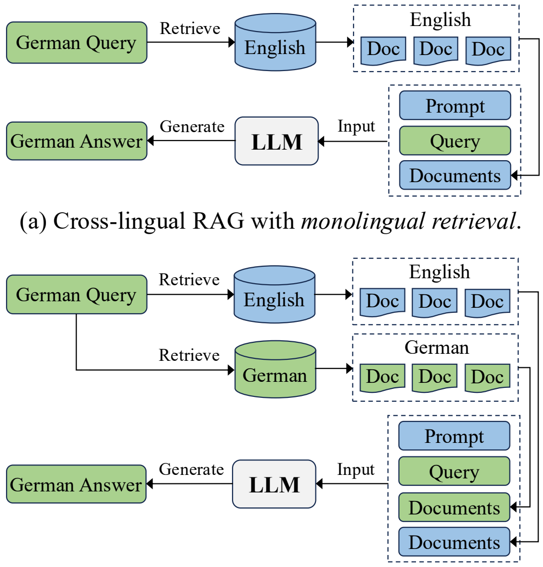
XRAG: Cross-lingual Retrieval-Augmented Generation

Wei Liu[1*], **Sony Trenous**[2], **Leonardo F. R. Ribeiro**[2], **Bill Byrne**[2], **Felix Hieber**[2]
[1]Heidelberg Institute for Theoretical Studies gGmbH
[2]Amazon AGI
wei.liu@h-its.org, {trenous,leonribe,willbyrn,fhieber}@amazon.com

Abstract

We propose XRAG, a novel benchmark designed to evaluate the generation abilities of LLMs in cross-lingual Retrieval-Augmented Generation (RAG) settings where the user language does not match the retrieval results. XRAG is constructed from recent news articles to ensure that its questions require external knowledge to be answered. It covers the real-world scenarios of monolingual and multilingual retrieval, and provides relevancy annotations for each retrieved document. Our novel dataset construction pipeline results in questions that require complex reasoning, as evidenced by the significant gap between human and LLM performance. Consequently, XRAG serves as a valuable benchmark for studying LLM reasoning abilities, even before considering the additional cross-lingual complexity. Experimental results on five LLMs uncover two previously unreported challenges in cross-lingual RAG: 1) in the monolingual retrieval setting, all evaluated models struggle with response language correctness; 2) in the multilingual retrieval setting, the main challenge lies in reasoning over retrieved information across languages rather than generation of non-English text.[1]



(a) Cross-lingual RAG with *monolingual retrieval*.

(b) Cross-lingual RAG with *multilingual retrieval*.

Figure 1: Two cases of cross-lingual RAG: (a) monolingual retrieval, where the LLM uses retrieved English documents to respond to a German query; (b) multilingual retrieval, where the LLM uses retrieved English and German documents to respond to a German query.

1 Introduction

Retrieval-augmented generation (RAG) augments large language models (LLMs) by retrieval of relevant documents with the aim of improving response quality (Lewis et al., 2020). The widespread adoption of RAG has prompted many recent studies to evaluate specific capabilities of LLMs in RAG settings, such as robustness to noise (Wang et al., 2024), information integration (Chen et al., 2024b), time sensitivity (Kasai et al., 2023), multi-hop reasoning (Tang and Yang, 2024) and conversational QA (Roy et al., 2024). Notably, these evaluations are in monolingual settings in which questions and retrieved documents are in the same language.

Real-world deployments of RAG systems also need to handle cross-lingual use cases, where the user's language does not match that of the retrieved documents. The simplest scenario is **Cross-lingual RAG with Monolingual Retrieval** (Asai et al., 2023), where users in multiple locales are served by a single RAG system that accesses an English-only knowledge base, as illustrated in Figure 1a. This setup applies to, for example, a general-purpose RAG system that relies solely on English web search or a corporate helpdesk with an internal database available only in English. A more complex scenario is **Cross-lingual RAG with Multilingual Retrieval**, where RAG systems combine information from both English and the user's language to generate a response (see Figure 1b). This is a

[*] Work done during an internship at Amazon.
[1] Data and Code of XRAG are available at https://huggingface.co/datasets/AmazonScience/XRAG

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 15669–15690
November 4-9, 2025 ©2025 Association for Computational Linguistics

common situation in that native-language sources often contain culturally or geographically specific knowledge, with English resources providing additional, more general information.[2]

Due to the absence of relevant benchmarks, we lack an understanding of how well LLMs can handle such cross-lingual RAG scenarios. A potential solution is to use existing cross-lingual open-domain question-answering datasets, such as XQA (Liu et al., 2019a) and XOR QA (Asai et al., 2021), for evaluation (Chirkova et al., 2024). Yet these datasets only cover limited cross-lingual scenarios; in particular, the documents used to answer questions are in English, which hinders the evaluation of LLMs in more complex multilingual scenarios (i.e., Figure 1b). Moreover, the questions in these datasets tend to be relatively simple (e.g. span extraction questions) and often can be answered without retrieval.[3] Due to these shortcomings, these datasets do not measure the true cross-lingual capabilities of LLMs in RAG settings.

To address this gap, we introduce XRAG, a benchmark for evaluating the Question Answering capabilities of LLMs in cross-lingual RAG scenarios, where some information must be extracted from retrieved documents that are not in the user's language. The benchmark features natural-sounding questions that require cross-document reasoning and are **challenging for LLMs even in an English monolingual RAG setting** (GPT-4o achieves only 62.4% accuracy, see Table 4). We develop a novel LLM-based question generation workflow using recent news articles, ensuring that current frontier models are **unable to answer the questions without retrieval** (GPT-4o accuracy is 6.3% without retrieval, see Table 3). To guarantee a high-quality dataset, we employ extensive human Quality Assurance, resulting in **few ambiguous or noisy questions** (under 8%, see Section 5.2). In addition to English, the benchmark spans **four widely spoken and linguistically diverse languages** (Arabic, Chinese, German, and Spanish).

XRAG comprises two sub-tasks, corresponding to the *monolingual retrieval* and the *multilingual retrieval* settings of cross-lingual RAG. For each non-English language, we provide a directly comparable English monolingual RAG baseline task. Each instance in the XRAG benchmark consists of a question, a gold answer, two supporting articles that together answer the question, and six topically related but non-answering distracting articles. This allows us to approximate realistic RAG settings with imperfect retrieval in evaluating the Question Answering abilities of LLMs.

We evaluate five LLMs, including both closed- and open-source models, on XRAG. In summary, our contributions are:

(1) We introduce XRAG, a novel benchmark designed to evaluate the performance of LLMs in two cross-lingual RAG scenarios.

(2) We propose a novel method for generating challenging cross-document QA pairs from News Crawl, resulting in natural questions that current LLMs cannot answer using only their parametric knowledge.

(3) We find that in the *monolingual retrieval* setting, all evaluated LLMs face issues with Response Language Correctness-an issue that has received little attention from the research community.

(4) In the *multilingual retrieval* setting, the primary challenge for LLMs does not lie in non-English generation, but in reasoning over retrieved information across languages.

2 Related Work

There are extensive recent investigations into characterizing the Question Answering capabilities of LLMs in RAG settings. Vu et al. (2024) construct a a dynamic QA benchmark, FreshQA, that tests the ability of LLMs to use up-to-date world knowledge to solve questions. Chen et al. (2024c) introduce RGB to analyze fundamental abilities of LLMs in RAG systems, such as noise robustness and negative rejection. Tang and Yang (2024) propose MultiHop-RAG, which focuses on whether retrieval-enhanced LLMs can retrieve and reason over multiple pieces of supporting evidence. The Comprehensive RAG Benchmark, created by Yang et al. (2024), aims to assess whether LLMs are able to answer different types of questions, ranging from simple to complex. Thakur et al. (2024b) present MIRAGE-Bench, a multilingual RAG benchmark constructed from Wikipedia, to evaluate RAG systems performance in different languages. Zhu et al.

[2] An initial study on a proprietary dataset of real-world LLM traffic from non-English users in Germany, Japan, and Spain found that using only English or native-language search results was inferior to combining both (see Appendix A).

[3] Chirkova et al. (2024) shows that 47.5% of questions in XORQA can be answered by Command-R without retrieval.

RAG Setting	Field	Language	Content
Monolingual Retrieval	Question	German	Wie viel haben Walmart und ALDI zusammen für die Opfer des Hurrikans Helene 2024 gespendet?
	Answer	German	Die gesamten Spenden überstiegen 11 Millionen Dollar.
	Supporting Articles	English	Walmart, Sam's Club and the Walmart Foundation are increasing their commitment to $10 million to Hurricane Helene Relief Effort...
		English	The American Red Cross recognizes ALDIfor its pledge of $1,000,000. By making a donation to Hurricane Helene Relief...
	Distracting Articles	English	Walmart Canada reaches new giving milestone of $750 million raised and donated to charities and non-profits across Canada...
		English	Aldi has donated £2,000 to charities in Gloucestershire to help support those in need during the school holidays. The donations...
Multilingual Retrieval	Question	German	Welches Land gewann seine erste Goldmedaille bei den Olympischen Spielen 2024 früher, die Vereinigten Staaten oder Deutschland?
	Answer	German	Deutschland. Beide Länder gewannen am 27. Juli bei den Schwimmwettbewerben ihre ersten Goldmedaillen, aber die USA gewannen ihre Medaille erst im letzten Wettkampf des Tages – später als Deutschland.
	Supporting Articles	German	Lukas Märtens hat am 27. Juli in Paris den olympischen Titel über 400 m Freistil gewonnen. Der Magdeburger siegte in 3:41,78...
		English	In the last swimming race of July 27, the U.S. took its first gold medal of the 2024 Olympics, winning the 4×100-meter freestyle...
	Distracting Articles	German	Im Rahmen von noch bevorstehenden Qualifikationsevents können sich weitere Sportler noch für die Spiele in Paris qualifizieren...
		English	The United States Olympic & Paralympic Committee have announced the 592-member 2024 U.S. Olympic team ready to compete...

Table 1: Two instances from XRAG, each consisting of a question, a gold answer, two supporting articles, and six distracting articles (two are shown). In the *monolingual retrieval* setting, all supporting and distracting articles are in English; in the *multilingual retrieval* setting, the supporting and distracting articles are in the question language and in English. LLMs should answer these questions based on the supporting articles while ignoring the distractors.

(2025) introduce RAGEval, a framework for evaluating RAG systems across diverse domains, such as medicine and law. As noted, these are in monolingual settings, whereas we aim to benchmark LLMs performance in cross-lingual RAG scenarios.

Chirkova et al. (2024) is one of few studies that evaluate the cross-lingual capabilities of LLMs in RAG systems. They conduct an analysis of existing cross-lingual open-domain question-answering datasets (Asai et al., 2021). Motivated by this prior work, XRAG is a new cross-lingual benchmark that covers a wider range of scenarios and consists of questions designed to require external knowledge to answer, thereby providing a more accurate reflection of the cross-lingual capabilities of LLMs in RAG settings. Li et al. (2024) present BordIRlines, which also considers cross-lingual RAG settings but with a fundamentally different objective. It examines hallucination and bias arising from documents in different languages, emphasizes sensitive topics (e.g., geopolitical disputes), and provides qualitative case studies rather than large-scale empirical evaluation. In contrast, XRAG targets general-purpose cross-lingual RAG scenarios, focuses on factual cross-document questions grounded in recent news, and supports scalable and reproducible benchmarking of LLM question-answering ability in cross-lingual RAG settings.

Evaluation of cross-lingual NLP systems is a long-standing research problem. Relatively recent work has focused on performance in specific NLP tasks such as NLI (Conneau et al., 2018; Liu et al., 2023), summarization (Wang et al., 2022), retrieval question answering (Roy et al., 2020), and open-domain question answering (Toutanova et al., 2021). With the advent of large language models, cross-lingual evaluation has expanded to include few-shot or even zero-shot settings. Wang et al. (2023) investigate GPT-4 performance for cross-lingual summarization in a zero-shot setting and find that it performs competitively with finetuned mBART-50. Ahuja et al. (2023) evaluate the performance of generative models on 15 tasks, covering classification, sequence labeling, and generation.

We note that our focus is on retrieval augmented generation from multilingual document retrieval, and not on the document retrieval task itself. However in Sections 4.1 and 4.4 we discuss how we use monolingual and multilingual dense document retrieval techniques in constructing XRAG. Our work aligns with recent efforts in evaluating cross-lingual performance of LLMs, with a focus on retrieval augmented generation and cross-lingual answer generation in particular.

3 XRAG - Cross-lingual RAG Benchmark

We define the task of cross-lingual RAG as follows: given a question q, the LLM is prompted to generate an answer $\tilde{a}$ in the same language as the question by referring to a collection of m articles $\mathrm{D} = \{d_1, d_2, ..., d_m\}$ that contains articles in a language different than the question:

$$\begin{aligned} &\tilde{a} \leftarrow \mathtt{LLM}(q, \mathrm{D}, \mathtt{prompt}) \\ &\mathtt{Language}(q) = \mathtt{Language}(\tilde{a}) \\ &\exists\, d_i \in \mathrm{D},\ \mathtt{Language}(d_i) \neq \mathtt{Language}(q) \end{aligned} \tag{1}$$

Figure 1 shows two cases, in which LLMs need to use information from *English* articles to generate *German* responses to *German* questions. The goal of our benchmark is to enable an understanding of **how well LLMs perform generation in such cross-lingual RAG scenarios**.

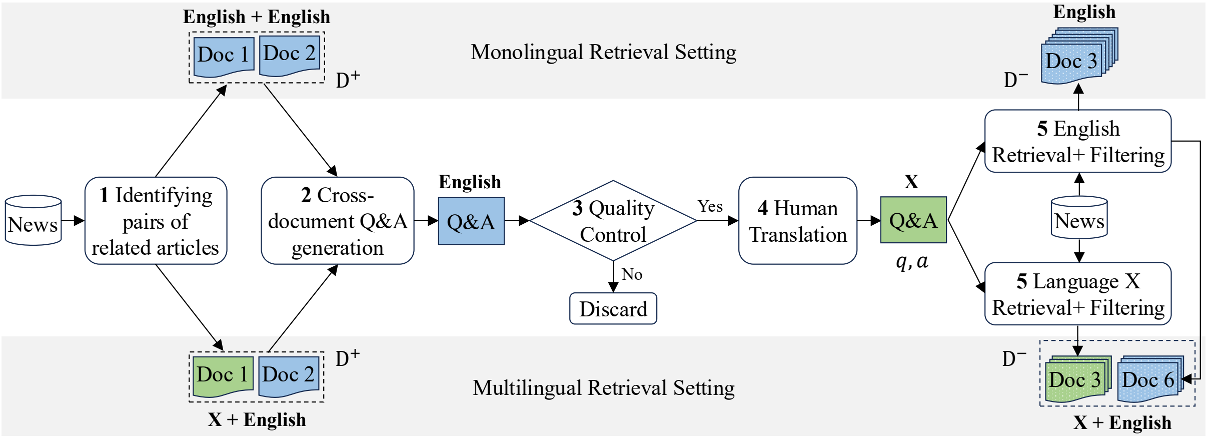

English + English
Doc 1
Doc 2
D+
Monolingual Retrieval Setting
English
Doc 3
D−
News
1 Identifying pairs of related articles
2 Cross-document Q&A generation
English
Q&A
3 Quality Control
Yes
No
Discard
4 Human Translation
X
Q&A
q, a
5 English Retrieval+ Filtering
News
5 Language X Retrieval+ Filtering
Doc 1
Doc 2
D+
X + English
Multilingual Retrieval Setting
D−
Doc 3
Doc 6
X + English

Figure 2: Each instance $(q, a, \mathrm{D}^+, \mathrm{D}^-)$ in XRAG — where q is the question, a the gold answer, D^+ the supporting articles, and D^- the distractors — is constructed as follows: (1) find two related articles; (2) generate an English cross-document Q&A pair using the two articles; (3) evaluate the quality of the Q&A pair; (4) translate the Q&A pair into language X ∈ {German, Spanish, Chinese, Arabic}; and (5) collect distracting articles for the question.

Each instance $(q, a, \mathrm{D}^+, \mathrm{D}^-)$ in XRAG consists of a question q, a golden answer a, two supporting articles D^+, and several distracting articles D^-. The supporting articles each contribute partial information needed to answer the question, and only together do they provide a complete answer; in contrast, the distracting articles are topically related but cannot answer the question. Taken together, this simulates a realistic RAG scenario with imperfect retrieval, where we can control the quality of the grounding by the inclusion of distractors. Questions in XRAG are cross-document questions, requiring reasoning across the two supporting articles to answer, while ignoring the distracting articles. Our benchmark considers two real-world cross-lingual RAG scenarios: the *monolingual retrieval* scenario and the *multilingual retrieval* scenario.

In the *monolingual retrieval* setting, LLMs rely on English articles to generate an answer. This occurs when users in multiple locales are served by a single cross-lingual RAG system that has access only to an English knowledge base. In this paper, we consider questions in four languages: German (de), Spanish (es), Chinese (zh), and Arabic (ar):

$$\begin{aligned}&\texttt{Language}(q) \in \{\mathrm{de}, \mathrm{es}, \mathrm{zh}, \mathrm{ar}\}\\&\texttt{Language}(\mathrm{D}^+) = \texttt{Language}(\mathrm{D}^-) = \mathrm{en}\end{aligned} \tag{2}$$

These four are widely used in the research community (Macko et al., 2023) and represent a range of cross-lingual challenges, ranging from easy (es-en) to challenging (zh-en) (Yang et al., 2022).

In a *multilingual retrieval* setting, LLMs use articles in both the question language and other languages to answer a question. This corresponds to a cross-lingual RAG scenario where documents in a resource-rich language provide additional information for LLMs to answer questions in a second language. Similarly, we consider four languages:

$$\begin{aligned}&\texttt{Language}(q) \in \{\mathrm{de}, \mathrm{es}, \mathrm{zh}, \mathrm{ar}\}\\&\texttt{Language}(\mathrm{D}^+) = \{\mathrm{en}, \texttt{Language}(q)\}\\&\texttt{Language}(\mathrm{D}^-) = \{\mathrm{en}, \texttt{Language}(q)\}\end{aligned} \tag{3}$$

Table 1 gives examples of *monolingual retrieval* and *multilingual retrieval* from XRAG.

4 XRAG Construction

Figure 2 shows the overall XRAG construction process. We begin with English, German, Spanish, Chinese, and Arabic news articles from News Crawl between June 1, 2024, and November 30, 2024. This timeframe ensures that the articles are dated after the knowledge cutoff of LLMs such as GPT-4 and Claude 3.5. Questions created from these articles are more likely to require external knowledge to answer.

4.1 Identifying pairs of related articles

To generate natural cross-document questions from a pair of articles, the articles must be topically related; otherwise, the generated questions may seem artificial (Welbl et al., 2018).

For the *monolingual retrieval* setting, we construct a bipartite graph linking English articles with the entities in their titles. We then use depth-first search to find pairs of articles that share at least two entities in their titles. These article pairs serve

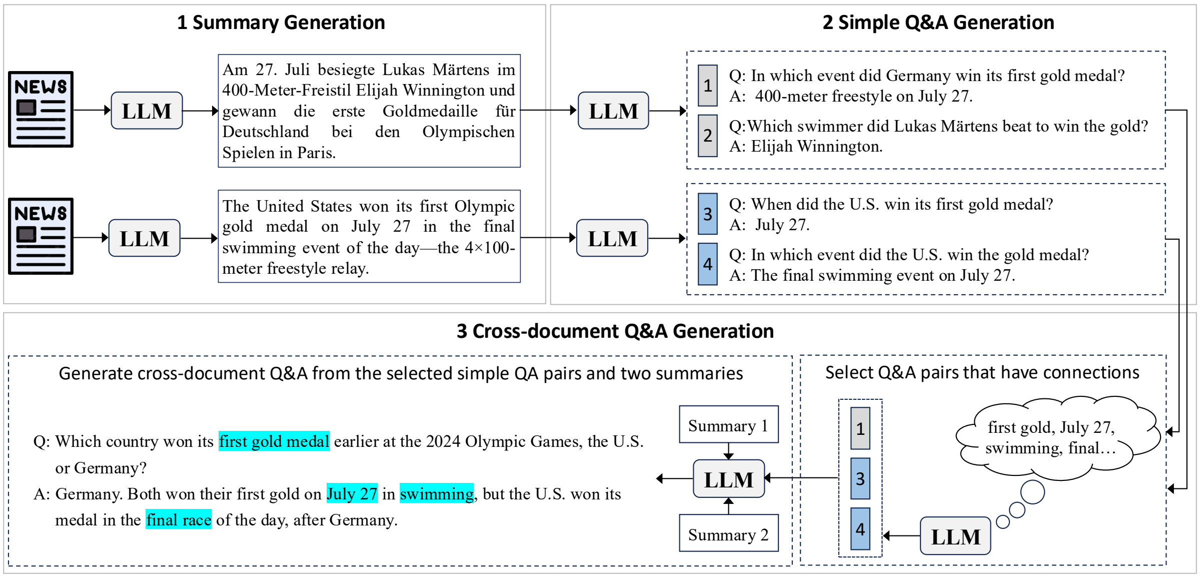

1 Summary Generation
NEWS
LLM
Am 27. Juli besiegte Lukas Märtens im 400-Meter-Freistil Elijah Winnington und gewann die erste Goldmedaille für Deutschland bei den Olympischen Spielen in Paris.
NEWS
LLM
The United States won its first Olympic gold medal on July 27 in the final swimming event of the day—the 4×100-meter freestyle relay.
2 Simple Q&A Generation
LLM
1 Q: In which event did Germany win its first gold medal? A: 400-meter freestyle on July 27.
2 Q:Which swimmer did Lukas Märtens beat to win the gold? A: Elijah Winnington.
LLM
3 Q: When did the U.S. win its first gold medal? A: July 27.
4 Q: In which event did the U.S. win the gold medal? A: The final swimming event on July 27.
3 Cross-document Q&A Generation
Generate cross-document Q&A from the selected simple QA pairs and two summaries
Q: Which country won its first gold medal earlier at the 2024 Olympic Games, the U.S. or Germany?
A: Germany. Both won their first gold on July 27 in swimming, but the U.S. won its medal in the final race of the day, after Germany.
Summary 1
LLM
Summary 2
Select Q&A pairs that have connections
1
3
4
first gold, July 27, swimming, final…
LLM

Figure 3: LLM-based workflow for generating English cross-document questions from a pair of related articles: (1) generate a summary for each article; (2) create simple English Q&A pairs from each summary that require only one-step reasoning; (3) identify connections between the two sets of Q&A pairs, select related ones, and construct a new Q&A pair that requires reasoning across multiple pieces of information from the selected pairs and summaries.

as related English articles for generating cross-document questions.

For the *multilingual retrieval* setting, we use international events from Wiki 2024 and a multilingual dense retriever to search across different languages for articles related to the events. We then group articles in different languages about the same event to form related article pairs.

We provide a more detailed explanation of how to locate relevant articles in English or across languages in Appendices B.2.

4.2 English cross-document Q&A generation

We design an LLM-based workflow[4] to generate natural and coherent English cross-document questions from news articles. Figure 3 shows an overview of the generation workflow.

Step 1: Summary Generation. Given a pair of related articles either in English or in English and another language, we prompt the LLM to create a summary for each article that (1) is accurate and concise; (2) covers the key points; and (3) has little lexical overlap with the article (the prompt is shown in Figure 10). These summaries are then used to generate questions in Step 2. There are two reasons for this: generating questions directly from articles often leads to questions with high overlap in wording, making it easy to answer through string matching; and direct question generation from articles can focus on trivial details, whereas generating questions from summaries tends to produce questions about the main points of the articles.

Step 2: Simple Q&A Generation. Our goal is to create cross-document questions that require information from two articles to answer. However, we find that generating such questions in one step is difficult. LLMs often create questions that simply link two separate questions with "and". Instead, we first prompt the LLM to generate simple English Q&A pairs from each summary that can be answered with one step of reasoning (the prompt is shown in Figure 11). For example, from a German report about Germany's first gold medal at the 2024 Olympics, the LLM generates Q&A pairs like: (**q**: In which event did Germany win its first gold medal? **a**: 400-meter freestyle on July 27).

Step 3: Cross-document Q&A Generation. After generating simple Q&A pairs from two summaries, we prompt the LLM to: (1) identify connections between the two sets of Q&A pairs; (2) select related Q&A pairs from the two sets, ensuring that at least one pair is chosen from each source; and (3) formulate new questions that require reasoning across multiple pieces of information drawn from the selected Q&A pairs. Since the selected Q&A pairs originate from different source articles, answering the newly generated questions necessitates integrating information from both sources, thus resulting

[4] We use GPT-4o-2024-08-06.

in cross-document questions. For example, using the simple Q&A pairs in Figure 3, the LLM finds links such as “first gold medal”, “swimming race”, “final” and the date “July 27” between the two sets of simple Q&A pairs. The LLM then generates a comparison question: “Which country won its first gold medal earlier at the 2024 Olympic Games, the U.S. or Germany?”. We then ask the LLM to generate an answer to the question using information from the selected simple Q&A pairs and the two summaries (the prompt for answer generation is in Figure 16). Inspired by Yang et al. (2024), we focus on four types of cross-document questions: aggregation, comparison, multi-hop, and set questions. We present the definition of the four types of questions in Table 8, and the prompts to generate these questions in Figures 12, 13, 14, and 15.

4.3 Quality Control and Human Translation

The generated Q&A pairs may contain factual errors due to LLM hallucinations (Huang et al., 2024). To avoid these, we ask a professional multilingual annotation team to verify the quality of the generated Q&A pairs (the annotation guideline is shown in Figure 17). They select examples where the question is natural and answerable and the answer is either correct or correctable by them. For the *monolingual retrieval* setting, we engage a professional translation team to translate the verified Q&A pairs into German, Spanish, Chinese, and Arabic. For the *multilingual retrieval* setting, translations are performed only into language X for Q&A pairs derived from X-English article pairs (e.g., into German for Q&A pair created from German-English article pairs). See Appendix B.5 for more details on human translation.

Table 2 presents the dataset statistics after human verification and translation.

4.4 Selecting the Distracting Articles

The grounding articles for each question consist of a set of supporting documents and distracting documents. The two articles used in cross-document question generation serve as supporting documents.

For distracting documents, we search for documents that are topically related to the question but do not answer it. In the *monolingual retrieval* setting, we use a multilingual dense retriever to search for English documents. In the *multilingual retrieval* setting we search for documents in both English and the question language. In both settings we select distracting documents that are published at least two weeks before the supporting articles to ensure that the distracting documents do not answer the question.

		Monolingual Retrieval	Multilingual Retrieval			
		De / Es / Zh / Ar	De	Es	Zh	Ar
Example Number		1000	300	300	300	300
Question	Aggregation	313	86	99	106	95
	Comparison	260	98	109	81	85
	Multi-hop	215	45	40	65	57
	Set	212	71	52	48	63
Answer	Original	872	291	296	284	263
	Corrected	128	9	4	16	37

Table 2: Statistics of XRAG question types in *monolingual* and *multilingual* retrieval settings. Answers that are corrected during quality control are also considered.

This process yields a set of grounding documents for each question. In the *monolingual retrieval* setting, each question will have six distracting documents and two supporting documents, all in English. In the *multilingual retrieval* setting, each question will have one supporting document and three distracting documents in English, and the same again in the question language.

5 Benchmarking with XRAG

5.1 Experimental Settings

Models. We benchmark five models on XRAG: GPT-4o (OpenAI, 2024), Claude Sonnet-3.5 v1 (Anthropic, 2024), Mistral-large (Jiang et al., 2023), Command-R+ (Cohere, 2024), and Nova Pro (Amazon, 2024). These are leading closed- and open-source multilingual LLMs, and have been widely used in RAG research. Unless otherwise specified, the evaluation is conducted by providing the LLM with a question, two supporting documents, and six distracting documents (we show the prompt used to instruct LLMs in using articles to answer questions in Figure 18). Figure 4 shows the evaluation workflow on XRAG.

Evaluation Metrics. The answers in XRAG are usually simple facts stated in one or two sentences. Following previous work (Yang et al., 2024; Wang et al., 2024; Thakur et al., 2024a), we use the LLM-as-a-Judge method (Zheng et al., 2023), which has proven good at recognizing when two short answers mean the same thing (Kamalloo et al., 2023). To avoid *self-preference* (Panickssery et al., 2024), we use a panel of three LLM judges (GPT-4o, Claude Sonnet-3.5, and Mistral-large) with a majority vote. We also use a language detector[5] to

[5] https://github.com/pemistahl/lingua

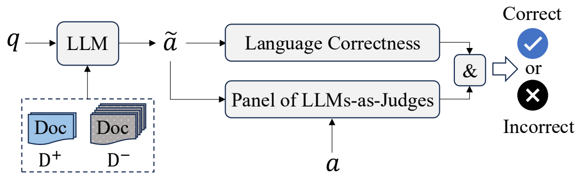


Figure 4: Evaluation workflow on XRAG: (1) the evaluated LLM generates a response $\tilde{a}$ for a question q based on two supporting articles D^+, and six distracting articles D^-; (2) the response is checked for language correctness; (3) a panel of three LLM judges independently assess the factual accuracy of the response based on the question q and a gold answer a, with the final judgment based on majority vote; (4) the final evaluation combines the factual judgment and language correctness.

check if the model answer is in the same language as the question; it is marked as incorrect otherwise. To confirm that automatic judging works well, we compare the LLM judge panel decisions with human judges and find a Cohen's kappa score of 0.71. Finally, we report each model's accuracy (%), as determined by the LLM judge panel, which includes the assessment of language correctness. The template used for the LLM-as-a-Judge is shown in Figure 20, and more details regarding the correlation experiments between the LLM judge panel and human evaluations are provided in Appendix C.2.

5.2 Establishing QA Performance Bounds

Our goal is to create a cross-lingual RAG benchmark with two key properties: (1) questions should not be answerable using only the parametric knowledge of LLMs, and (2) the task includes challenging questions that require complex reasoning to answer. To assess whether XRAG meets these properties, we evaluate performance of several LLMs on English questions from the *monolingual retrieval* setting of XRAG (see English Q&A in Figure 2) under two conditions: without retrieval, and with the correct supporting articles. Table 3 presents the results of models evaluated by the LLM judge panel. All LLMs perform poorly without retrieval, with accuracy rates falling below 16%. This indicates that these LLMs cannot answer XRAG questions by relying solely on their parametric knowledge. Even when given supporting articles, simulating ideal retrieval, the best result, achieved by GPT-4o,[6] reaches only 75.40% accuracy, which is still far below human accuracy, as we discuss next. These findings show that XRAG questions are challenging even for advanced LLMs.

	No Retrieval	Oracle Retrieval
GPT-4o	6.30	75.40
Claude Sonnet 3.5	11.70	67.60
Mistral-Large	15.20	66.40
Command-R+	15.30	63.50
Nova-Pro	13.70	68.20

Table 3: LLM QA accuracy in answering XRAG questions without retrieval, and with XRAG supporting articles (but without distracting articles). Questions and supporting articles are in English (see Figure 21).

Doc. Lang.	En					
Query. Lang.	**En**	**De**	**Es**	**Zh**	**Ar**	**Avg.**
GPT-4o	62.40	55.90	56.80	54.70	54.70	55.50
Claude 3.5	42.80	37.40	40.10	37.60	38.50	38.40
Mistral-large	43.30	36.50	39.50	30.60	18.90	31.40
Command-R+	45.70	39.80	41.20	34.30	33.80	37.30
Nova-Pro	54.00	44.80	49.30	37.30	34.30	41.43

Table 4: LLM QA accuracy in the XRAG *monolingual retrieval* setting. Grounding documents consist of two supporting articles and six distracting articles, all in English (see Figure 23). QA accuracy with English queries provides a monolingual RAG baseline for comparison.

Performance Upper Bounds. To establish a human upper bound on our dataset, we hire human annotators to answer 200 English questions from the *monolingual retrieval* setting (see English Q&A in Figure 2) by carefully reading the article pairs used to create the questions. Their performance is evaluated at 85% by the LLM judge panel, which is much higher than that of the best LLM.[6] This shows that even without the cross-lingual challenge, the dataset is a strong benchmark for LLM reasoning. A manual review of answers judged incorrect by the automated evaluator (see Appendix B.4 for more details on this manual review) finds that 2% are actually correct, 5% are wrong with gold answers being correct, and 8% involve noisy or ambiguous questions. This sets two separate upper bounds: 85% for human performance, and 92% allowing for noisy questions.

5.3 XRAG in Monolingual Retrieval Setting

We first benchmark LLMs in cross-lingual RAG with the *monolingual retrieval* setting of XRAG. To highlight the cross-lingual challenges, we compare results with an English monolingual RAG baseline, where the input question, supporting articles, and distracting articles are all in English. Grounding

[6] N.B.: QA pairs are generated by GPT-4o, and evaluation may be biased in its favor.

Doc. Lang. Query. Lang.	En+En$_{\text{De}}$ En	En+De De	En+En$_{\text{Es}}$ En	En+Es Es	En+En$_{\text{Zh}}$ En	En+Zh Zh	En+En$_{\text{Ar}}$ En	En+Ar Ar	Avg. (crossling.)
GPT-4o	63.33	61.67	59.33	56.00	63.00	59.33	60.67	53.33	57.58
Claude 3.5	51.00	45.67	46.33	42.67	46.67	48.00	47.33	39.67	44.00
Mistral-large	45.67	42.00	43.00	39.33	48.67	37.33	43.67	32.00	37.67
Command-R+	43.67	40.00	42.33	40.33	49.67	36.33	43.33	32.00	37.17
Nova-Pro	56.33	53.00	49.67	45.33	57.67	49.33	57.33	44.67	48.08

Table 5: LLM QA accuracy in the XRAG *multilingual retrieval* setting, which for each language **X** consists of a set of questions each accompanied by a supporting document and three distracting documents in language **X** and the same again in English. **En$_X$** refers to English translations of documents from language **X** using Google Translate (see Figure 24). **En+En$_X$** is a monolingual retrieval baseline setting for the language pair **En+X**.

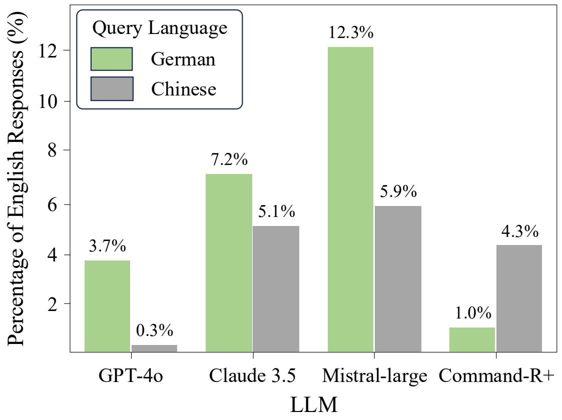


Figure 5: Percentage of instances in cross-lingual RAG with *monolingual retrieval* (English documents) where LLMs respond in English instead of the German or Chinese question language.

articles in the *monolingual retrieval* setting are already in English so this baseline experiment simply replaces the translated question with its original English (see English Q&A in Figure 2).

Table 4 shows the results as assessed by the LLM judge panel. XRAG in the *monolingual retrieval* setting poses a significant challenge for LLMs, with all models performing poorly. Among them, GPT-4o achieves the highest average accuracy at 55.50%,[6] while others score considerably lower, ranging from 31.4% to 41.43%. The cross-lingual capabilities of LLMs vary across languages. Compared to the English monolingual RAG baseline, all models experience a performance drop when answering non-English questions, but the severity of this drop differs. GPT-4o and Claude exhibit the smallest and most consistent declines across languages, suggesting more robust multilingual handling. Command-R+ and Mistral show larger variability, indicating potential language-specific weaknesses, particularly for Mistral, which suffers a 56.3% relative drop in Arabic.

Surprisingly, we find that **LLMs have issues with Response Language Correctness** (RLC), i.e., they respond in English instead of the question language. Figure 5 lists the percentage of cases in the *monolingual retrieval* setting where LLMs respond in the wrong language. GPT-4o and Command-R+ produce the fewest RLC errors, while Mistral-large is most affected.

5.4 XRAG in Multilingual Retrieval Setting

We now benchmark LLMs in cross-lingual RAG with the *multilingual retrieval* setting of XRAG. As with *monolingual retrieval*, we construct an English monolingual RAG setting for comparison. We replace the original non-English questions with their English counterparts (questions before human translation; see English Q&A in Figure 2). We also translate non-English supporting and distracting articles into English using Google Translate.

Table 5 presents LLM performance in cross-lingual QA in the *multilingual retrieval* setting of XRAG. All models exhibit poor performance in this cross-lingual scenario, with GPT-4o having the highest average accuracy at 57.58% and Command-R+ the lowest at 37.17%. LLMs also show accuracy degradations relative to their corresponding English monolingual RAG baseline, despite the latter being constructed with the assistance of machine translation.

To identify the most challenging aspect of the cross-lingual RAG with *multilingual retrieval*, we conduct a controlled analysis by gradual conversion to the English monolingual RAG setting. Specifically, we successively replace the question, supporting articles, and distracting articles by their English counterparts from the monolingual RAG baseline. Table 6 shows the results of GPT-4o. Changing the question (and expected answer) language from non-English to English only brings a relatively small average improvement, suggesting that **non-English generation may not be the core**

GPT-4o	**En+De**	**En+Es**	**En+Zh**	**En+Ar**	**Avg.**
XRAG-MultiR	61.67	56.00	59.33	53.33	57.58
+EQ	63.00	52.67	60.67	56.67	58.25
+EQ, +ES	65.00	57.33	62.00	60.33	61.16
MonoRAG	63.33	59.33	63.00	60.67	61.58

Table 6: Controlled analysis of GPT-4o on the *multilingual retrieval* setting of XRAG, replacing questions (EQ), supporting articles (ES), and distracting articles (ED) with their English counterparts from the English monolingual RAG settings (see Figure 25). "XRAG-MultiR" is the *multilingual retrieval* setting, and "MonoRAG" (+EQ, +ES, +ED) is the English monolingual RAG baseline setting.

challenge.[7] By contrast, replacing non-English supporting articles with their English translations improves average accuracy noticeably, indicating that **reasoning over retrieved information across languages is challenging for LLMs**. Translating distracting articles into English also improves performance, implying that identifying useful information in a mixed-language context is harder than from a wholly English one. Similar results are observed with other LLMs, see Appendix D.1.

6 Conclusions

We introduce XRAG, a benchmark for evaluating the generation abilities of LLMs in cross-lingual RAG settings. We introduce a novel LLM-based workflow for creating questions that require complex reasoning and external documents to answer. Experiments reveal that LLMs significantly underperform humans on XRAG, even without cross-lingual elements, highlighting its utility for assessing reasoning ability. Further analysis shows that LLMs struggle with response language correctness in the XRAG *monolingual retrieval* setting and with reasoning over retrieved content across languages in the XRAG *multilingual retrieval* setting.

7 Limitations

Our work has some limitations. First, due to budget constraints, XRAG currently covers only four non-English languages. However, we believe that these four linguistically diverse languages are sufficient to enable robust cross-lingual RAG research. Moreover, we provide comprehensive implementation details, allowing others to extend our approach to additional languages easily. Second, our multilingual retrieval setting solely covers the scenario of two languages (English and the question language). However, there may be cases involving retrieval across a set of languages (more than two). Note that our construction pipeline can support exploration in this setup by using more articles to generate questions, which we leave for future work. Third, we only benchmarked five models in this work because of legal concerns. It would be interesting to see how other LLMs perform on XRAG. Finally, we could conduct more insightful controlled analyses on XRAG, such as exploring the impact of the number of distracting articles. Due to space limitations, we leave this for future work.

Acknowledgements

Bill Byrne holds concurrent appointments as an Amazon Scholar and as Professor of Information Engineering at the University of Cambridge. This paper describes work performed at Amazon.

We thank Tobias Domhan, Rexhina Blloshmi, and Ke Tran for their valuable comments during the early stages of this project. We are also grateful to Tobias Domhan and Zach Hille for their guidance on using Amazon's internal platform for efficient LLM inference, which was essential for the completion of this project. Finally, we would like to thank the anonymous reviewers for their constructive feedback.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore. Association for Computational Linguistics.

Amazon. 2024. The amazon nova family of models: Technical report and model card.

Anthropic. 2024. Claude sonnet-3.5: Next-generation ai model.

Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021. XOR QA: Cross-lingual open-retrieval question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*,

[7] We also find that LLMs rarely have issues with Response Language Correctness in the *multilingual retrieval* setting.

pages 547–564, Online. Association for Computational Linguistics.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 2318–2335, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024b. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 17754–17762.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024c. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Nadezhda Chirkova, David Rau, Hervé Déjean, Thibault Formal, Stéphane Clinchant, and Vassilina Nikoulina. 2024. 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.

Cohere. 2024. Introducing command-r+: Advanced retrieval-augmented generation model.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* Just Accepted.

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

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.

Jungo Kasai, Keisuke Sakaguchi, yoichi takahashi, Ronan Le Bras, Akari Asai, Xinyan Velocity Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2023. Realtime QA: What's the answer right now? In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Bryan Li, Samar Haider, Fiona Luo, Adwait Agashe, and Chris Callison-Burch. 2024. BordIRlines: A dataset for evaluating cross-lingual retrieval augmented generation. In *Proceedings of the First Workshop on Advancing Natural Language Processing for Wikipedia*, pages 1–13, Miami, Florida, USA. Association for Computational Linguistics.

Tian Liang, Xing Wang, Mingming Yang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Addressing entity translation problem via translation difficulty and context diversity. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11628–11638, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2019a. XQA: A cross-lingual open-domain question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2358–2368, Florence, Italy. Association for Computational Linguistics.

Wei Liu, Yi Fan, and Michael Strube. 2023. HITS at DISRPT 2023: Discourse segmentation, connective detection, and relation classification. In *Proceedings of the 3rd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2023)*, pages 43–49, Toronto, Canada. The Association for Computational Linguistics.

Wei Liu, Xiyan Fu, Yue Zhang, and Wenming Xiao. 2021. Lexicon enhanced Chinese sequence labeling using BERT adapter. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5847–5858, Online. Association for Computational Linguistics.

Wei Liu, Tongge Xu, Qinghua Xu, Jiayu Song, and Yueran Zu. 2019b. An encoding strategy based word-character LSTM for Chinese NER. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2379–2389, Minneapolis, Minnesota. Association for Computational Linguistics.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987, Singapore. Association for Computational Linguistics.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. SemEval-2022 task 11: Multilingual complex named entity recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1412–1437, Seattle, United States. Association for Computational Linguistics.

Sebastian Nagel. 2016. News dataset available. Accessed: 4 October 2016.

OpenAI. 2024. Introducing gpt-4o.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. LLM evaluators recognize and favor their own generations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nirmal Roy, Leonardo F. R. Ribeiro, Rexhina Blloshmi, and Kevin Small. 2024. Learning when to retrieve, what to rewrite, and how to respond in conversational QA. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10604–10625, Miami, Florida, USA. Association for Computational Linguistics.

Uma Roy, Noah Constant, Rami Al-Rfou, Aditya Barua, Aaron Phillips, and Yinfei Yang. 2020. LAReQA: Language-agnostic answer retrieval from a multilingual pool. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5919–5930, Online. Association for Computational Linguistics.

Yixuan Tang and Yi Yang. 2024. MultiHop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries. In *First Conference on Language Modeling*.

Nandan Thakur, Luiz Bonifacio, Crystina Zhang, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Boxing Chen, Mehdi Rezagholizadeh, and Jimmy Lin. 2024a. “knowing when you don‘t know”: A multilingual relevance assessment dataset for robust retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12508–12526, Miami, Florida, USA. Association for Computational Linguistics.

Nandan Thakur, Suleman Kazi, Ge Luo, Jimmy Lin, and Amin Ahmad. 2024b. Miragebench: Automatic multilingual benchmark arena for retrieval-augmented generation systems. *Preprint*, arXiv:2410.13716.

Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors. 2021. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. 2024. FreshLLMs: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13697–13720, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Jiaan Wang, Yunlong Liang, Fandong Meng, Beiqi Zou, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2023. Zero-shot cross-lingual summarization via large language models. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 12–23, Singapore. Association for Computational Linguistics.

Jiaan Wang, Fandong Meng, Duo Zheng, Yunlong Liang, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2022. A survey on cross-lingual summarization. *Transactions of the Association for Computational Linguistics*, 10:1304–1323.

Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, Kun Yu, Yuxing Yuan, Yinghao Zou, Jiquan Long, Yudong Cai, Zhenxiang Li, Zhifeng Zhang, Yihua Mo, Jun Gu, Ruiyi Jiang, Yi Wei, and Charles Xie. 2021. Milvus: A purpose-built vector data management system. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD ’21, page 2614–2627, New York, NY, USA. Association for Computing Machinery.

Shuting Wang, Jiongnan Liu, Shiren Song, Jiehan Cheng, Yuqi Fu, Peidong Guo, Kun Fang, Yutao Zhu, and Zhicheng Dou. 2024. Domainrag: A chinese benchmark for evaluating domain-specific retrieval-augmented generation. *Preprint*, arXiv:2406.05654.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Jian Yang, Shaohan Huang, Shuming Ma, Yuwei Yin, Li Dong, Dongdong Zhang, Hongcheng Guo, Zhoujun Li, and Furu Wei. 2022. CROP: Zero-shot cross-lingual named entity recognition with multilingual labeled sequence translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 486–496, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Gui, Ziran Jiang, Ziyu JIANG, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas SCHEFFER, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen tau Yih, and Xin Luna Dong. 2024. CRAG - comprehensive RAG benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Kunlun Zhu, Yifan Luo, Dingling Xu, Yukun Yan, Zhenghao Liu, Shi Yu, Ruobing Wang, Shuo Wang, Yishan Li, Nan Zhang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2025. RAGEval: Scenario specific RAG evaluation dataset generation framework. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8520–8544, Vienna, Austria. Association for Computational Linguistics.

User Locale	Native	English	Both
Germany	59.8%	56.3%	71.5%
Japan	61.1%	44.3%	68.3%
Spain	57.9%	48.4%	68.9%

Table 7: Percentage of satisfactory retrieval results using Google search as retriever on real-world information seeking LLM traffic. Search results are evaluated by Claude 3.5 Sonnet if they contain sufficient information for a satisfactory answer to the user query. *Native* uses Google search in the user locale, *English* performs English Google search using a translation of the user query, and *Both* combines the two search results. LLM traffic is obtained via SimilarWeb.

A Retrieval Quality Investigation

We evaluate the relevance of English and native-language search results for real-world queries submitted to large language models (LLMs) by non-English users in Germany, Spain, and Japan. The analysis is based on a proprietary dataset of real LLM traffic provided by SimilarWeb[8]. User queries are filtered by locale and language using langid to retain only those submitted in the respective native languages. For each query, we retrieve two sets of search results: one from a U.S.-based Google search using the English translation of the query, and another from a country-specific Google domain using the original native-language version. Relevance is assessed using Claude 3.5 Sonnet, which evaluates whether the retrieved results contain sufficient information to generate a satisfactory response, taking into account the user's locale (e.g., a tax-related query from Germany must include references to German tax regulations). Table 7 reports the percentage of queries for which the English-only, native-only, or both sets of results independently provided sufficient information. The results indicate that combining English and native-language search results significantly improves the proportion of queries for which a comprehensive response can be generated, compared to using either language alone.

B Data Construction

B.1 Data Source

We use articles in News Crawl (Nagel, 2016) as the data source to create questions and prepare supporting and distracting articles. Specifically, we download news articles between June 1, 2024 and November 30, 2024 from NEWS Crawl using newplease[9]. This timeframe exceeds the knowledge cutoff of widely used LLMs, such as GPT-4o and Claude 3.5 sonnet. Therefore, questions created from these articles are more likely to require LLMs to use external knowledge to answer. We only keep articles that contain more than 1200 tokens and are in English, German, Spanish, Chinese, or Arabic, obtaining around 1700k, 250k, 600k, 180k, and 460k news articles in these languages.

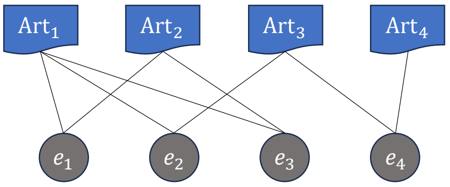


Figure 6: Example of a bipartite graph between articles and entities.

B.2 Identify Pairs of Related Articles

B.2.1 Identify English-English Article pairs

Inspired by the concept of "bridge entity" in Yang et al. (2018), we use a bipartite graph between articles and entities to find related English article pairs. Specifically, we randomly sample around 100k English articles from the data downloaded from News Crawl, covering various topics such as Politics, Sports, Economy, and Entertainment. Then, we use stanza (Qi et al., 2020) to identify entities in the titles of the sampled articles and construct a bipartite graph between the articles and entities (as shown in Figure 6). In this graph, nodes are entities and articles, and an edge will be added between an entity and an article if the entity is contained in the title of the article. Finally, we perform the Depth-First Search on the graph to find pairs of articles sharing at least two entities in their titles (e.g., articles 1 and 2 in Figure 6) and having a publication time gap of no more than two weeks. Here we use the title instead of the main text because entities in the title are often the key entities of the news, and two articles containing the same key entities are more likely to be related.

B.2.2 Identify X-English Article Pairs

To construct cross-document questions for the *multilingual retrieval* setting in XRAG, we need to find related articles across languages. We empirically find that the article-entity graph performs

[8] https://www.similarweb.com/

[9] https://github.com/fhamborg/news-please

Event	Language	Topic
2024 presidential election 2024	English	Politics
Halbfinale der UEFA EURO 2024 (en: semi-finals of UEFA EURO 2024)	German	Sports
Inundaciones repentinas en Valencia (en: Flash floods in Valencia)	Spanish	Disaster
秘鲁钱凯港开港 (en: Opening of the Port of Chancay, Peru)	Chinese	Economy
وقف إطلاق النار في قطاع غزة (en: Ceasefire in the Gaza Strip)	Arabic	Politics

Figure 7: Examples of events collected from Wiki 2024 between June and November, which will be used to retrieve related articles across languages.

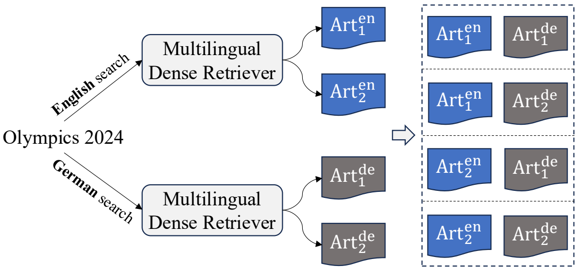

English search
German search
Olympics 2024
Multilingual Dense Retriever
Multilingual Dense Retriever
Art$_1^{en}$
Art$_2^{en}$
Art$_1^{de}$
Art$_2^{de}$
Art$_1^{en}$ Art$_1^{de}$
Art$_1^{en}$ Art$_2^{de}$
Art$_2^{en}$ Art$_1^{de}$
Art$_2^{en}$ Art$_2^{de}$

Figure 8: Example of using the event "Olympics 2024" and a multilingual retrieve to find related articles in English and German.

Topic relevance verification

You are an AI assistant tasked with verifying whether the given two texts are topically related. Considering the following steps to generate your response:
1. Read the two texts carefully and get the "high level" topics.
2. Compare whether the two texts are about related entities.
3. Compare whether the high level topics of the two texts are related (e.g., both describe stock price, sport, film, a country, a party, or the same event).
4. Do not pay much attention to details but high level topics!
5. Create your response.

Here are the two given texts:
<text1> {{ text1 }} </text1>
<text2> {{ text2 }} </text2>

If they are topically related, format your response as: <answer>yes</answer>
otherwise, output: <answer>no</answer>

Figure 9: Prompt for topic relevance verification.

poorly here, due to (1) inaccurate entity recognition in non-English texts (Liu et al., 2019b, 2021; Malmasi et al., 2022), and (2) challenges in cross-lingual entity mapping (Liang et al., 2024).

To address this, we collect 117 international events between June and November 2024 from different language versions of Wiki 2024[10], covering topics such as politics, sports, astronomy, and natural disasters (see some examples in Figure 7). We also build a multilingual dense retriever[11] based on the multilingual model BGE-M3 (as text encoder, Chen et al., 2024a) and Milvus (as vector database, Wang et al., 2021). The retriever operates on news articles downloaded from NEWS Crawl. Then, we use the retriever to search across languages for articles related to the events. Finally, we group articles in different languages about the same event to form related article pairs. Figure 8 show an example, where we use the event "Olympics 2024" to search for English and German articles and create related English-German article pairs.

In both cases, we further use GPT-4o to verify the topical relevance of the found article pairs, passing only those confirmed to be truly related to the next step. Figure 9 shows the prompt we used to instruct GPT-4o for relevancy verification. Furthermore, if computational resources were not a constraint, a more generalized approach to identifying pairs of related articles would involve using randomly selected documents as queries and retrieving related pairs through a multilingual dense retriever. This method would be applicable across a wide range of domains and languages.

B.3 English cross-document Q&A Generation

Figure 10 shows the prompt used to generate a summary from an article, as described in Step 1 of Section 4.2. Figure 11 shows the prompt for generating a set of simple Q&A pairs from a summary, as described in Step 2 of Section 4.2. Table 8 shows the definition of four types of cross-document questions: aggregation, comparison, multi-hop, and set questions, as mentioned in Step 3 of Section 4.2. We show the prompts used for generating these four types of questions in Figures 12, 13, 14, and 15. The prompt used for generating an answers to the created cross-document question is presented in Figure 16.

B.4 Human Verification

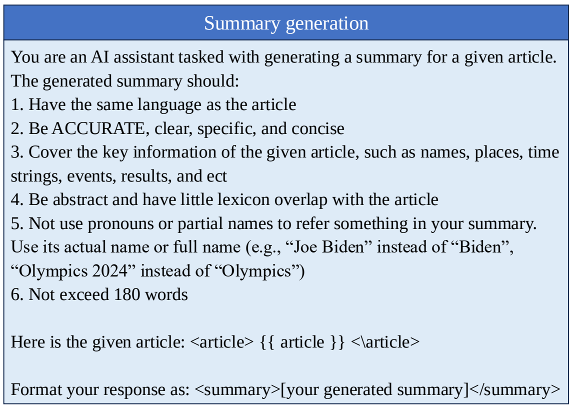
Due to the existence of LLM hallucinations (Huang et al., 2024), the Q&A pairs generated by the LLM may contain factual errors. Therefore, we ask a professional annotation team to verify the quality of the generated Q&A pairs. Figure 17 presents the guidelines we prepared for the annotation team. Each language had 4–6 annotators with verified

[10] https://en.wikipedia.org/wiki/2024
https://de.wikipedia.org/wiki/2024
https://es.wikipedia.org/wiki/2024
https://zh.wikipedia.org/zh-cn/2024
https://ar.wikipedia.org/wiki/2024

[11] The dense retriever is also used in selecting distracting articles for each question, see Section 4.4.

Quesiton type	Definition
Aggregation	Questions that require the aggregation of information across articles to answer (e.g., “how many Oscar awards did Meryl Streep win?”)
Comparison	Questions that compare information in two articles (e.g., “who started performing earlier, Adele or Ed Sheeran?”)
Multi-hop	Questions that require chaining multiple pieces of information from two articles to compose the answer (e.g., “who acted in Ang Lee’s latest movie?”)
Set	Questions that expect a set of entities, objects, or events from two articles as the answer (e.g., “what are the continents in the southern hemisphere?”)

Table 8: Definition of four types of cross-document questions: aggregation, comparison, multi-hop, and set questions, as mentioned in Step 3 of Section 4.2.



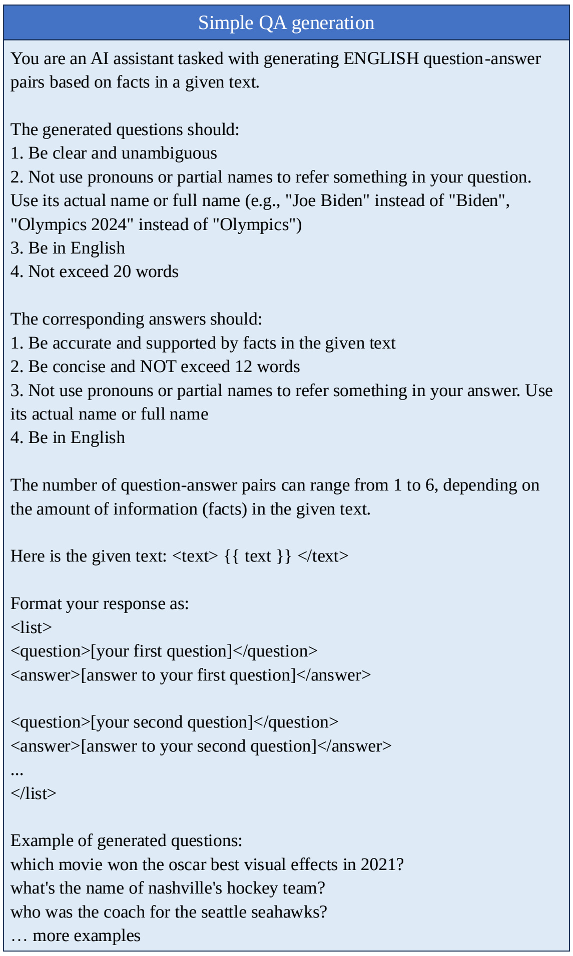
Summary generation

You are an AI assistant tasked with generating a summary for a given article. The generated summary should:
1. Have the same language as the article
2. Be ACCURATE, clear, specific, and concise
3. Cover the key information of the given article, such as names, places, time strings, events, results, and ect
4. Be abstract and have little lexicon overlap with the article
5. Not use pronouns or partial names to refer something in your summary. Use its actual name or full name (e.g., “Joe Biden” instead of “Biden”, “Olympics 2024” instead of “Olympics”)
6. Not exceed 180 words

Here is the given article: <article> {{ article }} <\article>

Format your response as: <summary>[your generated summary]</summary>

Figure 10: Prompt for summary generation, as described in Step 1 of Section 4.2.



Simple QA generation

You are an AI assistant tasked with generating ENGLISH question-answer pairs based on facts in a given text.

The generated questions should:
1. Be clear and unambiguous
2. Not use pronouns or partial names to refer something in your question. Use its actual name or full name (e.g., "Joe Biden" instead of "Biden", "Olympics 2024" instead of "Olympics")
3. Be in English
4. Not exceed 20 words

The corresponding answers should:
1. Be accurate and supported by facts in the given text
2. Be concise and NOT exceed 12 words
3. Not use pronouns or partial names to refer something in your answer. Use its actual name or full name
4. Be in English

The number of question-answer pairs can range from 1 to 6, depending on the amount of information (facts) in the given text.

Here is the given text: <text> {{ text }} </text>

Format your response as:
<list>
<question>[your first question]</question>
<answer>[answer to your first question]</answer>

<question>[your second question]</question>
<answer>[answer to your second question]</answer>
...
</list>

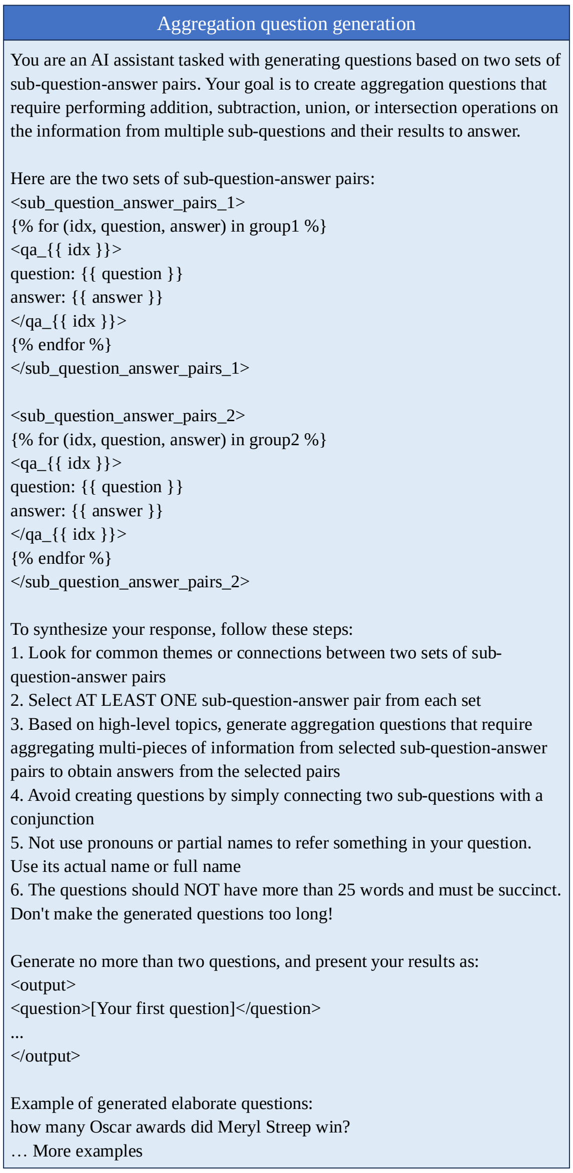
Example of generated questions:
which movie won the oscar best visual effects in 2021?
what's the name of nashville's hockey team?
who was the coach for the seattle seahawks?
… more examples

Figure 11: Prompt for simple Q&A generation, as described in Step 2 of Section 4.2.

bilingual proficiency. Annotators were trained on 150 pilot examples with iterative feedback. Final inter-annotator agreement (on other 200 examples) yielded Cohen’s kappa coefficient of 0.6182, indicating substantial agreement.

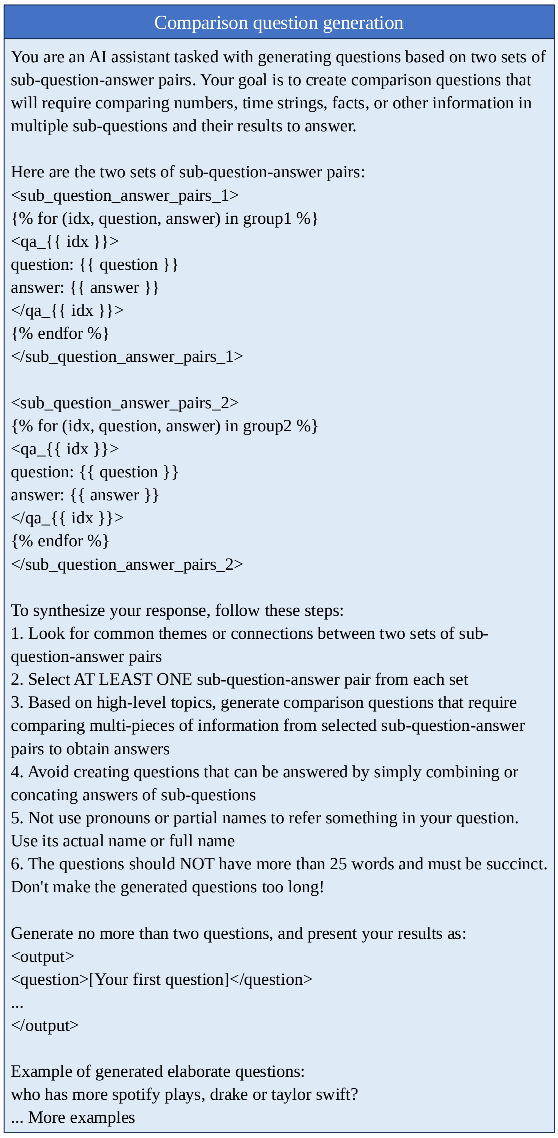
We generate 2,950 raw cross-document Q&A pairs from English article pairs. Following manual verification, approximately 90% of the questions are deemed natural, 72% are considered answerable, and 54% of the answers (including some that were manually corrected) are judged correct, yielding a final set of approximately 1,500 verified Q&A pairs. To further assess quality, we sample 200 Q&A pairs and task a separate group of annotators to answer the questions by carefully reading the article pairs used to generate these questions. Annotators are explicitly instructed to answer no more than one question per hour (to ensure thorough reading and accurate responses). If their answers align with the reference answers, it suggests high-quality Q&A pairs. We use the majority vote of three LLM-as-a-Judge (see Evaluation Metrics in Section 5.1) to calculate the accuracy of human responses to 200 questions, resulting in 85%. Upon manual review of the 30 failed cases, we find: (i) 4 human answers are correct but incorrectly judged by the LLMs; (ii) 10 are genuinely incorrect, with the reference answers being valid; and (iii) 16 are difficult to assess due to ambiguity or poor question quality, and are thus categorized as low-quality examples. This performance is comparable to established QA benchmarks, for instance, human accuracy on SQuAD (Rajpurkar et al., 2016) is 86.8%. We then randomly sample 1000 examples from the 1500 Q&A pairs and send them for human translation.

We generate approximately 1,000 Q&A pairs each from article pairs of the following language pairs: English–German, English–Spanish, En-



```
Aggregation question generation

You are an AI assistant tasked with generating questions based on two sets of
sub-question-answer pairs. Your goal is to create aggregation questions that
require performing addition, subtraction, union, or intersection operations on
the information from multiple sub-questions and their results to answer.

Here are the two sets of sub-question-answer pairs:
<sub_question_answer_pairs_1>
{% for (idx, question, answer) in group1 %}
<qa_{{ idx }}>
question: {{ question }}
answer: {{ answer }}
</qa_{{ idx }}>
{% endfor %}
</sub_question_answer_pairs_1>

<sub_question_answer_pairs_2>
{% for (idx, question, answer) in group2 %}
<qa_{{ idx }}>
question: {{ question }}
answer: {{ answer }}
</qa_{{ idx }}>
{% endfor %}
</sub_question_answer_pairs_2>

To synthesize your response, follow these steps:
1. Look for common themes or connections between two sets of sub-
question-answer pairs
2. Select AT LEAST ONE sub-question-answer pair from each set
3. Based on high-level topics, generate aggregation questions that require
aggregating multi-pieces of information from selected sub-question-answer
pairs to obtain answers from the selected pairs
4. Avoid creating questions by simply connecting two sub-questions with a
conjunction
5. Not use pronouns or partial names to refer something in your question.
Use its actual name or full name
6. The questions should NOT have more than 25 words and must be succinct.
Don't make the generated questions too long!

Generate no more than two questions, and present your results as:
<output>
<question>[Your first question]</question>
...
</output>

Example of generated elaborate questions:
how many Oscar awards did Meryl Streep win?
… More examples
```

Figure 12: Prompt for aggregation question generation, as described in Step 3 of Section 4.2.



```
Comparison question generation

You are an AI assistant tasked with generating questions based on two sets of
sub-question-answer pairs. Your goal is to create comparison questions that
will require comparing numbers, time strings, facts, or other information in
multiple sub-questions and their results to answer.

Here are the two sets of sub-question-answer pairs:
<sub_question_answer_pairs_1>
{% for (idx, question, answer) in group1 %}
<qa_{{ idx }}>
question: {{ question }}
answer: {{ answer }}
</qa_{{ idx }}>
{% endfor %}
</sub_question_answer_pairs_1>

<sub_question_answer_pairs_2>
{% for (idx, question, answer) in group2 %}
<qa_{{ idx }}>
question: {{ question }}
answer: {{ answer }}
</qa_{{ idx }}>
{% endfor %}
</sub_question_answer_pairs_2>

To synthesize your response, follow these steps:
1. Look for common themes or connections between two sets of sub-
question-answer pairs
2. Select AT LEAST ONE sub-question-answer pair from each set
3. Based on high-level topics, generate comparison questions that require
comparing multi-pieces of information from selected sub-question-answer
pairs to obtain answers
4. Avoid creating questions that can be answered by simply combining or
concating answers of sub-questions
5. Not use pronouns or partial names to refer something in your question.
Use its actual name or full name
6. The questions should NOT have more than 25 words and must be succinct.
Don't make the generated questions too long!

Generate no more than two questions, and present your results as:
<output>
<question>[Your first question]</question>
...
</output>

Example of generated elaborate questions:
who has more spotify plays, drake or taylor swift?
... More examples
```

Figure 13: Prompt for comparison question generation, as described in Step 3 of Section 4.2.

glish–Chinese, and English–Arabic. After manual verification, about 90% of the questions from the English–German and English–Spanish sets are deemed natural, while the proportion for English–Chinese and English–Arabic is slightly lower, at approximately 84%. Finally, we obtain 487, 680, 332, and 420 high-quality Q&A pairs from the English–German, English–Spanish, English–Chinese, and English–Arabic article pairs, respectively. From each set, 300 Q&A pairs are randomly sampled and submitted for translation.

B.5 Human Translation

Q&A pairs generated from English-English article pairs are translated into German, Spanish, Chinese, and Arabic to simulate a cross-lingual retrieval-augmented generation (RAG) with *monolingual retrieval*. Given the importance of named entities in Q&A, translators are instructed to consult Wikipedia or other reliable sources to find commonly used translations in the target language. If no appropriate translation exists, the original English term is retained. For example, “Microsoft updated the Copilot” is translated into Chinese as “微软更新了Copilot,” where “Microsoft” is translated (微软) and “Copilot” remains in English due to the absence of a standard Chinese equivalent.

For Q&A pairs generated from article pairs in different languages, such as English-German and English-Chinese, translation is performed only into the language of the non-English input article. For example, Q&A pairs from English–German article

```
Multihop question generation

You are an AI assistant tasked with generating questions based on two sets of sub-
question-answer pairs. Your goal is to create multi-hop questions that will require
chaining multiple pieces of information from multiple sub-questions and their
results to answer.

Here are the two sets of sub-question-answer pairs:
<sub_question_answer_pairs_1>
{% for (idx, question, answer) in group1 %}
<qa_{{ idx }}>
question: {{ question }}
answer: {{ answer }}
</qa_{{ idx }}>
{% endfor %}
</sub_question_answer_pairs_1>

<sub_question_answer_pairs_2>
{% for (idx, question, answer) in group2 %}
<qa_{{ idx }}>
question: {{ question }}
answer: {{ answer }}
</qa_{{ idx }}>
{% endfor %}
</sub_question_answer_pairs_2>

To synthesize your response, follow these steps:
1. Look for common themes or connections between two sets of sub-question-
answer pairs
2. Select AT LEAST ONE sub-question-answer pair from each set
3. Based on high-level topics, generate multi-hop questions which need reasoning
over multi-pieces of information from selected sub-question-answer to obtain
answers
4. Avoid creating questions that can be answered by simply combining or concating
answers of sub-questions.
5. Not use pronouns or partial names to refer something in your question. Use its
actual name or full name.
6. The questions should NOT have more than 25 words and must be succinct. Don't
make the generated questions too long!

Generate no more than two questions, and present your results as:
<output>
<question>[Your first question]</question>
...
</output>

Example of generated elaborate questions:
where did the ceo of salesforce previously work?
... More examples
```

Figure 14: Prompt for multi-hop question generation, as described in Step 3 of Section 4.2.

pairs are translated into German, and so on for others. These examples are used to simulate a cross-lingual RAG with *multilingual retrieval*.

For each language in the XRAG dataset, the cost is about $8,000 for quality assurance and $12,000 for professional translation.

C Experimental Settings

C.1 Models

We benchmark five models on XRAG, including GPT-4o-2024-08-06, Claude 3.5 Sonnet (2024-06-20), Mistral-Large-Instruct-2407, Command-r+, and Nova-pro. Figure 18 shows the template we use to prompt LLMs to respond to a given answer by reading the retrieved articles. Figure 19 shows the template used to prompt LLMs to answer questions using their own parametric knowledge.

```
Set question generation

You are an AI assistant tasked with generating questions based on two sets of sub-
question-answer pairs. Your goal is to create "set" questions that expect a set of entities
or objects of the same category as answers.

Here are the two sets of sub-question-answer pairs:
<sub_question_answer_pairs_1>
{% for (idx, question, answer) in group1 %}
<qa_{{ idx }}>
question: {{ question }}
answer: {{ answer }}
</qa_{{ idx }}>
{% endfor %}
</sub_question_answer_pairs_1>

<sub_question_answer_pairs_2>
{% for (idx, question, answer) in group2 %}
<qa_{{ idx }}>
question: {{ question }}
answer: {{ answer }}
</qa_{{ idx }}>
{% endfor %}
</sub_question_answer_pairs_2>

To synthesize your response, follow these steps:
1. Look for common themes or connections between two sets of sub-question-answer
pairs
2. Select AT LEAST ONE sub-question-answer pair from each set
3. Based on high-level topics, generate "set" questions which need entities or objects of
same category from selected sub-question-answer pairs to obtain answers
4. Avoid creating questions by simply connecting two sub-questions with a conjunction
5. Not use pronouns or partial names to refer something in your question. Use its actual
name or full name
6. The questions should NOT have more than 25 words and must be succinct. Don't
make the generated questions too long!

Generate no more than two questions, and present your results as:
<output>
<question>[Your first question]</question>
...
</output>

Example of generated elaborate questions:
what are ben stiller 3 most recent movies?
... More examples
```

Figure 15: Prompt for set question generation, as described in Step 3 of Section 4.2.

C.2 Evaluation Metrics

We use LLM-as-a-Judge to determine whether an LLM's response is correct. Specifically, each time we input a question, a golden answer, and an answer generated by a model to an LLM and ask the LLM to determine whether the generated answer is correct or incorrect following the guideline we provided (see prompt in Figure 20). To avoid the *self-preference* problem (Panickssery et al., 2024), we use three LLM judges, including GPT-4o-2024-08-06, Claude Sonnet-3.5 (2024-06-20), and Mistral-Large-Instruct-2407, and take the majority vote as the final result. In the prompt, we explicitly instruct LLM judges to consider the language of models' responses, but they sometimes fail to do so. To address this, we apply a language detection tool, lingua[12], to verify whether the response is in the same language as the corresponding input question, and if not, we consider it incorrect. Finally, we

[12] https://github.com/pemistahl/lingua

Answer generation for cross-document question

```
You are an AI assistant tasked with generating an ENGLISH answer for a
given question. Your goal is to create an ENGLISH answer based solely on
the given sub-question-answer pairs, two reference texts, and their
publication dates.

The given question is: <question> {{ question }} </question>

Here are the given sub-question-answer pairs:
<sub_question_answer_pairs>
{% for (idx, question, answer) in group %}
<qa_{{ idx }}>
question: {{ question }}
answer: {{ answer }}
</qa_{{ idx }}>
{% endfor %}
</sub_question_answer_pairs>

Here are the two reference texts and their publication dates:
<text_1> {{ text1 }} </text_1>
<date_1> {{ date1 }} </date_1>

<text_2> {{ text2 }} </text_2>
<date_2> {{ date2 }} </date_2>


1. Look for connections between the given questions and sub-questions
2. Read answers of sub-questions and the two reference texts
3. Generate an ENGLISH answer based solely on those information. DO
NOT generate answers use information OUTSIDE of sub-question-answer
pairs and two reference texts, such as your own knowledge
4. Format your response as: <answer>[your answer]</answer>
```

Figure 16: Prompt for creating an answer for a generated cross-document question, as described in Step 3 of Section 4.2.

report each model's accuracy by the LLM judge panel, which includes the assessment of language correctness.

To assess the reliability of the LLM judge panel, we compare its evaluations with those provided by human annotators. Specifically, we collect responses from five different LLMs to 300 English questions in the *monolingual retrieval* setting, yielding a total of 1,500 responses. The LLM judge panel is then used to evaluate the correctness of each response against the gold answer. In parallel, we recruit three annotators via Amazon Mechanical Turk[13] to independently assess the same set of 1,500 responses. These annotators follow the same evaluation guidelines as those used by the LLM-as-a-Judge (see Figure 20). The majority vote among the three annotators is taken as the final human judgment. To quantify the level of agreement between the LLM judge panel and the human evaluators, we compute Cohen's kappa, which yields a score of 0.71, indicating substantial agreement between the two evaluation approaches.

Claude 3.5	En+De	En+Es	En+Zh	En+Ar	Avg.
XRAG-MultiR	45.67	42.67	48.00	39.67	44.00
+EQ	49.00	40.67	45.33	42.67	44.42
+EQ, ES	46.67	44.00	46.33	47.67	46.17
MonoRAG	51.00	46.33	46.67	47.33	47.83

Table 9: Controlled analysis of Claude Sonnet 3.5 on the *multilingual retrieval* setting of XRAG, replacing questions (EQ), supporting articles (ES), and distracting articles (ED) with their English counterparts from the English monolingual RAG settings (see Figure 25). "XRAG-MultiR" is the *multilingual retrieval* setting, and "MonoRAG" (+EQ, +ES, +ED) is the English monolingual RAG baseline setting.

Mistral-large	En+De	En+Es	En+Zh	En+Ar	Avg.
XRAG-MultiR	42.00	39.33	37.33	32.00	37.67
+EQ	42.67	34.67	40.67	38.00	39.00
+EQ, ES	43.33	40.00	47.33	45.33	44.00
MonoRAG	45.67	43.00	48.67	43.67	45.25

Table 10: Controlled analysis of Mistral-large on the *multilingual retrieval* setting of XRAG, replacing questions (EQ), supporting articles (ES), and distracting articles (ED) with their English counterparts from the English monolingual RAG settings (see Figure 25).

Command-R+	En+De	En+Es	En+Zh	En+Ar	Avg.
XRAG-MultiR	40.00	40.33	36.33	32.00	37.17
+EQ	41.00	34.00	40.67	40.33	39.00
+EQ, ES	43.33	36.00	50.33	43.33	43.25
MonoRAG	43.67	42.33	49.67	43.33	44.75

Table 11: Controlled analysis of Command-R+ on the *multilingual retrieval* setting of XRAG, replacing questions (EQ), supporting articles (ES), and distracting articles (ED) with their English counterparts from the English monolingual RAG settings (see Figure 25).

C.3 Settings in Different Tables

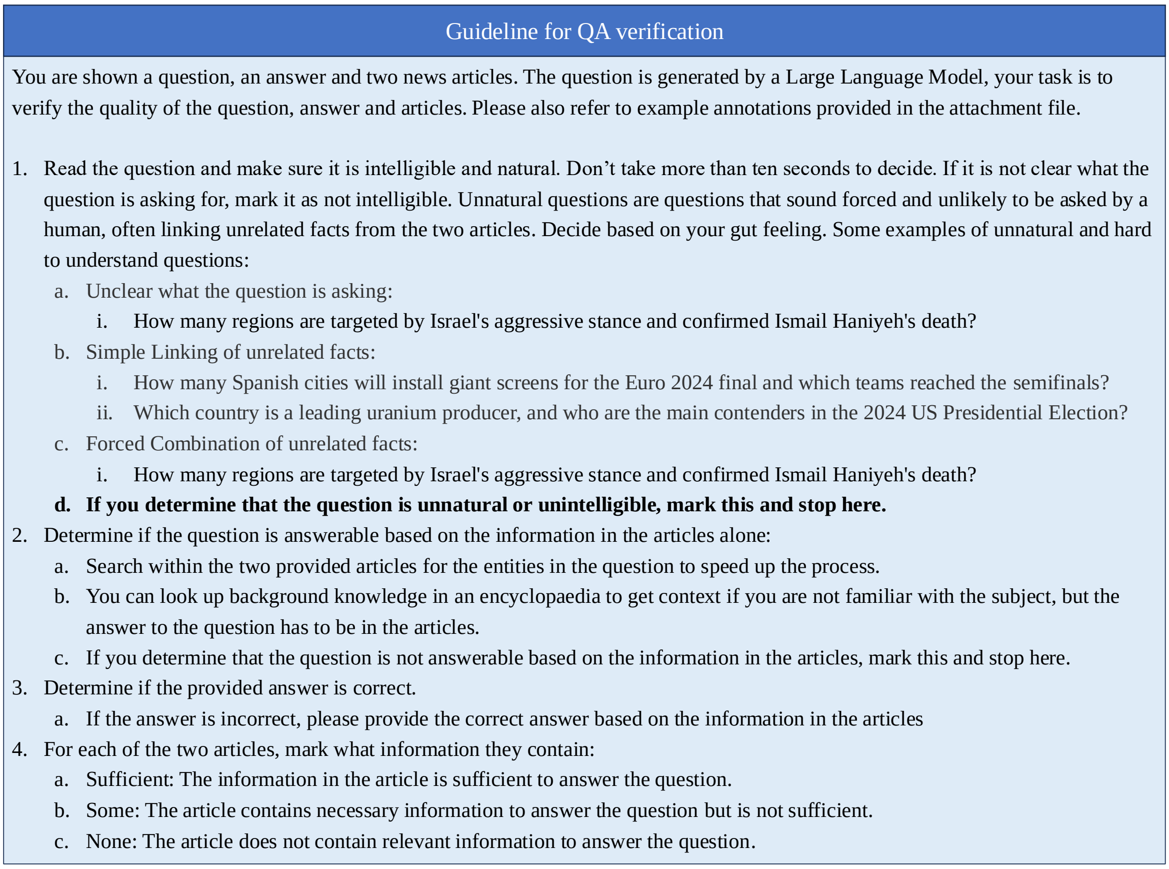
To facilitate the interpretation of the results presented in different tables, we provide diagrams illustrating the corresponding experimental setups. Specifically, Figures 21, 23, 24, and 25 depict the experimental configurations associated with Tables 3, 4, 5, and 6, respectively.

D More Experimental Results

D.1 Controlled Analysis

In Section 5.4, we perform a controlled analysis on the language of query, distracting articles, and supporting articles in the cross-lingual RAG with *multilingual retrieval*, using GPT-4o as the primary model. Here, we extend the analysis to additional LLMs: results for Claude 3.5 Sonnet are shown in Table 9, Mistral-large in Table 10, and Command-R+ in Table 11. Similar results are ob-

[13] Given the straightforward nature of the evaluation task, we opted to use Mechanical Turk instead of a professional annotation team.

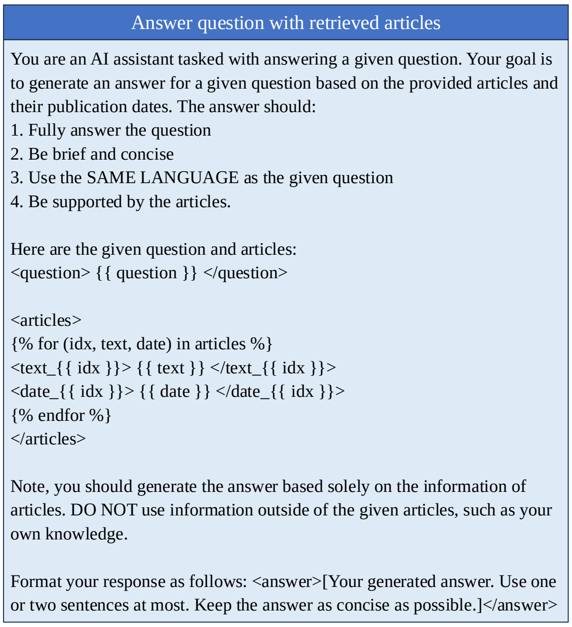


Guideline for QA verification

You are shown a question, an answer and two news articles. The question is generated by a Large Language Model, your task is to verify the quality of the question, answer and articles. Please also refer to example annotations provided in the attachment file.

1. Read the question and make sure it is intelligible and natural. Don't take more than ten seconds to decide. If it is not clear what the question is asking for, mark it as not intelligible. Unnatural questions are questions that sound forced and unlikely to be asked by a human, often linking unrelated facts from the two articles. Decide based on your gut feeling. Some examples of unnatural and hard to understand questions:
   a. Unclear what the question is asking:
      i. How many regions are targeted by Israel's aggressive stance and confirmed Ismail Haniyeh's death?
   b. Simple Linking of unrelated facts:
      i. How many Spanish cities will install giant screens for the Euro 2024 final and which teams reached the semifinals?
      ii. Which country is a leading uranium producer, and who are the main contenders in the 2024 US Presidential Election?
   c. Forced Combination of unrelated facts:
      i. How many regions are targeted by Israel's aggressive stance and confirmed Ismail Haniyeh's death?
   d. If you determine that the question is unnatural or unintelligible, mark this and stop here.
2. Determine if the question is answerable based on the information in the articles alone:
   a. Search within the two provided articles for the entities in the question to speed up the process.
   b. You can look up background knowledge in an encyclopaedia to get context if you are not familiar with the subject, but the answer to the question has to be in the articles.
   c. If you determine that the question is not answerable based on the information in the articles, mark this and stop here.
3. Determine if the provided answer is correct.
   a. If the answer is incorrect, please provide the correct answer based on the information in the articles
4. For each of the two articles, mark what information they contain:
   a. Sufficient: The information in the article is sufficient to answer the question.
   b. Some: The article contains necessary information to answer the question but is not sufficient.
   c. None: The article does not contain relevant information to answer the question.

Figure 17: Guidelines for verifying the quality of generated cross-document Q&A pairs, as described in Section 4.3. We also provide additional examples to guide the annotation.



Answer question with retrieved articles

You are an AI assistant tasked with answering a given question. Your goal is to generate an answer for a given question based on the provided articles and their publication dates. The answer should:
1. Fully answer the question
2. Be brief and concise
3. Use the SAME LANGUAGE as the given question
4. Be supported by the articles.

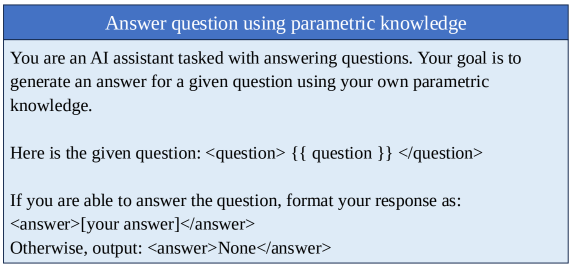
Here are the given question and articles:
<question> {{ question }} </question>

```
<articles>
{% for (idx, text, date) in articles %}
<text_{{ idx }}> {{ text }} </text_{{ idx }}>
<date_{{ idx }}> {{ date }} </date_{{ idx }}>
{% endfor %}
</articles>
```

Note, you should generate the answer based solely on the information of articles. DO NOT use information outside of the given articles, such as your own knowledge.

Format your response as follows: <answer>[Your generated answer. Use one or two sentences at most. Keep the answer as concise as possible.]</answer>

Figure 18: Prompt used to instruct LLMs in using articles to answer questions, as described in Section 5.1.



Answer question using parametric knowledge

You are an AI assistant tasked with answering questions. Your goal is to generate an answer for a given question using your own parametric knowledge.

Here is the given question: <question> {{ question }} </question>

If you are able to answer the question, format your response as:
<answer>[your answer]</answer>
Otherwise, output: <answer>None</answer>

Figure 19: Prompt for answering question using parametric knowledge of LLMs. This corresponds to the "No Retrieval" setting in Section 5.2 and Table 3.

served across these LLMs: translating the supporting articles from non-English to English leads to the most improvement (i.e., +ES). This suggests that the primary challenge does not appear to lie in non-English text generation but rather in reasoning over retrieved information across languages.

E Effect of Chain-of-Thought

By construction, a correct answer in XRAG requires combining – or reasoning over – multiple pieces of information. Because our evaluations do not include CoT (see the prompt in Figure 18), and the evaluated models do not reason over the answer

```
LLM-as-a-Judge

You are an AI assistant tasked with result evaluation. Given a question, a
golden answer, and a generated answer, your goal is to judge whether the
generated answer is correct according to the question and golden answer.

Here is the question:
<question> {{ question }} </question>

Here is the golden answer:
<golden_answer> {{ answer }} </golden_answer>

Here is the generated answer:
<generated_answer> {{ pred }} </generated_answer>

To synthesize your response, follow these steps:
1. Analyze the question type and conditions in the question
2. Identify the key information in the golden answer that can completely
solve the question (e.g., names for "Who" questions and numbers for "how
many" questions)
3. Check whether the key information exists in the generated answer
4. Ignore differences in punctuation and phrasing between the golden answer
and the generated answer. It is OK if the generated answer contains more
information than the golden answer, as long as it does not contain any
conflicting statements.
5. If the generated answer is in a different language than the golden answer,
it should be considered as wrong.

Your response should be in json format as follows:
{
   "justification": [Explain why you think the GENERATED ANSWER is
Correct or Incorrect. Use one or two sentences at most. Keep the
explaination as concise as possible.],
   "answer": [correct or incorrect]
}
```

Figure 20: Prompt used for LLM-as-a-Judge, as described in Section 5.1.

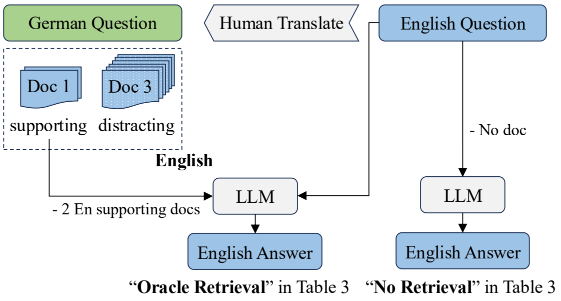


Figure 21: Experimental settings in Table 3.

```
Answer question with chain-of-thought

You are an AI assistant tasked with answering a given question. Your goal is
to generate an answer for a given question based on the provided articles and
their publication dates. The answer should:
1. Fully answer the question
2. Be brief and concise
3. Use the SAME LANGUAGE as the given question
4. Be supported by the articles.

Here are the given question and articles:
<question>
{question}
</question>

<articles>
{search_results}
</articles>

Your response should be a json dictionary containing two fields: "reasoning"
containing a step-by-step reasoning to arrive at the answer, and "answer"
with a concise answer to the question. Example:

{{"reasoning": "A step-by-step reasoning to arrive at the answer", "answer":
"A concise answer to the question"}}

Note, you should generate the answer based solely on the information of
articles. DO NOT use information outside of the given articles, such as your
own knowledge.
```

Figure 22: Chain-of-thought prompt for answering question using the retrieved documents.

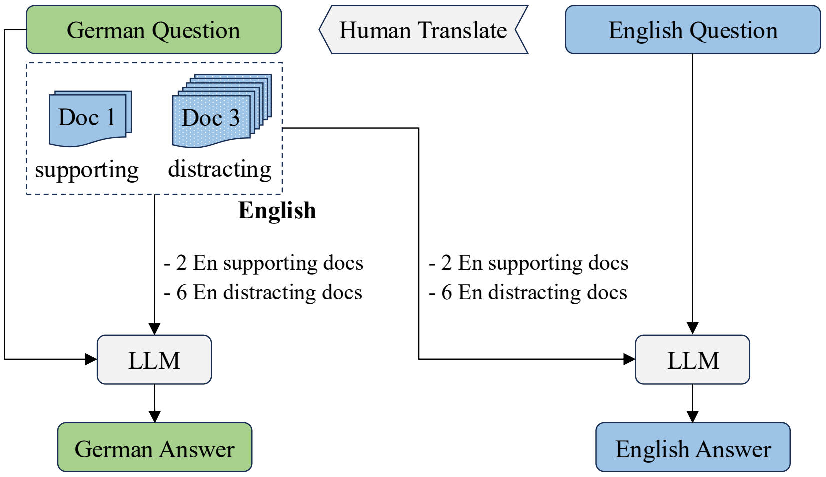
Model	**Default Prompt**	**CoT Prompt**
Claude 3.7	47.70	63.80
Nova Pro	49.80	57.20

Table 12: Comparison of default and Chain-of-Thought prompting on XRAG Task 1.

by default, this "reasoning" has to be performed in implicit form in the model representations for all models evaluated in our paper. In contrast, "reasoning" LLMs are models that produce an explicit reasoning trace.

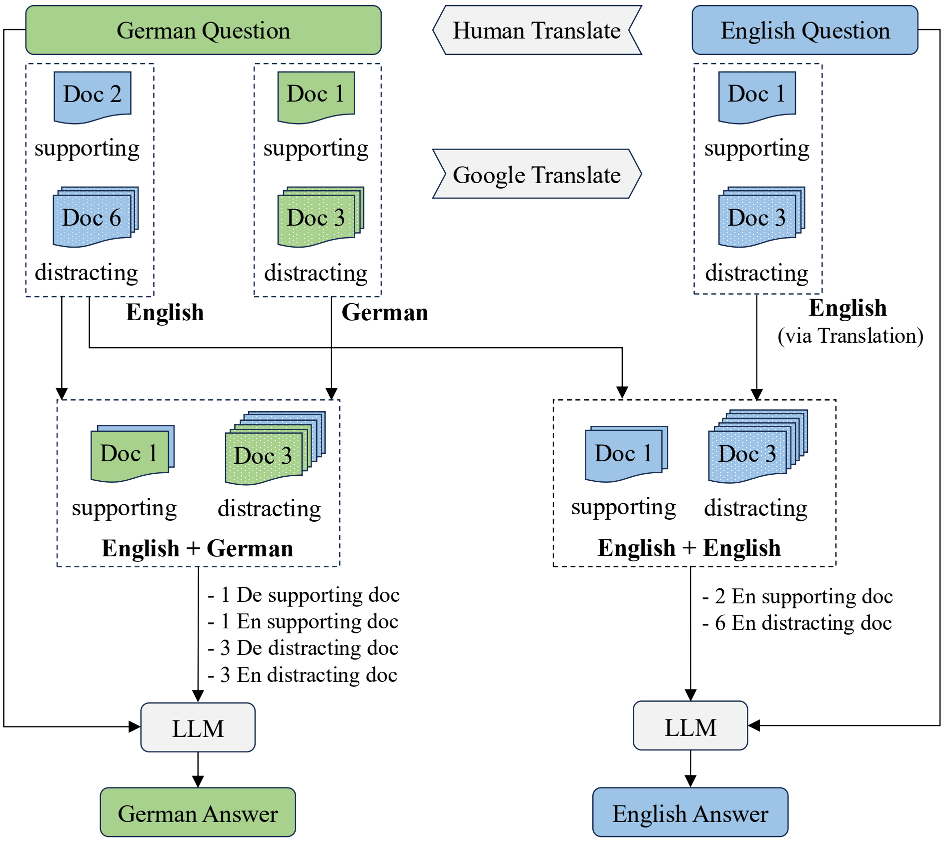
To connect the implicit reasoning required by our benchmark with explicit reasoning traces, we conducted additional experiments comparing a default prompt and a chain-of-thought (CoT) prompting setup. Table 12 shows average accuracy on Task 1 of XRAG using a single-judge (GPT-4) setup (non-English queries with English documents):

These results show that models prompted to produce explicit reasoning traces perform significantly better than those providing a direct answer, giving credence to the claim that the implicit reasoning required by our dataset construction benefits from explicit reasoning trace generation (the prompt is illustrated in Figure 22).



Doc. Lang.=**En**, Query. Lang.=**De** in Table 4 Doc. Lang.=**En**, Query. Lang.=**En** in Table 4

Figure 23: Experimental settings in Table 4. Here, we use English (**En**) and German (**De**) as examples.



Doc. Lang.=**En+De**, Query. Lang.=**De** in Table 5 Doc. Lang.=**En+En**$_{\text{De}}$, Query. Lang.=**En** in Table 5

Figure 24: Experimental settings in Table 5. Here, we use English + German (**En+De**) as an example.

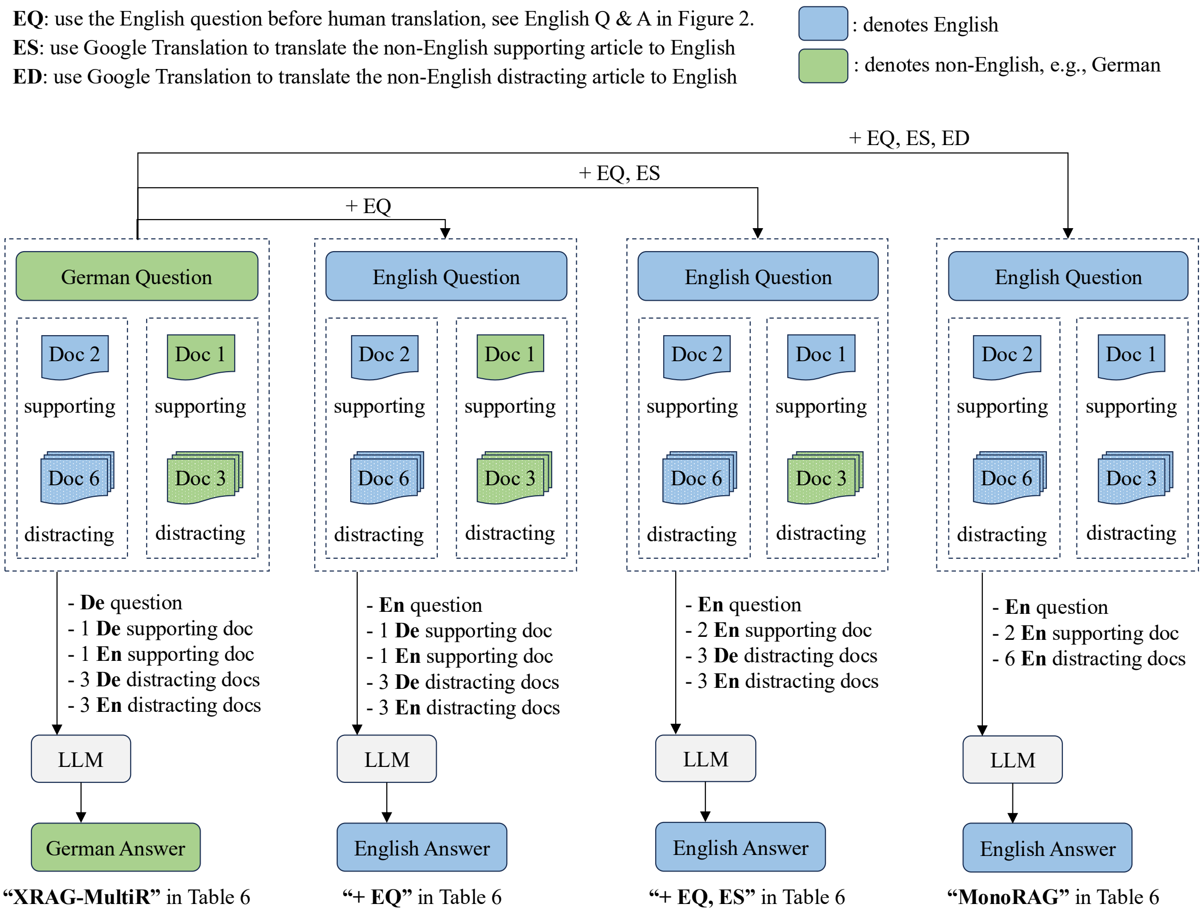


Figure 25: Experimental settings in Table 6. Here, we use English + German (**En+De**) as an example.