applied sciences

MDPI

Article

BOLT: Building Open-Source LLMs for Your Target Domain via Automated Hierarchical Knowledge Distillation

Runze Lu, Zhaoyu Fan, Guanjie Wang and Qingjiang Shi *

School of Computer Science and Technology, Tongji University, Shanghai 201804, China; 2333104@tongji.edu.cn (R.L.); zhaoyu_fan@tongji.edu.cn (Z.F.); 2354089@tongji.edu.cn (G.W.)
* Correspondence: shiqj@tongji.edu.cn

Abstract

Adapting open-source large language models (LLMs) to specialized domains remains a critical challenge due to domain knowledge gaps, data scarcity, and reference hallucination. Existing approaches often neglect the structural characteristics of domain knowledge and fail to provide principled estimations of knowledge scope, resulting in data homogenization and suboptimal adaptation, while leaving reference hallucination unmitigated. This paper introduces BOLT(Building Open-source LLMs for your Target domain), a modular end-to-end framework that tailors open-source LLMs for domain-specific scenarios. BOLT systematically estimates domain scope, constructs structured hierarchical knowledge trees, distills diverse and semantically aligned training data from advanced teacher LLMs, and employs curriculum learning for progressive model optimization. To address reference hallucination, BOLT substitutes generative methods, which are susceptible to hallucinations, with a matching-based strategy, thereby alleviating the problem and significantly improving reference recommendation accuracy. Extensive experiments across diverse domains and models demonstrate that BOLT enables the efficient modeling of structured hierarchical domain knowledge and effectively enhances reference recommendation accuracy while preserving both training efficiency and robustness throughout the adaptation process.

Keywords: large language models; knowledge distillation; domain-specific models; reference hallucination



Academic Editor: Ming Liu

Received: 24 September 2025
Revised: 16 October 2025
Accepted: 19 October 2025
Published: 24 October 2025

Citation: Lu, R.; Fan, Z.; Wang, G.; Shi, Q. BOLT: Building Open-Source LLMs for Your Target Domain via Automated Hierarchical Knowledge Distillation. *Appl. Sci.* **2025**, *15*, 11393. https://doi.org/10.3390/app152111393

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The rapid advancement of large language models (LLMs) has garnered substantial attention, owing to their impressive performance across a wide spectrum of tasks [1,2]. Adapting LLMs to specialized domains and real-world deployment has become a prominent and rapidly growing research trend [3–5]. Notably, mainstream open-source models [6–8], trained on large-scale and diverse corpora, have established strong performance baselines, thereby accelerating the development of domain-adapted variants.

Despite their strong general capabilities, open-source models often lack sufficient knowledge and proficiency in specialized domains, requiring further adaptation. As shown in Figure 1, this process faces several challenges, including the scarcity of high-quality domain data and the often ill-defined structure of domain knowledge. Manual data collection is prohibitively costly in terms of both time and human effort. Consequently, distilling knowledge from cutting-edge LLMs has become a widely adopted and scalable alternative [9,10]. However, existing methods often overlook the structure of domain knowledge

and fail to estimate how much training data are needed for different domains. These omissions can degrade data quality and limit downstream performance. Furthermore, many real-world applications require models to generate faithful and verifiable references, yet the generative nature of LLMs leads to hallucination [11], including fabricated citations [12]. All these challenges highlight the urgent need for principled estimation of domain knowledge scope and for more structured representations of domain expertise. In addition, addressing reference hallucination calls for mechanism-level solutions beyond surface-level filtering.

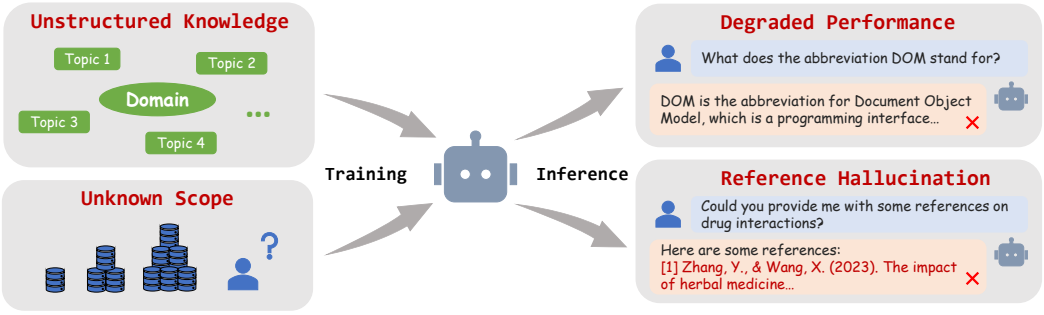


Figure 1. Challenges C applying open-source LLMs to specialized domains.

In this paper, we introduce BOLT (Building Open-source LLMs for your Target domain), a systematic framework for automating the adaptation of open-source LLMs to specialized domains. The framework comprises five key stages, each designed to progressively adapt open-source LLMs to target domains:

1. **Scope Estimation**. The breadth of domain knowledge is quantified through a recursive traversal of Wikipedia categories and domain-relevant articles. This stage provides an approximate boundary of the knowledge scope, ensuring that subsequent modeling efforts are neither too narrow to miss essential concepts nor too broad to dilute domain specificity.
2. **Knowledge Tree Construction**. Based on the estimated scope, hierarchical knowledge structures are constructed with the assistance of teacher LLMs. Each node in the knowledge tree corresponds to a domain concept, and the tree is expanded by recursively identifying prerequisite sub-concepts. In this manner, leaf nodes represent foundational knowledge that underpins their parent nodes (e.g., algebra as a prerequisite for advanced mathematics). The resulting knowledge tree captures both hierarchical dependencies and conceptual granularity, serving as a structured blueprint for data distillation.
3. **Data Distillation**. Guided by the hierarchical knowledge tree, teacher LLMs generate training samples that are not only semantically aligned with specific nodes but also enriched with explicit attributes such as length, difficulty level, and task format. The attributed distillation process ensures that the training dataset reflects both vertical and horizontal coverage, preventing data homogenization.
4. **Curriculum Learning**. The distilled data are organized into a curriculum aligned with the hierarchical tree. The student LLM is progressively optimized from general fundamental concepts to fine-grained domain-specific knowledge, facilitating stable convergence and reducing the risk of catastrophic forgetting.
5. **Reference Recommendation**. To alleviate reference hallucination, generative methods are replaced with a matching-based strategy. BOLT automatically collects relevant references according to the constructed knowledge tree and further retrieves appropriate citations by combining word-based matching and embedding-based matching against the query content. This hybrid strategy improves both the accuracy and reliability of references accompanying generated responses.

Through the integration of these components, BOLT provides an end-to-end framework for automating the adaptation of open-source LLMs to specialized domains. It enables reasonable estimation of the knowledge scope and organizes domain knowledge in a structured hierarchical manner, thereby facilitating efficient data construction and model training. Moreover, BOLT alleviates reference hallucination at the mechanism level, effectively improving the accuracy of reference recommendations. In summary, our contributions are as follows:

- A modular end-to-end framework, BOLT, that automates the domain adaptation of open-source LLMs through five integrated stages.
- A principled approach to domain knowledge modeling, featuring scope estimation and knowledge tree construction, which mitigates data homogenization and enhances the efficiency of knowledge acquisition.
- A novel matching-based method for reference recommendation, which mitigates hallucination while ensuring the authenticity of references.
- Extensive experiments demonstrating BOLT's effectiveness across various tasks, models, and domains, showcasing its superiority over baselines and strong generalizability across domain adaptation settings.

2. Related Work

2.1. Data Acquisition for LLM's Domain Adaptation

Acquiring high-quality and semantically aligned data is a core challenge in adapting LLMs to specialized domains. Various methods have involved manually collecting and processing datasets to train LLMs [13,14]. However, these approaches are typically labor-intensive, and it can be challenging to access many domain-specific knowledge areas due to their scarcity or confidentiality. In response, several studies [3,9,15] have explored leveraging the generative capabilities of LLMs to automate the creation of datasets, aiming to achieve results comparable to those of manually curated datasets. Despite their potential, these efforts overlook the varying knowledge scopes across domains and lack fine-grained knowledge structures, leading to data homogenization [16] and ultimately compromising data quality and diversity.

2.2. Training Strategies for Domain-Specific LLMs

The training strategy is a key factor that influences the effectiveness of domain-specific adaptation. Existing methods can be roughly classified into three categories. Supervised fine-tuning (SFT) is the most practical and widely used approach [17,18], directly optimizing model parameters on in-domain data. Divergence- and similarity-based methods [19,20] utilize internal signals from white-box teacher models to provide distributional guidance. Reinforcement learning [21] and ranking-based optimization [22] incorporate feedback signals from reward models or preference annotations to further enhance domain alignment. However, these methods often ignore the inherent logicality of domain knowledge and fail to adapt accordingly, leading to weak domain retention and even catastrophic forgetting [23].

2.3. Hallucination in LLMs

LLMs are known to generate fictitious references that do not correspond to real sources [12,24], a phenomenon referred to as reference hallucination, which is a specific case of broader hallucination in LLMs [11]. The underlying causes of hallucination generally fall into three categories: data, training, and inference [25]. To address data-related issues, prior work has applied rigorous filtering and curation of training corpora [26,27]. Training-phase solutions include model editing techniques [28,29] and architectural modifications [30],

as well as alignment-based methods [31] to reduce hallucination risk. At the inference level, factuality-aware decoding strategies [32,33] and faithfulness-enhanced methods [34] have been proposed to improve consistency and factual grounding in generated outputs. Although these methods offer partial relief from hallucination, they fail to eliminate it fundamentally and are thus inadequate for tasks such as reference recommendation, where factual accuracy is imperative.

3. Method

As illustrated in Figure 2, the BOLT framework comprises five stages. A detailed explanation of each stage is provided in the following sections.

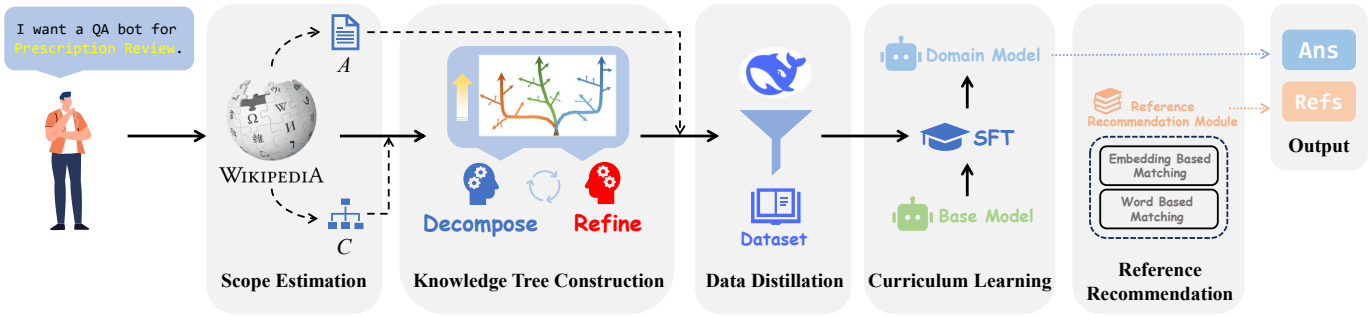


Figure 2. An overview of the BOLT framework.

3.1. Scope Estimation

BOLT begins with a specified target domain $\mathcal{D}$ and estimates its knowledge coverage using a traversal-based method grounded in the hierarchical structure of Wikipedia categories. Starting from the root category corresponding to $\mathcal{D}$, BOLT recursively collects subordinate categories and their associated articles. The resulting counts of categories and articles, denoted as C and A, serve as an intuitive yet informative proxy for the scale of domain knowledge, guiding the subsequent construction of the knowledge tree and dataset.

3.2. Knowledge Tree Construction

Inspired by the need to systematically expand from a specific domain to a comprehensive set of knowledge points, BOLT adopts a tree structure to represent the hierarchical and expandable nature of domain knowledge. BOLT initiates the construction of the knowledge tree $\mathcal{T}$ by setting the target domain $\mathcal{D}$ as its root and enqueuing it into the decomposition queue Q. The tree is then incrementally expanded under the guidance of two teacher models, denoted as $\mathcal{G}_1$ and $\mathcal{G}_2$, which alternately carry out knowledge decomposition and critique-driven refinement. The prompts used during the knowledge tree construction process are provided in Appendix A.

3.2.1. Knowledge Decomposition

If Q is not empty, the next node to be expanded is dequeued, denoted as k. Leveraging $\mathcal{G}_1$, a set of prerequisite knowledge points required for learning k in the context of $\mathcal{D}$ is generated, denoted as $\mathcal{K}$. The prompt used in this process is detailed in Appendix A.1.

3.2.2. Critique-Driven Refinement

Subsequently, the teacher model $\mathcal{G}_2$ evaluates the appropriateness of the knowledge points in $\mathcal{K}$. It supplements any critical yet omitted concepts while removing those deemed overly fine-grained or excessively elementary. In cases where the original point k is already

sufficiently specific and requires no further elaboration, the entire set $\mathcal{K}$ is discarded. If $\mathcal{K}$ remains nonempty, its elements are enqueued into Q for subsequent processing. The prompt template used in this process is provided in Appendix A.2.

In practice, it is necessary to impose constraints on both the degree and depth of the knowledge tree $\mathcal{T}$. These constraints can be heuristically linked to C introduced in Section 3.1, such that the degree d and depth h of the tree are both proportional to C, i.e., $d, h \propto C$. Accordingly, the two procedures described above are iteratively applied until the depth of the tree approaches the upper bound h. Subsequently, $\mathcal{G}_2$ is employed once more to review and refine $\mathcal{T}$. In this step, duplicate or highly similar topics are removed, following the principle of prioritizing nodes with a lower depth. The overall process is summarized in Algorithm 1, and an example of the resulting knowledge tree is shown in Figure 3.

Algorithm 1 Knowledge Tree Construction.

Input: Target domain $\mathcal{D}$, teacher models $\mathcal{G}_1$, $\mathcal{G}_2$, number of categories C
Output: Knowledge tree $\mathcal{T}$
Initialize the root of $\mathcal{T}$ as $\mathcal{D}$ and add $\mathcal{D}$ to an empty queue Q
Set degree d and depth h such that $d, h \propto C$
while $Q \neq \emptyset$ and depth$(\mathcal{T}) < h$ **do**
  $k \leftarrow Q$.Dequeue()
  $\mathcal{K} \leftarrow$ Decompose$(\mathcal{G}_1, d, k, \mathcal{D})$
  $\mathcal{K} \leftarrow$ Refine$(\mathcal{G}_2, \mathcal{K}, k, \mathcal{D})$
  if $\mathcal{K} \neq \emptyset$ **then**
    Q.Enqueue$(\mathcal{K})$
    $\mathcal{T}$.AddChildren$(k, \mathcal{K})$
  end if
end while
Refine$(\mathcal{G}_2, \mathcal{T})$
return $\mathcal{T}$

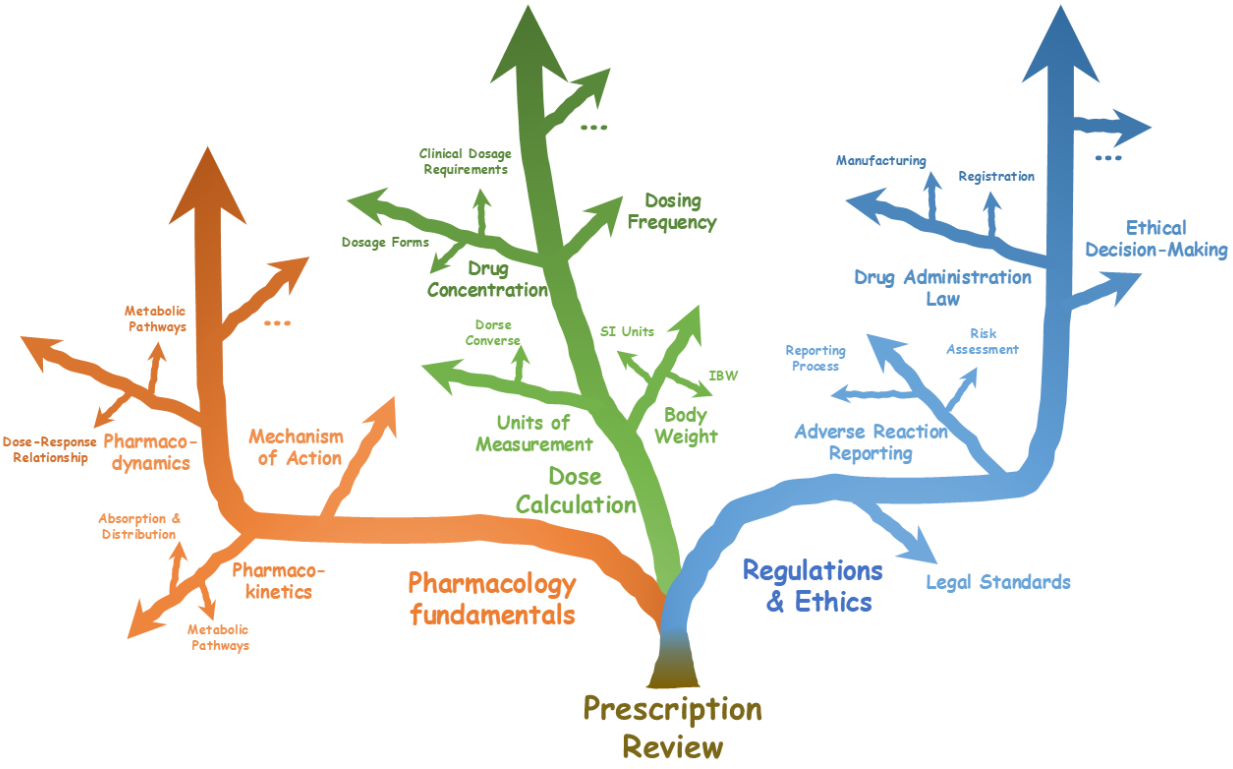


Figure 3. An example of the knowledge tree.

3.3. Data Distillation

After constructing the knowledge tree, BOLT generates the corresponding dataset. The overall dataset scale is informed by C and A, as introduced in Section 3.1. Based on the observation of a previous work [35] that effective alignment can be achieved with approximately 1000 high-quality training instances, and considering the empirical values of C and A in practice, the data volume N for each task in $\mathcal{D}$ is heuristically defined as $N = C \cdot \ln A$, which reflects both the conceptual complexity and the content richness associated with the domain. To enrich training data, BOLT incorporates an attribute set $\mathcal{B}$, which is a collection of data attributes, including length, mission format, difficulty level, etc. Furthermore, assuming that the importance of knowledge nodes varies with their depth in the tree, nodes closer to the root are considered to be more crucial and thus require a larger volume of data, and then another model $\mathcal{G}_3$ can be leveraged to distill data:

$$D_k = \text{Distill}(\mathcal{G}_3, \mathcal{D}, k, h_k, \gamma, N, \mathcal{B}), \tag{1}$$

where D_k denotes the dataset corresponding to knowledge point k, h_k indicates the depth of k, and γ is a parameter to quantify the amount of data allocated to nodes at different depths, so the amount of data for k is proportional to γ^{h_k}.

3.4. Curriculum Learning

BOLT employs a curriculum learning approach for model training. Specifically, the training data are organized in order from foundational knowledge points to the root node, and within the same layer, the data are arranged from easy to difficult. Furthermore, after completing the training on a given layer of knowledge points, a certain amount of samples from previously learned knowledge points are randomly inserted to enable the model to perform rehearsal learning. This curriculum learning strategy simulates human learning processes, allowing the model to learn in a structured manner from shallow to deep and from easy to hard while simultaneously mitigating catastrophic forgetting.

3.5. Reference Recommendation

After curriculum learning, a base model is transformed into a domain-specific model, denoted as $\mathcal{M}_\mathcal{D}$. To facilitate reference recommendation, responses generated by $\mathcal{M}_\mathcal{D}$ are enhanced through a question–reference matching method. For the target domain $\mathcal{D}$, BOLT automatically constructs a reference repository by pre-selecting relevant references from online academic sources, guided by the structure of the knowledge tree $\mathcal{T}$, denoted as $\mathcal{R}_\mathcal{D} = \{r_1, r_2, \ldots, r_{|\mathcal{R}_\mathcal{D}|}\}$. Given a question q, BOLT computes the similarity between each reference in $\mathcal{R}_\mathcal{D}$ and q, denoted as $\phi(q, r)$, where $r \in \mathcal{R}_\mathcal{D}$. Specifically, BOLT employs two matching algorithms: embedding-based matching and word-based matching.

3.5.1. Embedding-Based Matching

BOLT utilizes a sentence encoder to encode the question q and each reference $r \in \mathcal{R}_\mathcal{D}$ for similarity matching. The underlying concept of this approach is inspired by Retrieval-Augmented Generation (RAG) [36]. Specifically, BOLT employs a sentence encoder, denoted as $\mathcal{E}$, to embed q and r as vectors, and subsequently computes the cosine similarity between q and r:

$$\phi_\text{e}(q, r) = \frac{\mathcal{E}(q) \cdot \mathcal{E}(r)}{\|\mathcal{E}(q)\|_2 \|\mathcal{E}(r)\|_2}. \tag{2}$$

While inspired by RAG, BOLT differs in how retrieved information is utilized. In RAG, the retrieved documents only serve as contextual input to the model, and the model still freely generates text; thus, hallucinations may persist. In contrast, BOLT directly outputs

references that are explicitly matched through embedding similarity, ensuring that all cited references are real and verifiable.

3.5.2. Word-Based Matching

BOLT proposes a word-wise matching algorithm as a supplementary method for reference recommendation. Let the function $h(\cdot)$ denote the set of all distinct words from a given text segment; then, the similarity between q and r can be expressed as follows:

$$\phi_{\mathrm{w}}(q, r) = \frac{|h(q) \cap h(r)|}{|h(q)|}. \tag{3}$$

This measure closely resembles the Jaccard similarity [37], but it replaces the denominator in the original formula with a single set, $h(q)$. This adjustment addresses the imbalance caused by the varying lengths of reference titles.

Then, three references are recommended for each algorithm, denoted as $\mathcal{R}_{\mathrm{e}}$ and $\mathcal{R}_{\mathrm{w}}$:

$$\mathcal{R}_{\mathrm{e}} = \mathrm{argmax}_{\mathcal{R}_{\mathrm{e}} \subseteq \mathcal{R}_{\mathcal{D}}, |\mathcal{R}_{\mathrm{e}}|=3} \sum_{r \in \mathcal{R}_{\mathrm{e}}} \phi_{\mathrm{e}}(q, r), \tag{4}$$

$$\mathcal{R}_{\mathrm{w}} = \mathrm{argmax}_{\mathcal{R}_{\mathrm{w}} \subseteq \mathcal{R}_{\mathcal{D}}, |\mathcal{R}_{\mathrm{w}}|=3} \sum_{r \in \mathcal{R}_{\mathrm{w}}} \phi_{\mathrm{w}}(q, r), \tag{5}$$

The results from both algorithms are aggregated to provide the final set of references:

$$\mathcal{R} = \mathcal{R}_{\mathrm{e}} \cup \mathcal{R}_{\mathrm{w}}. \tag{6}$$

4. Experiments

4.1. Experimental Setup

4.1.1. Data Preparation

We focus on three domains: prescription review, digital payments, and copyright law. Each domain comprises specific tasks that can generally be categorized as generative or classification tasks, as summarized in Table 1. All teacher models were configured as DeepSeek-V3 [2]. In terms of scope estimation, the maximum recursion depth for querying articles and categories was constrained to two to achieve an optimal balance between coverage and relevance. Subsequently, the depth and maximum degree of the knowledge tree were configured based on parameters C and A, followed by the computation of the data size, as detailed in Table 2. To maintain a relatively controllable and appropriate number of nodes, we uniformly set the tree depth to 2, with the root at height 2 and the bottom nodes at height 0. To reflect the varying importance of knowledge points across different layers, we set γ to 1.25. The attribute set $\mathcal{B}$ in data distillation includes three categories: difficulty (easy, medium, hard), length (50–200 words), and type (conceptual understanding, reasoning and computation, practical application). Attribute values are randomly assigned during data generation. The BOLT framework automatically constructs reference repositories by retrieving the relevant academic literature from online search engines. The resulting reference counts are summarized in Table 3.

Table 1. Description of the three domains targeted in the experiment and the corresponding tasks along with their types. "Gen." represents generative tasks, while "Cls." represents classification tasks.

Prescription Review	Digital Payments	Copyright Law	Type
Domain Knowledge Quiz			Gen.
Multiple-Choice Question		Legal Knowledge Applying	Cls.
Rational Drug-Use Assessment	Sentiment Analysis	Legal Knowledge-Understanding	

Table 2. Information on domain scope, knowledge trees, and data size (number of training samples per task) across three domains (denoted by initials).

Domain	Scope		Tree			Data Size
	Categories	Articles	Degree	Depth	Nodes	
P.R.	178	33,190	4	2	41	1853
D.P.	127	3119	3	2	30	1021
C.L.	192	4550	4	2	44	1617

Table 3. Results on reference recommendation, including **Avg. Len.** (the average number of references recommended by BOLT) and **Repo. Size** (the number of references in the repository).

Domain	Avg. Len.	Repo. Size	BOLT (%)	DeepSeek (%)
Prescription Review	5.43	220	**51.57**	28.21
Digital Payments	5.31	94	**56.31**	43.59
Copyright Law	5.59	102	**52.42**	10.26

4.1.2. Models and Training

We selected six open-source models as base models: Llama3-8B [38], Mistral-7B [7], Baichuan2-7B [39], Qwen2-0.5B, Qwen2-1.5B, and Qwen2-7B [40]. Due to the constraints of computational resources, we employed a combination of mixed-precision training and LoRA fine-tuning [41], with the objective of maximizing the likelihood of predicting the next token in the supervised fine-tuning setting:

$$L = -\frac{1}{|\mathbf{y}|}\sum_{i=1}^{|\mathbf{y}|}\log P(y_i|\mathbf{x}, y_1, y_2, \ldots, y_{i-1}; \boldsymbol{\theta}_{\text{LoRA}}), \tag{7}$$

where $\mathbf{x}$ denotes the input sequence, $\mathbf{y}$ denotes the target output sequence, $y_1, y_2, \ldots, y_{|\mathbf{y}|}$ represents the tokens in $\mathbf{y}$, and $\boldsymbol{\theta}_{\text{LoRA}}$ denotes the trainable parameters introduced by LoRA on top of the frozen backbone model. This strategy made it feasible to complete the training on a single NVIDIA RTX 4090 GPU, with some essential hyperparameters summarized in Table 4. For reference recommendation, we employed stsb-distilroberta-base as the sentence encoder.

Table 4. Essential parameters and their settings for model training.

Model Type	Learning Rate	Training Epochs	Optimizer	LoRA Rank	LoRA α	Batch Size	Gradient Acc. Steps
bfloat16	5×10^{-4}	3	AdamW	8	8	1	16

4.1.3. Evaluation

We compare our method against three baselines: (1) the base model (e.g., Llama3-8B), (2) its official fine-tuned version (e.g., Llama3-8B-Instruct), and (3) a topic-guided knowledge distillation variant inspired by [10]. Specifically, we first partition the target domain into topics using a teacher model and then generate 3000 training instances per task to guide fine-tuning. For generative tasks, using cutting-edge LLMs such as GPT-4 as evaluators has become common practice [35]. Accordingly, we employ GPT-4-Turbo [1] to jointly evaluate the responses from our method and the three baselines on five dimensions: precision, relevance, understandability, professionalism, and logicality. Each dimension is rated on a 10-point scale. For classification tasks, model performance is measured using standard accuracy. The evaluation datasets are constructed from real-world examinations in their respective domains, with each task containing approximately 50–200 test cases. To evaluate the performance of the reference recommendation, we manually count the number of relevant references for each query. To evaluate the effectiveness of BOLT in terms of the reference recommendation, we conduct a manual assessment on the domain knowledge quiz (DKQ) task by calculating the proportion of references generated by BOLT that are genuinely relevant to the given queries. This is then compared against the proportion of verifiable references produced by DeepSeek-V3 (three references per query).

4.2. Main Results

4.2.1. Results Across All Domains

Table 5 presents the results of all models across three domains, including nine tasks. Figure 4 shows the scores of the Qwen2-7B series models across five dimensions on the DKQ task. The score for the DKQ task in Table 5 was obtained by calculating the average score across the five dimensions presented in Figure 4. Our proposed BOLT framework significantly enhances the performance of the base model and outperforms all baselines in most cases. The models evaluated in this experiment include different sizes within the same series (Qwen2-0.5B, Qwen2-1.5B, and Qwen2-7B). Following the application of BOLT, smaller models exhibit more pronounced relative performance gains, whereas larger models consistently achieve superior absolute performance, indicating a higher performance upper bound. It is worth noting that, across the Qwen2-0.5B model series, the BOLT framework underperforms the topic-guided approach on more than half of the tasks. We hypothesize that this may be due to the limited capacity of the model, under which the quantity of knowledge has a greater impact on performance than the way in which the knowledge is structured.

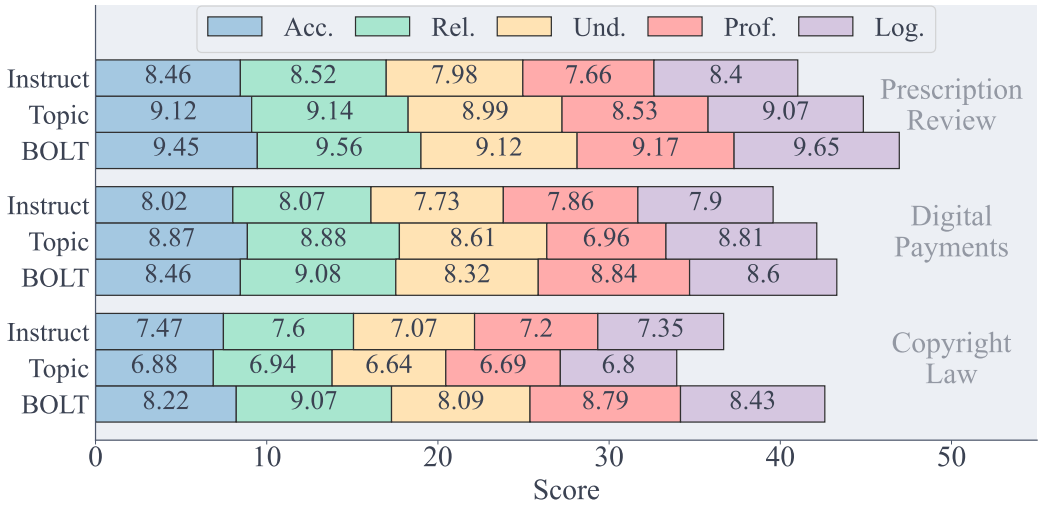


Figure 4. Performance details of the Qwen2-7B series models on the domain knowledge quiz task across three domains.

Table 5. Results of all models across three domains, covering a total of nine tasks. Each task is represented by its abbreviation. The optimal results within the same model series are highlighted in **bold**. All metrics are "the higher, the better".

Model	Prescription Review			Digital Payments			Copyright Law		
	DKQ	RDUA (%)	MCQs (%)	DKQ	SA (%)	MCQs (%)	DKQ	LKU (%)	LKA (%)
Llama3-8B	2.69	34.05	22.56	5.29	49.52	25.83	3.12	33.33	28.41
+Instruct	7.40	57.93	45.12	7.65	76.70	60.00	5.48	54.60	54.55
+Topic	8.91	63.41	66.46	7.70	81.55	**89.17**	5.74	79.89	76.70
+BOLT	**9.27**	**79.88**	**70.73**	**8.12**	**85.47**	87.50	**8.23**	**81.61**	**89.20**
Mistral-7B	3.73	30.49	18.29	6.10	27.18	26.67	2.20	20.69	17.05
+Instruct	5.49	46.95	28.66	6.07	66.02	28.33	5.36	23.56	27.27
+Topic	8.31	53.69	60.37	7.32	80.58	**85.00**	4.36	81.03	71.02
+BOLT	**9.03**	**73.78**	**65.85**	**8.66**	**82.69**	81.67	**8.10**	**82.18**	**85.80**
Baichuan2-7B	3.54	34.15	23.78	3.85	18.45	19.17	4.45	21.26	28.41
+Instruct	7.73	58.54	49.39	6.99	82.52	62.50	5.66	51.72	53.41
+Topic	8.11	65.85	68.90	7.86	90.29	83.33	6.16	77.01	73.86
+BOLT	**8.92**	**79.27**	**71.95**	**8.26**	**92.74**	**86.67**	**7.93**	**87.36**	**88.64**
Qwen2-0.5B	5.35	39.02	26.22	6.23	23.30	28.33	2.15	24.71	25.57
+Instruct	6.72	46.10	42.07	7.18	49.52	50.83	3.39	58.62	45.45
+Topic	7.08	48.17	**56.10**	**7.64**	**86.41**	65.00	3.01	**68.97**	**64.20**
+BOLT	**7.13**	**64.63**	54.27	7.26	83.80	**73.30**	**5.56**	64.94	57.39
Qwen2-1.5B	6.47	45.29	34.15	6.63	38.84	60.00	3.38	48.28	36.93
+Instruct	7.27	59.85	59.15	7.67	52.43	75.00	4.61	70.83	66.48
+Topic	7.60	67.68	66.46	7.32	81.55	**85.00**	4.43	73.56	69.32
+BOLT	**8.34**	**69.51**	**70.12**	**8.34**	**90.50**	82.50	**7.73**	**82.76**	**77.27**
Qwen2-7B	7.09	48.78	57.32	7.11	54.37	68.30	4.07	65.52	60.23
+Instruct	8.20	75.00	75.61	7.76	85.44	87.50	7.34	84.48	82.39
+Topic	8.97	76.22	**77.44**	8.62	87.38	87.50	6.79	86.21	81.25
+BOLT	**9.39**	**81.70**	76.22	**8.66**	**91.06**	**94.17**	**8.52**	**87.93**	**85.23**

4.2.2. Results on Reference Recommendation

Table 3 reports the results of the reference recommendation task. More than half of the recommended references are valid, suggesting strong relevance to the questions

and clearly outperforming the generation-based baseline. Notably, BOLT's performance is relatively stable across different repository sizes and appears to be more influenced by the proportion of valid references than by the absolute size of the repository. While BOLT does not guarantee perfect topical alignment of all recommended references, every citation it produces is authentic and verifiable. In contrast, generation-based approaches built on large language models often yield references that appear contextually relevant but are prone to hallucinations, citing works that do not actually exist. We contend that such hallucinations are substantially more detrimental to user experience than occasional mismatches in topical relevance, thereby underscoring the practical advantage of BOLT's matching-based strategy.

4.3. Analysis

Finding 1: The scope estimation module in BOLT reliably provides accurate assessments of data volume. During the data distillation stage, we construct five training datasets of varying sizes: 25%, 50%, 75%, 125%, and 150% of the standard data volume (as shown in Table 2) across three domains. These datasets are used to train the Llama3-8B base model. Model performance is evaluated on three tasks: multiple-choice questions (MCQs) in prescription review and digital payments, and legal knowledge application (LKA) in copyright law. The results are shown in Figure 5. Performance improves markedly as data size increases from 25% to 100% across all domains, but it plateaus beyond 100%, indicating diminishing returns. This saturation trend underscores the role of the scope estimation step in identifying an efficient data volume that balances training cost and model effectiveness.

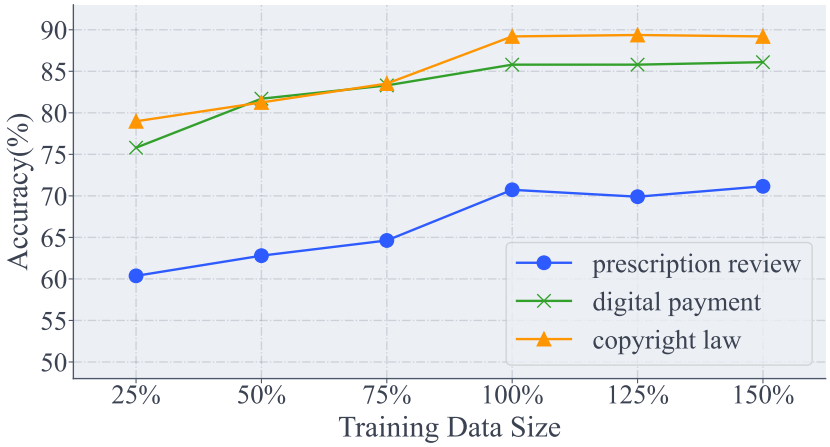


Figure 5. Model performance across domains with varying training data sizes.

Finding 2: Critique-driven refinement contributes to improving the structural organization of the knowledge tree. We prompt the teacher model $\mathcal{G}_2$ to explain its refinement process, with two representative modifications illustrated in Figure 6. The model exhibits a reasonable capacity to assess structural coherence and apply targeted adjustments that enhance the organization of the knowledge tree. However, the impact of critique-driven refinement on the overall quality of the knowledge tree remains underexplored and calls for more fine-grained analysis, which presents a promising direction for future research.

Finding 3: Curriculum learning enhances knowledge acquisition and strengthens model performance. We randomly shuffled the training data obtained by data distillation and used them to fine-tune the Llama3-8B base model. The model was then evaluated on classification tasks across three domains, with the results shown in Table 6. Curriculum learning outperforms random-order training on most tasks, highlighting its overall advantage. However, in the Legal Knowledge Understanding (LKU) task, the randomly trained

model achieves better performance, likely because LKU focuses on conceptual recall (e.g., legal provisions) rather than structural understanding of the domain.

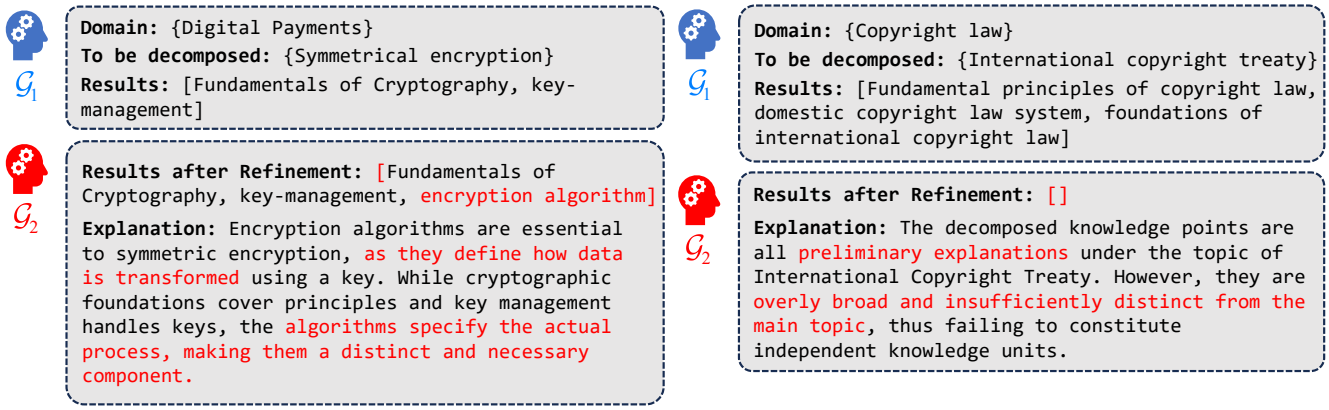


Figure 6. Representative modifications from critique-driven refinement.

Table 6. Comparison between curriculum learning and random-order training.

Domain	Task	Random-Order	Curriculum
Prescription Review	RDUA (%)	78.05	**79.88**
	MCQs (%)	65.24	**70.73**
Digital Payments	SA (%)	80.45	**85.47**
	MCQs (%)	86.67	**87.50**
Copyright Law	LKU (%)	**83.91**	81.86
	LKA (%)	85.23	**89.20**

Finding 4: BOLT enables the model to maintain stable performance even in high-uncertainty environments. To evaluate the stability of model outputs, we conducted an experiment on the LKU task, in which each test sample was used to generate 10 outputs under varying temperature settings. The temperature parameter controls the randomness of model outputs: lower values make responses more deterministic, while higher values increase diversity and may cause instability when exceeding one. We then computed the accuracy and information entropy [42] of these outputs, with the results presented in Figure 7. As the temperature increases, the accuracy of the baseline model (i.e., the officially fine-tuned version) declines, whereas the BOLT model consistently maintains high accuracy, exhibiting robustness to temperature changes. Furthermore, higher temperature settings lead to a more dispersed response distribution in the baseline model, while the BOLT model's outputs remain notably more concentrated. These results indicate that BOLT enhances the stability of model outputs without compromising performance.

Finding 5: The extracted knowledge varies across different teacher LLMs, leading to discrepancies in model performance. We incorporated GPT-4o mini [1] and GLM-4-Air [43] as two additional teacher models into the framework, with prescription review designated as the target domain. Subsequently, we trained the Llama3-8B base model using the generated dataset. Table 7 presents the comparison results, revealing that knowledge extracted from different teacher LLMs leads to varying model performance across distinct tasks. This highlights the importance and potential of further research in effectively leveraging the complementary strengths of diverse teacher models.

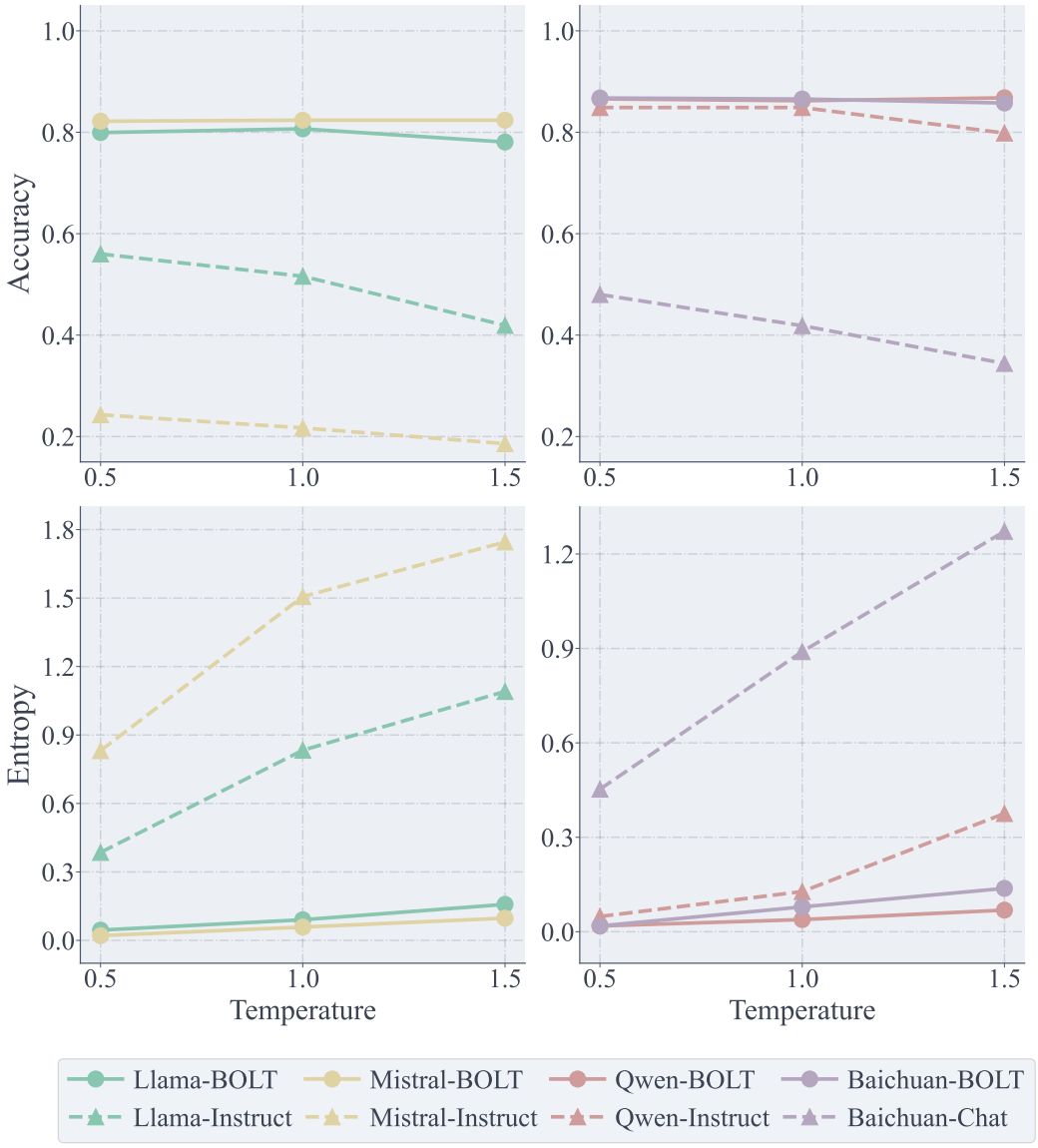


Figure 7. Performance and stability comparison of different models under varying temperature settings.

Table 7. Comparison of the impact of different teacher LLMs on the performance across three tasks in the prescription review domain.

Teacher LLM	DKQ	RDUA (%)	MCQs (%)
DeepSeek-V3	9.39	81.70	76.22
GPT-4o mini	9.34	74.10	71.59
GLM-4-Air	9.27	76.00	77.37

5. Discussion

The experimental results demonstrate that BOLT effectively addresses critical challenges in adapting open-source LLMs to specialized domains, including domain knowledge gaps, data scarcity, and reference hallucination. By systematically estimating domain scope and constructing hierarchical knowledge trees, BOLT captures the structural characteristics of domain knowledge, which are often overlooked by existing adaptation methods. Compared to conventional approaches, BOLT provides more principled guidance for model training, reduces data homogenization, and improves the accuracy of reference recommendation. Furthermore, its curriculum-guided knowledge distillation from advanced

teacher LLMs ensures efficient and robust domain adaptation, mitigating issues such as catastrophic forgetting and unstructured knowledge integration.

BOLT's design enables broad applicability across diverse domain-specific tasks. The hierarchical knowledge modeling and curriculum-based learning facilitate coherent acquisition of prerequisite concepts, making it particularly effective for scientific question answering, automated content summarization, and specialized information extraction. Furthermore, the matching-based reference strategy significantly reduces hallucination, highlighting its utility in scenarios requiring high-precision reference recommendations. These properties suggest that BOLT can serve as a generalizable framework for various domain-adaptive LLM applications where structured knowledge and reliable outputs are crucial.

Despite its advantages, BOLT has potential limitations. In particular, its performance ceiling is inherently constrained by the capabilities of teacher LLMs. If teacher models lack comprehensive domain knowledge or exhibit biases, the adaptation of the student model will be limited accordingly. Furthermore, the reliance on hierarchical trees may not fully capture highly interdependent or nonlinear relationships in certain domains, and the efficiency of knowledge tree construction could be further improved. Understanding and addressing these limitations are essential to extend the framework to broader or more complex settings.

Several promising avenues for future research arise from these observations. First, exploring multi-model knowledge distillation could enable BOLT to integrate the complementary strengths of multiple teacher LLMs, enhancing domain adaptation performance. Second, investigating alternative knowledge representations, such as graphs, ontologies, or hypergraphs, can better capture complex interdependencies within domain knowledge. Third, improving the efficiency and expertise of knowledge structure construction, including automated methods for high-quality domain knowledge acquisition, could further strengthen BOLT's practicality. Future work could also evaluate BOLT in highly dynamic or interdependent domains to assess its scalability and real-world applicability.

6. Conclusions

This paper presents BOLT, a modular end-to-end framework designed to adapt open-source language models to domain-specific scenarios. By employing a multi-stage pipeline, BOLT systematically estimates the scope of domain knowledge and structures it into a hierarchical knowledge tree, enabling more effective and targeted knowledge distillation. To address reference hallucination at its root, BOLT replaces generative mechanisms with a matching-based approach, substantially mitigating hallucination risks. Comprehensive experiments across diverse domains and open-source models demonstrate that BOLT consistently enhances domain-specific performance while suppressing reference hallucination. Further analysis confirms the effectiveness of each framework component, highlighting BOLT's ability to improve model stability and robustness and providing valuable insights and inspiration for future research on domain adaptation in large language models.

Author Contributions: Conceptualization, R.L.; methodology, R.L. and Z.F.; software, R.L. and Z.F.; validation, R.L., Z.F. and G.W.; formal analysis, R.L. and Z.F.; investigation, G.W. and R.L.; resources, Q.S.; data curation, R.L. and Z.F.; writing—original draft preparation, R.L.; writing—review and editing, R.L., Z.F. and Q.S.; visualization, R.L.; supervision, Q.S.; project administration, R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Fundamental Research Funds for the Central Universities (2120230311).

Data Availability Statement: All research data could be provided via email 2333104@tongji.edu.cn.

Acknowledgments: The authors gratefully acknowledge the support provided by the School of Computer Science and Technology, Tongji University. During the preparation of this manuscript and the execution of the experiments, the authors used OpenAI's ChatGPT (GPT-5, 2025 release) for language refinement, stylistic editing, and to improve prompts. The authors have carefully reviewed and edited all outputs and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

For the sake of transparency and reproducibility, this Appendix documents the prompts employed in our study when interacting with teacher models during the construction of the knowledge tree. Although the original prompts were formulated in Chinese, we present their English translations here to ensure accessibility to a broader readership.

Appendix A.1. Prompt Used for Knowledge Decomposition

```
# Task
You are now an expert in the field of {domain}, tasked with designing a learning path on {
    topic} for a learner.

Based on the knowledge structure of {topic} within {domain}, analyze the prerequisite
    knowledge required to master {topic}, and output around {width} knowledge points that
    are meaningful for discussion and learning. The knowledge points should be logically
    distinct and consistent with textbook-style pedagogical logic. If {topic} itself is
    already very fundamental, or there is no further need to break it down, please clearly
    state this and briefly explain the reason.

# Requirements
Please follow these requirements:
1. The output knowledge points should be prerequisite or core foundational content for
    learning {topic}.
2. Each knowledge point must be independently discussable or expandable, not merely a term.
3. Do not write definitions or explanations, only list the names.
4. The number of knowledge points should be around {width}; slight adjustments are allowed
    if necessary.
5. Ensure a reasonable structure; do not simply enumerate terms.
6. If {topic} itself is already very fundamental, or there is no further need to break it
    down, please clearly state this and briefly explain the reason.
7. The knowledge points should be foundational for learning {topic}, rather than simply
    subdivisions of {topic}.

# Output Format Specification
‘‘‘json
{
  "domain": "Domain Name",
  "topic": "Topic to be decomposed",
  "prerequisites": [
    {
      "name": "Knowledge Point 1",
      "significance": "Explanation of its necessity"
    },
    // Other knowledge points...
  ],
  "stop_reason": "Reason if decomposition is unnecessary"
}
```

Appendix A.2. Prompt Used for Critique-Driven Refinement

```
# Task
As an expert in the field of {domain}, you are responsible for conducting a rigorous review
of the knowledge system design regarding the topic of {topic}. You must apply critical
thinking to evaluate all submitted content.

# Review Rules
1. Necessity Check:
   Clear the knowledge point list in the following cases:
     - The topic is already an atomic-level concept (e.g., "real number", "alphabet")
     - The generated knowledge points have little relevance to the topic
     - The knowledge points are self-repetitive

2. Completeness Check:
   Supplement missing elements if any of the following are absent:
     - Core prerequisite knowledge recognized in the field (e.g., learning calculus must
     include limits)
     - Key methodologies supporting the topic (e.g., gradient computation in machine
     learning)

3. Conciseness Check:
   Remove redundant items in the following cases:
     - Multiple knowledge points essentially discuss the same concept
     - A higher-level concept already encompasses the lower-level concepts

# Input and Output Content
    - Input: A knowledge branch containing a topic and several knowledge points
    - Output: The processed knowledge points along with the reasoning for such processing

# Output Format Specification
‘‘‘json
{
  "domain": "Domain Name",
  "topic": "Topic to be evaluated",
  "knowledge_points": ["Knowledge Point 1", "Knowledge Point 2"]  // or an empty list if no
     points remain
  "reason": "..."
}
```

References

1. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774. [CrossRef]
2. Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. Deepseek-v3 technical report. *arXiv* **2024**, arXiv:2412.19437.
3. Zhang, H.; Chen, J.; Jiang, F.; Yu, F.; Chen, Z.; Li, J.; Chen, G.; Wu, X.; Zhang, Z.; Xiao, Q.; et al. Huatuogpt, towards taming language model to be a doctor. *arXiv* **2023**, arXiv:2305.15075. [CrossRef]
4. Cui, J.; Li, Z.; Yan, Y.; Chen, B.; Yuan, L. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv* **2023** , arXiv:2306.16092
5. Zhang, X.; Yang, Q. Xuanyuan 2.0: A large chinese financial chat model with hundreds of billions parameters. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, Birmingham, UK, 21–25 October 2023; Association for Computing Machinery: New York, NY, USA, 2023; pp. 4435–4439.
6. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv* **2023**, arXiv:2307.09288. [CrossRef]
7. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; Casas, D.d.l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. *arXiv* **2023**, arXiv:2310.06825. [CrossRef]
8. Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. Qwen technical report. *arXiv* **2023**, arXiv:2309.16609. [CrossRef]

9. Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N.A.; Khashabi, D.; Hajishirzi, H. Self-instruct: Aligning language models with self-generated instructions. *arXiv* **2022**, arXiv:2212.10560.
10. Ding, N.; Chen, Y.; Xu, B.; Qin, Y.; Hu, S.; Liu, Z.; Sun, M.; Zhou, B. Enhancing Chat Language Models by Scaling High-quality Instructional Conversations. *arXiv* **2023**, arXiv:2305.14233. [CrossRef]
11. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.J.; Madotto, A.; Fung, P. Survey of hallucination in natural language generation. *ACM Comput. Surv.* **2023**, *55*, 1–38. [CrossRef]
12. Day, T. A preliminary investigation of fake peer-reviewed citations and references generated by ChatGPT. *Prof. Geogr.* **2023**, *75*, 1024–1027. [CrossRef]
13. Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H.P.D.O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. Evaluating large language models trained on code. *arXiv* **2021**, arXiv:2107.03374. [CrossRef]
14. Wu, S.; Irsoy, O.; Lu, S.; Dabravolski, V.; Dredze, M.; Gehrmann, S.; Kambadur, P.; Rosenberg, D.; Mann, G. Bloomberggpt: A large language model for finance. *arXiv* **2023**, arXiv:2303.17564. [CrossRef]
15. Li, Y.; Bubeck, S.; Eldan, R.; Del Giorno, A.; Gunasekar, S.; Lee, Y.T. Textbooks are all you need ii: Phi-1.5 technical report. *arXiv* **2023**, arXiv:2309.05463. [CrossRef]
16. Yu, Y.; Zhuang, Y.; Zhang, J.; Meng, Y.; Ratner, A.J.; Krishna, R.; Shen, J.; Zhang, C. Large language model as attributed training data generator: A tale of diversity and bias. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 55734–55784.
17. Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; Hashimoto, T.B. Stanford Alpaca: An iNstruction-Following Llama Model. 2023. Available online: https://www.kaggle.com/discussions/general/394598 (accessed on 23 September 2025).
18. Xu, C.; Sun, Q.; Zheng, K.; Geng, X.; Zhao, P.; Feng, J.; Tao, C.; Jiang, D. Wizardlm: Empowering large language models to follow complex instructions. *arXiv* **2023**, arXiv:2304.12244. [CrossRef]
19. Chen, H.; Quan, X.; Chen, H.; Yan, M.; Zhang, J. Knowledge distillation for closed-source language models. *arXiv* **2024**, arXiv:2401.07013. [CrossRef]
20. Liang, C.; Zuo, S.; Zhang, Q.; He, P.; Chen, W.; Zhao, T. Less is more: Task-aware layer-wise distillation for language model compression. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; PMLR; pp. 20852–20867.
21. Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; Zhang, D. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv* **2023**, arXiv:2308.09583. [CrossRef]
22. Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C.D.; Ermon, S.; Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 53728–53741.
23. Xu, X.; Li, M.; Tao, C.; Shen, T.; Cheng, R.; Li, J.; Xu, C.; Tao, D.; Zhou, T. A survey on knowledge distillation of large language models. *arXiv* **2024**, arXiv:2402.13116.
24. Wagner, M.W.; Ertl-Wagner, B.B. Accuracy of information and references using ChatGPT-3 for retrieval of clinical radiological information. *Can. Assoc. Radiol. J.* **2024**, *75*, 69–73.
25. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* **2023**, *43*, 42. [CrossRef]
26. Gao, L.; Biderman, S.; Black, S.; Golding, L.; Hoppe, T.; Foster, C.; Phang, J.; He, H.; Thite, A.; Nabeshima, N.; et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv* **2020**, arXiv:2101.00027. [CrossRef]
27. Gunasekar, S.; Zhang, Y.; Aneja, J.; Mendes, C.C.T.; Del Giorno, A.; Gopi, S.; Javaheripi, M.; Kauffmann, P.; de Rosa, G.; Saarikivi, O.; et al. Textbooks are all you need. *arXiv* **2023**, arXiv:2306.11644. [PubMed]
28. Sinitsin, A.; Plokhotnyuk, V.; Pyrkin, D.; Popov, S.; Babenko, A. Editable neural networks. *arXiv* **2020**, arXiv:2004.00345. [CrossRef]
29. Zhang, N.; Yao, Y.; Tian, B.; Wang, P.; Deng, S.; Wang, M.; Xi, Z.; Mao, S.; Zhang, J.; Ni, Y.; et al. A comprehensive study of knowledge editing for large language models. *arXiv* **2024**, arXiv:2401.01286. [CrossRef]
30. Liu, B.; Ash, J.; Goel, S.; Krishnamurthy, A.; Zhang, C. Exposing attention glitches with flip-flop language modeling. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 25549–25583.
31. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 27730–27744.
32. Lee, N.; Ping, W.; Xu, P.; Patwary, M.; Fung, P.N.; Shoeybi, M.; Catanzaro, B. Factuality enhanced language models for open-ended text generation. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 34586–34599.
33. Li, K.; Patel, O.; Viégas, F.; Pfister, H.; Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 41451–41530.
34. Shi, W.; Han, X.; Lewis, M.; Tsvetkov, Y.; Zettlemoyer, L.; Yih, S.W.t. Trusting your evidence: Hallucinate less with context-aware decoding. *arXiv* **2023**, arXiv:2305.14739. [CrossRef]

35. Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; Yu, L.; et al. Lima: Less is more for alignment. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 55006–55021.
36. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
37. Jaccard, P. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull. Soc. Vaudoise Sci. Nat.* **1901**, *37*, 547–579.
38. Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. The Llama 3 herd of models. *arXiv* **2024**, arXiv:2407.21783. [CrossRef]
39. Yang, A.; Xiao, B.; Wang, B.; Zhang, B.; Bian, C.; Yin, C.; Lv, C.; Pan, D.; Wang, D.; Yan, D.; et al. Baichuan 2: Open large-scale language models. *arXiv* **2023**, arXiv:2309.10305. [CrossRef]
40. Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; et al. Qwen2 Technical Report. *arXiv* **2024**, arXiv:2407.10671.
41. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-rank adaptation of large language models. *arXiv* **2021**, arXiv:2106.09685.
42. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
43. GLM, T.; Zeng, A.; Xu, B.; Wang, B.; Zhang, C.; Yin, D.; Zhang, D.; Rojas, D.; Feng, G.; Zhao, H.; et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv* **2024**, arXiv:2406.12793.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.