Language Bias in Multilingual RAG: A Case Study in the Japanese Medical Domain

Anonymous ACL submission

Abstract

Despite the significant achievements of LLMs in recent years, their performance in low-resource language-domain pairs remains less than satisfactory. Although RAG is often considered a solution, we identify the paradox: the LLM's poor performance in low-resource language-domain pairs is due to a lack of corpora, but RAG also relies on comprehensive and high-quality corpora. We show that this paradox could lead to failure of RAG in certain low-resource language-domain pairs, like the Japanese medical domain. We propose to use high-resource corpora to enhance the knowledge coverage. We also identify and address the language bias issue when using multilingual corpora, which prevents the RAG framework from fully utilizing the multilingual corpus. Through our proposed RAG framework and reranker training method, the RAG performance of LLMs is improved by 4.36-7.96 percentage point on JMedBench.

1 Introduction

In recent years, the development of large language models (LLMs) has revolutionized a broad range of natural language processing (NLP) tasks. However, in certain low-resource language-domain pairs, such as the Japanese medical domain, LLMs still have room for improvement. According to Jiang et al. (2024), LLMs struggle with several tasks in the Japanese medical domain, including multiple-choice question answering (MCQA), named entity recognition (NER), and document classification (DC). Their performances on Japanese tasks are worse than equivalent tasks in English (Pal et al., 2022; Jin et al., 2019).

The main reason for the poor performance of LLMs in low-resource language-domain pairs is the lack of training data. Retrieval augmented generation (RAG) is believed to be an efficient method to improve the performance of generation tasks in such domains (Gao et al., 2023). RAG includes a retrieval module that obtains knowledge from an external trustful database to assist LLMs in generating the answer. This module ensures the answers are grounded in the retrieved evidence, enhancing the relevance and accuracy of the output. However, RAG also requires high-quality and comprehensive corpora, which is lacking in low-resource language-domain pairs. This paradoxically suggests that RAG may not be an ideal solution for tasks in low-resource language-domain pairs. To solve this low-resource paradox, we propose using rich English resource in the same domain to enhance the coverage of knowledge.

In this paper, we will use the Japanese medical domain as a case study, aiming to enhance the performance of LLMs in low-resource language-domain pairs with multilingual RAG frameworks. We propose three pipelines to improve the performance of RAG with multilingual corpora, as shown in Figure 1. Figure 1(a) provides a monolingual RAG baseline. Figure 1(b) provides a naive cross-lingual RAG pipeline, which is also the most common cross-lingual RAG pipeline at present (Chirkova et al., 2024; Li et al., 2024). However, we find that language bias exists within this pipeline. Language bias refers to that when using a query in a certain language, the retrieved paragraphs are highly likely to be in the same language, leading to insufficient use of the full multilingual corpora. To solve the language bias, we propose a translation-based cross-lingual RAG pipeline in Figure 1(c), which effectively eliminates language bias and improves the utilization of multilingual knowledge. However, this approach relies heavily on a strong translation system. To avoid the performance loss caused by the translation system, and also to further improve the quality of retrieved documents, we propose a reranker-based cross-lingual pipeline in Figure 1(d).

Overall, we aim to improve the performance of

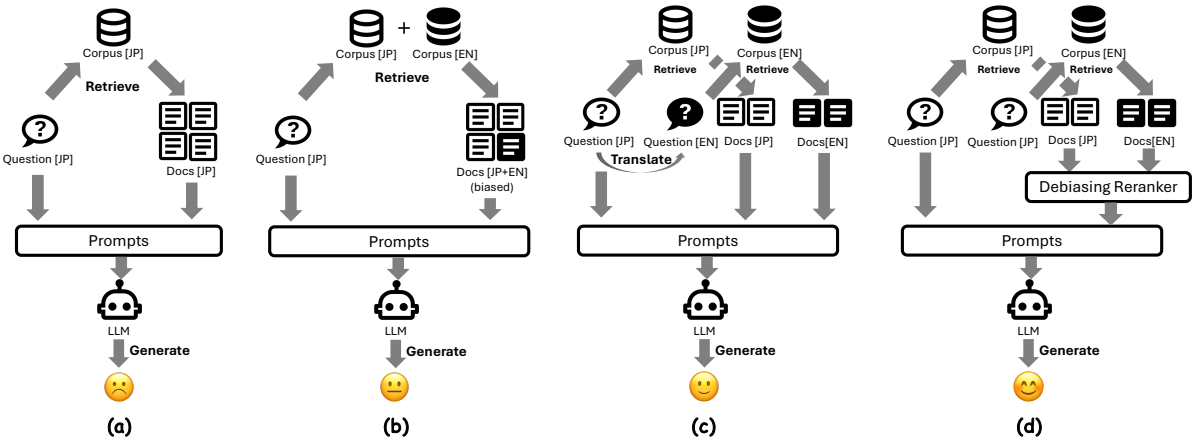

Figure 1: Four RAG pipelines experimented in this paper. Figure 1(a) provides a monolingual RAG baseline, and Figure 1(b), 1(c), 1(d) are the proposed cross-lingual RAG pipelines.

LLMs in the Japanese medical domain using cross-lingual RAG framework. The contributions in this paper include the following:

- We indicate and verify the low-resource paradox, which infers that RAG is not an effective solution to low-resource language-domain pairs, and propose using multilingual corpus to solve it.
- We discover the problem of language bias in multilingual RAG and introduce translation-based and reranker-based pipelines as solutions.
- We propose an innovative training method for multilingual rerankers that can eliminate the language bias among multilingual corpora and improve the quality of retrieved documents as well.

2 Monolingual RAG: Evidence of the Low-Resource Paradox

In this paper, we will use the Japanese medical domain as an example of low-resource language-domain pairs. We will use JMedBench, a recently released comprehensive benchmark in the Japanese medical field, to evaluate the performance of various models both in non-RAG and RAG frameworks. In this section, we will first give an introduction to JMedBench, and then test the performance of LLMs on JMedBench within the RAG framework to show the existence of the low-resource paradox.

2.1 Introduction to JMedBench

JMedBench (Jiang et al., 2024) includes 20 Japanese datasets across five task types with 38K testing samples in total. It includes native Japanese datasets like IgakuQA (Kasai et al., 2023), which compiles the Japanese medical lincensing examination in past years. It also includes high-quality datasets translated from English like MedMCQA (Pal et al., 2022), MedQA (Jin et al., 2021), PubMedQA (Jin et al., 2019), etc.

Considering that multiple-choice question answering (MCQA) tasks can comprehensively assess LLMs' understanding of medical knowledge while allowing for quick and accurate evaluation, in this paper we will primarily use MCQA datasets to evaluate the performance of LLMs and the RAG frameworks.

2.2 Experiment Settings

LLMs We follow the categorization of LLMs in JMedBench paper (Jiang et al., 2024) and select the most representative model in each category according to their performances: **general LLMs in non-Japanese languages** (Llama3-8B (Dubey et al., 2024), Qwen-2-7B (Yang et al., 2024)), **biomedical LLM in non-Japanese languages** (Meditron-7B (Chen et al., 2023)), **Japanese general LLMs** (llm-jp-v3-13B (Aizawa et al., 2024)), **Japanese biomedical LLM** (MMed-Llama3-8B (Qiu et al., 2024)).

Prompts We conduct experiments using zero-shot and few-shot prompts. In zero-shot prompts, we sequentially list the context (retrieved documents), question, options, and then let the LLM

provide an answer. Detailed prompt templates are presented in Table 5, Appendix A. In few-shot prompts, we follow the setting in the JMedBench paper, where samples are randomly selected from the training set and included as few-shot examples before the actual question.

Corpora We basically use biomedical-related corpora as the knowledge base for RAG. The Japanese corpus includes the following: MSD manual [1], J-Dream abstracts [2] and clinical guidelines released by academic societies.

Retriever In this paper, we hope to discover the potential of multilingual corpora, so we will mainly focus on the performance of mContriever (Izacard et al., 2021), which is a multilingual, unsupervised dense retrieval model.

2.3 Experiment Results

For a clearer visualization of monolingual RAG's performance, we directly display the performances of the monolingual RAG framework with different LLM, corpus and number of retrievals in Figure 2. Detailed data can be found in Table 6, Appendix A. It is notable that the RAG framework does not improve LLMs' performance; on the contrary, it decreases the accuracy of LLMs' responses in most cases. We mainly attribute the failure to issues with the corpora and the retriever. On one hand, the corpora cannot cover all the knowledge required to answer these questions. Therefore, in some cases, the RAG pipeline may retrieve paragraphs that are irrelevant to the question, thereby affecting the model's ability to provide correct answers. On the other hand, only the retriever cannot retrieve the paragraphs that best meet the needs of the questions.

Therefore, in the next section, we will first utilize a multilingual corpus to enhance the knowledge coverage of the knowledge base; then, we will introduce a new reranker to improve the retrieval effectiveness, taking into account the characteristics of the multilingual corpus.

3 Multilingual RAG: Solution to the Low-Resource Paradox

3.1 Proposed Pipelines

As discussed in the previous section, the monolingual RAG framework underperforms in the Japanese medical domain, mainly for two reasons:

[1] https://www.merckmanuals.com/
[2] https://jdream3.com/

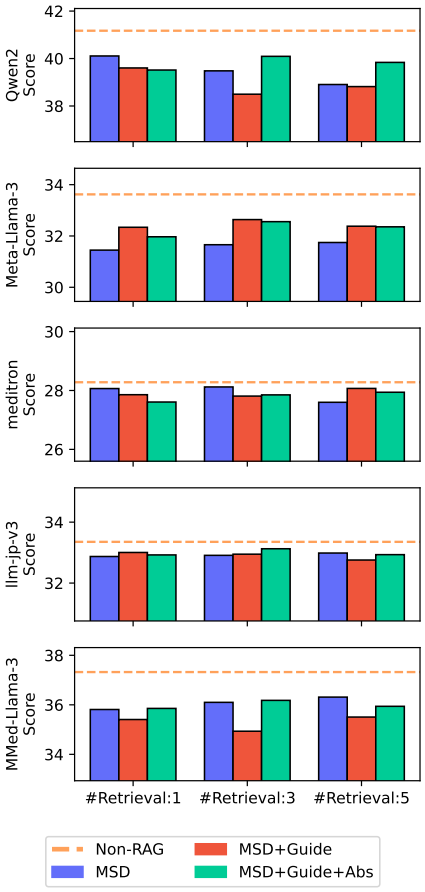

Figure 2: Comparison of zero-shot performances of the monolingual RAG framework.

the insufficiency of the corpus and the inadequacy of the retriever. Therefore, we propose three new multilingual RAG pipelines for the Japanese medical field in Figure 1, attempting to address the low-resource paradox.

Naive Cross-Lingual RAG Pipeline To address the insufficiency of the monolingual corpus, we propose to add English corpora to the Japanese corpora and use a multilingual retriever to directly retrieve reference paragraphs from it, as shown in Figure 1(b). However, our experiments reveal that this approach has a significant issue: language bias. In other words, when the query is in Japanese, the multilingual retriever tends to retrieve Japanese paragraphs. This makes the inclusion of English corpora pointless, as the RAG framework is likely to continue referencing Japanese paragraphs.

Translation-Based Cross-Lingual RAG Pipeline Language bias in multilingual retrievers is indeed difficult to eliminate because language is inherently a crucial component of the sentence embedding. We propose to first translate all Japanese queries

into English. Then, use the Japanese and English queries to retrieve paragraphs from corpora in their respective language. Finally, merge the retrieved paragraphs from both languages based on their retrieval scores. This pipeline is presented in Figure 1(c), and is proven to effectively solve the inherent language bias in the retriever.

Reranker-Based Cross-Lingual RAG Pipeline Although the translation-based pipeline can simultaneously address the issues of insufficient corpus and language bias, it still faces a few challenges. First, we haven't solved the issue of the retriever's inadequacy, as it still struggles to identify gold paragraphs that can effectively aid in the question answering. Second, not all LLMs have strong translation capabilities, and improper translations can further diminish the retriever's performance. In the translation-based pipeline, we directly used the datasets already translated in JMedBench, thus avoiding this issue. However, in practical applications, we should still strive to minimize the performance loss caused by the translation process.

Therefore, we propose a reranker-based cross-lingual RAG pipeline in Figure 1(d). We first use the original Japanese query to retrieve paragraphs from the Japanese and English corpora, independently. Then, we train a reranker to rerank these paragraphs. This pipeline can rank retrieved English and Japanese paragraphs without language bias, while also avoiding the performance loss during the translation process.

3.2 Reranker Training

Motivation The reranker is a common component in the RAG pipeline, responsible for the second stage of ranking the documents retrieved by the retriever. Although rerankers indeed improve the performance of the RAG framework (Pradeep et al., 2022; Khattab and Zaharia, 2020), we hope to understand the underlying reasons for their necessity. We hypothesize that, for the RAG framework, the most helpful paragraph is the one most relevant to the question. An example is that a question and itself share the highest relevance because their sentence embedding are identical, but the question itself cannot provide any useful information for answering it. Instead, the most helpful paragraph should be the one most relevant to the ground truth answer, as it will guide the question in the right direction. In other words, the paragraphs which are retrieved with the answer can assist LLMs answering the questions most effectively.

However, it is impossible to use the ground truth answer for retrieval in practical scenarios, but we can still use the "retrieve with the answer" method to construct training data for rerankers and provide an upper bound for RAG performance. We will elaborate on how to use this method to train a reranker and verify the correctness of this hypothesis.

Dataset Construction In our reranker, the input is a query and a paragraph, and the output is a score representing how well the paragraph can assist the query in generating the correct answer. To train such a reranker, we construct a dataset using the "retrieve with the answer" method.

Queries are sourced from the training sets of different datasets in JMedBench. We chose a subset of questions from different datasets to serve as query input.

Paragraphs consist of golden paragraphs and random paragraphs. We first use the ground truth answer for each question to perform retrieval from the corpora, obtaining the golden paragraphs for each query. To ensure a balanced paragraph distribution, we also sample several random paragraphs from the corpora for each query.

We use the relevance between the ground truth answer for each question and the paragraphs (inner product of the sentence embeddings) as the scores. Notably, the training dataset for the reranker is also multilingual, aiming to eliminate language bias through the reranker. Thus, if a paragraph is in Japanese, we compute its relevance to the Japanese ground truth answer; if it is in English, we compute its relevance to the corresponding English ground truth answer.

Using the above method, we construct a dataset of 688K entries, with 80% used as the training set and 20% as the validation set.

Training Settings We use a multilingual BERT model[3] with an additional fully-connected layer as the base model, and then finetune it with the training set described before. For the model input, we concatenate the query and paragraph with the [SEP] token. We use mean squared error (MSE) as the loss function and Adam as the optimizer with learning rate of 6×10^{-6}. We finetune the model for 30 epochs using eight NVIDIA A100-SXM4-40GB, which takes around 14 hours.

[3] https://huggingface.co/google-bert/bert-base-multilingual-uncased

3.3 Experiment Settings

Datasets We will test our pipelines on these four translated datasets from JMedBench in the main experiments: MedMCQA_JP, USMLE-QA_JP, MedQA_JP, MMLU-medical_JP.

LLMs & Prompts We will retain the settings in Section 2.2.

Corpora We will retain the corpora in Section 2.2 as the Japanese knowledge source. Besides, we will use English corpora to supplement the Japanese ones. The English corpora reference the settings in MedRAG (Xiong et al., 2024) and include the following: Meditron training corpora [4], biomedical abstracts from PubMed [5], articles from StatPearls [6] and English medical textbooks (Jin et al., 2021).

Corpus	Lang.	#Chunk	#Chunk pct.(%)	
MSD	JP	36K	0.13	10.49
GDL		376K	1.30	
Abs		2.6M	9.05	
MDT	EN	49K	0.17	89.51
PBM		25M	87.78	
SP		324K	1.12	
TB		125K	0.44	

Table 1: Statistical information on the number of chunks for each corpus, including MSD, Clinical Guideline (GDL), J-Dream Abstract (Abs), Meditron training corpora (MDT), PubMed(PBM), StatPearls (SP) and English textbook (TB).

The statistics of all corpora are presented in Table 1. We can observe that the English corpus makes up about 90% of the entire corpus, thereby covering a broader range of knowledge. Due to the dominance of English in scientific research, this phenomenon is quite common across various fields. Therefore, our approach has the potential to be extended to more low-resource language-domain pairs.

3.4 Evidence of Language Bias

In this section, we will verify the hypothesis of language bias: the retriever tends to retrieve paragraphs in the same language as the query.

Figure 3 presents the results of our investigation into language bias. We use Japanese questions to retrieve paragraphs from the multilingual corpus,

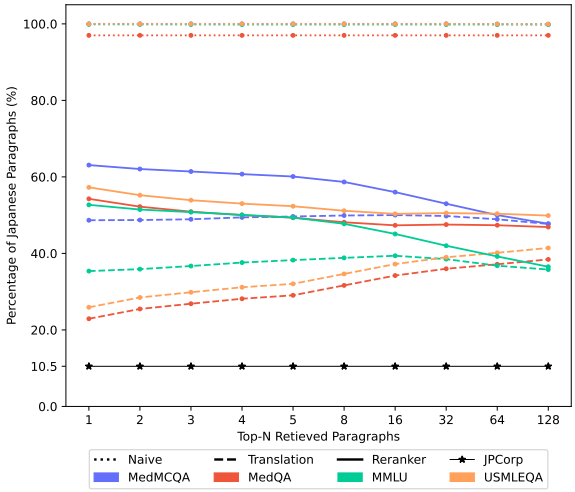

Figure 3: The average proportion of Japanese paragraphs among the top N paragraphs. Line styles represent different pipelines, while colors represent different datasets. JPCorp line indicates the proportion of Japanese paragraphs in the entire corpus.

and the proportion of Japanese paragraphs can reflect whether language bias exists. We can draw the following conclusions from this figure.

Firstly, language bias is a severe problem. Although Japanese paragraphs make up just 10.49% of the total corpus, almost only Japanese paragraphs are retrieved with Japanese queries in the naive pipeline, because language is an important component of the sentence embedding. It means the addition of the English corpus will hardly bring any changes to the results, even though it contains a vast amount of new knowledge.

Secondly, translation-based and reranker-based pipelines can effectively address the issue of language bias. By using our proposed pipelines, the proportion of retrieved Japanese paragraphs can be reduced from nearly 100% to around 40%-60%, with variation depending on the pipeline, the dataset and the number of retrievals.

In summary, language bias is a severe issue to multilingual RAG. We are confident that employing a multilingual dataset for RAG is advantageous, as it enables a broader spectrum of knowledge and diverse viewpoints in multiple languages. However, this is not a naive problem; we must use a more suitable pipeline to deal with it.

3.5 Experiment Results

Table 2 and Table 7 respectively present the performance of our proposed RAG pipelines with five retrieved paragraphs under zero-shot and few-shot

[4] https://huggingface.co/datasets/epfl-llm/guidelines

[5] https://pubmed.ncbi.nlm.nih.gov/

[6] https://www.statpearls.com/

settings across different tasks. To provide a clearer insight into the results, we visualized the overall performance of the LLMs, which is presented in Figure 4. "JPCorp" represents that the pipeline only use the Japanese corpus, and "FullCorp" represents the full corpus. We will analyze this data from the following perspectives.

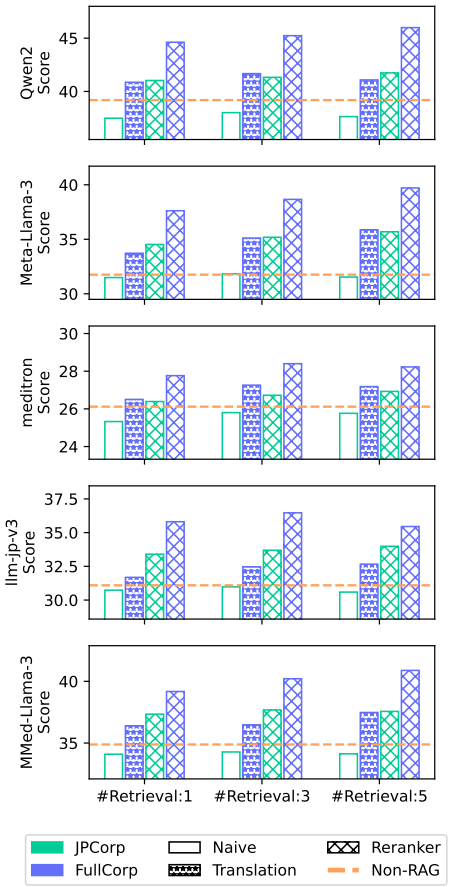

Figure 4: Comparison of zero-shot performances of the RAG pipelines with different hyperparameters.

Corpus We can observe that using the full corpus generally results in better performance than using the Japanese-only corpus, with other conditions controlled. Using high-resource language knowledge to supplement the corpus can effectively address the paradox of RAG in low-resource language-domain pairs.

Surprisingly, according to the experimental results, LLMs are not only able to extract knowledge from paragraphs in different languages, but also capable of responding in the same language as the question, which exceeds our expectations.

Pipeline Due to language bias, the naive RAG pipeline's performance is almost same as the monolingual RAG pipeline. However, both the translation-based pipeline and the reranker-based pipeline effectively aid the LLM in correctly answering questions, thereby enhancing overall performance. Since our trained reranker can better understand which paragraphs are more helpful compared to a retriever without a reranker, the reranker-based pipeline achieves the best performance in most cases.

The construction of the upper bound is similar to the method used for constructing the training set for the reranker. The upper bound is significantly higher than all the results under the same conditions. This also proves our hypothesis: the most helpful paragraph should be the one most relevant to the ground truth.

Number of Retrievals Compared to monolingual RAG results, the LLM's performance with a reranker-based pipeline shows greater monotonicity: increased number of retrievals leads to better performances. This proves that our reranker is effective, because the top paragraphs after reranking, effectively aid the LLM in answering questions well. Therefore, moderately increasing the number of retrievals is beneficial, as it helps the LLM find some of the missing good paragraphs.

3.6 Discussions

In this section, we will further explore certain interesting topics. Unless otherwise specified, the results in this section utilize Qwen-2 as the base model, the number of retrievals set as 1, and with zero-shot prompting.

Further Study on Paragraph-Answer Relevance One of the hypothesis is that, the most helpful paragraph should be the one most relevant to the ground truth. Thus, we are also interested in how LLMs perform when referencing paragraphs with different relevance to the ground truth answer. For convenience, we will refer to "relevance to the ground truth" answer as the paragraph's score. To ensure a balanced distribution of paragraph scores, we set a golden paragraph and a randomly sampled paragraph for each question. The histogram in Figure 5 shows the distribution of paragraph scores, while the line graph reflects the accuracy of question responses given paragraphs within each score range. The overall trend indicates that the higher the relevance between a paragraph and the ground truth answer, the more likely the LLM is to provide the correct answer when referencing it.

Accuracy (%)	MedM	USM	MedQ	MML	Aver (Micro)
Zero-shot Evaluation					
Qwen2	38.03	37.78	29.93	49.01	39.19
+Naive	36.82	34.96	27.65	48.10	37.64
+Translation	41.45	36.45	29.38	51.36	41.08
+Reranker(JPCorp)	41.09	38.21	33.85	50.96	41.74
+Reranker(FullCorp)	**45.48**	**43.53**	**40.41**	**52.60**	**45.99**
Upper Bound(JPCorp)	*47.60*	*57.19*	*49.88*	*56.71*	*51.34*
Upper Bound(FullCorp)	*53.07*	*65.44*	*59.15*	*62.11*	*57.77*
Llama3	31.94	30.71	24.82	36.77	31.75
+Naive	32.27	31.03	23.10	35.97	31.53
+Translation	37.03	32.05	27.18	41.74	35.86
+Reranker(JPCorp)	35.54	34.21	29.45	41.28	35.69
+Reranker(FullCorp)	**38.98**	**36.78**	**35.52**	**46.19**	**39.71**
Upper Bound(JPCorp)	*50.32*	*59.07*	*56.25*	*52.43*	*52.95*
Upper Bound(FullCorp)	*54.67*	*66.46*	*62.61*	*55.69*	*57.81*
Meditron	27.76	25.61	21.92	25.65	26.12
+Naive	27.99	25.92	21.05	23.89	25.76
+Translation	29.52	26.16	21.05	**26.83**	27.18
+Reranker(JPCorp)	28.93	26.24	23.29	25.39	26.93
+Reranker(FullCorp)	**30.15**	**27.81**	**24.45**	26.78	**28.23**
Upper Bound(JPCorp)	*40.88*	*48.16*	*41.08*	*34.79*	*40.66*
Upper Bound(FullCorp)	*42.98*	*47.76*	*41.40*	*37.52*	*42.27*
llm-jp	30.65	31.42	25.37	35.76	31.09
+Naive	30.60	30.40	24.35	34.95	30.59
+Translation	33.80	30.48	24.90	36.88	32.66
+Reranker(JPCorp)	34.11	33.11	29.61	**37.28**	33.99
+Reranker(FullCorp)	**35.82**	**35.13**	**33.34**	36.29	**35.45**
Upper Bound(JPCorp)	*47.07*	*60.09*	*55.70*	*49.55*	*50.81*
Upper Bound(FullCorp)	*51.78*	*66.61*	*61.82*	*52.97*	*55.72*
MMed-Llama3	35.43	34.72	28.28	38.27	34.88
+Naive	34.28	33.62	27.65	38.54	34.13
+Translation	38.13	35.51	28.99	43.13	37.48
+Reranker(JPCorp)	37.69	36.33	31.81	42.03	37.56
+Reranker(FullCorp)	**41.06**	**38.74**	**36.40**	**45.01**	**40.89**
Upper Bound(JPCorp)	*48.34*	*51.77*	*46.82*	*50.03*	*48.99*
Upper Bound(FullCorp)	*50.35*	*54.67*	*49.18*	*50.72*	*50.90*

Table 2: The performance of different LLMs with RAG framework on JMedBench with zero-shot prompting. The bold text in the first column represents the model names, while the italic text indicates the names of the RAG pipeline used. The best and second-best pipeline for each model's performance are highlighted in bold and underlined, respectively. The upper bound is not included in the performance ranking.

Contributions of corpora We also hope to study the contributions of each corpus to the reranker-based pipeline, particularly the differences between the Japanese and English corpora. Table 3 provides the distribution of corpus sources for the best paragraphs retrieved by the rerank-based pipeline, as well as the accuracy of answering questions using paragraphs from each corpus. We can observe that the LLM performs quite well when Japanese paragraphs are retrieved using the reranker-based pipeline, even surpassing that with English paragraphs. One possible reason might be that the LLM is better at handling monolingual inputs. Itt also indicates that the paragraphs in the Japanese corpus are of high quality and can enhance the LLM's performance in proper conditions.

We also conduct an ablation study on the corpora in different languages. We force the LLM not to use the optimally retrieved paragraphs and instead retrieve from the corpus in the other language. When the Japanese corpus was ablated, the LLM's performance was not significantly affected; conversely, the LLM's performance significantly declines on questions originally referencing English paragraphs. It indicates that the shortcoming of the Japanese corpus is the limited knowledge

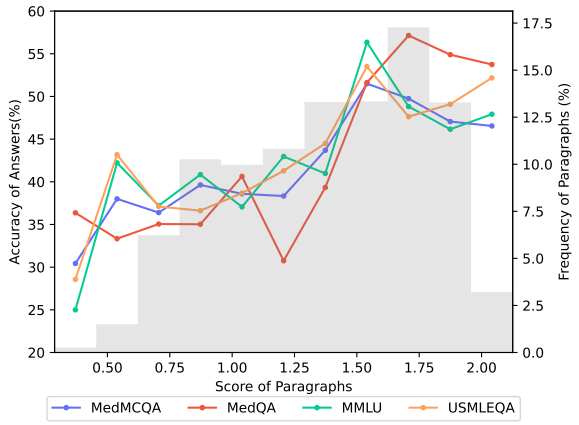

Figure 5: The score distribution of paragraphs and the relationship between accuracy and paragraph scores.

Corpus	Lang.	Retrieval (%)	Acc.(%)	Overall Acc. (%)	Ablation Acc. (%)
MSD	JP	4.58	43.65	44.76	43.11
GDL		7.83	45.17		
Abs		50.22	44.80		
MDT	EN	5.88	48.22	44.37	34.89
PBM		25.78	43.39		
SP		3.66	45.08		
TB		2.07	44.38		

Table 3: The distribution of corpus sources for the optimal paragraphs retrieved by the rerank-based pipeline and the accuracy when using each corpus source to answer the questions.

coverage, which leads to the drastic performance drop when the English corpus is forbidden.

Nevertheless, the Japanese corpus is not replacable by the English one. We list the performance of the LLM with Japanese, English, and the full corpus in Table 4 for comparison. Although the English corpus outperforms the Japanese corpus due to its comprehensiveness, its performance is not as good as the full corpus, regardless of whether a reranker is used. Diversity still matters in the RAG corpus.

We also compare the performance of our reranker with other open-source rerankers in Table 4. Our reranker outperforms other ones because it is specifically trained for the Japanese medical domain, which proves the efficacy our proposed reranker training method. On the other hand, our reranker is even more outstanding on the full corpus, due to its capability of mitigating language bias.

Reranker	JPCorp	ENCorp	FullCorp
Proposed pipelines			
Naive	37.49	40.75	-
Translation	-	-	40.86
Our Reranker	**41.03**	**43.58**	**44.62**
Open-source rerankers			
multilingual-e5 [7]	39.89	42.38	40.12
Jina-reranker-v2 [8]	40.84	42.27	41.03
gte-multilingual [9]	40.92	41.48	40.96

Table 4: Comparison of RAG performance using different corpus, pipelines and rerankers.

4 Conclusions

In this paper, we identify the low-resource paradox, demonstrating RAG's limitations for low-resource language-domain pairs, and leverage high-resource language corpora as a solution. We address language bias in multilingual RAG through translation-based and reranker-based pipelines. Additionally, we introduce a novel training method for multilingual rerankers, reducing language bias and enhancing retrieval quality across diverse languages.

We believe that using multilingual corpora for RAG will be the trends in the future. On one hand, the existence of high-resource language corpora effectively addresses the issue of knowledge coverage for low-resource language corpora. On the other hand, low-resource language corpora can offer different cultural perspectives to avoid biases inherent in high-resource language users. We hope our work can serve as a reference for further researches.

5 Related Work

Retrieval Augmented Generation (RAG) The Retrieval-Augmented Generation (RAG) framework enhances LLMs by integrating external knowledge retrieval, addressing hallucinations issues (Fan et al., 2024). RAG could employs sparse (Robertson et al., 2009) or dense (Karpukhin et al., 2020) retrieval methods, or joint optimization (Guu et al., 2020).

Cross-lingual RAG While RAG enhances LLMs' factuality and reduces hallucinations, it struggles in cross-lingual settings. The BORDIRLINES dataset (Li et al., 2024) reveals biases in geopolitical disputes, while the mRAG pipeline (Chirkova et al., 2024) highlights challenges like code-switching and fluency errors.

6 Limitations

There are still some limitations in this paper that we hope to improve upon in future work, which is listed as followed.

- In this paper, we explored only two languages: Japanese as a representative of low-resource languages and English as a representative of high-resource languages. Introducing corpora from more languages would broaden the knowledge coverage and allow for a better study of the impact of different language corpora on LLM performance.

- To our best knowledge, mContriever is currently the most popular multilingual retriever. However, the quality gap in retrieval by the retriever itself cannot be entirely compensated for by a reranker. Therefore, we hope to explore different retrievers in the future and compare their performance.

- This paper, particularly in the discussion part, we primarily examined the performance with the number of retrievals equal to one. The main reason is that the interactions among multiple retrieved documents are significantly more complex than those involving a single retrieved document. When the content of multiple articles is complementary, similar, or contradictory, the RAG pipeline may exhibit different performances. We hope to explore this further in future research.

References

Akiko Aizawa, Eiji Aramaki, Bowen Chen, Fei Cheng, Hiroyuki Deguchi, Rintaro Enomoto, Kazuki Fujii, Kensuke Fukumoto, Takuya Fukushima, Namgi Han, et al. 2024. Llm-jp: A cross-organizational project for the research and development of fully open japanese llms. *arXiv preprint arXiv:2407.03963*.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. 2023. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*.

Nadezhda Chirkova, David Rau, Hervé Déjean, Thibault Formal, Stéphane Clinchant, and Vassilina Nikoulina. 2024. Retrieval-augmented generation in multilingual settings. *arXiv preprint arXiv:2407.01463*.

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

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

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*.

Junfeng Jiang, Jiahao Huang, and Akiko Aizawa. 2024. Jmedbench: A benchmark for evaluating japanese biomedical large language models. *arXiv preprint arXiv:2409.13317*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

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

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*.

Jungo Kasai, Yuhei Kasai, Keisuke Sakaguchi, Yutaro Yamada, and Dragomir Radev. 2023. Evaluating gpt-4 and chatgpt on japanese medical licensing examinations. *arXiv preprint arXiv:2303.18027*.

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.

Bryan Li, Samar Haider, Fiona Luo, Adwait Agashe, and Chris Callison-Burch. 2024. Bordirlines: A dataset for evaluating cross-lingual retrieval-augmented generation. *arXiv preprint arXiv:2410.01171*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Ronak Pradeep, Yuqi Liu, Xinyu Zhang, Yilin Li, Andrew Yates, and Jimmy Lin. 2022. Squeezing water from a stone: a bag of tricks for further improving cross-encoder effectiveness for reranking. In *European Conference on Information Retrieval*, pages 655–670. Springer.

Pengcheng Qiu, Chaoyi Wu, Xiaoman Zhang, Weixiong Lin, Haicheng Wang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2024. Towards building multilingual language model for medicine. *arXiv preprint arXiv:2402.13963*.

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

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. *arXiv preprint arXiv:2402.13178*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

A Prompt for RAG Pipelines

The prompt template used in the pipelines is listed in Table 5.

In few-shot prompts, we follow the setting in the JMedBench paper, where samples are randomly selected from the training set and included as few-shot examples before the actual question. However, `{context}` in the few-shot examples are not instantiated, to prevent knowledge leak. We replace `{context}` with a piece of text: `***質問に関する文脈***`, which means `"*** Context related to the question ***"`.

Prompt Template	**Prompt Content**
Original	文脈：`{context}` 質問：`{question}` 選択肢：`{options}` 答え：

Translated	`Context：{context}`
	`Question：{question}`
	`Options：{options}`
	`Answer：`

Table 5: The template for MCQA task with RAG framework. In few-shot prompting, before the actual question, the prompt content will be repeated for `num_shot` more times, instantiated with `question`, `options` and `answer` sampled from the training set. However, `{context}` is replaced with a piece of text: `***質問に関する文脈***`.

B Experiment Results of the Monolingual RAG Pipeline

The performances of the monolingual RAG pipeline is listed in Table 6.

C Experiment Results (Few-shot) of the Multilingual RAG Pipeline

The few-shot performances of the multilingual RAG pipeline is listed in Table 7.

D Reranker training

Figure 6 shows the relationship between the accuracy of answers from the reranker-based pipeline and the number of reranker training epochs. As training progresses, the reranker gradually learns which paragraphs can more effectively assist in answering questions, resulting in an increase in accuracy. We believe there are many ways to optimize the training of the reranker, such as through dataset construction and the setting of the loss function. However, due to space constraints, we will not cover these in detail. Nonetheless, we are confident that our proposed reranker training method is easy to implement, scalable, can effectively enhance model performance, and can effectively mitigate language bias.

Accuracy (%)	IGA	JMM	MedM	USM	MedQ	MML	Pub	Aver (Micro)
Zero-shot Evaluation								
Qwen2	**40.75**	**43.90**	**38.03**	**37.78**	**29.93**	**49.01**	55.20	**41.17**
+*MSD*	38.56	41.46	36.22	36.45	29.46	46.50	55.00	39.48
+*MSD+GDL*	35.44	40.13	35.24	36.14	28.28	45.86	55.30	38.50
+*MSD+GDL+Abs*	39.44	42.72	37.22	36.06	27.41	48.32	55.30	40.09
Llama3	25.94	35.09	31.94	30.71	24.82	36.77	55.40	33.62
+*MSD*	28.50	33.91	30.79	31.11	23.25	35.86	39.10	31.66
+*MSD+GDL*	25.88	32.10	32.37	30.16	23.57	34.26	52.80	32.64
+*MSD+GDL+Abs*	29.05	34.23	32.85	30.79	23.64	35.81	40.20	32.56
Meditron	21.94	25.65	27.76	25.61	21.92	25.65	56.50	28.28
+*MSD*	21.75	24.70	28.40	27.34	20.74	24.37	55.00	28.12
+*MSD+GDL*	21.25	24.78	27.90	25.92	20.90	24.48	55.20	27.81
+*MSD+GDL+Abs*	22.06	24.70	27.49	26.63	20.90	24.80	55.20	27.85
llm-jp	27.56	36.43	30.65	31.42	25.37	35.76	54.60	33.35
+*MSD*	30.56	35.56	31.58	30.32	25.22	33.08	50.20	32.91
+*MSD+GDL*	29.69	33.60	31.27	31.89	25.22	33.67	52.20	32.95
+*MSD+GDL+Abs*	31.81	34.54	31.25	30.87	24.98	34.53	51.10	33.13
MMed-Llama3	31.50	36.43	35.43	34.72	28.28	38.27	**65.20**	37.32
+*MSD*	32.82	35.48	34.21	35.35	29.85	38.21	53.00	36.10
+*MSD+GDL*	31.38	34.23	34.23	34.33	26.47	36.77	50.40	34.93
+*MSD+GDL+Abs*	35.19	35.80	34.54	33.70	27.26	38.86	54.00	36.18
Few(2)-shot Evaluation								
Qwen2	51.75	**52.95**	**42.43**	**42.66**	**36.45**	**62.43**	**56.50**	**48.06**
+*MSD*	50.94	49.88	41.52	41.01	35.43	58.31	54.80	46.26
+*MSD+GDL*	50.94	49.41	41.67	40.69	34.96	58.74	55.20	46.28
+*MSD+GDL+Abs*	**52.88**	50.51	**42.43**	41.40	36.37	59.33	55.30	47.16
Llama3	33.63	37.21	35.29	34.49	27.73	41.42	52.20	36.85
+*MSD*	35.60	36.35	34.62	34.80	27.02	40.19	52.80	36.51
+*MSD+GDL*	34.44	35.48	34.64	33.46	26.55	39.87	52.10	36.02
+*MSD+GDL+Abs*	37.00	37.77	35.02	33.78	27.10	44.03	51.90	36.39
Meditron	21.94	27.38	28.90	26.63	23.17	28.06	55.00	29.37
+*MSD*	21.06	25.96	27.25	25.45	21.13	25.39	55.20	27.81
+*MSD+GDL*	21.50	26.59	26.99	25.69	20.97	25.76	55.00	27.87
+*MSD+GDL+Abs*	22.34	26.36	27.21	24.74	21.05	26.24	54.80	27.96
llm-jp	36.75	36.90	31.72	31.66	26.47	40.35	53.90	35.36
+*MSD*	36.69	34.30	31.53	30.95	25.94	39.02	51.50	34.57
+*MSD+GDL*	36.44	35.80	31.51	31.66	26.55	37.79	49.50	34.37
+*MSD+GDL+Abs*	37.94	37.20	31.51	31.97	27.26	38.48	51.30	35.03
MMed-Llama3	43.50	44.53	38.90	40.06	33.94	48.00	56.30	42.38
+*MSD*	43.19	41.62	38.03	39.43	32.44	45.43	54.80	40.98
+*MSD+GDL*	44.00	41.15	37.80	38.26	32.21	45.59	54.40	40.76
+*MSD+GDL+Abs*	46.13	41.46	38.25	39.98	32.60	45.70	55.00	41.42

Table 6: The performance of different LLMs with RAG framework on JMedBench. The bold text in the first column represents the model names, while the italic text indicates the names of the corpora used, including *MSD* (MSD Manual), *GDL* (Clinical Guideline), *Abs* (J-Dream Abstract). The best and second-best performances are highlighted in bold and underlined, respectively.

Accuracy (%)	MedM	USM	MedQ	MML	Aver (Micro)
Few(2)-shot Evaluation					
Qwen2	42.43	42.66	36.45	62.43	45.93
+*Naive*	41.69	41.56	35.98	59.06	44.60
+*Translation*	45.37	40.93	36.29	60.18	46.59
+*Reranker(JPCorp)*	44.34	42.82	38.97	61.15	46.98
+*Reranker(FullCorp)*	**46.99**	**44.50**	**41.08**	**61.49**	**48.90**
Upper Bound(JPCorp)	*47.76*	*53.26*	*46.43*	*62.69*	*51.63*
Upper Bound(FullCorp)	*51.97*	*58.76*	*54.36*	*66.72*	*56.54*
Llama3	35.29	34.49	27.73	41.42	35.39
+*Naive*	34.57	33.62	26.55	39.87	34.40
+*Translation*	40.78	34.49	28.28	**47.78**	39.52
+*Reranker(JPCorp)*	38.82	35.59	31.51	43.56	38.29
+*Reranker(FullCorp)*	**40.99**	**36.09**	**34.56**	47.70	**40.77**
Upper Bound(JPCorp)	*42.58*	*44.93*	*41.79*	*48.80*	*44.16*
Upper Bound(FullCorp)	*46.31*	*51.61*	*47.60*	*52.43*	*48.62*
Meditron	28.90	**26.63**	**23.17**	28.06	27.53
+*Naive*	27.83	23.57	19.72	23.78	25.12
+*Translation*	**29.93**	25.45	21.21	29.98	**27.99**
+*Reranker(JPCorp)*	28.15	24.98	20.85	26.35	26.21
+*Reranker(FullCorp)*	29.72	25.75	19.64	**30.24**	27.75
Upper Bound(JPCorp)	*36.31*	*38.96*	*34.09*	*33.83*	*35.83*
Upper Bound(FullCorp)	*39.21*	*41.40*	*35.43*	*36.50*	*38.39*
llm-jp	31.72	31.66	26.47	40.35	32.81
+*Naive*	31.70	31.50	26.55	37.89	32.25
+*Translation*	36.17	32.76	27.18	42.44	35.70
+*Reranker(JPCorp)*	33.53	33.31	30.18	42.39	34.93
+*Reranker(FullCorp)*	**36.78**	**35.54**	**32.75**	**43.26**	**37.41**
Upper Bound(JPCorp)	*41.48*	*52.00*	*48.70*	*48.05*	*45.54*
Upper Bound(FullCorp)	*46.43*	*59.39*	*53.97*	*51.10*	*50.48*
MMed-Llama3	38.90	40.06	33.94	48.00	40.32
+*Naive*	38.23	39.28	32.99	45.80	39.26
+*Translation*	**42.55**	37.08	33.54	51.10	42.27
+*Reranker(JPCorp)*	40.42	39.76	36.66	48.48	41.52
+*Reranker(FullCorp)*	42.11	**40.49**	**38.56**	**52.24**	**43.55**
Upper Bound(JPCorp)	*42.43*	*45.72*	*42.18*	*52.00*	*44.96*
Upper Bound(FullCorp)	*45.82*	*49.80*	*46.90*	*54.20*	*48.39*

Table 7: The performance of different LLMs with RAG framework on JMedBench with few(2)-shot prompting. The bold text in the first column represents the model names, while the italic text indicates the names of the RAG pipeline used. The best and second-best pipeline for each model's performance are highlighted in bold and underlined, respectively. It should be noted that the upper bound is not included in the performance ranking.

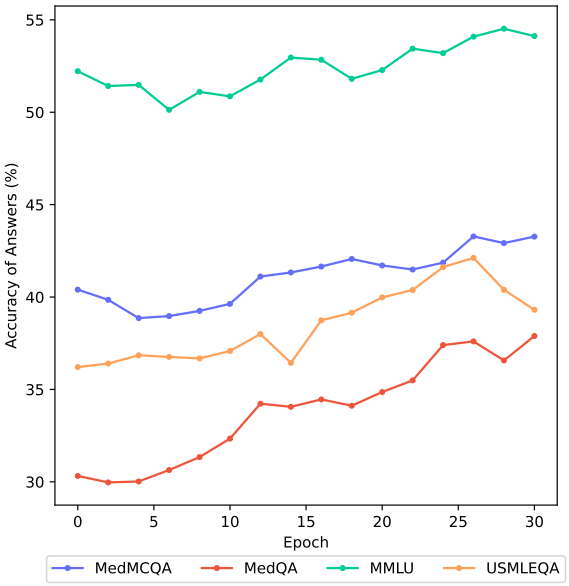

Figure 6: The performance of the reranker-based pipeline with respect to the number of training epochs.