BERGEN: A Benchmarking Library for Retrieval-Augmented Generation

David Rau[1,*], Hervé Déjean[2], Nadezhda Chirkova[2], Thibault Formal[2], Shuai Wang[3,*], Stéphane Clinchant[2], Vassilina Nikoulina[2]
[1] University of Amsterdam, [2] Naver Labs Europe, [3] The University of Queensland
[1]`d.m.rau@uva.nl`, [2] `{firstname.lastname}@naverlabs.com`, [3] `shuai.wang2@uq.edu.au`,

Abstract

Retrieval-Augmented Generation allows to enhance *Large Language Models* with external knowledge. In response to the recent popularity of generative LLMs, many RAG approaches have been proposed, which involve an intricate number of different configurations such as evaluation datasets, collections, metrics, retrievers, and LLMs. Inconsistent benchmarking poses a major challenge in comparing approaches and understanding the impact of each component in the pipeline. In this work, we study best practices that lay the groundwork for a systematic evaluation of RAG and present *BERGEN*, an end-to-end library for reproducible research standardizing RAG experiments. In an extensive study focusing on QA, we benchmark different state-of-the-art retrievers, rerankers, and LLMs. Additionally, we analyze existing RAG metrics and datasets. Our open-source library *BERGEN* is available under `https://github.com/naver/bergen`.

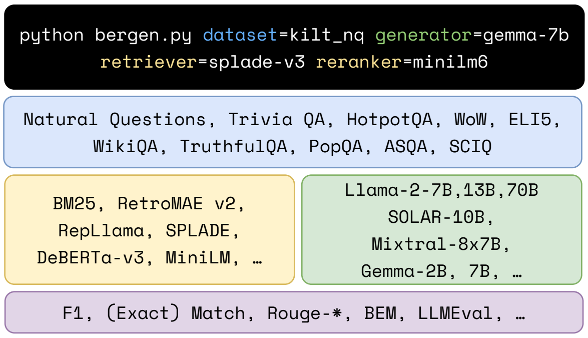


Figure 1: Summary of features in *BERGEN*. *BERGEN* enables a reproducible and comprehensive study of state-of-the-art retrievers, rerankers and LLMs in RAG (we conduct 500+ experiments –see Table 4).

1 Introduction

With billions of learnable parameters, Large Language Models (LLMs) hold the capacity to store vast amounts of the information contained in the pretraining data, transcending mere common sense knowledge (Devlin et al., 2019; Radford et al., 2019; Touvron et al., 2023; Kim et al., 2023; Team, 2023; OpenAI et al., 2024; Wei et al., 2022). This knowledge, embedded in the model weights, can be accessed through model prompting after an alignment step (Ouyang et al., 2022; Zhang et al., 2023), transforming LLMs into universal Question Answering (QA) tools and sparking an unprecedented surge in commercial and scientific interest.

However, a major limitation of such LLMs is that their knowledge is static and can not be directly manipulated. Consequently, inaccurately memorized or outdated information within the model's parameters cannot be easily identified, let alone updated, and can lead to erroneous responses. Therefore, ensuring factual accuracy has become a major concern when millions of users interact with LLMs or when addressing domain-specific QA scenarios where LLMs must rely on external information.

Such challenges are addressed by Retrieval-Augmented Generation (RAG) (Das et al., 2019; Seo et al., 2019; Lewis et al., 2020), where relevant information, *retrieved* from a given external collection, is *explicitly* provided as context to the LLM to generate an answer that can go beyond its internal knowledge. Due to their multi-step nature, RAG pipelines are complex systems whose final performance is influenced by a myriad of possible configurations and design choices.

New RAG approaches are usually characterized by fragmented and often suboptimal experimental setups, e.g. using outdated retrievers or unreliable metrics. The importance of the evaluation metrics is even more important in *zero-shot* RAG settings, where LLM-generated answers are more verbose compared to standard QA short answers, and surface-matching metrics fail to capture whether the answer is correct. The described inconsistency between setups makes new methods hardly comparable, and the absence of a systematic

[*]Work performed while at Naver Labs Europe.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 7640–7663
November 12-16, 2024

Source	Dataset	Metric	Models	Collection	Top-n docs	Setting
Izacard and Grave (2021)	NQ, TriviaQA (unfiltered), SQuAD Open	Exact Match, F1	BM25, DPR, T5	Wikipedia '16, '18	5, 10, 25, 50, 100	Full-FT
Asai et al. (2024a)	PopQA, TriviaQA (unfiltered), PubHealth, ARC-C, Bio, ASQA	Match, Precision, Recall, Accuracy, Mauve	Contriever, Search Engine, GTR-XXL, Llama2 7B, 13B	Wikipedia '18, '20, '23	5, 10	
Lin et al. (2024)	MMLU, NQ, TQA, ELI5, HotpotQA, FEVER, AIDA, zsRE, T-REx, WoW	Exact Match, Accuracy	DRAGON+, Llama 65B	Wikipedia '17-'20, Wiki21 from Common Crawl	10	0-Shot, Few-Shot, Full-FT
Ma et al. (2023)	HotpotQA, PopQA, AmbigNQ, MMLU	Exact Match, F1	Bing-API, BM25, ChatGPT, T5, Vicuna 13B		1	Few-Shot
Kim et al. (2024)	NQ, WebQ, 2Wiki, HotpotQA	Exact Match, F1	Contriever, BM25, ChatGPT, Llama2-chat-70B	KILT Wikipedia	10	0-Shot
Kamalloo et al. (2023)	NQ-Open	Exact Match, F1, BEM	DPR, Contriever, InstructGPT, FID, R2-D2 EMDR[2]	KILT Wikipedia	25, 50, 100	0-Shot

Table 1: Non-exhaustive examples of experimental setups in the RAG literature: Everybody uses their own setup!

evaluation of the impact of various RAG components complicates understanding the effectiveness of the proposed approaches as well as the interactions between the retrieval system and the LLM.

Our contribution. To address the challenges described above, we introduce *BERGEN* –short for *BEnchmark on Retrieval augmented-GENeration*– a Python library for easy and reproducible end-to-end RAG experiments. Through *BERGEN*, we conduct a comprehensive study benchmarking state-of-the-art retrievers, rerankers, and LLMs in 500+ experiments. By comparing a large number of prominent datasets and metrics, we derive *best practices for testing RAG approaches*, laying the groundwork for comparable results and future advancements in this field. *BERGEN* also supports multilingual datasets to promote RAG development beyond English. In a nutshell, our main findings are as follows:

- It is important to perform more semantic evaluation, e.g. LLM-based evaluation, beyond commonly used surface-matching metrics (e.g. exact match, F1, Rouge-L, etc.).
- Retrieval quality matters for RAG response generation, hence the importance of usage of SoTA retrievers and rerankers in RAG.
- We highlight the importance of reviewing standard benchmarks for knowledge-intensive tasks commonly used for RAG: some datasets evaluating general knowledge might not be suitable for RAG in the context of modern LLMs which have acquired most of such knowledge from the Web and Wikipedia.
- LLMs of any size can benefit from retrieval.

2 Related Work

RAG libraries. First, LangChain (LangChain, Accessed 2024) and LlamaIndex (LlamaIndex, 2024) offer generic off-the-shelf application modules for high-level RAG development tailored for production-ready applications. Furthermore, Khattab et al. (2023) present DSPy, a programming-based approach that creates compositional and declarative modules to build complex LLM operations. More recently, RAGGED (Hsia et al., 2024) explores optimal RAG pipeline designs such as exploring encoder-decoder *vs* decoder-only models for generation. FlashRAG (Jin et al., 2024) introduces a modular open-source toolkit designed for RAG experiments. Both have been developed concurrently with this work and as such are most similar to our framework.

However, neither LlamaIndex, DSPy, nor RAGGED offer sufficient flexibility for a research environment and focus on a limited selection of retrievers, datasets, and metrics. Additionally, FlashRAG lacks an in-depth analysis of such components. Furthermore, a reranking functionality is often overlooked and none of the works analyze or highlight enough the importance of retrieval quality. In contrast, our framework prioritizes flexibility and extensibility by simply writing configuration files for models and datasets to cover a wide range of supported configurations.

Inconsistent Setups. Amidst the growing interest in LLMs, numerous RAG approaches have been introduced recently (Izacard and Grave, 2021; Izacard et al., 2022b; Jiang et al., 2023; Lin et al., 2024; Asai et al., 2024a; Jiang et al., 2023; Kim et al., 2024; Ram et al., 2023; Ma et al., 2023; Xu et al., 2024). Among those works, the experimental setups are *fragmented at best*. Works vary in the use of evaluation datasets, collections, evaluation metrics, retrieval systems, and LLMs. We present examples of experimental setups in Table 1 highlighting the current chaotic state of RAG evaluation that does not allow a systematic comparison across methods or components in the pipeline.

Retrieval in RAG. The impact of the retrieval quality as well as its relative impact w.r.t. the size of the LLM remain unclear. While efforts have focused on mitigating hallucinations (Chen et al., 2023; Ji et al., 2023; Mishra et al., 2024) and dealing with noisy contexts (Cuconasu et al., 2024) within the LLM component, the impact of the retrieval component to improve responses remains underexplored (Asai et al., 2024b). Recent state-of-the-art approaches employ *outdated retrievers* without refining the ranking, which is a critical aspect for retrieval quality (Craswell et al., 2023). For instance, none of the works presented in Table 1 employ a re-ranking stage.

Data processing. For providing external context to the LLM, different sources can be utilized. While Wikipedia is the most common practice, utilizing snapshots with different timestamps causes additional inconsistencies among approaches. Variations in data preprocessing can further complicate comparisons (Tamber et al., 2023) and have an impact on observed performance.

To streamline the puzzling number of different experimental configurations, what is needed is a unified framework to systematically train and evaluate RAG systems. Asai et al. (2024b) acknowledge this challenge and call for a “standardized and open-sourced library for retrieval-based LMs".

3 Task Definition

RAG consists of a ranking system $\mathcal{R}$ and a parametric generative language model θ, where the ranking system can be multi-staged. First, the ranking system builds a search index $\mathcal{I}$ based on a collection. Then, at request time, the index $\mathcal{I}$ is searched yielding context segments[1] c that are relevant to the user input x: $c = f_{\mathcal{I},\mathcal{R}}(x)$. Next, the LLM generates a response r based on the context c and user input x both embedded in a model-specific instruction template i: $r = f_\theta(i, x, c)$.

[1]The segments can be at different granularities for instance sentences, passages, or entire documents. In this work, we focus on passages.

4 Benchmarking Library *BERGEN*

We present *BERGEN*, an open-source Python library that standardizes RAG experiments[2]. *BERGEN* supports a wide range of model architectures as well as training and evaluation configurations and at its core is designed to be extendable with minimal code. The main goal is to simplify the currently fragmented experimental setup of RAG research. Our library allows reproducing experiments end-to-end including data download, preprocessing, indexing, retrieval, generation, and training for a wide range of state-of-the-art models with a simple command:

```
python bergen.py retriever='splade-v3'
reranker='minilm6'
generator='SOLAR-10.7B'
dataset='kilt_nq' train='lora'
```

To accommodate the fast-paced efforts in open-sourcing models and datasets, *BERGEN* is built on top of the Hugging Face (HF) hub to handle datasets (Lhoest et al., 2021) and models (Wolf et al., 2020), allowing for a straightforward extension with all available resources hosted on the hub, as well as locally stored ones. *BERGEN* further includes a wide set of popular QA datasets, including multilingual datasets, as well as surface-based and LLM-based metrics for evaluation. For an overview of all features included, we refer to our github repository. The library supports zero-shot evaluation as well as different fine-tuning configurations. We rely on Hydra (Yadan, 2019) to handle complex experiment configurations. For instance, adding a new LLM to *BERGEN* is as simple as adding a `yaml` config file:

```
init_args:
  _target_: models.generators.llm.LLM
  model_name:
  "Upstage/SOLAR-10.7B-Instruct-v1.0"
  max_new_tokens: 128
  max_length: 2048
  quantization: "int4"
batch_size: 16
```

We now give an overview of models, datasets, collections, evaluation metrics, and training *currently* supported in *BERGEN*.

4.1 Retrievers

BERGEN supports indexing and retrieval with the most popular first-stage retrievers spanning traditional, dense, and sparse bi-encoders. We support

[2]https://github.com/naver/bergen

Pyserini's BM25 (Lin et al., 2021), various sparse SPLADE models (Formal et al., 2022; Lassance et al., 2024), as well as dense (encoder-only) models such as CoCondenser (Gao and Callan, 2022), RetroMAE (Shitao et al., 2022), or BGE (Xiao et al., 2023). *BERGEN* also supports decoder-based retrievers like RepLLaMA (Wang et al., 2024b), or models like BGE-M3 (Chen et al., 2024) for multilingual scenarios. Since our library builds on top of the HF hub, including any other dense or sparse model is straightforward.

4.2 Rerankers

Modern retrieval systems refine the initial ranking using rerankers such as Cross-Encoders (Nogueira and Cho, 2020). In contrast to the initial retrieval which encodes queries and passages independently for efficiency purposes, rerankers contextualize passages w.r.t. queries and thus produce more effective representations. Using a reranker is crucial to improve ranking quality at early ranks – this is particularly important since only a limited number of passages can be provided as context to the LLM. *BERGEN* supports Cross-Encoders such as MiniLM (Wang et al., 2020), DeBERTa-v3 (Lassance and Clinchant, 2023), or BGE(-M3) (Xiao et al., 2023; Chen et al., 2024).

4.3 LLMs

BERGEN supports the most popular open-weights LLMs such as Llama2 (Touvron et al., 2023), Llama3 (AI@Meta, 2024), SOLAR (Kim et al., 2023), Mixtral (Jiang et al., 2024), Gemma (Team, 2023), TinyLlama (Zhang et al., 2024a), and Command-R[3] (multilingual). To accommodate the fast-paced development of LLMs, our library allows adding new HF models simply by defining a config file as shown earlier.

4.4 Evaluation Datasets

Among the research community, there is a disparity regarding which datasets to use for evaluating RAG. We identified 40+ datasets among recently proposed RAG approaches, spanning (multi-hop)-Question Answering, multiple-choice, entity linking, conversational, fact-checking, and slot-filling.

In this work, we focus on QA and select the most popular publicly available datasets for *BERGEN*. These datasets cover different characteristics of QA such as short- and long-form Question Answering in different domains. We include Natural Questions (NQ) (Kwiatkowski et al., 2019), Trivia QA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), Wizard of Wikipedia (WoW) (Dinan et al., 2019), ELI5 (Fan et al., 2019), WikiQA (Yang et al., 2015), TruthfulQA (Lin et al., 2022), PopQA (Mallen et al., 2023), ASQA (Stelmakh et al., 2022), SCIQ (Welbl et al., 2017), MKQA (Longpre et al., 2021) and XOR-TyDi QA (Asai et al., 2021a) –the last two for multilingual RAG. New datasets can also be easily integrated into *BERGEN*.

4.5 Collection

The core strength of the RAG setup is that the LLM can be augmented with relevant context stemming from any source. Consequently, many different collections can be chosen from and vary among the proposed approaches. Different data preprocessing, such as splitting the data into smaller chunks, and downloading the data at different timestamps, can cause additional inconsistencies among setups. (Petroni et al., 2021) solve this by using a single fixed Wikipedia dump to retrieve from across different datasets. We utilize this publicly available KILT (Petroni et al., 2021) Wikipedia dump[4] and, similarly to (Tamber et al., 2023; Karpukhin et al., 2020), split articles into non-overlapping chunks of 100 words, and prepend the article title to each chunk, yielding around $24.8M$ passages in total. The resulting collection is in the Hugging Face Arrow dataset format to ensure memory-efficient and performant loading. We implement a dataset engine that allows for multi-threaded end-to-end processing (downloading, processing, and saving datasets) making the addition of new datasets straightforward. To enable experiments with a multilingual datastore, *BERGEN* also supports multilingual Wikipedia[5].

4.6 Evaluation

To date, it remains unclear which metrics are effective for evaluating open-ended generation. Typically, given a question, a reference answer, and a generated candidate answer, the task is to evaluate whether the question is answered sufficiently. The most common metrics can be categorized as surface- and LLM-based metrics. Surface-based metrics rely on exact lexical matching with either

[3] https://huggingface.co/CohereForAI/c4ai-command-r-v01

[4] https://huggingface.co/datasets/kilt_wikipedia

[5] https://huggingface.co/datasets/wikimedia/wikipedia

the entire reference label or its sub-string; on the other side, LLM-based metrics leverage semantic soft-matching. While surface-based metrics may excel at capturing short, factual equivalence, they naturally fall short in accurately capturing the semantic equivalence of longer reference-answer pairs.

We employ the widely-used surface-based metrics Match[6], Exact Match, Precision, Recall, F1[7], Rouge -1, -2, -L, as well as more advanced automatic metrics that are based on semantic similarity: BEM (Bulian et al., 2022), GPT-4 (OpenAI et al., 2024), as well as LLMeval, a simple yet effective LLM-based metric.

LLMeval. There exist numerous works using LLMs as evaluators (Saad-Falcon et al., 2023; Zheng et al., 2023; Kamalloo et al., 2023). Recently, RAGAS (Es et al., 2023) and RetrievalQA (Zhang et al., 2024b) have introduced better, automated evaluation of LLM-generated text. However, as a simple LLM-based metric, we leverage SOLAR-10.7B-Instruct-v1.0 (Kim et al., 2023) as a zero-shot answer equivalence evaluator –similar to Instruct-GPT in (Kamalloo et al., 2023)– providing a good compromise between parameter size (efficiency) and effectiveness. Based on an instruction prompt, we ask the model to judge whether a generated response answers a question compared to a reference answer, resulting in binary relevance judgments. We refer to Appendix F for details.

4.7 Training

BERGEN supports training the LLM end-to-end in different configurations. We support full fine-tuning (FT), as well as QLoRA FT (Dettmers et al., 2023) with 4-bit and 8-bit quantization.

5 Experiments

To our knowledge, the experiments we conduct with *BERGEN* present the largest RAG study yet, comparing a variety of different configurations of retrievers, rerankers, LLMs, datasets, and metrics –as (partly) summarized in Table 4. The computational demands of fine-tuning state-of-the-art LLMs limit us to evaluating the LLMs in this work mostly to zero-shot.

[6]Match measures whether the label is *contained* in the generated answer as an exact match following Schick et al. (2023); Mallen et al. (2023); Asai et al. (2024a); Zhang et al. (2024b).

[7]Precision, Recall, F1 compare the generated answer and the label on the token level.

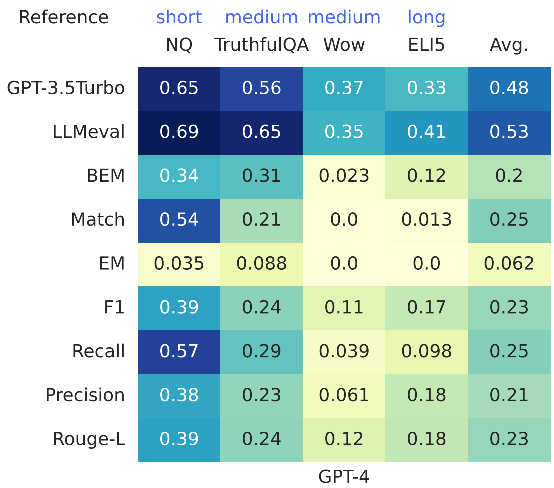


Figure 2: Correlation of different metrics with GPT-4-as-a-judge for datasets with varying reference label lengths (short, medium, and long).

We make several choices to speed up the inference and minimize the required GPU memory. We set the temperature T = 0 for anwer generation, and the number of maximum generated tokens to 128. Generation is done with vLLM (Kwon et al., 2023). Retrievers and rerankers are used in half-precision (Micikevicius et al., 2018). We run our experiments, depending on the size of the LLM, with a maximum of 2x A100 80GB GPUs. We detail our prompts in Appendix E. We retrieve top-50 passages –that are eventually re-ranked– of which we provide the top-5 to the LLM. This is in line with observations made by Hsia et al. (2024) showing that a small number of provided passages is sufficient for decoder-only models.

BERGEN allows us to easily investigate various research questions on evaluation, datasets, the benefit of retrieval, or the impact of LLM size. As such, we bridge the gap in the literature by systematically comparing common (5.1) metrics, (5.2) datasets, (5.3) retrieval systems, and (5.4) LLMs. Finally, we observe the performance that can be gained by (5.6) fine-tuning the LLMs.

5.1 Comparison of Metrics

We analyze a wide range of surface-based as well as LLM-based metrics systematically to answer **(RQ 1)** *Which metrics are most effective for evaluating open-ended text generation and comparing RAG systems?* To cover different characteristics, we select four representative datasets with different reference lengths: NQ (short), TruthfulQA and

WoW (medium), and ELI5 (long reference labels). We evaluate what we found to be a strong RAG system[8], with the motivation to identify metrics that can distinguish the best-performing models effectively.

We compare all our metrics against GPT-4-as-a-judge and measure correlation averaged over samples with Kendall's Tau in Figure 2. We find LLMeval on average to be closest to GPT-4, which is known to be one of the strongest baselines for evaluation tasks (Kamalloo et al., 2023). We further observe surface-based metrics and BEM failing to evaluate long answer-reference pairs, in reference to GPT-4. In contrast, LLMeval shows a strong correlation with GPT-4 for examples with long references, however, weaker compared to references with short- and medium-lengths –highlighting the difficulty of comparing longer answer-reference pairs. Exact Match (EM) fails to evaluate zero-shot responses effectively. Manual inspection reveals LLM responses are more verbose than the short references in NQ, making exact matches difficult, especially for medium and long references.

> Recommendation : **Evaluation**
>
> LLMeval closely aligns with GPT-4's evaluation, followed by Match and Recall, making them the most effective non-commercial metrics for (zero-shot) RAG evaluation, among the ones tested.

We use LLMeval in the remainder of this work and include results with the Match metric in Appendix A.

5.2 Datasets for RAG Evaluation

In this section, we analyze 10 QA datasets covering a wide set of characteristics such as different question lengths, reference label lengths, and domains to investigate **(RQ 2)** *Which datasets are suitable for RAG?* For this experiment, we are interested in how much performance can be gained by adding relevant context to the LLM compared to no retrieval (Closed Book). We argue that the more performance can be gained by adding retrieval, the more "suitable" the dataset is for RAG evaluation. For this experiment again, we leverage the same strong retrieval system.

[8]Retrieval: SPLADE-v3, re-ranking: DeBERTa-v3, and answer generation: SOLAR-10.7B-Instruct-v1.0 – See Sections 5.3 and 5.4.

Figure 3 shows that retrieval does not increase response generation quality for all datasets. Specifically, for TruthfulQA, ELI5, and WoW, generation performance deteriorates by adding retrieved context to the LLM. Even adding oracle retrieval to ELI5 and WoW does not lead to increased performance (see Figure C). There could be multiple explanations for such results that would require further investigation and detailed analysis in future work. First, some dataset labels are noisy or incomplete and LLMs answers may actually be better, while some questions and tasks may not require external knowledge. Second, most retrieval systems are not trained for very long questions, which could make it especially challenging for certain datasets. The evaluation of longer references is also more challenging –highlighting the importance of developing better evaluation metrics. Lastly, Wikipedia is often used in the pre-training collection of LLMs. Therefore, the models might have memorized the answers, rendering retrieval obsolete, which further highlights the importance of developing new datasets. A more detailed analysis of failure cases can be found in Appendix B.

On the other hand, ASQA, HotpotQA, NQ, TriviaQA, and PopQA gain the most performance by adding retrieval. For exact numbers, we further refer to Table 4 in Appendix A.

> Recommendation : **Datasets**
>
> ASQA, HotpotQA, NQ, TriviaQA, and PopQA benefit most from retrieval in zero-shot settings. In contrast, TruthfulQA, SCIQ, and ELI5, WoW do not benefit from current state-of-the-art retrieval nor from oracle retrieval (where available) and seem to be more challenging. This suggests that the current SoTA retrieval systems and evaluation are not sufficient for these datasets and highlight potential areas for future research directions.

5.3 Impact of Retrieval

Providing a high-quality ranking to the LLM is crucial, as only a limited set of passages can be provided as context for generation. To achieve this, modern retrieval systems refine the initial ranking using rerankers such as Cross-Encoders. The relation between retrieval quality and downstream generation performance remains relatively under-

Figure 3: Performance gain w/ and w/o retrieval (SPLADE-v3 + reranking (RR) with DeBERTa-v3) on different datasets with SOLAR-10.7B.

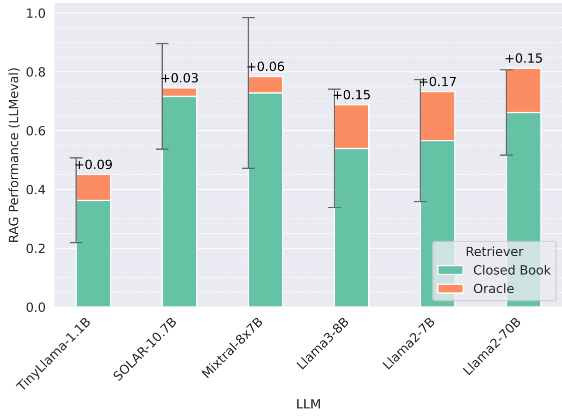


Figure 5: Performance gains w/ and w/o oracle retrieval for LLMs with different sizes. Comparing closed book *vs* oracle passages averaged over all QA datasets in KILT.

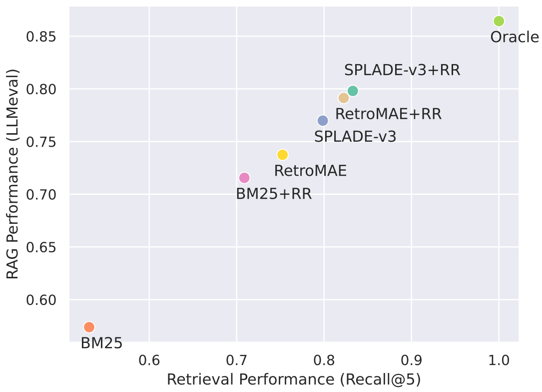


Figure 4: Impact of retrieval performance on RAG Performance for SOLAR-10.7B on NQ with different ranking systems. RR means with additional re-ranking using DeBERTa-v3.

explored, particularly relative to different LLM sizes. To answer **(RQ 3)** *Does retrieval quality positively impact generation quality?*, we compare the performance of LLMs with several retrievers, and with optional reranking. The QA datasets in KILT also contain relevance labels allowing us to additionally evaluate ranking –see Table 5 in Appendix C. Note that we focus here on *"zero-shot rankers"*, i.e. models typically trained on the MS MARCO passage ranking collection (Bajaj et al., 2018) – and not on the target collection. In Appendix D we further include more comprehensive ablations of modern SoTA retrievers from the MTEB benchmark (Muennighoff et al., 2023) – which are fine-tuned on the KILT collections.

In Figure 4 we measure LLMs' performance against retrieval effectiveness on the NQ dataset. We select three popular retrievers with different characteristics; namely BM25 (lexical sparse), RetroMAE (dense), and SPLADE-v3 (learned sparse). We additionally rerank the initial retrieval with a DeBERTa-v3 cross-encoder. We find that with increased retrieval quality, LLM performance improves across LLMs by a large margin. Overall, re-ranking largely boosts results, and SPLADE-v3 reranked with DeBERTa-v3 achieves the best performance across datasets and metrics. These observations hold similarly for other datasets –as seen in Table 4. To understand how much more performance could be gained if we had access to even better retrieval systems, we also provide passages that directly contain the answer (Oracle passages) as context to the LLM for datasets that contain relevance annotation. We find that improving ranking systems could further boost LLM performance for RAG (see Table 4 and Figure 5).

Recommendation : Retrieval

For RAG downstream performance, it is crucial to employ SoTA retrieval systems in the RAG pipeline. Reranking has been often overlooked and should be used to have strong baselines for future research.

5.4 Impact of LLM size

Next, we investigate whether adding retrieval is more beneficial for a specific model size. We select LLMs with different sizes ranging from 1 to 70B parameters. To answer **(RQ 4)** *What is the impact of the LLM size in RAG?*, we measure in Figure 5

	en	ar	fi	ja	ko	ru
			MKQA			
No Ret	0.67	0.29	0.32	0.37	0.32	0.48
En Wiki	**0.76**	0.54	0.58	0.63	0.59	0.71
Multi Wiki	0.74	**0.57**	**0.64**	**0.64**	**0.62**	**0.72**
			XORQA			
No Ret	0.63	0.56	0.41	0.40	0.54	0.47
En Wiki	**0.73**	0.57	0.59	0.51	0.58	0.65
Multi Wiki	0.69	**0.70**	**0.74**	**0.62**	**0.66**	**0.74**

Table 2: Impact of retrieval in the multilingual setting. Generator: Command-R, retriever/reranker: BGE-M3. Columns denote the language of user queries while rows denote the language of the datastore (English Wikipedia, or multilingual Wikipedia). Metric: LLMeval.

the performance of the LLMs with gold passages (Oracle) and without retrieval (Closed Books).

Our experiments show no clear relation between model size and performance gain by adding (perfect) retrieval. Llama2 7B gains the most performance, followed by Llama2 70B and Llama3 8B, TinyLlama 1.1B, Mixtral 8x7B, and SOLAR 10.7B. It is worth noting that Llama2 7B with retrieval outperforms its biggest counterpart Llama2 70B without retrieval. In conclusion, our results show that neither model size nor performance without retrieval is generally indicative of the usefulness of adding retrieval for zero-shot response generation. The same observations hold when considering retrieval systems –instead of Oracle (not shown).

5.5 Multilingual RAG

We further extend *BERGEN* to support multilingual experiments –see extended descriptions and analyses in Appendix H and Chirkova et al. (2024). Table 2 reports results for multilingual RAG . We observe that retrieving from the English Wikipedia datastore is already beneficial for non-English queries. Retrieval from multilingual Wikipedia boosts results further.

5.6 Fine-Tuning the LLMs

Finally, we want to understand whether the performance gap between the different model sizes can be closed by fine-tuning the models by answering **(RQ 5)** *How much performance can be gained by fine-tuning?* Due to the computational cost, we limit our experiments to a single dataset. We select NQ as significant performance can be gained by adding retrieval as shown by the previous experiment. We fine-tune the LLMs using QLoRa –for further details, see Appendix G.

LLM	M	LLMeval
TinyLlama-1.1B-chat	0.56 (+0.13)	0.77 (+0.41)
Llama-2-7B-chat	0.64 (+0.03)	0.82 (+0.24)
Llama-3-8B-chat	0.66 (+0.02)	0.78 (+0.04)
SOLAR-10.7B	0.67 (-0.03)	0.84 (+0.05)
Mixtral-8x7B-inst.	0.68 (+0.01)	0.84 (+0.05)
Llama-2-70B-chat	0.69 (+0.04)	0.85 (+0.06)

Table 3: LLMs fine-tuned on NQ, for retrieval with SPLADE-v3 and reranking with DeBERTa-v3. Performance gains in absolute points compared to zero-shot is indicated in brackets.

We observe in Table 3 that smaller LLMs gain more performance with fine-tuning compared to their bigger counterparts. Our results also demonstrate that fine-tuning significantly reduces the performance gap between the smallest (1.1B) and the largest LLM (70B), compared to the zero-shot evaluation setting.

6 Conclusion

In this work, we present *BERGEN*, a library for benchmarking RAG systems. We conduct hundreds of experiments with various configurations allowing us to analyze each part of the RAG pipeline, to derive recommendations for testing and provide strong baselines for future RAG experiments.

We highlight it is crucial to perform semantic evaluations, in addition to commonly used surface-matching metrics We show retrieval quality significantly impacts RAG response generation, underscoring the importance of using state-of-the-art retrievers, specifically rerankers. We emphasize the need to review standard benchmarks for knowledge-intensive tasks in RAG. Additionally, we show that LLMs of any size can benefit from improved retrieval methods. To keep up with the rapid development of LLMs and the constant release of models, we plan to add more retrieval models, LLMs, and datasets in the future. Additionally, by designing the library to be easily extendable, we make it straightforward for the research community to contribute. To conclude, we provide a modular framework, alongside data and runs, for systematically evaluating RAG pipelines and contributing to better reproducibility and understanding of the effectiveness of current and future RAG systems.

Limitations

Despite conducting a very large set of experiments to understand the effect of various RAG components, including different retrievers, rerankers, and LLMs, this work comes with limitations. First, limited by the computational demands of the most recent LLMs, we are restricted to choosing a set of models and datasets, while at the same time primarily focusing on evaluating LLMs zero-shot.

Second, we conduct all experiments using a single Wikipedia-based collection, which is similar to the data on which the LLMs were trained. It would be interesting to explore out-of-domain collections with different characteristics, such as those in the medical or legal domains, to better understand how both the retrievers and LLMs operate in diverse contexts.

Lastly, our experiments are limited to focusing mostly on QA RAG, which only highlights one out of many possible RAG applications such as summarization, open-domain dialogue, slot-filling, and fact verification. We encourage the research community to extend our insights by evaluating more models and datasets and experimenting with multi-lingual settings.

References

AI@Meta. 2024. Llama 3 model card.

Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021a. 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*, pages 547–564, Online. Association for Computational Linguistics.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024a. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

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

Akari Asai, Zexuan Zhong, Danqi Chen, Pang Wei Koh, Luke Zettlemoyer, Hannaneh Hajishirzi, and Wen-tau Yih. 2024b. Reliable, adaptable, and attributable language models with retrieval. *arXiv preprint arXiv:2403.03187*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Boerschinger, and Tal Schuster. 2022. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. *arXiv preprint arXiv:2202.07654*.

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

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2023. Benchmarking Large Language Models in Retrieval-Augmented Generation. ArXiv:2309.01431 [cs].

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)*, page 177–188, Bangkok, Thailand. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2023. Overview of the trec 2022 deep learning track. In *Text REtrieval Conference (TREC)*. NIST, TREC.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. In *International Conference on Learning Representations*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural ir models more effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2353–2359, New York, NY, USA. Association for Computing Machinery.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853, Dublin, Ireland. Association for Computational Linguistics.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *Proc. of SIGIR*.

Jennifer Hsia, Afreen Shaikh, Zhiruo Wang, and Graham Neubig. 2024. Ragged: Towards informed design of retrieval augmented generation systems.

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

Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. ArXiv:2007.01282 [cs].

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022b. Atlas: Few-shot Learning with Retrieval Augmented Language Models. ArXiv:2208.03299 [cs].

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. 2024. Flashrag: A modular toolkit for efficient retrieval-augmented generation research.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Ehsan Kamalloo, Nouha Dziri, Charles L. A. Clarke, and Davood Rafiei. 2023. Evaluating Open-Domain Question Answering in the Era of Large Language Models. ArXiv:2305.06984 [cs].

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. 2023. Solar 10.7b: Scaling large language models with simple yet effective depth upscaling.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Improving open-domain question answering of LLMs via summarized retrieval. In *The Twelfth International Conference on Learning Representations*.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention.

LangChain. Accessed 2024. LangChain Documentation. `https://python.langchain.com/`.

Carlos Lassance and Stéphane Clinchant. 2023. Naver labs europe (splade) @ trec deep learning 2022.

Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. Splade-v3: New baselines for splade.

Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. 2024. Open source strikes bread - new fluffy embeddings model.

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.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xianming Li and Jing Li. 2024. Angle-optimized text embeddings.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, New York, NY, USA. Association for Computing Machinery.

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

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2024. RA-DIT: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*.

LlamaIndex. 2024. Llamaindex: Data framework for llm applications. Accessed: 2024-05-28.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Luke Merrick, Danmei Xu, Gaurav Nuti, and Daniel Campos. 2024. Arctic-embed: Scalable, efficient, and accurate text embedding models.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In *International Conference on Learning Representations*.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained Hallucination Detection and Editing for Language Models. ArXiv:2401.06855 [cs].

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage re-ranking with bert.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic embed: Training a reproducible long context text embedder.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *TREC*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2023. Ares: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441, Florence, Italy. Association for Computational Linguistics.

Xiao Shitao, Liu Zheng, Shao Yingxia, and Cao Zhao. 2022. Retromae: Pre-training retrieval-oriented language models via masked auto-encoder. In *EMNLP*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Manveer Singh Tamber, Ronak Pradeep, and Jimmy Lin. 2023. Pre-processing Matters! Improved Wikipedia Corpora for Open-Domain Question Answering. In *Advances in Information Retrieval*, pages 163–176, Cham. Springer Nature Switzerland.

Google Gemini Team. 2023. Gemini: A family of highly capable multimodal models.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024a. Text embeddings by weakly-supervised contrastive pre-training.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. Improving Text Embeddings with Large Language Models. ArXiv:2401.00368 [cs].

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Retrieval meets long context large language models. In *The Twelfth International Conference on Learning Representations*.

Omry Yadan. 2019. Hydra - a framework for elegantly configuring complex applications. Github.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

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.

Jiacheng Ye, Xijia Tao, and Lingpeng Kong. 2023. Language versatilists vs. specialists: An empirical revisiting on multilingual transfer ability.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024a. Tinyllama: An open-source small language model.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. Instruction tuning for large language models: A survey.

Zihan Zhang, Meng Fang, and Ling Chen. 2024b. Retrievalqa: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering.

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*.

A Main Results

Our main results of evaluating different LLMs with context provided through different retrieval systems on 10 datasets can be found in Table 4. The table comprises the results of 450+ experiments.

B Dataset Analysis

We provide additional support and analysis on the two datasets ELI5 and WoW. More specifically, we lay out reasons why they may not be suited for RAG evaluation in our benchmark. We use different retrievers and two LLMs (SOLAR 10.7B and Llama2 70B) to illustrate our points. Additional results with more retrievers and LLMs can be found in Figure 7 –the conclusions remain however similar.

In Figure 6a, we plot the retrieval performance against the LLMEval metric on the ELI5 dataset for various retrievers. The Closed Book setting (no retrieval) outperforms the Oracle retrieval for which only gold passages (that contain the answer) are provided as context. Surprisingly, the different retrievers have low retrieval performance (<0.3 R@5), but improve generation quality when compared to Oracle. This may indicate partial annotation and/or missing relevant documents. In any case, the performance is much lower than in the Closed Book setting. This is why we consider that ELI5 is probably not appropriate at the moment for testing RAG systems.

In Figure 6b we present a similar analysis of the WoW dataset. Similarly, the Closed Book setting outperforms other systems –including the approach providing the LLM with the oracle passages. In this case, none of the systems with retrieval outperforms the Oracle. Looking closely at the task and some examples, it is actually not clear why this dialogue task should rely on retrieved knowledge from Wikipedia. As an example of a dataset that we find suitable for RAG, we list NQ (Figure 6c). We observe increasing benefits from using stronger retrieval systems, with the oracle retrieval achieving the highest performance.

C Retrieval Evaluation on KILT

KILT contains passage- and document-level annotations of gold documents containing the answer. However, these annotations are not compatible with our 100-word passage split, therefore we map our passages to the document-level ranking annotations, essentially indicating whether a retrieved passage is contained in a document that has been annotated as relevant, serving as a good indication of relevance.

In Table 6, we measure the retrieval effectiveness of different retrieval systems on all datasets in KILT containing ranking labels. We use Recall@5, as this reflects the number of passages used as context to the LLM. We select the models discussed in Section 5.3. We further provide in Appendix D a more exhaustive evaluation of SoTA retrievers on the NQ dataset.

D Retrieval Analysis

We provide comprehensive ablations on the impact of retrieval quality on generation. We study modern SoTA retrievers –including models from the MTEB benchmark which have been fine-tuned on datasets like NQ. Table 6 lists all the models we consider, and Table 7 present the retrieval performance alongside the generation quality (with and without re-ranking respectively). Overall, we observe that SoTA models from MTEB achieve better performance in both aspects. These results are somewhat expected, as fine-tuning ranking models on the target collection improves ranking quality and therefore the relevance of input contexts. However, it does not measure the "zero-shot" performance of the RAG pipeline –especially given the inability of learned retrievers to generalize to out-of-domain collections (Thakur et al., 2021). In the meantime, re-ranking closes the gap between approaches.

E LLM prompts

We opted to use a single general prompt, rather than dataset-specific ones, to minimize the impact of prompt variations and to simplify experimentation. When providing context in the form of retrieved passages to the model, we used the following prompt embedded into the chat-template of the respective model:

		Retrieval System															
		Closed Book		Oracle		RetroMAE		RetroMAE+RR		BM25		BM25+RR		SPLADE-v3		SPLADE-v3+RR	
Dataset	LLM	M	LLM	M	LLM	M	LLM	M	LLM	M	LLM	M	LLM	M	LLM	M	LLM
ASQA (dev, 1k)	Llama2-70B	0.496	0.671	-	-	0.669	0.744	0.712	0.784	0.564	0.665	0.652	0.739	0.705	0.789	0.732	0.815
	Llama2-7B	0.373	0.526	-	-	0.601	0.614	0.673	0.694	0.485	0.511	0.620	0.627	0.650	0.685	0.684	0.718
	Llama3-8B	0.348	0.456	-	-	0.652	0.667	0.690	0.732	0.478	0.506	0.620	0.672	0.682	0.719	0.719	0.762
	Mixtral-8x7B	0.561	0.724	-	-	0.680	0.744	0.716	0.792	0.547	0.645	0.664	0.766	0.724	0.793	0.735	0.819
	SOLAR-10.7B	0.527	0.690	-	-	0.692	0.723	0.743	0.786	0.517	0.554	0.675	0.725	0.722	0.764	0.762	0.811
	TinyLlama-1.1B	0.180	0.200	-	-	0.441	0.369	0.515	0.453	0.305	0.253	0.431	0.385	0.459	0.400	0.528	0.449
KILT ELI5 (dev, 1.5k)	Llama2-70B	0.000	0.706	0.000	0.627	0.000	0.648	0.000	0.663	0.000	0.603	0.000	0.638	0.000	0.670	0.000	0.684
	Llama2-7B	0.000	0.683	0.000	0.469	0.000	0.566	0.000	0.640	0.000	0.429	0.000	0.533	0.000	0.585	0.000	0.626
	Llama3-8B	0.000	0.569	0.000	0.304	0.000	0.501	0.000	0.541	0.000	0.358	0.000	0.438	0.000	0.495	0.000	0.565
	Mixtral-8x7B	0.000	0.749	0.000	0.438	0.000	0.557	0.000	0.605	0.000	0.451	0.000	0.516	0.000	0.569	0.000	0.608
	SOLAR-10.7B	0.000	0.808	0.000	0.386	0.000	0.532	0.000	0.626	0.000	0.368	0.000	0.495	0.000	0.557	0.000	0.626
	TinyLlama-1.1B	0.000	0.449	0.000	0.268	0.000	0.284	0.000	0.329	0.000	0.233	0.000	0.273	0.000	0.288	0.000	0.305
KILT HotpotQA (dev, 5.6k)	Llama2-70B	0.310	0.458	0.749	0.910	0.463	0.610	0.507	0.662	0.503	0.658	0.521	0.690	0.515	0.674	0.536	0.705
	Llama2-7B	0.243	0.372	0.687	0.831	0.386	0.504	0.418	0.541	0.416	0.544	0.455	0.582	0.437	0.561	0.459	0.589
	Llama3-8B	0.228	0.373	0.749	0.887	0.407	0.510	0.450	0.586	0.447	0.562	0.489	0.631	0.463	0.597	0.497	0.643
	Mixtral-8x7B	0.388	0.547	0.776	0.901	0.473	0.592	0.511	0.652	0.506	0.639	0.534	0.679	0.518	0.661	0.545	0.703
	SOLAR-10.7B	0.351	0.501	0.793	0.891	0.446	0.493	0.501	0.575	0.481	0.533	0.530	0.613	0.496	0.567	0.539	0.637
	TinyLlama-1.1B	0.165	0.349	0.539	0.586	0.255	0.286	0.286	0.329	0.286	0.322	0.315	0.361	0.288	0.325	0.315	0.363
KILT NQ (dev, 2.8k)	Llama2-70B	0.448	0.651	0.794	0.905	0.617	0.742	0.642	0.779	0.517	0.652	0.606	0.737	0.637	0.765	0.658	0.791
	Llama2-7B	0.338	0.515	0.776	0.859	0.567	0.652	0.606	0.688	0.443	0.526	0.540	0.632	0.595	0.672	0.616	0.701
	Llama3-8B	0.329	0.493	0.794	0.848	0.595	0.668	0.631	0.732	0.446	0.508	0.571	0.652	0.618	0.711	0.643	0.747
	Mixtral-8x7B	0.526	0.721	0.841	0.899	0.620	0.731	0.664	0.779	0.501	0.631	0.602	0.732	0.640	0.764	0.671	0.790
	SOLAR-10.7B	0.444	0.659	0.829	0.864	0.642	0.731	0.689	0.792	0.498	0.574	0.623	0.716	0.674	0.768	0.702	0.803
	TinyLlama-1.1B	0.168	0.300	0.606	0.530	0.352	0.300	0.417	0.356	0.232	0.187	0.366	0.298	0.371	0.319	0.437	0.364
KILT TriviaQA (dev, 5.3k)	Llama2-70B	0.832	0.855	0.937	0.933	0.873	0.870	0.911	0.904	0.875	0.870	0.907	0.906	0.909	0.905	0.923	0.917
	Llama2-7B	0.657	0.676	0.887	0.880	0.805	0.794	0.854	0.848	0.799	0.783	0.854	0.848	0.850	0.842	0.879	0.866
	Llama3-8B	0.707	0.731	0.892	0.857	0.831	0.799	0.884	0.858	0.815	0.785	0.878	0.856	0.881	0.855	0.902	0.882
	Mixtral-8x7B	0.875	0.873	0.933	0.908	0.866	0.844	0.906	0.882	0.867	0.842	0.900	0.881	0.904	0.885	0.918	0.899
	SOLAR-10.7B	0.805	0.810	0.904	0.836	0.858	0.811	0.915	0.877	0.845	0.789	0.907	0.870	0.911	0.868	0.928	0.898
	TinyLlama-1.1B	0.320	0.447	0.700	0.548	0.604	0.480	0.679	0.568	0.603	0.480	0.695	0.575	0.671	0.551	0.728	0.608
KILT Wow (dev, 3k)	Llama2-70B	0.000	0.713	0.001	0.685	0.000	0.639	0.000	0.644	0.001	0.602	0.000	0.607	0.001	0.639	0.001	0.631
	Llama2-7B	0.000	0.677	0.002	0.622	0.001	0.480	0.000	0.474	0.000	0.435	0.001	0.462	0.001	0.515	0.000	0.498
	Llama3-8B	0.000	0.530	0.001	0.542	0.000	0.452	0.000	0.465	0.000	0.370	0.001	0.421	0.000	0.491	0.000	0.484
	Mixtral-8x7B	0.000	0.765	0.001	0.773	0.000	0.713	0.000	0.663	0.000	0.659	0.001	0.683	0.000	0.732	0.000	0.726
	SOLAR-10.7B	0.000	0.808	0.002	0.747	0.000	0.612	0.001	0.609	0.001	0.527	0.001	0.558	0.000	0.640	0.000	0.623
	TinyLlama-1.1B	0.000	0.461	0.001	0.321	0.000	0.242	0.000	0.238	0.000	0.215	0.000	0.229	0.001	0.268	0.001	0.255
POPQA (test, 15.3k)	Llama2-70B	0.327	0.366	-	-	0.618	0.598	0.672	0.635	0.410	0.419	0.484	0.484	0.605	0.588	0.655	0.625
	Llama2-7B	0.226	0.257	-	-	0.562	0.527	0.610	0.565	0.375	0.374	0.449	0.438	0.558	0.535	0.602	0.560
	Llama3-8B	0.242	0.276	-	-	0.616	0.559	0.664	0.599	0.383	0.377	0.468	0.455	0.599	0.568	0.657	0.601
	Mixtral-8x7B	0.397	0.415	-	-	0.631	0.579	0.685	0.615	0.397	0.412	0.484	0.472	0.620	0.588	0.679	0.619
	SOLAR-10.7B	0.307	0.392	-	-	0.660	0.579	0.720	0.632	0.410	0.386	0.508	0.463	0.645	0.585	0.712	0.631
	TinyLlama-1.1B	0.152	0.170	-	-	0.386	0.344	0.421	0.379	0.293	0.275	0.341	0.310	0.398	0.357	0.435	0.422
SCIQ (test, 1k)	Llama2-70B	0.586	0.833	-	-	0.598	0.830	0.596	0.844	0.563	0.779	0.594	0.822	0.597	0.833	0.610	0.851
	Llama2-7B	0.468	0.756	-	-	0.508	0.714	0.517	0.760	0.445	0.659	0.520	0.717	0.515	0.721	0.514	0.753
	Llama3-8B	0.526	0.775	-	-	0.516	0.744	0.538	0.777	0.458	0.647	0.532	0.755	0.521	0.750	0.541	0.786
	Mixtral-8x7B	0.657	0.900	-	-	0.592	0.841	0.614	0.867	0.544	0.793	0.595	0.839	0.576	0.852	0.599	0.854
	SOLAR-10.7B	0.637	0.902	-	-	0.586	0.821	0.616	0.857	0.519	0.746	0.599	0.836	0.589	0.846	0.618	0.872
	TinyLlama-1.1B	0.221	0.526	-	-	0.372	0.436	0.391	0.514	0.300	0.353	0.370	0.446	0.372	0.463	0.415	0.507
WIKIQA (test, 6.1k)	Llama2-70B	0.000	0.844	-	-	0.008	0.901	0.004	0.889	0.008	0.786	0.004	0.827	0.004	0.885	0.008	0.918
	Llama2-7B	0.004	0.745	-	-	0.004	0.786	0.004	0.794	0.000	0.584	0.004	0.695	0.004	0.765	0.004	0.782
	Llama3-8B	0.000	0.745	-	-	0.000	0.786	0.008	0.840	0.000	0.543	0.004	0.716	0.004	0.802	0.004	0.844
	Mixtral-8x7B	0.000	0.881	-	-	0.004	0.864	0.012	0.885	0.008	0.700	0.004	0.794	0.004	0.914	0.008	0.909
	SOLAR-10.7B	0.004	0.893	-	-	0.004	0.872	0.004	0.885	0.004	0.671	0.004	0.790	0.004	0.914	0.004	0.922
	TinyLlama-1.1B	0.000	0.465	-	-	0.008	0.362	0.008	0.383	0.004	0.198	0.004	0.329	0.008	0.412	0.004	0.428
TruthfulQA (dev, 0.8k)	Llama2-70B	0.033	0.520	-	-	0.064	0.457	0.075	0.501	0.061	0.427	0.065	0.465	0.064	0.496	0.072	0.496
	Llama2-7B	0.035	0.450	-	-	0.051	0.345	0.058	0.378	0.048	0.308	0.059	0.356	0.043	0.370	0.058	0.365
	Llama3-8B	0.020	0.446	-	-	0.045	0.392	0.054	0.425	0.040	0.349	0.044	0.419	0.040	0.431	0.045	0.419
	Mixtral-8x7B	0.056	0.706	-	-	0.066	0.586	0.080	0.578	0.062	0.508	0.075	0.559	0.066	0.580	0.082	0.574
	SOLAR-10.7B	0.055	0.701	-	-	0.049	0.474	0.054	0.526	0.040	0.435	0.051	0.485	0.045	0.531	0.050	0.512
	TinyLlama-1.1B	0.034	0.262	-	-	0.023	0.208	0.033	0.213	0.028	0.175	0.028	0.180	0.028	0.217	0.038	0.217

Table 4: Zero-shot performance on various datasets with varying retrieval systems, where RR stands for additional re-ranking with DeBERTa-v3. Match metric (M) and LLMeval (LLM)

Method	Dataset				
	ELI5	HotpotQA	NQ	TriviaQA	WoW
BM25	0.132	0.580	0.531	0.508	0.447
BM25+RR	0.198	0.680	0.709	0.617	0.528
RetroMAE	0.241	0.522	0.753	0.571	0.648
RetroMAE+RR	0.257	0.628	0.822	0.645	0.675
SPLADE-v3	0.240	0.645	0.799	0.641	0.688
SPLADE-v3+RR	0.264	0.704	0.833	0.663	0.684

Table 5: Retrieval Performance (R@5) on KILT QA tasks with different retrieval systems, where RR indicates additional re-ranking using DeBERTa-v3.

Model	`Checkpoint`
Sparse	
BM25 (Robertson et al., 1994)	-
SPLADE++ (Formal et al., 2022)	`naver/splade-cocondenser-selfdistil`
SPLADE-v3 (Lassance et al., 2024)	`naver/splade-v3`
Dense (MS MARCO)	
TAS-B (Hofstätter et al., 2021)	`sebastian-hofstaetter/distilbert-dot-tas_b-b256-msmarco`
CoCondenser (Gao and Callan, 2022)	`Luyu/co-condenser-marco-retriever`
Contriever (Izacard et al., 2022a)	`facebook/contriever-msmarco`
RetroMAE (Shitao et al., 2022)	`Shitao/RetroMAE_MSMARCO_distill`
DRAGON+ (Lin et al., 2023)	`facebook/dragon-plus-context-encoder`
	`facebook/dragon-plus-query-encoder`
Dense (MTEB)	
GTE (Li et al., 2023)♣	`Alibaba-NLP/gte-base-en-v1.5`
	`Alibaba-NLP/gte-large-en-v1.5`
BGE (Xiao et al., 2023)♣	`BAAI/bge-small-en-v1.5`
	`BAAI/bge-base-en-v1.5`
	`BAAI/bge-large-en-v1.5`
E5 (Wang et al., 2024a)♣	`intfloat/e5-small-v2`
	`intfloat/e5-base-v2`
	`intfloat/e5-large-v2`
AnglE (Li and Li, 2024)♣	`WhereIsAI/UAE-Large-V1`
MXBAI Embed (Lee et al., 2024)†	`mixedbread-ai/mxbai-embed-large-v1`
Nomic Embed (Nussbaum et al., 2024)♣	`nomic-ai/nomic-embed-text-v1`
Jina Embed (Günther et al., 2023)♣	`jinaai/jina-embeddings-v2-base-en`
Arctic Embed (Merrick et al., 2024)♣	`Snowflake/snowflake-arctic-embed-l`

Table 6: Retrieval Systems and corresponding HuggingFace checkpoints. We include standard dense and sparse approaches trained on the MS MARCO passage ranking dataset (Bajaj et al., 2018). We further include recent models that report strong performance on the MTEB benchmark (Muennighoff et al., 2023)[9]. These models are usually fine-tuned on a larger pool of annotated datasets, which include MS MARCO but also QA datasets like NQ. In such a case, the RAG performance evaluated on datasets like KILT NQ is not "zero-shot". ♣ *indicates that models have been explicitly fine-tuned on NQ. Note that MXBAI Embed is excluding MTEB data from its training set – but relies on proprietary data*†.

```
system: "You are a helpful assistant. Your task is to extract relevant
information from provided documents and to answer to questions as briefly as
possible."
user: f"Background:\n{docs}\n\nQuestion:\ {question}"
```

For closed-book experiments, where no context is provided to the LLMs we used a simple prompt:

```
system: "You are a helpful assistant. Answer the questions as briefly as
possible."
user: f"Question:\ {question}"
```

Model	Re-ranking R@5 (↓)	Re-ranking LLMEval	Ranking R@5	Ranking LLMEval
BM25	0.709	0.716	0.531	0.574
TAS-B	0.821	0.783	0.728	0.698
RetroMAE	0.822	0.792	0.753	0.731
CoCondenser	0.825	0.783	0.744	0.715
SPLADE++	0.827	0.803	0.778	0.754
DRAGON+	0.833	0.793	0.791	0.753
SPLADE-v3	0.833	0.795	0.799	0.768
Contriever	0.837	0.793	0.783	0.728
jina-embeddings-v2-base-en♣	0.837	0.804	0.795	0.750
gte-base-en-v1.5♣	0.846	0.809	0.823	0.782
snowflake-arctic-embed-l♣	0.847	0.819	0.830	0.787
bge-small-en-v1.5♣	0.849	0.810	0.786	0.754
bge-base-en-v1.5♣	0.854	0.809	0.808	0.756
nomic-embed-text-v1♣	0.854	0.809	0.843	0.789
bge-large-en-v1.5♣	0.854	0.815	0.821	0.788
mxbai-embed-large-v1†	0.855	0.811	0.830	0.780
AnglE♣	0.856	0.815	0.834	0.789
gte-large-en-v1.5♣	0.858	0.813	0.854	0.790
e5-small-v2♣	0.864	0.813	0.856	0.788
e5-base-v2♣	0.866	0.808	0.870	0.805
e5-large-v2♣	0.867	0.822	0.883	0.808

Table 7: RAG performance (LLMEval) on NQ for SOLAR-10.7B for various retrievers w/ re-ranking (DeBERTa-v3). We sort models by ascending R@5 (re-ranking performance). ♣ *indicates that models have been explicitly fine-tuned on NQ*. Note that re-ranking even hurts E5 retrieval's effectiveness – indicating that the model captured NQ's ranking signals well.

F LLMeval: LLM-based Answer Equivalence Evaluation

For LLM eval we leverage the SOLAR-10.7B-Instruct-v1.0 by providing the question, reference answer, and the generated candidate answer to the model and asking the model to judge based on the following prompt:

```
f"You are an evaluation tool. Just answer by {{Yes}} or {{No}}. Here is a
question, a golden answer and an AI-generated answer. Judge whether the
AI-generated answer is correct according to the question and golden answer,
answer with {{Yes}} or {{No}}.\nQuestion: {question}.\nGolden answer:
{answer}\nGenerated answer: {prediction} Response: {{"
```

Based on this instruction the model generates "true" or "false", yielding in binary labels. In cases where the model generates any other tokens we default to "false". Upon manual inspection, we found this to be the case very rarely. To speed up inference we use vLLM. We also tried extracting the logits for "true" or "false" to obtain a continuous score between 0 and 1 but found this to perform comparably to directly generating a single token ("true" or "false").

G Training Details

In Table 8 we list the Hyperparameters used for our fine-tuning experiments.

H Multilingual RAG

To promote experimentation with RAG in multilingual settings, we incorporate components needed to support multilingual datasets in *BERGEN*, for 12 non-English languages[10]. Our goal is to build a strong baseline for zero-shot multilingual RAG which could be used in future works for experimentation with new approaches.

[10]Arabic, Simplified Chinese, Finnish, French, German, Italian, Japanese, Korean, Portuguese, Russian, Spanish, Thai.

Hyperparameter	Assignment
learning Rate	1e-4
lr scheduler type	linear
warmup ratio	0.05
weight dacay	0.1
batch size	max. possible
optimizer	AdamW
epochs	1
LoRa layers	all linear layers
LoRa alpha	64
LoRa dropout	0.1
LoRa r	32
LoRa bias	None
num GPUs	1
GPU	A100 80GB
retriever(s)	SPLADE-v3 (+ DeBERTa-v3)
num passages	5

Table 8: Hyperparameters for Fine-tuning

Multilingual Retrieval. Multilinguality in RAG comes in two faces: non-English user queries and non-English datastores. Such a setting requires a strong retriever and reranker, which supports both monolingual and cross-lingual retrieval. The former case corresponds to the user query and the datastore being in the same language. The latter case corresponds to retrieving from the datastore in a language different from the language of the user query. We also consider a scenario with a multilingual datastore. We pick the recently released (and publicly available) BGE-M3 model[11] (Chen et al., 2024) which provides all listed functionalities and includes all languages we consider in its training data.

Multilingual Generation. We rely on the Command-R-35B[12] model as a generator for multilingual experiments in *BERGEN*. Command-R-35B has been developed with keeping RAG application in mind and officially supports 11 languages[13], including most of our considered languages, and also includes 13 more languages (incl. Russian) in pretraining but not instruction tuning.

Recent studies (Ye et al., 2023) show that even English-centric LLMs possess multilingual understanding and generation capabilities. As a result, they can also be used for multilingual experiments, especially with auxiliary system prompts, as described below.

System Prompt. In our preliminary experiments, we found that models sometimes reply in English even when prompted in non-English. For example, Command-R, augmented with the English retrieved context and prompted in non-English, replies in English in $\sim 50\%$ of cases. For English-centric models, such a behavior happens frequently even with no context or same-language context. To enable generation in the user language (expected behavior), we augment the model's system prompt with an explicit instruction to generate in the given language and also translate the system prompt into user languages[14]. We found that this combination enables the highest chance of generation in the user language for all models.

Datasets. Following (Asai et al., 2021b), we use MKQA (Longpre et al., 2021) and XOR-TyDi QA (Asai et al., 2021a) datasets for multilingual evaluation in our experiments. MKQA consists of $10k$ examples

[11]Retriever: `https://huggingface.co/BAAI/bge-m3` (dense version). Reranker: `https://huggingface.co/BAAI/bge-reranker-v2-m3`.

[12]`https://huggingface.co/CohereForAI/c4ai-command-r-v01`

[13]Command-R official languages: Arabic, Brazilian Portuguese, English, French, German, Italian, Japanese, Korean, Simplified Chinese, and Spanish.

[14]We translate system prompts using Google Translate and ask employees of our laboratory, native or fluent in given languages, to check translated prompts.

from the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019), translated into 25 languages. This dataset is therefore parallel between languages and grounds knowledge primarily in English Wikipedia. In our experiments we select a subset of $2.7k$ samples, overlapping between MKQA and KILT NQ datasets[15], thus recovering relevant passages information from KILT NQ. XOR-TyDi QA comprises $40k$ information-seeking questions in 7 languages (of which we use $3k$ validation questions) and grounds questions in Wikipedia in the same language as the question or in English. To provide English for comparison, we include results for English on the TyDi QA dataset (Clark et al., 2020).

Datastore. We follow (Asai et al., 2021b) and (Karpukhin et al., 2020) and construct passages by splitting Wikipedia article into chunks of 100 words (or 100 Unicode characters for non whitespace separated languages, namely Chinese, Japanese, and Thai) and prepending the article title to each chunk. In most of the experiments, we retrieve either from English Wikipedia (KILT version[16]) or Wikipedia in the user language[17], but we also experiment with retrieving from a concatenation of the two mentioned Wikipedias and from Wikipedia in all considered languages.

Metrics. In our preliminary experiments, we noticed a pattern arising sometimes in the scenario with cross-lingual retrieval, when models generate a transliteration of named entities in other languages different from the one contained in the ground-truth label. This is not a weakness of the system, but needs to be accounted for in the evaluation metric. Since word-level matching fails to capture similarity in the described case, we propose to evaluate *recall on character n-gram level*. We first split ground-truth labels into tokens, extract all character 3-grams from each token, and evaluate which percentage of such n-grams is present in the model-generated response –see Table 11 for illustration.

In addition to the task metric, we also control the correct language rate, CLR, which measures which percentage of model outputs are written in the user language. We detect languages using `fasttext` library (Joulin et al., 2017, 2016) and its `lid.176.bin` model[18]. Due to high erroneous level of language identification for short sequences, we only evaluate the CRL metric for model responses longer than 20 characters.

The experimental setting is the same as in English experiments, e.g. we use greedy decoding, retrieve top-50 passages, and use re-ranking after retrieval.

Table 12 reports correlation between LLMeval metric and other surface-based metrics, including *recall on character n-gram level*. We notice that overall character-level recall correlates better with LLMeval metric. This is even more striking for non latin-script languages. It worth noting that overall the correlation between LLMeval and Char3-recall is relatively low. Manual inspection of the results highlights that LLMeval only assess whether an answer is valid or not, even if was not generated in the same language as query or gold label. Further research is required to better design reliable multilingual evaluation metrics.

Results. Tables 9 and 10 reports results with two multilingual datasets and various retrieval options: retrieval from English Wikipedia, from Wikipedia in the user language, from their concatenation, or from the concatenation of Wikipedia in all languages. In the latter two cases with run retrieval over the embeddings of passages in multiple languages, so that the selected passages may be also in multiple languages.

Comparing retrieval from English and user language, we observe different behavior on the two considered datasets. On the MKQA dataset, retrieval from English is more beneficial, which is expected since questions in MKQA were initially written by relying on the English Wikipedia and then translated into other languages. At the same time, XOR-TyDi QA includes questions grounded in both English and user languages (see statistics in Table 2, Longpre et al., 2021), and we observe that retrieval from Wikipedia in the user language is more beneficial.

Overall, we find that BGE-M3 also successfully manages to retrieve from the concatenated multilingual Wikipedia and thus dynamically choose the appropriate datastore, often reaching performance higher than

[15]NQ dataset in KILT benchmark available at https://huggingface.co/datasets/kilt_tasks
[16]https://huggingface.co/datasets/facebook/kilt_wikipedia
[17]https://huggingface.co/datasets/wikimedia/wikipedia
[18]https://fasttext.cc/docs/en/language-identification.html

with any of the two monolingual Wikipedias.

	No retrieval	Retrieval from Wiki in English	User lang	English+UL	All langs
MKQA					
English	58.4	**70.2**	—	—	68.5
Arabic	26.4	45.9	36.3	**49.0**	48.2
Chinese	21.4	29.1	22.5	27.2	**31.0**
French	48.4	62.6	56.3	65.0	**66.2**
Finnish‡	29.7	55.8	45.2	59.8	**60.7**
German	47.8	64.6	54.8	65.5	**66.9**
Italian	51.5	61.2	56.8	64.8	**66.3**
Japanese	31.7	**42.7**	28.8	40.2	42.1
Korean	21.5	32.2	31.5	**38.4**	38.1
Portuguese	48.4	62.3	54.9	65.2	**66.9**
Russian†	38.1	55.0	51.0	**61.0**	59.4
Spanish	52.5	63.3	57.3	65.7	**67.1**
Thai‡	12.4	23.7	10.1	23.2	**24.5**
XOR TyDi QA					
English	47.5	**64.2**	—	—	59.4
Arabic	47.7	52.9	65.5	66.6	**66.8**
Finnish‡	30.8	45.2	58.9	**60.9**	59.1
Japanese	21.0	25.2	30.0	24.8	**31.8**
Korean	31.0	33.4	40.8	40.0	**41.8**
Russian†	40.5	53.9	62.3	63.8	**64.6**

Table 9: Metric: **character 3-gram recall**. Performance of mRAG for various languages on MKQA and XOR-TyDi QA datasets (TyDi QA for English), with different retrieval options. Retriever: BGE-M3. Reranker: BGE-M3 Generator: Command-R-35B. Prompt: translated into user languages with an instruction to generate in the given user language (UL). † denotes languages included in Command-R pretraining but not instruction tuning. ‡ denotes languages not included in Command-R pretraining nor tuning. *RAG brings substantial performance improvement in all languages, and retrieval from multilingual Wikipedia is beneficial in most cases.*

	No retrieval	Retrieval from Wiki in English	User lang	English+UL	All langs
MKQA					
English	0.67	**0.76**	–	–	0.74
Arabic	0.29	0.54	0.41	0.56	**0.57**
Chinese	0.37	0.60	0.39	0.58	**0.61**
French	0.48	0.63	0.55	0.64	**0.65**
Finnish‡	0.32	0.58	0.47	0.62	**0.64**
German	0.47	0.64	0.55	**0.66**	**0.66**
Italian	0.52	0.61	0.54	0.63	**0.64**
Japanese	0.37	0.63	0.36	0.59	**0.64**
Korean	0.32	0.59	0.45	**0.62**	**0.62**
Portuguese	0.51	0.63	0.55	0.65	**0.67**
Russian†	0.48	0.71	0.58	**0.72**	**0.72**
Spanish	0.55	0.65	0.58	0.66	**0.68**
Thai‡	0.34	**0.59**	0.22	0.57	**0.59**
XOR TyDi QA					
English	0.63	0.73	–	–	**0.69**
Arabic	0.56	0.57	0.68	0.69	**0.70**
Finnish‡	0.41	0.59	0.72	**0.74**	0.74
Japanese	0.40	0.51	0.52	0.49	**0.62**
Korean	0.54	0.58	0.64	0.64	**0.66**
Russian†	0.47	0.65	0.71	0.73	**0.74**

Table 10: Metric: LLMeval. Performance of mRAG for various languages on MKQA and XOR-TyDi QA datasets (TyDi QA for English), with different retrieval options. Retriever: BGE-M3. Reranker: BGE-M3 Generator: Command-R-35B. Prompt: translated into user languages with an instruction to generate in the given user language (UL). † denotes languages included in Command-R pretraining but not instruction tuning. ‡ denotes languages not included in Command-R pretraining nor tuning. *RAG brings substantial performance improvement in all languages, and retrieval from multilingual Wikipedia is beneficial in most cases.*

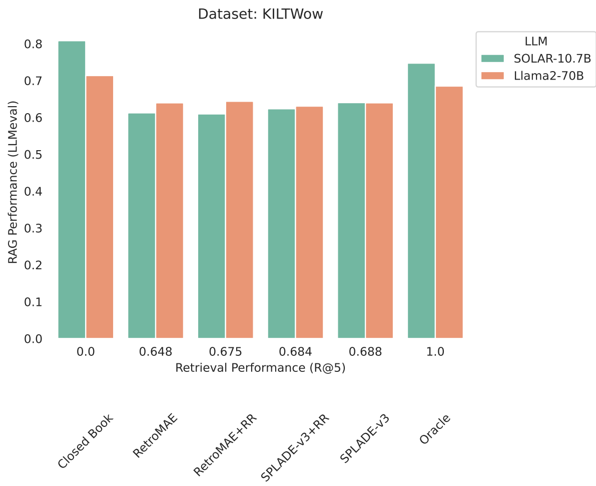
	Text	**Character 3-grams**
Ground truth	sofya kovalevskaya	[sof ofy fya kov ova val ale lev evs vsk ska kay aya]
Model response	sofia kovalevskaia	[sof ofi fia kov ova val ale lev evs vsk ska kai aia]
Recall	0	$9/13 = 69.2\%$

Table 11: Illustration of the proposed character 3-gram recall metric, designed to be more robust to different possible transliterations of named entities. Tokens matching between groundtruth and model response are underlined.

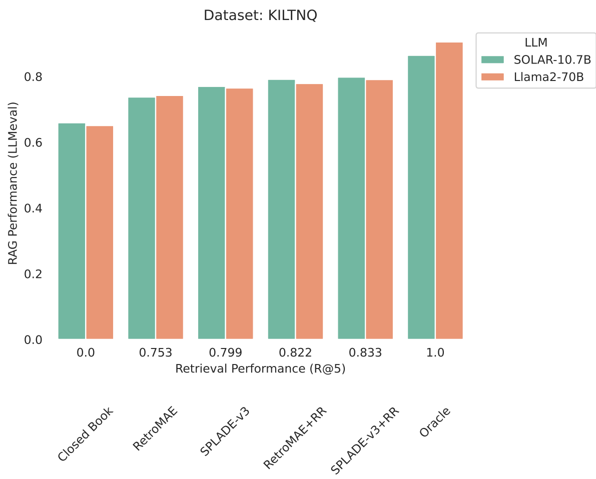
	Recall	Rouge-1	Rouge-L	Char3-recall	Match	LID
English	0.34	0.37	0.37	0.45	0.43	0.06
Arabic	0.35	0.37	0.37	0.47	0.40	0.18
Chinese	0.07	0.07	0.07	0.37	0.33	0.13
French	0.34	0.40	0.40	0.48	0.46	0.11
Finnish	0.36	0.39	0.39	0.51	0.46	0.21
German	0.39	0.40	0.40	0.48	0.45	0.10
Italian	0.37	0.41	0.41	0.48	0.47	0.01
Japanese	0.15	0.15	0.15	0.43	0.43	0.10
Korean	0.34	0.33	0.33	0.44	0.42	0.11
Portuguese	0.35	0.41	0.41	0.47	0.45	0.07
Russian	0.29	0.31	0.31	0.42	0.32	0.13
Spanish	0.37	0.40	0.40	0.47	0.45	0.01
Thai	0.20	0.21	0.21	0.22	0.21	0.09

Table 12: Kendall-Tau correlation between surface-based metrics and LLMeval metric.

(a)



(b)



(c)

Figure 6: Comparison of suitable and non-suitable datasets for RAG evaluation. For datasets Eli5 (a) and WoW (b) the Closed Book setting (no retrieval) is much better than the Oracle making them less suitable. On the other hand for NQ (c) Oracle is much better than Closed Book, and retrieval improves generation quality depending on their effectiveness. This makes it a suitable dataset for RAG evaluation.

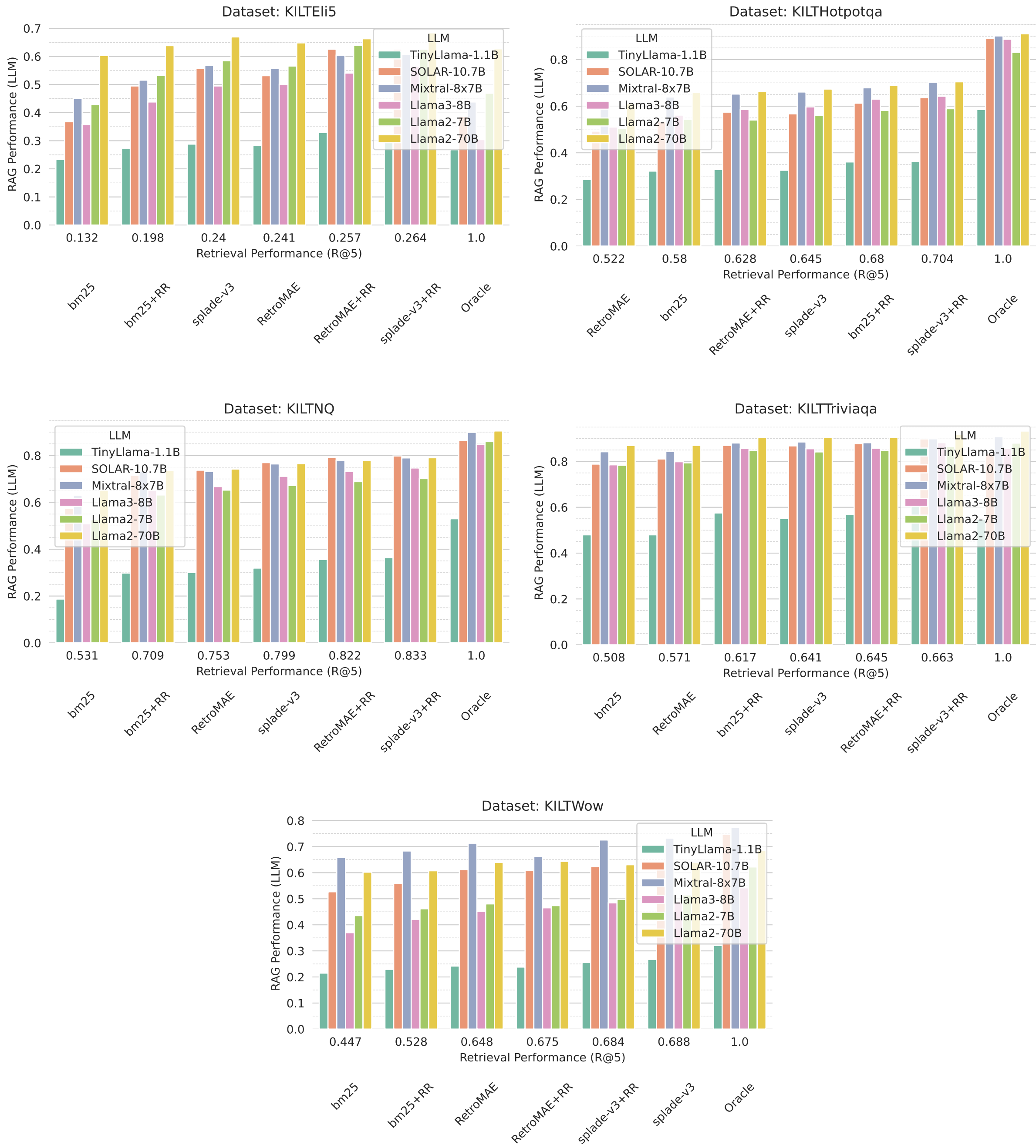


Figure 7: Impact of retrieval performance on different LLMs for zero-shot RAG.