AQuilt: Weaving Logic and Self-Inspection into Low-Cost, High-Relevance Data Synthesis for Specialist LLMs

Xiaopeng Ke[1] **Hexuan Deng**[1] **Xuebo Liu**[1*] **Jun Rao**[1]
Zhenxi Song[2] **Jun Yu**[2] **Min Zhang**[1]
[1]Institute of Computing and Intelligence, Harbin Institute of Technology, Shenzhen
[2]School of Intelligence Science and Engineering, Harbin Institute of Technology, Shenzhen
{xiaopk7, hxuandeng, rao7jun}@gmail.com
{liuxuebo,songzhenxi,yujun,zhangmin2021}@hit.edu.cn

Abstract

Despite the impressive performance of large language models (LLMs) in general domains, they often underperform in specialized domains. Existing approaches typically rely on data synthesis methods and yield promising results by using unlabeled data to capture domain-specific features. However, these methods either incur high computational costs or suffer from performance limitations, while also demonstrating insufficient generalization across different tasks. To address these challenges, we propose **AQuilt**, a framework for constructing instruction-tuning data for any specialized domains from corresponding unlabeled data, including **A**nswer, **Q**uestion, **U**nlabeled data, **I**nspection, **L**ogic, and **T**ask type. By incorporating logic and inspection, we encourage reasoning processes and self-inspection to enhance model performance. Moreover, customizable task instructions enable high-quality data generation for any task. As a result, we construct a dataset of 703k examples to train a powerful data synthesis model. Experiments show that AQuilt is comparable to DeepSeek-V3 while utilizing just 17% of the production cost. Further analysis demonstrates that our generated data exhibits higher relevance to downstream tasks. Source code, models, and scripts are available at https://github.com/Krueske/AQuilt.

1 Introduction

With the rapid development of large language models (LLMs), their general capabilities have demonstrated significant success (Dubey et al., 2024; Jaech et al., 2024; Yang et al., 2024b; Guo et al., 2025). However, performance in specialized domains like law and medicine remains constrained (Ganin et al., 2016; Wang and Sennrich, 2020; Rao et al., 2025). To enhance model performance in these fields, synthetic data has emerged as a promising solution, thanks to its high-quality outputs (Zhang et al., 2024; Abdin et al., 2024; Wang et al., 2024; Guan et al., 2025; Rao et al., 2024).

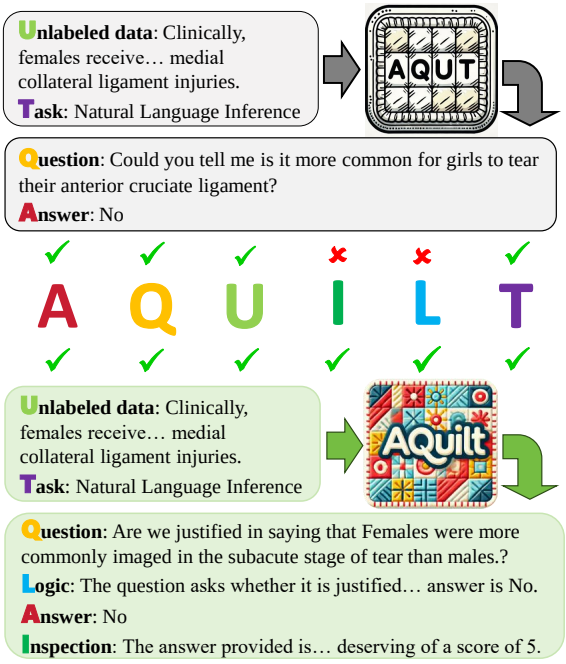

Figure 1: Traditional data synthesis models (top) can only generate question-answer pairs from unlabeled data, while AQuilt (bottom) additionally generates logic and inspection for the synthetic data.

Existing approaches rely on large models' domain priors for data synthesis. Yet, domain-specific knowledge and linguistic patterns, including lexical, syntactic, and stylistic characteristics, which are embedded in domain corpora rather than fully captured by model priors (Zhou et al., 2024b). This limitation motivates methodologies leveraging unlabeled data, which inherently encode domain-specific features (Hamilton et al., 2016; Mudinas et al., 2018). Recent advances demonstrate that unlabeled data-driven data synthesis enhances domain-specific data quality and task performance (Ziegler et al., 2024), supporting our focus on optimizing unlabeled data-driven methodologies.

* Corresponding Author

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 5764–5797
November 4-9, 2025

However, although some of the methods focus on domain-specific data synthesis using unlabeled data, they still have some problems. For example, current domain synthetic data generation methods often depend on powerful commercial models or large LLMs (Taori et al., 2023; Xu et al., 2024; Chen et al., 2024a). Though they perform well, these models are usually too expensive, restricting accessibility (Bansal et al., 2024). Using smaller, specialized models is an alternative (Zelikman et al., 2022; Chen et al., 2024b; Li et al., 2024c), but their covered tasks are limited, and the generation is too simple for complex tasks.

To address limitations, we propose **AQuilt**, a framework for constructing data that incorporates **A**nswer, **Q**uestion, **U**nlabeled data, **I**nspection, **L**ogic, and **T**ask type from any unlabeled data. We train a smaller data synthesis model to synthesize domain-specific instruct-tuning data and reduce synthesis costs. We introduce **L**ogic and **I**nspection to enhance model reasoning and ensure the quality of the synthesized data (Zelikman et al., 2022; Hosseini et al., 2024). Furthermore, **T**ask type is expanded to facilitate generalization to unseen tasks during training. We then synthesize a high-quality bilingual dataset (Chinese and English) containing 703k examples using DeepSeek-V3 (Liu et al., 2024), which is used to train a low-cost, high-relevance data synthesis model.

AQuilt demonstrates performance comparable to the distillation source model, DeepSeek-V3, across experiments involving two base models and five tasks, while requiring only 17% of the production cost. Furthermore, compared to previous specialized models for data synthesis, e.g., Bonito (Nayak et al., 2024), which performs well but is limited to generating data for English tasks requiring unlabeled data, our method demonstrates superior performance across these same tasks. Further analysis confirms the effectiveness of incorporating logic and inspection, as well as the higher relevance of our synthetic data to downstream tasks, which further contributes to the model's high performance.

Our contributions are as follows:

- We propose AQuilt, a framework for synthesizing high-relevance data for any task from any unlabeled dataset at a low cost. By incorporating logic and inspection, we enhance model reasoning and improve data quality.
- Experiment results show that AQuilt is comparable to DeepSeek-V3 with 17% of the production cost.
- Further analysis shows that logic and self-inspection contribute to better performance and more relevant generated data.
- We will publicly release our data synthesis model, training data, and code, contributing to developing more powerful specialized LLMs and data synthesis models.

2 Related Works

Domain Data Synthesis without Unlabeled Data. Recent works leverage the parametric knowledge of general LLMs for domain-specific data synthesis, avoiding domain unlabeled data (Bao et al., 2023; Deng et al., 2025a; Luo et al., 2025). Domain-oriented innovations include Zhou et al. (2024c) using Self-Instruct (Wang et al., 2023b) to synthesize legal question-answer pairs, and Li et al. (2024b) employing GPT-4 to generate scientific questions through knowledge distillation (Peng et al., 2023; Rao et al., 2023). Eldan and Li (2023) demonstrates constrained domain adaptation by generating children's stories with controlled vocabulary. While existing methods leverage strong LLMs to synthesize training data directly (Gilardi et al., 2023; Xie et al., 2024; Hwang et al., 2024), which mainly rely on preexisting domain knowledge of commercial LLMs (Achiam et al., 2023; Yang et al., 2023), their efficiency of domain data synthesis remains limited (Palepu et al., 2024). However, small models struggle to synthesize data with domain-specific knowledge without external input (Deng et al., 2024; Harbola and Purwar, 2025), spurring research on data synthesis by combining small models with domain-specific unlabeled data.

Domain Data Synthesis with Unlabeled Data. Thus, we focus on domain data synthesis based on unlabeled data, which better balances performance and efficiency. Recently, specialized methods have been proposed to integrate unlabeled data to address domain gaps (Bartz et al., 2022; Deng et al., 2023, 2025b; Upadhyay et al., 2025). For instance, Nayak et al. (2024) train models on datasets with unlabeled data (e.g., summary, reading comprehension) for task-specific synthesis. Ziegler et al. (2024) combines retrieval with in-context learning to generate data requiring specialized knowledge. Iterative refinement techniques, such as reinforced self-training (Dou et al., 2024) and back-translation

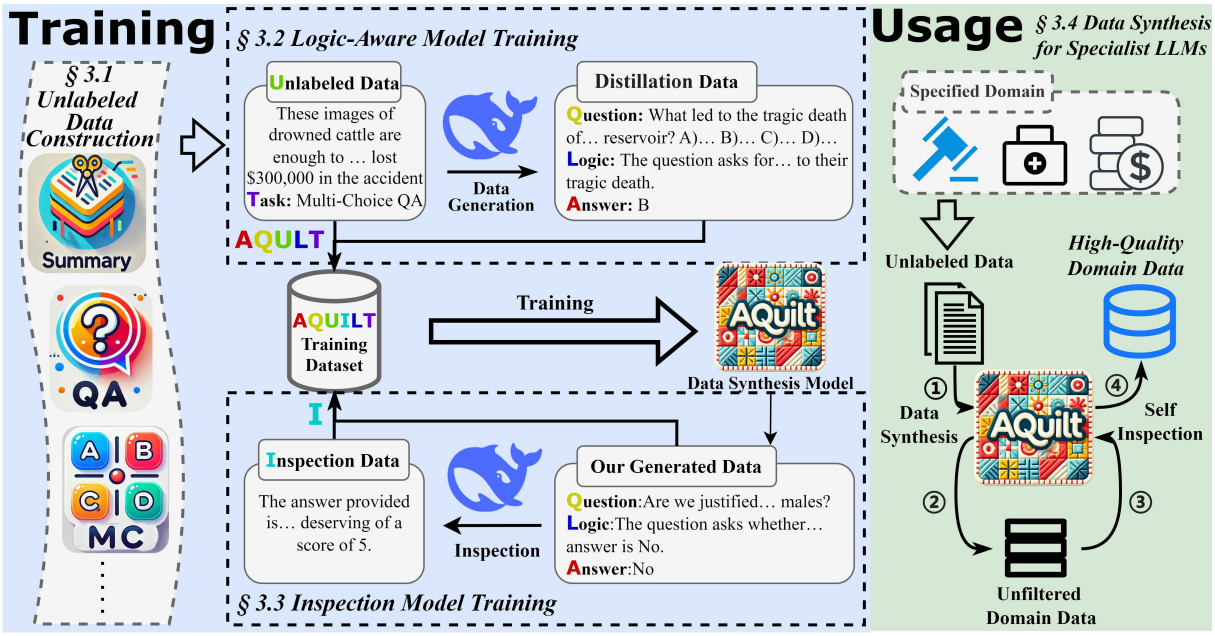

Figure 2: Overview of the proposed AQuilt framework. The left side illustrates the training process of our data synthesis model, while the right side demonstrates how the trained model automatically synthesizes high-quality domain-specific data. The synthesized data is subsequently used to train Specialist LLMs.

(Li et al., 2024c), further improve synthetic data using domain resources. However, existing solutions face two key challenges: (1) high costs and inefficiency when relying on commercial LLMs (Bansal et al., 2024), and (2) poor generalization of specialized models to out-of-distribution tasks. For example, models like Chen et al. (2024b), trained on GPT-generated seeds, lack the ability to define task types explicitly. These limitations underscore the urgent demand for frameworks capable of reconciling cost efficiency, domain specificity, and task generalization. In response, we present a 7B-parameter data synthesis model that integrates open-domain QA for task generalization while balancing efficiency and quality without expensive LLMs.

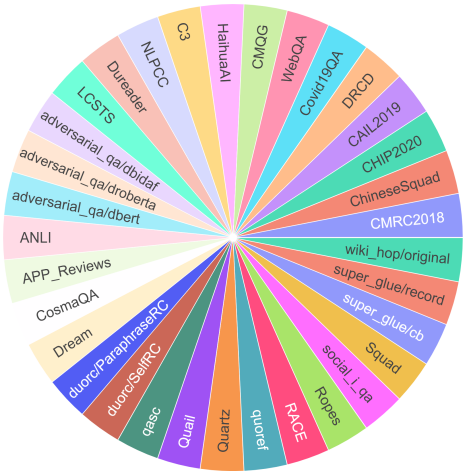

Figure 3: Overview of the sources of the unlabeled data we collect, which covers 33 different kinds of datasets, ensuring the diversity of the data we construct.

3 Our Proposed AQuilt Framework

To develop a low-cost, high-relevance data synthesis model with cross-task generalization capabilities for multiple domains, we propose the framework illustrated in Figure 2. It starts with building a large source data corpus (§3.1). Next, we distill **AQULT** quintuplets to enhance the model's data synthesis ability (§3.2). We then evaluate the generated data to obtain **I**nspection data, which further improves the model's self-inspection capability (§3.3). Finally, using this synthesized data, we train a robust data synthesis model, **AQuilt**, which is applied to generate high-quality domain-specific data for training specialist LLMs (§3.4).

3.1 Unlabeled Data Construction

Task Definition. To enable our data synthesis model to synthesize various types of tasks, we follow Nayak et al. (2024) and cover a wide range of tasks, including extractive QA, natural language inference, multi-choice QA (single-answer / multi-answer), text generation, text summarization, text classification, and natural language understanding. Furthermore, to enhance generalization across downstream tasks, we introduce two additional

task types: **open-book QA** and **closed-book QA**. Because of the customizable questions, these two tasks are not confined to specific categories. When synthesizing data for a new task type, we designate it as either closed-book QA or open-book QA, depending on whether it requires unlabeled data as input. Additionally, we prepend the instruction of the new task as a prefix to the question. This effectively enhances the data synthesis generalization for novel domain-specific task types.

Data Type. To facilitate multi-domain generalization in data synthesis, we aggregate diverse unlabeled data spanning 33 Chinese-English bilingual datasets covering news, encyclopedias, reviews, and multiple specialized domains, illustrated in Figure 3 and detailed in Appendix A.

3.2 Logic-Aware Model Training

Logic-Aware Data Generation. The incorporation of intermediate thought rationales has been shown to enhance LLM performance (Zelikman et al., 2022). Motivated by this, we incorporate the model's intermediate reasoning process, i.e., *logic*, into the data synthesis procedure, fostering a more structured reasoning process and improving the overall data quality. Specifically, for each task type t, we randomly associate it with unlabeled data u and employ a strong commercial LLM, i.e., DeepSeek-V3, to generate distilled data, including the question q, logic l, and answer a. Formally,

$$(a, q, l) = \texttt{LLM}_{\text{Strong}}^{\text{GenData}}(u, t). \tag{1}$$

Furthermore, we collect original datasets, including extractive question answering, natural language inference, multichoice question answering (single answer), and summarization datasets from labeled data to enhance the diversity of QA pairs and address LLMs' challenges in generating extractive QA data. We prompt the model to supplement the missing logic l for these collected datasets. Formally:

$$l = \texttt{LLM}_{\text{Strong}}^{\text{GenLogic}}(a, q, u, t). \tag{2}$$

Collecting all synthesized data, we obtain the dataset $\mathcal{D}_L = \{(a, q, u, l, t)\}_N$, which comprehensively covers all the task types defined in AQuilt.

Relevence-Aware Data Filtering. Existing methods, e.g., Bonito, often heavily rely on unlabeled data u to generate (q, a) pairs, introducing the risk of synthesizing low-relevance data for certain tasks that do not depend on unlabeled data. In contrast, we ensure that our synthesized (q, a) pairs remain meaningful without u in multi-choice or closed-book QA tasks that usually do not require u as input. To achieve this, we explicitly guide the model's preferences through prompt engineering. We also filter out cases not meeting this criterion by identifying prohibited words, such as "the context" and "the text". This ensures that the generated questions are applicable to downstream tasks with or without unlabeled data.

Additionally, to mitigate potential biases in LLMs (Zhou et al., 2024a; Guo et al., 2024), we analyze word frequency statistics. For each task, we identify the most frequent words, excluding stopwords. If any word appears in more than 10% of the data, it may indicate stylistic bias. In such cases, we eliminate questions containing these keywords, reducing their prevalence and ensuring a diverse and unbiased final training set.

After filtering, we obtain the refined dataset $\mathcal{D}_L^{'} = \{(a, q, u, l, t)\}_{N'}$. The total dataset size is summarized in Table 1.

Model Training. Using the large-scale dataset $\mathcal{D}_L^{'}$ obtained above, we train the data synthesis model. To enable the model to synthesize task-specific data from unlabeled data, we use u and t as inputs and train the model to generate q, a, and l. Formally:

$$\mathcal{L}_{\text{AQuilt}} = -\sum_{\mathcal{D}_L^{'}}^{j} \log P_\theta(a_j, q_j, l_j \mid u_j, t_j). \tag{3}$$

Using the above loss function $\mathcal{L}$, we obtain the data synthesis model, $\texttt{LLM}_{\text{AQuilt}}$.

3.3 Inspection Model Training

The above model is capable of generating data for any task from domain-specific texts. However, the generated data may still be low-quality in some situations. To mitigate this, we train the model to acquire self-inspection capabilities.

Inspection Data Generation. To train the self-inspection capability, we need to collect training data with varying quality levels. However, since the data synthesized by strong commercial LLMs is generally of high quality, we utilize our previously trained LLM, $\texttt{LLM}_{\text{AQuilt}}$, to generate new data. This ensures that the synthesized data aligns with the distribution of our final generation process, which is beneficial for model training. Specifically, for each task t, we randomly sample u and input it into

Task Type	English	Chinese
Extractive QA	16k	16k
Natural Language Inference	49k	33k
Multi-Choice QA (Single Answer)	49k	49k
Multi-Choice QA (Multiple Answers)	28k	31k
Text Generation	33k	33k
Text Summarization	49k	43k
Text Classification	33k	33k
Natural Language Understanding	32k	32k
Open-Book QA	33k	31k
Closed-Book QA	33k	33k
Self-Inspection	7k	7k
Total	362k	341k

Table 1: The number of generated training data from different tasks. A total of 703k data is collected, covering both English and Chinese.

our trained model, $\texttt{LLM}_{\text{AQuilt}}$, to synthesize data. Subsequently, we use DeepSeek-V3 to score these samples. Formally,

$$\begin{aligned}(a', q', l') =& \texttt{LLM}_{\text{AQuilt}}(u, t),\\ i =& \texttt{LLM}^{\text{GenInsp}}_{\text{Strong}}(a', q', u, l', t).\end{aligned} \tag{4}$$

As a result, we obtain the dataset $\mathcal{D}'_I = \{(a', q', u, i, l', t)\}_M$ for training the self-inspection capability of AQuilt.

Self-Inspection Model Training. For training, we continue fine-tuning the previously trained model, $\texttt{LLM}_{\text{AQuilt}}$, by incorporating a LoRA adapter (Hu et al., 2022), formally denoted as $\texttt{LLM}^{\text{LoRA}}_{\text{AQuilt}}$. This modification enables the model to score its own instruction-tuning dataset (a', q', l'), which is generated based on (u, t). Formally, we optimize the LoRA-augmented model using the following loss function:

$$\mathcal{L}^{\text{LoRA}}_{\text{AQuilt}} = -\sum_{\mathcal{D}'_I}^{j} \log P_{\theta_{\text{LoRA}}}(i_j \mid a'_j, q'_j, u_j, l'_j, t_j). \tag{5}$$

3.4 Data Synthesis for Specialist LLMs

We use the constructed AQuilt model to generate high-quality domain-specific instruct-tuning data from unlabeled data. Then, we could train the specialist model on the generated data to enhance its domain-specific task performance.

Domain Data Synthesis. When conducting downstream task learning for LLMs, we synthesize high-quality domain-specific data using only the specified task type t and the relevant domain unlabeled data u. Leveraging our data synthesis model, we can efficiently generate training data tailored to the given domain and task. Formally,

$$(a', q', l') = \texttt{LLM}_{\text{AQuilt}}(u, t). \tag{6}$$

Notably, if the task type t has not been observed in the training set, we designate t as either *closed-book QA* or *open-book QA*, depending on whether it requires the unlabeled data as input. Additionally, we prepend the instruction of the new task as a prefix to the question.

Data Self-Inspection. To ensure the quality of the generated data, we apply filtering based on self-inspection. Specifically, we first generate an inspection score using the trained model:

$$i' = \texttt{LLM}^{\text{LoRA}}_{\text{AQuilt}}(a', q', u, l', t). \tag{7}$$

Subsequently, we filter out low-quality data. By default, we remove data with an inspection score of 2 or lower (on a 5-point scale). If more than 20% of the data receives a score of 2, which suggests that the task is inherently simpler, we only remove data with a score of 1. As a result, we obtain high-quality training data, facilitating efficient adaptation of the model to specific domains.

Training Specialist LLM We train the target model on high-quality domain-specific data synthesized by AQuilt model to enhance its performance on domain-specific tasks.

4 Experiment

4.1 Training Setup of AQuilt

Data. As summarized in Table 1, we construct a bilingual data set (EN / ZH) that covers 10 types of tasks, with the aim of improving model generalization through diversified coverage. To prevent task dominance and ensure balanced distributions, we apply downsampling: logic training data are capped at 50k samples per (task, language) pair, while self-inspection data contain no more than 2k samples per score for each language, with scores ranging from 1 to 5. The final aggregated dataset comprises 703k samples, with full prompts detailed in Appendix D.

Training. The experiments are conducted on Qwen2.5-7B-Base with 8 NVIDIA 4090 24GB GPUs. For both training procedures described above, we use the AdamW optimizer, with a learning rate of 1e-4, batch size of 32, and LoRA r and alpha both set to 64, training for 2 epochs.

Model	Source	SquadQA		PubMedQA		CEVAL		Translation		EssayQA		Avg.	
		Score	Cost	Score	Cost	Score	Cost	Score	Cost	Score	Cost	Score	Cost
Qwen2.5-7B	**None**	3.12	0	56.60	0	87.46	0	32.23	0	19.21	0	39.72	0
	TAPT	3.16	0.90	56.40	1.13	87.72	2.32	33.00	1.88	19.11	1.61	39.88	1.57
	Bonito	22.78	1.19	71.40	1.42	NA	NA	NA	NA	NA	NA	NA	NA
	DeepSeeek-V3 w/ Self-Instruct	NA	NA	74.00	10.40	87.81	16.73	36.95	20.31	**24.07**	26.02	NA	NA
	DeepSeeek-V3 w/ Unlabeled Data	16.09	3.91	75.80	4.65	**88.55**	7.21	36.89	9.14	20.73	12.96	47.61	7.57
	DeepSeeek-V3 w/ SI+UD	30.20	6.33	**76.80**	7.55	88.34	12.88	36.81	18.32	23.22	24.47	**51.47**	13.91
	AQuilt	**34.69**	1.48	74.00	1.75	88.44	2.90	**38.00**	2.44	22.11	2.25	51.45	2.16
Llama3-8B	**None**	3.68	0	73.60	0	58.32	0	27.79	0	15.37	0	35.75	0
	TAPT	3.67	1.22	73.60	1.56	58.50	3.73	28.13	3.07	15.20	2.38	35.82	2.39
	Bonito	23.05	1.51	72.20	1.85	NA	NA	NA	NA	NA	NA	NA	NA
	DeepSeeek-V3 w/ Self-Instruct	NA	NA	74.80	10.75	61.96	17.95	34.07	21.50	21.26	26.57	NA	NA
	DeepSeeek-V3 w/ Unlabeled Data	16.69	4.23	75.80	4.91	59.25	8.43	34.67	10.33	19.27	13.52	41.14	8.28
	DeepSeeek-V3 w/ SI+UD	32.01	6.65	**76.40**	7.81	**64.16**	14.10	**35.57**	19.51	**21.93**	25.03	46.01	14.62
	AQuilt	**40.89**	1.79	75.20	2.10	63.16	3.91	34.50	3.35	19.65	2.77	**46.68**	2.78

Table 2: Main results for downstream task learning. "Model" indicates the base models used for training, all of which are the instruction versions. "Source" represents the source of the training data synthesis, detailed in §4.2. "Cost" represents the total expense for data synthesis and training. "Avg." represents the average scores across all tasks. **Bold** indicates the best performance, while underline indicates the second-best for each task.

4.2 Evaluation Setup of AQuilt

Benchmark. In this work, we conduct experiments across distinct downstream tasks, covering various task types. For extractive QA, we use **SquadQA** (Rajpurkar et al., 2018) and follow the online adaptation setting (Hu et al., 2023), which teaches LLM domain knowledge contained in unlabeled data. For yes/no QA, we select **PubMedQA** (Jin et al., 2019), an English natural language inference task related to medical research papers. For multi-choice QA, we choose eight subjects from **CEVAL** (Huang et al., 2023), covering compulsory courses from middle school to university in China. For translation and open-ended QA, we utilize the Legal **Translation** and Legal **EssayQA** tasks from LexEval (Li et al., 2024a). These tasks span different domains, validating the cross-domain and cross-task capabilities of our data synthesis model.

We evaluate PubMedQA and CEVAL using accuracy. Following Rajpurkar et al. (2018), we use the SQuAD F1 score to evaluate the SquadQA test dataset. For the Translation and EssayQA tasks, as in LexEval (Li et al., 2024a), we compute the Rouge-L score on the generated output. Also, we compute the BERTScore for the Translation and EssayQA tasks, listed in the Appendix B, to ensure the robustness of the evaluation metrics.

Domain Data Generation. AQuilt requires domain-specific unlabeled data, which we source as follows: SquadQA uses test set data (Hu et al., 2023), PubMedQA uses its original training set, CEVAL collects textbooks, while legal tasks (Translation and EssayQA) use Chinese CAIL (china-ai-law challenge, 2024) and English MAUD/UK-Absp (Wang et al., 2023a; Shukla et al., 2022) datasets. Task-specific data is generated following §3.4, with unseen tasks like Translation and EssayQA framed as closed-book QA via question prefixes. With vLLM (Kwon et al., 2023) (temperature=0.7, top_p=0.95, max_length=1024), we synthesize 20k training samples per task.

Baselines. We compare different baselines based on the source of synthetic training data. For the "None" baseline, we directly prompt the model for evaluation without any training. For the "TAPT" baseline, we follow Gururangan et al. (2020) to use task-adaptive pretraining, training the model only on unlabeled data. For the "Bonito" and "DeepSeek-V3 (w/ unlabeled data)" baselines, we use these models to synthesize data based on domain-specific texts and tasks, then fine-tune the models on the synthesized data. For the "DeepSeek-V3 (w/ Self-Instruct)" baseline, we generate data by applying Self-Instruct (Wang et al., 2023b) (a method enabling model to autonomously create training instructions) to DeepSeek-V3 and adapt the prompt from SELF-GUIDE (Zhao et al., 2024) for the synthesis of domain-specific task data. For the "DeepSeek-V3 (w/ Self-Instruct + Unlabeled Data)" baseline, which is abbreviated as "DeepSeek-V3 (w/ SI + UD)" in Table 2, we incorporate unlabeled data into the Self-Instruct data synthesis process (Specific prompts in Appendix D).

Training Details for Specialist LLMs. To validate the effect of the synthetic data, we per-

form experiments based on instruct models, including Qwen2.5-7B-Instruct (Yang et al., 2024a) and Llama3-8B-Instruct (Dubey et al., 2024). We finetune these models with LoRA on different supervision sources for 3 epochs per task. All other settings are consistent with those in Section 4.1. Note that we use a lower LR of 1e-7 and single-epoch training for CEVAL based on Qwen2.5 (prone to overfitting given Qwen2.5's strong baseline) and TAPT (to preserve instruction-following capability). For TAPT specifically, we reformat tasks into tuning prompts like "Please output [Domain] text in English: [Sentence]" to maintain alignment.

4.3 Main Results

Superior Performance. As shown in Table 2, AQuilt outperforms most baselines on average and is comparable to the best setting using DeepSeek-V3 (w/ SI + UD). Besides, TAPT, which relies solely on unlabeled data, shows no significant improvement across tasks, highlighting the value of labeled data synthesis. SquadQA tests whether synthetic data enables effective learning of domain-specific knowledge from unlabeled data. Since DeepSeek-V3 (w/ Self-Instruct) lacks unlabeled data, it cannot follow this setup, hence its results are marked as NA. In contrast, AQuilt significantly improves results on this task, confirming LLMs' poor performance on extractive QA tasks and justifying the use of original QA from labeled data (introduced in §3.2) for such tasks.

Low-Cost Generation. We compute the cost of different data synthesis and training methods in dollars. Given the DeepSeek-V3 model's 671B size makes local deployment impractical, we use its official API and base the cost calculation on total data synthesis expenditure. For other sources, we instead use local NVIDIA 4090 24GB GPUs, calculating costs via Vast AI[1]'s GPU rental prices. For production costs, we calculate the total cost based on the GPU Hours used.

AQuilt matches DeepSeek-V3 (w/ SI + UD) in performance at 17% of its cost, and achieves better results than DeepSeek-V3 (w/ Unlabeled Data) using 31% of the cost, demonstrating AQuilt's significant efficiency advantages in data synthesis.

Cross-Task Generalization. Bonito is unable to generate data for three tasks (results are marked as NA) due to its support for only English tasks that rely on unlabeled data. In contrast, AQuilt achieves better task generalization by incorporating Chinese data and defining more flexible task types. Specifically, AQuilt assigns the task type as closed/open-book QA and uses task requirements as question prefixes. Experimental results demonstrate that AQuilt consistently outperforms various baselines on these tasks, highlighting its strong ability to generalize across different tasks.

[1] https://vast.ai/

Model	SquadQA	CEVAL	Translation	Avg.
AQuilt	**40.89**	**63.16**	**34.50**	**46.18**
w/o Logic	40.68	59.64	33.61	44.64
w/o Self-Inspection	40.00	60.95	34.22	45.06
w/ Low-Quality	39.81	59.22	33.15	44.06

Table 3: Ablation results for logic and self-inspection. We independently remove logic and self-inspection to observe performance changes. w/ Low-Quality refers to using self-inspection to select low-quality data, in contrast to the main experiment settings.

Effectiveness of Unlabeled Data. By comparing DeepSeek-V3 (w/ SI + UD) with DeepSeek-V3 (w/ Self-Instruct), it's evident that even on the strong DeepSeek-V3 model foundation, incorporating unlabeled data during domain data synthesis can further improve synthetic data quality. This highlights the effectiveness of using domain-specific unlabeled data for data synthesis.

5 Analysis

We provide a comprehensive analysis to demonstrate the effect of our method and the underlying reasons for its success. Given the implementation cost, unless otherwise specified, the experiments are based solely on the results obtained from Llama3-8B-Instruct as the base model and focus on SquadQA, CEVAL, and Translation.

5.1 Analysis on Logic and Self-Inspection

To evaluate the effects of logic and self-inspection, we independently remove each component during the training process. Results are in Table 3.

Effect of Logic. To validate the effect of logic, we remove the logic component from the entire training pipeline (w/o Logic) and retrain AQuilt using data without logic, while maintaining all other settings and experimental setups. In subsequent data synthesis, no logic is incorporated. The results show a significant drop in model performance after removing logic, demonstrating its critical role within the entire data synthesis framework.

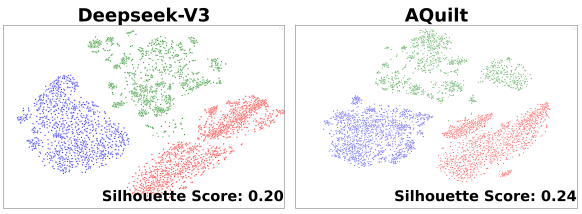

Figure 4: Relevance analysis of synthesized domain data. We convert generated questions into sentence vectors and analyze the distribution across CEVAL (Red), Translation (Green), and SquadQA (Purple).

Effect of Self-Inspection. To assess the impact of self-inspection, we conduct the following evaluation: the optimal setting for filtering low-quality data (AQuilt), a no-filtering setup (w/o Self-Inspection), and the use of only low-quality data identified by self-inspection (w/ Low-Quality). To ensure comparability, consistent data volumes are applied across all settings. The results confirm the significant effect of self-inspection. Further, under w/ Low-Quality, we observe a slight decrease in performance, indicating that the presence of low-quality data negatively impacts model performance. Fortunately, when logic is applied, overall performance remains at a relatively high level.

5.2 Domain Relevance of Generated Data

We demonstrate that our method achieves superior performance by generating data with higher relevance to the target domain and lower noise. To demonstrate this, we select CEVAL, Translation, and SquadQA as three distinct domain tasks, each with 2k synthetic data samples. Using the Qwen2.5-7B, we compute sentence embeddings for the synthetic questions generated by DeepSeek-V3 (w/ Unlabeled Data) and AQuilt, which are both created based on unlabeled data.

As shown in Figure 4, we apply t-SNE for dimensionality reduction and plot a 2D scatter plot. The results show that the generated data of AQuilt is more concentrated and contains fewer noises. To confirm this quantitatively, we compute the Silhouette Score, which reflects consistency and relevance within data clusters, with higher values indicating stronger domain relevance. The results align with the scatter plot, demonstrating that our method generates more concentrated data with reduced noise and increased relevance.

Model	SquadQA	CEVAL	Translation	Avg.
DeepSeek-V3	6.90%	8.18%	0.00%	5.23%
AQuilt	**0.40%**	**5.15%**	0.00%	**1.85%**

Table 4: Independence analysis of synthetic data with unlabeled data. We assess the percentage of synthetic questions generated by DeepSeek-V3 and AQuilt that are highly dependent on unlabeled data, revealing their low relevance to downstream tasks.

Source	SquadQA	CEVAL	Translation	Avg.	
				Score	Cost
Qwen2.5-72B	21.19	59.06	**34.82**	38.36	19.92×
AQuilt	**40.89**	**63.16**	34.50	**46.18**	**1**×

Table 5: Comparison with Qwen2.5-72B-Instruct. All abbreviations are consistent with those in Table 2.

5.3 Analysis of Relevance-Aware Filtering

For multi-choice and closed-book QA tasks, the synthetic labeled data must remain independent of the unlabeled data; otherwise, introducing false correlations during training may lead to low relevance to these tasks. To address this, we apply relevance-aware data filtering (introduced in Section 3.2).

In the unlabeled data-based synthetic generation setting (Table 2), while AQuilt demonstrates consistent advantages across tasks, DeepSeek-V3 (w/ Unlabeled Data) shows curiously limited gains on CEVAL. To analyze the underlying cause, we examine the proportion of generated questions that rely on unlabeled data to generate answers, which may contribute to hallucinations. We sample 2,000 instances from the generated dataset and use GPT-4o to evaluate. As shown in Table 4, the results indicate that even when explicitly instructed in the prompt, strong models like DeepSeek-V3 still tend to generate questions with spurious correlations, resulting in low relevance to downstream tasks and consequently reducing overall performance.

5.4 Ablation of Base Model

To ensure a fair comparison of improvement sources, we conduct controlled experiments under identical settings (based on the same unlabeled data) between AQuilt (trained on Qwen2.5-7B) and the Qwen family's larger 72B model, confirming our enhancements originate from methodology rather than base model capacity.

As shown in Table 5, our method outperforms the 72B model on average, while requiring only about 1/20 GPU hours on NVIDIA A800 80G. The overall experimental results align with the main

Model	SquadQA	CEVAL	Translation	Avg.
Bonito	2.43	NA	NA	NA
AQuilt	2.98	4.16	3.58	3.57

Table 6: Using GPT-4o to evaluate 1,000 samples randomly drawn from the training data synthesized by AQuilt and Bonito for the three test tasks.

Model	1	2	3	4	5
Bonito	28.75%	40.60%	24.90%	4.20%	1.55%
AQuilt	3.97%	5.92%	19.67%	27.60%	22.84%

Table 7: Probability distribution of GPT-4o's 1-5 scores for 1,000 randomly drawn samples from the AQuilt and Bonito synthesized training data.

experimental trends, further validating the effect of our method and confirming that the superior performance is not due to a stronger base model.

5.5 GPT-4o Evaluation

To validate the quality of the questions generated by AQuilt and to compare its performance with models of comparable scale, Bonito, we present the scoring results (without Self-Inspection filtering) obtained in Table 6 by using GPT-4o (temperature=0.7, top_p=0.95) to evaluate 1,000 samples randomly drawn from the training data synthesized by AQuilt and Bonito for each task. The prompts used by GPT-4o are completely consistent with those shown in the Appendix D, with a scoring scale ranging from 1 to 5 points.

Based on the prompt we provide to GPT-4o, a score of 2 points meets the basic quality requirement. As seen in Table 6, the average score for AQuilt-synthesized data across most tasks exceed 3 points. However, Bonito-synthesized data receive lower score. Meanwhile, since Bonito could not synthesize data for the CEVAL and Translation tasks, these are marked as NA. This indicates that the majority of data synthesized by AQuilt is relatively high-quality. For comparatively simpler tasks such as SquadQA and Translation, GPT-4o tends to assign slightly lower scores, demonstrating that the difficulty of domain-specific tasks can influence the final evaluation results to some extent.

Meanwhile, Table 7 shows the distribution of GPT-4o's scores for the synthesized data of the aforementioned five tasks generated by AQuilt and Bonito (note that Bonito can only synthesize data for two of these tasks).

Based on the score distribution from GPT-4o above, across all five tasks, nearly half of the data received a score of 4 points. Only 3.97% of the data fail to meet the basic quality requirements (assigned a score of 1 point), and this portion will be filtered out by AQuilt's subsequent Self-Inspection module.

6 Conclusion

In this paper, we present AQuilt, a framework for generating data that incorporates **I**nspection, **Q**uestion, **U**nlabeled data, **A**nswer, **L**ogic, and **T**ask type from unlabeled data. Specifically, AQuilt enhances data synthesis quality through the introduction of **L**ogic and **I**nspection. The inclusion of **T**ask type, encompassing both open-book QA and closed-book QA, enables cross-task generalization for downstream data synthesis. Experimental results demonstrate that AQuilt outperforms Bonito, a widely used data synthesis model, in both task generalization and performance. Our synthetic data is even comparable to that of DeepSeek-V3, while requiring less than 17% of the production cost. Further analysis reveals that while the use of open-source LLMs yields generally favorable results, it shows drawbacks in format adherence and downstream task relevance. This underscores the necessity of dedicated data synthesis models. We will release all the training details and models to encourage further research.

Limitations

Additional Data Synthesis Sources. In this work, we used only DeepSeek-V3 as a source of distilled data. Expanding the data sources to include human-curated existing training datasets and data synthesized by more powerful models could provide a diverse mix of training data. This expansion may further enhance the diversity of the synthesized data styles, as well as improve downstream model performance and robustness.

Data Synthesis for Various Languages. In this work, we extend the language capabilities of the data synthesis model to two high-resource languages. In future work, we are interested in exploring the model's performance on mid- to low-resource languages, where the model may have poorer performance and less data availability. Additionally, we will investigate whether the model exhibits zero-shot generalization capabilities when encountering languages not seen during training.

Advanced Data Synthesis Frameworks. With the introduction of DeepSeek-R1 (Guo et al., 2025) and Kimi-K1.5 (Du et al., 2025), more advanced data synthesis frameworks have emerged, which use iterative data synthesis, high-quality evaluation, and reinforcement learning to generate progressively stronger data. Our self-inspection training framework has the potential to generalize to these frameworks. It is a future work in this setting.

Ethics Statement

Our work adheres to the ACL Ethics Policy and publicly released the code for reproducibility. LLMs may exhibit racial and gender biases, so we strongly recommend users assess potential biases before applying the models in specific contexts. Additionally, due to the difficulty of controlling LLM outputs, users should be cautious of issues arising from hallucinations.

Acknowledgments

This work was supported in part by Guangdong S&T Program (Grant No. 2024B0101050003), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515011491), and Shenzhen Science and Technology Program (Grant Nos. ZDSYS20230626091203008, KJZD20231023094700001, KQTD20240729102154066). We would like to thank the anonymous reviewers and meta-reviewer for their insightful suggestions.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, et al. 2024. Phi-4 Technical Report. *arXiv preprint arXiv:2412.08905*.

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

Hritik Bansal, Arian Hosseini, Rishabh Agarwal, Vinh Q. Tran, and Mehran Kazemi. 2024. Smaller, weaker, yet better: Training LLM reasoners via compute-optimal sampling. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Zhijie Bao, Wei Chen, Shengze Xiao, Kuang Ren, Jiaao Wu, Cheng Zhong, Jiajie Peng, Xuanjing Huang, and Zhongyu Wei. 2023. Disc-medllm: Bridging general large language models and real-world medical consultation. *CoRR*, abs/2308.14346.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: investigating adversarial human annotation for reading comprehension. *Trans. Assoc. Comput. Linguistics*, 8:662–678.

Christian Bartz, Hendrik Raetz, Jona Otholt, Christoph Meinel, and Haojin Yang. 2022. Synthesis in style: Semantic segmentation of historical documents using synthetic data. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3878–3884.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2024a. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yongrui Chen, Haiyun Jiang, Xinting Huang, Shuming Shi, and Guilin Qi. 2024b. DoG-instruct: Towards premium instruction-tuning data via text-grounded instruction wrapping. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4125–4135, Mexico City, Mexico. Association for Computational Linguistics.

china-ai-law challenge. 2019. Cail2019 - china ai and law challenge 2019.

china-ai-law challenge. 2024. Cail (china ai and law challenge) official website.

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2019. A span-extraction dataset for Chinese machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5883–5889, Hong Kong, China. Association for Computational Linguistics.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932, Hong Kong, China. Association for Computational Linguistics.

DataFountain. 2020. Covid-19 government affairs question answering assistant dataset.

Hexuan Deng, Liang Ding, Xuebo Liu, Meishan Zhang, Dacheng Tao, and Min Zhang. 2023. Improving simultaneous machine translation with monolingual data. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 12728–12736. AAAI Press.

Hexuan Deng, Wenxiang Jiao, Xuebo Liu, Jun Rao, and Min Zhang. 2025a. REA-RL: Reflection-Aware Online Reinforcement Learning for Efficient Large Reasoning Models. *arXiv preprint arXiv:2505.19862*.

Hexuan Deng, Wenxiang Jiao, Xuebo Liu, Min Zhang, and Zhaopeng Tu. 2024. Newterm: Benchmarking real-time new terms for large language models with annual updates. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Hexuan Deng, Wenxiang Jiao, Xuebo Liu, Min Zhang, and Zhaopeng Tu. 2025b. DRPruning: Efficient Large Language Model Pruning through Distributionally Robust Optimization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. 2024. Re-ReST: Reflection-reinforced self-training for language agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15394–15411, Miami, Florida, USA. Association for Computational Linguistics.

Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, et al. 2025. Kimi k1.5: Scaling Reinforcement Learning with LLMs. *arXiv preprint arXiv:2501.12599*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Ronen Eldan and Yuanzhi Li. 2023. TinyStories: How Small Can Language Models Be and Still Speak Coherent English? *arXiv preprint arXiv:2305.07759*.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT Outperforms Crowd-Workers for Text-Annotation Tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Giovanni Grano, Andrea Di Sorbo, Francesco Mercaldo, Corrado A Visaggio, Gerardo Canfora, and Sebastiano Panichella. 2017. Android apps and user feedback: a dataset for software evolution and quality improvement. In *Proceedings of the 2nd ACM SIGSOFT international workshop on app market analytics*, pages 8–11.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rStar-Math: Small LLMs Can Master Math Reasoning with Self-Evolved Deep Thinking. *arXiv preprint arXiv:2501.04519*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*.

Yufei Guo, Muzhe Guo, Juntao Su, Zhou Yang, Mengqiu Zhu, Hongfei Li, Mengyang Qiu, and Shuo Shuo Liu. 2024. Bias in Large Language Models: Origin, Evaluation, and Mitigation. *arXiv preprint arXiv:2411.10915*.

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, Online. Association for Computational Linguistics.

Institute for Frontier Information Haihua and Institute for Interdisciplinary Information Sciences Tsinghua. 2021. 2021 hai hua ai competition.

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Austin, Texas. Association for Computational Linguistics.

Chitranshu Harbola and Anupam Purwar. 2025. Knowslm: A framework for evaluation of small language models for knowledge augmentation and humanised conversations. *arXiv preprint arXiv:2504.04569*.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, and et al. 2018. DuReader: a Chinese machine reading comprehension dataset from real-world applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 37–46, Melbourne, Australia. Association for Computational Linguistics.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-STar: Training verifiers for self-taught reasoners. In *First Conference on Language Modeling*.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A large scale Chinese short text summarization dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1967–1972, Lisbon, Portugal. Association for Computational Linguistics.

Edward J. 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, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Nathan Hu, Eric Mitchell, Christopher Manning, and Chelsea Finn. 2023. Meta-learning online adaptation of language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4418–4432, Singapore. Association for Computational Linguistics.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Hyeonbin Hwang, Doyoung Kim, Seungone Kim, Seonghyeon Ye, and Minjoon Seo. 2024. Self-explore: Enhancing mathematical reasoning in language models with fine-grained rewards. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1444–1466, Miami, Florida, USA. Association for Computational Linguistics.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. OpenAI o1 System Card. *arXiv preprint arXiv:2412.16720*.

Su Jianlin. 2017. Baidu's chinese question-answering dataset webqa.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Pluto Junzeng. 2020. ChineseSquad: Chinese Read Comprehension Dataset.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A dataset for question answering via sentence composition. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8082–8090. AAAI Press.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Haitao Li, You Chen, Qingyao Ai, Yueyue Wu, Ruizhe Zhang, and Yiqun Liu. 2024a. Lexeval: A comprehensive chinese legal benchmark for evaluating large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 25061–25094. Curran Associates, Inc.

Sihang Li, Jin Huang, Jiaxi Zhuang, Yaorui Shi, Xiaochen Cai, Mingjun Xu, Xiang Wang, Linfeng Zhang, Guolin Ke, and Hengxing Cai. 2024b. ScilitLLM: How to adapt LLMs for scientific literature understanding. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2024c. Self-alignment with instruction backtranslation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 58–62, Hong Kong, China. Association for Computational Linguistics.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi

Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. 2025. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *The Thirteenth International Conference on Learning Representations*.

Andrius Mudinas, Dell Zhang, and Mark Levene. 2018. Bootstrap domain-specific sentiment classifiers from unlabeled corpora. *Transactions of the Association for Computational Linguistics*, 6:269–285.

Nihal Nayak, Yiyang Nan, Avi Trost, and Stephen Bach. 2024. Learning to generate instruction tuning datasets for zero-shot task adaptation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 12585–12611, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

NLPCC. 2017. NLPCC 2017 Shared Task Data.

Anil Palepu, Vikram Dhillon, Polly Niravath, Wei-Hung Weng, Preethi Prasad, Khaled Saab, Ryutaro Tanno, Yong Cheng, Hanh Mai, Ethan Burns, et al. 2024. Exploring large language models for specialist-level oncology care. *arXiv preprint arXiv:2411.03395*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *arXiv preprint arXiv:2304.03277*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Jun Rao, Zepeng Lin, Xuebo Liu, Xiaopeng Ke, Lian Lian, Dong Jin, Shengjun Cheng, Jun Yu, and Min Zhang. 2025. APT: Improving specialist LLM performance with weakness case acquisition and iterative preference training. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20958–20980, Vienna, Austria.

Jun Rao, Xuebo Liu, Lian Lian, Shengjun Cheng, Yunjie Liao, and Min Zhang. 2024. CommonIT: Commonality-aware instruction tuning for large language models via data partitions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10064–10083, Miami, Florida, USA.

Jun Rao, Xv Meng, Liang Ding, Shuhan Qi, Xuebo Liu, Min Zhang, and Dacheng Tao. 2023. Parameter-efficient and student-friendly knowledge distillation. *IEEE Trans. Multim.*, pages 1–12.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to AI complete question answering: A set of prerequisite real tasks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8722–8731. AAAI Press.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693, Melbourne, Australia. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Chih-Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. 2018. DRCD: a chinese machine reading comprehension dataset. *CoRR*, abs/1806.00920.

Abhay Shukla, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh. 2022. Legal case document summarization: Extractive and abstractive methods and their evaluation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1048–1064, Online only. Association for Computational Linguistics.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge dataset and models for dialogue-based reading comprehension. *Trans. Assoc. Comput. Linguistics*, 7:217–231.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2020. Investigating prior knowledge for challenging chinese machine reading comprehension. *Trans. Assoc. Comput. Linguistics*, 8:141–155.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. QuaRTz: An open-domain dataset of qualitative relationship questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5941–5946, Hong Kong, China. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Alibaba Cloud Tianchi. 2020. Tianchi competition: Traditional chinese medicine literature question generation challenge.

Alibaba Cloud Tianchi. 2024. "wanchuang cup" traditional chinese medicine question generation challenge dataset.

Ojasw Upadhyay, Abishek Saravankumar, and Ayman Ismail. 2025. Synlexlm: Scaling legal llms with synthetic data and curriculum learning. *arXiv preprint arXiv:2504.18762*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.

Steven Wang, Antoine Scardigli, Leonard Tang, Wei Chen, Dmitry Levkin, Anya Chen, Spencer Ball, Thomas Woodside, Oliver Zhang, and Dan Hendrycks. 2023a. MAUD: An expert-annotated legal NLP dataset for merger agreement understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16369–16382, Singapore. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yutong Wang, Jiali Zeng, Xuebo Liu, Fandong Meng, Jie Zhou, and Min Zhang. 2024. TasTe: Teaching large language models to translate through self-reflection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6144–6158, Bangkok, Thailand. Association for Computational Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024. Monte carlo tree search boosts reasoning via iterative preference learning. In *The First Workshop on System-2 Reasoning at Scale, NeurIPS'24*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

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

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. 2024b. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. *arXiv preprint arXiv:2409.12122*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The Dawn of LMMs: Preliminary Explorations with GPT-4V(ision). *arXiv preprint arXiv:2309.17421*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *CoRR*, abs/1810.12885.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. RAFT: Adapting language model to domain specific RAG. In *First Conference on Language Modeling*.

Chenyang Zhao, Xueying Jia, Vijay Viswanathan, Graham Neubig, and Tongshuang Wu. 2024. Self-guide: Better task-specific instruction following via self-synthetic finetuning. In *First Conference on Language Modeling*.

Ben Zhou, Hongming Zhang, Sihao Chen, Dian Yu, Hongwei Wang, Baolin Peng, Dan Roth, and Dong Yu. 2024a. Conceptual and Unbiased Reasoning in Language Models. *arXiv preprint arXiv:2404.00205*.

Xiaomao Zhou, Qingmin Jia, and Yujiao Hu. 2024b. Advancing general sensor data synthesis by integrating llms and domain-specific generative models. *IEEE Sensors Letters*, 8(11):1–4.

Zhi Zhou, Jiang-Xin Shi, Peng-Xiao Song, Xiao-Wen Yang, Yi-Xuan Jin, Lan-Zhe Guo, and Yu-Feng Li. 2024c. Lawgpt: A chinese legal knowledge-enhanced large language model. *Preprint*, arXiv:2406.04614.

Ingo Ziegler, Abdullatif Köksal, Desmond Elliott, and Hinrich Schütze. 2024. CRAFT Your Dataset: Task-Specific Synthetic Dataset Generation Through Corpus Retrieval and Augmentation. *arXiv preprint arXiv:2409.02098*.

A Unlabeled Data Construction Details

We introduce the sources of the collected datasets, including unlabeled data and labeled data dataset, in which we collect (u, q, a, t) tuples to ensure higher quality.

Sources of Unlabeled Data. We collect unlabeled data from the following 33 datasets: CMRC2018 (Cui et al., 2019), ChineseSquad (Junzeng, 2020), CHIP2020 (Tianchi, 2020), CAIL2019 (china-ai-law challenge, 2019), DRCD (Shao et al., 2018), Covid19QA (DataFountain, 2020), WebQA (Jianlin, 2017), CMQG (Tianchi, 2024), HaihuaAI (Haihua and Tsinghua, 2021), C3 (Sun et al., 2020), NLPCC (NLPCC, 2017), Dureader (He et al., 2018), LCSTS (Hu et al., 2015), AdversarialDbidaf, AdversarialDroberta, AdversarialDbert (Bartolo et al., 2020), ANLI (Nie et al., 2020), APPReviews (Grano et al., 2017), CosmaQA (Huang et al., 2019), Dream (Sun et al., 2019), Duorc (Saha et al., 2018), Qasc (Khot et al., 2020), Quail (Rogers et al., 2020), Quartz (Tafjord et al., 2019), Quoref (Dasigi et al., 2019), RACE (Lai et al., 2017), Ropes (Lin et al., 2019), SocialQA (Sap et al., 2019), Squad (Rajpurkar et al., 2016), SuperGLUE (Wang et al., 2019), Record (Zhang et al., 2018), WikiHop (Welbl et al., 2018).

Sources of Labeled Data. We collect labeled data from the following 22 datasets: CMRC2018 (Cui et al., 2019), ChineseSquad (Junzeng, 2020), CHIP2020 (Tianchi, 2020), CAIL2019 (china-ai-law challenge, 2019), HaihuaAI (Haihua and Tsinghua, 2021), C3 (Sun et al., 2020), DRCD (Shao et al., 2018), Covid19QA (DataFountain, 2020), WebQA (Jianlin, 2017), CMQG (Tianchi, 2024), Dureader (He et al., 2018), LCSTS (Hu et al., 2015), AdversarialDbidaf, AdversarialDroberta, AdversarialDbert (Bartolo et al., 2020), ParaphraseRC, SelfRC (Saha et al., 2018), Quoref (Dasigi et al., 2019), Ropes (Lin et al., 2019), Squad (Rajpurkar et al., 2016), Record (Zhang et al., 2018), ANLI (Nie et al., 2020).

B Translation and EssayQA Task Evaluation Results

In the Table 8, we show the results of Translation and EssayQA tasks evaluated by Rouge-L and BERTScore. It can be seen that the data synthesized by AQuilt significantly improves the performance of the Instruct model on these tasks, with improvements comparable to DeepSeek-V3.

C Case Study

We present a synthetic data example generated by AQuilt in the table below.

D Prompts

We present all the prompts used in our paper in the following tables, including:

- **Prompts for Synthetic Training Dataset:** The prompt is used by the training dataset constructed in §3.2. The same prompt format is adhered to when generating downstream domain data with our dataset.
- **Logic Generation Prompts For Extractive QA:** The prompt is used when generating Logic for extractive QA tasks.
- **Prompts for Synthetic Inspection Training Dataset:** The prompt is used when generating inspection data with DeepSeek-V3.
- **Different Task Prompts and Self-Inspection Prompts for AQuilt:** The prompt is used when generating downstream task data with AQuilt, covering all tasks we defined.
- **Prompts for General LLM Downstream Task Generation Based on Unlabeled Data:** The prompt is used when generating data based on unlabeled data with a general-purpose LLM.
- **Prompts for General LLM Downstream Task Generation using SelfInstuct:** The prompt is used when generating data with a general-purpose LLM using SelfInstuct.
- **Prompts for Evaluation on Downstream Tasks:** The prompt is used during the evaluation of downstream tasks.
- **GPT-4o Prompts for Independence Analysis in Synthetic Data:** The prompt is used when analyzing the independence of synthetic data with unlabeled data.

Model	Source	Translation		EssayQA	
		Rouge-L	BERTScore	Rouge-L	BERTScore
Qwen2.5-7B	**None**	32.23	91.53	19.21	86.27
	TAPT	33.00	91.52	19.11	86.27
	Bonito	NA	NA	NA	NA
	DeepSeek-V3 w/ Self-Instruct	<u>36.95</u>	93.10	**24.07**	**87.30**
	DeepSeek-V3 w/ Unlabeled Data	36.89	92.95	20.73	86.52
	DeepSeek-V3 w/ SI + UD	36.81	<u>93.15</u>	<u>21.93</u>	<u>86.96</u>
	AQuilt	**38.00**	**93.46**	22.11	86.52
Llama3-8B	**None**	27.79	89.94	15.37	82.51
	TAPT	28.13	91.62	15.20	82.46
	Bonito	NA	NA	NA	NA
	DeepSeek-V3 w/ Self-Instruct	34.07	92.94	<u>21.26</u>	<u>86.96</u>
	DeepSeek-V3 w/ Unlabeled Data	<u>34.67</u>	<u>93.06</u>	19.27	86.29
	DeepSeek-V3 w/ SI + UD	**35.57**	**93.13**	**23.22**	**87.02**
	AQuilt	34.50	92.97	19.65	86.18

Table 8: The evaluation results of Translation and EssayQA tasks using Rouge-L and BERTScore as evaluation metrics respectively. **Bold** indicates the best performance, while <u>underline</u> indicates the second-best for each task.

AQuilt Synthetic Data Sample (from the PubMedQA Task)

Context:
Direct pressure applied on the inner ear cannot . . . . . the other group.
Task Type:
natural language inference
Question Generated by AQuilt:
Does direct pressure applied on the inner ear during scuba diving induce hearing loss, based on the findings that perforating the eardrums of guinea pigs before diving resulted in no variation in auditory threshold and cochlear spontaneous activity, even during extreme diving conditions?
Logic Generated by AQuilt:
1. Understand the Question: The question asks whether direct pressure on the inner ear during scuba diving causes hearing loss, based on experimental findings with guinea pigs. 2. Analyze the Text: The text describes an experiment where guinea pigs with perforated eardrums were subjected to diving conditions, including extreme depths and durations. No changes in auditory threshold or cochlear spontaneous activity were observed in this group. 3. Logical Reasoning: Since the experiment did not observe any hearing loss or changes in auditory function in the group with perforated eardrums, it suggests that direct pressure on the inner ear does not induce hearing loss. 4. Choose the Best Answer: The text supports the conclusion that direct pressure on the inner ear does not induce hearing loss, as evidenced by the lack of variation in auditory parameters in the experimental group.
Answer Generated by AQuilt:
No.
Quality Analysis Generated by AQuilt:
1. The question is clear and specific, asking whether direct pressure on the inner ear during scuba diving induces hearing loss based on experimental findings. 2. The response is relevant and directly addresses the question by referencing the experimental findings with guinea pigs. 3. The response provides a logical reasoning process, explaining how the lack of observed changes in auditory parameters supports the conclusion that direct pressure does not induce hearing loss. 4. The grammar and expression in the response are fluent and clear, with no spelling errors. 5. The response provides a comprehensive solution by not only answering the question but also explaining the reasoning behind the conclusion, which helps the user understand the context and implications of the findings.
Quality Score Generated by AQuilt:
4

Prompts for Synthetic Training Dataset (Meta Prompts)

You are a professional Q&A pair generation assistant. Your responsibility is to create complete, clear, accurate, and useful {Task Type} Q&A pairs based on the provided text content. You need to deeply analyze the text to create reasonable questions and provide appropriate, detailed answers, ensuring that each Q&A pair is relevant and useful.
I. The {Task Type} questions you create should meet the following requirements:
Requirement 1: {Question Requirement}
Requirement 2: The {Task Type} questions generated can be answered without <text>, and the question provides comprehensive information, complete context, including the core content or key information of <text>. Specifically: The question must explicitly contain the key information of <text> to ensure that the question itself is self-contained. - Do not rely on external context or assume that the user already understands the content of <text>. Avoid using phrases such as according to the above text, in the context, above content, or based on the information provided;
Requirement 3: The {Task Type} questions generated should be as complex as possible, requiring multi-step reasoning to determine the final answer;
Requirement 4: The generated answer is accurate and error-free, based on credible facts and data from the provided text;
Requirement 5: The generated answer is complete, not only selecting the correct option but also providing explanation and thinking steps to exclude other distractors;
Requirement 6: {Thought Process Requirement}
Requirement 7: When generating Q&A pairs and thought process, assume there is no text as a reference, which means do not include phrases like the text, the context, or the information provided in the Q&A pairs and thought process you create.
II. Please generate a Q&A pair in the following format:
JSON
{
question:{{The question you create}},
thought process:{{The thought process you create}},
answer:{{The answer you create}}
}

III. Please study the above requirements carefully and create a {Task Type} Q&A pair:

Prompts for Synthetic Training Dataset (Specific Task Content 1/3)

Multi-Choice QA (single answer):
Task Type: single-choice
Question Requirement: The intent of the single-choice questions generated is clear and the semantics are explicit, including the question and necessary answers as well as distractors;
Thought Process Requirement: The thinking process should include the following steps: 1. Read the question: Understand the provided question. 2. Analyze the options: Assess the relationship and correctness of each candidate with the provided question. 3. Choose the best answer: Select the most accurate and contextually relevant option;

Multi-Choice QA (multi answer):
Task Type: multi-choice
Question Requirement: The intent of the generated multiple-choice question is clear and the semantics are explicit, including the question and necessary answers (there can be multiple answers that meet the requirements of the question) as well as distractors;
Thought Process Requirement: The thought process should include the following steps: 1. Read the question: Understand the question provided and clarify what is required. 2. Analyze the options: Assess each candidate option's relationship and correctness in relation to the provided question. 3. Choose reasonable answers: Based on the analysis of each option, select all reasonable answers;

Closed-Book QA:
Task Type: closed-book
Question Requirement: The generated questions should be answerable without external knowledge and should provide comprehensive information, complete context, and contain relevant background information; do not generate questions about specific small events, as such questions are meaningless;
Thought Process Requirement: The thought process should include the following steps: 1. Understanding the question: Understand the question and clarify its requirements. 2. Analyzing the question: Analyze relevant information based on your own knowledge without relying on external resources. 3. Formulating a response: Construct a reasonable and accurate answer.

Open-Book QA:
Task Type: open-book
Question Requirement: Open Q&A refers to identifying and extracting specific information segments from the given text to answer the question. The generated question should explicitly include the text needed to answer the question and should not directly use the text provided by the user.
Thought Process Requirement: The generation of the thought chain should include the following steps: 1. Read the text: fully understand the problem and the provided text or paragraph. 2. Identify the relevant parts: locate the specific text segments that contain the answer to the question. 3. Construct the final reply: summarize the answer to the question.

Prompts for Synthetic Training Dataset (Specific Task Content 2/3)

Text Classification:
Task Type: text classification
Question Requirement: The generated classification question should have a clear intent and unambiguous semantics, including the text content and predefined categories.
Thought Process Requirement: The generated thought process should include the following steps: 1. Analyze the content: Examine the content of the <text>, identifying themes, keywords, or other indicative features.2. Map to labels: Match the analyzed features to predefined labels or categories.3. Confirm classification: Verify that the assigned label accurately reflects the content.4. Record the result: Record or output the classification result.

Natural Language Inference:
Task Type: natural language inference
Question Requirement:The question generated can be answered with Yes/No/Maybe.
Thought Process Requirement: The generated thought process should include the following steps:1.Understand the question: Understand the provided question. 2. Analyze the question: Identify which specific parts of the text the question is related to. 3. Logic Reasoning: provide logic reasoning to answer. 4. Provide the best answer: Select yes/no/maybe to answer the question.

Text Generation:
Task Type: text generation
Question Requirement: The text generation problem should have a clear intent and be semantically clear. It must include the user's instructions and the conditions for generation.
Thought Process Requirement: The thought process for generation should include the following steps: 1. Understand the input conditions: Review the user's instructions and conditions to grasp the required scope, tone, and structure. 2. Brainstorm content: Develop key ideas or themes consistent with the input conditions. 3. Generate output: Create a well-structured and coherent response that follows the user's instructions.

Text Summarization:
Task Type: text summarization
Question Requirement: The text summarization problem should have a clear intent and be semantically clear, including the text content and summarization requirements.
Thought Process Requirement: 1. Identify key points: Extract the most important information that represents the overall content of the source material. 2. Organize information: Logically arrange the extracted key points to ensure clarity and coherence. 3. Generate summary: Condense the key points into a concise and clear summary.

Prompts for Synthetic Training Dataset (Specific Task Content 3/3)

Natural Languaga Understanding:
Task Type: natural language understanding
Question Requirement: The generated natural language understanding tasks can specifically include sentiment analysis, intent recognition, entity recognition, part-of-speech tagging, semantic analysis, etc.
Thought Process Requirement: The generated thinking steps should include the following steps: 1. Read the task: Understand the provided task. 2. Analyze the problem: Evaluate the relationship and correctness of the input text in relation to the task. 3. Provide the best answer: Respond with the most accurate and contextually relevant answer.

Logic Generation Prompts for Extractive QA

You are a professional assistant for generating thought process. Your responsibility is to synthesize useful thought process data based on the provided text content and question-answer pairs. You need to deeply analyze the text and construct a logical connection between the question and the answer.
1. The thought process you create should meet the following requirements: Requirement 1: The generated thought process should be rich in content and logically clear, demonstrating how to infer the <Answer> from the <Question>. Requirement 2: The thought process should include the following steps: (1). Read the question: Understand the provided question. (2). Analyze the question: Identify which specific parts of the text the question is related to. (3). Provide the best answer: Select the most relevant content from the text to answer the question.
2. Please generate the thought process in the following format:
JSON
{
“thought_process”: “{{The thought process you created}}”
}
3. Please carefully study the above requirements and then create a thought process based on the following <text>, <question>, and <answer> provided by the user:

Prompts for Synthetic Inspection Training Dataset (1/2)

You are an AI instruction and response quality assessment assistant, please score the quality of the user's instruction and response according to the following scoring criteria, and you can refer to the <text> provided by the user to evaluate the correctness of the question and answer pair:
1. The scoring criteria are as follows:
1 point - Low quality, minimal requirements met (Low Level):
- The response is only partially relevant to the question and lacks depth or detail.
- The response contains noticeable grammatical errors, spelling mistakes, or awkward phrasing.
- The response fails to address the user's questions or needs adequately.
- The response provides no additional explanation, context, or background information, leaving the user with limited understanding.
2 points - Basic requirements met (Qualified Level):
- The response is relevant and can basically meet the user's needs.
- The response has correct grammar and no obvious spelling errors.
- The response can solve the user's problem, but the solution may not be comprehensive or in-depth enough.
- The response provides basic explanations and background information, but not in detail.
3 points - Good quality, meeting most requirements (Good Level):
- The response is highly relevant and can well meet the user's needs.
- The response is fluent in grammar, without spelling errors, and clearly expressed.
- The response provides a comprehensive solution that can solve the user's problem and considers possible follow-up issues.
- The response provides detailed explanations and background information, which helps users understand.
4 points - High quality, meeting all requirements and exceeding expectations (Excellent Level):
- The response is highly relevant, not only meeting user needs but also anticipating and resolving potential issues.
- The response has perfect grammar, precise expression, and well-chosen words.
- The response provides an in-depth solution that can comprehensively solve the problem from multiple angles and provides additional useful information or suggestions.
- The response provides in-depth explanations and background information, which helps users gain a deeper understanding of the problem and solution.
5 points - Excellent quality, exceeding all requirements with professional contributions (Outstanding Level):
- The response is highly relevant, not only meeting user needs but also providing solutions beyond expectations.
- The response has impeccable grammar, elegant expression, and precise, impactful wording.
- The response provides an in-depth and professional solution that can solve the problem from a unique perspective and provides extremely valuable additional information or suggestions.

Prompts for Synthetic Inspection Training Dataset (2/2)

- The response provides in-depth explanations and background information, demonstrating a high level of professionalism and profound understanding of the issue.
2. Please analysis the quality in the following format:
JSON
{
analysis_steps: {{your analysis for the quality}},
score: {{your rate to the qa_pair}}
}

3. Please carefully study the above scoring criteria and strictly follow the scoring criteria above to score the following <qa_pair> based on the following <text> provided by the user:

Different Task Prompts for AQuilt (1/5)

single choice question answering: """Please generate a single-choice question from the provided reference materials to help students better grasp the relevant knowledge: The single-choice question should include a question, four options labeled A, B, C, and D, one of which is the answer to the question;
At the same time, you also need to generate the thinking steps for solving the question, as well as the answer to this question.
And output in the following JSON format:
JSON
{"question": "xxx", "thinking_steps": "xxx", "answer": "xxx"}

""",
multi choice question answering: """Please generate a multiple-choice question from the references provided to help students better grasp the knowledge:
The multiple-choice question should include a question with multiple options tags A, B, C, D, E (and so on), one or more of which are the answers to the questions; At the same time, you also need to generate the thinking steps for solving the question, as well as the answer to this question.
And output in the following JSON format:
JSON
{"question": "xxx", "thinking_steps": "xxx", "answer": "xxx"}
"""

Different Task Prompts for AQuilt (2/5)

close-book question answering: """Please generate a closed-book question and answer pair from the provided reference materials that do not require reference text to answer to help students better grasp the relevant knowledge:
This Q&A pair should include a question, and you also need to generate the thinking steps for solving the question, as well as the answer to this question. And output in the following JSON format:
JSON
{"question": "xxx", "thinking_steps": "xxx", "answer": "xxx"}
""",
open-book question answering: """Please generate an open-book Q&A pair from the provided reference materials to help students better grasp the relevant knowledge: This Q&A pair should include a question, and you also need to generate the thinking steps for solving the question, as well as the answer to this question. And output in the following JSON format:
JSON
{"question": "xxx", "thinking_steps": "xxx", "answer": "xxx"}
"""

Different Task Prompts for AQuilt (3/5)

text summarization: """Please generate a concise summary Q&A pairs of the provided text to help students better understand the main points:
The summary should capture the key ideas and essential information from the text. The content you generate should include a question, and you also need to generate the thinking steps for solving the question, as well as the answer to this question. And output in the following JSON format:
JSON
{"question": "xxx", "thinking_steps": "xxx", "answer": "xxx"}
""",
text generation: """Please generate a text-generated Q&A pair based on the text provided to help students learn:
The resulting text should be well-structured and relevant to the given text. The content you generate should include a question, and you also need to generate the thinking steps for solving the question, as well as the answer to this question. And output in the following JSON format:
JSON
{"question": "xxx", "thinking_steps": "xxx", "answer": "xxx"}
"""

Different Task Prompts for AQuilt (4/5)

natural language inference: """Please generate a logical inference question from the provided reference materials to help students better grasp the relevant knowledge:
Logical inference questions generally ask whether a judgment or piece of knowledge is correct, with answers including "yes, no, maybe" three options.
The content you generate should include a question, and you also need to generate the thinking steps for solving the question, as well as the answer to this question.
And output in the following JSON format:
JSON
{"question": "xxx", "thinking_steps": "xxx", "answer": "xxx"}
""",
text classification: """Generate a text classification task based on the text provided to help students understand the content of the text:
Classifications should be accurate and relevant to the given text.
The content you generate should include a question, and you also need to generate the thinking steps for solving the question, as well as the answer to this question.
And output in the following JSON format:
JSON
{"question": "xxx", "thinking_steps": "xxx", "answer": "xxx"}
"""

Different Task Prompts for AQuilt (5/5)

extractive question answering: """Please generate an extractive question answering task based on the provided reference materials to help students better understand the main points:
The content you generate should include a question, and you also need to generate the thinking steps for solving the question, as well as the answer to this question. And output in the following JSON format:
JSON
{"question": "xxx", "thinking_steps": "xxx", "answer": "xxx"}
""",
natural language understanding: """Please generate a natural language understanding question (such as sentiment analysis, semantic analysis, entity recognition, etc.) based on the provided reference materials to help students better grasp the relevant knowledge:
The content you generate should include a question, and you also need to provide the thinking steps to solve the question, as well as the answer to the question. Please output in the following JSON format:
JSON
{"question":"xxx", "thinking_steps": "xxx", "answer": "xxx"}
"""

Self-Inspection Prompts for AQuilt

Please score the quality of the user's instruction and response to help students understand the quality of the question and response based on the provided text. There are 5 levels of quality, which are: 1 point, 2 points, 3 points, 4 points, 5 points. The higher the score, the better the quality.
You'll first need to analyze the quality of the question and response before grading it. And output in the following JSON format:
JSON
{"analysis_steps": "xxx", "score": "xxx"}

Prompts for General LLM Downstream Task Generation Based on Unlabeled Data (SquadQA)

You are a professional Q&A pair generation assistant. Your responsibility is to create complete, clear, accurate, and useful extractive Q&A pairs based on the provided text.
I. The Q&A pairs you create should meet the following requirements:
Requirement 1: The generated questions should have clear intentions and semantics;
Requirement 2: The generated questions should be answerable without external knowledge; Do not rely on external context or assume that the user already understands the content of <text>.
Requirement 3: The question generated can be answered with the provided reference material.
Requirement 4: The generated answers should be accurate and based on credible facts and data;
Requirement 5: The generated answer can be found in the reference materials.
II. Please generate a Q&A pair in the following format:
JSON
{
"question": "{{The question you create}}",
"answer": "{{The answer you create(Extracted from reference materials)}}"
}
III. Please study the above requirements carefully and create a extractive Q&A pair based on the <text> provided by the user below:

Prompts for General LLM Downstream Task Generation Based on Unlabeled Data (PubMedQA)

You are a professional Q&A pair generation assistant. Your responsibility is to create complete, clear, accurate, and useful Yes/No Q&A pairs based on the provided text.
I. The Q&A pairs you create should meet the following requirements:
Requirement 1: The generated questions should have clear intentions and semantics;
Requirement 2: The generated questions should be answerable without external knowledge;
Requirement 3: The question generated can be answered with either Yes or No.
Requirement 4: The generated answers should be accurate and based on credible facts and data;
Requirement 5: The generated answer has only two options: Yes/No.
II. Please generate a Q&A pair in the following format:
JSON
{
“question”: “{{The question you create}}”,
“answer”: “{{The answer you create(Yes/No)}}”
}
III. Please study the above requirements carefully and create a Yes/No Q&A pair based on the <text> provided by the user below:

Prompts for General LLM Downstream Task Generation Based on Unlabeled Data (CEVAL)

You are a professional question-answer pair generation assistant. Your responsibility is to create complete, clear, accurate, and useful single-choice question-answer pairs based on the provided text content. You need to analyze the text in-depth, create reasonable questions, and provide appropriate and detailed answers to ensure that each question-answer pair is relevant and useful.
I. The single-choice questions you create should meet the following requirements:
Requirement 1: The single-choice questions should have clear intentions and be semantically clear, including the question and necessary options as well as distractors;
Requirement 2: The single-choice questions should be answerable without the <text> and the information provided in the question should be comprehensive, with complete context and relevant background information;
Requirement 3: The answers generated should be accurate and based on credible facts and data from the provided text;
Requirement 4: The answers generated should be complete and not omit any necessary information;
II. Please generate the question-answer pairs in the following format:
JSON
{
“question”:“{{The question you create and the options}}”,
“answer”:“{{The correct answer you create}}”
}
III. After carefully studying the above requirements, please create a single-choice question-answer pair based on the <text> provided below:

Prompts for General LLM Downstream Task Generation Based on Unlabeled Data (Translation)

You are an expert in generating question-and-answer pairs. Your task is to create complete, clear, accurate, and useful closed-book question-and-answer pairs based on the provided text content. You need to analyze the text in depth, formulate reasonable questions, and provide appropriate and detailed answers, ensuring that each question-and-answer pair is relevant and useful.
I. The question-and-answer pairs you create should meet the following requirements:
Requirement 1 The questions should have clear intentions and be semantically clear.
Requirement 2 The questions should be answerable without external knowledge, and the information provided in the question should be comprehensive and contextually complete.
Requirement 3 The questions should be legal translation questions. You need to first determine the language of the given text. If it is in Chinese, the question should be of the Chinese-to-English type. Conversely, if the given text is in English, the question should be of the English-to-Chinese type.
Requirement 4 The type of question you generate can be randomly selected from the following four options: “Please translate the following sentence from the contract into Chinese/English:”, “Please translate the following legal term into Chinese/English:”, “Please translate the following sentence into Chinese/English:”, “Please translate the following legal provision into Chinese/English:”
Requirement 5: The answers you generate should be accurate and based on reliable facts and data.
II. Please generate the question-and-answer pairs in the following format:
JSON
{
“question”:“{{the question you create}}”,
“answer”:“{{the answer you create}}”
}
III. After carefully studying the above requirements, please create a question-and-answer pair based on the <text> provided by the user below:

Prompts for General LLM Downstream Task Generation Based on Unlabeled Data (EassyQA)

You are a professional question-and-answer pair generation assistant. Your responsibility is to create complete, clear, accurate, and useful question-and-answer pairs based on the provided text content. You need to analyze the text in depth, create reasonable questions, and provide appropriate and detailed answers to ensure that each question-and-answer pair is relevant and useful.
I. The question-and-answer pairs you create should meet the following requirements:
Requirement 1: The generated questions should have clear intentions and be semantically clear.
Requirement 2: The questions should be answerable without external knowledge, and the information provided in the questions should be comprehensive and contextually complete.
Requirement 3: The questions should be legal essay questions, starting with: "Please analyze the following essay question, elaborate on your views, and cite relevant legal provisions and principles. Ensure that you provide sufficient arguments and analysis for each question to clearly demonstrate your deep understanding and flexible application of legal issues."
Requirement 4: The answers generated should be accurate and based on reliable facts and data.
II. Please generate the question-and-answer pairs in the following format:
JSON
{
"question": "{{the question you create}}",
"answer": "{{the answer you create}}"
}
III. Please carefully study the above requirements and then create a Q&A pair based on the <text> provided by the user.

Prompts for General LLM Downstream Task Generation using Self-Instruct (Input Generation w/o context)

As an InputGenerator , your task is to generate a new [input] based on the [instruction] and some example [input].
Try your best to ensure that the new [input] you generate is distinct from the provided [input] while maintaining a diverse, detailed, precise, comprehensive, and high-quality response. Avoid generating a new [input] that is the same as the provided [input].
Start of instruction
{{instruction}}
End of instruction
Here are some high-quality [input] for the [instruction]. These [input] can provide you with very strict format requirements .
Below are [N] [input] examples:
{{Input Examples}}
Please generate 1 [input] based on the examples and requirements:

Prompts for General LLM Downstream Task Generation using Self-Instruct (Input Generation w/ context)

```
As an InputGenerator , your task is to generate a new [input] based on the
[instruction], [context] and some example [input].
Try your best to ensure that the new [input] you generate is distinct from the
provided [input] while maintaining a diverse, detailed, precise, comprehensive,
and high-quality response. Avoid generating a new [input] that is the same as
the provided [input].
Start of instruction
{{instruction}}
End of instruction
Here are some high-quality [input] and provided [context] for the [instruction].
These [input] can provide you with very strict format requirements .
Below are [N] [input] examples:
{{Input Examples}}
The provided context content is: {{context}}
Please generate 1 [input] based on the examples, requirements, and the provided
above context:
```

Prompts for General LLM Downstream Task Generation using Self-Instruct (Output Generation)

```
You are an AI question-answering bot, acting as an expert in the field of
{{domain}}. Please refer to the question provided by the user and answer the
question carefully.
User:Please answer the following question:
{{question}}
```

Prompts For evaluation on Downstream Tasks

SquadQA

Input: {question}

PubMedQA

Input: Context:{context} Based on the context above, please answer the following question:{question}

CEVAL

Input: Below is a multiple-choice question from a Chinese {subject} exam. Please select the correct answer.
{question}
A. {A} B. {B} C. {C} D. {D}
What is the answer?

Translation

Input: Please complete the following legal translation task and provide the translation directly. Translate the following {text type} into Chinese/English: {legal text}.

EssayQA

Input: Please analyze the following essay question. Elaborate on your views in detail and may cite legal provisions and relevant legal principles. Ensure that you provide sufficient arguments and analysis for each question to clearly demonstrate your profound understanding and flexible application ability of legal issues.
{material}
Question: {question}.

GPT-4o Prompts for Independence Analysis in Synthetic Data

You are a professional question analysis assistant, responsible for determining whether a question relies on a text for its answer based on the provided question.
Criteria for Judgment:
The question contains some obvious keywords that indicate reliance on a text, such as “the above content,” “according to the text,” “the above text,” “in the text,” “in the passage,” etc.
If the question is about understanding or inquiring about the content of a certain text, then it is also considered a question that relies on a text for its answer.
Formatting Requirements:
Please carefully review the above criteria.
Determine whether the question provided by the user has text dependency.
If it does, please answer directly with ’Yes.’
If it does not, please answer directly with ’No.