Understanding and Improving Information Preservation in Prompt Compression for LLMs

Weronika Łajewska [1], Momchil Hardalov [2], Laura Aina [2], Neha Anna John [2], Hang Su [2], Lluís Màrquez [3]

[1] University of Stavanger, Work done during an internship at AWS AI Labs, [2] Amazon, [3] Technical University of Catalonia (UPC), Work done while at Amazon

WHY DO WE NEED PROMPT COMPRESSION?

LLMS ARE POWERFUL BUT COSTLY

- Large prompts → high computation and latency
- Redundant or irrelevant info → degraded performance, biased outputs

INFORMATION-INTENSIVE TASKS MAKE IT WORSE

- Tasks like QA, summarization, retrieval-augmented generation
- Prompts easily exceed model limits

EXISTING COMPRESSION HELP — BUT HOW WELL?

- Many techniques focus only on shorter input length
- Limited understanding of what information is lost and how it impacts performance

OUR WORK

- We propose a framework for evaluating prompt compression in long-context generation.
- We analyze representative soft (xRAG, PISCO) and hard (LLMLingua) methods, revealing their key strengths and limitations.
- We introduce improved xRAG variants addressing information loss at high compression rates

HOLISTIC EVALUATION FRAMEWORK FOR PROMPT COMPRESSION

DOWNSTREAM TASK PERFORMANCE

Method	HotpotQA (EM)	HotpotQA* (EM)	arXiv-sum. (F1)	QuAC (F1)	TriviaQA (EM)	GSM8K (EM)
Mistral-7B	0.664	0.772	0.834	0.869	0.773	0.477
Mistral-7B (no cont.)	0.276 (-58%)	0.276 (-64%)	---	0.834 (-4%)	0.590 (-24%)	0.440 (-1%)
xRAG	0.297 (-55%)	0.374 (-52%)	0.803 (-4%)	0.838 (-4%)	0.691 (-11%)	0.336 (-30%)
PISCO	0.297 (-55%)	0.589 (-24%)	0.818 (-2%)	0.861 (-1%)	0.738 (-5%)	0.393 (-18%)
LLMLingua	0.297 (-55%)	0.696 (-10%)	0.805 (-4%)	0.846 (-3%)	0.727 (-6%)	0.305 (-36%)

→ All methods cause performance drops, especially on multi-hop reasoning
→ Smaller gap on summarization and conversational QA (high-level info tasks)
→ Compressed ICL examples (GSM8K) perform worse than no examples

GROUNDING

Method	HotpotQA	HotpotQA*	arXiv-sum.	QuAC	TriviaQA	GSM8K
Mistral-7B	0.8	0.75	0.97	0.93	0.78	0.5
xRAG	0.52	0.57	0.39	0.45	0.73	0.42
PISCO	0.59	0.76	0.74	0.63	0.84	0.48
LLMLingua	0.45	0.75	0.62	0.49	0.72	0.44

→ Compression reduces grounding by 30–50 points, especially in HotpotQA, QuAC, and arXiv-sum
→ xRAG: high compression → more hallucinations, even in first claims
→ LLMLingua: fewer hallucinations (text-level compression preserves context)
→ PISCO: most faithful outputs despite high compression (it benefits from sequence-level knowledge distillation)

INFORMATION PRESERVATION

Method	Data	Encodings	BERTScore F1	Preserved Entities
xRAG	Unseen	1 per sample	0.66	0.28
		1 per sent.	0.42	0.19
	Seen	1 per sample	0.65	0.25
		1 per sent.	0.35	0.12
PISCO	Unseen	8 per sample	0.89	0.49
		8 per sent.	0.9	0.59

→ xRAG struggles to reconstruct details (loss increases with one token per sentence)
→ xRAG preserves topics, but loses fine-grained info (esp. dates, numbers, people)
→ PISCO achieves higher semantic and entity preservation

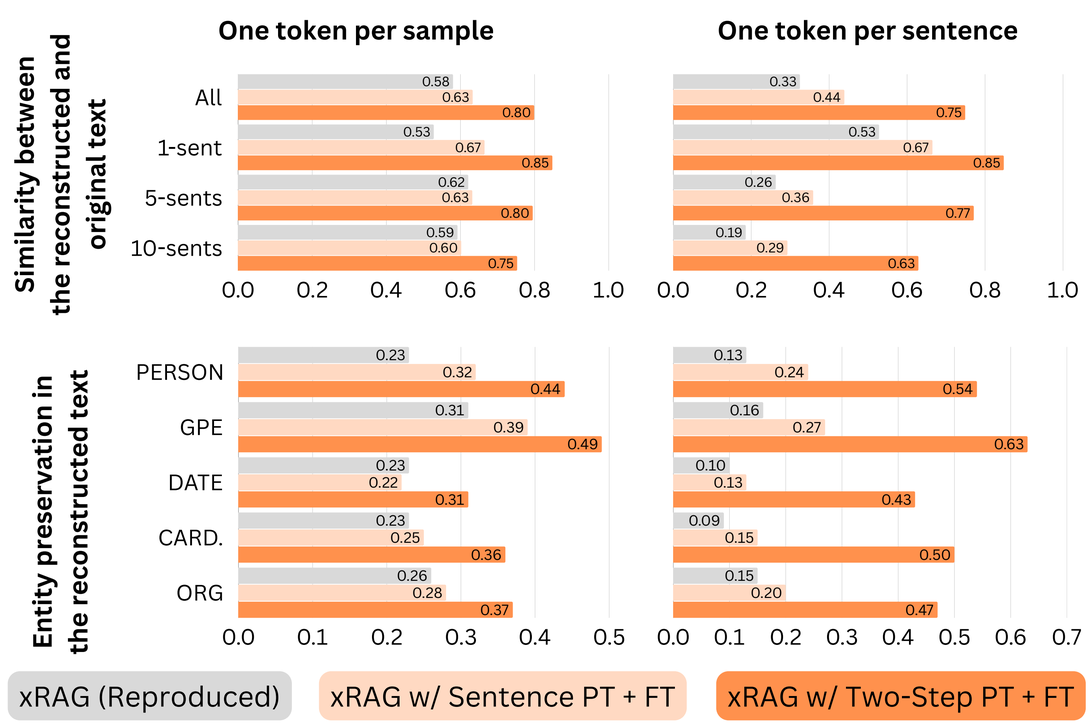
IMPROVING XRAG FOR BETTER INFORMATION PRESERVATION

SENTENCE-LEVEL PRE-TRAINING & FINE-TUNING (SENTENCE PT + FT)

- Pre-train on single sentences for finer granularity.
- Replace token-based chunking with sentence-based segmentation.
- Each sentence → one xRAG token.

TWO-STEP PRE-TRAINING (TWO-STEP PT + FT)

- Stage 1: Encode one sentence per sample.
- Stage 2: Chunk into sentences with separate encoding.
- Helps model integrate info across multiple tokens.



LESSONS LEARNED

Current compression methods struggle with detail preservation, especially at high compression rates and for long contexts.

xRAG tokens tend to encode general topics, not fine-grained details (e.g., names, dates, numbers).

Modifying training to control information granularity improves retention, grounding, and downstream performance — even with few tokens.

FUTURE DIRECTIONS

MULTI-TOKEN CONTEXT UNDERSTANDING

Enable reasoning across multiple soft prompt tokens and sequence-level training (as in PISCO) to improve grounding and detail capture

TASK-SPECIFIC EMBEDDINGS

Replace dense retrieval encoders with instruction-tuned embeddings for richer, detail-aware representations.

CONTEXT-AWARE ADAPTIVE COMPRESSION

Adjust compression dynamically by task and input complexity to move away from a one-size-fits-all compression ratio

HYBRID SOFT–HARD PROMPTING

Combine soft prompts for efficiency with hard prompts to preserve key factual details (e.g., entities, numbers).