DeepRetrieval: Powerful Query Generation for Information Retrieval with Reinforcement Learning

Pengcheng Jiang*
UIUC CS

Abstract

Information retrieval systems are crucial for enabling effective access to large document collections. Recent approaches have leveraged Large Language Models (LLMs) to enhance retrieval performance through query augmentation, but often rely on expensive supervised learning or distillation techniques that require significant computational resources and hand-labeled data. In this paper, we introduce DeepRetrieval, a novel reinforcement learning-based approach that trains LLMs to perform query augmentation directly through trial and error, without requiring supervised data. By using the retrieval recall as a reward signal, our system learns to generate effective queries that maximize document retrieval performance. Our preliminary results demonstrate that DeepRetrieval significantly outperforms existing state-of-the-art methods, including the recent LEADS system, achieving 60.82% recall on publication search and 70.84% recall on trial search tasks while using a smaller model (3B vs. 7B parameters) and requiring no supervision data. These results suggest that our reinforcement learning approach offers a more efficient and effective paradigm for information retrieval, potentially changing the landscape of document retrieval systems. code is available at https://github.com/pat-jj/DeepRetrieval.

1 Introduction

Information retrieval (IR) systems play a critical role in helping users find relevant documents within large collections. Traditional approaches to IR rely on keyword matching and statistical methods, which often struggle with understanding the semantic meaning behind user queries. Recent advances in Large Language Models (LLMs) have shown promise in addressing these limitations through query augmentation (Bonifacio et al., 2022), where LLMs expand or reformulate user queries to better capture relevant documents.

However, current LLM-based approaches to query augmentation typically employ supervised learning or distillation techniques, which have several significant limitations:

- They require expensive computational resources to generate training data, often costing thousands of dollars
- The quality of the augmented queries depends on the quality of the supervision data
- They rely on larger models to generate data for smaller ones, introducing potential biases and limitations

In this work, we introduce DeepRetrieval, a novel approach that uses reinforcement learning (RL) to train LLMs for query augmentation. Rather than relying on supervision data, DeepRetrieval allows the model to learn through direct trial and error, using the retrieval recall as the reward signal. This approach has several key advantages:

- No need for expensive supervision data generation

*This document represents a preliminary technical report. The current author list is incomplete and will be updated upon project completion.

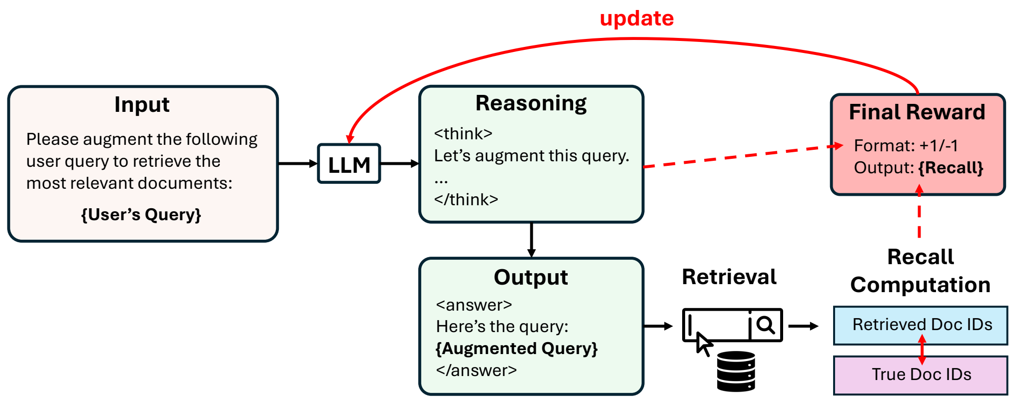

Figure 1: DeepRetrieval: The LLM generates an augmented query, which is used to retrieve documents. The recall is computed and used as a reward to update the model.

- Direct optimization for the end-goal (recall performance)
- Ability to learn effective strategies without human demonstration

Our preliminary results demonstrate that DeepRetrieval significantly outperforms existing state-of-the-art methods, including the recent LEADS system (Wang et al., 2025), achieving 60.82% recall on publication search and 70.84% recall on trial search tasks. Notably, these results were obtained using a smaller model (3B parameters) compared to LEADS (7B parameters) and without requiring any supervision data, highlighting the efficiency and effectiveness of our approach.

2 Methodology

Our DeepRetrieval approach builds upon recent advances in reinforcement learning for LLMs, applying this paradigm to the specific task of query augmentation for information retrieval. Our methodology is directly inspired by DeepSeek-R1-Zero (DeepSeek-AI et al., 2025), which demonstrated that RL can be used to train models with advanced reasoning capabilities without relying on supervised data. Figure 1 illustrates the overall architecture of our system.

2.1 Problem Formulation

Let D be a collection of documents and q be a user query. The goal of an information retrieval system is to return a set of documents $D_q \subset D$ that are relevant to q. In query augmentation, the original query q is transformed into an augmented query q' that is more effective at retrieving relevant documents.

Traditionally, this augmentation process is learned through supervised learning, where pairs of (q, q') are provided as training data. In contrast, our approach uses reinforcement learning, where the model learns to generate effective augmented queries through trial and error, similar to how DeepSeek-R1-Zero learns to solve reasoning problems.

2.2 Reinforcement Learning Framework

We formulate the query augmentation task as a reinforcement learning problem:

- **State**: The user's original query q
- **Action**: The augmented query q' generated by the model
- **Reward**: The recall achieved when using q' to retrieve documents

The model is trained to maximize the expected reward, i.e., to generate augmented queries that achieve high recall. This direct optimization for the end goal differs from supervised approaches, which optimize for similarity to human-generated or larger-model-generated augmentations.

2.3 Model Architecture and Output Structure

We utilize Qwen-2.5-3B-Instruct (Yang et al., 2024) as the base LLM for our system. The model takes the user query as input and generates an augmented query. The model is structured to first generate reasoning steps in a <think> section, followed by the final augmented query in an <answer> section in JSON format. This structured generation allows the model to consider various aspects of the query and explore different augmentation strategies before finalizing its response.

For our initial experiments, we focused on medical literature retrieval using a specialized prompt based on the PICO framework (see Appendix A for details). The JSON format with Boolean operators (AND, OR) and appropriate parentheses for grouping is required for compatibility with search systems. However, our methodology is general and can be applied to traditional IR datasets with appropriate modifications to the prompt and query format.

2.4 Reward Mechanism

Our reward function is designed to directly optimize for retrieval performance. The process works as follows:

1. The model generates an augmented query in response to a PICO framework query
2. The augmented query is executed against a document collection (PubMed or ClinicalTrials.gov)
3. Recall is computed as the proportion of relevant documents retrieved
4. A composite reward is calculated based on:
    - Format correctness (JSON structure, proper tagging)
    - Retrieval recall, with higher rewards for better recall

Specifically, our reward function uses a tiered scoring system for recall performance, as shown in Table 1.

Recall	≥ 0.7	≥ 0.5	≥ 0.4	≥ 0.3	≥ 0.1	≥ 0.05	< 0.05
Reward	+5.0	+4.0	+3.0	+1.0	+0.5	+0.1	-3.5

Table 1: Reward tiers based on recall performance. Higher recall values receive significantly larger rewards, incentivizing the model to generate more effective queries.

Additionally, correct formatting receives +1 point, while incorrect formatting receives -4 points. Importantly, if the format is incorrect (missing tags, improper JSON structure, etc.), the answer reward is not computed at all, resulting in only the formatting penalty. This reward structure strongly encourages the model to generate well-formed queries that maximize recall while adhering to the required output format.

2.5 Training Process

Our training process follows these steps:

1. Initialize the model with pre-trained weights
2. For each query in the training set:
    (a) Generate an augmented query

    (b) Execute the query against the search system
    (c) Compute the recall (proportion of relevant documents retrieved)
    (d) Use the recall-based reward to update the model
3. Repeat until convergence

This process allows the model to learn effective query augmentation strategies directly from the retrieval performance, without requiring explicit supervision. The model gradually improves its ability to transform PICO framework queries into effective search terms that maximize recall of relevant medical literature.

3 Experiments

3.1 Datasets

We evaluate our approach on two medical literature retrieval tasks:

- **Publication search**: Retrieving relevant medical publications from PubMed based on user queries expressed in the PICO framework
- **Trial search**: Retrieving relevant clinical trials from ClinicalTrials.gov based on similar PICO framework queries

These datasets are particularly challenging for information retrieval systems due to the specialized terminology and complex relationships in medical literature. For each query, we have a set of ground truth relevant documents (identified by their PMIDs) that should ideally be retrieved by the augmented query.

3.2 Evaluation Metrics

We use recall as our primary evaluation metric, which measures the proportion of relevant documents retrieved. Specifically, we report:

- **Recall (Publication search)**: The proportion of relevant publications retrieved
- **Recall (Trial search)**: The proportion of relevant clinical trials retrieved

3.3 Baselines

We compare our approach against several baselines:

- **GPT-4o**: Various configurations (Zero-shot, Few-shot, ICL, ICL+Few-shot)
- **GPT-3.5**: Various configurations (Zero-shot, Few-shot, ICL, ICL+Few-shot)
- **Haiku-3**: Various configurations (Zero-shot, Few-shot, ICL, ICL+Few-shot)
- **Mistral-7B** (Jiang et al., 2023): Zero-shot configuration
- **LEADS** (Wang et al., 2025): A state-of-the-art approach for medical literature retrieval using Mistral-7B trained with distillation

3.4 Implementation Details

We implemented DeepRetrieval using the VERL framework[1], which is an open-source implementation of the HybridFlow RLHF framework Sheng et al. (2024).

Our training configuration used Proximal Policy Optimization (PPO) with the following key parameters:

- Base model: Qwen-2.5-3B-Instruct (Yang et al., 2024)

[1]https://github.com/volcengine/verl

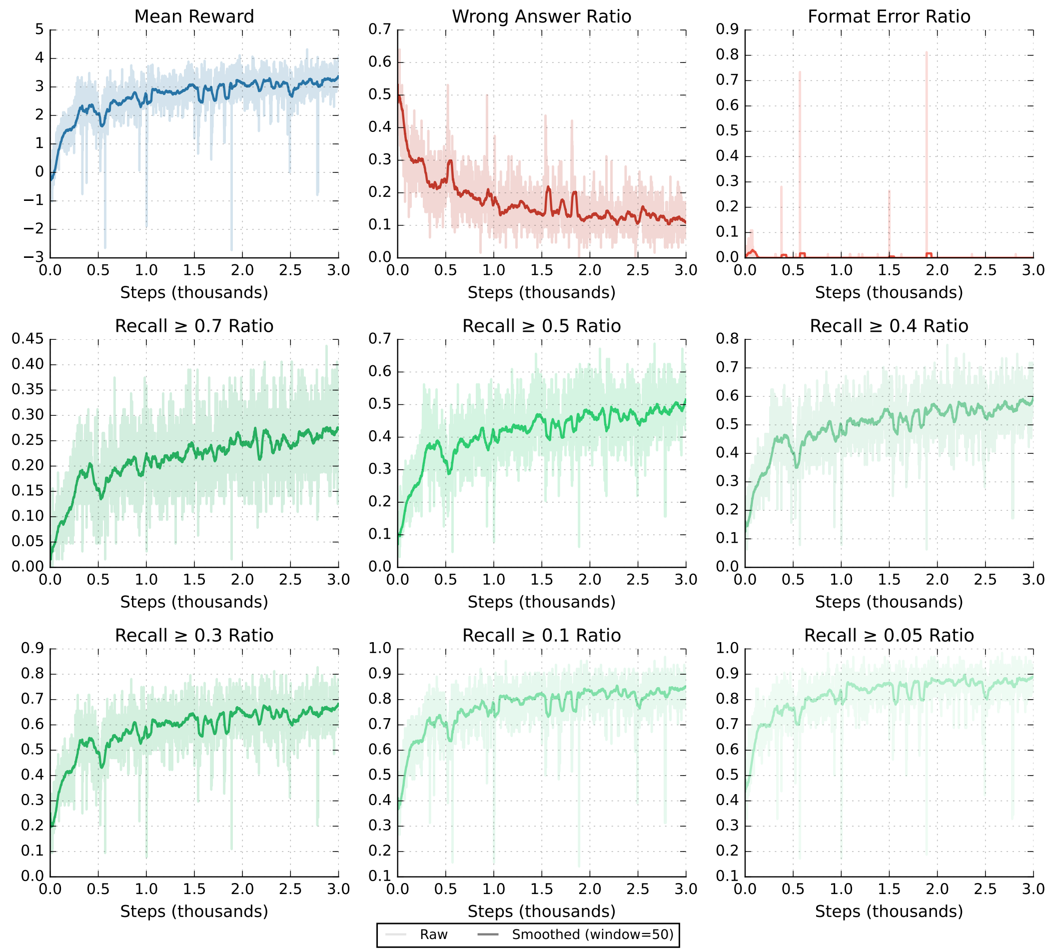

Figure 2: Training dynamics of DeepRetrieval. Recall computation is based on PubMed retrieval during the training.

- PPO mini batch size: 16
- PPO micro batch size: 8
- Learning rate: 1e-6 for actor, 1e-5 for critic
- KL coefficient: 0.001
- Maximum sequence lengths: 500 tokens for both prompts and responses

We trained the model on two NVIDIA A100 80GB PCIe using FSDP strategy with gradient checkpointing enabled to optimize memory usage. The training process ran for 5 epochs.

As shown in Figure 2, the training dynamics reveal steady improvement in performance metrics over the course of training. The mean reward (top left) shows a consistent upward trend, starting negative but quickly becoming positive and continuing to improve throughout training. Simultaneously, both wrong answer ratio (top center) and format error ratio (top right) decrease substantially, indicating that the model is learning to generate properly structured queries that retrieve relevant documents.

The most notable trend is the systematic improvement across all recall thresholds. The ratio of queries achieving high recall values ($\geq$0.5, $\geq$0.7) increases steadily, with the highest recall tier ($\geq$0.7) growing from nearly zero to approximately 0.25 by the end of training. Mid-range recall ratios ($\geq$0.4, $\geq$0.3) show stronger growth, reaching around 0.6-0.7, while lower

Model	Method	Recall (Publication)	Recall (Trial)
GPT-4o	Zero-shot	5.79	6.74
	Few-shot	7.67	4.69
	ICL	19.72	14.26
	ICL+Few-shot	11.95	7.98
GPT-3.5	Zero-shot	4.01	3.37
	Few-shot	4.15	3.34
	ICL	18.68	13.94
	ICL+Few-shot	7.06	5.54
Haiku-3	Zero-shot	10.98	11.59
	Few-shot	14.71	7.47
	ICL	20.92	24.68
	ICL+Few-shot	19.11	9.27
Mistral-7B	Zero-shot	7.18	8.08
LEADS	Zero-shot	24.68	32.11
DeepRetrieval	Zero-shot	**60.82**	**70.84**

Table 2: Comparison of different models and methods on publication search and trial search tasks. Bold numbers indicate the best performance.

recall thresholds (≥0.1, ≥0.05) quickly approach and stabilize near 0.8-0.9. This progression clearly demonstrates how reinforcement learning gradually enhances the model's ability to generate effective query augmentations by directly optimizing for retrieval performance.

4 Results

4.1 Main Results

Table 2 presents the main results of our experiments. DeepRetrieval achieves 60.82% recall on publication search and 70.84% recall on trial search, significantly outperforming all baselines, including the state-of-the-art LEADS system.

4.2 Analysis

Several key observations emerge from our results:

- **Superior performance**: DeepRetrieval outperforms LEADS by a large margin (60.82% vs 24.68% on publication search, 70.84% vs 32.11% on trial search), despite using a smaller model (3B vs 7B parameters).
- **Cost efficiency**: Unlike LEADS, which requires expensive distillation (estimated at over $10,000 for training data generation), DeepRetrieval requires no supervision data, making it significantly more cost-effective.
- **Generalizability**: The consistent performance across both publication and trial search tasks suggests that our approach generalizes well across different retrieval scenarios.
- **Effectiveness of structured generation**: The <think>/<answer> structure allows the model to reason through complex queries before finalizing its response, improving overall quality.

5 Discussion

5.1 Why Reinforcement Learning Works

The superior performance of DeepRetrieval can be attributed to several factors:

- **Direct optimization**: By directly optimizing for recall, the model learns to generate queries that are effective for retrieval, rather than queries that match some predefined pattern.
- **Exploration**: The reinforcement learning framework allows the model to explore a wide range of query augmentation strategies, potentially discovering effective approaches that might not be present in supervised data.
- **Adaptive learning**: The model can adapt its augmentation strategy based on the specific characteristics of the query and the document collection, rather than applying a one-size-fits-all approach.
- **Structured reasoning**: The two-phase generation approach with separate thinking and answering components allows the model to work through the problem space before committing to a final query.

5.2 Limitations and Future Work

While our preliminary results are promising, several limitations and directions for future work remain:

- **Evaluation on classic IR datasets**: Our current experiments are focused on medical literature retrieval with the PICO framework. A critical next step is to evaluate DeepRetrieval on standard IR benchmarks such as MS MARCO, TREC, and BEIR to test its effectiveness in more general retrieval scenarios.
- **Comparison with more advanced methods**: Additional comparisons with recent query augmentation approaches would further validate our findings.
- **Model scaling**: Investigating how the performance scales with larger models could provide insights into the trade-offs between model size and retrieval performance.
- **Reward engineering**: Exploring more sophisticated reward functions that incorporate other metrics beyond recall (e.g., precision, nDCG) could potentially lead to further improvements.
- **Integration with retrieval pipelines**: Exploring how DeepRetrieval can be integrated into existing retrieval pipelines, including hybrid approaches that combine neural and traditional retrieval methods.

6 Conclusion

In this paper, we introduced DeepRetrieval, a novel reinforcement learning-based approach for query augmentation in information retrieval. By training a 3B parameter language model to directly optimize for retrieval recall, we achieved state-of-the-art performance on medical literature retrieval tasks, significantly outperforming existing approaches that rely on supervised learning or distillation.

The key innovation of our approach lies in its ability to learn effective query augmentation strategies through trial and error, without requiring expensive supervision data. This makes DeepRetrieval not only more effective but also more cost-efficient than existing methods.

Our results suggest that reinforcement learning offers a promising paradigm for information retrieval, potentially changing the landscape of document retrieval systems. We believe that this approach can be extended to other information retrieval tasks and domains, offering a general framework for improving retrieval performance across a wide range of applications.

References

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144*, 2022.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Zifeng Wang, Lang Cao, Qiao Jin, Joey Chan, Nicholas Wan, Behdad Afzali, Hyun-Jin Cho, Chang-In Choi, Mehdi Emamverdi, Manjot K. Gill, Sun-Hyung Kim, Yijia Li, Yi Liu, Hanley Ong, Justin Rousseau, Irfan Sheikh, Jenny J. Wei, Ziyang Xu, Christopher M. Zallek, Kyungsang Kim, Yifan Peng, Zhiyong Lu, and Jimeng Sun. A foundation model for human-ai collaboration in medical literature mining, 2025. URL https://arxiv.org/abs/2501.16255.

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

A PICO Prompt

For our medical literature retrieval experiments, we used the following specialized prompt:

```
The Assistant is a clinical specialist. He is conducting
research and doing a medical literature review. His task
is to create query terms for a search URL to find relevant
literature on PubMed or ClinicalTrials.gov.

The research is defined using the PICO framework:
P: Patient, Problem or Population - Who or what is the
   research about?
I: Intervention - What is the main intervention or
   exposure being considered?
C: Comparison - What is the intervention being compared to?
O: Outcome - What are the relevant outcomes or effects
   being measured?

The Assistant should show his thinking process in <think>
</think> tags. The Assistant should return the final answer
in JSON format in <answer> </answer> tags,
For example:
<think>
[thinking process]
</think>
<answer>
{
    "query": "...."
}
</answer>.
Note: The query should use Boolean operators (AND, OR) and
parentheses for grouping terms appropriately.
```

This specialized prompt is tailored to medical literature retrieval but could be adapted for other IR domains by modifying the task description and query structure guidance accordingly.