CoSelector: Multi-Agent Collaboration for Multilingual Retrieval-Augmented Generation

Long Yuan[1,2], **Rui Qi**[1,2], **Xue Zhang**[1,2], **Jinan Xu**[1,2], **Kaiyu Huang**[1,2*]
[1]Key Laboratory of Big Data & Artificial Intelligence in Transportation
(Beijing Jiaotong University), Ministry of Education
[2]School of Computer Science and Technology, Beijing Jiaotong University
{24120408, kyhuang}@bjtu.edu.cn

Abstract

General Retrieval-Augmented Generation (RAG) systems would suffer from performance degradation in multilingual scenarios due to the complexity of integrating different retrieval results from multilingual collections. In this paper, we propose CoSelector, a multi-agent collaborative framework for multilingual retrieval selection, which is optimized by the centralized training with decentralized execution. Under this paradigm, agents share the same goal but execute decisions using only local observations. CoSelector consists of a Selector agent and a Generator agent. The Selector dynamically filters and composes the most pertinent multilingual documents into a coherent context, while the Generator processes this context to understand and integrate information for answer generation. Across the 15 languages evaluated on MKQA and XOR-TyDi QA, CoSelector achieves competitive improvement in response language consistency, demonstrating its effectiveness in both linguistic alignment and the selection of retrieved results.

1 Introduction

Retrieval-Augmented Generation (RAG) systems (Lewis et al., 2020; Mo et al., 2026) aim to integrate external evidence into the content generation process, which effectively overcomes the static nature of large language models' (LLMs) parameter-based knowledge. Despite its success, the prevailing RAG paradigm remains predominantly English-centric (Wang et al., 2026), failing to account for the inherently multilingual landscape of global information ecosystems (Chirkova et al., 2024a). In a multilingual scenario, a given query in the source language might not necessarily have supporting evidence from the native-language collections, but might exist in other languages, so-called multilingual RAG (Park and Lee, 2025; Qi et al., 2026b).

Method	Recall@5	Accuracy
Qwen2.5-3B-Instruct	/	14.17
w/. Retriever	81.42	30.40
w/. Retriever&Reranker	87.43	30.12
w/. Multi-Agent	86.27	39.13

Table 1: Performance of retrieval (Recall@5) and generation (Accuracy) using different RAG strategies on the NQ benchmark (Kwiatkowski et al., 2019). Retriever is bge-m3 and reranker is bge-m3-reranker.

However, RAG methods for retrieving evidence from multilingual collections encounter two critical challenges. First, multilingual retrievers do not perform well on low-resource languages due to biases in the pre-training data resources, leading to a significant amount of noise in the retrieved results (Huang et al., 2025; Qin et al., 2024). Second, cultural differences and variations in expression across languages make it difficult to integrate retrieved evidence from diverse sources into stable and consistent knowledge, which prevents the generator from reliably leveraging such multilingual information and thus results in degrading performance (Chirkova et al., 2024a). Overall, the challenge of multilingual RAG lies in how to effectively fuse the useful multi-source information by reducing the negative impact of the noise.

Existing RAG approaches rank candidate evidences according to relevance score before incorporating them into the context, such as using a re-ranker model to remove redundant ones, and thereby mitigate noise in multilingual retrieval results (Zhang et al., 2025b; Guo et al., 2025b). However, these methods operate at the query level in a pipeline, e.g., with the retriever, re-ranker, and generator functioning independently, leaving relevant evidence detached from the generator's feedback. As our preliminary test shown in Table 1, although

*Corresponding author.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1838–1856
November 6-10, 2026 ©2026 Association for Computational Linguistics

prior RAG pipelines can achieve high retrieval recall, they cannot guarantee that the accuracy of the generated answers can be improved simultaneously. Therefore, it is important to explore whether the retriever and generator can work collaboratively to enable action-level control, thereby facilitating more precise knowledge integration.

Inspired by multi-agent RAG (Chen et al., 2025), we formulate multilingual RAG as a multi-agent collaboration task, which is composed of a Selector and a Generator as a decision-making problem. The Selector is responsible for filtering out useless retrieved evidence and optimizing for collaboration with the Generator to maximize shared rewards (e.g., answer accuracy). Reinforcement learning (RL) (Sutton et al., 2018; Haarnoja et al., 2018; Silver et al., 2017) is a common practice for this setting, as it can optimize agent behaviors through trial-and-error interactions with an environment to maximize cumulative reward. However, multi-agent RL is notoriously challenging due to non-stationarity and credit assignment issues (Cheruiyot et al., 2025; Zhang et al., 2025a). When turning into the multilingual RAG setting, multi-agent RL suffers from inconsistent agent objectives, noisy cross-lingual feedback, and heterogeneous linguistic knowledge that complicates inter-agent collaboration.

To this end, we propose CoSelector, a multi-agent collaboration framework that adopts a centralized critic and decentralized Proximal Policy Optimization (PPO) actors. Specifically, the centralized critic provides a global value function during training, guiding the decentralized Selector and Generator policies toward cooperative strategies that maximize the shared utility, thereby directly addressing the credit assignment challenge. Simultaneously, the PPO component ensures training stability through its clipping mechanism, mitigating the non-stationarity issue.

Our CoSelector framework employs the Centralized Training with Decentralized Execution (CTDE) paradigm (Amato, 2024). In the observable setting, the Selector and Generator agents hold disjoint local observations: the former accesses the query and raw top-K multilingual retrieval results, while the latter only observes the query and filtered results screened by the Selector. Under the CTDE paradigm, the centralized critic leverages global information to stabilize joint training, whereas each agent acts as an independent PPO actor to make decisions purely based on its own local observation during inference.

Experiments conducted on comprehensive multilingual benchmarks covering 15 distinct languages demonstrate that our CoSelector can refine the initial multilingual retrieval results. Specifically, the Selector filters out redundant and irrelevant information, ensuring that only the most pertinent cross-lingual knowledge is retained to construct a concise and useful context for the Generator.

To sum up, our contributions are as follows:

- We introduce CoSelector that employs an document-level knowledge selection mechanism based on RL. It effectively filters redundant multilingual knowledge and alleviates knowledge conflicts across languages.
- The Selector agent breaks down each query into a sequence of fine-grained selection actions, dynamically determining which multilingual knowledge to incorporate into the context for each language and document, replacing the role of a reranker.
- Experimental results show that CoSelector achieves competitive performance in both accuracy for answering, while also exhibiting strong response language consistency and selection precision.

2 Related Work

Multilingual RAG. Multilingual RAG primarily faces the challenges of knowledge concentration in high-resource languages and conflicts among multilingual knowledge (Liu et al., 2025; Li et al., 2025a; Thakur et al., 2025; Qi et al., 2026a). In the literature, Chirkova et al. (2024a) built the first systematic baseline dubbed Bergen, which explicitly instructs LLMs to generate answers consistent with the user's language via prompt engineering and designs a 3-Character Recall (C3Recall) metric that is more suitable for evaluating answer correctness in multilingual scenarios. Then, Ranaldi et al. (2025a) proposed CrossRAG, which translates documents into English to reduce LLM processing, showing particular benefits for low-resource languages. However, existing multilingual RAG methods mostly rely on static prompt design or translation-based pipeline paradigms (Ranaldi et al., 2025a). They either directly concatenate retrieved multilingual documents as raw context (Li et al., 2025a; Chirkova et al., 2024a) or adopt universal translation strategies (Ranaldi et al., 2025a),

without conducting document-level adaptive selection.

Information Filtering in RAG. Representative RAG filtering paradigms in monolingual settings mainly include pipeline-based solutions such as naive RAG equipped with standalone rerankers (Glass et al., 2022), the cascading filtering pipeline DeepSieve (Guo et al., 2025c), and the metric-driven InfoGain-RAG (Wang et al., 2025). These mainstream monolingual methods follow a sequential pipeline, where the retriever, reranker, and generator operate independently in a decoupled manner (Karpukhin et al., 2020a; Lewis et al., 2020; Chuang et al., 2023). Different from existing decoupled pipeline schemes, our CoSelector establishes a collaborative Selector–Generator agent paradigm, whose agents are jointly optimized by sharing the global reward derived from the final answer.

Agentic RAG. Training-based RAG methods leverage RL for end-to-end optimization. Chen et al. (2025) formulate the RAG pipeline as a multi-agent system optimized by multi-agent reinforcement learning (MARL), while Jin et al. (2025) introduce the Search-R1 framework for multi-round retrieval-reasoning. Despite their success, applying RL to optimize the intricate document-level cross-lingual decision-making required for effective multilingual RAG remains unexplored.

3 Methods

3.1 Problem Formulation as Cooperative Multi-Agent Task

In our scenario, we formulate the multilingual RAG pipeline as a multi-agent collaboration problem. Specifically, we introduce a Selector between the retriever and generator in terms of Naive RAG, with system prompts to differentiate their roles and functionalities. The Selector and Generator share the same parameters and global reward, enabling efficient joint optimization. The Selector is responsible for secondary filtering and evidences selection on the full set of multilingual retrieval documents, and its output serves as the input to the Generator.

Formally, we define a tuple $\langle \mathcal{G}, \mathcal{O}, \mathcal{A}, \mathcal{R} \rangle$ where: $\mathcal{G}$ is the agent set; $\mathcal{O}$ collects observations available to each agent; $\mathcal{A}$ specifies the permissible actions; and $\mathcal{R}$ denotes the shared reward. All agents work collectively to maximize this global reward, which directly corresponds to the final answer accuracy.

3.2 Overall of CoSelector

Figure 1 presents the framework of CoSelector, comprising two learnable agents:

- **Selector** ($M^{(s)}$): Takes as input a query $\mathbf{q}$ and a multilingual retrieved documents set $\mathcal{C}_{\text{ret}}$ from a frozen external retriever. It first groups retrieved documents by language, then performs sliding-window based inclusion decisions for each language group independently to produce a filtered subset $\mathcal{C}_{\text{sel}} \subseteq \mathcal{C}_{\text{ret}}$.
- **Generator** ($M^{(g)}$): Consumes the original query $\mathbf{q}$ together with the filtered context $\mathcal{C}_{\text{sel}}$ from the Selector, and generates the final predicted answer $\mathbf{a}_{\text{pred}}$.

Following the classical paradigm of cooperative multi-agent reinforcement learning (MARL) (Hong et al., 2024; Guo et al., 2024), we adopt three core modules: shared reward formulation, parameter-sharing architecture, and role differentiation via distinct system prompts.

3.3 Agent Specifications

In this section, we detail the observation space, action space, and reward function for each agent $i \in \mathcal{G} = \{\text{Selector (S)}, \text{Generator (G)}\}$.

3.3.1 Selector Agent

Observation. The Selector's observation (case in Appendix E.1) includes its system prompt, the input query, and the complete retrieved document set:

$$O_S = \{\text{prompt}_S,\ \mathbf{q},\ \mathcal{C}_{\text{ret}}\} \tag{1}$$

Action space. The Selector outputs a binary decision per document, restricted to two tokens:

$$A_S = \{\text{“yes"},\ \text{“no"}\} \tag{2}$$

To guarantee valid outputs, we apply a `LogitsProcessor`[1] at each decoding step. Logits for all tokens except “yes” and “no” are masked to $-\infty$, forcing the model to select one of the two valid options.

Reward. The reward of Selector is defined as whether its output equals the global correctness score as:

$$R_S = \mathcal{R}_{\text{global}} = \text{C3Recall}(\mathbf{a}_{\text{pred}}, \mathbf{a}_{\text{gt}}) \tag{3}$$

[1] https://huggingface.co/docs/transformers.js/api/generation/logits_process

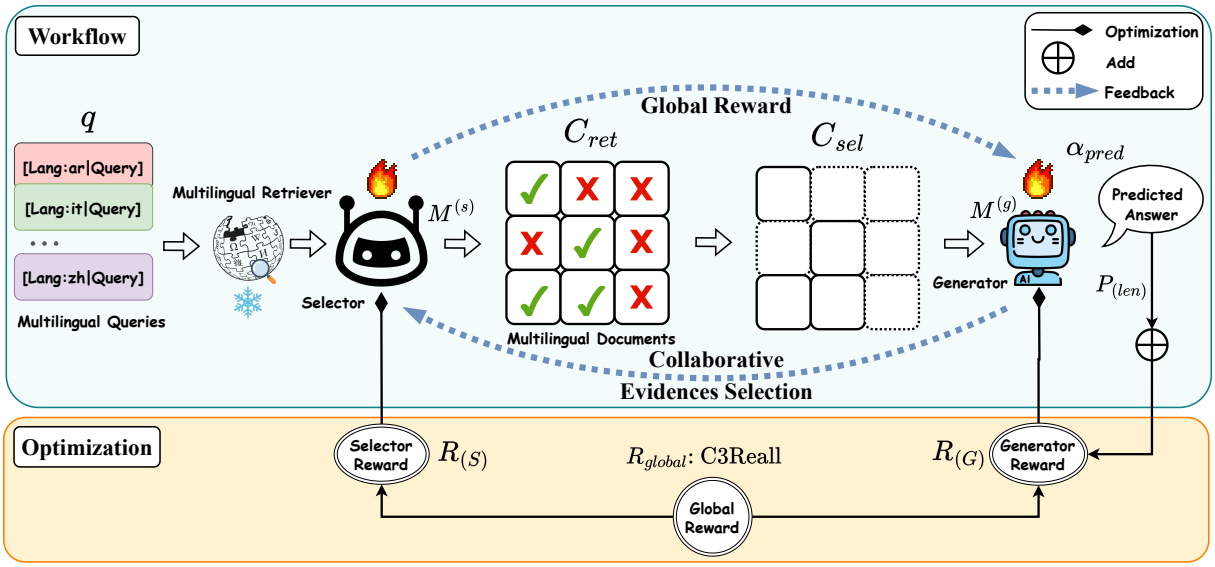


Figure 1: Architecture of CoSelector. The Selector filters multilingual retrieval results $\mathcal{C}_{\text{ret}}$ to produce a condensed context $\mathcal{C}_{\text{sel}}$. The Generator then produces the final answer (arrows indicate information flow from retriever to Selector and Selector to Generator). Both agents share the same reward signal based on answer correctness.

3.3.2 Generator Agent

Observation. The Generator observes its prompt, the query, and the filtered documents from the Selector (case in Appendix E.2):

$$O_G = \{\text{prompt}_G,\ \mathbf{q},\ \mathcal{C}_{\text{sel}}\} \tag{4}$$

Action space. The Generator produces free-form answers token by token from the LLM's vocabulary $\mathcal{V}$:

$$A_G = \mathcal{V} \tag{5}$$

Reward. The Generator receives a combination of the global correctness reward and a length penalty term as:

$$R_G = \mathcal{R}_{\text{global}} + \mathcal{P}_{\text{len}} \tag{6}$$

where $\mathcal{R}_{\text{global}} = \text{C3Recall}(\mathbf{a}_{\text{pred}}, \mathbf{a}_{\text{gt}})$ is the shared correctness reward. The penalty $\mathcal{P}_{\text{len}}$ is defined as:

$$\mathcal{P}_{\text{len}} = \begin{cases} -0.5, & \text{if } |\mathbf{a}_{\text{pred}}| > 40 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

This penalty serves two purposes: (1) preventing the joint optimization from collapsing into generating gibberish or excessively long outputs; (2) balancing the mixed distribution of answer lengths in the test set, where short answers (3–5 tokens) and long answers (exceeding 40 tokens) co-exist. Without this penalty, the model tends to favor longer generations that may sacrifice precision on short-answer questions.

3.4 Training Procedure

Stage I: Supervised Fine-Tuning for Cold-Start Before joint RL training, we cold-start both agents via supervised fine-tuning (SFT) on a task-specific cold-start dataset. To ensure balanced learning, this dataset is constructed by uniformly sampling instances across answer types and languages, covering diverse linguistic and answer patterns. For the Selector, we generate "yes"/"no" labels using an *answer-in-document* heuristic (see Appendix B). For the Generator, we train on ground-truth answers. This initialization reduces the exploration space and accelerates subsequent MARL convergence.

Stage II: Joint Multi-Agent Optimization with MAPPO After SFT, we jointly optimize both agents using MAPPO (Yu et al., 2022), a centralized training and decentralized execution (CTDE) algorithm tailored for cooperative multi-agent tasks. The main differences of the MAPPO algorithm compared to the single-agent PPO (Schulman et al., 2017) are threefold: (1) Both agents share a unified global reward based on final answer accuracy; (2) Trajectories from the Selector and Generator are collected simultaneously for joint policy updates; (3) A parameter-sharing backbone network is updated using experiences from both agents, distinguishing agent roles via system prompts.

Specifically, let $\tau_S = (O_S, a_S, R_S)$ and $\tau_G = (O_G, a_G, R_G)$ denote the trajectories (observation,

action, reward) of the Selector and Generator, respectively, both agents receive the identical global reward $\mathcal{R}_{\text{global}} = \text{C3Recall}(\mathbf{a}_{\text{pred}}, \mathbf{a}_{\text{gt}})$ at the terminal step T. Then, during training, the combined experience tuple (τ_S, τ_G) from both agents is stored in a shared replay buffer. We update the shared policy network π_θ using the MAPPO objective, which averages the clipped PPO loss across all agents as

$$\mathcal{L}_{\text{MAPPO}}(\theta) = \mathbb{E}_{\tau_S,\tau_G}\left[\frac{1}{2}\sum_{i\in\{\text{S,G}\}}\min\big(r^i\hat{A}^i, \text{clip}(r^i, 1-\epsilon, 1+\epsilon)\hat{A}^i\big)\right] \tag{8}$$

where $r^i = \pi_\theta(a^i|O^i)/\pi_{\theta_{\text{old}}}(a^i|O^i)$ denotes the importance sampling ratio. We adopt generalized advantage estimation to compute individual advantage values for each agent. The calculation form is consistent with standard single-agent PPO, with only separate estimation conducted for each agent.

Crucially, in the backward pass, trajectories and rewards from both agents are fed together to update the shared network. This joint update mechanism enables the Selector to optimize document filtering for Generator success, fully aligning both agents toward the shared objective.

4 Experiments

4.1 Experimental Settings

Models. We employ mE5 (Wang et al., 2024) as the retriever and bge-reranker-v2-m3 (Chen et al., 2024a) as the reranker. For the generation phase, we instantiate CoSelector with three LLMs: Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct (Team, 2024) and Llama3-8B-Instruct (AI@Meta, 2024) to evaluate among different backbone models.

Datasets. We evaluate CoSelector on the multilingual open-domain QA benchmarks (MKQA (Longpre et al., 2020)). For MKQA, we follow (Chirkova et al., 2024a) to construct training and evaluation sets with 13 languages. We also conduct an additional experiment on XOR-TyDi QA (Asai et al., 2021a) in Appendix C.1. Overall, the experiments cover 15 typologically diverse languages with varying resource levels (details in Appendix A.1).

Evaluation Protocols. We adopt C3Recall (Chirkova et al., 2024b) that is used to measure the answer accuracy of multilingual RAG systems, and four other fine-grained metrics to evaluate the practical contribution of the Selector (Details are provided in Appendix A.2).

Implementation Details. The retrieval corpus is constructed from a multilingual Wikipedia dump[2], segmented into contiguous 100-word docs following standard practice (Karpukhin et al., 2020b; Asai et al., 2021b). Furthermore, we adopt a consistent retrieval strategy for the top-K setting. We initially retrieve the top-3 docs for each language and further select appropriate candidates to form the final top-10 docs according to relevance scores. With the retrieved documents, CoSelector is optimized in a two-stage process: (1) SFT for cold-starting the policy, and (2) RL via MAPPO for refining the collaborative agent policies. The hyperparameter settings for both stages are listed in Appendix A.2.

Baseline Methods. The compared baselines: (1) **Direct** denotes direct query inference without retrieved contextual information. (2) **Naive RAG** and **Naive RAG w/. SFT** introduce retrieved context and additional supervised fine-tuning upon the vanilla LLM respectively. **Naive RAG w/. reranker** denotes that a reranker is additionally introduced into the pipeline. (3) **CrossRAG** (Ranaldi et al., 2025b) is a translation-based multilingual RAG pipeline that unifies all contextual documents into English. (4) **Bergen** (Chirkova et al., 2024a) retrieves and concatenates corpora from user query language and English sources. (5) **Search-o1** (Li et al., 2025b) integrates agentic search modules into reasoning workflows, while (6) **Search-R1** (Jin et al., 2025) is a reinforcement learning paradigm facilitating interleaved reasoning and searching. For fair comparison, we restrict the retrieval scope of both Search-o1 and Search-R1 to the multilingual Wikipedia dump in all experiments.

4.2 Main Results

Overall. As shown in Table 2, we conduct comprehensive comparisons between CoSelector and various baseline methods across 13 languages covered in MKQA. The results demonstrate that CoSelector achieves steady and remarkable performance gains across different model sizes and families. In general, benefiting from the introduction of external multilingual information, RAG-based methods can synthesize cross-lingual evidences to accurately answer language-specific and time-sensitive querys, achieving suboptimal performance. Restricted by weak multilingual instruction following ability of LLMs and inherent noise in retrieved documents, RAG-based methods fail to achieve fur-

[2] https://huggingface.co/datasets/wikimedia/wikipedia/tree/main

Methods	Query Language													
	ar	**de**	**es**	**en**	**fi**	**fr**	**it**	**ja**	**ru**	**th**	**zh**	**pt**	**ko**	**Avg.**
Qwen2.5-3B-Instruct														
Direct	12.13	26.63	29.67	38.59	17.03	27.30	25.91	10.61	17.29	13.02	12.00	28.86	7.35	20.49
Naive RAG	15.50	42.87	**53.43**	44.42	38.64	44.11	43.64	17.90	13.16	42.14	23.42	16.65	18.06	31.84
Naive RAG w/. SFT	27.39	41.69	40.90	47.82	34.55	41.94	40.44	30.19	31.66	24.43	29.43	41.40	23.27	35.01
Naive RAG w/. reranker	20.70	39.33	42.81	**51.61**	35.56	42.32	41.36	20.44	30.39	18.09	19.00	41.88	15.45	32.23
CrossRAG	15.94	42.90	54.16	46.93	38.91	44.68	44.23	16.94	15.21	**42.16**	23.48	17.62	18.39	32.43
Bergen	25.31	43.44	45.20	46.76	39.40	41.28	**46.21**	24.74	30.88	22.68	24.61	**47.33**	18.11	35.07
Search-o1	15.50	27.72	28.39	30.81	26.65	27.01	29.06	18.91	25.18	17.66	17.46	29.89	15.66	21.32
Search-R1	19.40	38.15	43.80	51.01	23.85	41.29	40.10	15.11	29.65	16.59	14.52	39.36	10.60	29.36
CoSelector (Ours)	**29.97**	**45.86**	45.36	50.49	**41.19**	45.93	**45.61**	**34.90**	**36.95**	27.53	**31.40**	46.34	**27.15**	**39.13**
Qwen2.5-7B-Instruct														
Direct	17.48	34.75	38.26	46.20	23.79	36.99	36.77	16.71	23.98	17.84	17.19	38.61	11.35	27.69
Naive RAG	24.15	46.53	48.68	57.12	42.99	47.44	47.19	26.29	32.80	22.10	23.39	46.92	18.98	37.27
Naive RAG w/. SFT	31.55	45.11	44.19	52.62	40.68	46.42	46.08	35.71	36.63	**29.42**	33.79	45.63	30.01	39.83
Naive RAG w/. reranker	23.22	46.23	48.11	56.67	43.64	46.78	47.11	25.28	32.80	21.85	23.32	47.10	18.49	36.90
CrossRAG	24.32	47.01	48.73	**59.12**	41.68	46.96	47.25	27.23	33.16	21.52	24.47	46.13	19.96	37.50
Bergen	25.13	48.68	50.60	52.79	45.58	50.57	50.63	27.50	35.24	22.74	21.70	51.04	17.86	38.47
Search-o1	21.62	35.39	34.05	40.65	27.49	33.50	34.67	21.79	24.69	21.25	20.69	33.49	17.38	28.20
Search-R1	21.98	43.48	46.47	54.10	31.16	44.72	44.98	20.78	31.41	20.45	17.13	44.27	14.13	33.46
CoSelector (Ours)	**38.36**	**53.82**	**53.17**	57.15	**48.00**	**53.68**	**51.55**	**40.90**	**42.94**	28.41	**37.12**	**53.48**	**32.78**	**45.49**
Qwen2.5-7B-Instruct (Retriever: BGE-M3)														
Direct	13.44	29.20	32.49	40.47	18.06	28.89	29.88	13.32	19.57	14.26	14.69	30.66	09.18	22.70
Naive RAG	25.06	43.12	45.96	50.26	39.56	43.84	49.53	21.93	33.93	21.31	20.11	45.59	18.64	35.02
Naive RAG w/. SFT	30.81	42.69	41.88	47.31	38.38	43.73	43.93	32.60	35.43	28.66	32.52	42.83	27.32	37.55
Naive RAG w/. reranker	26.58	46.84	48.88	57.74	43.85	47.86	47.42	25.67	36.34	24.08	22.19	47.81	20.71	38.15
CrossRAG	25.91	43.97	46.81	51.11	40.41	44.69	50.38	22.78	34.78	22.16	20.96	46.44	19.49	35.87
Bergen	24.02	38.96	41.08	40.57	40.11	39.88	45.24	21.02	34.26	20.41	17.33	44.58	19.19	32.82
Search-o1	14.57	26.05	28.64	35.96	21.03	28.44	26.62	16.33	18.94	14.05	15.77	28.67	10.55	21.97
Search-R1	11.91	18.14	21.87	31.81	11.75	20.85	21.31	14.16	11.27	8.79	12.64	20.67	8.43	16.43
CoSelector (Ours)	**36.76**	**48.63**	**49.33**	**53.16**	**44.95**	**49.61**	**57.61**	**37.68**	**40.18**	**33.74**	**35.84**	**49.30**	**31.81**	**42.97**
Llama3-8B-Instruct														
Direct	11.71	44.34	49.18	56.25	36.90	45.72	44.56	11.99	16.44	16.42	11.34	43.38	10.63	30.68
Naive RAG	13.85	50.45	58.03	51.13	47.79	50.13	49.53	14.99	13.53	49.38	18.54	16.75	14.17	34.48
Naive RAG w/. SFT	34.64	48.14	47.31	52.88	47.37	48.12	48.68	39.06	**40.04**	33.42	32.48	48.61	32.35	42.55
Naive RAG w/. reranker	16.88	50.03	52.29	**61.41**	46.16	50.54	50.28	16.07	23.76	18.41	14.33	48.70	15.75	35.74
CrossRAG	13.12	48.14	56.03	50.23	44.85	49.23	46.19	12.68	12.51	48.69	18.39	12.86	13.94	32.84
Bergen	25.13	48.68	50.60	52.79	45.58	50.57	50.63	27.50	35.24	22.74	21.70	**51.04**	17.86	34.50
Search-o1	20.98	32.73	34.71	40.03	29.16	32.97	33.59	22.70	29.39	17.89	15.71	34.17	16.11	27.70
Search-R1	13.10	33.60	47.90	37.50	25.60	36.40	34.20	19.50	9.30	34.30	23.60	15.90	10.20	26.20
CoSelector (Ours)	**40.05**	**58.29**	**66.55**	58.78	**55.91**	**58.79**	**61.11**	**46.76**	38.57	**61.54**	**49.27**	34.59	**36.04**	**51.25**

Table 2: Main results on MKQA across 13 languages with C3Recall (%), where the best performance are **bold**.

ther performance improvement. Specifically, the performance degradation of the RL-based method (Search-R1) stems from the confusion introduced when retrieving from the full multilingual corpus. Existing methods retrieve from large and linguistically diverse datasets, which introduces additional noise and severe cross-lingual knowledge conflicts, thereby amplifying errors and degrading performance, as illustrated in the case study in Appendix D. CoSelector adopts global reward as implicit feedback and unified collaborative optimization objective, enabling mutual information exchange between the Selector and Generator agents to form a closed feedback loop, and ultimately boosting answer accuracy.

Effect of Reranker. Notably, reranker-based Naive RAG reranks retrieved results and picks other top-K documents with the highest scores from its perspective. However, it only returns documents matching the inherent preference and lies in the isolation between retrieval and generation, as the retriever cannot perceive feedback signals from the generator. Table 1 reveals that strong retrieval performance does not equate to high answer accuracy. Some high-scoring documents thus hinder the generator from comprehending and extracting valid evidence. However, CoSelector adopts collaborative MARL, taking final answer correctness as the global reward signal for agent cooperation. Besides, as shown in Figure 3, the quality of generated answers guides

Settings	Query Language													
	ar	de	es	en	fi	fr	it	ja	ru	th	zh	pt	ko	Avg.
Full Model	**38.36**	**53.82**	**53.17**	**57.15**	**48.00**	**53.68**	**51.55**	**40.90**	**42.94**	28.41	37.12	**53.48**	**32.78**	**45.49**
BP w/o. Generator	33.91	49.98	50.18	54.13	44.82	49.40	48.55	38.27	40.58	26.89	34.93	49.85	29.41	42.38
BP w/o. Selector	36.56	51.41	50.12	54.27	45.38	50.93	47.61	40.03	40.47	27.76	36.32	49.78	29.91	43.12
w/o. Selector, w/o. RL	33.05	47.21	47.13	51.22	42.37	47.05	46.18	35.16	37.42	**29.83**	32.97	46.32	27.14	40.42
w/o. Selector	37.33	51.60	50.88	52.98	46.19	52.10	47.54	39.71	41.87	29.12	**38.07**	50.92	31.41	43.82
w/o. RL	30.14	45.52	44.97	49.56	40.78	44.97	45.76	36.03	38.11	29.05	32.12	44.98	27.71	39.21

Table 3: Ablation studies on MKQA (C3Recall %). "BP w/o." indicates whether each agent participates in the backpropagation of the joint optimization. "w/o. RL" denotes the training paradigm that follows SFT.

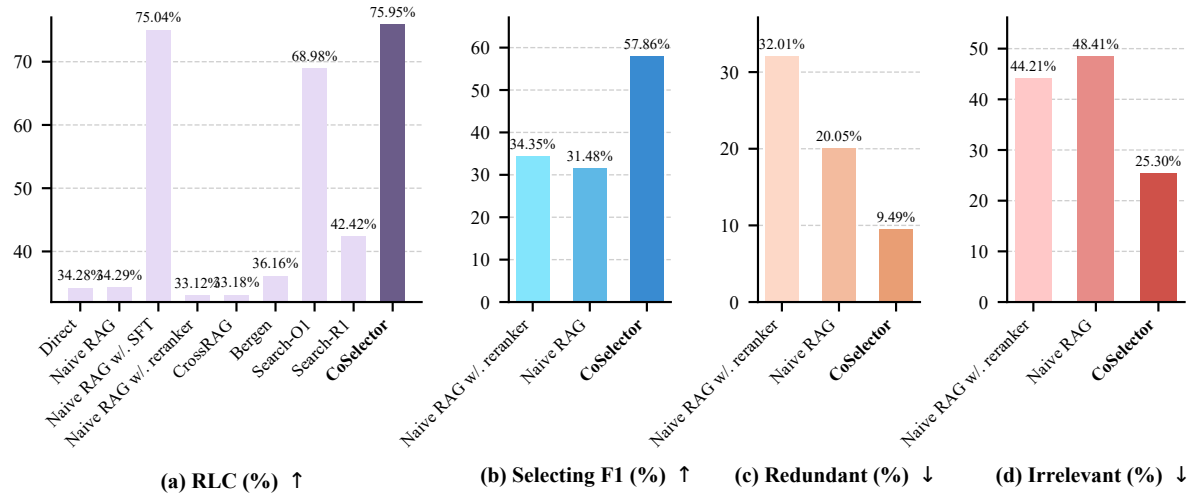


Figure 2: Multi-dimensional analysis of CoSelector on MKQA. RLC = Response Language Consistency. Selecting F1 = document selection accuracy (*answer-in-document* heuristic). Redundancy Ratio and Irrelevance Ratio respectively quantify the proportion of redundant and irrelevant information within retrieved context, respectively. All metrics are averaged values.

the Selector agent to select evidence-accessible documents instead of merely top-scoring ones, eliminating inherent language selection bias existing in retrievers and rerankers.

4.3 Ablation Studies

Joint Optimization. As shown in Table 3, we exclude the trajectory of the Selector agent or the Generator agent in PPO backpropagation, respectively. The results show that multi-agents collaboration is effective with joint optimization. When two agents are jointly optimized through shared reward, the selector learns to select the optimal combination of multilingual evidences under the guidance of a shared outcome-based reward, and the generator further enhances its denoising and multilingual knowledge integration capabilities on this basis. Optimizing only the selector allows it to retain higher-quality multilingual evidences, but the generator's evidence integration capability relies solely on the SFT, resulting in performance degradation sharply. On the other hand, optimizing only the generator leverages its partial denoising ability, achieving only a moderately improved performance, yet it still falls short of our method.

Training Procedures. To valid that the performance gains of CoSelector originate from the introduction of the Selector rather than vanilla SFT or RL training, we exclude the Selector from the feedback loop, so that the Generator directly takes the top-10 retrieved documents as input. The comparison between "Full Model" and "w/o. Selector + RL" shows an average performance improvement of 1.67%, which verifies that the Selector agent effectively enhances the overall quality of multilingual evidence. We only observe slight performance declines on Thai and Chinese, while obvious gains are obtained on the other eleven languages, with a notable rise of 4.17% on English. This reflects that the evidence selection ability of the Selector agent varies across different languages. We further ex-

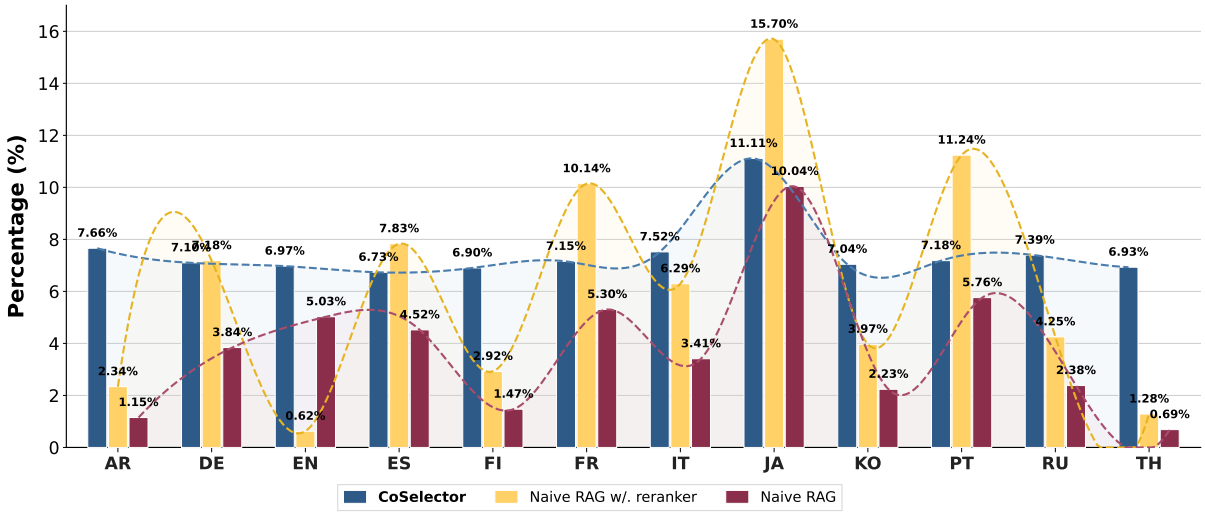


Figure 3: Distribution of language of selected evidence with different approaches in the MKQA dataset.

plore the performance advantages of RL over SFT. Experimental results demonstrate that RL brings an average improvement of over 3.4% compared with SFT. It indicates that as trainable optimization paradigms, RL possesses stronger advantages in multilingual knowledge fusion and the internalization of multilingual RAG capabilities.

5 Contributions of Selector

Overall. We analyze CoSelector through the lens of selection quality, language bias, resource-tier performance, and inference efficiency, demonstrating its effectiveness as an efficient and equitable pre-selector for multilingual RAG.

Overall Performance. As shown in Figure 2, CoSelector achieves better performance in terms of Selecting F1 and answer accuracy. This indicates that our method can not only locate valid answer information more precisely, but also reliably distinguish high-quality multilingual documents for generation, instead of leaving all discrimination difficulties to the generator.

CoSelector attains an average context length of 3770 characters, comparable to 3589 chars of Naive RAG, yet markedly shorter than Naive RAG with reranking, 4878 chars. It delivers prominent denoising effects under equivalent input scale. The redundancy ratio falls from 20.05% to 9.49%, and irrelevant content ratio drops from 48.41% to 25.30%. Acting as an effective pre-filter, it removes extensive noise, enables models to concentrate on key valid information, and boosts reasoning efficiency and prediction robustness. As a core metric for real-world multilingual RAG, Response Language Consistency (RLC) hits the peak value of 75.95% with CoSelector. This outperforms the 75.04% of the second-best Naive RAG with SFT and greatly exceeds Search-R1's 42.42%. The method enhances cross-lingual alignment, improving RLC and practical usability of multilingual RAG systems.

Documents Language Distribution Analysis. As shown in Figure 3, we analyze the document language distribution selected by different methods to explore the internal causes of performance improvements. CoSelector actively adjusts the severe language selection bias introduced by mE5 and bge-m3-v2-reranker, striving to keep the proportion of selected documents for each language roughly balanced. In contrast, baseline methods show significant linguistic preference: Naive RAG with mE5 retriever is overwhelmingly dominated by Chinese (zh), accounting for 54.19%, while the reranker-based Naive RAG with bge-m3-v2-reranker exhibits a strong inclination towards Chinese (26.23%) and Japanese (15.70%), leaving low-resource languages including Arabic, Finnish and Thai severely underrepresented. Notably, under the adjustment of CoSelector, the proportions of Chinese and Japanese are still slightly higher than other languages, which is attributed to the secondary filtering mechanism of the Selector agent and the inherent preference of the original retrieval results for Chinese and Japanese documents. By leveraging the balanced retrieval strategy (see in Appendix C.2) and secondary selection mechanism, CoSelector effectively mitigates the distribution bias, achieving fair and stable coverage for both high-resource and low-resource languages, and lay-

Tier	Data	#L	Direct	Naive	Ours
High	MKQA	10	32.22	40.82	**46.65**
Mid	MKQA	3	19.49	30.18	**35.69**
High	XOR	3	22.04	29.31	**31.70**
Mid	XOR	3	27.65	**39.59**	37.50
Low	XOR	1	17.55	**38.53**	29.79

Table 4: C3Recall (%) by resource tier on MKQA / XOR-TyDi (Qwen2.5-7B-Instruct). #L: #languages.

Method	Lat. (ms)	Calls	In	Out
Naive RAG	5940	1	3589	110
Naive RAG w/ Reranker	8210	1	4578	121
Search-o1	11650	1.73	615	1743
Search-R1	12800	2.44	919	2758
CoSelector	7950	5	3770	10

Table 5: Inference cost on MKQA (Qwen2.5-7B-Instruct). In/Out: tokens per query.

ing a solid foundation for multilingual RAG systems.

Performance Across Language Resource Tiers. Table 4 reports the C3Recall performance grouped by language resource tiers. We observe consistent trends across two datasets: For low-resource languages, the absolute performance gain brought by RAG is the largest. Direct LLM inference suffers severe performance degradation on low-resource languages, and Naive RAG yields a large boost, +20.98% on XOR-TyDi low-resource setting. For mid-resource languages, RAG also brings substantial improvements. The core bottleneck lies in the generation capability of the LLM itself. For high-resource languages, the gain of vanilla RAG is relatively limited, where the main bottleneck shifts to retrieval quality rather than generation. CoSelector further improves over Naive RAG for most high and mid-resource cases by learning a collaborative selection policy with generator feedback. An exception appears in the XOR-TyDi mid-resource setting, where Naive RAG slightly outperforms CoSelector. We attribute this to language distribution shift in retrieved evidence.

Inference Efficiency. Table 5 compares inference overhead on MKQA with Qwen2.5-7B-Instruct. CoSelector only introduces a small latency overhead compared with Naive RAG, and is much faster than Search-o1 and Search-R1. Although CoSelector requires multiple model calls for the Selector agent, the total input token budget is kept at a moderate level. Notably, CoSelector produces far fewer output tokens, since the Selector filters noisy evidence before generation and the Generator only needs to predict concise answers. Compared with reranker-based and agentic RAG baselines, CoSelector achieves better answer quality without excessive inference cost.

6 Conclusions

We propose CoSelector, a collaborative multi-agent reinforcement learning framework that unifies optimization goals and iteratively refines results via feedback signals. Its tailored multilingual evidence selection mechanism alleviates cross-lingual knowledge conflicts. The selector agent dynamically filters and assembles relevant cross-lingual evidence to build high-quality context, reducing the processing load of large language models.

Acknowledgements

The research work descried in this paper has been supported by the National Nature Science Foundation of China (No. 62376019, 62476023, 61976015, 61976016, 61876198 and 61370130). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

Limitations

Optimization Methodology. The current optimization approach presents two avenues for improvement. First, the credit assignment and non-stationarity challenges inherent in our multi-agent RAG task necessitate highly stable RL training. While MAPPO was effective, alternative algorithms offering potentially greater robustness (e.g., GRPO (Guo et al., 2025a), which operates with a simpler, lower-variance objective) warrant investigation. Second, the use of LoRA (Hu et al., 2021) for RL, while efficient, imposes a low-rank constraint on policy updates. A full-parameter RL fine-tuning strategy could potentially circumvent this limitation, allowing the model to learn more complex and nuanced collaborative policies, thereby pushing the performance boundary of multilingual RAG further.

References

AI@Meta. 2024. Llama 3 model card.

Christopher Amato. 2024. An introduction to centralized training for decentralized execution in cooper-

ative multi-agent reinforcement learning. *Preprint*, arXiv:2409.03052.

Akari Asai, Jungo Kasai, Jonathan H. Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021a. Xor qa: Cross-lingual open-retrieval question answering. *Preprint*, arXiv:2010.11856.

Akari Asai, Xinyan Yu, Jungo Kasai, and Hannaneh Hajishirzi. 2021b. One question answering model for many languages with cross-lingual dense passage retrieval. *Preprint*, arXiv:2107.11976.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024b. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Yiqun Chen, Lingyong Yan, Weiwei Sun, Xinyu Ma, Yi Zhang, Shuaiqiang Wang, Dawei Yin, Yiming Yang, and Jiaxin Mao. 2025. Improving retrieval-augmented generation through multi-agent reinforcement learning. *arXiv preprint arXiv:2501.15228*.

Kemboi Cheruiyot, Nickson Kiprotich, Vyacheslav Kungurtsev, Kennedy Mugo, Vivian Mwirigi, and Marvin Ngesa. 2025. A survey of multi agent reinforcement learning: Federated learning and cooperative and noncooperative decentralized regimes. *Preprint*, arXiv:2507.06278.

Nadezhda Chirkova, David Rau, Hervé Déjean, Thibault Formal, Stéphane Clinchant, and Vassilina Nikoulina. 2024a. Retrieval-augmented generation in multilingual settings. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 177–188, Bangkok, Thailand. Association for Computational Linguistics.

Nadezhda Chirkova, David Rau, Hervé Déjean, Thibault Formal, Stéphane Clinchant, and Vassilina Nikoulina. 2024b. Retrieval-augmented generation in multilingual settings. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 177–188.

Yung-Sung Chuang, Wei Fang, Shang-Wen Li, Wen-tau Yih, and James Glass. 2023. Expand, rerank, and retrieve: Query reranking for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12131–12147.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Minghao Guo, Qingcheng Zeng, Xujiang Zhao, Yanchi Liu, Wenchao Yu, Mengnan Du, Haifeng Chen, and Wei Cheng. 2025b. Deepsieve: Information sieving via llm-as-a-knowledge-router. *Preprint*, arXiv:2507.22050.

Minghao Guo, Qingcheng Zeng, Xujiang Zhao, Yanchi Liu, Wenchao Yu, Mengnan Du, Haifeng Chen, and Wei Cheng. 2025c. Deepsieve: Information sieving via llm-as-a-knowledge-router. *arXiv preprint arXiv:2507.22050*.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Steven Yau, Zijuan Lin, Liyang Zhou, and 1 others. 2024. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, volume 2024, pages 23247–23275.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Kaiyu Huang, Fengran Mo, Xinyu Zhang, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, Jinan Xu, Jian-Yun Nie, and Yang Liu. 2025. A survey on large language models with multilingualism: Recent advances and new frontiers. *Preprint*, arXiv:2405.10936.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *Preprint*, arXiv:2503.09516.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020a. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 6769–6781.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and

Wen-tau Yih. 2020b. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Bryan Li, Fiona Luo, Samar Haider, Adwait Agashe, Siyu Li, Runqi Liu, Miranda Muqing Miao, Shriya Ramakrishnan, Yuan Yuan, and Chris Callison-Burch. 2025a. Multilingual retrieval augmented generation for culturally-sensitive tasks: A benchmark for cross-lingual robustness. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4215–4241.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making large language models a better foundation for dense retrieval. *Preprint*, arXiv:2312.15503.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025b. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5420–5438.

Wei Liu, Sony Trenous, Leonardo FR Ribeiro, Bill Byrne, and Felix Hieber. 2025. Xrag: Cross-lingual retrieval-augmented generation. *arXiv preprint arXiv:2505.10089*.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2020. Mkqa: A linguistically diverse benchmark for multilingual open domain question answering.

Fengran Mo, Zhan Su, Yuchen Hui, Jinghan Zhang, Jia Ao Sun, Zheyuan Liu, Chao Zhang, Tetsuya Sakai, and Jian-Yun Nie. 2026. Opendecoder: Open large language model decoding to incorporate document quality in rag. In *Proceedings of the ACM Web Conference 2026*, pages 2252–2262.

Jeonghyun Park and Hwanhee Lee. 2025. Investigating language preference of multilingual rag systems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5647–5675.

Rui Qi, Fengran Mo, Yufeng Chen, Xue Zhang, Shuo Wang, Hongliang Li, Jinan Xu, Meng Jiang, Jian-Yun Nie, and Kaiyu Huang. 2026a. Language-coupled reinforcement learning for multilingual retrieval-augmented generation. *arXiv preprint arXiv:2601.14896*.

Rui Qi, Fengran Mo, Sijin Lu, Yufeng Chen, Jian-Yun Nie, and Kaiyu Huang. 2026b. Crosearch-r1: Better leveraging cross-lingual knowledge for retrieval-augmented generation. *arXiv preprint arXiv:2604.25182*.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S. Yu. 2024. Multilingual large language model: A survey of resources, taxonomy and frontiers. *Preprint*, arXiv:2404.04925.

Leonardo Ranaldi, Barry Haddow, and Alexandra Birch. 2025a. Multilingual retrieval-augmented generation for knowledge-intensive task. *arXiv preprint arXiv:2504.03616*.

Leonardo Ranaldi, Federico Ranaldi, Fabio Massimo Zanzotto, Barry Haddow, and Alexandra Birch. 2025b. Improving multilingual retrieval-augmented language models through dialectic reasoning argumentations. *Preprint*, arXiv:2504.04771.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, and 1 others. 2017. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359.

Richard S Sutton, Andrew G Barto, and 1 others. 2018. Reinforcement learning: An introduction 2nd ed. *MIT press Cambridge*, 1(2):25.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Nandan Thakur, Suleman Kazi, Ge Luo, Jimmy Lin, and Amin Ahmad. 2025. Mirage-bench: Automatic multilingual benchmark arena for retrieval-augmented generation systems. 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 274–298.

Dan Wang, Guozhao Mo, Yafei Shi, Cheng Zhang, Bo Zheng, Boxi Cao, Xuanang Chen, Yaojie Lu, Hongyu Lin, Ben He, and 1 others. 2026. All languages matter: Understanding and mitigating language bias in multilingual rag. *arXiv preprint arXiv:2604.20199*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *Preprint*, arXiv:2402.05672.

Zihan Wang, Zihan Liang, Zhou Shao, Yufei Ma, Huangyu Dai, Ben Chen, Lingtao Mao, Chenyi Lei, Yuqing Ding, and Han Li. 2025. Infogain-rag: Boosting retrieval-augmented generation through document information gain-based reranking and filtering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7201–7215.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita-Velez, Yue Liao, Hongru Wang, and 6 others. 2025a. The landscape of agentic reinforcement learning for llms: A survey. *Preprint*, arXiv:2509.02547.

Haozhen Zhang, Tao Feng, and Jiaxuan You. 2025b. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning. *Preprint*, arXiv:2506.09033.

A Expermental Details

A.1 Dataset and Model Details

MKQA. The MKQA (Multilingual Knowledge Questions and Answers) dataset, introduced by Apple, is a critical benchmark for evaluating cross-lingual question answering models. It contains 10,000 question-answer pairs aligned across 26 languages. Derived from the English Google Natural Questions dataset, MKQA employs professional translation and alignment, and provides entity annotations alongside automated evaluation scripts.

XOR-TyDi QA. To evaluate the comprehensive capability of CoSelector in multilingual answer generation, we adopt the XOR-Full subtask. XOR-TyDi QA is a multilingual open-retrieval question answering dataset whose challenge lies in requiring systems to retrieve answers directly from a multilingual document collection and generate them in the target language. This subtask demands models to perform end-to-end cross-lingual retrieval and generation, providing a holistic assessment of their information integration and multilingual generation abilities.

Language List. Spanish (es), German (de), English (en), Finnish (fi), French (fr), Italian (it), Japanese (ja), Russian (ru), Thai (th), Chinese (zh), Portuguese (pt), Korean (ko), Russian (ru), and Telugu (te) are languages with distinct linguistic and cultural footprints across the globe.

Multilingual-e5-base. The Multilingual-E5-base (mE5) model (Wang et al., 2024) is a multilingual text embedding model based on XLM-RoBERTa, supporting 94 languages. It requires input texts to be prefixed with "query: " or "passage: " and produces L2-normalized embeddings suitable for direct similarity comparison. In this work, M-E5 serves as the retriever for all compared systems and our proposed CoSelector framework.

Qwen2.5-3B-Instruct and 7B. In the CoSelector framework, we instantiate the Selector and Generator agents with Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct, respectively, to assess the robustness of our multi-agent collaboration across different model scales. Both models belong to Alibaba's Qwen2.5 series and exhibit strong multilingual understanding and generation capabilities across 29 languages, with notable improvements in knowledge retention, code generation, and instruction following compared to earlier versions. Their robust multilingual competence and instruction-following attributes make them well-suited to handle the complex decision-making and answer-generation demands inherent in multilingual retrieval-augmented generation.

Qwen3-32B-Instruct (Team, 2025). We employ Qwen3-32B-Instruct to construct the parallel translation data for XOR-TyDi QA. The translation prompt template is provided in Figure 4.

bge-reranker-v2-m3 (Li et al., 2023; Chen et al., 2024b). We use bge-reranker-v2-m3 as the reranker for Naive RAG to perform a secondary reranking of the original multilingual retrieved results.

A.2 Implementation Details

Training Process. The optimization of CoSelector proceeds in two stages: a warm-start Supervised Fine-Tuning (SFT) stage and a subsequent Reinforcement Learning (RL) training stage. For SFT, the training data are first sampled in a balanced manner based on language and answer type. We then conduct separate SFT for the Selector and the Generator. The Generator is trained on the balanced data to map from the query and retrieved context to the answer. For the Selector, we additionally downsample examples in which all candidate documents are uniformly relevant (yes) or irrelevant (no) to prevent trivial selection policies. SFT is performed for 1 epoch.

Before RL training, we use the SFT-initialized model to perform rejection sampling on the RL training set. The resulting predictions are bucketed by their C3Recall scores (with a bin size of 0.1) to ensure a balanced distribution of sample difficulties. RL is carried out with the MAPPO for 300 steps, using a batch size of 64, a PPO clipping ratio of 0.2, a temperature of 0.9, a KL coefficient of 0.2, and whitened and normalized rewards. All SFT experiments run on 8× NVIDIA A800 (80GB) GPUs, and RL training is conducted on NVIDIA H20 (96GB) GPUs.

Dataset Partitioning. MKQA. We first remove samples with empty or null answers from the MKQA dataset, resulting in 6,584 valid samples. For each of the 13 selected languages, we sample 2,000 queries, forming a test set of 26,000 samples in total. The remaining 4,584 samples are further split into a warm-start SFT subset (485 samples) and an RL training set (4,099 samples). **XOR-TyDi QA.** Since the official test set of XOR-TyDi QA does not provide ground-truth answers, we use its validation set as the test set for evaluation.

When training Search-R1, we construct a bal-

Methods	Query Language							
	ar	**bn**	**fi**	**ja**	**ko**	**ru**	**te**	**Avg.**
Qwen2.5-3B-Instruct								
Direct	27.79	13.26	22.44	8.89	16.17	23.67	13.78	18.00
Naive RAG	44.78	26.32	32.72	20.22	20.25	**39.20**	**26.98**	31.98
Naive RAG w/. SFT	32.09	27.27	24.87	20.80	18.51	15.61	11.90	22.92
Naive RAG w/. reranker	**45.62**	24.18	33.14	21.30	19.84	38.62	24.97	29.67
CrossRAG	45.18	**26.77**	33.81	20.71	20.54	40.76	26.63	30.63
Search-o1	19.22	10.95	06.59	07.39	15.41	12.45	11.42	11.92
Search-R1	41.84	10.32	34.40	18.29	24.09	29.62	5.27	26.08
CoSelector (Ours)	43.51	22.86	**39.06**	**29.24**	**27.24**	35.83	24.87	**33.53**
Qwen2.5-7B-Instruct								
Direct	36.89	17.47	28.58	15.03	20.92	30.18	17.55	23.80
Naive RAG	**49.01**	**28.85**	40.91	20.53	23.94	**43.46**	**38.53**	**36.93**
Naive RAG w/. SFT	38.72	34.42	33.56	26.88	23.66	22.48	14.05	29.24
Naive RAG w/. reranker	47.24	25.24	40.16	21.69	24.11	43.19	37.67	34.18
CrossRAG	48.12	29.01	40.12	21.78	21.63	44.76	37.61	34.72
Search-o1	27.05	17.24	11.43	01.91	18.77	07.37	04.08	12.25
Search-R1	41.92	10.27	**41.76**	19.17	25.73	33.22	6.62	28.38
CoSelector (Ours)	45.97	24.78	41.74	**29.46**	**26.85**	38.78	29.79	35.71

Table 6: Comparison on 7 Languages of XOR-TyDi QA. The metric is C3Recall (%).

anced training set by uniformly sampling 1,000 instances per language. Retrieval is performed over the full multilingual corpus.

Training Tools. We use the LLaMA-Factory[3] framework for both supervised fine-tuning (SFT) and reinforcement learning (RL), with modifications to accommodate our specific training pipeline.

Language Detector. Because answers are multilingual short texts, we employ the lingua-language-detector[4] tool to ensure accurate RLC.

Evaluation Protocols. We adopt five fine-grained metrics to evaluate the practical contribution of the Selector. Response Language Consistency (RLC) evaluates the consistency between generated responses and user query languages. Selecting F1 assesses the evidence selection accuracy of the Selector Agent, with ground truth labels obtained from the answer-in-document heuristic. We further adopt Context Length to count the character number of input context, and employ Qwen3-32B to calculate Redundancy Ratio and Irrelevance Ratio, which respectively quantify the proportion of redundant and irrelevant information within retrieved context.

B Heuristic Algorithms for Constructing SFT data

Selector SFT Data Construction. The training and evaluation of CoSelector involve 15 languages.

[3] https://github.com/hiyouga/LlamaFactory

[4] https://pypi.org/project/lingua-language-detector/

XOR dataset translate example:

<|im_start|>system\n你是一个多语言翻译专家，精通te（泰卢固语），bn（孟加拉语），ru（俄语），ja（日语），ko（韩语），
ar（阿拉伯语），fi（芬兰语）这7种语言之间的互相翻译。需要将以下内容从ru（俄语）翻译到te（泰卢固语）。输入格式(Translation: *You are a multilingual translation expert, proficient in translating between 7 languages: Telugu, Bengali, Russian, Japanese, Korean, Arabic, and Finnish. You need to translate the following content from Russian to Telugu. Input format:*)：[Language:ru|Query]: {query content}输出格式：[Language:te|Query]: {translated query}
<|im_end|>\n<|im_start|>user\n[Language:ru|Query]: Какой климат в Ишимбае? (Translation: *What is the climate like in Ishimbay?*)<|im_end|>\n<|im_start|>assistant\n/no_think

Figure 4: XOR-TyDi QA query translation template.

To obtain supervisory signals indicating which multilingual documents are relevant to an answer, we employ an efficient heuristic based on lexical overlap: a document is considered relevant if the answer string appears within it. For each document, we first filter out standard stopwords. We then compute a lexical matching score between the document and each candidate answer. Documents with a matching score greater than zero are considered relevant and are ranked in descending order of their score. others are assigned a score of zero and deemed irrelevant. We construct binary sequences (yes/no) for each language group, indicating document relevance. Due to significant differences in tokenization across languages, we use a combination of tokenization tools: spaCy 3.8.0 for de, en, es, fi, fr, it, ja, ko, pt, ru, zh. CAMeL Tools for ar. PyThaiNLP for th; the indic-nlp-library for te; and the bnlp-

toolkit for bn.

Generator SFT Data Construction. For the Generator's SFT stage, the input is constructed by concatenating the query with the top-ranked relevant documents (labeled yes by the above heuristic) retrieved for that query. The corresponding gold answer serves as the target output sequence. This teaches the Generator to produce answers conditioned on the filtered, multi-document context provided by the Selector.

C Other Results

C.1 Results on XOR-TyDi QA dataset

As shown in Table 6, our method achieves competitive performance across most languages, yet suffers from the performance decline for Russian (ru) and Telugu (te) on the XOR-TyDi QA benchmark. This phenomenon can be attributed to the differences in data characteristics, processing pipelines, and dataset scales. MKQA is a high-quality, human-translated multilingual parallel corpus that guarantees precise and consistent cross-lingual alignment. In contrast, XOR-TyDi QA consists of non-parallel multilingual data. To support cross-lingual retrieval, we adopt Qwen3-32B-Instruct to translate the corpus for cross-lingual alignment, and the cumulative noise introduced during translation severely degrades the model performance on Russian and Telugu. Furthermore, the limited test scale of XOR-TyDi QA serves as another critical factor. XOR-TyDi QA only contains 3,743 test samples, while MKQA includes up to 26,000 test instances. The insufficient test data of XOR-TyDi QA leads to less stable and generalizable evaluation results, exacerbating the performance drop on low-resource languages. Details of the translation procedure are provided in Figure 4.

C.2 Results with Different Retrieval Strategies

We explore two distinct retrieval strategies regarding language balance of retrieved documents. One is the balanced strategy: we first select the top-3 documents per language, then score candidates via retriever or reranker, and finally retain the top-10 documents to maintain language diversity. The other is the unbalanced strategy: we directly pick the top-10 documents with the highest global scores, which may lead to missing results of certain languages.

As shown in Table 7, the unbalanced strategy generally achieves slightly better results, which may stem from the introduction of multilingual noise. The balanced strategy consistently yields higher performance with SFT training, while such regularity cannot be observed under the unbalanced setting.

For CoSelector, we did not adopt the unbalanced strategy. Our core consideration is to avoid language selection bias at the early data preparation stage of model training. Uniform sampling across all languages ensures that information from low-resource languages can be fully leveraged.

D Case Study

We highlight a critical failure mode of retrieval-augmented systems lacking cross-lingual routing and the corresponding strength of our framework through three groups of case studies.

As shown in Figure 5, we discuss the phenomenon of language-agnostic retrieval. When processing the Korean query "*When did Mother's Day become a public holiday?*", the Search-R1 baseline, which lacks explicit multilingual filtering, demonstrates its weakness. Its iterative retrieval fails to navigate the multilingual corpus effectively. The initial search retrieves irrelevant documents about Mother's Day memorials in Korea, and a subsequent search erroneously focuses on Japanese documents. Consequently, the model is misled by this noisy and linguistically misaligned context, producing the wrong answer "1956". This underscores how language barriers and retrieval noise can severely degrade performance in the absence of a dedicated routing mechanism. In contrast, CoSelector successfully addresses this challenge through its intelligent routing module. For the same query, the Selector analyzes the intent and accurately filters the multilingual document pool (containing 39 documents in 13 languages). It identifies and selects key evidence fragments from English, German, and Spanish documents, all of which consistently cite the 1914 U.S. legislation that established Mother's Day. Leveraging this coherent, cross-lingually corroborated context, CoSelector correctly deduces the answer "1914-5-9", aligning perfectly with the ground truth "1914". This example validates the Selector's ability to perform precise, intent-aware selection across languages, which is fundamental to accurate multilingual QA.

As shown in Figure 6, in this case, CoSelector

Strategy	Methods	Query Language													
		ar	de	es	en	fi	fr	it	ja	ru	th	zh	pt	ko	**Avg.**
Qwen2.5-3B-Instruct															
Unbalanced	Naive RAG	20.50	42.24	45.83	**51.15**	37.49	43.98	44.20	21.14	29.61	18.70	19.87	44.55	15.36	33.43
	Naive RAG w/. SFT	21.84	39.40	39.42	43.67	33.10	40.18	40.09	25.11	27.41	22.32	25.91	40.46	19.18	32.25
Balanced	Naive RAG	15.50	42.87	**53.43**	44.42	38.64	44.11	43.64	17.90	13.16	42.14	23.42	16.65	18.06	31.84
	Naive RAG w/. SFT	27.39	41.69	40.90	47.82	34.55	41.94	40.44	30.19	31.66	24.43	29.43	41.40	23.27	35.01
	CoSelector (Ours)	**29.97**	**45.86**	45.36	50.49	**41.19**	**45.93**	**45.61**	**34.90**	**36.95**	**27.53**	**31.40**	**46.34**	**27.15**	**39.13**
Qwen2.5-7B-Instruct															
Unbalanced	Naive RAG	24.52	48.98	50.64	56.18	45.72	48.78	50.04	26.45	33.61	23.01	23.53	50.11	19.20	38.52
	Naive RAG w/. SFT	34.30	46.90	45.70	51.90	43.30	48.00	48.60	37.10	38.10	**30.00**	35.30	47.90	30.10	41.30
Balanced	Naive RAG	24.15	46.53	48.68	57.12	42.99	47.44	47.19	26.29	32.80	22.10	23.39	46.92	18.98	37.27
	Naive RAG w/. SFT	31.55	45.11	44.19	52.62	40.68	46.42	46.08	35.71	36.63	29.42	33.79	45.63	30.01	39.83
	CoSelector (Ours)	**38.36**	**53.82**	**53.17**	**57.15**	**48.00**	**53.68**	**51.55**	**40.90**	**42.94**	28.41	**37.12**	**53.48**	**32.78**	**45.49**

Table 7: C3Recall (%) on MKQA between **Unbalanced** (Global top-10) and **Balanced** (top-3 to top-10).

CoSelector (Ours)

[Language:ko|Query]: 언제 어머니의 날이 공휴일로 만들어졌나요 (Translation: *When was Mother's Day made a public holiday?*)

Selector Input: There are 13 sets in 13 languages, with 3 documents in each set, totaling 39 multilingual documents. These documents cover multilingual texts related to "Mother's Day", "Parents' Day", "public holidays" and other completely unrelated holidays (such as Father's Day, National Day, Earth Day, International Women's Day, etc.).

Selector Output:
[Language:de|Document:1]: Clearly states that in 1914, the U.S. Congress passed legislation establishing Mother's Day as a national holiday for the first time. (Translation: *Clearly states that in 1914, the U.S. Congress passed legislation establishing Mother's Day as a national holiday for the first time.*)

[Language:en|Document:0]: Clearly states that in 1914, President Woodrow Wilson signed a proclamation designating Mother's Day as a national holiday.

[Language:en|Document:1]: States Congress passed a law establishing Mother's Day on May 8, 1914.

[Language:en|Document:2]: In 1914, Congress approved a joint resolution, which the President signed establishing Mother's Day.

CoSelector Answer: 1914-5-9

Ground Truth: 1914

Search-R1

[Language:ko|Query]:언제 어머니의 날이 공휴일로 만들어졌나요 (Translation: *When was Mother's Day made a public holiday?*)

Retrieval Process:

First Search: <search>어머니의 날이 공휴일로 만들어진 날짜 (Translation: *When was Mother's Day made a public holiday?*) </search> Result:- [KO] Doc 2: "국가적으로 어머니의 날을 지키게 된 것은 **1956년** 국무회의의 결정에 따른 것으로" (Translation: *The national observance of Mother's Day was based on a decision by the Cabinet in 1956.*)

Second Search: <search>어머니의 날이 공휴일로 지정된 날짜 (Translation: *The date Mother's Day is designated as a public holiday*) </search> Result:- [JA] Doc 2: "**1914年**に「母の日」はアメリカの記念日になり" (Translation: *"Mother's Day became an American holiday in 1914."*)

Search-R1 Answer: 어머니의 날이 공휴일로 지정된 날짜는 **1956년 5월 8일**로, 이는 한국의 문화와 어머니를 공경하는 전통을 결합하여 제정되었습니다. 이 날은 1973년에 '어버이날'로 명명되어 남녀를 불문하고 부모를 존경하는 날로 바뀌었습니다. (Translation: *Mother's Day was designated as a public holiday on May 8, 1956, and was established by combining Korean culture with the tradition of honoring mothers. In 1973, the day was renamed 'Parents' Day' and transformed into a day to respect parents regardless of gender.*)

Figure 5: A detailed case of the phenomenon of language-agnostic retrieval.

accurately filtered multilingual documents directly related to L&T (e.g., describing its history and business in German, English, and Spanish). Although an irrelevant Arabic document was incorrectly included, the predominance of correct information enabled the model to correctly answer "no". In contrast, Search-R1 conducted two retrieval rounds: the first retrieved irrelevant information, and the second failed to mention the Fortune 500 ranking, ultimately leading to an incorrect answer.

As shown in Figure 7, this case highlights a key behavioral difference: Search-R1 answers the query by relying solely on its internal parametric knowledge without invoking the retrieval module, whereas CoSelector correctly grounds its response in precisely filtered and consolidated multilingual evidence.

Failure mode proportions for English and Thai. Table 8 categorizes failure cases for English and Thai queries. "No Native Doc" dominates failures for both languages, meaning the retrieved pool lacks documents written in the query language. This accounts for 56% and 51% of errors in English and Thai respectively. Numerical Collapse is another major failure source, where the model misreuses numerical values across unrelated questions. Thai exhibits high rates of Semantic Drift and Asian Language Proxy failure: polysemy in Thai and cross-lingual proxy substitution with Japanese/Korean are the key failure patterns for small models. Nearly all failed samples lack ground-truth supporting evidence inside the selected context, indicating that evidence selection quality is the root cause of failure. This table explains why CoSelector underperforms Naive RAG on Thai under Qwen2.5-3B-Instruct: small models struggle to execute the multi-agent selection procedure reliably.

CoSelector (Ours)	Search-R1
[Language:it|Query]: larsen and toubro è una delle compagnie del fortune 500 (Translation: *Larsen and Toubro is one of the fortune 500 companies*)	**[Language:it|Query]**:larsen and toubro è una delle compagnie del fortune 500 (Translation: *Larsen and Toubro is one of the fortune 500 companies*)
Selector Input: There are 13 sets in 13 languages, with 3 documents in each set, totaling 39 multilingual documents. The content includes:1. Documents related to Larsen & Toubro itself (e.g., de-0, en-0, es-0, fi-0, etc.), describing its history, business, and size.2. General documents related to the Fortune 500 list (e.g., de-2, es-2, fi-2, etc.), explaining what the list is, but without mentioning L&T.3. Documents completely unrelated to the query (e.g., ar-0 discusses the Swedish economy, ko-0 discusses cameras, th-0 discusses the S&P 500 index). **Selector Output:** **[Language:de|Document:0]:** Introduces Larsen & Toubro as an Indian construction and engineering company.... **[Language:de|Document:1]:** Recounts L&T's early history, including its contract with Caterpillar.... **[Language:en|Document:0]:** Describes L&T as an Indian multinational conglomerate and one of the world's leading construction companies.... **[Language:en|Document:1]:** Lists only L&T-related cross-references.... **[Language:en|Document:2]:** Introduces one of L&T's financial subsidiaries.... **[Language:es|Document:0]:** Introduces L&T as an Indian multinational conglomerate with $17 billion in revenue in 2017.... **[Language:es|Document:1]:** Recounts the founding history of L&T.... **[Language:ar|Document:0]~[Language:ar|Document:2]:** These three Arabic documents are completely unrelated to L&T or Fortune 500 and should theoretically not be selected, yet they appear in the generator input. This may indicate an error in the filter's judgment or a special case in the workflow... example. **Ground Truth: no** **CoSelector Answer: no**	**Retrieval Process:** **First Search:** <search>Quali sono le principali imprese di Larsen & Toubro? (Translation: *What are Larsen & Toubro's main businesses?*)</search> Result: - [IT] Doc 1: "Tra le aziende di importanza interregionale... (Translation: *Among the companies of interregional importance...*)"(Unrelated) **Second Search:** <search>È Larsen & Toubro nella Fortune 500? (Translation: *Is Larsen & Toubro in the Fortune 500?*)</search>Result: - [EN] Doc 1: "Larsen & Toubro Ltd, commonly known as L&T, is an Indian multinational conglomerate company..." (Fortune 500 was not mentioned.) **Search-R1 Answer: Sì**, Larsen & Toubro è una delle imprese Fortune 500. È una delle principali imprese congiunte indiane con interessi nel settore dell'ingegneria, costruzioni, manifattura, tecnologia, informatica e altri. (Translation: *Yes, Larsen & Toubro is a Fortune 500 company. It is one of India's leading joint ventures with interests in engineering, construction, manufacturing, technology, information technology, and other sectors.*)

Figure 6: A detailed case of filtering effects.

Failure Mode	en %	th %	Query Example
Numerical Collapse	29%	24%	2017 life expectancy, Montgomery distance
No Native Doc	56%	51%	10 N·m = J, Las Vegas elevation
No GT Support	∼100%	∼100%	Nearly all failures
Entity Confusion	High	Medium	Fugees vs Flack, Bee Gees yes/no
Semantic Drift	Low	High	Large flat plate, CB radio
Asian Language Proxy	Low	High	Will Arnett, Lynyrd Skynyrd
Native GT Discarded	—	Medium	Día de los Muertos (Mexico)

Table 8: Failure mode proportions for English and Thai.

Qualitative examples. Table 9 provides qualitative case studies showing how the Selector agent keeps useful cross-lingual evidence and discards redundant or misleading documents. In successful cases, the selector retains cross-lingual supporting clues while filtering redundant and interfering documents (e.g., wrong airports in the Phoenix airport example). In failure cases, the selector may discard valid ground-truth evidence or keep irrelevant paragraphs. These examples illustrate the core advantage of CoSelector: it can perform context composition across multiple languages, rather than only ranking documents within a single language. Meanwhile, the cases also reveal limitations: cross-lingual ambiguity may confuse the Selector and lead to incorrect filtering decisions.

E Prompt Template

E.1 Selector Agent

As illustrated in Figure 8, the Selector takes the multilingual retrieval results as input and produces relevance labels for each.

E.2 Generator Agent

As illustrated in Figure 9, the Generator then produces the final answer based on the multilingual documents selected and ranked by the Selector.

CoSelector (Ours)	Search-R1
[Language:de|Query]: wer sang das lied first time ever i saw your face	**[Language:de|Query]**: wer sang das lied first time ever i saw your face
Selector Input: There are 13 sets in 13 languages, with 3 documents in each set, totaling 39 multilingual documents, covering information such as the song's creator, singer, background, and album. **CoSelector Output:** The following 11 documents were identified as relevant (relevance="yes") and entered into the generator: **[Language:de|Document:0]:** Introduces that Ewan McColl wrote this song for Peggy Seeger, which was later sung by Roberta Flack and won a Grammy Award. **[Language:de|Document:1]:** Similar content, emphasizing that McColl wrote it for Seeger and that Flack sang it and won an award. **[Language:en|Document:0]:** Provides details that the song was written by Ewan MacColl for Peggy Seeger, and that Roberta Flack sang it in 1972 and won a Grammy Award. **[Language:en|Document:1]:** Indicates that the song was written by Ewan MacColl and lists the relevant albums. **[Language:en|Document:2]:** Emphasizes that Roberta Flack's version was successful on the charts and won a Grammy Award. 6. **[Language:es|Document:0]:** Indicates the song was written by Ewan MacColl for Peggy Seeger. **[Language:es|Document:1]:** Mentions that Roberta Flack's version appeared in a movie and became a hit song. **[Language:fi|Document:0]:** The song list includes "The First Time Ever (I Saw Your Face)," written by Ewan MacColl. **[Language:fr|Document:0]:** Indicates the song was written by Ewan MacColl, recorded by Roberta Flack, and won a Grammy Award. [Language:fr|Document:1]: Mentions that Roberta Flack sang this song in 1972. However, **[Language:ar|Document:0]** describes a song first performed in Boston in 1857 by Johnny Bill, which is completely unrelated to the song in the query. **CoSelector Answer:** Roberta Flack **Ground Truth:** Roberta Flack	**Retrieval Process:** No Search **Search-R1Answer:** Peter, Paul and Mary

Figure 7: A case of retrieval availability.

Query	GT / Pred	Action
(EN) who is the presiding officer of the senate when the vice president is not there	President pro tempore of the United States Senate / president pro tempore	**Keep**: [en:0] explicitly states "president pro tempore . . . presiding officer when VP absent". **Drop**: [en:1] same topic but redundant phrasing; [fr:2], [ja:0] cross-lingual repetition, non-critical. **Keep**: [ar:0] cross-lingual relevance but suboptimal.
(DE) Wie heißt der Flughafen von Phoenix?	Phoenix Sky Harbor International Airport / Phoenix Sky Harbor International Airport	**Keep**: [en:0] "Phoenix Sky Harbor International Airport" disambiguation list. **Drop**: [de:0] Deer Valley; [de:2] Gateway Airport (other airports in the same city causing interference). **Keep**: [ar:0] Arabic description of the same airport.
(DE) durchschnittliche Lebensdauer eines Menschen im Jahr 2017	72.0 Jahr / 78.6 Jahr	**Keep**: [en:0] US life expectancy at age 65; [ja:0] healthy life expectancy 72.14; [ja:1] US total population 78.6 years; [ja:2] irrelevant text. **Drop**: [de:0] global life expectancy trend; [fi:1] 72.

Table 9: Qualitative examples: kept and discarded evidence by CoSelector's Selector agent. GT means Ground Truth.

CoSelector-Selector example:

{"**instruction**": "Judge the relevance between the documents and query for the current language. There are 3 documents. Output a space-separated sequence of 'yes' or 'no' to indicate relevance for each document in order.",

"**input**": "**[Language:de|Query]**:was ist die höhe der Internationale Raumstation(Translation: *What is the altitude of the International Space Station?*)\n\n**[Language:ja|Document:0]** 比較として、国際宇宙ステーションの軌道小路は408 kmである。 また、1969年のアポロ9号以来の大西洋への有人の着水だった。(Translation: *For comparison, the International Space Station's orbital path is 408 km. Furthermore, this was the first manned splashdown in the Atlantic Ocean since Apollo 9 in 1969.*)\n\n**[Language:ja|Document:1]** グローバルスターの軌道高は1,420km、イリジウムの軌道高は670kmである。(Translation: *Globalstar's orbital altitude is 1,420 km, while Iridium's orbital altitude is 670 km.*)\n\n**[Language:ja|Document:2]** 例 \n国際宇宙ステーションは、公転周期が91.74分で、軌道長半径は6738kmである。関連項目 \n 軌道短半径\n\n軌道\n力学\n天文学に関する記事 (Translation: *Example: The International Space Station has an orbital period of 91.74 minutes and an orbital semi-major axis of 6738 km. Related topics: Orbital semi-minor axis, Orbit, Dynamics, Articles on astronomy*)",

"**output**": "yes no no",

"**system**": "You are a helpful, respectful and honest assistant. Your task is to judge the relevance between the documents and query for a specific language.",

"**history**": []}

Figure 8: An example for Selector.

CoSelector-Generator example:

{"**instruction**": "The language tag for a multilingual query is [Language:lang|Query], and the language tag for a multilingual document is [Language:lang|Document:num]. Generate a concise answer consistent with the query language based on the multilingual document.",

"**input**": "**[Language:de|Query]**:was ist die höhe der Internationale Raumstation(Translation: *What is the altitude of the International Space Station?*)\n**[Language:it|Document:1]** Il trasferimento da Hubble alla stazione spaziale \n\nNella realtà il trasferimento dal telescopio Hubble alla Stazione Spaziale Internazionale non potrebbe avvenire come proposto nel film, innanzi tutto perché si trovano in orbite completamente diverse: Hubble ad un'altitudine di circa 559 chilometri con un'inclinazione orbitale di 28,5 gradi, mentre la Stazione Spaziale Internazionale si trova ad un'altitudine di circa 420 chilometri con un'inclinazione orbitale di 51,65 gradi.(Translation: *The transfer from Hubble to the space station \n\nIn reality, the transfer from the Hubble telescope to the International Space Station could not take place as proposed in the film, first of all because they are in completely different orbits: Hubble at an altitude of about 559 kilometers with an orbital inclination of 28.5 degrees, while the International Space Station is at an altitude of about 420 kilometers with an orbital inclination of 51.65 degrees.*)\n**[Language:ja|Document:0]** 比較として、国際宇宙ステーションの軌道小路は408 kmである。 また、1969年のアポロ9号以来の大西洋への有人の着水だった。(Translation: *For comparison, the International Space Station's orbital path is 408 km. Furthermore, this was the first manned splashdown in the Atlantic Ocean since Apollo 9 in 1969.*)\n**[Language:th|Document:1]** สถานีอวกาศ () เป็นสิ่งก่อสร้างที่ออกแบบโดยมนุษย์ เพื่อใช้เป็นที่อยู่การดำรงชีพในอวกาศ โดยอยู่ในวงโคจรต่ำ (LEO)\n\nสถานีอวกาศนานาชาติ (International Space Station) หรือ ISS เป็นห้องปฏิบัติการลอยฟ้าซึ่งโคจรรอบโลกที่ระยะสูง 410 กิโลเมตร เคลื่อนที่ด้วยความเร็ว 27,744 กิโลเมตร/ชั่วโมง โคจรรอบโลก (Translation: *A space station is a man-made structure designed for habitation in space, located in low Earth orbit (LEO).\n\nThe International Space Station (ISS) is a floating laboratory orbiting the Earth at an altitude of 410 kilometers, traveling at a speed of 27,744 kilometers per hour.*)\n\nNow, answer the Question: was ist die höhe der Internationale Raumstation, based on the above Multilingual Documents as briefly and accurately as possible.",

"**output**": "408.0 Kilometer",

"**system**": "You are a helpful, respectful and honest assistant. Your task is to generate an accurate answer based on multilingual documents.",

"**history**": []}

Figure 9: An example for Generator.