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 conducted 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

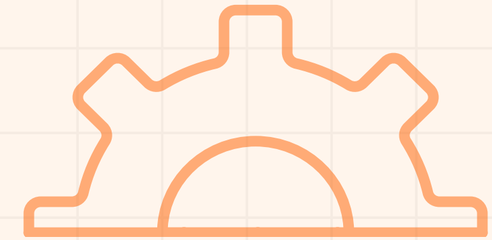
INFORMATION-INTENSIVE TASKS MAKE IT WORSE

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

EXISTING COMPRESSION METHODS HELP — BUT HOW WELL?

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

What makes a good prompt compression method?



HOLISTIC EVALUATION FRAMEWORK

Downstream Task Performance

Information Preservation

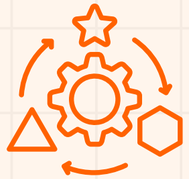
Grounding to Original Context

→ Evaluation across multiple generation tasks: multi-hop QA, conversation QA, long-document summarization, and mathematical reasoning (850–6.5K tokens).

Prompt Compression Methods

XRAG (CHENG ET AL., 2024)

- Encodes the full input into a single embedding token.
- Uses a trainable modality bridge to map encoder outputs to the LLM embedding space.



PISCO (LOUIS ET AL., 2025)

- Builds on xRAG with encoder and decoder adapters.
- Compresses context into embedding vectors and generates responses using both document and query embeddings.

LLMLINGUA (JIANG ET AL., 2023)

- Applies dynamic compression via a budget controller.
- Uses token-level pruning to shorten text while maintaining meaning.
- Outputs a compressed textual prompt for the LLM.

[1] Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, SiQing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. *xRAG: Extreme context compression for retrieval-augmented generation with one token*. In The Thirty-eighth Annual Conference on Neural Information Processing Systems.
[2] Maxime Louis, Hervé Déjean, and Stéphane Clinchant. 2025. *PISCO: Pretty Simple Compression for Retrieval-Augmented Generation*. In Findings of the Association for Computational Linguistics: ACL 2025, pages 15506–15521, Association for Computational Linguistics.
[3] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. *LLMLingua: Compressing prompts for accelerated inference of large language models*. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 13358–13376

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%)

Model performance on long-context datasets. Percentages in brackets show the relative drop from the Mistral-7B baseline, calculated as (score - mistral_score) / mistral_score. "No context" (no cont.) for GSM8K means removing ICL demos, while for arXiv-sum, the context is the document itself, making this setup not applicable.

→ All methods cause performance drops (3–55%), especially on multi-hop reasoning (HotpotQA)

→ Compression removes key details, limiting reasoning and aggregation

→ Smaller gap on summarization and conversational QA (high-level info tasks)

→ Compressed ICL examples (GSM8K) perform worse than no examples → poor interpretability

Grounding Results

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

FABLES grounding scores for responses generated with different methods, averaged over 5 random sets of 100 samples (stdev (σ) < 0.04).

→ 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 → benefits from sequence-level knowledge distillation.

Information Preservation in Soft Prompting

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

Information preservation results: BERTScore F1 between original and reconstructed context for different context lengths, and fraction of preserved entities.

→ xRAG struggles to reconstruct details → avg. BERTScore F1 ≈ 0.66

→ Loss increases (~20–30 pts) with one token per sentence

→ Preserves topics, but loses fine-grained info — esp. dates, numbers, people

→ PISCO achieves higher semantic and entity preservation, consistent across encoding levels

Improving xRAG for Better Information Preservation

MOTIVATION

- Original xRAG fails to retain fine-grained information in long contexts.
- Goal: enhance detail preservation while keeping high compression efficiency.

EXPECTED BENEFITS

- Better handling of sentence-level context.
- Improved grounding and factual consistency.
- Higher entity preservation and semantic retention.

1 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.

2 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.

Improved xRAG — Results and Insights

→ Improved xRAG variants produce more faithful and less hallucinated responses.
→ Higher alignment between generated text and source documents.

→ Entity retention ↑ from 13% to 50% (multi-token encoding).
→ Semantic similarity (BERTScore F1) ↑ from 0.19 to 0.63.

Information preservation results for xRAG variants. Similarity is measured with BERTScore between the original and reconstructed text. Entity preservation is based on EM of entities in the reconstruction.

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.
- Sequence-level training (as in PISCO) improves 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.
- 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).

Summary

OUR WORK

- Proposed a framework for evaluating prompt compression in long-context generation.
- Analyzed soft (xRAG, PISCO) and hard (LLMLingua) methods — revealing key strengths and limitations.
- Introduced improved xRAG variants addressing information loss at high compression rates.

KEY RESULTS

- +23% improvement on downstream tasks.
- +8 BERTScore increase in grounding.
- 2.7× more entities preserved compared to the original xRAG.

MAIN CONTRIBUTIONS

- Identified major limitations of existing prompt compression methods.
- Proposed a holistic evaluation framework beyond downstream metrics.
- Demonstrated that granularity control during compression significantly improves performance and fidelity.

Thank you for your attention!

CODE REPOSITORY

PAPER

Feel free to reach out if you have any questions: lajewska@amazon.lu