BLADE: Enhancing Black-Box Large Language Models with Small Domain-Specific Models

Haitao LI[1,2] **Qingyao AI**[1,2*], **Jia CHEN**[3], **Qian DONG**[1,2], **Zhijing WU**[4], **Yiqun LIU**[1,2†]

[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2] Institute for Internet Judiciary, Tsinghua University, Beijing, China
[3] Xiaohongshu Inc
[4] School of Computer Science and Technology, Beijing Institute of Technology
liht22@mails.tsinghua.edu.cn

Abstract

Large Language Models (LLMs) like ChatGPT and GPT-4 are versatile and capable of addressing open-domain question-answering(QA) tasks effectively. However, general LLMs, which are developed on open-domain data, may lack the domain-specific knowledge essential for tasks in vertical domains, such as legal, medical, etc. To address this issue, previous approaches either conduct continuous pre-training with domain-specific data or employ retrieval augmentation to support general LLMs in handling QA tasks. Unfortunately, these strategies are either cost-intensive or unreliable in practical applications. To this end, we present a novel framework named BLADE, which enhances **B**lack-box **LA**rge language models with small **D**omain-sp**E**cific models. BLADE consists of a black-box LLM and a small domain-specific LM. The small LM preserves domain-specific knowledge and offers specialized insights, while the general LLM contributes robust language comprehension and reasoning capabilities. Specifically, our method involves three steps: 1) pre-training the small LM with domain-specific data, 2) fine-tuning this model using knowledge instruction data, and 3) joint Bayesian optimization of the general LLM and the small LM. In our experiments, we verify the effectiveness of BLADE on diverse LLMs and datasets across different domains. This shows the potential of BLADE as an effective and cost-efficient solution in adapting general LLMs for vertical domains.

Code — https://github.com/CSHaitao/BLADE

Introduction

Recently, large language models (LLMs) have attracted considerable attention in both academia and industry (OpenAI 2023; Zeng et al. 2022). These models, driven by expansive neural networks and trained on extensive data sets, exhibit remarkable ability in comprehending and generating natural language. The wide application of LLMs trained with open-domain data, denoted in this paper as *General LLMs*, has profoundly impacted various aspects of daily life and professional environments.

Despite their superior capabilities, large language models often face challenges in addressing QA tasks of vertical domains (e.g., medicine, legal) that require access to a large amount of domain knowledge (Chalkidis 2023; Cheng, Huang, and Wei 2023). How to adapt general LLMs for domain-specific applications has become an important problem for the research community (Arefeen, Debnath, and Chakradhar 2023).

Existing methods for adapting general LLMs to specific domains can be broadly divided into two main categories: domain data continuous pre-training and retrieval augmentation. Continuous pre-training involves infusing domain knowledge into general LLMs by training them on a domain-specific corpus (Aharoni and Goldberg 2020; Sachidananda, Kessler, and Lai 2021). While straightforward, this paradigm requires direct access to large-scale domain data and LLM parameters, which are not available in many conditions. Also, even with access to general LLM parameters and sufficient domain-specific data, directly tuning a general LLM (such as GPT-4) can be prohibitively expensive and poses a risk of overfitting. Aware of these challenges, researchers propose retrieval augmentation as a new paradigm, aiming to enhance general LLMs by leveraging their in-context learning ability (Shi et al. 2023). It involves first using a text retriever to find relevant content from the domain corpus, which is then incorporated into the LLM's input to aid in understanding domain-specific knowledge. However, there may exist two problems in this paradigm. For example, retrievers primarily rely on exact matches or semantic similarity, lacking inferential capabilities. This limitation means they may not always retrieve documents that fully address specific queries. Additionally, retrievers can only provide content that is already available in the corpus and lack the ability to integrate and summarize different information.

When humans face questions in new domains, besides taking classes (i.e., continuous pre-training) or conducting online searches via platforms like Google (i.e., retrieval augmentation), a more direct and practical approach is to seek advice from experts possessing domain-specific knowledge. With this idea in mind, we present BLADE, a novel paradigm where the general LLM is viewed as a black box and the small domain-specific LM is added as a tuneable module. BLADE leverages the superior language comprehension and logical

*Corresponding authors.
†Corresponding authors.

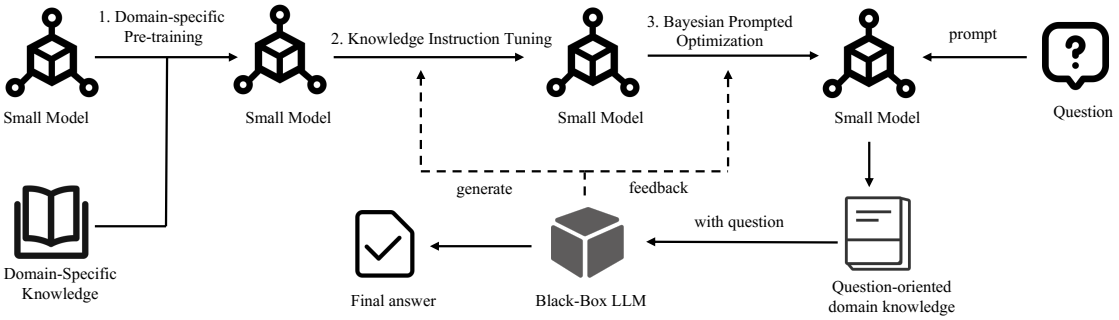

Figure 1: The workflow of BLADE. There are three steps in BLADE: (1) Domain-specific Pre-training. (2) Knowledge Instruction Tuning. (3) Bayesian Prompted Optimization.

reasoning capabilities of the general LLM, while also incorporating the domain-specific expertise and precision provided by the smaller, domain-focused LM. This approach includes Domain-specific Pretraining (DP) of the smaller LM and introduces two strategies: Knowledge Instruction Tuning (KIT) and Bayesian Prompted Optimization (BPO). Knowledge Instruction Tuning leverages general LLMs to generate pseudo data, which refines the smaller LM, equipping it with instruct-following ability. Then, the Bayesian Prompted Optimization aligns the output of small LMs with general LLMs using derivative-free optimization on soft embeddings. We verify the effectiveness of our method on public legal and medical benchmarks, which require a large amount of domain knowledge and strong reasoning abilities. The experiments show that BLADE can improve the performance of diverse general LLMs across legal and medical benchmarks. Compared with existing retrieval augmentation methods, the domain-specific small LM can generate more in-depth, comprehensive, and contextually appropriate external knowledge. This capability significantly improves the application of general LLMs in specialized domains.

Related Work

Domain Adaptation of LLMs

The domain adaptation of LLMs is an extensively researched field. Researchers have explored methods such as continuous pre-training, and retrieval augmentation to improve the performance of LLMs in a specific domain (Aharoni and Goldberg 2020; Li et al. 2024b). The most intuitive approach is continuous pre-training a language model on domain-specific corpora. Previous research has focused on data selection (Aharoni and Goldberg 2020) as well as adjusting or extending tokenizers (Sachidananda, Kessler, and Lai 2021) to achieve superior performance in the target domain. Despite being effective, fully training these models is often impractical due to extensive computational costs. Additionally, high-quality LLMs can only accessed through the inference API as black boxes. One possible alternative is to answer domain-specific questions by retrieving relevant information from a specific knowledge base (Borgeaud et al. 2022; Shi et al. 2023; Lewis, Stenetorp, and Riedel 2020). Retrieval augmentation has been shown to be effective in improving performance on various tasks. For instance, RETRO (Borgeaud et al. 2022) modifies the model architecture to incorporate retrieved text. Furthermore, REPLUG (Shi et al. 2023) treats the language model as a black box and enhances it using a retrieval model. Additionally, recent research has explored substituting traditional document retrievers with large language model generators (Yu et al. 2022; Li et al. 2023a; Sun et al. 2022b).

Method

Overview

Figure 1 illustrates the overall framework of BLADE. To be specific, BLADE solves domain-specific tasks through a collaborative approach between general black-box LLMs and small white-box LMs. General black-box LLMs, such as ChatGPT and GLM-130B, excel in reasoning and inference but are usually expensive and difficult to fine-tune in downstream applications. Conversely, small white-box LMs may be weak in sufficient reasoning ability, but can easily be adapted to memorize domain-specific knowledge. Inspired by this observation, our BLADE framework proposes to build a small LM to capture domain-specific knowledge and use it to generate tailored knowledge and prompts for general LLMs to synthesize and create responses for user queries.

Domain-specific Pre-training (DP)

A variety of studies have shown that pre-trained language models implicitly capture high-quality substantial knowledge hidden in the training data (Wang et al. 2022; Liu et al. 2021; Lewis et al. 2020; Joshi et al. 2020). This knowledge can be elicited from the language model through instructional prompts (Liu et al. 2021). Therefore, we first inject domain-specific knowledge to our small LMs via Domain-specific Pre-training (DP). Specifically, given domain-specific unsupervised text $T = \{t_1, t_2, ..., t_n\}$, we optimize the model by maximizing the following training objective:

$$G(T) = \sum_i \log P\left(t_i \mid t_{i-k}, \ldots, t_{i-1}; \Theta\right), \qquad (1)$$

where Θ is the parameter of the model. P is the conditional probability of generating the current token based on the previous tokens.

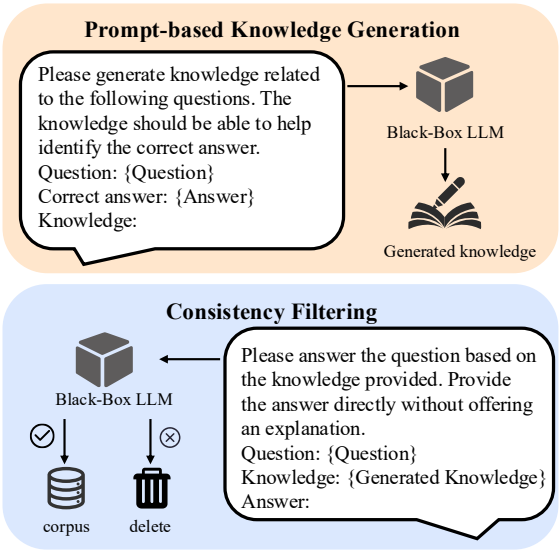

Figure 2: The process of generating data for Knowledge Instruction Tuning. Only knowledge that can help the black-box LLM correctly answer a question is reserved.

Knowledge Instruction Tuning (KIT)

After domain-specific pre-training, instruction tuning is often employed to equip the LMs with the ability to follow specific instructions. However the annotation of high quality instruction fine-tuning data is usually expensive. We introduce Knowledge Instruction Tuning (KIT), which focuses on leveraging LLMs-as-Judges (Li et al. 2024c) as annotators to enhance the instruction-following capabilities of smaller language models. This process enables the small LM to leverage its inherent knowledge for a particular task and provide useful information for downstream systems.

To be specific, KIT consists of three components: Prompt-based Knowledge Generation, Consistency Filtering, and Instruction Tuning. Figure 2 show the process of generating data of KIT. During Prompt-based Knowledge Generation, we use the training data from a specific domain or task (e.g., QA) to formulate training query-answer pairs. Subsequently, a general black-box LLM (e.g., ChatGPT, GPT4) is employed to generate explanations that assists in accurately answering the questions. The process creates instruction-tuning data for the small LMs where the instruction is the original training query and the desired output is the explanations we need to answer the query correctly. Since the general black-box LLM in KIT may not possess enough knowledge to answer all training query-answer pairs, specifically when the data is collected from domain-specific tasks, we filtered the generated instruction data based on round consistency, i.e., whether the general black-box LLM can produce the correct answer based on the knowledge they extracted. Consistency Filtering has been shown to be effective for query generation in QA tasks and various information retrieval tasks (Dai et al. 2022; Lewis et al. 2021). The instruction prompts are structured as depicted in Figure 2. We filter the generated knowledge and only retain the information that can help the LLM to answer questions correctly. The refined data is then used to fine-tune the small LM and teach it to produce knowledge for a given query instead of predicting the next token.

It is worth noting that the goal of the process is to extracte instruction-following capabilities from LLMs into small LM, rather than extracting specific domain knowledge from LLMs. We propose a method to efficiently generate fine-tuned instructions without manual annotation.

Bayesian Prompted Optimization (BPO)

Now we propose the Bayesian Prompted Optimization (BPO) method that teachs the domain-specific small LM to communicate with the general LLMs. Figure 3 illustrates the process of Bayesian Prompted Optimization (BPO). To be specific, the optimization objective is to enhance the performance $f(\cdot)$ of the general LLM on domain-specific tasks. Consider an example (X, Y) from the dataset $\mathcal{T}_t$. Let k represents the domain knowledge that is specific to the query X. In our framework, k is generated by the small domain-specific LM $g(\cdot)$. Let $h(\cdot, \cdot)$ be the evaluation metric for output $f(k, X)$ and ground truth Y. For example, in multiple-choice tasks, $h(\cdot, \cdot)$ can be accuracy. The optimization objective is to maximize the performance with appropriate knowledge, i.e.,

$$\max_{k} \mathbb{E}_{(X,Y)\sim\mathcal{T}_t} h(f([k; X]), Y), \text{ s.t. } k = g(X), \quad (2)$$

As discussed by Chen et al (Chen et al. 2023), the above problems can be seen as a combinatorial optimization with structural constraints. Since $f(\cdot)$ is a black-box model, traditional optimization via backpropagation is not feasible. Therefore, we apply derivative-free optimization to refine the soft prompt $\boldsymbol{p_h}$ on small model $g(\cdot)$. Specifically, we concatenate n soft tokens $p_{h_1:h_n} \in \mathbb{R}^D$ with input queries X as inputs to the small model to generate the domain knowledge $k = g(p_{h_1:h_n}, X)$. Therefore, our objective is to identify the optimal soft prompt:

$$\boldsymbol{p}^*_{\boldsymbol{h}} = \arg\max_{\boldsymbol{p_h}\in\mathbb{R}^D} \mathbb{E}_{(X,Y)\sim\mathcal{T}_t} h(f([g(\boldsymbol{p_h}, X); X]), Y). \quad (3)$$

Although the original optimization problem is transformed into a feasible continuous optimization problem, derivative-free black-box optimization remains challenging due to the high dimensionality of the optimized soft prompt. To address this problem, we propose to optimize a lower dimensional vector $\boldsymbol{p} \in \mathbb{R}^d$ where $d \ll D$ and apply a random projection $A \in \mathbb{R}^{d\times D}$ to project $\boldsymbol{p}$ into the original space. The intuitions behind this are two-fold: (1) As shown by previous studies (Sun et al. 2022a; Hu et al. 2021), the knowledge encoded by pre-trained LMs usually has low dimensionality by nature, which means that effective optimization doesn't require a full exploration of the high-dimensional parameter space; (2) According to Johnson-Lindenstrauss Lemma (Kleinberg 1997), the random projection is distance-preserving, so the kernel similarity of the low dimensional knowledge representation (i.e., p) can be maintained after the projection. Thus, the optimization objective is transformed into the following formula:

$$\boldsymbol{p}^* = \arg\max_{A\boldsymbol{p}\in\mathbb{R}^D} \mathbb{E}_{(X,Y)\sim\mathcal{T}_t} h(f([g(A\boldsymbol{p}, X); X]), Y). \quad (4)$$

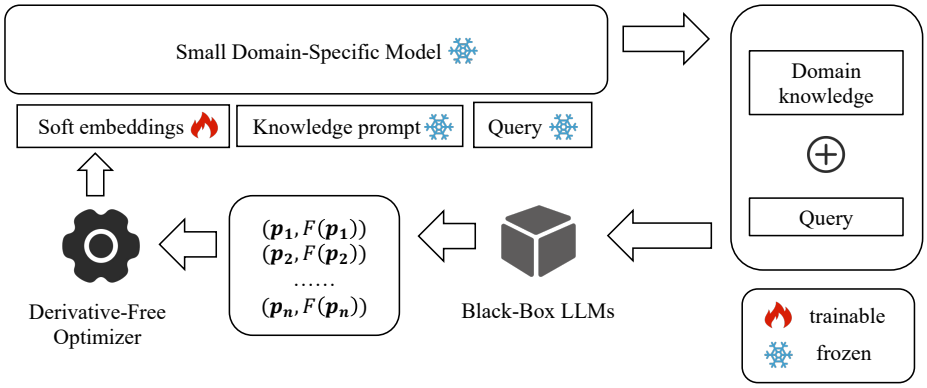

Figure 3: Illustration of the Bayesian Prompted Optimization where only soft embeddings are trainable. $F(\boldsymbol{p})$ is the objective score corresponding to soft embedding $\boldsymbol{p}$.

We employ Bayesian optimization (BO) (Frazier 2018) to handle the above optimization. BO is an effective technique for updating the posterior distribution of the objective function by iteratively incorporating new sample points. To be specific, we first define the objective function as $F(\boldsymbol{p}) = \mathbb{E}_{(X,Y)\sim\mathcal{T}_i} h(f([g(A\boldsymbol{p}, X); X]), Y)$. Then, we employ the Gaussian Process (GP) as the prior to estimate the distribution of $F(\cdot)$, i.e.,

$$F(\boldsymbol{p}) \sim GP(\mu, \sigma^2), \tag{5}$$

where μ is the mean function and σ^2 is the variance function. This GP can be updated iteratively as the optimization process progresses, incorporating new observations to better approximate the true function and reduce uncertainty w.r.t. its behavior. For each $\boldsymbol{p_i}$, we can obtain a score $F(\boldsymbol{p_i})$. Let $\mathcal{D}$ denote all collected data in previous BO steps, i.e., $\mathcal{D} = \{(\boldsymbol{p_1}, F(\boldsymbol{p_1})), ..., (\boldsymbol{p_n}, F(\boldsymbol{p_n}))\}$. Then the μ and σ^2 of the GP can be updated as follows:

$$\mu(\boldsymbol{p}) = c(\boldsymbol{p}, \boldsymbol{P}) \left(\boldsymbol{C} + \sigma_n^2 \boldsymbol{I}\right)^{-1} F(\boldsymbol{P}), \tag{6}$$

$$\sigma^2(\boldsymbol{p}) = c(\boldsymbol{p}, \boldsymbol{p}) - c(\boldsymbol{p}, \boldsymbol{P}) \left(\boldsymbol{C} + \sigma_n^2 \boldsymbol{I}\right)^{-1} c(\boldsymbol{P}, \boldsymbol{p}), \tag{7}$$

where $\boldsymbol{P} = [\boldsymbol{p_1}, ..., \boldsymbol{p_n}]$ and p_i is the soft embedding in the ith exploration, $c(\cdot, \cdot)$ is the covariance function, $\boldsymbol{C}$ is the covariance matrix of $\boldsymbol{P}$, σ_n is the noise variance, $\boldsymbol{I}$ represents the identity matrix. $\boldsymbol{p_1}$ is randomly initialized. After finishing an iteration, we employ Expected Improvement (EI) to find the next $\boldsymbol{p_{n+1}}$. The Expected Improvement (EI) is a popular acquisition function that balances exploration and exploitation. It quantifies the potential improvement over the current best observed value. Formally, the next soft prompt $\boldsymbol{p_{n+1}}$ is defined as follows:

$$\boldsymbol{p_{n+1}} \in \arg\max_{\boldsymbol{p}\in\mathbb{R}^d} \mathbb{E}_{F(\boldsymbol{p})} \left[\max \left\{ 0, F(\boldsymbol{p}) - \max_{i\in[n]} F\left(\boldsymbol{p}_i\right) \right\} \right], \tag{8}$$

In practice, we employ an evolutionary search algorithm known as CMA-ES (Hansen 2016) as the optimization method to identify the most effective soft prompts. When obtaining $\boldsymbol{p_{n+1}}$, we evaluate performance $F(\boldsymbol{p_{n+1}})$ of BLADE on the training batch. Subsequently, the pair $(\boldsymbol{p_{n+1}}, F(\boldsymbol{p_{n+1}}))$is incorporated into the $\mathcal{D}$ to update μ and σ^2. This process is performed iteratively until convergence (effectiveness gains less than a threshold for a given number of steps) or reaches the maximum iteration number.

Experiment

Datasets and Metrics

We conduct extensive experiments on two widely used datasets in the legal and medical domains.

JEC-QA (Zhong et al. 2020) is the largest Chinese multiple-choice dataset in the legal domain. The legal questions in JEC-QA require high-level reasoning ability and are divided into two types: Knowledge-Driven Questions (KD-questions) and Case-Analysis Questions (CA-questions). There are 26,365 questions in JEC-QA, of which 5,289 of them comprising the test set. It's worth noting that the number of correct options for each question is uncertain.

MLEC-QA (Li, Zhong, and Chen 2021) is the multi-choice biomedical QA dataset. This dataset contains five subsets: Clinic (Cli), Stomatology (Sto), Public Health (PH), Traditional Chinese Medicine (TCM), and Traditional Chinese Medicine Combined with Western Medicine (CWM), all of them collected from the National Medical Licensing Examination in China. There are 136,236 questions in MLEC-QA, each presenting five options with one correct answer.

Accuracy serves as the primary evaluation metric. When the correct answer contains more than one option, the model prediction is considered correct only if it exactly matches the golden answer. Due to the limited availability of high-quality pre-training corpora, our experiments are primarily conducted in Chinese benchmarks. For more experiments on English datasets, please refer to Appendix (Li et al. 2024a).

Baselines

We adopt four groups of baselines for comparison: General LLMs, Legal-specific LLMs, Medical-specific LLMs, and Retrieval-augmented LLMs. General models include a range of popular LLM: ChatGLM-6B (Du et al. 2022),

ChatGLM2-6B (Du et al. 2022), Baichuan2-7B-Chat/13B-Chat (Baichuan 2023), Qwen-7B-Chat (Bai et al. 2023), ChatGPT (Floridi and Chiriatti 2020). Due to the high cost, we sampled a subset of the data to evaluate GPT-4 and report it in the Appendix. Legal-specific LLMs are further fine-tuned in the legal corpus to improve the understanding of the law, including LaywerLLaMA (Huang et al. 2023), LexiLaw, ChatLaw-13B/33B (Cui et al. 2023). For Medical-specific LLMs, Taiyi (Luo et al. 2023) and Zhongjing (Yang et al. 2023) are selected as baseline models. In the Retrieval-augmented framework, BM25, BGE (Xiao et al. 2023), M3E (Wang Yuxin 2023), GTE (Li et al. 2023b) are employed as the retriever. BGE (Xiao et al. 2023), M3E (Wang Yuxin 2023), GTE (Li et al. 2023b) are advanced text embedding systems within Chinese contexts. Due to space limitations, a detailed description of the baselines and the reasons for selecting them are provided in the appendix.

Implementation Details

We implement BLADE with BLOOMZ (Le Scao et al. 2023) as the small LM due to BLOOMZ's popularity and availability in various sizes. BLOOMZ with 1b7 parameters are employed to complete the main experiment. To construct the Chinese legal pre-training corpus, we collect legal articles, legal books, legal cases, and other resources from official websites [1]. In the medical domain, our corpus comprises medical Wikipedia entries and various medical texts. More details can be found in Appendix.

Experiment Result

Main result Table 1 presents the results from the baselines and BLADE on the JEC-QA dataset. We derive the following observations from the experiment results: (1) Legal-specific LLMs show relatively poor results. There is even some performance degradation after domain-specific fine-tuning. For example, LexiLaw underperforms compared to ChatGLM-6B. We hypothesize that although continuous tuning can enhance domain knowledge, it also significantly impacts the model's ability in prompt processing. This observation is also in line with Cheng et al (Cheng, Huang, and Wei 2023). Furthermore, it's worth noting that the legal-specific LLMs demonstrate substantial improvement when integrated with BLADE, implying that these models may not be fully leveraging the domain knowledge encoded in their parameters. (2) BLADE consistently enhances performance across various models. For example, Baichuan2-7B-Chat achieves 26.4% performance improvement, while ChatGPT achieves 31.3% performance improvement. This indicates that BLADE is applicable to diverse language models with different sizes. BLADE effectively utilizes domain knowledge without affecting the reasoning ability of the original model.

Table2 shows the performance of BLADE on the medical domain dataset MLEC-QA. Similar to the legal domain, the Medical-specific LLMs exhibit unsatisfactory performance. The challenge of integrating domain knowledge through continuous training without compromising the original capabilities of LLMs deserves further investigation. Another interesting finding is that, with the assistance of BLADE, the gap between general LLMs has been narrowed. We attribute this to the nature of the dataset itself, where the majority of the questions are relatively straightforward and do not require overly complex reasoning abilities. Once domain-specific knowledge is provided, the initial performance gap between these LLMs is further diminished. Overall, in the medical domain, BLADE demonstrates consistent performance improvements across all five subsets, with Bacihuan2-13B-Chat achieving the best performance.

Comparison with Retrieval Augmented LLMs To further explore the effectiveness of BLADE, we compare it to the retrieval augmentation paradigm. Due to the more challenging nature of the Case-Analysis Questions in JEC-QA, we primarily conducted our experiments on JEC-QA in this section. Experiments on other datasets can be found in Appendix. More specifically, We employ three different legal corpora, including legal_article, legal_book, and legal_all. The legal_article corpus contains all the Chinese legal provisions. Legal_book denotes the National Unified Legal Professional Qualification Examination Counseling Book, which consists of 15 topics and 215 chapters organized in a hierarchical manner. Legal_all corpus is consistent with the corpus in the DP phase, which contains all documents from legal_article and legal_book.

We select ChatGLM2-6B and ChatGPT, which have shown the best results in open-source and closed-source models respectively on the JEC-QA dataset, to conduct the experiment. To ensure a fair comparison, we exclusively use the top-1 document from the retrieval results and employ an identical prompt to BLADE, which also generates only a single document for analysis. Table 3 demonstrates the comparison results. We have the following observations: (1) Retrieval augmentation is proven to be effective in enhancing the performance of general LLMs in specific domains. However, its effectiveness is significantly influenced by the retrieval model and the corpus. Consequently, not all retrieved knowledge contributes positively to the task at hand. (2) Knowledge-Driven questions, focusing on the definition and explanation of legal concepts, tend to benefit more from the retrieved knowledge. However, Case-Analysis Questions, involving the analysis of real-life scenarios, may not see significant improvement from retrieved knowledge. This reflects the limitations of the retrieval augmentation paradigm, which lacks causal inference ability to identify question-specific knowledge. (3) Regardless of Knowledge-Driven or Case Analysis questions, BLADE consistently provides stable enhancements and achieves optimal performance. We attribute this to small LMs generating knowledge through deep token-level cross-attention, unlike the shallow interactions seen in modern retrieval models. This makes the generated knowledge more in-depth and specific to the question. We provide a detailed analysis of the strengths and weaknesses of BLADE and RAG in the appendix and thoroughly explain why BLADE outperforms RAG.

We further explore the impact of the number of retrieved documents on the performance of ChagtGLM2-6B. Specifically, we use M3E as the retrieval model and legal_all as the

[1] https://wenshu.court.gov.cn/

Model	# Parameters	KD-questions		CA-questions		All	
		Original	+BLADE	Original	+BLADE	Original	+BLADE
Legal Specific LLMs							
LaywerLLaMA	13B	9.76	12.94**(32.6%)	6.05	8.66**(43.1%)	7.45	10.26**(37.7%)
LexiLaw	6B	15.50	19.63**(26.6%)	14.35	18.07**(25.9%)	14.78	18.66**(26.5%)
ChatLaw-13B	13B	10.32	17.32**(67.8%)	5.03	8.08**(60.6%)	7.01	11.55**(64.8%)
ChatLaw-33B	33B	15.66	21.80**(39.2%)	17.01	20.46**(20.3%)	16.50	20.96**(27.0%)
General LLMs							
ChatGLM-6B	6B	17.08	21.19**(24.1%)	16.64	18.62**(11.9%)	16.81	19.58**(16.5%)
ChatGLM2-6B	6B	**27.39**	30.81**(12.5%)	24.09	**26.34**(9.3%)**	**25.32**	**28.01**(10.6%)**
Qwen-7B-Chat	7B	25.78	**31.26**(21.2%)**	**24.52**	25.07*(2.2%)	24.99	27.39**(9.6%)
Baichuan2-7B-Chat	7B	19.23	24.27**(26.2%)	19.53	21.73**(11.3%)	19.41	22.68**(16.8%)
Baichuan2-13B-Chat	13B	25.78	28.29**(9.73%)	21.80	24.22**(11.1%)	23.29	25.75**(10.5%)
ChatGPT	-	20.53	28.45**(38.6%)	18.70	23.67**(26.6%)	19.38	25.46**(31.3%)

Table 1: Overall accuracy on JEC-QA dataset. The gain % shows the relative improvement of methods compared to the original language model. */** denotes that BLADE performs significantly better than the original language model at $p < 0.05/0.01$ level using the fisher randomization test (Rubin 1980). Best results are marked bold.

Model	Cli		CWM		PH		Sto		TCM	
	Ori.	+BLADE	Ori.	+BLADE	Ori.	+BLADE	Ori.	+BLADE	Ori.	+BLADE
Medical Specific LLMs										
Zhongjing_base	15.58	35.74**	19.03	37.52**	16.55	36.98**	14.48	34.86**	17.41	36.65**
Zhongjing_sft	16.00	47.92**	18.50	49.64**	15.85	50.24**	15.76	46.12**	18.88	47.82**
Taiyi	43.42	49.72**	32.71	42.99**	35.11	45.63**	31.53	41.77**	32.83	43.65**
General LLMs										
ChatGLM-6B	30.04	53.42**	30.84	55.06**	30.47	55.66**	27.56	52.24**	32.96	53.64**
ChatGLM2-6B	48.86	60.20**	44.82	57.23**	44.39	59.75**	41.77	57.61**	46.12	55.72**
Qwen-7B-Chat	56.57	59.78*	52.59	58.20**	52.64	62.26**	49.33	57.39**	51.53	56.62**
Baichuan2-7B-Chat	51.10	59.99**	51.14	58.69**	50.00	62.45**	45.29	57.61**	51.79	56.82**
Baichuan2-13B-Chat	**58.98**	**61.62***	**54.39**	**58.79****	**57.92**	**63.80****	**50.39**	**57.84****	**54.87**	**57.34***
ChatGPT	47.56	58.92**	38.69	57.91**	47.73	63.37**	43.32	57.58**	36.49	56.40**

Table 2: Overall performance on the medical dataset MLEC-QA. */** denotes that BLADE performs significantly better than baselines at $p < 0.05/0.01$ level using the fisher randomization test (Rubin 1980). Best results are marked bold.

corpus because they achieve the best results in the retrieval augmentation paradigm. As shown in Table 4, when an appropriate number of documents are retrieved, there is a slight performance improvement due to more relevant documents being recalled. However, the performance of ChatGLM2-6B degrades when too many documents are retrieved, probably due to excessive noise introduced by the additional documents. In contrast, BLADE achieves the best results by generating only one piece of knowledge, suggesting its proficiency in producing more targeted and refined knowledge.

Ablation Studies

To better illustrate the effectiveness of our approach, we further conduct ablation studies on JEC-QA in zero-shot setting. Table5 shows the impact of different strategies. It's noticeable that while Domain-specific Pretraining successfully imparts domain knowledge to the small LM, it falls short in enabling instruction-following capabilities and in generating suitable knowledge, leading to a decrease in performance. With the integration of Knowledge Instruction Tuning, the small LM begins to offer beneficial knowledge. Bayesian Prompted Optimization further enhances the performance. The above experiments verify the effectiveness of each process within our approach.

Impact of Sizes

In this section, we aim to investigate the impact of the small LM's size. We conducted experiments on the JEC-QA dataset, utilizing ChatGLM2-6B as the general model. Three versions of the small LM, namely BLOOMZ_560m (Le Scao et al. 2023), BLOOMZ_1b1 (Le Scao et al. 2023), and BLOOMZ_1b7 (Le Scao et al. 2023), were tested, each trained with the same training parameters and datasets. The results are shown in Table 6. We can observe that the small model with 560m parameters can also lead to performance gains. As the parameters of the small LM increase, the performance improvement brought by BLADE also increases. This phenomenon could be attributed to larger models' enhanced capability to generate more accurate and reliable knowledge.

Case Study

In this section, we conduct a case study to facilitate a clear understanding of the effectiveness of BLADE. Figure 4 in Appnendix illustrates the comparison of retrieved knowledge retrieved by M3E-base from the legal_all corpus with the knowledge generated by BLADE. This question involves

Retrieval_model	Corpus	ChatGLM2-6B			ChatGPT		
		KD-questions(%)	CA-questions(%)	All(%)	KD-questions(%)	CA-questions(%)	All(%)
-	-	27.39	24.09	25.33	20.53	18.70	19.38
BM25	legal_article	28.19*(2.9%)	24.03(-0.2%)	25.59(1.0%)	21.70**(5.7%)	19.01**(1.6%)	20.02**(3.3%)
BM25	legal_book	29.60**(8.1%)	24.25(0.6%)	26.25(3.6%)	22.66**(10.3%)	19.26(2.9%)	20.53**(5.9%)
BM25	legal_all	28.14**(2.7%)	23.16(-3.8%)	25.03*(-1.1%)	20.19**(-1.6%)	18.74**(0.2%)	19.20**(-0.5%)
BGE	legal_article	28.51*(5.3%)	24.95(3.6%)	26.41*(4.3%)	26.73**(30.2%)	19.73*(5.5%)	22.36(15.4%)
BGE	legal_book	27.54(0.5%)	23.49(-2.4%)	25.01(-1.2%)	27.19**(32.4%)	19.55(4.5%)	22.42**(15.7%)
BGE	legal_all	30.11**(9.9%)	24.13(0.2%)	26.38*(4.1%)	27.75**(35.2%)	20.54**(9.8%)	23.25**(19.9%)
GTE	legal_article	27.09(-1.1%)	23.55(-2.2%)	24.88(-1.8%)	22.15**(7.8%)	19.04(1.8%)	20.21(4.3%)
GTE	legal_book	25.58(-6.6%)	22.84(-5.2%)	23.86(-5.7%)	21.90**(6.6%)	19.55(4.5%)	20.43(5.4%)
GTE	legal_all	25.43(-7.1%)	23.28(-3.4%)	24.09(-4.9%)	22.25**(8.3%)	19.10(2.1%)	20.28(4.6%)
M3E	legal_article	28.55*(4.2%)	24.77(2.8%)	26.19(3.4%)	26.03**(26.7%)	20.58**(10.1%)	22.63**(16.7%)
M3E	legal_book	27.74(1.3%))	24.77(2.8%)	25.88(2.2%)	26.28**(28.0%)	20.98**(12.2%)	22.97**(18.5%)
M3E	legal_all	30.56**(11.6%)	24.88(3.3%)	27.02*(6.6%)	28.20**(37.3%)	21.19**(13.3%)	23.82**(22.9%)
BLADE		**30.81**(12.5%)**	**26.34**(9.3%)**	**28.01**(10.6%)**	**28.45**(38.6%)**	**23.67**(26.6%)**	**25.46**(31.3%)**

Table 3: The performance comparison of BLADE and Retrieval-augmented LLMs on JEC-QA. The gain % shows the relative improvement of methods compared to the original language model. */** denotes that BLADE performs significantly better than the original language model at $p < 0.05/0.01$ level using the fisher randomization test (Rubin 1980). The best method in each column is marked in bold.

Model	doc_num	KD-questions	CA-questions	All
-	0	27.39	24.09	25.33
M3E	1	30.56	24.88	27.02
M3E	3	30.71	24.67	26.93
M3E	5	30.36	25.29	27.19
M3E	7	29.75	24.28	26.33
M3E	9	29.63	24.40	26.36
BLADE	-	**30.81**	**26.34**	**28.01**

Table 4: Impact of the number of retrieved documents on JEC-QA. Best results are marked bold.

Small Model	KD-questions(%)	CA-questions(%)	All(%)
-	27.39	24.09	25.33
BLOOMZ_1b7	26.38	22.40	23.89
+ DP	26.87	23.63	24.85
+ DP & KIT	28.45	24.89	26.23
+ DP & KIT & BPO	**30.81**	**26.34**	**28.01**

Table 5: Ablation study on JEC-QA under zero-shot setting. The general LLM is ChatGLM2-6B. Best results are marked bold.

the assessment of civil conduct capacity in the context of a contract dispute. The appropriate legal procedure involves suspending the ongoing proceedings and initiating a specialized process by Li's parents to affirm Li's status as a person with limited civil capacity. The retrieval model returns the article about proceedings for people with mental illnesses, which fails to directly address the civil litigation process and the implications of limited civil capacity in contract disputes. BLADE's response is more accurate and directly relevant to the question. It correctly identifies the key issue – the civil litigation process concerning the assessment of civil conduct capacity in the context of a contract dispute. This case shows BLADE's strength in providing domain-specific, contextually appropriate responses. The domain-specific LM, trained on nuanced legal knowledge, is adept at interpreting the underlying legal implications of the described events. Therefore, BLADE can effectively bridge the gap between the specific details of an event and the relevant legal principles or precedents. More examples can be found in Appendix.

Small Model	KD-questions(%)	CA-questions(%)	All(%)
-	27.39	24.09	25.33
BLOOMZ_560m	29.05	24.92	26.47
BLOOMZ_1b1	29.80	25.52	27.13
BLOOMZ_1b7	**30.81**	**26.34**	**28.01**

Table 6: Impact of sizes on JEC-QA. The general LLM is ChatGLM2-6B. Best results are marked bold.

Conclusion

This paper proposes BLADE, a new framework for applying general large language models to new domains. At its core, BLADE employs small language models to assimilate and continually update domain-specific knowledge. The framework solves problems by realizing collaboration between general large language models and a small domain-specific model. Through extensive experiments on legal datasets, we find BLADE consistently demonstrates performance improvement across various language models with different sizes. In the future, we will investigate approaches to minimize hallucinations in small models and explore additional methods for joint optimization. A limitation is that our experiments are conducted only in multiple-choice datasets, the feasibility of our approach in generative tasks still deserves further investigation.

References

Aharoni, R.; and Goldberg, Y. 2020. Unsupervised domain clusters in pretrained language models. *arXiv preprint arXiv:2004.02105*.

Arefeen, M. A.; Debnath, B.; and Chakradhar, S. 2023. LeanContext: Cost-Efficient Domain-Specific Question Answering Using LLMs. *arXiv preprint arXiv:2309.00841*.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Baichuan. 2023. Baichuan 2: Open Large-scale Language Models. *arXiv preprint arXiv:2309.10305*.

Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; Van Den Driessche, G. B.; Lespiau, J.-B.; Damoc, B.; Clark, A.; et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, 2206–2240. PMLR.

Chalkidis, I. 2023. ChatGPT may Pass the Bar Exam soon, but has a Long Way to Go for the LexGLUE benchmark. arXiv:2304.12202.

Chen, L.; Chen, J.; Goldstein, T.; Huang, H.; and Zhou, T. 2023. InstructZero: Efficient Instruction Optimization for Black-Box Large Language Models. *arXiv preprint arXiv:2306.03082*.

Cheng, D.; Huang, S.; and Wei, F. 2023. Adapting Large Language Models via Reading Comprehension. arXiv:2309.09530.

Cui, J.; Li, Z.; Yan, Y.; Chen, B.; and Yuan, L. 2023. ChatLaw: Open-Source Legal Large Language Model with Integrated External Knowledge Bases. arXiv:2306.16092.

Dai, Z.; Zhao, V. Y.; Ma, J.; Luan, Y.; Ni, J.; Lu, J.; Bakalov, A.; Guu, K.; Hall, K. B.; and Chang, M.-W. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; and Tang, J. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 320–335.

Floridi, L.; and Chiriatti, M. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30: 681–694.

Frazier, P. I. 2018. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Hansen, N. 2016. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, Q.; Tao, M.; An, Z.; Zhang, C.; Jiang, C.; Chen, Z.; Wu, Z.; and Feng, Y. 2023. Lawyer LLaMA Technical Report. *ArXiv*, abs/2305.15062.

Joshi, M.; Lee, K.; Luan, Y.; and Toutanova, K. 2020. Contextualized representations using textual encyclopedic knowledge. *arXiv preprint arXiv:2004.12006*.

Kleinberg, J. M. 1997. Two algorithms for nearest-neighbor search in high dimensions. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, 599–608.

Le Scao, T.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; Gallé, M.; et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474.

Lewis, P.; Stenetorp, P.; and Riedel, S. 2020. Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets. arXiv:2008.02637.

Lewis, P.; Wu, Y.; Liu, L.; Minervini, P.; Küttler, H.; Piktus, A.; Stenetorp, P.; and Riedel, S. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9: 1098–1115.

Li, H.; Ai, Q.; Chen, J.; Dong, Q.; Wu, Z.; Liu, Y.; Chen, C.; and Tian, Q. 2024a. BLADE: Enhancing Black-box Large Language Models with Small Domain-Specific Models. arXiv:2403.18365.

Li, H.; Chen, Y.; Ai, Q.; Wu, Y.; Zhang, R.; and Liu, Y. 2024b. LexEval: A Comprehensive Chinese Legal Benchmark for Evaluating Large Language Models. arXiv:2409.20288.

Li, H.; Dong, Q.; Chen, J.; Su, H.; Zhou, Y.; Ai, Q.; Ye, Z.; and Liu, Y. 2024c. LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods. arXiv:2412.05579.

Li, J.; Zhong, S.; and Chen, K. 2021. MLEC-QA: A Chinese multi-choice biomedical question answering dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 8862–8874.

Li, Y.; Wu, Y.; Li, J.; and Liu, S. 2023a. Prompting large language models for zero-shot domain adaptation in speech recognition. *arXiv preprint arXiv:2306.16007*.

Li, Z.; Zhang, X.; Zhang, Y.; Long, D.; Xie, P.; and Zhang, M. 2023b. Towards General Text Embeddings with Multi-stage Contrastive Learning. arXiv:2308.03281.

Liu, J.; Liu, A.; Lu, X.; Welleck, S.; West, P.; Bras, R. L.; Choi, Y.; and Hajishirzi, H. 2021. Generated knowledge prompting for commonsense reasoning. *arXiv preprint arXiv:2110.08387*.

Luo, L.; Ning, J.; Zhao, Y.; Wang, Z.; Ding, Z.; Chen, P.; Fu, W.; Han, Q.; Xu, G.; Qiu, Y.; Pan, D.; Li, J.; Li, H.; Feng, W.; Tu, S.; Liu, Y.; Yang, Z.; Wang, J.; Sun, Y.; and Lin, H. 2023. Taiyi: A Bilingual Fine-Tuned Large Language Model for Diverse Biomedical Tasks. *arXiv preprint arXiv:2311.11608*.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Rubin, D. B. 1980. Randomization analysis of experimental data: The Fisher randomization test comment. *Journal of the American statistical association*, 75(371): 591–593.

Sachidananda, V.; Kessler, J. S.; and Lai, Y.-A. 2021. Efficient domain adaptation of language models via adaptive tokenization. *arXiv preprint arXiv:2109.07460*.

Shi, W.; Min, S.; Yasunaga, M.; Seo, M.; James, R.; Lewis, M.; Zettlemoyer, L.; and Yih, W.-t. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Sun, T.; Shao, Y.; Qian, H.; Huang, X.; and Qiu, X. 2022a. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, 20841–20855. PMLR.

Sun, Z.; Wang, X.; Tay, Y.; Yang, Y.; and Zhou, D. 2022b. Recitation-augmented language models. *arXiv preprint arXiv:2210.01296*.

Wang, W.; Srikumar, V.; Hajishirzi, H.; and Smith, N. A. 2022. Elaboration-generating commonsense question answering at scale. *arXiv preprint arXiv:2209.01232*.

Wang Yuxin, H. s., Sun Qingxuan. 2023. M3E: Moka Massive Mixed Embedding Model.

Xiao, S.; Liu, Z.; Zhang, P.; and Muennighof, N. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. *arXiv preprint arXiv:2309.07597*.

Yang, S.; Zhao, H.; Zhu, S.; Zhou, G.; Xu, H.; Jia, Y.; and Zan, H. 2023. Zhongjing: Enhancing the Chinese Medical Capabilities of Large Language Model through Expert Feedback and Real-world Multi-turn Dialogue. arXiv:2308.03549.

Yu, W.; Iter, D.; Wang, S.; Xu, Y.; Ju, M.; Sanyal, S.; Zhu, C.; Zeng, M.; and Jiang, M. 2022. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.

Zeng, A.; Liu, X.; Du, Z.; Wang, Z.; Lai, H.; Ding, M.; Yang, Z.; Xu, Y.; Zheng, W.; Xia, X.; et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Zhong, H.; Xiao, C.; Tu, C.; Zhang, T.; Liu, Z.; and Sun, M. 2020. JEC-QA: a legal-domain question answering dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 9701–9708.