Qeios

Peer Review

Review of: "TrustRAG: Enhancing Robustness and Trustworthiness in RAG"

Iliano Fasolino[1]

1. University of Milan, Italy

The problem the paper takes on is the right one. Corpus poisoning that can outnumber clean neighbors is exactly where majority voting and simple aggregation break, and Table 1 makes that failure visible for RobustRAG Keyword, InstructRAG ICL, and ASTUTE RAG under PoisonedRAG. Framing TrustRAG as a training-free, post-retrieval filter that sits after the retriever, with code at https://github.com/HuichiZhou/TrustRAG, is a clean engineering posture. The two-stage split—K-means with ROUGE-L n-gram preservation, then conflict removal that mixes internal knowledge, consolidation, and self-assessment—is also a coherent design. My concern is that several strong claims in the abstract and §6 sit ahead of what the threat model and the Stage-1 numbers can carry.

Start with the “first” and “State-of-the-Art” language. The introduction says TrustRAG is “the first defense framework to effectively defend corpus poisoning attack where the number of poisoned documents exceeds the number of clean ones,” and later claims SOTA against advanced RAG systems. Related Work already discusses RobustRAG and perplexity / paraphrasing defenses; the novelty that is actually shown is better behavior under multi-injection PoisonedRAG on NQ, HotpotQA, and MS-MARCO with Mistral-Nemo-12B, Llama-3.1-8B, and GPT-4o. That is enough. The priority wording should be cut or tightly scoped to the multi-injection setting the authors actually stress-test.

The Stage-1 results themselves argue for more humility. On PIA, which the authors treat as a single-injection case, TrustRAG stage 1 often looks worse than the later full pipeline: for Mistral-Nemo-12B, the HotpotQA PIA line is 37.0 / 51.0 ASR before Conflict Removal and 77.0 / 9.0 after; Llama-3.1-8B moves from 28.0 / 61.0 to 73.0 / 3.0 on the same column. So the clustering stage is not a general poison filter; it mainly helps when malicious neighbors form a tight, multi-document cluster. That matches the embedding-density story in §5.1 and Figure 3, and it also means an attacker who injects one strong inductive passage, or who spreads several poisons so they do not share a dense cluster, is outside what Stage 1 was built for. I

 doi.org/10.32388/IV9WW4

would like an explicit adaptive discussion: what happens if the adversary knows k = 2 and optimizes for embedding diversity, or plants poisons that look like clean neighbors under the chosen SimCSE / BERT / BGE encoders in Table 2?

Conflict Removal raises a different issue. The pipeline asks the same LLM family to (1) emit "internal knowledge," (2) mark conflicts between that text and retrieved passages, and (3) self-assess which source to trust. Under corrupted retrieval, that loop is useful when the model's parametric memory is roughly right. It is fragile when the model is wrong, outdated, or itself steerable by the remaining poison that Stage 1 let through. The paper treats internal knowledge as a consensus from pre-training; it does not measure how often that consensus is factually wrong on the same NQ / HotpotQA items, nor how often self-assessment prefers a fluent wrong internal draft over a correct external passage. Without that error breakdown, "trustworthiness" is mostly attack-success reduction, not a calibrated account of when the system should refuse or abstain.

I also read the clean columns in Table 1 as a real trade-off, not a free lunch. On several clean cells, TrustRAG stage 2 is close to Vanilla RAG, and sometimes a point or two below (for example, Mistral on NQ clean: 66 vs. 69). That may be acceptable, but the abstract's "maintains high contextual relevance while strengthening defenses" should report the clean drop beside the ASR win, especially for readers who will pay the ~2× inference cost shown in Table 3. The RedditQA numbers in §5.6 (Vanilla 27.3% ACC / 43.8% ASR vs. TrustRAG 72.2% / 11.9%) are encouraging as a non-synthetic check; they need the same transparency about sample size, how "factual errors" were labelled, and whether the LLM's internal knowledge simply dominated noisy Reddit text.

Two reporting gaps would make the next version much easier to trust. First, say clearly how many queries sit behind each ACC / ASR cell and whether decoding was single-shot (the text already prefers one internal-knowledge draw over multi-temperature sampling—good for cost, bad for variance). Second, deposit the Stage-1 cluster assignments and the Conflict Removal rationales for a public subset, not only aggregate tables. The prompts are pointed to in Appendix D; releasing them with a few worked poisoned examples would let others probe the circular self-assessment path I worry about.

I would keep the multi-injection defense goal and the open-source filter design. I would rewrite the priority / SOTA sentences, add an adaptive and single-poison stress test aimed at breaking k = 2, report clean-accuracy cost next to ASR, and measure how often internal knowledge and self-assessment are themselves wrong. With those fixes, this becomes a solid robustness paper rather than an over-claimed "trust" framework.

Declarations

Potential competing interests: No potential competing interests to declare.