PAVE: PREMISE-GROUNDED ANSWER VALIDATION AND EDITING FOR RETRIEVAL-AUGMENTED LLMS

Tianyi Huang*
Ryquo
tianyi@ryquo.com

Caden Yang
App-In Club
caden@appinclub.org

Emily Yin
App-In Club
emily@appinclub.org

Eric Wang
App-In Club
eric.wang@appinclub.org

Michael Zhang
App-In Club
michael@appinclub.org

ABSTRACT

Retrieval-augmented language models can retrieve relevant evidence yet still commit to answers before explicitly checking whether the retrieved context supports the conclusion. We present PAVE (*Premise-Grounded Answer Validation and Editing*), an inference-time validation layer for evidence-grounded question answering. PAVE decomposes retrieved context into question-conditioned atomic facts, drafts an answer, scores how well that draft is supported by the extracted premises, and revises low-support outputs before finalization. The resulting trace makes answer commitment auditable at the level of explicit premises, support scores, and revision decisions. In controlled ablations with a fixed retriever and backbone, PAVE outperforms simpler post-retrieval baselines in two evidence-grounded QA settings, with the largest gain reaching 32.7 accuracy points on a span-grounded benchmark. We view these findings as proof-of-concept evidence that explicit premise extraction plus support-gated revision can strengthen evidence-grounded consistency in retrieval-augmented LLM systems.

1 INTRODUCTION

Retrieval-augmented language models are increasingly used in settings where answers must be not only fluent, but also well supported by the evidence shown to the model. In medicine, science, law, and policy, a response can be harmful even when it sounds plausible if its conclusion is unsupported by the retrieved premises or if it overlooks an exception stated in the evidence. Recent work shows that LLMs remain brittle under hallucination, conflicting evidence, and multi-step reasoning demands, including in retrieval-augmented settings intended to improve grounding (Ji et al., 2023; Tonmoy et al., 2024; Wang et al., 2023; Feldman et al., 2024).

Retrieval-augmented generation (RAG) addresses part of this problem by conditioning answer generation on external documents rather than solely on parametric memory (Lewis et al., 2021). Yet standard RAG still leaves a consequential gap between *retrieving evidence* and *verifying that a conclusion is supported by that evidence*. In many pipelines, the model consumes one or more passages and emits a final answer in a single step. When that answer is wrong, users often cannot tell whether the failure came from retrieval, from overlooking a key sentence, from contradictory premises, or from unsupported inference beyond the evidence.

We study this gap at a deliberately narrow but practically important level: premise-grounded consistency between retrieved evidence and generated answers. The question is not whether the model can produce a longer free-form explanation, but whether its final answer remains faithful to an explicit set of evidence units derived from the retrieved context. This target is narrower than full formal reasoning, but it captures a common failure mode in deployed RAG systems: the model should have revised, softened, or withheld a conclusion once the supporting premises were inspected more carefully.

*Corresponding author.

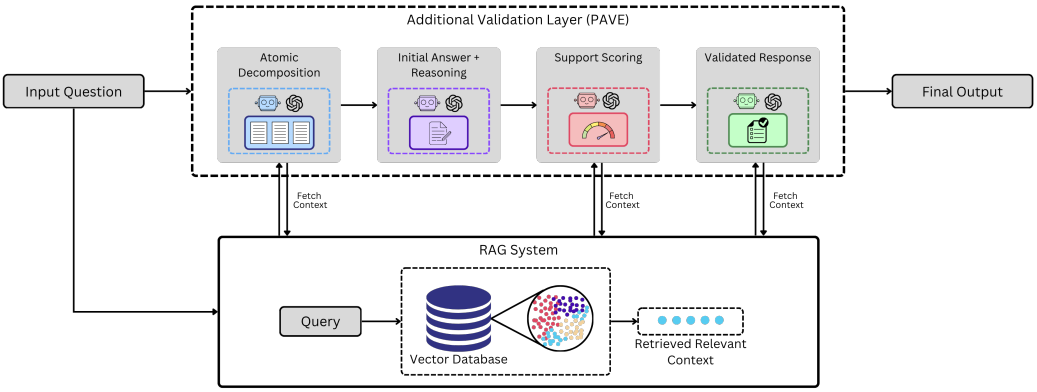

Figure 1: High-level view of PAVE. Instead of committing immediately to a final answer, the system inserts an explicit validation stage between retrieval and answer finalization: retrieved context is decomposed into atomic facts, used to draft an answer and short rationale, scored for evidence support, and revised only when the support signal is insufficient.

Recent research suggests two complementary directions for addressing this issue. First, fine-grained retrieval work shows that smaller evidence units, adaptive chunking, and query refinement can improve downstream question answering by matching retrieval granularity more closely to the task (Chen et al., 2024; Ma et al., 2023; Chan et al., 2024; Asai et al., 2023; Yan et al., 2024; Jeong et al., 2024; Sarthi et al., 2024). Second, verification-oriented methods such as SelfCheckGPT and Chain-of-Verification show that inference-time checking can reduce unsupported generations (Manakul et al., 2023; Dhuliawala et al., 2023). Our proposal combines these perspectives in a lightweight post-retrieval layer.

We present **PAVE** (*Premise-Grounded Answer Validation and Editing*), a modular workflow that converts retrieved context into atomic facts, drafts an answer, scores its premise support, and revises low-support answers before returning the final response. The method name emphasizes the central intervention: before answer commitment, the model is asked to make its working premises explicit, validate the draft against them, and edit low-support outputs. Our aim is not to present a fully optimized end-to-end RAG stack. Instead, we study PAVE as a controlled validation interface: the retriever and backbone are held fixed so that the contribution of atomic decomposition and support-gated revision can be examined directly.

Contributions. We make three contributions. First, we introduce PAVE, a post-retrieval validation layer that combines atomic decomposition, support scoring, and thresholded revision to reduce unsupported answer commitment. Second, we evaluate the method as a controlled ablation study on a fixed retriever and backbone, which isolates the effect of the validation layer and positions the paper as a controlled study. Third, we clarify the interpretability claim by formalizing the audit trace and by providing a paired SQuAD diagnostic that shows how PAVE changes decisions on the same examples.

2 Related Work

2.1 Verification and Hallucination Mitigation

A large literature studies hallucination detection and mitigation in LLMs (Ji et al., 2023; Tonmoy et al., 2024; Wang et al., 2023). Some methods rely on self-consistency or self-checking. SelfCheckGPT, for example, measures divergence across multiple sampled generations and treats disagreement as a signal of factual uncertainty (Manakul et al., 2023). Chain-of-Verification drafts an answer, asks targeted verification questions, answers them independently, and then synthesizes a revised response (Dhuliawala et al., 2023). These methods show that inference-time verification can improve reliability without retraining the base model.

PAVE shares this inference-time perspective, but differs in its verification target. Rather than checking agreement among multiple generations alone, it checks whether a draft conclusion is supported by an explicit question-conditioned set of evidence units extracted from retrieved context. This makes the validation problem more local and easier to audit: the central question becomes whether the answer is supported by the retrieved premises, not merely whether the model agrees with itself.

2.2 Fine-grained retrieval and RAG optimization

The retrieval side of the literature has moved well beyond fixed passage retrieval. Query rewriting methods improve retrieval by transforming user questions into retrieval-friendly forms (Ma et al., 2023; Chan et al., 2024). Adaptive systems such as Self-RAG, CRAG, and Adaptive-RAG dynamically decide when to retrieve or how aggressively to critique retrieved evidence (Asai et al., 2023; Yan et al., 2024; Jeong et al., 2024). Hierarchical retrieval methods such as RAPTOR support long-document reasoning through recursive summaries (Sarthi et al., 2024). Other work argues that retrieval granularity itself matters: Dense X Retrieval shows that proposition-level indexing can outperform sentence- or passage-level retrieval for open-domain QA (Chen et al., 2024).

PAVE is complementary to these directions. It does not prescribe a specific retriever, chunking strategy, or query-rewriting policy. Instead, it operates after retrieval and improves the *reasoning interface* between retrieved context and answer generation. Better retrieval still helps PAVE, but PAVE addresses a distinct failure mode: unsupported answer commitment even when useful evidence is already present.

2.3 Logical reasoning and solver-augmented methods

Work on logical reasoning benchmarks shows that LLMs remain unreliable on formal and semi-formal reasoning tasks despite strong surface fluency. FOLIO evaluates natural-language reasoning with first-order logic structure (Han et al., 2024), while LogicBench tests a broader range of logical patterns, including negation and non-monotonic reasoning (Parmar et al., 2024). Solver-augmented systems such as Logic-LM combine LLMs with symbolic solvers to improve faithfulness on logic-heavy tasks (Pan et al., 2023). Related work on logical preference consistency argues that robust reasoning should satisfy invariances such as transitivity and negation consistency (Liu et al., 2025).

Compared with solver-based pipelines, PAVE does not perform formal proof search. Instead, it introduces an explicit premise-grounded validation step for evidence-based question answering. This makes it useful both as a standalone consistency layer and as a front end to stronger downstream verifiers: the atomic facts produced by PAVE provide a natural intermediate representation that can be checked by an entailment model or symbolic module.

3 Method

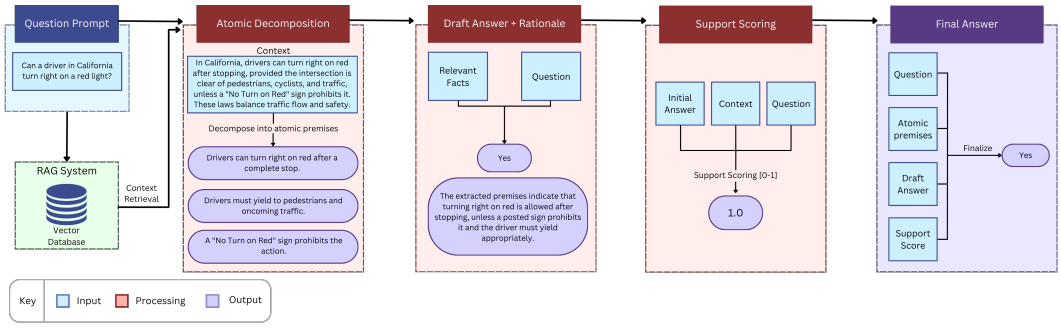

Figure 2: Illustrative execution trace for PAVE on a traffic-rule question. The retrieved context is decomposed into atomic premises, used to produce a draft answer and rationale, scored for support, and then finalized. In this example, the decomposition makes the governing rule, the relevant condition, and the exception explicit before final answer commitment.

3.1 Problem setup

Let q denote a user question and let a retriever return context

$$C = \mathcal{R}(q) = \{c_1, \ldots, c_m\}, \tag{1}$$

where each c_j is a retrieved passage or document chunk. Standard RAG directly maps (q, C) to an answer. PAVE inserts an explicit validation layer that separates evidence extraction, draft generation, support scoring, and answer finalization.

The atomic decomposition stage produces a question-conditioned fact list

$$F = \mathcal{D}(q, C) = [f_1, \ldots, f_n], \tag{2}$$

where each f_i is intended to be a concise, standalone statement that is directly relevant to answering q. A generator then drafts an answer and short rationale,

$$(a^{(0)}, r^{(0)}) = \mathcal{G}(q, F), \tag{3}$$

and a support scorer returns a scalar score

$$s = \mathcal{S}(q, F, a^{(0)}, r^{(0)}) \in [0, 1], \tag{4}$$

where higher values indicate stronger support of the draft answer by the atomic fact set. Finalization is governed by a fixed revision threshold $\tau \in [0, 1]$:

$$a^\star = \begin{cases} a^{(0)}, & \text{if } s \geq \tau, \\ \mathcal{U}(q, F, a^{(0)}, r^{(0)}, s), & \text{if } s < \tau, \end{cases} \tag{5}$$

where $\mathcal{U}$ is a revision step invoked only for low-support drafts. The corresponding audit trace is

$$\mathcal{T} = (F, a^{(0)}, r^{(0)}, s, a^\star). \tag{6}$$

This trace is the main explainability artifact returned by the system.

3.2 Atomic decomposition as premise extraction

The first stage converts retrieved context into a compact set of premises. In practice, this decomposition is question-conditioned: the model is asked to retain only information needed to answer q, and to express that information in short, self-contained statements rather than long copied passages. This has two benefits. First, it reduces distraction from irrelevant text in later stages. Second, it makes support checking more precise because exceptions and preconditions remain explicit instead of being buried inside longer paragraphs. In qualitative traces, the most useful decompositions separate the main claim, the relevant condition, and any exception into distinct premises, since those are precisely the elements the support scorer must inspect.

The core PAVE pipeline does *not* require learned or hand-crafted importance weights over atomic facts. We explored salience weighting in an earlier prototype, but it proved less stable than the simpler decomposition-plus-scoring design. For clarity, we therefore treat importance weighting as an ablation rather than part of the main method.

3.3 Support scoring and revision

Given the atomic fact list F, the generator produces a draft answer $a^{(0)}$ and short rationale $r^{(0)}$. A separate scoring prompt then evaluates whether the conclusion is sufficiently supported by the premises. This score is best interpreted as a *premise-support signal*: values near 1 indicate that the answer follows directly or is strongly supported by the extracted facts, while lower values indicate weak support, contradiction, or missing conditions.

The score is not merely reported; it changes system behavior. High-support drafts are finalized immediately, while low-support drafts trigger revision. This gating mechanism is central to the design: the point of the scorer is not only to annotate the answer, but to prevent premature commitment when the available premises do not support the conclusion strongly enough.

3.4 What the trace adds beyond direct LLM interaction

A direct interaction with an LLM can already produce an answer and an explanation, so it is reasonable to ask what PAVE adds. The key difference is structural. A free-form explanation is post hoc and unconstrained; it does not tell the user which evidence units were explicitly considered, whether the answer crossed a validation threshold, or whether a revision step was triggered. By contrast, PAVE returns a machine-readable trace with three concrete affordances: evidence locality through atomic premises, an explicit decision signal through the support score, and failure localization through the separation of retrieval, draft generation, and revision. This is why the method is useful even when the final answer text overlaps with what a strong LLM might have produced directly.

4 Experimental Setup

4.1 Datasets

We report results on two evidence-grounded QA settings.

PubMedQA. PubMedQA is a biomedical benchmark whose answers are restricted to `yes`, `no`, or `maybe` (Jin et al., 2019). The `maybe` label makes the task especially relevant to evidence-sensitive reasoning because inconclusive or mixed evidence must not be forced into an overconfident binary decision. We report results on a 1,000-question labeled subset.

SQuAD. SQuAD is a factoid QA benchmark in which answers are short spans supported by the context (Rajpurkar et al., 2016). Relative to PubMedQA, it places greater emphasis on precise evidence alignment and less on three-way uncertainty calibration. We report results on a 1,000-example subset and an additional 5,000-example subset used as a scale check for the final PAVE pipeline.

4.2 Ablation design

The original draft presented several pipelines side by side in a way that obscured the main contribution. We recast them here as a clean ablation study.

- **Baseline RAG**: one-shot answer generation from retrieved context with no atomic decomposition and no support scoring.
- **RAG + importance weighting**: an exploratory variant that decomposes context into atomic statements and attaches salience scores, but does not use the final PAVE design.
- **RAG + support scoring**: a two-stage variant that scores a draft answer against retrieved context but does not use atomic decomposition.
- **PAVE**: the full method, combining atomic decomposition with support scoring and thresholded revision.

This organization isolates the two core questions behind the method: whether decomposition helps on its own, whether scoring helps on its own, and whether the combination is stronger than either component in isolation.

4.3 Implementation notes

All compared variants use the same underlying LLM backbone, GPT-4o mini, in the original experiments (OpenAI, 2024). The retrieval backend is held fixed across the ablations so that observed differences reflect the validation layer rather than a change in retriever or base model. Each stage is implemented with a specialized prompt for decomposition, draft generation, support scoring, and, when needed, revision. Because PAVE is training-free, it introduces no task-specific training cost; its main practical cost is additional inference latency. The full pipeline uses three LLM calls by default and invokes a fourth call only when revision is triggered.

We report answer-level accuracy as the primary metric. For PubMedQA, this is exact agreement with the gold `yes`/`no`/`maybe` label. For SQuAD, the available experiment logs contain binary correctness

labels for each prediction. These labels credit semantically correct span-equivalent answers rather than only verbatim string matches, so the reported SQuAD numbers should be read as answer-level accuracy under a fixed shared protocol rather than as canonical EM/F1. All compared SQuAD variants use the same labeling rule.

5 RESULTS

Table 1: Main results on two evidence-grounded QA tasks, PubMedQA and SQuAD (1,000-example subset).

Pipeline	**PubMedQA**	**SQuAD**
Baseline RAG	71.20	62.40
RAG + importance weighting	69.80	47.70
RAG + support scoring	71.90	63.60
PAVE (full)	**73.30**	**95.10**

The full PAVE pipeline achieves the strongest performance on both datasets. On PubMedQA, the gain over the baseline is modest but consistent (+2.1 points), corresponding to a 7.3% relative reduction in error. On SQuAD, the gain is much larger (+32.7 points on the 1,000-example subset), which corresponds to an 87.0% relative reduction in error against the baseline and suggests that atomic premise extraction and support-gated revision are especially effective when the answer must stay tightly aligned with the available evidence.

Support scoring without atomic decomposition yields only a small improvement over baseline, suggesting that a verification stage is useful but limited when it must reason over raw passages. The importance-weighting variant performs worse than the baseline on both datasets, implying that salience annotations can amplify extraction noise when they are treated as part of the decision rule. The strongest results therefore come from the combined design: premise extraction plus support scoring plus revision.

On the larger 5,000-example SQuAD subset, the full PAVE pipeline retains 92.76 accuracy. While lower than the 95.10 observed on the smaller subset, this still suggests that the improvement is not confined to a tiny pilot slice. The overall pattern is consistent with the method's design objective: when answers are short and directly recoverable from retrieved evidence, explicitly checking premise support before finalization can substantially reduce avoidable errors.

5.1 PAIRED TRACE ANALYSIS ON SQUAD

Logged outputs for both the RAG + support scoring variant and the full PAVE pipeline are available for a 100-example SQuAD slice, which we use for a paired diagnostic comparison. Because this analysis is paired at the example level, it helps explain where the aggregate gain comes from rather than only restating benchmark-level accuracy.

Figure 3 shows that PAVE corrects 23 of the 25 errors made by the support-scoring variant and introduces only one new error, reducing the error count by 88% on the same examples. The same slice also shows a marked tightening of answer form: using a simple whitespace token count on the logged outputs, median answer length drops from 71 tokens for the support-scoring variant to 2 tokens for PAVE, exactly matching the median gold answer length. Within this slice, the largest gain appears on 3–4 token answers, where accuracy rises from 66.7 to 96.7.

Although this evidence is indirect, it is consistent with the claim that decomposition quality matters. When the answer is first grounded in a compact set of question-conditioned premises, the downstream scorer appears better able to localize the answer span and suppress irrelevant explanation. A more direct evaluation of decomposition faithfulness, coverage, and failure modes remains an important next step.

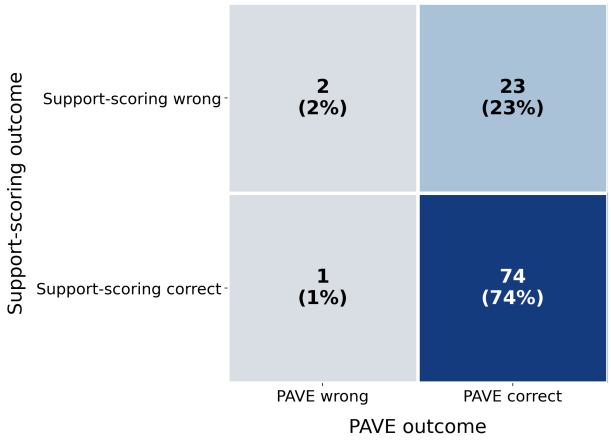

Figure 3: Paired outcome transition on 100 shared SQuAD examples comparing the support-scoring variant with PAVE.

6 Scope, Limitations, and Future Work

This paper is best read as a preliminary study of a post-retrieval validation interface. The experiments intentionally keep the retriever and backbone fixed so that the effect of the validation layer can be isolated. That design makes the central claim easier to interpret, but it also narrows the scope of what the current evidence can establish.

Several limitations remain. Paired per-example outputs are currently available only for a diagnostic 100-example SQuAD slice, which constrains how far the present analysis can go beyond aggregate results. In addition, decomposition quality is assessed only indirectly through downstream behavior; a more direct study of decomposition faithfulness, coverage, and failure modes would strengthen the claim that the atomic representation is the right interface for validation. The method also introduces additional inference overhead, since decomposition and support scoring require extra model calls in exchange for greater auditability and control. Finally, because the current experiments hold both the backbone and the retriever fixed, broader evaluation on stronger language models and stronger retrieval pipelines remains an important next step.

From an application perspective, PAVE is most useful when reliability is a main priority. In high-stakes settings, returning an answer together with its supporting premises and a revision signal can make human review substantially easier than relying on a polished but unstructured explanation. A high support score, however, should not be interpreted as a guarantee of truth. The score reflects agreement with the retrieved evidence, and that evidence can itself be incomplete, outdated, or biased.

7 Conclusion

This paper studies a simple question with practical consequences for retrieval-augmented language models: should a system commit to an answer before it has explicitly checked whether the retrieved evidence supports that conclusion? PAVE is one answer to that question. By inserting an explicit premise-grounded validation stage between retrieval and final answer commitment, it turns evidence use into something more inspectable and more revisable than one-shot generation alone. Across the reported ablations, the clearest pattern is that support scoring becomes substantially more useful when it operates over compact, question-conditioned premises rather than over raw retrieved passages. We view these results as proof-of-concept evidence for a broader design principle: in evidence-grounded generation, making premises explicit before finalizing an answer can improve not only accuracy, but also the discipline with which answers are formed.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023. URL https://arxiv.org/abs/2310.11511.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. Rq-rag: Learning to refine queries for retrieval augmented generation, 2024. URL https://arxiv.org/abs/2404.00610.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. Dense x retrieval: What retrieval granularity should we use?, 2024. URL https://arxiv.org/abs/2312.06648.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models, 2023. URL https://arxiv.org/abs/2309.11495.

Philip Feldman, James R. Foulds, and Shimei Pan. Ragged edges: The double-edged sword of retrieval-augmented chatbots, 2024. URL https://arxiv.org/abs/2403.01193.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alex Wardle-Solano, Hannah Szabo, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander R. Fabbri, Wojciech Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. Folio: Natural language reasoning with first-order logic, 2024. URL https://arxiv.org/abs/2209.00840.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity, 2024. URL https://arxiv.org/abs/2403.14403.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, March 2023. ISSN 1557-7341. doi: 10.1145/3571730. URL http://dx.doi.org/10.1145/3571730.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering, 2019. URL https://arxiv.org/abs/1909.06146.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL https://arxiv.org/abs/2005.11401.

Yinhong Liu, Zhijiang Guo, Tianya Liang, Ehsan Shareghi, Ivan Vulić, and Nigel Collier. Aligning with logic: Measuring, evaluating and improving logical preference consistency in large language models, 2025. URL https://arxiv.org/abs/2410.02205.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting for retrieval-augmented large language models, 2023. URL https://arxiv.org/abs/2305.14283.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models, 2023. URL https://arxiv.org/abs/2303.08896.

OpenAI. Gpt-4o mini: advancing cost-efficient intelligence, 2024. URL https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning, 2023. URL https://arxiv.org/abs/2305.12295.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. Logicbench: Towards systematic evaluation of logical reasoning ability of large language models, 2024. URL https://arxiv.org/abs/2404.15522.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016. URL https://arxiv.org/abs/1606.05250.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. Raptor: Recursive abstractive processing for tree-organized retrieval, 2024. URL https://arxiv.org/abs/2401.18059.

S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. A comprehensive survey of hallucination mitigation techniques in large language models, 2024. URL https://arxiv.org/abs/2401.01313.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, Yidong Wang, Linyi Yang, Jindong Wang, Xing Xie, Zheng Zhang, and Yue Zhang. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity, 2023. URL https://arxiv.org/abs/2310.07521.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation, 2024. URL https://arxiv.org/abs/2401.15884.