting, we insert "## Background Knowledge" and oracle facts concatenated with "#### Knowledge *i*," where *i* is an index of each oracle fact. We actually used Japanese prompts in the experiments, but we also show their English translation for explanation purposes in this paper.

Our LLM-as-a-judge focuses on the correctness of generated answers. Ideally, a correct answer should cover all required information, and it does not contain misinformation or irrelevant content. Nevertheless, it is difficult to determine the latter as there is an immense variety of patterns. Accordingly, we focus on the former; evaluating whether generated answers satisfy essential conditions for correctness. Figure 8 shows a prompt used for LLM-as-a-judge. If an evaluator LLM outputs "Yes" or "yes" for this prompt, the generated answer is judged to satisfy a given condition. This

[4] https://pypi.org/project/pdfminer/20191125/

Name	Usage	Description	# Documents	Size (MB)	# Tokens (M)
JP1 Manuals (Hitachi, 2025b)	CPT	Text converted from PDF manuals covering JP1 Version 13.	105	56.8	14.3
JP1 Release Notes	CPT	Text converted from PDF release notes of JP1.	42	14.9	3.9
Other JP1 Docs	CPT	Other references for managing JP1	46	0.4	0.1
llm-jp-corpus-v2	CPT	Subset of llm-jp-corpus-v2, which contains various Japanese texts	-	651.4	-

Table 4: Detailed list of training data used for CPT

Parameter	Value	
	CPT	SFT
Global batch size	120	120
Learning rate	1.0e-5	1.0e-5
Weight decay	0.1	0.1
Adam β 1	0.9	0.9
Adam β 2	0.95	0.95
Optimizing Scheduler	constant	-
Warm up ratio	-	-
Max sequence length	2048	4096

Table 5: Training hyper-parameters

judgment process is applied to all conditions in all prepared checklists.

In the knowledge evaluation, we evaluated LLM's elicitation capability by a prompt shown in Figure 9. Generated answers are first normalized. Specifically, we split each generated text by newline character and normalize the first part of split texts by removing punctuation from it. After normalization, we computed accuracy based on whether normalized answers exactly match the ground-truth answers.

```
以下の文書はソフトウェア製品のマニュアルです。\n
次の手順でこの文書に関する質問、回答、回答の根拠を日本語で作成してください。\n
\n
1. この製品を使うユーザーになりきって、Question:という文字列の後に、与えられた文書に基づいて、実際に起きたトラブルや聞きたいこと、分からないことなどを質問として作成して下さい。その際、トラブルに加えて、やりたいことや、状況説明、トラブルシューティングに役立つ情報(例えば、実行環境、バージョン、実行コマンド、エラー詳細)なども記載するといいかもしれません。\n
2. 次に、この製品を扱う会社の一流のコンタクトセンターの職員になりきって、Answer:という文字列の後に、文書に基づいて初心者のユーザーの質問に丁寧に答えて下さい。\n
3. 最後に、Citation:という文字列の後に、回答の根拠となった文書内の記述をそのままコピー&ペーストして書いてください(変更は加えないで抜き出してください)。\n
\n
注意点として、質問や回答を複数個つくったりしないでください。また、Question:, Answer:, Citation:以外のフォーマットを使わないでください。\n
\n
では、質問を以下の文書の情報を基に日本語で作成してください。\n
文書:{chunk}
```

```
(A prompt translated into English)
The following document is a manual for a software product.\n
Please create a question about this document, the answer, and the rational of the answer in Japanese according to the steps below.\n
\n
1. Pretend to be a user of this product and, after the string Question:, create a question based on the provided document, such as an actual problem that occurred, something you want to ask, or something you don't understand. In addition to the problem itself, it may be better to include what you are trying to do, a description of the situation, and information useful for troubleshooting (for example, the execution environment, version, execution command, error details), etc.\n
2. Next, assume the role of an elite contact center employee of the company that handles this product, and after the string Answer:, politely answer the beginner user's question based on the document.\n
3. Finally, after the string Citation:, copy and paste the exact passage from the document that forms the basis of your answer (do not make any changes; extract it as is).\n
\n
Do not create multiple questions or answers. Also, do not use any format other than Question:, Answer:, and Citation:.\n
\n
Now, please create the question in Japanese based on the information in the document below.\n
Document:{chunk}
```

Figure 5: Prompt used for synthesizing JP1-QA. Chunks extracted from JP1 manuals are inserted to *{chunk}*. We use the Japanese prompt in our experiments.

以下の「### Document」には、ITシステムを制御・管理するためのミドルウェアであるJP1に関する説明が記載されています。\n
「### Document」の文章を言い換えることで質問と回答のペアを1つ作ってください。\n
\n
ただし、必ず下記の条件を守ってください。\n
・回答が必ず「### Document」の文章に記載されている1つの単語となる\n
・「### 専門文書」の内容に基づいて、必ず回答可能な質問である\n
・回答が一意に定まる\n
・質問は「### Question」の後に続けて書く\n
・回答は「### Answer」の後に続けて書く\n
\n
下記は作成例です。 \n
Document\n
(2)␣計画実行登録\n
計画実行登録は，ジョブネットのスケジュール定義やジョブネットが属するジョブグループのカレンダー情報に基づいて実行予定をスケジュールします。計画実行登録の場合，実行登録後は初回のジョブネットの実行予定だけが確定されたスケジュールで，それ以降のスケジュールは擬似予定（シミュレーションされたスケジュール）という扱いになります。擬似予定については，「4.4.2(1) スケジュールシミュレーション」を参照してください。次回の実行予定は，前回の実行予定のジョブネットが開始された時点でスケジュール確定します。\n
\n
Question\n
ジョブネットのスケジュール定義やジョブネットが属するジョブグループのカレンダー情報に基づいて実行予定をスケジュールする方法は何と言いますか？\n
\n
Answer\n
計画実行登録\n
\n
\n
それでは、下記の「### Document」に対して上述の条件を守って、質問と回答をつくってください。絶対に質問と回答以外を出力してはいけません。解説も不要です。\n
\n
Document\n
{fact}\n
\n

(A prompt translated into English)
The following “### Document” contains an explanation of JP1, middleware for controlling and managing IT systems.\n
Create one question–answer pair by paraphrasing the sentences in “### Document.”\n
\n
Be sure to observe the following conditions.\n
・The answer must be a single word that appears in the sentences of “### Document.”\n
・The question must be answerable based on the content of “### Document.”\n
・The question must have only one answer.\n
・Write the question immediately after “### Question.”\n
・Write the answer immediately after “### Answer.”\n
\n
Below is an example.\n
\n
Document\n
(2)␣Planned execution\n
In planned execution, the jobnet is scheduled based on its schedule definition and the calendar information set for the job group to which the jobnet belongs.\n
When you register a jobnet for planned execution, only its first execution is fixed, and subsequent executions are treated as dummy runs (simulated schedules). For details on dummy runs, see 4.4.2(1) Schedule simulation. The jobnet’s next run is finalized when the current run starts.\n
\n
Question\n
What is the execution method called that sets schedules for a jobnet based on the jobnet’ s schedule definitions and the calendar information of the job group to which the jobnet belongs?\n
\n
Answer\n
Planned execution\n
\n
\n
Now, based on the above conditions, create one question–answer pair for the “### Document” below. You must output nothing other than the question and answer. No explanation is needed.\n
\n
Document\n
{fact}\n
\n

Figure 6: Prompt used for synthesizing closed-book QAs used in the knowledge evaluation. The oracle facts are inserted to *{fact}*. We use the Japanese prompt in our experiments.

回答戦略について\n
上記の「### 質問文」に回答するには以下の回答戦略に遵守する必要があります。\n
{conclusions}\n
それでは「### 質問文」への回答を作成してください。\n
\n

(A text translated into English)
About the Answer Strategy\n
To answer the above "### Question," you must adhere to the following answer strategy.\n
{conclusions}\n
\n
Now, please create your answer to the "### Question."\n

Figure 7: Texts inserted into prompts when using oracle conclusions that specify the reasoning direction. We itemize such oracle conclusions and fill *{conclusions}* with them. After that, we insert this filled texts into prompts. We use the Japanese text in our experiments.

「## 評価対象」に書かれている文章が「## 評価基準」で示す要件を満たしているかを判断してください。\n
ただし、書かれていない内容の推測はせず、文章内の文字列に基づき客観的に「## 評価基準」と照らし合わせてください。\n
判断結果を「## 回答」に続けて、YesかNoで回答してください。絶対にYesかNo以外で回答してはいけません。\n
以上の指示に厳密に正確に従ってください。\n
\n
評価対象\n
{generated_answer}\n
\n
評価基準\n
{criteria}\n
\n
回答\n

(A prompt translated into English)
Determine whether the text written under "## Evaluation Target" satisfies the requirements indicated under "## Evaluation Criteria."\n
However, do not infer any unstated content; instead, objectively compare the strings in the text with the "## Evaluation Criteria."\n
Write your judgment result immediately after "## Answer" and respond with either Yes or No. You must not answer with anything other than Yes or No.\n
Follow the above instructions strictly and precisely.\n
\n
Evaluation Target\n
{generated_answer}\n
\n
Evaluation Criteria\n
{criteria}\n
\n
Answer\n

Figure 8: Prompt used for LLM-as-a-judge. We insert a generated answer into *{generated_answer}* and each condition of checklists into *{criteria}*. We use the Japanese prompt in our experiments.

統合システム運用管理向けソフトウェア・ミドルウェア製品であるJP1に関する一問一答形式の質問が「### Question」に記載されています。\n
「### Answer」に続けて、質問の答えとなる名詞を回答してください。\n
\n
ただし、絶対に質問の答えとなる名詞のみを簡潔に回答してください。それ以外は回答してはいけません。解説も不要です。\n
下記は回答例です。\n
\n
Question\n
JP1/IM - ManagerがJP1/Base（イベントサービス）からJP1イベントを取得する際の条件を設定するフィルターを何といいますか？\n
\n
Answer\n
イベント取得フィルター\n
\n
Question\n
JP1製品のうち、複数の業務の内容と実行順序を定義することで定型的・定期的な業務を自動化することを目的とした製品は何ですか？\n
\n
Answer\n
JP1/Automatic Job Management System 3\n
\n
それでは、下記の「### Question」に回答してください。\n
\n
\n
Question\n
{question}\n

(A prompt translated into English)
A question in a Q&A format about JP1, an integrated system operations management middleware product, is provided under "### Question."\n
Please write the noun that answers the question immediately after "### Answer."\n
\n
Be sure to answer concisely with only the noun that serves as the answer to the question. Do not output anything else. No explanation is needed.\n
Below is an example.\n
\n
Question\n
What is the filter called that specifies the conditions under which JP1/IM – Manager acquires JP1 events from JP1/Base (event service)?\n
\n
Answer\n
Event acquisition filter\n
\n
Question\n
Among JP1 products, which product is designed to automate routine and periodic tasks by defining the contents and execution order of multiple tasks?\n
\n
Answer\n
JP1/Automatic Job Management System 3\n
\n
Now, please answer the "### Question" below.\n
\n
Question\n
{question}\n

Figure 9: Prompt used for evaluating the elicitation capability in our knowledge evaluation. We insert synthesized a closed-book QA into *{question}*. We use the Japanese prompt in our experiments.