CoEvo: Coevolution of LLM and Retrieval Model for Domain-Specific Information Retrieval

Ang Li[1], **Yiquan Wu**[2†], **Yinghao Hu**[1], **Lizhi Qing**[3], **Shihang Wang**[3], **Chengyuan Liu**[1]
Tao Wu[1], **Adam Jatowt**[4], **Ming Cai**[1†], **Fei Wu**[1,5], **Kun Kuang**[1†]

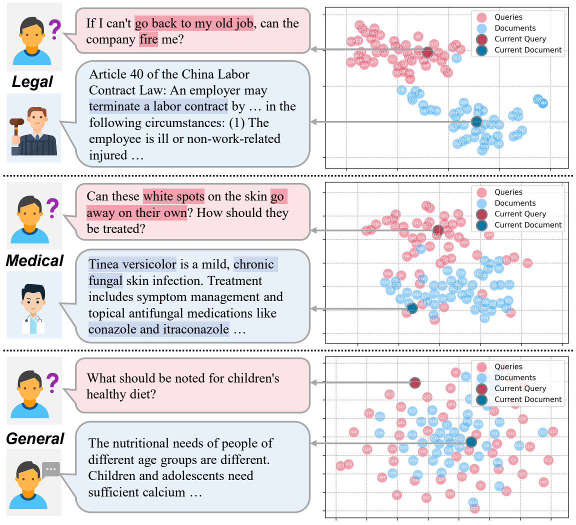
[1]College of Computer Science and Technology, Zhejiang Unirersity, [2]Guanghua Law School, Zhejiang University,
[3]Alibaba Group, Hangzhou, China, [4]University of Innsbruck, Innsbruck, Austria, [5]Shanghai AI Laboratory
{leeyon, wuyiquan, huyinghao, liucy1, cm, wufei, kunkuang}@zju.edu.cn
{yekai.qlz, wangshihang.wsh}@alibaba-inc.com, adam.jatowt@uibk.ac.at

Abstract

Information retrieval in specialized domains (e.g., legal or medical) faces challenges in aligning user queries, often expressed in colloquial language, with highly structured, terminology-rich documents. This discrepancy creates a distribution gap in the text representation. Recent methods aim to enhance queries by generating intermediary elements (e.g., keywords, pseudo-documents) before performing retrieval with large language models (LLMs). However, by treating LLMs and retrievers separately, these approaches risk producing unreliable or irrelevant intermediaries, which can significantly degrade retrieval performance. To address this issue, we propose CoEvo, an alternating optimization framework that facilitates the coevolution of LLMs and retrieval models. CoEvo operates through two key steps: L-step directs the LLM in generating intermediaries by leveraging an archive of historical examples known to enhance retrieval. R-step trains the retriever using contrastive learning on the intermediaries produced by the LLM. Finally, we evaluate and flexibly leverage content generated by the LLM to amplify the effectiveness of coevolution. Experimental results demonstrate significant improvements in retrieval performance across both legal and medical domains.

1 Introduction

Information Retrieval (IR) technologies are a cornerstone among data processing techniques when it comes to acquiring information (Manning et al., 2008). Given a typically short input query, retrieval aims to obtain relevant documents from external data collections (Kobayashi and Takeda, 2000; Singhal, 2001). IR has found multiple applications in various fields (Buettcher et al., 2010; Lee et al., 2019; Yin et al., 2016), such as online search, question answering, and recommender systems. In the early days, researchers mainly focused on lexical-based sparse retrieval utilizing statistical features like BM25 (Robertson and Zaragoza, 2009). With the advent of extensive labeled datasets and advanced model training, embedding-based dense retrieval (Xiong et al., 2020a; Qu et al., 2021) has emerged as a superior paradigm that involves vectorizing both queries and documents, computing their similarities and finally selecting the most relevant documents to the query.



If I can't go back to my old job, can the company fire me?

Article 40 of the China Labor Contract Law: An employer may terminate a labor contract by … in the following circumstances: (1) The employee is ill or non-work-related injured …

Can these white spots on the skin go away on their own? How should they be treated?

Tinea versicolor is a mild, chronic fungal skin infection. Treatment includes symptom management and topical antifungal medications like conazole and itraconazole …

What should be noted for children's healthy diet?

The nutritional needs of people of different age groups are different. Children and adolescents need sufficient calcium …

Figure 1: Examples of vector distribution for queries and documents in professional (legal and medical) and general domains, where a mismatch between queries and documents in professional domains can be seen.

Retrieval is critical in professional domains, including medicine, journalism, legal, and finance areas (Kolomiyets and Moens, 2011). Unlike general retrieval tasks, queries in domain-specific retrieval are often articulated by laypersons, who tend to use common or colloquial terms, whereas domain-specific documents usually contain specific terminology that may not be known to users, as illustrated in Fig. 1. This gap can result in a significant mismatch in vector distributions between queries

† Corresponding Author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 14980–14999
November 4-9, 2025 ©2025 Association for Computational Linguistics

and documents when using embedding-based dense retrieval, which can degrade retrieval performance. While large language models (LLMs) excel in text understanding and bring significant improvement for various tasks (Touvron et al., 2023; Achiam et al., 2023), they operate in a generation paradigm, making them less suited for direct application in retrieval tasks. Recently, researchers have explored utilizing LLMs to optimize queries before retrieval (Ma et al., 2023; Wang et al., 2023a; Peng et al., 2024), typically by generating query-related information (e.g., keywords, pseudo documents) to enhance retrieval. For simplicity, in this paper, we refer to query-related information as intermediary. However, in that paradigm, LLMs and retrievers function independently, raising a significant concern: **LLMs may generate unreliable or irrelevant information, which can mislead subsequent retrieval process**. This begs the question: *Can we provide more effective collaboration between LLMs and retrievers to improve retrieval?*

Building on this vision, we propose the **CoEvo**, an alternating optimization framework to facilitate the **coevo**lution of an LLM and a retriever. Initially, we employ the LLM to generate the intermediary for each query. Then, the CoEvo framework alternates between the LLM optimization (L-step) and the retriever optimization (R-step) in an iterative scheme: In the L-step, based on the retriever's feedback, we construct an example archive, which stores exemplary query-intermediary pairs, where the intermediary led to finding relevant documents. These examples aim to guide the LLM in generating more accurate intermediaries through in-context learning (ICL). In the R-step, sourcing from intermediaries newly generated by the LLM, we construct training data consisting of positive and negative samples, which are employed in contrastive learning to train the retriever for document retrieval using intermediaries. Throughout this iterative process, both the LLM's example archive and the retriever's training data can be continuously updated, fostering their coevolution. Finally, we introduce an intermediary evaluator that assigns quality-based weights to each intermediary, enhancing the impact of high-quality intermediaries on final retrieval. It is trained solely on the feedback from L-step and R-step iterations, requiring no additional data, and further amplifying the effectiveness of the coevolution.

We evaluate our method on both legal and medical retrieval datasets. Extensive experiments demonstrate that our CoEvo framework yields superior results compared to baseline models.

To summarize, our contributions are:

- We investigate the domain-specific retrieval problem of the collaboration of LLM and retriever, optimizing the retrieval process from query to intermediary and then to document.
- We propose CoEvo, a framework that alternates between an L-step, where the retriever guides the LLM to generate intermediaries, and an R-step, where the LLM generates training data for the retriever using intermediaries. We further introduce an evaluator to assess and flexibly leverage LLM-generated content.
- We conduct extensive experiments on two professional domains (i.e., legal and medical), demonstrating that our approach achieves superior retrieval performance.
- We make the code and data publicly available to encourage other scholars to investigate this problem[1].

2 Related Work

2.1 Domain-specific Retrieval

Retrieval is widely applied across various professional fields, including law (Cui et al., 2024; Li et al., 2025b), medicine, and finance (Chen et al., 2023c). Unlike general retrieval tasks, queries in domain-specific retrieval are often written in a non-professional or even colloquial manner while the professional documents are highly-structured in terms of domain-specific terminology (Li et al., 2024; Zhou et al., 2024), which makes the task more challenging.

Traditional retrieval models consist of two types: sparse models and dense models. Sparse models, such as TF-IDF (Sparck Jones, 1972) and BM25 (Robertson et al., 2009), typically rely on inverted index matching and raw data input. While training-free and easy to use, their performance is highly sensitive to query and database quality, struggling with mismatched, colloquial, or complex queries in professional fields (Drozdov et al., 2022). Dense models like DPR (Karpukhin et al., 2020), contriever (Izacard et al., 2021), E5 (Wang et al., 2022), and BGE (Chen et al., 2023b), typically embed queries and documents into a con-

[1] https://github.com/LIANG-star177/CoEvo/tree/master

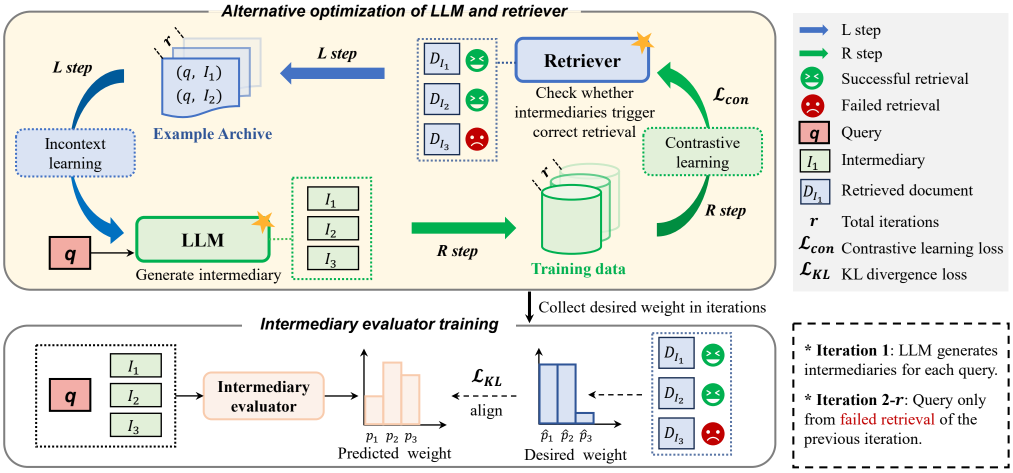

Alternative optimization of LLM and retriever
L step
(q, I_1)
(q, I_2)
r
Example Archive
L step
D_{I_1}
D_{I_2}
D_{I_3}
Retriever
Check whether intermediaries trigger correct retrieval
$\mathcal{L}_{con}$
Incontext learning
Contrastive learning
r
R step
q
LLM
Generate intermediary
I_1
I_2
I_3
R step
Training data
L step
R step
Successful retrieval
Failed retrieval
q Query
I_1 Intermediary
D_{I_1} Retrieved document
r Total iterations
$\mathcal{L}_{con}$ Contrastive learning loss
$\mathcal{L}_{KL}$ KL divergence loss
Collect desired weight in iterations
Intermediary evaluator training
q
I_1
I_2
I_3
Intermediary evaluator
p_1 p_2 p_3
Predicted weight
$\mathcal{L}_{KL}$
align
$\hat{p}_1$ $\hat{p}_2$ $\hat{p}_3$
Desired weight
D_{I_1}
D_{I_2}
D_{I_3}
* Iteration 1: LLM generates intermediaries for each query.
* Iteration 2-r: Query only from failed retrieval of the previous iteration.

Figure 2: The training framework of CoEvo.

tinuous vector space that adheres to certain standards, such as semantic similarity (Karpukhin et al., 2020). Although dense retrieval methods are trainable and flexible, they often struggle with significant vector distribution mismatches between queries and documents in domain-specific retrieval, reducing retrieval performance. While some studies propose query expansion methods (Lavrenko and Croft, 2001; Lv and Zhai, 2009) to bridge the gap between them, these methods are designed for sparse retrieval and do not adapt well to the higher-performing dense retrieval models. In this paper, we try to leverage the advantages of LLMs to bridge the gap in domain-specific retrieval.

2.2 LLM-Assisted Retrieval

Large Language Models (LLMs) excel in text understanding and embedding, and are frequently utilized for generative tasks. Some researchers have attempted to optimize queries using LLMs for enhanced retrieval (Wang et al., 2023a; Peng et al., 2024; Liu et al., 2025). Typically, these approaches involve generating additional information with LLMs and referencing this to retrieve documents. Recently, some researchers have used LLMs to enhance traditional retrievers. For example, Wang et al. (2023b) starts by generating pseudo-documents through the few-shot prompting of LLMs. These pseudo-documents are then used to retrieve documents. Anand et al. (2023a) proposed context-aware query rewriting. First, ambiguous training queries are rewritten by employing context-aware prompting of LLMs, where relevant documents are used as context. Subsequently, a ranker is fine-tuned on these rewritten queries rather than the original ones during training. However, both of these approaches merely leverage the LLM's generative capabilities and the retriever's retrieval functions, without fostering heuristic collaboration between them. In this paper, we introduce the CoEvo framework, which uses a coevolutionary synergy paradigm between LLMs and retrievers to achieve better performance in domain-specific retrieval.

3 Preliminaries

In this section, we first follow the previous LLM-assisted retrieval work to do domain-specific retrieval. Researchers use LLMs to enrich the user's original query's information (Ma et al., 2023; Wang et al., 2023a; Anand et al., 2023a). Following this, LLMs can generate intermediaries for the queries and use them to retrieve relevant documents.

Given a query q, relevant documents $\hat{D} = \{d_i\}_{i=1}^{\hat{k}}$, and $\hat{k}$ denoting the number of the relevant documents, LLM is asked to construct the intermediary I based on q. Then their concatenation $q^+ = \text{concat}(q, [\text{SEP}], I)$ is input into the retriever (using bi-encoder structure, details in Appendix I) to encode it into the vector h_{q^+}. Meanwhile, the document base is also vectorized as $\{h_{d_j}\}_{j=1}^{m}$ by the model, where m is the total number of documents. Then their cosine similarity is calculated:

$$s_j = \frac{h_{q^+} \cdot h_{d_j}}{\|h_{q^+}\| \|h_{d_j}\|}. \tag{1}$$

Thus, the documents with the highest score are selected as the retrieval result. For better performance, dense retrieval can be employed with labeled data training. The embedding model transforms the concatenation and document into vectors h_{q^+} and h_d. Then, documents in $\hat{D}$ are treated as positive samples, while hard negatives $\mathbb{N}$ are found using BM25 as negative samples (details in Appendix H). The contrastive learning loss is used:

$$\mathcal{L}_{con} = -\sum_{d\in\hat{D}} \log \frac{e^{h_{q^+}\cdot h_d}}{e^{h_{q^+}\cdot h_d} + \sum_{d_i\in\mathbb{N}} e^{h_{q^+}\cdot h_{d_i}}}. \tag{2}$$

However, in domain-specific retrieval, LLMs need to generate professional and precise intermediaries. Without mutual feedback between LLMs and retrievers, LLMs may introduce low-quality intermediaries and mislead the retrieval.

4 Methodology

4.1 An Overview of CoEvo

As illustrated in Fig. 2, CoEvo alternates between LLM optimization (L-step) and retriever optimization (R-step) in an iterative scheme. We first allow the LLM to generate intermediaries across all training data as the initialization, followed by alternating steps: In the L-step, the retriever selects intermediaries that trigger relevant documents and stores them in an example archive. Then we use examples from the archive to guide the LLM in generating better intermediaries through in-context learning (ICL). In the R-step, we create training data to train the retriever for document retrieval using intermediaries through contrastive learning. Finally, data from L-step and R-step iterations are used to train the intermediary evaluator to weight intermediates by quality, enhancing the influence of high-quality ones on retrieval.

To reduce cost, CoEvo iterates only on the previous iteration's failed retrievals, gradually refining the example archive for the LLM and the training data for the retriever. We condense the training process into pseudocode as portrayed by Alg. 1.

4.2 L-step: LLM Optimization with ICL

L-step is designed to help the LLM generate better intermediaries for the retriever. We first store intermediaries that trigger successful retrieval in the example archive with the help of the retriever's feedback. In t-th iteration, for each query q_i, we concatenate it with the intermediary I_i generated by the LLM and input it into the retriever to retrieve documents, denoted as d_{I_i}. We stipulate that if the retrieved document is relevant (i.e., $d_{I_i} \in \hat{D}_i$), this is a successful retrieval, and the query-intermediary pair (q_i, I_i) should be put into the example archive EA as a new good example:

$$\Delta EA = \bigcup_{i=1}^{n_t} \{(q_i, I_i)\}, \text{if } d_{I_i} \in \hat{D}_i, \tag{3}$$

here, ΔEA denotes a newly added example, n_t is the data size of the t-th iteration. In this way, in the subsequent iterations, we can select relevant query-intermediary pairs from the example archive to guide LLMs' generation by in-context learning. The selection of examples can also be achieved using retrieval methods.

Algorithm 1 The pseudocode of CoEvo.

Input: Query q, Relevant document $\hat{D}$, Total iteration r, LLM parameters θ_{LLM}.
Output: Retriever parameters θ_{R_r}, Intermediary evaluator parameters θ_E

```
iteration $t = 0$
while $t < r$ do
    Select example $e$ from $EA_t$
    $I = \theta_{LLM}(q, e)$
    $D_I = \theta_{R_t}(q, I)$                          ▷ Eq. 2
    $\Delta(EA, T, P) = eval(D_I, \hat{D})$             ▷ Eq. 3,4,5
    $EA_{t+1} = EA_t + \Delta EA$
    $T_{t+1} = T_t + \Delta T$
    $P_{t+1} = P_t + \Delta P$
    $\theta_{R_{t+1}} = \mathcal{L}_{con}(T_{t+1})$     ▷ Equ. 1
    $t = t + 1$
end while
$\theta_E = \mathcal{L}_D(q, I, P_r)$                   ▷ Equ. 8
```

4.3 R-step: Retriever Optimization with Contrastive Learning

R-step helps the retriever better utilize the query and intermediary to find the corresponding document. We differentiate between intermediaries that trigger successful retrieval and those that fail to construct the training data for the retriever.

Specially, for i-th query: (1) If the intermediary I_i triggers the successful retrieval, we use the query-intermediary concatenation $q_i^+ = \text{concat}(q_i, \text{[SEP]}, I_i)$ as anchor, relevant documents $\hat{D}_i$ as positive samples, and BM25 hard negatives $\mathbb{N}_i$ as negative samples. (2) If the intermediary triggers the failed retrieval, we treat $\hat{D}_i$ as both the anchor and positive samples at the same time, while the other intermediaries in the batch are treated as batch negatives $\mathbb{B}_i$ (details in Appendix H),

$$\Delta T = \bigcup_{i=1}^{n_t} \begin{cases} \{(I_i, \hat{D}_i, \mathbb{N}_i)\}, & \text{if } d_{I_i} \in \hat{D}_i \\ \{(\hat{D}_i, \hat{D}_i, \mathbb{B}_i)\}, & \text{if } d_{I_i} \notin \hat{D}_i \end{cases}. \quad (4)$$

By forming triplets through permutation and constructing the training data, we train the retriever using contrastive learning with the loss function in Eq. 2. This approach not only avoids introducing low-quality intermediaries into the training data but also ensures the retriever is trained across all data samples, preserving training effectiveness.

4.4 Amplification of Coevolution Effectiveness

To amplify the effectiveness of coevolution, we introduce an intermediary evaluator trained to weight generated intermediates by quality, enhancing their influence on final retrieval. It is trained solely on data from L-step and R-step iterations, requiring no additional data. Specifically, the desired weight distribution of intermediaries is defined and recorded based on the feedback of the retriever as follows:

$$\Delta P = \bigcup_{i=1}^{n_t} \begin{cases} 1, & \text{if } d_{I_i} \in \hat{D}_i \\ 0, & \text{if } d_{I_i} \notin \hat{D}_i \end{cases}. \quad (5)$$

If a query has k^I intermediaries, the desired weight distribution is $\hat{\mathbf{p}} = \{\hat{p}_i\}_i^{k^I}$. After completing r iterations, we derive P_r to train the intermediary evaluator. For deeper interaction, we concatenate the query and its i-th intermediary and input them into a cross-encoder model (details in Appendix I) to obtain the output of the final hidden layer. We extract the hidden state corresponding to the [CLS] token:

$$h_{i,\text{[CLS]}} = \text{Encoder}(x_i)[0], \quad (6)$$

where $x_i = \text{concat}(\text{[CLS]}, q, \text{[SEP]}, I_i)$. Then we use a linear layer to map $h_{i,[CLS]}$ to a score:

$$p_i = W \cdot h_{i,\text{[CLS]}} + b, \quad (7)$$

where W is the weight matrix and b is the bias vector. For all k^I intermediaries, we calculate the corresponding scores and do softmax operation to obtain predicted weight distribution $\mathbf{p} = \text{softmax}\{p_i\}_{i=1}^{k^I}$. To align the predicted distribution as closely as possible with the desired distribution, we utilize KL Divergence as the loss function:

$$\mathcal{L}_D = D_{\text{KL}}(\hat{\mathbf{p}} \parallel \mathbf{p}) = \sum_{i=1}^{k^I} \hat{p}_i \log\left(\frac{\hat{p}_i}{p_i}\right). \quad (8)$$

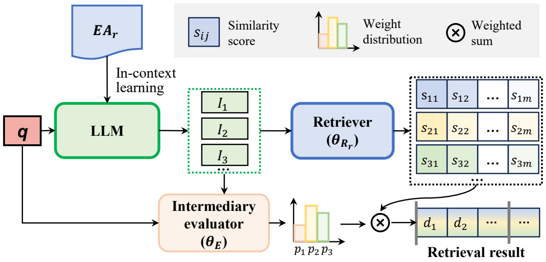


Figure 3: The inference of CoEvo.

4.5 Inference

In the inference stage shown in Fig. 3, we use the example archive EA_r, the retriever with parameter θ_{R_r}, and the intermediary evaluator with parameter θ_E. Given a query, we let the LLM generate k^I intermediaries in parallel referring to the examples from EA_r. For i-th intermediary, the retriever calculates the similarity score s_{ij} with each document following Eq. 1, and the intermediary evaluator assigns its weight p_i. The final similarity score is computed as follows:

$$\tilde{s} = \{\sum_{i=1}^{k^I} p_i s_{ij}, j = 1, \ldots, m\}. \quad (9)$$

We then select the documents with the highest scores as the final retrieval result.

5 Experiments

5.1 Datasets

We conduct our experiments on two Chinese datasets, including legal and medical domains. QLAR is a dataset that consists of user queries and the corresponding applicable law articles, supplemented by data from the EQUALS dataset (Chen et al., 2023a). This dataset features flexible queries and numerous law articles, posing a more challenging retrieval task compared to previous work that focused solely on criminal law (Yue et al., 2021; Xu et al., 2020; Zhong et al., 2018). KUAKE-IR is a dataset derived from the medical paragraph retrieval task in the CBLUE benchmark (Zhang et al., 2022). The input is a user query, and the output is a relevant medical paragraph. More details can be found in Appendix G. The statistics of these two datasets are shown in Tab. 1. To ensure the fairness of the verification, we randomly divided each dataset into training set, validation set, and test set, maintaining the ratio of 80%: 10%: 10%.

Type	QLAR	KUAKE-IR
# of Samples	17794	4630
# of Document base	17043	8520
Avg. # of Tokens in Query	18.59	10.34
Avg. # of Tokens in Document	70.18	98.19
Avg. # of Document in Sample	1.70	1.00

Table 1: Statistics of the dataset.

5.2 Metrics

Following the evaluation approach employed in prior work (Long et al., 2022), we evaluate the retrieval performance by Recall precision at top 1, 5, 10, 100 (R@1, R@5, R@10, R@100) and Mean Reciprocal Rank at 10 documents (MRR@10). We use BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) to evaluate the quality of the intermediary.

5.3 Baselines

In the experiments, we employ both the traditional sparse and dense retrieval. **BM25** (Robertson and Zaragoza, 2009) is a sparse model based on term frequency and document length. **Contriever** (Izacard et al., 2022) is a fine-tuning dense model through contrastive learning with a shared bi-encoder architecture. **DRAGON** (Lin et al., 2023) has separate models for encoding queries and documents. **ColBERTv2** (Santhanam et al., 2022) uses a token-level late-interaction approach, keeping all token-level embeddings for computing similarities. We also explore domain-specific retrieval methods. **PEG** (Wu et al., 2023) is trained across several specific domains, including legal and medicine. **SAILER** (Li et al., 2023) is a SOTA legal case retrieval model that leverages structural information and legal rules. **Chatlaw** (Cui et al., 2024) is a text similarity model trained on court case data. **MedBert**[2] is a BERT-based text embedding model (Devlin et al., 2018) trained on clinical data.

For the LLM-assisted retrieval methods, we evaluate two general LLMs, **GPT-4o mini** (Achiam et al., 2023) and **Qwen2.5-turbo** (Yang et al., 2024), as well as two Domain LLMs, **Farui-plus**[3] in legal domain and **HuatuoGPT2-7B-4bits** (Zhang et al., 2023) in medical domain. For the collaboration of the LLM and the retriever, we implement three methods to compare with our method: (1) **LLM w/o training retriever**, in which LLM directly generates intermediaries and then retrieves relevant documents. (2) **query2doc** (Wang et al., 2023a) uses few shot prompting to guide LLMs generating intermediaries but exclude intermediaries in training retriever. (3) **CAR** (Anand et al., 2023b) finetunes the retriever with the rewritten queries by LLM through context-aware prompting.

5.4 Implementation Details

For all methods, we set the training to 5 epochs with a learning rate of 5e-5, and a batch size of 4. This also means that our CoEvo undergoes 5 iterations. For retrieval, referring to Langchain Chatchat (Liu et al., 2024), we use FAISS to prevectorize the knowledge base, allowing for quick computation of similarity scores. In all LLM-assisted retrieval methods, the retriever is initialized using PEG retriever. The prompt for the legal domain is *"Please answer the legal articles applicable to the following questions in the Chinese legal system. (Format like Article 273 of the Criminal Law of the People's Republic of China:...)"*. The prompt for the medical domain is *"Please answer the patient's question with professional medical knowledge."* In CoEvo, we select $k^e = 3$ ICL examples with the best-performing retriever PEG in dense retrieval. In the inference stage, we let LLM generate $k^I = 3$ intermediaries in parallel. We have chosen those as the above settings achieve the best performance. The impact of ICL examples and prompt design is explored in Appendix A.

To evaluate the robustness and generalizability of CoEvo, we conduct experiments in zero-shot and few-shot settings, as well as on the English biomedical dataset TREC-COVID. We further analyze resource consumption and present case studies. Due to space limitation, we include these experiments in Appendices B, C, D, E, F.

5.5 Main Results

From Tab. 2, we conclude that: (1) Compared to sparse or dense retrieval, LLM-assisted retrieval shows notable performance gains. This suggests that LLMs, memorizing an enormous amount of knowledge, can enrich queries and guide retrieval systems. (2) CoEvo surpasses all baselines on two datasets, particularly in Legal. This shows the effectiveness of our collaborative evolution framework of LLM and retriever. Compared to query2doc and CAR, we guide the generation of the intermediary through feedback design and use the intermediary more flexibly. (3) When training retriever, LLM-assisted retrieval may perform worse than tradi-

[2] https://github.com/trueto/medbert

[3] https://tongyi.aliyun.com/farui

Model	Training retriever	QLAR					KUAKE-IR				
		R@1	R@5	R@10	R@100	MRR@10	R@1	R@5	R@10	R@100	MRR@10
Sparse or Dense Retrieval											
BM25	×	8.82	21.97	28.60	52.53	30.22	23.91	34.41	40.04	60.86	28.40
Contriever	✓	10.62	27.45	38.88	73.93	37.15	26.96	42.27	49.36	77.36	33.62
Dragon	✓	15.73	40.53	52.84	85.04	48.85	29.02	47.24	57.04	86.28	37.01
Colbertv2	✓	13.20	35.79	48.37	79.69	43.10	26.36	41.27	48.50	76.95	32.70
PEG	✓	24.33	53.76	65.39	92.92	62.02	34.56	59.18	69.55	95.68	45.41
SAILER	✓	9.32	24.83	34.33	71.91	33.94	-	-	-	-	-
Chatlaw	✓	3.48	10.62	16.40	53.88	15.08	-	-	-	-	-
MedBert	✓	-	-	-	-	-	31.53	53.78	61.77	92.22	40.73
LLM-assisted Retrieval (w/ PEG)											
GPT-4o mini	×	11.29	35.48	43.55	84.68	41.07	28.49	37.02	46.74	72.39	33.25
Qwen2.5-turbo	×	14.52	33.06	47.58	81.45	43.72	28.05	39.02	48.78	73.17	32.85
+ query2doc	✓	24.13	52.45	64.23	89.37	60.72	36.28	60.48	71.49	95.89	47.39
+ CAR	✓	26.10	54.18	66.83	91.72	61.58	37.05	60.46	70.93	95.57	47.06
+ CoEvo	✓	**29.55**	58.86	67.68	92.93	63.29	38.44	61.12	72.35	**96.33**	**47.87**
DomainLLM	×	20.62	40.21	50.52	82.24	45.82	34.85	48.48	51.52	74.24	40.11
+ query2doc	✓	26.38	55.15	69.66	93.18	63.61	37.58	60.83	71.49	**96.33**	47.41
+ CAR	✓	27.44	58.95	71.03	93.29	64.50	37.40	60.47	72.05	95.87	47.48
+ CoEvo	✓	28.17	**59.15**	**73.24**	**94.82**	**65.17**	**38.61**	**61.42**	**73.26**	95.71	47.75

Table 2: Performance Comparison of Different Models on Article Retrieval and Medical Retrieval Tasks. DomainLLM refers to Farui-plus on the QLAR dataset and HuatuoGPT2-7B-4bits on the KUAKE-IR dataset.

Method	Example selection	Intermediary Quality				Retrieval Performance			
		ROUGE-1	ROUGE-2	ROUGE-L	BLEU-N	R@1	R@5	R@10	MRR@10
	PEG	**72.93**	**45.96**	**57.34**	34.53	**29.55**	**58.86**	**67.68**	**63.29**
CoEvo	KNN	65.25	41.61	52.78	29.51	26.25	52.16	62.63	58.75
	Random	65.01	33.46	45.93	23.37	21.18	37.75	43.53	43.35
w/o EA	PEG	69.11	39.64	52.59	29.05	25.61	52.89	60.02	57.17
w/o ICL	-	64.33	34.36	47.37	23.81	23.03	45.88	53.37	52.64
w/ GPT2	-	59.95	41.58	51.56	**41.93**	22.53	41.07	50.73	48.36
w/ T5	-	63.55	40.73	49.05	39.52	21.46	45.96	53.11	50.51

Table 3: Performance comparison of different intermediary generation methods.

tional dense retrieval (e.g., on MRR@10, Qwen2.5-turbo + query2doc gets 60.72%, which is less than trained PEG's 62.02%). This proves the irrelevant information of LLM-generated intermediaries can mislead retriever training. (4) DomainLLMs deliver better performance than general LLMs for our method because their specialized knowledge can produce the intermediary with more relevant information to enrich the query. (5) The bi-encoder model outperforms the cross-encoder model (e.g., Dragon outperforms Contriever and ColBERTv2). This indicates that separately encoding queries and documents can mitigate the problem caused by the gap between them. (6) Domain embedding does not result in better performance. Although domain embeddings are better at understanding domain documents, they may lose the ability to handle colloquial queries. (7) The overall performance of the KUIKE-IR dataset outperforms the QLAR dataset, likely due to the greater gap between queries and documents in the QLAR dataset.

5.6 Analysis Study

We further analyze each design in CoEvo through ablation and replacement experiments.

Example selection and intermediary generation. We explore different methods for generating the quality of intermediaries and the impact on retrieval performance. From the results on the QLAR dataset with Qwen2.5-turbo presented in Tab. 3, we can draw several conclusions: (1) Three methods are tested for selecting examples from the example archive: **Random**, **KNN**, and **PEG**. The examples are used to guide the LLM in generating the intermediary through in-context learning (ICL). **PEG** selects the intermediary with better quality

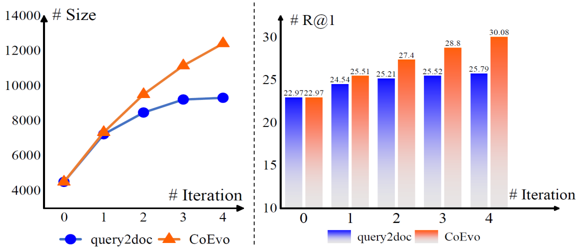


Figure 4: Example archive size and retrieval performance in different iterations.

Method	R@1	R@5	R@10	MRR@10
CoEvo	**29.55**	**58.86**	**67.68**	**63.29**
w/o multi	27.93	56.08	64.38	60.11
w/ concat	27.42	54.33	59.94	56.36
w/ sum	27.81	54.78	60.20	58.88
w/ cos-sim	26.79	56.17	62.99	61.58

Table 4: Intermediary fusion method's performance on QLAR dataset.

Method	R@1	R@5	R@10	MRR@10
PEG	24.33	53.76	65.39	62.02
+ CoEvo	29.55$^{+5.22}$	58.86$^{+5.10}$	67.68$^{+2.29}$	63.29$^{+1.27}$
Dragon	15.73	40.53	52.84	48.85
+ CoEvo	20.79$^{+5.06}$	45.31$^{+4.78}$	58.93$^{+6.09}$	53.77$^{+4.92}$
Contriever	10.62	27.45	38.88	37.15
+ CoEvo	19.10$^{+8.48}$	36.52$^{+9.07}$	47.23$^{+8.35}$	45.61$^{+8.46}$

Table 5: Performance variation with different retrievers on QLAR dataset.

Method	R@1	R@5	R@10	MRR@10
GPT-4o mini	27.19	57.96	64.10	60.82
Qwen2.5-turbo	**29.55**	**58.86**	**67.68**	**63.29**
Farui-plus	28.17	59.15	73.24	65.17

Table 6: Performance comparison of different LLMs.

and gets better retrieval performance. (2) **w/o EA** retrieves examples from the training data instead of the example archive for ICL. Both with PEG, it decreases the performance, which indicates that examples from the example archive are more useful to guide intermediary generation. (3) **w/o ICL** lets LLM generate the intermediary directly without examples. The performance decline indicates the effectiveness of in-context learning. (4) Moreover, we replace LLMs with small language models, denoted as **w/ GPT2** (Radford et al., 2019) and **w/ T5** (Raffel et al., 2020). Small language models underperform compared to LLMs in generating intermediaries because it is hard for them to generate knowledge-dense intermediaries from queries with sparse information.

Performance sensitivity to iterations. In the training iterations, CoEvo evaluates the intermediary to expand the example archive and the training data, enabling the interaction between the LLM and the retriever. However query2doc excludes the intermediary in the retriever training, cutting off this interaction. We explore changes of the example base size and the retrieval performance in iterations as shown in Fig. 4. As the iteration increases, we draw the following insights: (1) CoEvo expands bigger example archive size and achieves better retrieval performance. (2) The performance of query2doc diverges further from CoEvo, due to the retriever being limited to handling the original query. In contrast, CoEvo solves the problem by incorporating the intermediary into training with the corresponding feedback design.

Intermediary fusion. In the inference stage, CoEvo uses the intermediary evaluator to fuse retrieval results from multiple intermediaries. We compare several other fusion methods: **w/o multi** only retrieves based on one intermediary. **w/ concat** concatenates all intermediaries and retrieves one document list directly. **w/ sum** aggregates similarity scores of the same documents for all intermediaries and then reranks. **w/ cos-sim** weights with cosine similarity between query and intermediary, and then sum. From Tab. 4, we observe that: (1) The intermediary evaluator trained based on retriever feedback outperforms other methods, indicating that it can well predict the weight of the intermediary, thereby ranking the more likely retrieval results further ahead. (2) Using cosine similarity as a weight provides minimal additional gain, suggesting that semantic similarity does not fully capture intermediary importance.

Performance using different retrievers and LLMs. We assess the generalizability of our CoEvo framework using PEG as the baseline retriever and compare its performance with other retrievers, such as Dragon and Contriever, on the QLAR dataset. As shown in Tab. 5, CoEvo demonstrates significant improvements despite different retrievers (e.g., with Dragon, CoEvo increases R@1 by 5.06, R@5 by 4.78, R@10 by 6.09, and MRR@10 by 4.92). Tab. 6 evaluates the performance of using different LLMs in CoEvo, with Qwen2.5-turbo achieving the best results.

6 Conclusion

In this paper, we investigate domain-specific retrieval tasks in vertical domains (e.g. law and medical). The queries are often colloquial, while the documents contain abundant domain-specific terminology. We identify that the current LLM-assisted retrieval paradigm, where LLM and retriever perform their own respective works, may introduce irrelevant information. To effectively harness the capabilities of LLMs, we propose an alternating optimization framework, namely CoEvo, for achieving the coevolution between LLM and retriever. Extensive experiments demonstrate that CoEvo improves the quality of LLM generation and further enhances retrieval performance.

7 Ethical Statement

With the development of AI, more and more DomainAI technologies (e.g. LegalAI and MedAI) are proposed to assist people, especially those who suffer from an intense workload (Chalkidis et al., 2019; Topol, 2019; Li et al., 2025a). Professional domains tend to be critical and sensitive areas, hence any subtle miscalculation may trigger serious consequences, so it is imperative to discuss the related ethical issues. Our model aims to provide people with reference by retrieving domain documents, which is an algorithmic investigation where still many potential risks remain (e.g., LLM generation security, document database quality). The algorithm can be beneficial to address the consultation problem. Nevertheless, the algorithm only intends to assist experts and should not “replace” human experts. The retrieval operation should be conducted or verified based on manual verification.

8 Limitations

Despite its effectiveness, the CoEvo framework has several limitations, which are as follows:

- The quality of improvements relies on the diversity and accuracy of collected examples and training data, which may be biased or insufficient in practice.
- Due to the involvement of LLMs, query rewriting or query transformation methods incur notable resource overhead, as analyzed in Appendix E. While dynamically selecting whether to involve LLMs based on task difficulty is a promising approach (Jeong et al., 2024).

9 Acknowledgments

This work was supported in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (2025C02037), National Natural Science Foundation of China (62376243), and the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study (SN-ZJU-SIAS-0010). All opinions in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

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

Abhijit Anand, Vinay Setty, Avishek Anand, et al. 2023a. Context aware query rewriting for text rankers using llm. *arXiv preprint arXiv:2308.16753*.

Abhijit Anand, Venktesh V, Vinay Setty, and Avishek Anand. 2023b. Context aware query rewriting for text rankers using llm. *Preprint*, arXiv:2308.16753.

Stefan Buettcher, Charles L. A. Clarke, and Gordon V. Cormack. 2010. *Information Retrieval: Implementing and Evaluating Search Engines*. Cambridge, MA.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in english. In *ACL (1)*, pages 4317–4323. Association for Computational Linguistics.

Andong Chen, Feng Yao, Xinyan Zhao, Yating Zhang, Changlong Sun, Yun Liu, and Weixing Shen. 2023a. Equals: A real-world dataset for legal question answering via reading chinese laws. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL ’23, page 71–80, New York, NY, USA. Association for Computing Machinery.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2023b. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2309.07597.

Wei Chen, Qiushi Wang, Zefei Long, Xianyin Zhang, Zhongtian Lu, Bingxuan Li, Siyuan Wang, Jiarong Xu, Xiang Bai, Xuanjing Huang, and Zhongyu Wei. 2023c. Disc-finllm: A chinese financial large language model based on multiple experts fine-tuning. *arXiv preprint arXiv:2310.15205*.

Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2024. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model. *Preprint*, arXiv:2306.16092.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. 2022. Compositional semantic parsing with large language models. In *The Eleventh International Conference on Learning Representations*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Preprint*, arXiv:2112.09118.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong-Cheol Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 7036–7050. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Mei Kobayashi and Koichi Takeda. 2000. Information retrieval on the web. *ACM Comput. Surv.*, 32(2):144–173.

Oleksandr Kolomiyets and Marie-Francine Moens. 2011. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 120–127, New York, NY, USA. Association for Computing Machinery.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

Ang Li, Yiquan Wu, Ming Cai, Adam Jatowt, Xiang Zhou, Weiming Lu, Changlong Sun, Fei Wu, and Kun Kuang. 2025a. Legal judgment prediction based on knowledge-enhanced multi-task and multi-label text classification. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6957–6970, Albuquerque, New Mexico. Association for Computational Linguistics.

Ang Li, Yiquan Wu, Yifei Liu, Ming Cai, Lizhi Qing, Shihang Wang, Yangyang Kang, Chengyuan Liu, Fei Wu, and Kun Kuang. 2025b. UniLR: Unleashing the power of LLMs on multiple legal tasks with a unified legal retriever. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11953–11967, Vienna, Austria. Association for Computational Linguistics.

Ang Li, Yiquan Wu, Yifei Liu, Fei Wu, Ming Cai, and Kun Kuang. 2024. Enhancing court view generation with knowledge injection and guidance. *Preprint*, arXiv:2403.04366.

Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. Sailer: Structure-aware pre-trained language model for legal case retrieval. *Preprint*, arXiv:2304.11370.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *Preprint*, arXiv:2302.07452.

Qian Liu, Jinke Song, Zhiguo Huang, Yuxuan Zhang, glide the, and liunux4odoo. 2024. Langchain-chatchat. `https://github.com/chatchat-space/Langchain-Chatchat`. GitHub repository.

Yibin Liu, Ang Li, and Shijian Li. 2025. Hierarchical interaction summarization and contrastive prompting for explainable recommendations. *Preprint*, arXiv:2507.06044.

Dingkun Long, Qiong Gao, Kuan Zou, Guangwei Xu, Pengjun Xie, Ruijie Guo, Jian Xu, Guanjun Jiang, Luxi Xing, and Ping Yang. 2022. Multi-cpr: A multi domain chinese dataset for passage retrieval. In *SIGIR*, pages 3046–3056. ACM.

Yuanhua Lv and ChengXiang Zhai. 2009. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 1895–1898, New York, NY, USA. Association for Computing Machinery.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *Preprint*, arXiv:2305.14283.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Wenjun Peng, Guiyang Li, Yue Jiang, Zilong Wang, Dan Ou, Xiaoyi Zeng, Derong Xu, Tong Xu, and Enhong Chen. 2024. Large language model based long-tail query rewriting in taobao search. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 20–28.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *Preprint*, arXiv:2112.01488.

Amit Singhal. 2001. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24:35–43.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Eric J. Topol. 2019. High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine*, 25:44 – 56.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Simlm: Pre-training with representation bottleneck for dense passage retrieval. *arXiv preprint arXiv:2207.02578*.

Liang Wang, Nan Yang, and Furu Wei. 2023a. Query2doc: Query expansion with large language models. *Preprint*, arXiv:2303.07678.

Liang Wang, Nan Yang, and Furu Wei. 2023b. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*.

Tong Wu, Yulei Qin, Enwei Zhang, Zihan Xu, Yuting Gao, Ke Li, and Xing Sun. 2023. Towards robust text retrieval with progressive learning. *arXiv preprint arXiv:2311.11691*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020a. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *Preprint*, arXiv:2007.00808.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020b. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish confusing law articles for legal judgment prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3086–3095, Online. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng

Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Dawei Yin, Yuening Hu, Jiliang Tang, Tim Daly, Mianwei Zhou, Hua Ouyang, Jianhui Chen, Changsung Kang, Hongbo Deng, Chikashi Nobata, Jean-Marc Langlois, and Yi Chang. 2016. Ranking relevance in yahoo search. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 323–332, New York, NY, USA. Association for Computing Machinery.

Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021. Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 973–982.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 1503–1512.

Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, Xiang Wan, Benyou Wang, and Haizhou Li. 2023. Huatuogpt, towards taming language models to be a doctor. *arXiv preprint arXiv:2305.15075*.

Ningyu Zhang, Mosha Chen, Zhen Bi, Xiaozhuan Liang, Lei Li, Xin Shang, Kangping Yin, Chuanqi Tan, Jian Xu, Fei Huang, Luo Si, Yuan Ni, Guotong Xie, Zhifang Sui, Baobao Chang, Hui Zong, Zheng Yuan, Linfeng Li, Jun Yan, Hongying Zan, Kunli Zhang, Buzhou Tang, and Qingcai Chen. 2022. CBLUE: A Chinese biomedical language understanding evaluation benchmark. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7888–7915, Dublin, Ireland. Association for Computational Linguistics.

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3540–3549, Brussels, Belgium. Association for Computational Linguistics.

Xiang Zhou, Yudong Wu, Ang Li, Ming Cai, Yiquan Wu, and Kun Kuang. 2024. Unlocking authentic judicial reasoning: A template-based legal information generation framework for judicial views. *Knowledge-Based Systems*, 301:112232.

A Exploring Different Prompts and Varying Numbers of ICL Examples

We also evaluate the impact of different prompts and varying numbers of in-context learning (ICL) examples. We conducted these tests using the Farui LLM on the QLAR dataset. As shown in the Tab. 7, we observe that specifying the format in the prompt (Prompt 2) reduces ROUGE-1, ROUGE-2, and ROUGE-L scores, but improves BLEU-n scores and retrieval performance. This improvement can be attributed to the shorter response length and increased specificity, which resulted in more professional outputs.

Moreover, when using examples to guide ICL, we found that the quality of the generated intermediaries and their retrieval performance peaked when the number of examples was set to $k^e = 3$.

B Performance in Zero-shot and Few-shot Scenarios

In zero-shot scenario, CoEvo degrades to query2doc, where the LLM expands queries and then performs retrieval, without further iterative feedback between the LLM and retriever. In few-shot scenario, we use BM25 to retrieve the top-k most relevant query-document pairs from the training set for each test sample. This provides k-shot reference data for the LLM during inference. The LLM first generates k = {1, 2, 3} intermediaries based on each reference query and its corresponding ground truth document. Then, using in-context learning (ICL), the LLM mimics the intermediary for the current query, which is subsequently used for retrieval. The performance comparison on QLAR is shown in Tab. 8. We observe that: (1) The full-shot setting achieves the best performance because its examples can be iteratively optimized. (2) The 1-shot setting ranks second best since more examples may introduce irrelevant information, reducing the quality of intermediary generation.

C Performance on TREC-COVID

We evaluate CoEvo on the dataset TREC-COVID, which consists of large-scale biomedical documents. This allows us to assess the robustness of our method on large-scale data and in the English language. Additionally, TREC-COVID is

Type	Intermediary Quality				Retrieval Performance			
	ROUGE-1	ROUGE-2	ROUGE-L	BLEU-N	R@1	R@5	R@10	MRR@10
				w/o example				
prompt 1	76.39	53.77	58.93	31.08	13.03	38.88	43.37	46.31
prompt 2	63.65	45.31	45.61	36.96	17.42	39.33	44.94	48.36
				w example				
$k^e = 1$	63.60	46.00	47.23	37.11	19.10	36.52	45.51	49.20
$k^e = 2$	65.36	47.53	**49.05**	38.60	18.54	37.08	45.51	50.82
$k^e = 3$	**65.48**	**48.12**	48.05	**39.52**	20.22	**38.20**	**49.44**	**52.17**
$k^e = 4$	64.25	47.24	48.53	38.13	**21.91**	37.64	47.75	51.61

Table 7: Performance comparison of different prompts and example numbers. **Prompt 1**:*Please provide the law articles in the Chinese legal system applicable to this question.* **Prompt 2**:*Please provide the law articles in the Chinese legal system applicable to the question. Respond only in the specified format without explanations. (Format like Article 273 of the Criminal Law of the People's Republic of China:...)*

Method	R@1	R@5	R@10	R@100	MRR@10
zero-shot	24.33	53.76	65.39	92.92	62.02
1-shot	24.75	55.83	66.11	92.78	62.17
2-shot	24.36	53.49	65.98	92.37	61.54
3-shot	22.24	52.38	64.68	90.04	60.29
full-shot	28.17	59.15	73.24	94.82	65.17

Table 8: Performance in zero-shot and few-shot settings.

Method	P@5	P@10	P@100	Recip Rank
BM25	46.40	44.00	28.82	65.71
PEG	51.35	46.41	31.83	66.76
query2doc	54.73	52.24	36.97	70.67
CoEvo	55.92	53.83	36.80	71.26

Table 9: Performance on the dataset TREC-COVID.

from the biomedical domain, showcasing the extension of our study. We use the July 16, 2020 release of CORD-19 from the official dataset. Since each query in the TREC-COVID dataset has many matching documents (on average 1,565), which differs significantly from our QLAR and KUAKE-IR datasets, where each query has only one or two matching documents, we used P@5, P@10, P@100, and Reciprocal Rank metrics. We test BM25, PEG, query2doc, and CoEvo (ours), considering both comprehensiveness and time constraints. In both query2doc and CoEvo, the LLM used is Qwen2.5-turbo, and the retriever is PEG. The experimental results are shown in Fig. 9. We found that CoEvo achieved the best performance, demonstrating its generalizability across various applications.

Method	R@1	R@10	MRR@10
PEG w/o I in train&infer	24.33	65.39	62.02
PEG w/ I in infer	20.43	56.00	53.54
PEG w/ I in train&infer (CoEvo)	28.17	73.24	65.17

Table 10: Performance comparison of different intermediary settings. "I" refers to "intermediary".

D Intermediary Training Discussion

We explore two settings to validate the effectiveness of using intermediaries in fine-tuning. (1) "PEG w/o I in train&infer": This setting fine-tunes the model directly with the query and corresponding document, actually is the baseline PEG in Tab. 2. (2) "PEG w/ I in infer": Here, the intermediary is used only during inference, without participation in training. (3) "PEG w/ I in train&infer": Our CoEvo follows this setting. From Tab. 10, we find "PEG w/ I in infer" reduces performance significantly as the trained PEG is limited to handling the original query and can't adapt to query-intermediary concatenation. So CoEvo involves the intermediary in both training and inference, performing better.

E Consumption Analysis

We conduct a resource consumption analysis of our method using Qwen as the LLM. As shown in Tab. 11, in terms of time, the training phase consists of two steps. The L-step, where the LLM generates intermediaries and the retriever evaluates them, takes 1.43 hours per epoch. The R-step, where the retriever is trained, takes 0.45 hours per epoch. With 5 epochs in total, the entire training process takes 9.4 hours. During inference, the LLM generates an intermediary in 0.96 seconds, and retrieval takes

	Phase	Details	Consumption
Time	Training (per epoch)	L-step R-step Total	1.43 h 0.45 h **1.88 h**
	Inference (per sample)	LLM Retriever Total	0.96 s 0.046 s **1.01 s**
Space	Training	-	**33,928 MB**
	Inference	-	**4,356 MB**

Table 11: Resource Consumption Analysis of CoEvo.

0.046 seconds, resulting in a total of 1.01 seconds per sample.

In terms of space, Qwen operates as an API and does not consume local storage. The retriever requires 33,928 MB of memory during training and 4,356 MB during inference.

F Case Study

As shown in Fig. 6, 7, 8, 9, we provide real-world cases from the legal and medical domains, along with their English-translated version. The prompts for generating intermediaries using the in-context learning (ICL) method are also demonstrated within the cases. In the context of In-Context Learning (ICL), query-intermediary pairs are retrieved from our Example Archive. In the legal domain, LLM generates multiple potential law articles as intermediaries. These intermediaries overlap with *Articles 44 and 46 of the Contract Law of the People's Republic of China* in the ground truth, significantly reducing the difficulty of subsequent retrieval tasks. In the medical domain, the LLM tends to generate only a single intermediary. To address this, we configured different temperatures and executed multiple runs of the LLM to produce several intermediaries.

We also explore the changes in context and intermediary across iterations. We select a case from the QLAR dataset, showing the changes in the 0th, 2nd, and 4th iterations of CoEvo in Fig. 10. We observe that: (1) In the 0th iteration, with no prior examples available, the LLM generated irrelevant and poorly formatted intermediaries that did not follow Chinese legal conventions, resulting in retrieval errors (e.g., incorrect retrieval of Article 38). (2) In the 2nd iteration, some examples had been collected, but their quality and relevance were still limited. This led to interference from irrelevant content, such as an example from insurance law ranking first. Despite this, the intermediary began to include correct references (e.g., Article 46 of the Labor Law), indicating early improvement. (3) In the 4th iteration, sufficient high-quality examples enabled the LLM to generate a more accurate and informative intermediary, successfully retrieving the correct document. As iterations progressed, the proportion and ranking of relevant examples in the context improved, leading to higher-quality intermediaries and assisting the retriever in finding the correct result. This demonstrates the effectiveness of our CoEvo framework.

G Dataset Construction Details

QLAR is a legal dataset derived from the public EQUALS dataset, which contains 6,914 high-quality, annotated query-article-answer triplets. We extract query-article pairs directly from EQUALS without additional annotation or preprocessing.

KUAKE-IR is a medical dataset collected from Alibaba's Quark Search, reflecting real-world business scenarios. The dataset ensures diversity by randomly sampling across different medical specialties. Query-paragraph pairs are derived from click behavior logs and verified through model-based checks and manual review to ensure accuracy. This publicly available dataset already includes matched query-paragraph pairs, requiring no additional annotation. We only performed preprocessing to maintain data quality. A query-paragraph pair is removed if: (1) The query explicitly mentions a disease name, as this significantly reduces retrieval difficulty. (2) The paragraph contains fewer than 20 characters.

H The Implementation of Negatives

H.1 BM25 Negatives

BM25 is a widely used and classic method for selecting hard negatives, helping the retriever learn to distinguish relevant from non-relevant content by choosing documents with high similarity scores but incorrect relevance. It is both efficient and computationally inexpensive. Several prior works have used BM25 for hard negatives, including DPR (Karpukhin et al., 2020) and ColBERT (Khattab and Zaharia, 2020). Mathematically, the BM25 score for a document D given a query Q can be approximated as:

$$\sum_{i \in q \cap d} IDF_i \cdot \frac{f_i(d) \cdot (k_1 + 1)}{f_i(d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})}$$

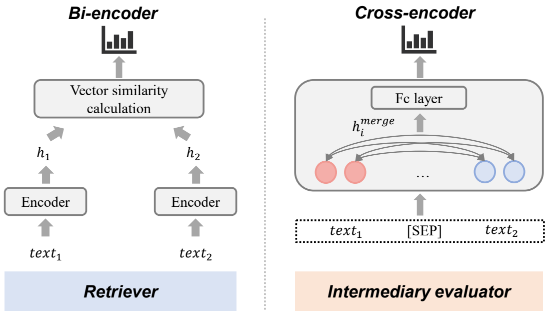


Figure 5: Model details.

where $f_i(d)$ is the term frequency of term i in document d, $|d|$ is the length of document d, $avgdl$ is the average document length in the corpus, k_1 and b are hyperparameters, IDF_i is the inverse document frequency of term i.

Another promising method is hybrid negative sampling. In this approach, a sparse method first selects hard negatives, and a dense retriever further refines them, forming a mixed negative sample pool. This strategy has been explored in works such as ANCE (Xiong et al., 2020b) and STAR (Zhan et al., 2021), and we plan to investigate it in future work.

H.2 Batch Negatives

Batch negatives are created within a batch of data during training. Let's assume we have a batch of intermediary embeddings $\{h_1^I, h_2^I, ..., h_N^I\}$, N is the batch size. Suppose the retrieval result of i-th of the batch is incorrect. Then, we replace the i-th intermediary embedding with the corresponding document embedding. The batch negatives are the other embeddings in the batch, i.e., $\mathbb{B} = \{h_j^I | j \neq i\}$.

In this way, the wrongly retrieved data is fully trained through a self-supervised manner in a reversed direction for the contrast between the intermediary and the document. At the same time, this approach avoids low-quality intermediaries participating in the training process of the retriever.

I Model Structure Details

In this work, we employed a bi-encoder architecture for the retriever, while a cross-encoder architecture was utilized for the intermediary evaluator, as shown in Fig. 5.

I.1 Bi-encoder

Two independent encoders are used to encode the text into vectors respectively, then calculate the similarity. The main advantage is that the similarity calculation is fast. For example, in the recall scenario of question-document, because what the two-tower model obtains is actually the representation of a single text, the calculation of similarity only makes a simple calculation at the end, and the most time-consuming document representation operation can be completed offline.

I.2 Cross-encoder

When the model is input, it is concatenated and input to the same encoder, so that the two texts have deeper interaction, and then through a fully connected layer. Therefore, the similarity calculation effect of the cross-encoder is also significantly better than that of the bi-encoder of the two-tower structure. However, the similarity calculation between a large number of texts in the cross-encoder requires real-time model inference calculation and consumes more time, which is suitable for the re-ranking stage after retrieval.

Query	我是退休人员，被一家公司招聘后，签了一年的协议，现在公司因投入不足提前解除劳动关系。有什么补偿吗？
Prompt	**Instruction:** 请回答中国法律体系中适用于下面问题的法律法条。只按照例如的格式回答，不用解释。（例如：《中华人民共和国刑法》第二百七十三条：……） **Example:** 下面是回答对应法律法条的例子： Question: 企业提前解除劳动合同，员工能否获得一年一个月工资的补偿？ Law Article: 《中华人民共和国劳动合同法》第四十七条：用人单位依照本法第四十条、第四十一条的规定解除劳动合同的，应当向劳动者支付经济补偿。经济补偿按照劳动者在本单位的工作年限计算，每满一年支付相当于一个月工资的经济补偿。工作满六个月不满一年的，按一年计算；不满六个月的，支付半个月工资的经济补偿。支付的经济补偿以劳动合同解除或终止前劳动者十二个月的平均工资为标准。 Question: 我与对方签订了一份合同，但是现在因为一些原因无法继续履行，我可以直接解除合同吗？是否需要通知对方？ Law Article: 《中华人民共和国合同法》第九十三条：当事人协商一致，可以解除合同。当事人可以在合同中约定一方解除合同的条件，解除合同的条件成就时，解除权人可以解除合同。当事人一方依照法律规定解除合同的，应当通知对方，合同自通知到达对方时解除。 Question: 我在矿山受伤，被认定为8级，解除劳动合同，工资约6000，我能赔偿多少？ Law Article: 《中华人民共和国工伤保险条例》第三十七条：职工因工致残被鉴定为七级至十级伤残的，劳动合同解除或终止时，用人单位应当按照下列标准支付一次性伤残就业补助金和一次性工伤医疗补助金：（一）七级伤残为本人工资的十三个月；（二）八级伤残为本人工资的十一个月；（三）九级伤残为本人工资的九个月；（四）十级伤残为本人工资的七个月。本人工资按照劳动合同解除或者终止前十二个月的平均月工资计算。 **Input:** Question: 我是退休人员，被一家公司招聘后，签了一年的协议，现在公司因投入不足提前解除劳动关系。有什么补偿吗？ Law Article:
Intermediary	《中华人民共和国劳动合同法》第四十四条：有下列情形之一的，劳动合同终止：（一）劳动合同期满的；······（五）法律、行政法规规定的其他情形。 《中华人民共和国劳动合同法》第四十六条：有下列情形之一的，用人单位应当向劳动者支付经济补偿：（一）用人单位依照本法第三十六条的规定向劳动者提出解除劳动合同并与劳动者协商一致解除劳动合同的；······（三）用人单位依照本法第四十条规定解除劳动合同的。 《中华人民共和国劳动合同法》第四十七条：经济补偿按劳动者在本单位工作的年限，每满一年支付一个月工资的标准向劳动者支付。六个月以上不满一年的，按一年计算；不满六个月的，向劳动者支付半个月工资的经济补偿。
Ground truth	《中华人民共和国劳动合同法》第四十六条：有下列情形之一的，用人单位应当向劳动者支付经济补偿：（一）劳动者依照本法第三十八条规定解除劳动合同的；（二）用人单位依照本法第三十六条规定向劳动者提出解除劳动合同并与劳动者协商一致解除劳动合同的；（三）用人单位依照本法第四十条规定解除劳动合同的；（四）用人单位依照本法第四十一条第一款规定解除劳动合同的；（五）除用人单位维持或者提高劳动合同约定条件续订劳动合同，劳动者不同意续订的情形外，依照本法第四十四条第一项规定终止固定期限劳动合同的；（六）依照本法第四十四条第四项、第五项规定终止劳动合同的；（七）法律、行政法规规定的其他情形。

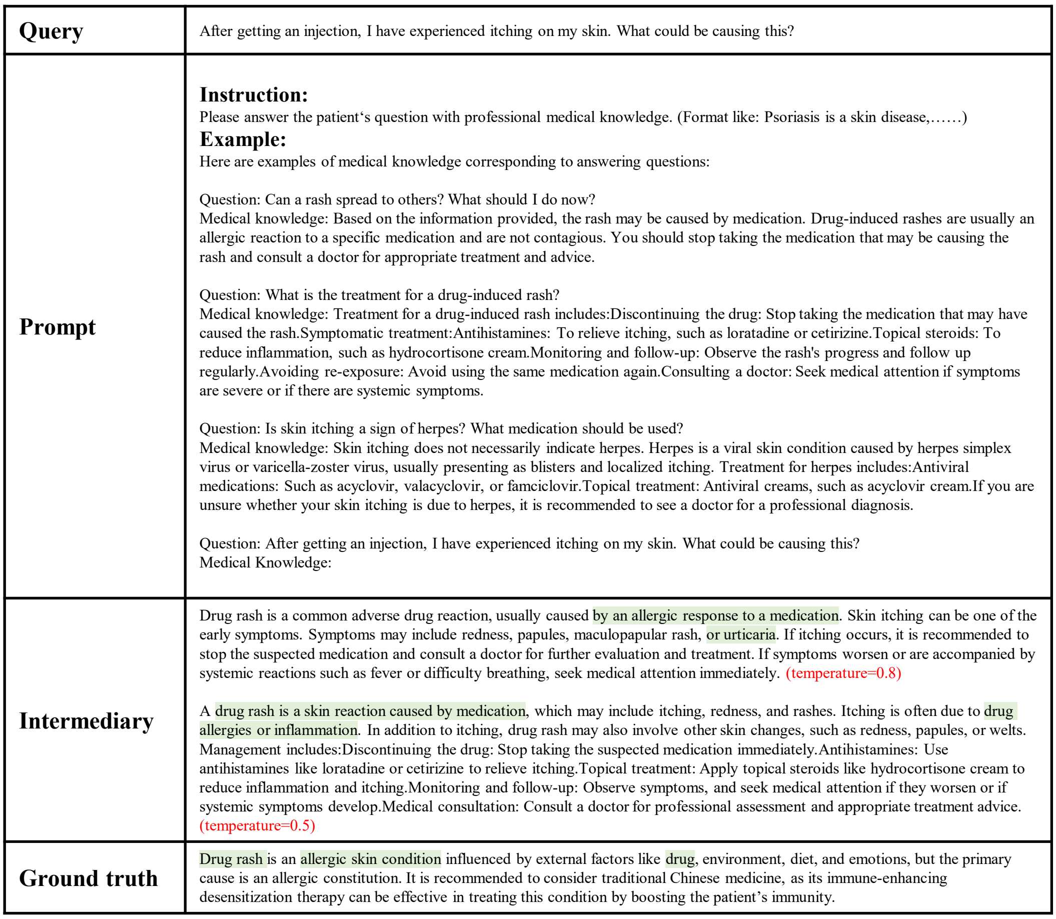
Figure 6: Chinese law case example.

Query	I am a retired employee who was recruited by a company and signed a one-year agreement. However, the company has terminated my employment relationship early due to insufficient investment. Is there any compensation available?
Prompt	**Instruction:** Please answer the legal articles applicable to the following questions in the Chinese legal system. (Format like Article 273 of the Criminal Law of the People's Republic of China:...) **Example:** Here are examples of law articles corresponding to answering questions: Question: If a company terminates an employment contract early, can the employee receive compensation equivalent to one month's salary for each year worked? Law Article: <Labor Contract Law of the People's Republic of China>, Article 47: If an employer terminates an employment contract according to Articles 40 and 41 of this law, the employer must pay economic compensation. The compensation is calculated based on the employee's years of service at the company, with one month's salary paid for each full year. For periods between six months and one year, compensation is calculated as one year; for periods less than six months, compensation is half a month's salary. The compensation is based on the employee's average salary for the twelve months before the termination or end of the contract. Question: I signed a contract with another party, but now I cannot fulfill it due to certain reasons. Can I terminate the contract directly? Do I need to notify the other party? Law Article: <Contract Law of the People's Republic of China>, Article 93: Parties may terminate the contract by mutual agreement. A contract may stipulate conditions under which one party can terminate it. When these conditions are met, the entitled party can terminate the contract. If one party terminates the contract according to legal provisions, they must notify the other party, and the contract is terminated upon the receipt of such notice by the other party. Question: I was injured in a mine, rated as a Level 8 injury, and my salary is about 6,000 RMB. How much compensation can I receive? Law Article: <Regulation on Work-Related Injury Insurance of the People's Republic of China>, Article 37: If an employee is disabled due to a work-related injury and is rated as Level 7 to 10 disability, when the employment contract is terminated or ended, the employer must pay a lump sum disability employment subsidy and a lump sum work injury medical subsidy as follows: (1) For Level 7 disability, 13 months of the employee's salary; (2) For Level 8 disability, 11 months of the employee's salary; (3) For Level 9 disability, 9 months of the employee's salary; (4) For Level 10 disability, 7 months of the employee's salary. The employee's salary is calculated based on their average monthly salary for the twelve months before the termination or end of the contract. Question: I am a retired employee who was recruited by a company and signed a one-year agreement. However, the company has terminated my employment relationship early due to insufficient investment. Is there any compensation available? Law Article:
Intermediary	<Article 44>: A labor contract shall be terminated under any of the following circumstances:The term of the labor contract expires;...;Other circumstances stipulated by laws or administrative regulations. <Article 46>: An employer shall pay economic compensation to the employee under any of the following circumstances:The employer proposes to terminate the labor contract in accordance with Article 36 of this law, and the employee agrees through negotiation;...;The employer terminates the labor contract in accordance with Article 40 of this law. <Article 47>: Economic compensation shall be paid to the employee based on the number of years they have worked for the employer, at the rate of one month's wage for each full year of service. For periods of six months or more but less than one year, it shall be calculated as one year; for periods of less than six months, half a month's wage shall be paid as economic compensation.
Ground truth	<Article 46 of the Labor Contract Law of the People's Republic of China>: The employer is required to pay economic compensation to the employee under any of the following circumstances:1. The employee terminates the labor contract according to Article 38 of this law.2. The employer proposes to terminate the labor contract according to Article 36 of this law, and both parties agree to the termination.3. The employer terminates the labor contract according to Article 40 of this law.4. The employer terminates the labor contract according to the first paragraph of Article 41 of this law.5. The employer terminates a fixed-term labor contract according to the first item of Article 44 of this law, except when the employer maintains or improves the conditions agreed upon in the contract and the employee does not agree to renew the contract.6. The labor contract is terminated according to items 4 or 5 of Article 44 of this law. 7. Other circumstances stipulated by laws and administrative regulations.

Figure 7: English version of law case example.

Query	我打完针就有皮肤瘙痒的现象，这是怎么回事呢？
Prompt	**Instruction:** 请用专业的医学知识回答病人的问题。只按照例如的格式回答，不用解释。（例如，银屑病是一种皮肤病，……） **Example:** 下面是回答问题对应医学知识的例子： Question: 疹子会传染给其他人吗？我现在要怎么办呢？ Medical knowledge:从您提供的信息来看，疹子可能是由药物引起的皮疹。药物引起的皮疹是一种皮肤反应，通常与特定药物的过敏反应有关。这种类型的皮疹通常不会传染给其他人。建议您立即停止使用可能导致皮疹的药物，并咨询医生以获得适当的治疗和建议。 Question:药物引起的疹子的治疗方式是怎么样的呢？ Medical knowledge:药物引起的疹子是一种由药物引发的皮肤反应。治疗方式包括： 1. 停用引发药物：立即停用可能引起皮疹的药物。2. 对症治疗：抗组胺药物：用于缓解瘙痒，如氯雷他定、开瑞坦。局部类固醇：用于减轻炎症，如氢化可的松膏。3. 观察与随访：观察皮疹的变化，并定期跟踪。4. 避免再接触：避免再次使用相同的药物。5. 咨询医生：如症状严重或伴有全身症状，应立即就医。 Question:皮肤瘙痒是长了疱疹吗？用什么药？！ Medical knowledge:皮肤瘙痒并不一定意味着长了疱疹。疱疹是一种由疱疹病毒（单纯疱疹病毒或带状疱疹病毒）感染引起的病毒性皮肤病，通常表现为水疱和局部瘙痒。针对疱疹的治疗建议包括：抗病毒药物：使用阿昔洛韦、伐昔洛韦或泛昔洛韦等药物。局部治疗：可使用含有抗病毒成分的外用药膏，如阿昔洛韦乳膏。如果不确定皮肤瘙痒是否由疱疹引起，建议就医进行专业诊断。 **Input:** Question: 我打完针就有皮肤瘙痒的现象，这是怎么回事呢？ Medical knowledge:
Intermediary	药疹是一种常见的药物不良反应，通常是由身体对某种药物产生过敏反应引起的。皮肤瘙痒可能是药疹的早期症状之一。药疹的症状可能包括皮肤红斑、丘疹、斑丘疹或荨麻疹等形式。出现皮肤瘙痒后，建议尽快停止可疑药物的使用，并联系医生进行进一步评估和治疗。如果药疹的症状加重或伴随其他全身性反应如发热、呼吸困难等，应立即就医。 (temperature=0.8) 药疹是一种药物引发的皮肤反应，可能表现为瘙痒、红斑、皮疹等症状。瘙痒通常是由于药物过敏或药物引发的皮肤炎症所致。除了瘙痒，药疹还可能伴有皮肤的其他改变，如红斑、丘疹或风团等。处理方法包括：停用药物：如果怀疑某种药物导致药疹，应立即停用该药物。抗组胺药物：可使用抗组胺药物如氯雷他定或开瑞坦来缓解瘙痒。局部治疗：局部类固醇药膏如氢化可的松膏可以减轻炎症和瘙痒。观察与随访：观察症状的变化，如有加重或出现全身症状，应及时就医。就医咨询：建议您尽快咨询医生，进行专业评估和获得适当的治疗建议。(temperature=0.5)
Ground truth	药疹是一种过敏性皮肤病，这种皮肤病的外界致病因素很多，比如药物、环境、饮食、心情等都有一定的关系的，但是最根本的原因还是本身的过敏性体质。建议可以采取中药的治疗方法，中医的增免脱敏疗法治疗这种皮肤病效果相对的是比较理想的，它的治疗的机理主要是通过口服中药来增加患者自身的抵抗力。

Figure 8: Chinese medicine case example.



Query	After getting an injection, I have experienced itching on my skin. What could be causing this?
Prompt	**Instruction:** Please answer the patient's question with professional medical knowledge. (Format like: Psoriasis is a skin disease,……) **Example:** Here are examples of medical knowledge corresponding to answering questions: Question: Can a rash spread to others? What should I do now? Medical knowledge: Based on the information provided, the rash may be caused by medication. Drug-induced rashes are usually an allergic reaction to a specific medication and are not contagious. You should stop taking the medication that may be causing the rash and consult a doctor for appropriate treatment and advice. Question: What is the treatment for a drug-induced rash? Medical knowledge: Treatment for a drug-induced rash includes:Discontinuing the drug: Stop taking the medication that may have caused the rash.Symptomatic treatment:Antihistamines: To relieve itching, such as loratadine or cetirizine.Topical steroids: To reduce inflammation, such as hydrocortisone cream.Monitoring and follow-up: Observe the rash's progress and follow up regularly.Avoiding re-exposure: Avoid using the same medication again.Consulting a doctor: Seek medical attention if symptoms are severe or if there are systemic symptoms. Question: Is skin itching a sign of herpes? What medication should be used? Medical knowledge: Skin itching does not necessarily indicate herpes. Herpes is a viral skin condition caused by herpes simplex virus or varicella-zoster virus, usually presenting as blisters and localized itching. Treatment for herpes includes:Antiviral medications: Such as acyclovir, valacyclovir, or famciclovir.Topical treatment: Antiviral creams, such as acyclovir cream.If you are unsure whether your skin itching is due to herpes, it is recommended to see a doctor for a professional diagnosis. Question: After getting an injection, I have experienced itching on my skin. What could be causing this? Medical Knowledge:
Intermediary	Drug rash is a common adverse drug reaction, usually caused by an allergic response to a medication. Skin itching can be one of the early symptoms. Symptoms may include redness, papules, maculopapular rash, or urticaria. If itching occurs, it is recommended to stop the suspected medication and consult a doctor for further evaluation and treatment. If symptoms worsen or are accompanied by systemic reactions such as fever or difficulty breathing, seek medical attention immediately. (temperature=0.8) A drug rash is a skin reaction caused by medication, which may include itching, redness, and rashes. Itching is often due to drug allergies or inflammation. In addition to itching, drug rash may also involve other skin changes, such as redness, papules, or welts. Management includes:Discontinuing the drug: Stop taking the suspected medication immediately.Antihistamines: Use antihistamines like loratadine or cetirizine to relieve itching.Topical treatment: Apply topical steroids like hydrocortisone cream to reduce inflammation and itching.Monitoring and follow-up: Observe symptoms, and seek medical attention if they worsen or if systemic symptoms develop.Medical consultation: Consult a doctor for professional assessment and appropriate treatment advice. (temperature=0.5)
Ground truth	Drug rash is an allergic skin condition influenced by external factors like drug, environment, diet, and emotions, but the primary cause is an allergic constitution. It is recommended to consider traditional Chinese medicine, as its immune-enhancing desensitization therapy can be effective in treating this condition by boosting the patient's immunity.

Figure 9: English version of medicine case example.

Query		I am a retired person. After being recruited by a company, I signed a one-year agreement. Now the company has prematurely terminated the labor relationship due to insufficient investment. What compensation is there?
Ground truth		**<Article 46 of the Labor Contract Law of the People's Republic of China>:** The employer is required to pay economic compensation to the employee under any of the following circumstances:1. The employee terminates the labor contract according to Article 38 of this law.2. The employer proposes to terminate the labor contract according to Article 36 of this law, and both parties agree to the termination.3. The employer terminates the labor contract according to Article 40 of this law.4. The employer terminates the labor contract according to the first paragraph of Article 41 of this law.5. The employer terminates a fixed-term labor contract according to the first item of Article 44 of this law, except when the employer maintains or improves the conditions agreed upon in the contract and the employee does not agree to renew the contract.6. The labor contract is terminated according to items 4 or 5 of Article 44 of this law. 7. Other circumstances stipulated by laws and administrative regulations.
Iteration	**0th**	**Context:** None **Intermediary:** <Article 38 of the Labor Contract Law>: a labor contract terminates upon expiration. Article 44 also states that a labor contract ends when it reaches its expiration date. In principle, based on Article 46, an employer is not required to pay severance for termination due to contract expiration unless special circumstances apply, such as the employer failing to offer renewal terms equal to or better than the original contract, and the employee refusing to renew. **Retrieval: <Article 38 of the Labor Contract Law>**
	2nd	**Context: Question1**: I signed a three-year labor contract with my company, but they terminated my employment after six months, citing my inability to adapt. Can I apply for unemployment insurance?Intermediary1: <Article 45 of the Social Insurance Law of the PRC> states that unemployed individuals are eligible for unemployment benefits if they meet the following conditions: (1) They have participated in unemployment insurance, and both the employer and employee have fulfilled the payment obligations for at least one year; (3) They have completed unemployment registration and are actively seeking employment. **Question2**: I have worked at my company for two years, but due to financial difficulties, the company proposed terminating my contract through mutual agreement. Am I entitled to severance pay?Intermediary2: <Article 46 of the Labor Contract Law of the PRC> states that an employer must provide severance pay if (2) the employer proposes terminating the labor contract under Article 36 and both parties reach a mutual agreement. <Article 44> further states that if a fixed-term contract expires and the employee does not request renewal, the employer must provide severance pay according to the law. **Question3**: I am experiencing water leakage in my rented apartment, but the landlord refuses to fix it. Can I request a rent reduction or terminate the lease?Intermediary3: <Article 721 of the Civil Code of the PRC> states that if a rental property requires repairs that affect the tenant's normal use, the tenant may request a rent reduction or an extension of the lease term. If the property's quality issues seriously affect the tenant's safety or health, the tenant may terminate the lease at any time. **Intermediary:** <Article 45 of the Social Insurance Law of the PRC>: Unemployed individuals may receive unemployment benefits if they meet the following conditions: (1) They have legally participated in unemployment insurance, with both the employer and employee having paid contributions for at least one year; (2) Unemployment was not caused by voluntary resignation; (3) They have completed unemployment registration and are willing to seek employment. <Article 44 of the Labor Contract Law of the PRC>: When a labor contract expires, the employer may choose not to renew it, leading to contract termination. <Article 46 of the Labor Contract Law of the PRC>: Employers must provide severance pay in certain situations, including: (2) When the employer proposes to terminate the contract under Article 36, and both parties reach a mutual agreement. **Retrieval: <Article 44 of the Labor Contract Law>**
	4th	**Context: Question1**: I am a contract worker with a fixed-term labor contract. My contract has expired, and the company has decided not to renew it. Am I entitled to severance pay? Intermediary1: <Article 44 of the Labor Contract Law of the PRC> states that a labor contract terminates upon expiration. <Article 46> stipulates that the employer must provide severance pay if the contract is terminated under Article 44(1), except when the employer offers renewal under the same or better conditions and the employee refuses. **Question2:** If a company terminates a labor contract early, can the employee receive one month's salary as severance for each year worked? Intermediary2: <Article 47 of the Labor Contract Law of the PRC> states that if an employer terminates a labor contract under <Article 40 or Article 41>, the employee is entitled to severance pay. The severance is calculated as one month's salary for each full year of service. If the employee has worked for at least six months but less than a year, it is rounded up to one year. If less than six months, half a month's salary is paid. The compensation is based on the employee's average monthly salary over the past 12 months. **Question3:** I was injured while working in a mine and was classified as having an 8th-degree disability. My contract has been terminated, and my salary is around 6,000 yuan. How much compensation can I receive? Intermediary3: <Article 37 of the Work-Related Injury Insurance Regulations of the PRC> states that if an employee suffers a work-related disability classified between 7th and 10th degree, they are entitled to a one-time disability employment subsidy and a one-time work-related injury medical subsidy upon contract termination. For an 8th-degree disability, the total compensation is 11 months' salary, calculated based on the employee's average monthly wage over the past 12 months. **Intermediary:** <Article 46 of the Labor Contract Law of the People's Republic of China>: An employer shall provide economic compensation to employees under the following circumstances: (2) The employer proposes to terminate the labor contract according to Article 36 of this law and reaches an agreement with the employee. (4) The employer terminates the labor contract in accordance with Paragraph 1 of Article 41 of this law. <Article 44 of the Labor Contract Law of the People's Republic of China>: A labor contract shall be terminated under any of the following circumstances: ...(4) The employer is legally declared bankrupt. When an employer is declared bankrupt due to poor business performance or other reasons, its labor contracts are also terminated, as the bankrupt company's assets will be liquidated, making it unable to fulfill labor contracts. (5) The employer's business license is revoked, the company is ordered to close, dissolved, or voluntarily decides to dissolve in advance. In these cases, the employer loses its legal basis for continuing operations or existence, leading to the termination of labor contracts. **Retrieval: <Article 46 of the Labor Contract Law>**

Figure 10: Changes in context and intermediary across iterations.