SEMI-PARAMETRIC LANGUAGE MODEL WITH SELECTIVE MEMORY

Anonymous authors

Paper under double-blind review

ABSTRACT

Pretrained on trillions of tokens, LLMs are known for their ability to store a large amount of factual knowledge in their parametric memory. However, recalling facts from this memory is known to be unreliable, particularly for longtail knowledge—obscure facts infrequently mentioned in training data. Although retrieval-augmented generation (RAG) is the standard solution, it introduces overheads such as increased inference costs due to longer input contexts and additional engineering complexity from preprocessing and indexing extensive document collections. In this work, we propose a novel approach to improve the factuality of LLMs on long-tail knowledge. We begin by identifying atomic facts—short statements detailing relationships of some entities—that are not present in a pretrained LLM's parametric memory. These facts are then stored in an external, non-parametric memory. Subsequently, the model undergoes continuous pretraining, enabling it to learn when to consult this external memory at inference time. Compared with existing approaches, our approach uses a compact external memory that selectively stores only the facts not clearly present in the LLM's parametric memory, resulting in minimal additional inference-time costs in terms of both time and space. Furthermore, our method outperforms fully trained models of comparable size on knowledge-intensive benchmarks and achieves competitive results against larger models.

1 Introduction

Large language models (LLMs) have demonstrated an outstanding ability to learn a substantial amount of world language from their training corpus, storing this knowledge in their parameters and excelling in a wide range of applications. However, despite these advanced capabilities, LLMs frequently encounter the problem of hallucination, particularly when dealing with long-tail knowledge that is less represented in their training data (Mallen et al., 2023; Asai et al., 2023; Kandpal et al., 2023; Wei et al., 2024).

Recent work has explored integrating external memory into language models to improve factuality and reduce memorization at inference time (Li et al., 2025) (Mallen et al., 2023) (Li et al., 2024) (He et al., 2023). However, a key limitation of this training-free approach is the need for a complex proposal procedure to decide between generating or retrieving from memory. Another line of work augments language model training with retrieval-based memory (Guu et al., 2020)(Borgeaud et al., 2021). This type of methods force all retrived knowledge into model parameters, which adds significant overhead during training and inference. Memory3Yang et al. (2024) integrates externalized knowledge into attention layers by compressing key value representations. More recently, Large Memory Language Model (LMLM) (Zhao et al., 2025) introduced a pretraining recipe that stores factual knowledge both in model weights and in an external database. While effective, LMLM does not differentiate long-tail knowledge from sufficiently learned knowledge, which can lead to unnecessary retrievals and even degrade downstream performance.

A central challenge still remains: when and how to decouple long-tail knowledge during pre-training to improve factuality without sacrificing other capabilities. Offloading knowledge too early may limit the model's ability to form long-range associations between knowledge entities, while offloading too late may fail to sufficiently remove long-tail knowledge from the model parameters.

We address this with continual pretraining and selective memory, which uses partially trained models to detect knowledge gaps and adaptively externalize them. Our method begins by employing model-based or frequency-based methods to score long-tail knowledge within the training corpus. We preprocess the corpus into interleaved sequences of standard text and extracted knowledge segments, where the long-tail knowledge are stored in an external non-parametric memory. During continued pretraining, we implement an adaptive masking strategy: when the cross-entropy loss for a knowledge segment exceeds a predefined threshold, indicating potential memorization difficulty or long-tail knowledge, we masks these segments from the next-token prediction objective. To enable effective retrieval from this external memory during inference, we also finetune a lightweight query adapter that learns to generate appropriate query representations and maintains the separation between parametric knowledge (encoded in model weights) and non-parametric knowledge (stored in external memory).

Our method demonstrates superior performance on memory-intensive benchmarks compared to fully trained models of equivalent size, and even achieves competitive results compared to larger models. Compared to full offloading, this approach achieves a favorable trade-off: it preserves performance on general and reasoning-heavy benchmarks while shows clear improvements on long-tail knowledge tasks.

2 Semi-parametric Language Models (SPLM)

2.1 Identify Long-Tail Knowledge

Common knowledge—frequent facts and patterns—are best captured in the model's parametric weights, allowing for fast, generalizable inference. Offloading these to external memory can force the model to rely on retrieval for even simple knowledge, increasing inference time and reducing generalizability. On the other hand, long-tail knowledge is difficult to internalize, requires repeated exposures, and offers little generalization benefit, yet it consumes disproportionate model capacity. Similar to Zhao et al. (2025), We preprocess the training corpus into interleaved free text and memory segments. Each memory segment contains a query—answer pair wrapped in special tokens: m_start [query text] m_retrieve [answer text] m_end], where m_start marks the beginning of a memory segment, and the m_retrieve token indicates the start of the answer which should be looked up from external memory. For example, the sentence "Sugeno received the IEEE Frank Rosenblatt Award in 2010 for his contributions to the field of fuzzy systems." is preprocessed as follows:

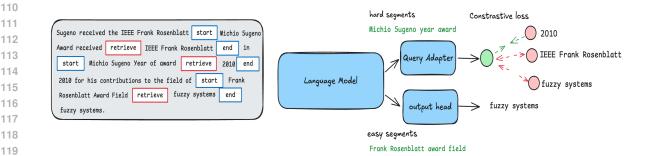
Sugeno received the m_start Michio Sugeno Award Received m_retrieve IEEE Frank Rosenblatt Award m_end in m_start Michio Sugeno Year of Award m_retrieve 2010 m_end 2010 for his contributions to the field of fuzzy systems.

In addition, we score each memory segments in terms of how rare or hard the knowledge segments. We experiment with a few variants for proxy of long tail knowledge, and leave the discussion on this part to section 3.5.2.

2.2 Training with selective memory

To prevent the model from inefficiently memorizing these rare or difficult knowledge pieces, we selectively delegate them to an external memory. In practice, this is achieved by masking out the answer tokens for hard knowledge segments during continued pre-training, so the model is trained to intermix free text generation with knowledge retrieval calls. This selective masking ensures that the model's parametric capacity is focused on common knowledge, while long-tail knowledge is handled via retrieval.

Selective Memory Offloading all knowledge to memory can bloat the retrieval store with trivial facts, slow down inference, and make the system brittle when retrieval fails. Prior work has shown that pretrained models' loss is a reliable proxy for knowledge gaps, where persistent high loss usually indicates unreliable memorization. For a given memory segment s_i , let A_i denote the set of token positions corresponding to its answer text. We compute the average answer loss ℓ_i as a proxy for the difficulty of the memory segment. We define hard memory segments as those with average loss



108

111

120 121

122

123

124

125 126 127

128

129

130

131

132

133 134

135

136 137 138

139

141 142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158 159 160

161

Figure 1: Training with selective memory and query adapter. Easy memory segments are trained directly with the standard next-token prediction loss, while hard segments are routed through a lightweight query adapter. The query adapter is optimized with a contrastive loss: it is pulled closer to the correct answer embedding (positive) and pushed away from incorrect answers (negatives) drawn from other segments within the same document.

above a threshold $S_H = \{s_i : \ell_i > \tau\}$. If the model finds a particular memory segment consistently difficult (loss beyond τ), we mark that segment as a candidate for externalization. In practice, we experiment with either a fixed threshold or an adaptive threshold based on the quantile of average losses across all memory segments. With selective memory delegation, we introduce a binary mask $\mathbf{M} = [m_1, m_2, \dots, m_T]$ where:

$$m_t = \begin{cases} 1 & \text{if } s_i \in \{S_H\} \text{ and } x_t \text{ is between } \underline{\mathbf{m}}_{\underline{\mathbf{s}}} \underline{\mathbf{t}} \underline{\mathbf{m}} \underline{\mathbf{m}}_{\underline{\mathbf{r}}} \underline{\mathbf{t}} \underline{\mathbf{r}} \underline{\mathbf{t}} \\ 0 & \text{otherwise} \end{cases}$$
 (1)

We apply this mask to the language modeling objective, so that the model is trained only on unmasked tokens. $\mathcal{L}_{\text{masked}} = -\frac{1}{\sum_{t=1}^{T} m_t} \sum_{t=1}^{T} m_t \cdot \log P(x_t \mid x_{< t}; \theta)$. This effectively removes the loss contributions from the answers of hard memory segments, discouraging the model from attempting to memorize them and instead encouraging reliance on external memory for factual recall. This adaptive masking mechanism enables the model to allocate its parametric capacity to common knowledge and compositional reasoning, while selectively offloading long-tail or hard-to-learn knowledge to a non-parametric memory.

Finetuning adapter While the above mechanism teaches the model when to retrieve, we also need to train how to retrieve the correct information from the external memory. To enable efficient and robust retrieval at inference time, we introduce a lightweight, contextualized query adapter that projects both query context and answer text into a shared embedding space where relevant query-answer pairs are close. During finetuning, only the adapter parameters are updated, while the LLM backbone remains frozen for efficiency and to preserve previously learned knowledge. The adapter is trained to map the high-dimensional contextual representation of a query (the LLM's hidden state at the m_retrieve token position) to a lower-dimensional dense query embedding that can be used to retrieve relevant answers. The final hidden representation at the position of the m_retrieve token is passed through the adapter to obtain a dense query embedding z_{q_i} . The adapter thus Similarly, for each memory segment's answer, we obtain a dense answer embedding by applying a pooling function over the LLM's final hidden states for the answer text to obtain z_{a_i} . We finetune the adapter using a contrastive InfoNCE loss: For a batch of N memory segments (i.e. Nquery-answer pairs (q_i, a_i)), we treat each (q_i, a_i) as a positive pair and all other N-1 answers in the batch as negatives for q_i :

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\sin(z_{q_i}, z_{a_i})/\tau)}{\sum_{j=1}^{N} \exp(\sin(z_{q_i}, z_{a_j})/\tau)},$$
 (2)

where $sim(u, v) = \frac{u^{\top}v}{\|u\| \|v\|}$ denotes the cosine similarity between vectors and $\tau > 0$ is a temperature hyperparameter controlling the sharpness of the distribution. We leverage in-batch negatives during training, since each batch typically contains multiple memory segments extracted from the same document or entity, naturally serving as hard negative examples.

Inference At inference time, if the model outputs the special $\underline{\mathbf{m}}$ -retrieve token, this signals that a knowledge retrieval is needed for the current query. At that point, we take the model's current hidden state at the $\underline{\mathbf{m}}$ -retrieve position (which encodes the context of the query) and feed it through the adapter to get a contextualized query embedding z_q . This embedding is then used to perform a nearest-neighbor search in the external memory store for the most relevant stored answer.

3 EXPERIMENTS

3.1 EXPERIMENT SETUP

Training details We pretrained a baseline LLAMA-3-1B model for 240k steps on the DCLM corpus (220B tokens). We then continued pretraining on the Wikipedia subset of the Dolmino corpus (3.7B tokens) for an additional 30k steps, using a learning rate of 4×10^{-4} , batch size of 4, and sequence length of 4096. Subsequently, we finetuned the adapter as a two layer MLP with 764 output dimension on the same corpus for 5k steps with a learning rate of 1×10^{-3} .

Inference details For generation, we compare using a off-the-shelf sentence embedding model (all-MinilM-L6-v2(Reimers & Gurevych, 2019; Wang et al., 2020)) to embed queries against using a finetuned adapter. For the sentence embedding model, we use a retrieval threshold of 0.6, and for the finetuned query adapter we apply a higher threshold of 0.7. To support efficient approximate nearest-neighbor search, we employ a FAISS dense index (Johnson et al., 2019), which clusters vectors into 16,384 centroids, and we set nprobe= 64 during retrieval to balance recall and efficiency.

3.2 EVALUATIONS

We evaluate our models on a set of benchmarks designed to test general or long-tail factual knowledge in language models. For general-purpose QA, we adopt widely used benchmarks that serve as guardrails for assessing open-domain question answering: NATURALQUESTION (NQ) (Kwiatkowski et al., 2019), ENTITYQ (Sciavolino et al., 2021) and HOTPOTQA (HQA) (Yang et al., 2018). These datasets emphasize broad coverage of knowledge and multi-hop reasoning, making them representative tests of general QA ability.

To specifically target long-tail factual knowledge, we use a suite of benchmarks including POPQA (Mallen et al., 2023), HEAD-TO-TAIL (Sun et al., 2024), and SIMPLEQA (Wei et al., 2024). POPQA is curated to cover questions on widely recognized entities and facts, while also incorporating a substantial proportion of less common, long-tail items. HEAD-TO-TAIL explicitly measures the popularity of entities: it consists of 18K question—answer pairs categorized into *head* (frequent), *torso* (moderately frequent), and *tail* (rare) entities. SIMPLEQA is a fact-seeking benchmark of short, unambiguous questions designed to challenge SOTA models.

3.3 BASELINES

Vanilla training As a baseline, we continued pretraining the model on the original Wikipedia subset of Dolmino for the same number of steps, without introducing any external memory or selective masking. This setup measures the effect of additional domain-adapted pretraining alone, isolating gains that come purely from further exposure to the corpus.

Full Memory Following the setup in the LMLM framework (Zhao et al., 2025), we construct a memory-only baseline in which all factual tokens are offloaded to external memory. In this setting, the model never learns to parametrize knowledge internally; instead, retrieval is performed at every m_retrieve call using a separate embedding model and fuzzy string matching against the memory index.

	NQ	HQA	EntityQ All	PopQA	Head- All	to-Tail Tail	SimpleQA
LLaMA3 8B	32.05	26.16	31.16	22.65	14.77	10.74	4.76
LLaMA3 1B	18.48	21.01	14.12	14.30	7.78	6.22	2.27
LLaMA3 1B - 240k	17.37	18.10	17.34	14.54	7.51	5.30	2.87
+Vanilla training	9.39	13.45	12.43	11.13	6.26	4.49	2.36
+Memory	13.85	13.86	24.7	20.31	11.44	11.52	7.35
+SM	14.85	13.53	24.82	23.91	11.45	10.84	9.96
+SPLM	17.65	16.24	25.86	26.3	12.24	11.3	10.28

Table 1: Answer-level recall (ALR) on open-domain QA benchmarks. The baselines include LLaMA3 8B, LLaMA3.2 1B, and a partially trained LLaMA3 1B (240k steps). For selective memory (SM), we report the result with an adaptive threshold of 0.6 on quantile of average answer token losses (40% of hard segments are included in memory).

3.4 MAIN RESULTS

In Table 1 we show the results on the answer-level recall (ALR) of a LLaMA-3 1B trained under different paradigms: continual pretraining on regular corpus (vanilla training), continual pretraining with fully offloaded memory (Memory), continual pretraining with selective memory (SM), and our full approach with selective memory combined with a query adapter (SPLM).

Continual pretraining with memory outperforms fully trained models on knowledge intensive tasks Compared to a fully trained Llama 1B model, training with memory demonstrates superior performance particularly on benchmarks that emphasize long-tail or entity-centric knowledge, such as EntityQ, PopQA, and the tail subset of head-to-tail. It also achieves comparable results compared to a much bigger model (Llama 8B), indicating that externalizing long-tail knowledge can enhance factual recall without increasing model size.

Selective memory outperforms full knowledge offloading We also observe clearly improvements when using selective memory compared to fully offloading memory. While offloading all memory improves recall on certain long-tail benchmarks such as PopQA, head-to-tail and simpleQA, it leads to degradation on general QA dataset (NQ) or those require multi-hop reasoning (HotpotQA). In contrast, selective memory achieves a better balance: it maintains competitive performance on general QA datasets while further improves the performance on long-tail benchmarks. This shows that adaptively masking only the hardest segments allows the model to preserve its parametric capacity for common knowledge and reasoning, while only relying on external memory for rare knowledge. Finally, incorporating a lightweight query adapter on top of selective memory (SPLM) recovers much of the degradation observed on general QA benchmarks while further improving long-tail factual recall. We observe the degradation on certain benchmarks is a result of the model generating imperfect and unnecessary retrieval. The query adapter mitigates this issue by learning to produce contextualized, discriminative, and richer query embeddings that are naturally aligned with the model's internal representation.

3.5 ABLATIONS

3.5.1 When should we decouple long-tail knowledge?

LMs are inefficient in memorizing long-tail knowledge in parametric memory Figure 2 plots the answer-level recall (ALR) of a Llama3 1B model trained on DCLM corpus as training progresses. We observe that some benchmarks exhibit clear improvements with additional training, while others remain stagnant. For example, NQ and HotpotQA continue to benefit from longer training, showing steady gains up to 24k steps. By contrast, simpleQA and popQA plateau quickly, indicating that continued training on the same corpus does little to improve performance on long-tail benchmarks.

pretraining vs continual pretraining Decoupling long-tail facts to external memory can be done during pre-training or during continuous pre-training time. However, it's hard to separate "factual

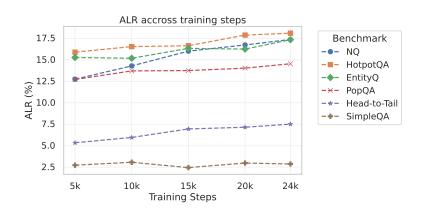


Figure 2: Answer-level recall (ALR) across training steps when the model is trained on regular corpus.

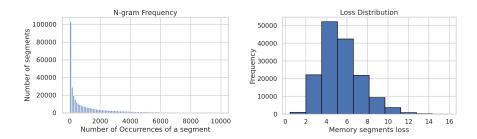


Figure 3: Model based and model agnostic scoring for long-tail knowledge.

memory" from "reasoning" in a clean way. Many reasoning capabilities are scaffolded and interleaved with factual text. For example, solving multi-step questions in math often relies on domainspecific facts or theorems embedded in the training data. If large spans of such text are masked, the model may undertrain on certain compositional patterns, hurting downstream reasoning.

3.5.2 DIFFERENT PROXIES FOR LONG-TAIL KNOWLEDGE

In this section, we compare model-based proxies and model-agnostic proxies for detecting long-tail knowledge in a corpus.

Model agnostic scoring Model-agnostic approaches estimate rarity directly from corpus statistics. One effective tool is *Infinigram* (Liu et al., 2024), which computes exact *n*-gram frequency distributions in a given corpus. We count the frequency if either the objective or the subjective entity in the memory segment appear in the corpus. While Infinigram provides an exact measure of *n*-gram frequencies, its distribution is extremely skewed (Figure 3), making it challenging to draw a clear boundary between head and long-tail knowledge.

Model based scoring Training loss provides a natural model-based signal of difficulty and has been widely used for curriculum learning. Figure 3 shows the distribution of memory segment losses at the start of continual pretraining: most losses concentrate around moderate values, while the long tail represents knowledge the model struggle to internalize. Alternatively, we also experimented with model based scoring, where we prompt the Llama3-70B model (see Appendix for prompt details) to assign a popularity score to each knowledge segment on a scale from 1 to 10, with higher scores corresponding to more frequently occurring knowledge and lower scores marking long-tail or rare facts.

3.5.3 DIFFERENT MEMORY THRESHOLD

Fixed vs adaptive threshold In the simplest variant, we apply a fixed threshold τ , masking any answer segment with average loss above τ . This provides a straightforward knob on how aggressively long-tail knowledge is offloaded: the higher τ is, the fewer segments are delegated to external memory. Alternatively, we experiment with an adaptive threshold, where at each step we compute the distribution of average losses across all answer segments and mask those falling in the top $\tau\%$ (e.g., 80th percentile). The adaptive strategy dynamically adjusts to training progress and corpus difficulty: early in training, more segments are masked, while later only the most challenging segments are delegated to memory. We report answer-level recall (ALR) under different fixed thresholds shown in figure 4. For fixed threshold, as the memory threshold increases, ALR generally decreases for the entity-centric and longtail benchmarks benchmarks such as EntityQ and PopQA, but increases for the common knowledge QA datsets such as NQ and HotpotQA. This aligns with our assumption that aggressive memory offloading (lower threshold) could hurt performances on general benchmarks. For adaptive threshold, we observe a non-monotonic trend, where an intermediate threshold leads to better performance. We hypothesize that this arises from how adaptive thresholds interact with the distribution of segment losses during model's training dynamics. If the percentile is too low, only the hardest segments are kept in memory and the performance improvement is limited. If the percentile is too high, segments that the model could have learned parametrically are masked and this could displaces useful training signal.

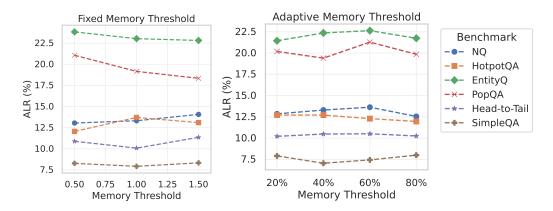


Figure 4: Answer level recall (ALR) for different memory threshold. The higher the threshold is, the fewer segments are delegated to external memory.

Trade-off between accuracy and latency Figure 5 compares the average number of retrieval calls per question across benchmarks as a function of the memory threshold. Long-tail datasets such as POPQA and SIMPLEQA trigger retrieval far more frequently than other benchmarks, indicating the model are more reliant on external memory for long tail questions. While one might expect retrieval frequency to increase linearly with more aggressive thresholds, this is not observed in practice. The reason is that segment losses are highly skewed and clustered as shown in previous histogram, so adjusting thresholds often has non-linear effects: small changes may have little impact or trigger sudden jumps in retrieval.

4 RELATED WORK

Non-parametric lanaguge modeling One recent line research has explored integrating non-parametric memory with language models generation to reduce memorization and improve factuality. The earliest work includes kNN-LM, which interpolates language model generation and retrieved nearest neighbors at inference time to adjust output distribution (Khandelwal et al., 2019). More recent inference-time only approaches include Chunk-Distilled LM (Li et al., 2025), which speculates multi-token chunks using a retrieval datastore to accelerate generation, as well as REST (He et al., 2023) and NEST (Li et al., 2024), which combine speculative decoding with retrieval to improve efficiency and factuality.

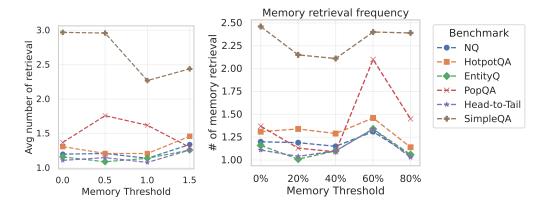


Figure 5: Average number of retrieval calls for different memory threshold. The higher the threshold is, the fewer segments are delegated to external memory.

Retrieval-augmented pretraining Several recent approaches have explored incorporating retrieval directly into the pretraining process. RETRO (Borgeaud et al., 2021) demonstrated that integrating retrieved context during pretraining improves generalization and reduces undesired memorization by allowing smaller models to match the performance of much larger purely parametric LMs. More recently and most relevant to us, Zhao et al. (Zhao et al., 2025) introduced Large Memory Language Models (LMLM), which explicitly separates factual knowledge storage from model weights by offloading specific factual details to an external database during pre-training. Our approach shares a similar motivation, but instead of we leverage signals from a partially trained model to adaptively decide what knowledge to offload.

5 CONCLUSION

We introduced semi-parametric language models with selective memory (SPLM), a framework that improves factuality on long-tail knowledge without sacrificing general capabilities. By masking high-loss segments and delegating them to external memory, SPLM preserves parametric capacity for common knowledge and reasoning while offloading rare facts. A lightweight query adapter enables contextualized retrieval aligned with the model's representations. Experiments demonstrate that SPLM achieves a favorable trade-off, outperforming memory-only baselines and rivaling larger models. This highlights selective memory as a promising direction for scaling factuality in LLMs efficiently.

There are several interesting directions for future research. We primarily targeted factual recall on short QA benchmarks, but the framework could naturally extend to other domains such as mathematical theorems or code snippets that occur in training data. An interesting question for future work is whether decoupling knowledge in this way could also improve performance on reasoning-intensive benchmarks. By externalizing rare factual content, the model could dedicate more of its parametric capacity and compute to learning reasoning strategies rather than memorizing infrequent facts. Another interesting direction would be to explore joint training of the adapter and the language model, which may further improve performance by enabling integration of external knowledge during inference.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. Retrieval-based language models and applications. In Yun-Nung (Vivian) Chen, Margot Margot, and Siva Reddy (eds.), *Proceedings*

of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts), pp. 41–46, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-tutorials.6. URL https://aclanthology.org/2023.acl-tutorials.6/.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, T. W. Hennigan, Saffron Huang, Lorenzo Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and L. Sifre. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*, 2021. URL https://api.semanticscholar.org/CorpusID: 244954723.

- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909, 2020. URL https://api.semanticscholar.org/CorpusID:211204736.
- Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D. Lee, and Di He. Rest: Retrieval-based speculative decoding. *ArXiv*, abs/2311.08252, 2023. URL https://api.semanticscholar.org/CorpusID:265157884.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. In *IEEE Transactions on Big Data*, 2019.
- Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.
- Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. *ArXiv*, abs/1911.00172, 2019. URL https://api.semanticscholar.org/CorpusID:207870430.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. 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 V. Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019. URL https://api.semanticscholar.org/CorpusID:86611921.
- Minghan Li, Xilun Chen, Ari Holtzman, Beidi Chen, Jimmy Lin, Wen tau Yih, and Xi Victoria Lin. Nearest neighbor speculative decoding for llm generation and attribution. *ArXiv*, abs/2405.19325, 2024. URL https://api.semanticscholar.org/CorpusID:270095057.
- Yanhong Li, Karen Livescu, and Jiawei Zhou. Chunk-distilled language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=nrvoWOWcyg.
- Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=u2vAyMeLMm.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.
- Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bertnetworks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. Simple entity-centric questions challenge dense retrievers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6138–6148, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.496. URL https://aclanthology.org/2021.emnlp-main.496/.

Kai Sun, Yifan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. Head-to-tail: How knowledgeable are large language models (LLMs)? A.K.A. will LLMs replace knowledge graphs? In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 311–325, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.18. URL https://aclanthology.org/2024.naacl-long.18/.

Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. arXiv preprint arXiv:2411.04368, 2024. URL https://arxiv.org/html/2411.04368.

Hongkang Yang, Zehao Lin, Wenjin Wang, Hao Wu, Zhiyu Li, Bo Tang, Wenqiang Wei, Jinbo Wang, Zeyun Tang, Shichao Song, Chenyang Xi, Yu Yu, Kai Chen, Feiyu Xiong, Linpeng Tang, and E Weinan. Memory3: Language modeling with explicit memory. *ArXiv*, abs/2407.01178, 2024. URL https://api.semanticscholar.org/CorpusID:270870116.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2018. URL https://api.semanticscholar.org/CorpusID:52822214.

Linxi Zhao, Sofian Zalouk, Christian K. Belardi, Justin Lovelace, Jin Peng Zhou, Kilian Q. Weinberger, Yoav Artzi, and Jennifer J. Sun. Pre-training large memory language models with internal and external knowledge. *ArXiv*, abs/2505.15962, 2025. URL https://api.semanticscholar.org/CorpusID:278789132.

A Appendix

- A ADDITIONAL IMPLEMENTATION DETAILS
- B SAMPLE GENERATIONS
- C PROMPT USED FOR SCORING

You are a capable model that can determine, based on the atomic factual knowledge query, whether the atomic factual knowledge belongs to a long-tailed (niche or less commonly referenced) domain. To assess the entities popularity, use the following rules:

1) The number of times the atomic factual knowledge potentially can referenced or cited by other documents on internet. 2) The number of times the atomic factual knowledge potentially can be linked to or duplicated on internet.

Please rate the atomic factual knowledge with a score between 1 and 10, where 10 indicates low popularity (long-tailed knowledge) and 1 indicates high popularity (widely referenced knowledge). Return the score in JSON format, for example: {"quality_score": predicted score } text: