Retrieval Helps or Hurts? A Deeper Dive into the Efficacy of Retrieval Augmentation to Language Models

Seiji Maekawa **Hayate Iso** **Sairam Gurajada** **Nikita Bhutani**
Megagon Labs
{seiji,hayate,sairam,nikita}@megagon.ai

Abstract

While large language models (LMs) demonstrate remarkable performance, they encounter challenges in providing accurate responses when queried for information beyond their pretrained memorization. Although augmenting them with relevant external information can mitigate these issues, failure to consider the necessity of retrieval may adversely affect overall performance. Previous research has primarily focused on examining how entities influence retrieval models and knowledge recall in LMs, leaving other aspects relatively unexplored. In this work, our goal is to offer a more detailed, fact-centric analysis by exploring the effects of combinations of entities and relations. To facilitate this, we construct a new question answering (QA) dataset called WiTQA (Wikipedia Triple Question Answers). This dataset includes questions about entities and relations of various popularity levels, each accompanied by a supporting passage. Our extensive experiments with diverse LMs and retrievers reveal when retrieval does not consistently enhance LMs from the viewpoints of fact-centric popularity. Confirming earlier findings, we observe that larger LMs excel in recalling popular facts. However, they notably encounter difficulty with infrequent entity-relation pairs compared to retrievers. Interestingly, they can effectively retain popular relations of less common entities. We demonstrate the efficacy of our finer-grained metric and insights through an adaptive retrieval system that selectively employs retrieval and recall based on the frequencies of entities and relations in the question.[1]

1 Introduction

Large language models (LMs) (Brown et al., 2020; OpenAI, 2023) have exhibited impressive capabilities owing to their ability to retrieve knowledge memorized during pre-training (Sanh et al., 2022; Wei et al., 2022; Ouyang et al., 2022). However, despite the increase in model size and complexity, LMs remain susceptible to factual inaccuracies in knowledge-intensive tasks such as open domain question answering and natural language generation, which demand access to a broader spectrum of information (Petroni et al., 2021; Chen et al., 2017; Lin et al., 2022). Retrieval-Augmented Language Models (RALMs) (Guu et al., 2020; Lewis et al., 2020; Izacard and Grave, 2021) have emerged as a promising solution for mitigating factual errors by incorporating relevant external information.

Nevertheless, recent studies suggest that RALMs are not a universal solution (Petroni et al., 2020; Li et al., 2023). Indiscriminately augmenting LMs with irrelevant passages can override potentially correct knowledge already possessed by the LM, resulting in incorrect responses (illustrated in Table 1). A robust RALM is characterized by its ability to accurately recall its prior knowledge while selectively incorporating retrieved information only when necessary.

Determining when to recall and when to retrieve external information thus requires a thorough investigation of the following questions:

1. Under what conditions LMs can recall correctly and what factors influence their ability?
2. When retrieval augmented models make errors and what factors affect their performance?
3. Are there any common error patterns between LMs and retrieval models responses?

While previous research has investigated factors influencing memorization in LMs as well as performance of retrievers, they have a few limitations: a) They solely focus on entities (Sun et al., 2023; Mallen et al., 2023), whereas real-world information comprises both entities and relations . b) They primarily focus exclusively on either retrievers or

[1]The code and data are available at https://github.com/megagonlabs/witqa.

Triple: (Chicago, country, United States of America) **Entity Popularity:** $95.0\%ile$
Question: What country is Chicago located in? **Entity-Relation Popularity:** $97.4\%ile$
LM Answer: United States [Correct]
Context: The Chicago Municipal Tuberculosis Sanitarium was located in Chicago, Illinois, USA.... [Correct Retrieval]
RALM Answer: USA [Correct]

Triple: (George H.W. Bush, educated at, Yale University) **Entity Popularity:** $89.5\%ile$
Question: What educational institution did George H.W. Bush attend? **Entity-Relation Popularity:** $41.8\%ile$
LM Answer: Yale University [Correct]
Context: The George H.W. Bush Presidential Library is located on a site on the west campus of Texas A&M University in College Station, Texas.... [Wrong Retrieval]
RALM Answer: Texas A&M University [Wrong]

Triple: (Ellen Litman, educated at, University of Pittsburgh) **Entity Popularity:** $10.3\%ile$
Question: What educational institution was Ellen Litman educated at? **Entity-Relation Popularity:** $17.9\%ile$
LM Answer: Stanford University [Wrong]
Context: Ellen Litman Ellen Litman (born 1973) is an American novelist. She received the Rona Jaffe Foundation Writers' Award in 2006. Born in Moscow, Russia, she emigrated with her parents in 1992 to Pittsburgh, Pennsylvania. She was educated at the University of Pittsburgh and earned a B.S. in Information Science. ... [Correct Retrieval]
RALM Answer: University of Pittsburgh [Correct]

Table 1: QA examples from WiTQA with predictions of varying popularity of question entity and entity-relation pair. We show the predictions from LM (GPT-3.5) with no augmentation and RALM (GPT-3.5+BM25). In the top row, both LM and RALM provide correct answers for the popular question. In the middle row, LM generates correct answer but RALM provides incorrect answer due to retrieval errors. In the bottom row, LM provides incorrect answer for an infrequent entity-relation pair.

recall in LMs, neglecting the interplay between them (Petroni et al., 2019; Sciavolino et al., 2021; Liu et al., 2023).

In this work, we aim to provide a finer-grained (fact-centric) analysis by investigating the impact of combinations of entities and relations on the performance of RALMs. We focus on question answering (QA) task, where we analyze 10 LMs of varying sizes with 5 different retrieval settings. To facilitate this, we require a QA benchmark that not only provides valid supporting passages for each QA pair but also integrates indicators for memorization in LMs. Additionally, the benchmark should encompass entities and relations of varying popularity. However, existing benchmarks such as PopQA (Mallen et al., 2023) and EntityQuestions (Sciavolino et al., 2021) are not suitable for this purpose as they are entity-centric and target long-tailed information. To facilitate fact-centric analysis, we develop a novel dataset called WiTQA (Wikipedia Triple Question Answers). We sample triples extracted from Wikipedia, considering the popularity of entities and entity-relation combinations. We then generate QA pairs for each triple, ensuring that each example in WiTQA is accompanied by supporting passages and popularity scores for the question entity-relation pair.

Our investigation of RALMs zero-shot performance on the proposed benchmark dataset (WiTQA) yields the following key findings:

- Even without retrieval, LMs can correctly recall entity-relation pairs frequently encountered during pre-training. Nonetheless, this capability is notably contingent on the model's size. Larger models can acquire long-tailed relations about renowned entities. However, there is still a significant drop in overall performance when addressing minor facts.

- For long-tailed entity-relation pairs, retrievers show more robust performance compared to recall abilities of LMs, suggesting that augmentation for such cases is beneficial. However, this observation does not extend to well-known entity-relation pairs, potentially leading to override issues.

- LMs achieve higher accuracy than retrievers for well-known entity-relation pairs regarding long-tailed entities while previous studies report that large LMs struggle with long-tailed entities.

Our findings reveal when retrieval does not assist LMs through the lens of fact-level popularity. Leveraging this insight, we propose a selective memory integration, which selectively employs retrieval augmentation and LMs' memory based on the frequencies of entities and relations. Our experiments demonstrate that this approach can enhance QA performance by up to 10.1%.

2 Background

2.1 Open domain Question Answering

Open domain question answering is a knowledge-intensive task that involves generating an answer as an output a given a question q as an input. This task typically involves retrieving relevant passage p based on the given question q using a retrieval

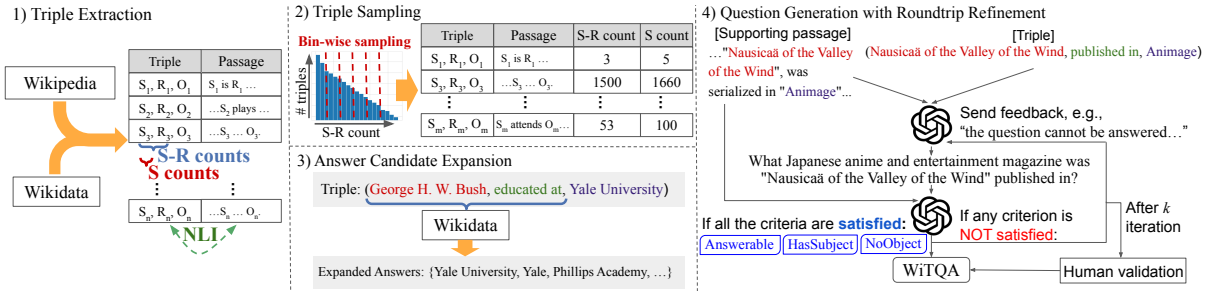

Figure 1: Overview of WiTQA dataset creation. First, we extract triples from Wikipedia and Wikidata, and compute the frequency of subject-relation pairs and subject entity (referred to as S-R counts and S counts) (§3.1.1). Second, we sample triples based on different ranges of S-R counts and select supporting passages based on entailment scores(§3.1.2). Third, we expand answer candidates using Wikidata (§3.1.3). Finally, we generate questions from triples and iteratively refine generated questions (§3.1.4).

model $\mathcal{M}_{\texttt{ret}}$. Subsequently, the retrieved passage is used to model the answer a using a reader model $\mathcal{M}_{\texttt{read}}$: $\mathcal{M}_{\texttt{read}}(a|q,p)$.

2.2 Parametric vs Non-Parametric Knowledge

Although, in the early study of the open domain question answering, a reader model $\mathcal{M}_{\texttt{read}}$ always relied on the external passage p to reliably answer the question q (Chen et al., 2017; Radford et al., 2019; Lee et al., 2019; Guu et al., 2020; Lewis et al., 2020), recent LM-based reader models started showing the potential to answer the question without relying on the external passage p: $\mathcal{M}_{\texttt{read}}(a|q,p=\phi)$ (Petroni et al., 2019; Roberts et al., 2020; Brown et al., 2020; Mallen et al., 2023; Yu et al., 2023a; Kandpal et al., 2023; Kang and Choi, 2023; Shi et al., 2023). This has actually shed light on the prospect that retrieval model $\mathcal{M}_{\texttt{ret}}$ can have a negative impact on RALMs. For example, Li et al. (2023) showcased that if the irrelevant passage is provided to the LMs, the parametric knowledge in LM is overridden by the non-parametric knowledge in the passage p. In the context of open domain question answering tasks, this suggests that the LM-based reader model $\mathcal{M}_{\texttt{read}}$ may produce incorrect answer $\bar{a}$ because it can be misguided by the irrelevant passage $\bar{p}$ provided by the retrieval model $\mathcal{M}_{\texttt{ret}}$.

To address this issue, various approaches have been proposed (Yu et al., 2023b; Mallen et al., 2023; Yoran et al., 2023; Asai et al., 2024). For instance, Mallen et al. (2023) reported that combining LMs and RALMs based on the entity popularity can yield improved QA accuracy. Asai et al. (2024) built a training dataset to determine when to engage in retrieval and to assess the relevance of retrieved passages using GPT-4 and then fine-tuned smaller-scale models to replicate similar behavior.

However, none of the above addresses the distinct strengths and weaknesses of LMs and retrieval models, nor have they clarified the causes of errors when these systems are interconnected. This primarily stems from the lack of datasets designed for such in-depth analysis. Thus, we introduce the WiTQA dataset, specifically created to analyze the error patterns of LMs and retrieval models, both independently and in tandem.

3 The WiTQA dataset

The WiTQA dataset is meticulously constructed with the underlying assumption that LMs are likely to recall facts frequently mentioned in their pre-training corpus (Petroni et al., 2019; Jiang et al., 2020). We curate question-answer pairs annotated with the frequency of mentions of entities and their relationships within Wikipedia—a predominant source of pre-training for LMs—along with their supporting passages. This enables us to explore the correlation between popularity in the pre-training corpus and the performance of LMs and retrieval models individually. Furthermore, when operating in tandem, it facilitates the analysis of whether RALM's errors stem from LM's reasoning or primarily arise from retrieval errors.

3.1 Dataset creation

Our process for creating the dataset involves four key steps: 1) extraction of triples from Wikipedia, 2) sampling of triples, 3) expansion of answer candidates, and 4) generation of questions with round-trip refinement, as illustrated in Figure 1.

3.1.1 Triple extraction

We first extract triples from Wikipedia to estimate the popularity of the subject entity (S count) and the co-occurrence of the subject entity and rela-

Dataset	Page view	S count	S-R count	Supporting passages	# of Relation Type	Question form
EntityQuestions (Sciavolino et al., 2021)	✗	✗	✗	✗	24	Template
PopQA (Mallen et al., 2023)	✓	✗	✗	✗	16	Template
WiTQA (Ours)	✓	✓	✓	✓	32	Model-assisted

Table 2: Summary of question-answering datasets. WiTQA includes question popularity indicators and valid supporting passages sourced from Wikipedia. It contains more diverse relation types. It employs a model-assisted approach for question generation, leading to a more versatile verbalization of triples.

tion predicate (S-R count) in the Wikipedia corpus.[2] However, building a scalable and robust information extraction system is a long-standing challenge (Chia et al., 2022; Kim et al., 2023). To address this, we opted to leverage the Wikipedia hyperlink and Wikidata for rule-based triple extraction (Elsahar et al., 2018; Huguet Cabot and Navigli, 2021). Specifically, we extract a list of entities from the Wikipedia abstract, map them to Wikidata ID using Wikimapper,[3] and finally extract all the triples mentioned in the text by matching them with the Wikidata database.

Given a list of extracted triples, we compute S count for each unique subject entity in the list. Similarly, we compute S-R count for each unique subject entity-relation pair in the list. Additionally, for each passage-triple pair, we compute entailment score using NLI models.[4] For a given triple, the passage with the highest entailment score is designated as the supporting passage. Please refer to Appendix A for more details.

3.1.2 Triple sampling

Concerning subject-relation (S-R) counts, the distribution of triples follows a long-tail pattern. To ensure dataset diversity, we employed sampling based on S-R counts. Specifically, we manually sampled 32 relations and categorized triples into intervals such as $[1, 5)$, $[5, 10)$, $[10, 50)$, $[50, 100)$, $[100, 500)$, $[500, 1000)$, and $1000+$. Subsequently, we sampled up to 200 triples for each relation within each interval.

3.1.3 Answer candidate expansion

Given that a question formulated with a subject S and relation R can have multiple valid answers, it is crucial to recognize that the object O in a sampled triple (S,R,O) might not be the only correct response to the question. To address this issue, we define acceptable answers for a question derived from a triple (S,R,O) as a set of entities E for which (S,R,E) exists in Wikidata. These acceptable answers include aliases of object entities listed in Wikidata.

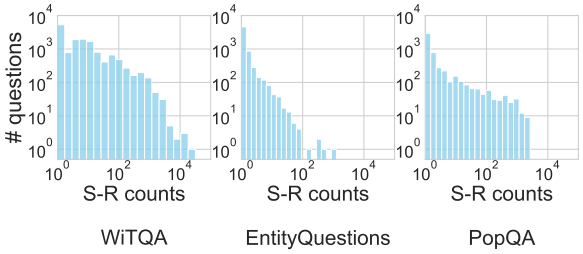

Figure 2: Histograms of question distributions. WiTQA exhibits greater diversity than existing benchmarks regarding question popularity, as indicated by the variation in S-R counts.

3.1.4 Question generation with roundtrip refinement.

Previous studies employ a template-based method to generate natural language questions for the triple-passage pairs. This method involves crafting a template for each relation with a placeholder for the subject entity, and treating the object entities as answers (Sciavolino et al., 2021; Mallen et al., 2023). However, template-based approaches are known to suffer from questions of poor quality and diversity. For instance, consider the template `What sport does [SUBJ] play?` for the relation `sport`. This template works well for triples like `(Shohei Ohtani, sport, baseball)`, resulting in a natural question: `What sport does Shohei Ohtani play?`. However, it becomes problematic with triples like `(2008-09 Maltese Premier League, sport, association football)`, leading to an awkward question: `What sport does the 2008-09 Maltese Premier League play?`.

One potential solution is to have each question written by a human; however, this approach is prohibitively expensive. Therefore, we adopt a model-assisted approach (Zhang et al., 2024) to automatically generate questions for each triple, based on

[2] https://archive.org/download/enwiki-20211020/enwiki-20211020-pages-articles-multistream.xml.bz2

[3] https://github.com/jcklie/wikimapper

[4] https://huggingface.co/cross-encoder/nli-deberta-v3-large

Questions	14,837
Unique subject entities	13,251
Unique object entities	7,642
Average length of supporting passages (characters)	214.3
Questions added in first roundtrip	12,856
Questions added in second roundtrip	823
Questions added in third roundtrip	283
Questions written by annotators	743

Table 3: Statistics of WiTQA.

the round-trip question generation method (Alberti et al., 2019). Specifically, we generate a question given a context. We then use that generated question and the context to generate the answer. To assess the validity of a generated question, we establish three criteria:

- `Answerable`: marked as True if the model accurately answers a generated question using its passage.
- `HasSubject`: marked as True if a generated question includes the subject entity of its triple.
- `NoObject`: marked as True if a generated question does not incorporate the object entity.

If a generated question fulfills all these criteria, it is considered valid; otherwise, feedback is provided to the model to enhance the question. We use GPT-3.5 (Ouyang et al., 2022) for this step due to its powerful language generation ability.

We observe that 95% of questions (14,094 /14,837) satisfy all the criteria after three round-trip iterations. The remainder of the questions (743/14,837) were rewritten by our three internal annotators. We divided the datasets into three overlapping sections and consulted with at least two annotators to obtain human-written questions. We used an annotation framework MEGAnno (Zhang et al., 2022) for this annotation due to its flexible labeling computational notebooks. Appendix B shows all the prompts that we used for the question generation and verification.

3.2 Dataset Statistics

By following the outlined data creation process, the resulting dataset, WiTQA, comprises a total of 14,837 questions. In order to position WiTQA within the landscape of factoid QA benchmarks, we compare it with two established benchmarks: EntityQuestions (Sciavolino et al., 2021) and PopQA (Mallen et al., 2023). Both of these benchmarks generate their questions from Wikidata triples. Table 2 illustrates that WiTQA is unique among QA benchmarks in that it includes supporting passages and question popularity. The histograms in Figure 2 showcase the distribution of questions based on the S-R counts. Thanks to our bin-wise triple sampling, WiTQA features a substantial number of questions with over 1,000 S-R counts, a characteristic that EntityQuestions and PopQA seldom possess. Additionally, we observe that 62% of questions in EntityQuestions and 65% of questions in PopQA have never appeared in triples extracted from the Wikipedia abstract. In this context, existing QA datasets have less diversity in terms of question popularity than WiTQA. For a more in-depth analysis of question popularity, we provide subject entity counts and subject page views[5] for each question. We show the statistics of WiTQA in Table 3. Detailed statistics of WiTQA are presented in the Appendix C.1.

4 Experiments: Recall or Retrieve

We evaluate 10 language models of varying sizes augmented with four different retrieval methods to quantify the recall capability of LMs and the performance of retrievers in isolation and jointly and share the insights.

4.1 Setup

Models. We use Flan-T5-small/base/large/xl (60M, 220M, 770M, and 3B) as small-scale LMs. We consider instruction fine-tuned Mistral-7B (Jiang et al., 2023) and Llama-2-7B/13B/70B-chat (Touvron et al., 2023) for medium-scale LMs, and GPT-3.5/GPT-4 for large-scale LMs(Ouyang et al., 2022; OpenAI, 2023).[6]

Retrievers. We consider seven retrievers that include BM25 (Robertson et al., 2009), Contriever (Izacard et al., 2021), GTR-large, GTR-xl (Ni et al., 2022), BGE[7] (Xiao et al., 2023), GenRead (Yu et al., 2023a) and Oracle. BM25 is a static term-based sparse retriever that doesn't require training. Contriever is a unsupervised dense retriever pre-trained on a large corpora and fine-tuned on MS-MARCO (Nguyen et al., 2016). GTR and BGE are supervised dense retrievers pretrained on a large corpora and then fine-tuned on various

[5]We obtain page views by querying the Wikipedia API.
[6]gpt-3.5-turbo-0613 and gpt-4-0613
[7]BAAI/bge-large-en

supervised datasets including NQ (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018). Both GTR and BGE retrieve from the Wikipedia corpus that is also employed to create our benchmark. GenRead generates relevant passages by prompting large language models. Oracle retriever always retrieves the correct supporting passage for a given QA pair. We include it to measure the reasoning capabilities of the models in isolation from retriever errors. Please refer to Appendix C.3 for more details.

Querying. We use the following two templates for prompting a model.[8] The first one (a) is generative prediction without any retrieval and (b) is contextual generative prediction that uses retrieved passage from one retrievers as a context.

(a) Generative Prediction:

```
Question:{{question}}
Answer:
```

(b) Contextual Generative Prediction:

```
Context:{{context}}
Question:{{question}}
Answer:
```

Metric. We mark a prediction as correct if any sub-string of the prediction is an exact match of any of the gold answers.

4.2 Analysis of Model's Recall Ability

We explore the models' capacity to recall factual knowledge, considering models of various sizes and questions with differing popularity of subject-relation pairs. As depicted in Figure 3, all models, regardless of their size, generally demonstrate good recall of popular facts. For instance, even Flan-T5-large can achieve up to 80% accuracy for questions with over 1000 S-R count. Predictably, larger models exhibit a superior ability to recall compared to smaller models. Remarkably, in the case of less popular questions, there is a notable discrepancy in accuracy between small models and medium-/large-scale models. Some of these findings are consistent with the insights from recent works on evaluating factual knowledge in LMs (Sun et al., 2023). Also, to compare S-R counts with subject page view counts which are used in the existing study (Mallen et al., 2023) as the question popularity, we demonstrate that the accuracy of vanilla LMs does not consistently increase with increasing

[8] Mallen et al. (2023) observed that more sophisticated instructions don't lead to significant improvements.

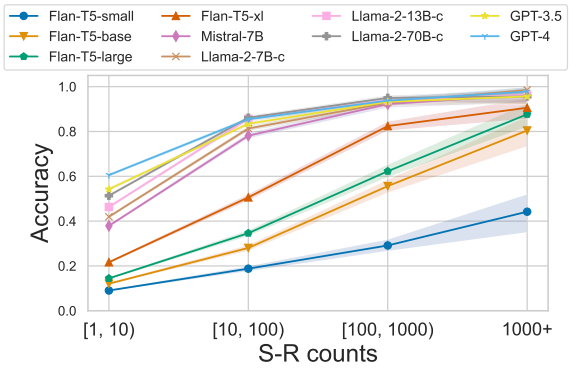

Figure 3: We categorize the questions into bins based on their S-R counts and present LMs accuracy across these bins. Shaded areas are the 95% bootstrap confidence intervals with 1000 samples. Larger models exhibit higher accuracy than smaller models. Even small models memorize factual knowledge about popular questions.

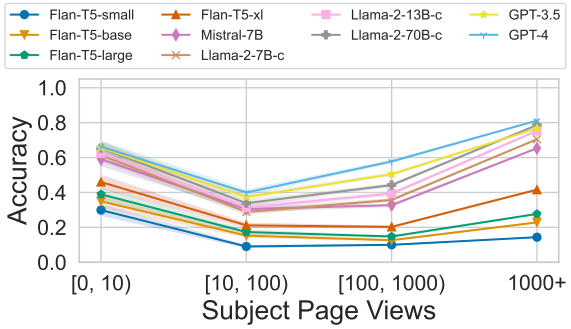

Figure 4: Accuracy over subject entity page views.

page view count in Figure 4 while accuracy consistently increases as S-R and S counts increase in Figures 3 and 5, respectively. This suggests that our proposed metrics are more robust than previously proposed metrics.

4.3 When Do Retrievers Help

Next, we augment the models with the retrievers and estimate the performances based on top-one retrieved passage. While expanding the context size could potentially enhance performance, we leave the exploration of augmentation with top-k passages for future investigations.

Figure 6 shows the results of various LMs with and without augmentation. Generally, retrieval augmentation enhances model performance, particularly for smaller models. However, the performance gap tends to be smaller, and sometimes even negative, for larger models like Llama-2 and the GPT series. GPT models often generate responses like "The context does not provide information on..." when the retrieved passages are insufficient to answer the question.We observe a similar pattern

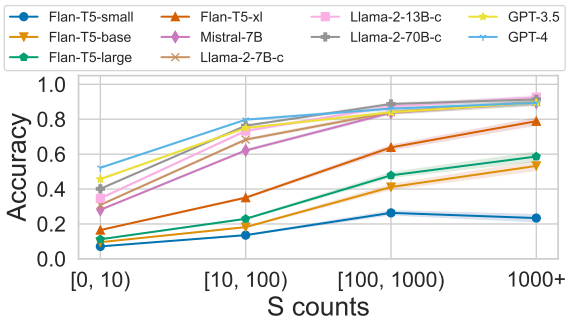

Figure 5: Accuracy over entity counts.

in Llama models, although it occurs less frequently. In essence, retrieved information often supersedes recall in larger models, suggesting that these models have high recall but are also more susceptible to retrieval errors.

Our observations indicate that BM25, Contriever, GTR, and BGE notably enhance the accuracy of small and medium models by up to 49.2%, while GenRead appears to be more effective for larger models with up to 4.1% improvement. This suggests that the ability to maintain a coherent chain-of-thought emerges more prominently when the model size is sufficiently large.

We observe a significant improvement in accuracy for all models, irrespective of their sizes, when augmented with Oracle. Even the smallest model, Flan-T5-small, achieves an accuracy of 78.1% with Oracle passages. However, substantial differences are observed between performances with BM25/Contriever/GTR/BGE and Oracle, which can be attributed to retrieval errors. When augmented with incorrect passages, most models struggle to provide correct answers.

Finally, we discuss why the small models augmented with Oracle passages obtain lower accuracy than larger models. We observe that when the context has multiple entities, the small models tend to predict the entity that appears close to the question words, which may not be correct. We give an example:

> Context:
> Susanna Wesley (née Annesley; 20 January 1669 – 23 July 1742) was the daughter of Dr Samuel Annesley and Mary White, and the mother of John and Charles Wesley
> Question:
> Who was the mother of Charles Wesley?

For example, Flan-T5-small answers "Mary White" while the true answer is "Susanna Wesley". Note that the context supports the true answer, but Flan-T5-small cannot answer correctly. Highlighted by the example, we observe that small models tend to extract an entity as an answer, which appears near from the words in questions, leading to lower accuracy compared to larger models.

4.4 Deep Dive into Errors

In order to closely examine the factors influencing recall and retrieval, we concentrate on GPT-3.5 as the baseline model, along with the four retrievers.

How does popularity affect RALM performance? Figure 7 shows the accuracy of GPT-3.5 on questions with different subject-relation (S-R) counts. Notably, the accuracy of Vanilla and GenRead is considerable for popular questions but significantly declines when S-R counts drop below 10. This drop is likely due to models memorizing popular facts. Conversely, RALMs utilizing BM25, Contriever, GTR, and BGE exhibit greater robustness on less popular questions. However, for popular questions, their performance is inferior compared to the Vanilla model without augmentation. Given that supporting passages (Oracle) consistently enhance the performance of models, we hypothesize that retrieval errors for popular questions negatively impact the performance of RALMs using BM25, Contriever, GTR, and BGE.

Is there a correlation between RALM performance and retrieval errors? To verify the above hypothesis, we closely examine the relationship between RALM performance and retrieval errors for questions of varying popularity. To estimate retriever performance, we compute passage accuracy, marking a passage as correct if any substring of the passage is an exact match for a gold answer. We report the passage accuracy of different retrieval models in Figure 8. This trend aligns closely with RALM performance, as illustrated in Figure 7. Specifically, agreement ratios between RALM performance and passage accuracy are 85.5%, 82.4%, 88.5%, 88.4%, 88.7%, and 93.1% for BM25, Contriever, GTR-large, GTR-xl, BGE, and Oracle, respectively. Upon closer examination, we find that although retriever accuracy is low for rare questions, the drop is less significant compared to Vanilla LMs (see the leftmost plots in Figures 7 and 8). This indicates that retrieval augmentation is still beneficial for rare facts compared

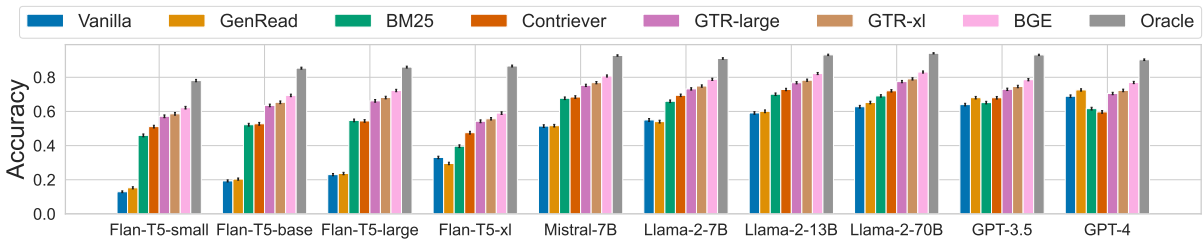

Figure 6: Overall accuracy of Vanilla LMs and LMs augmented with GenRead, BM25, Contriever, GTR-large, GTR-xl, BGE, and supporting passages (Oracle). Error bars show the 95% bootstrap confidence intervals with 1000 samples. Accuracy of Vanilla and GenRead improves with larger model sizes. Augmentation with retrievers enhances accuracy, especially for smaller models. However, the gap diminishes and, in some cases, becomes negative for larger models such as the GPT series. Supporting passages (Oracle) prove beneficial for all models.

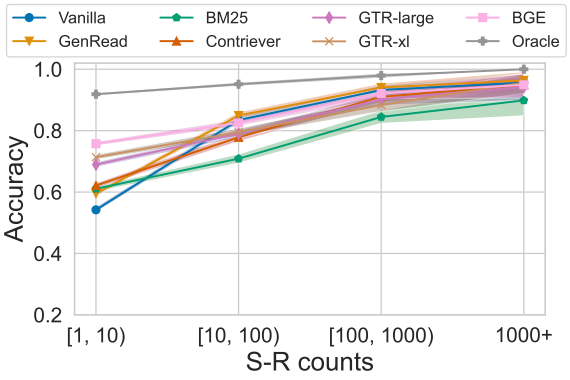

Figure 7: Accuracy of different retrievers with GPT-3.5 across varying question popularity.

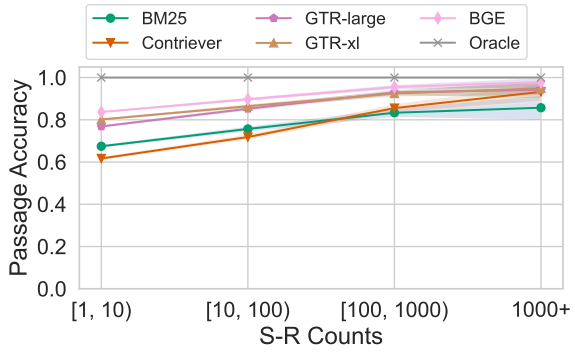

Figure 8: Passage accuracy across question popularity.

to recall. In contrast, Vanilla LMs perform better than retrievers for popular questions.

How does model size affect the need for augmentation? We further explore how the size of the model influences the need for augmentation based on the popularity of questions, considering that the ability to recall varies with model size. To this end, we categorize the questions into four groups based on the median of S-R counts (5) and S counts (12):

- head-head: both S-R and S counts are high
- head-tail: S-R count is high but S count is low
- tail-head: S-R count is low and S count is high
- tail-tail: both counts are low

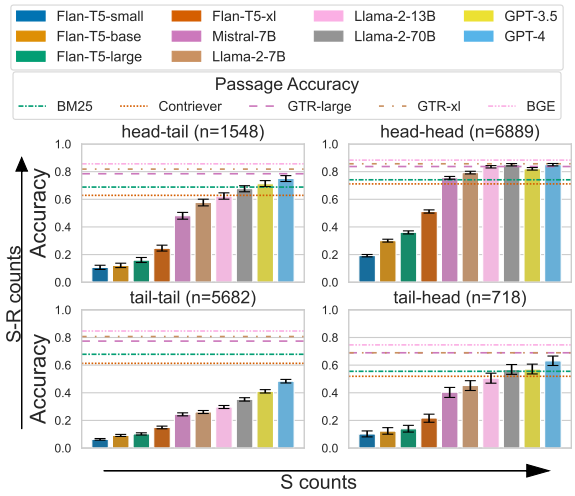

Figure 9: Analysis on Vanilla LMs with BM25, Contriver, GTR, and BGE passage accuracy over S-R counts and S counts (n = the number of questions in the group). In the top row, S-R counts are higher than the median. In the bottom row, they are less than or equal to the median. In the left column, S counts are less than or equal to the median and in the right column, they are higher than the median.

Figure 9 shows the accuracy of various models along with BM25, Contriver, GTR, and BGE passage accuracy across the four groups. Note that we plot QA accuracy and passage accuracy since we use the same criteria for them. We observe that supervised retrievers, GTR and BGE, outperform vanilla LMs in all groups. This is attributed to the fact that it was fine-tuned with QA datasets about Wikipedia. For the generalizability to broader domains, we discuss the results with an unsupervised retriever BM25 in the following.[9]

In the head-head group, medium and large models exhibit superior performance even without augmentation, e.g., the GPT series achieve higher than

[9]Similar results were observed for Contriever; hence, we focus on BM25 for simplicity.

80% accuracy. In contrast, there is a notable gap between the accuracy of the vanilla model and passage accuracy for small models. This suggests that only small models need augmentation for questions in this group.

In the head-tail and tail-head groups, Llama and GPT series achieve comparable or superior accuracy to the passage accuracy. This indicates that large models can memorize long-tailed relations of popular entities and popular relations of infrequent entities. On the other hand, medium-scale models fail to recall compared to passage accuracy, and hence, they will benefit from augmentation. Interestingly, the passage accuracy in the tail-head group is significantly lower (less than 60%) compared to other groups. This retrieval error is caused by frequent mentions of the entity but infrequent mentions of subject-relation pairs. Consequently, accurately identifying relevant passages from a large pool of passages containing the question entity becomes challenging.

In the tail-tail group, even GPT series struggle to recall long-tail information for rare entities. Since retrievers are robust for such information, we conclude that retrieval augmentation is generally helpful for all models.

Does a combination of recall and retrieval improve accuracy? Based on these observations, we hypothesize that an optimal combination of recall and retrieval can be obtained using S-R and S counts as thresholds. We develop and examine a selective memory integration method that uses augmentation based on S-R and S counts, while using vanilla LM as default. This method yielded an improvement of 10.1% and 8.1% over Vanilla and BM25 for GPT-3.5 and GPT-4, respectively. For more details on this investigation, refer to the Appendix C.2.

5 Further Related Work

Several recent studies (Asai et al., 2024; Yu et al., 2023b; Ma et al., 2023; Zhang et al., 2023) have focused on enhancing the robustness of RALMs by assessing the usefulness and relevance of retrieved passages to questions.

Yu et al. (2023b) has proposed the Chain-of-Note framework, which systematically evaluates the relevance and accuracy of retrieved passages, thereby improving the noise robustness of RALMs. Ma et al. (2023) has addressed query rewriting to improve retriever performance by using LMs or pre-trained LMs that are fine-tuned for a suitable query rewriter. Zhang et al. (2023) has introduced an approach that leverages two sources of information: retrieved passages and LM-generated passages. This approach is designed on the hypothesis that answers corroborated by both sources have a higher likelihood of being accurate.

These studies rely on model-centric approaches for assessing document relevance to queries, refining queries, and integrating retrieved and generated passages. Thus, they overlook the importance of data-centric question popularity as an indicator for deciding when to retrieve and augment information. In contrast, our study leverages question popularity metrics derived from Wikipedia, offering insights that are complementary and distinct from these existing methodologies.

6 Conclusion

We introduced WiTQA, a novel QA dataset comprising of QA pairs associated with supporting passages. This dataset enables us to assess the performance of LMs with respect to question popularity and retrieval-augmentation. We conducted extensive experiments investigating the zero-shot performance of 10 LMs with four different retrieval methods. Our findings reveal several key insights: 1) the ability to recall factual knowledge is influenced by the model's size, with even the GPT series facing challenges when addressing minor facts; 2) small RALMs demonstrate robust QA performance when provided with supporting passages, suggesting that errors in RALMs are primarily due to retrieval errors; and 3) retrievers exhibit greater robustness for long-tail information of long-tail entities compared to the recall capability of LMs.

Limitations

Distribution of pre-training corpus. This work hypothesizes that the distribution of the Wikipedia texts reflect that of pre-trained texts. However, the pre-trained texts of recent proprietary models such as GPT-4 are not accessible.

Prompt engineering. A limited amount of prompt tuning was conducted. For example, larger models tend to refrain from generating answers when the provided passage does not pertain to the question. From the viewpoint of maximizing QA accuracy, encouraging models to actively formulate answers regardless of passage relevance seems advantageous. Yet, this approach could elevate

the proportion of incorrect answers, which is undesirable in practical applications. Exploring this trade-off will be a focus of our future work.

Multi-hop relations. Real-world questions often exhibit a complexity beyond simple triple-based questions, for instance, encompassing multi-hop relations. In this case, determining the subject and relation can be challenging. However, to conduct a deep analysis of LMs with clearly characterized questions, we focus on triple-based questions in this work.

Ethical Consideration

We use internal annotators for question rewriting in Section 3.1.4, who were explained how data will be used by the authors directly and earned more than the minimal wage.

Acknowledgment

We thank Dan Zhang and Kushan Mitra for their help with annotation during the data creation process and Tanmay Laud and Estevam Hruschka for valuable discussion.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. RelationPrompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 45–57, Dublin, Ireland. Association for Computational Linguistics.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

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.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings*

of Machine Learning Research, pages 15696–15707. PMLR.

Cheongwoong Kang and Jaesik Choi. 2023. Impact of co-occurrence on factual knowledge of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7721–7735, Singapore. Association for Computational Linguistics.

Bosung Kim, Hayate Iso, Nikita Bhutani, Estevam Hruschka, Ndapa Nakashole, and Tom Mitchell. 2023. Zero-shot triplet extraction by template infilling. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 272–284, Nusa Dua, Bali. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2023. Large language models with controllable working memory. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1774–1793, Toronto, Canada. Association for Computational Linguistics.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. 2023. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv*.

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.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

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.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. 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. Technical report, OpenAI.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2023. Head-to-tail: How knowledgeable are large language models (llm)? a.k.a. will llms replace knowledge graphs?

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.

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*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

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.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023a. Generate rather than retrieve: Large language models are strong context generators. *The Eleventh International Conference on Learning Representations*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023b. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

Dan Zhang, Hannah Kim, Rafael Li Chen, Eser Kandogan, and Estevam Hruschka. 2022. MEGAnno: Exploratory labeling for NLP in computational notebooks. In *Proceedings of the Fourth Workshop on Data Science with Human-in-the-Loop (Language Advances)*, pages 1–7, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Haopeng Zhang, Hayate Iso, Sairam Gurajada, and Nikita Bhutani. 2024. XATU: A Fine-grained Instruction-based Benchmark for Explainable Text Updates. In *The 2024 Joint International Conference On Computational Linguistics, Language Resources and Evaluation*.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Merging generated and retrieved knowledge for open-domain QA. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

A Detailed Setup for NLI Scores of Supporting Passages

As discussed in Section 3.1, we use the entailment prediction to choose the best supporting passage for each triple. For each triple, we input the text containing both entities from the Wikipedia abstract, and the triple in their surface forms, subject + relation + object, separated by the <sep> token. Then, we simply select the passage with the highest NLI score for each unique triple and use it as the supporting passage of the triple.

B Prompts for Question Generation/Refinement and Experiments

First, we provide a prompt used for our question generation in Figure 10. Next, we show prompts used for QA tasks in Figures 11 and 12. Also, Figure 13 shows a prompt template to generate related passages by LMs for GenRead, which was introduced in its original paper.

Also, Algorithm 1 describes the roundtrip refinement. In line 1, we initialize Message as Question generated by a prompt in Figure 10. In line 3, we obtain an answer to the question with its supporting passage by using a prompt in Figure 12. Then, we check the three criteria "Answerable", "HasSubject", and "NoObject" for the question and answer pair. If the pair satisfies all the criteria, the question is added to WiTQA in line 6. Otherwise we append a message that indicates which criteria are not satisfied in lines 8–27. In lines 17–20, the algorithm describe an exceptional case in which the object of a triple is a substring of the subject and questions are answerable. Since questions cannot satisfy the criteria "HasSubject" and "NoObject" in this case, we use this condition to add such questions to WiTQA. Before proceeding to the next iteration, we regenerate a question with the message by using GPT-3.5 in line 28. If we cannot obtain a question-answer pair satisfying the three criteria through k iterations, the authors write the questions based on the triples.

```
Given a context and a triple (subject, relation, object), transform the triple to a question that asks "Object". The generated question must contain a given "Subject" and also be answerable without the context.

# Context:
{{ context }}

# Triple:
## Subject:
{{ subject }}

## Relation:
{{ Relation }}
(Meaning: {{ relation_description }})

## Object:
{{ object }}

# Output: <question only and must contain Subject>
```

Figure 10: A prompt for question generation. "{{ context }}", "{{ subject }}", "{{ relation }}", and "{{ object }}" are replaced with the supporting passage, subject, relation, and object of a given triple. "{{ relation_description }}" is a description of the relation, which is provided in Wikidata.

C Additional Experimental Details

Implementation. For open LMs, we execute all experiments on a GPU node with 8 NVIDIA A100-SXM cores. As for Llama-2-70B, we use AWQ quantization (Lin et al., 2023) to make it fit our GPU. We set the temperature parameter to 0 for all experiments.

C.1 Additional WiTQA Statistics

Table 4 shows 32 relations used in WiTQA and the number of questions containing each relation in WiTQA.

In Table 5, we show relation counts indicating how many times each relation appears in all extracted triples in Wikipedia abstract. Thanks to the bin-wise triple sampling described in Section 3.1,

Algorithm 1 Roundtrip question refinement

Input: Question, k
Output: Message

Message ← [Question] ▷ Use a question generation prompt template in Figure 10
for i in range(k)
  Answer ← GPT-3.5(Message) ▷ Use a RAG prompt template in Figure 12
  Answerable, HasSubject, NoObject ← Check_criteria(Question, Answer)
  if Answerable & HasSubject & NoObject
    Add Question to WiTQA
    break
  else if !Answerable & HasSubject & NoObject
    Message.append("It is good that the question contains 'Subject' and not 'Object', but the question cannot be answered. Make the question more detailed if needed. Try again.")
  else if !HasSubject & NoObject
    Message.append("The question you generated does not contain 'Subject', but 'Subject' must be in the question. Try again.")
  else if !HasSubject & !NoObject
    Message.append("The question you generated does not contain 'Subject' and does contain 'Object'. However, 'Subject' must be in the question. Also, 'Object' must not be in the question. Try again.")
  else if HasSubject & !NoObject
    if Question.subject in Question.object & Question.subject ! = Question.object
      Message.append("The question you generated contains 'Subject' and 'Object', but 'Object' must not be in the question. Though 'Subject' is the substring of 'Object', remove 'Object' and remain only 'Subject'.")
    else if Question.object in Question.subject
      if Answerable
        Add Question to WiTQA
        break
      else
        Message.append("It is good that the question contains 'Subject', but the question cannot be answered. Make the question more detailed if needed. Try again.")
      end if
    else
      Message.append("The question you generated contains 'Subject' and 'Object', but 'Object' must not be in the question. Remove 'Object' and remain only 'Subject'.")
    end if
  end if
  Message.append(GPT-3.5(Message)) ▷ Concatenate a refined question into Message
end for

```
# Question:
{{ question }}

# Answer: <answer only>
```

Figure 11: A prompt template for vanilla. "{{ question }}" is replaced with an actual question.

WiTQA successfully captures triples with long-tail relations such as "cuisine" and "medical condition", improving the diversity of its questions.

C.2 Selective Memory Integration

In Section 4.4, our analysis revealed that while RALMs exhibit enhanced performance over Vanilla LMs for less popular questions, Vanilla LMs achieve superior performance in handling more popular questions. This finding suggests a complementary relationship between Vanilla LMs and RALMs, contingent upon the popularity of the questions. The above observation motivates us to selectively integrate LMs and RALMs with the larger sizes to improve overall accuracy by using the question popularity as an indicator to decide when to augment or not. As shown in Figure 9, if both S-R and S counts are small (the tail-tail group), we need to augment LMs with retrieved passages; otherwise Vanilla and GenRead obtain higher accuracy. Hence, we take an approach that uses an RALM for the tail-tail group and Vanilla LM for other groups.

C.2.1 Settings

We estimate optimal thresholds of S-R counts and S counts for each relation by using 50% of questions in WiTQA, i.e., we find optimal thresholds for S-R and S counts so that the thresholds maximize the overall accuracy. Concretely, we use Vanilla LMs

```
Given a context and a question, answer the question.

# Context:
{{ context }}

# Question:
{{ question }}

# Answer: <answer only>
```

Figure 12: A prompt template for GenRead, BM25, Contriever, and Oracle. "{{ question }}" and "{{ context }}" are replaced with an actual question and context, respectively.

```
Generate a background document from Wikipedia to answer the given question. {{ question }}"
```

Figure 13: A prompt template for GenRead passage generation. "{{ question }}" is replaced with an actual question. This prompt comes from its original paper

if questions with smaller S-R counts and smaller S counts than their thresholds, otherwise we use LMs with BM25. Then, we apply selective LMs accordingly to the remaining questions in WiTQA. We run the experiments with 5 different random seeds to split the dataset and report the average and standard deviations.

C.2.2 Results

Figure 14 shows that selective method achieves better accuracy than Vanilla and BM25 across all models. In particular for large models, the selective memory integration improves 7.7%, 10.1%, and 8.1% over baselines for Llama-2-70B, GPT-3.5, and GPT-4, respectively. As discussed in Section 4.3, the GPT series with retrievers frequently output that the given contexts do not support the facts related to questions, rather than directly answering the questions. Consequently, their performance with selective integration is lower or comparable to that of Llama-2-70B.

Figure 15 illustrates the average retrieval ratio of LMs with the selective memory integration over 5 runs. We observe a trend that the retrieval ratio are shifting to smaller as the model size grows. For

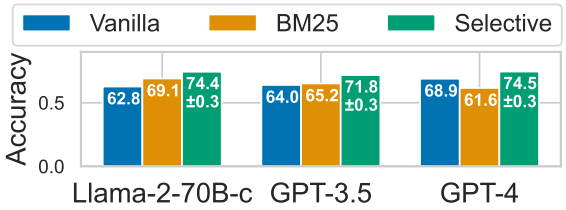

Figure 14: Accuracy of Vanilla, BM25, and LMs with a selective retriever. Accuracy (%) is displayed inside bars. As for selective LMs, we run 5 trials with different random seed to split a dataset and show the standard deviations.

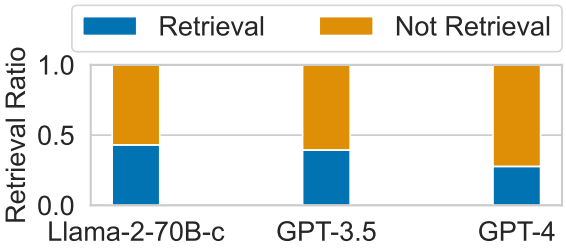

Figure 15: Average retrieval ratio of our selective memory integration method within 5 trials. "Retrieval" represents the ratio of answers sourced from LMs with BM25, while "Not Retrieval" denotes the ratio of answers sourced from Vanilla LMs.

example, GPT-4 uses a retriever for only 27.7% of questions while Llama-2-70B uses a retriever for 43.0% of questions. This is mainly because larger models can typically recall facts more correctly than smaller models.

C.3 Retriever setup

Passage chunking We use the llama-index to chunk the Wikipedia documents into chunks with `chunk_size = 256` and `chunk_overlap = 0`.

Relation label	Count
country	1,269
sport	1,034
capital	1,032
capital of	754
genre	658
author	654
language used	639
country of citizenship	619
father	560
characters	554
religion	550
composer	534
occupation	518
publisher	495
director	479
place of birth	443
educated at	392
mother	351
industry	345
relative	331
screenwriter	278
producer	271
doctoral advisor	214
broadcast by	214
published in	210
location of first performance	209
cuisine	208
executive producer	208
color	207
medical condition	204
architectural style	203
director of photography	200

Table 4: Number of questions containing relations in WiTQA.

Relation label	Count
country	1,979,521
sport	613,742
country of citizenship	278,973
place of birth	210,125
genre	169,649
capital	164,597
occupation	126,448
educated at	90,799
director	83,411
author	75,810
capital of	57,566
father	43,401
publisher	28,731
religion or worldview	23,986
composer	23,917
screenwriter	16,754
producer	16,578
language used	15,296
mother	10,348
industry	10,118
characters	9,783
architectural style	7,428
relative	4,327
doctoral advisor	2,544
published in	1,600
director of photography	1,254
location of first performance	1,086
broadcast by	874
color	787
medical condition	771
executive producer	707
cuisine	417

Table 5: Relation counts in all extracted triples in Wikipedia abstracts.