

A Lightweight Extractive Compression Pipeline for Long-Context Retrieval-Augmented Generation

Alikhan Zhamankhan¹, Iskander Akhmetov¹, Alex Pak¹, Assel Zhaxylykova¹,
Alexander Gelbukh^{2,*}

¹ Kazakh-British Technical University, Almaty,
Kazakhstan

² Instituto Politécnico Nacional,
Center for Computing Research, Mexico City,
Mexico

al_zhamankhan@kbtu.kz, gelbukh@cic.ipn.mx

Abstract. Long contexts in Retrieval-Augmented Generation (RAG) cause high latency and computational costs. We propose a lightweight, extractive sentence-level summarization method that extracts the top-k relevant sentences and expands them with a dynamic window (w) of adjacent sentences to preserve semantic flow. Evaluated on both single-hop and multi-hop QA datasets, our approach with a minimal window ($w = 1$) successfully mitigates semantic fragmentation and significantly reduces token consumption. On single-hop tasks, it achieves an optimal compression ratio of 27.5x, with extreme configurations reaching up to 73x while outperforming the full-document baseline. On multi-hop tasks, the method provides a highly efficient trade-off, maintaining competitive accuracy while drastically cutting computational overhead. While the method provides an efficient trade-off, it remains slightly behind the full-context baseline in multi-hop tasks due to the complexity of fact-chaining.

Keywords. Retrieval-augmented generation, extractive summarization, context compression, large language models.

1 Introduction

Retrieval-Augmented Generation (RAG) [10] has emerged as the de facto standard for mitigating hallucinations and injecting up-to-date knowledge into Large Language Models (LLMs) [16]. However, as RAG systems scale to process extensive

multi-document corpora, they encounter a critical bottleneck: the context window limitation. Feeding thousands of retrieved tokens into an LLM not only incurs prohibitive computational costs due to the quadratic complexity of the attention mechanism [17], but also severely degrades model performance. Recent studies highlight the “Lost in the Middle” phenomenon [12], demonstrating that LLMs systematically fail to utilize relevant information buried within long input sequences.

To address these challenges, multi-document text summarization has become an active area of research. Traditional summarization approaches generally fall into two categories: *abstractive* and *extractive*.

Abstractive methods employ specialized models to rewrite or summarize documents, which introduces unacceptable latency overheads for real-time question answering applications. Token-level extractive methods (e.g., LLMingua) selectively drop non-essential words to shorten the prompt. While effective in reducing token counts, aggressive token pruning often destroys the syntactic structure and grammatical coherence of the text, leading to coreference loss and confusing the generative model. Furthermore, standard RAG pipelines relying on static chunking (e.g., 500-word blocks) suffer from high noise-to-signal ratios, as a large chunk may contain only a single

relevant sentence surrounded by hundreds of irrelevant words.

In this paper, we propose a robust zero-shot extractive summarization pipeline that operates strictly at the sentence level. Our method combines the high recall of bi-encoder retrieval with the surgical precision of cross-encoder reranking. To prevent the semantic fragmentation caused by extracting isolated sentences, we introduce a *Sentence Windowing* mechanism. As illustrated in Figure 1, our pipeline systematically appending W adjacent sentences to the highly scored target sentence, our pipeline dynamically reconstructs the local context, resolving coreferences and maintaining narrative continuity without overwhelming the LLM with unnecessary text.

We evaluate our proposed pipeline across diverse open-domain and multi-hop QA datasets¹, including TriviaQA, Natural Questions, HotpotQA, and 2WikiMultiHopQA. Our empirical results reveal a significant trade-off between compression ratio and generative accuracy.

Our main contributions are summarized as follows:

- We introduce a zero-shot sentence-level context compression pipeline requiring no additional fine-tuning, leveraging a cross-encoder reranker coupled with dynamic sentence windowing.
- We find that zero-window sentence extraction negatively impacts performance because it breaks the logical flow of the text. Reintroducing a minimal local context ($w = 1$) resolves these missing links and leads to a significant improvement in the F1 metric.
- We show that our method can achieve up to a $27.5\times$ compression ratio (filtering out approximately 96% of retrieved noise) on single-hop QA tasks while maintaining a highly competitive F1-score, effectively mitigating the “Lost in the Middle” problem and significantly reducing token consumption.

¹To facilitate reproducibility, the complete source code and instructions for downloading the standardized datasets used in our experiments are publicly available at <https://github.com/alikhan2002/Adaptive-RAG-Compression>

2 Related Work

2.1 Long-Context Limitations in RAG

Although modern LLMs support increasingly long context windows, simply providing more retrieved text does not guarantee better reasoning quality. Prior work has shown that models often fail to effectively use information placed in the middle of long prompts, a phenomenon commonly referred to as *Lost in the Middle* [12]. In RAG settings, this problem is further amplified by retrieval noise, where irrelevant passages dilute the useful evidence required for answer generation.

2.2 Retrieval and Chunking Strategies

Several studies have attempted to improve retrieval quality before generation by modifying how documents are represented or segmented. Examples include methods such as *Landmark Embedding* and *Late Chunking*, which aim to preserve better document structure or improve retrieval granularity. Even though these chunking strategies produce better retrieval results, the language model still has to process thousands of tokens. The actual size of the prompt sent to the LLM is left unaddressed.

2.3 Extractive Text Summarization

Extractive text summarization aims to identify and extract the most salient sentences from a source text to form a concise summary. Early graph-based methods, such as LexRank [2] and TextRank [14], established the foundational principles of sentence importance scoring. With the advent of transformer architectures, approaches like BERTSum [13] and MatchSum [24] significantly advanced the field by framing sentence extraction as sequence classification or semantic matching tasks. In multi-document scenarios, frameworks such as PRIMERA [20] have been proposed to align and condense information across diverse sources. Our work builds upon these extractive principles but specifically adapts them for the latency and local-coherence constraints of modern RAG pipelines.

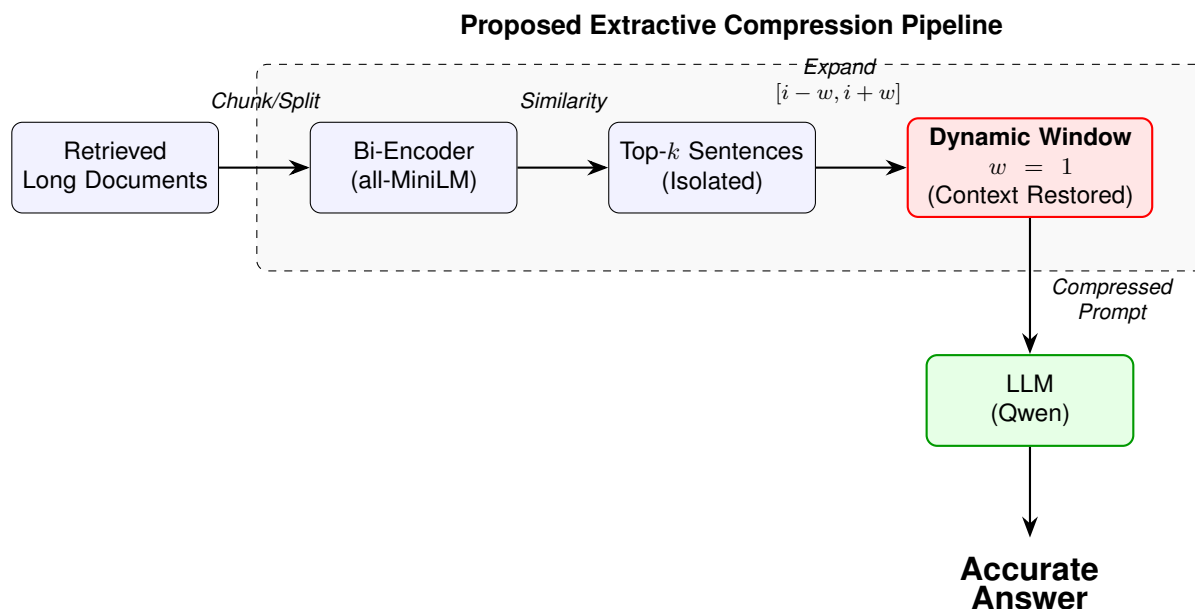


Fig. 1. Visual summary of our proposed method. Instead of feeding the entire document to the LLM (which leads to the "Lost in the Middle" effect) or extracting isolated sentences (which destroys semantic flow), our pipeline extracts highly relevant top- k sentences and deterministically expands them with a minimal local window ($w = 1$), achieving massive compression while preserving reasoning accuracy

2.4 Context Summarization and Compression Methods

To address long-context inefficiency, recent work has explored context compression for RAG. Abstractive methods such as *RECOMP-Abst*, *CompAct*, and *Refiner* generate condensed summaries of retrieved passages [21, 23, 11]. These methods can substantially reduce token counts, but their autoregressive compression stage introduces noticeable latency and may alter or omit critical evidence.

Extractive methods retain original text spans instead of rewriting them. *RECOMP-Extr* selects a small number of sentences from retrieved documents, while *LLMLingua* and *LongLLMLingua* perform token-level pruning to remove less informative content [21, 5, 6]. More recently, *EXIT* proposed context-aware extractive compression using a separate classifier that scores each sentence with access to broader document context [4].

Compared with these approaches, our method is simpler: it preserves sentence-level precision while restoring local coherence through a deterministic sentence window, without requiring a secondary compression model or query-time classifier.

2.5 Position of Our Method

Our approach is closest to extractive post-retrieval compression, but differs from prior work in two important ways. First, unlike token-level compression methods such as *LLMLingua*, which remove tokens and may disrupt syntactic structure, our method preserves complete local semantic units at the sentence level. Second, unlike learned or model-based compression approaches such as *RECOMP*, or classifier-based methods like *EXIT*, it does not introduce an additional trained compression stage or require supervision. This makes the method lightweight, fully zero-shot, retriever-agnostic, and easy to integrate into existing RAG pipelines.

3 Method

To effectively mitigate the computational overhead and cognitive degradation associated with long context windows, we propose a two-stage sentence-level compression pipeline equipped with dynamic local context restoration. Our approach strictly adheres to an “extract-then-abstract” paradigm, converting lengthy document corpora into highly dense, information-rich prompts.

Let $D = \{d_1, d_2, \dots, d_N\}$ be the set of retrieved documents relevant to a user query q . We first flatten the document set into a sequence of individual sentences, maintaining their original indices to preserve document-level origin. The flattened sentence corpus is defined as $S = \{s_1, s_2, \dots, s_M\}$.

3.1 Two-Stage Sentence Retrieval

Stage 1: Bi-Encoder Filtering. Given the computational expense of evaluating every sentence against the query using a cross-attention mechanism, we employ a fast bi-encoder, structurally similar to Sentence-BERT [15] and Dense Passage Retrieval (DPR) [8], to establish an initial candidate pool. The bi-encoder ϕ_{BE} independently maps the query q and each sentence s_i into a dense vector space $\mathbb{R}^d$. The similarity score is computed via cosine similarity:

$$\text{sim}_{BE}(q, s_i) = \frac{\phi_{BE}(q) \cdot \phi_{BE}(s_i)}{\|\phi_{BE}(q)\| \|\phi_{BE}(s_i)\|}. \quad (3.1)$$

We select the top- K sentences with the highest sim_{BE} scores to form an intermediate candidate set S_{cand} , where $K \ll M$. This stage ensures high recall while drastically reducing the search space.

Stage 2: Cross-Encoder Reranking. Bi-encoders are efficient, but they often miss fine-grained relations between the query and a candidate sentence because the two texts are encoded separately. To obtain a more accurate ranking, we pass each query–sentence pair from S_{cand} through a cross-encoder model ϕ_{CE} . In this setting, the query and the candidate sentence are processed together as a single input sequence,

which allows the attention layers to capture direct token-level interactions between them:

$$\text{score}_{CE}(q, s_i) = \phi_{CE}([\text{CLS}] \oplus q \oplus [\text{SEP}] \oplus s_i \oplus [\text{SEP}]). \quad (3.2)$$

We then rank all sentences in S_{cand} by $\text{score}_{CE}(q, s_i)$ and keep the top- k ones. We denote this final set as $S_{focal} \subset S_{cand}$.

3.2 Sentence Windowing for Context Restoration

Extracting isolated sentences inherently destroys narrative flow and disrupts coreference chains (e.g., pronouns losing their antecedents). To resolve this, we introduce a parameter w (window size), which dictates the number of adjacent sentences appended to each focal sentence. For every focal sentence $s_i \in S_{focal}$ located at index j within its original document d , we define its expanded window context $W(s_i)$ as:

$$W(s_i) = \bigcup_{t=j-w}^{j+w} s_t. \quad (3.3)$$

Crucially, the expansion is strictly bounded by document boundaries to prevent cross-document contamination. The final compressed context C_{comp} is constructed by taking the union of all expanded windows and sorting them by their original textual order:

$$C_{comp} = \text{Sort} \left(\bigcup_{s_i \in S_{focal}} W(s_i) \right). \quad (3.4)$$

This final string C_{comp} serves as the optimized prompt fed into the generative Large Language Model. By tuning k and w , we can dynamically control the trade-off between the compression ratio and semantic completeness.

4 Results

We evaluate our proposed extractive compression pipeline across four distinct QA datasets: TriviaQA and Natural Questions (single-hop), 2WikiMultihopQA and HotpotQA (multi-hop).

Our primary objective is to analyze the trade-off between the compression ratio (prompt length reduction) and the generative accuracy (measured by F1 and Exact Match scores).

4.1 Experimental Setup

To rigorously evaluate our proposed compression pipeline, we established a standardized experimental framework across all tasks.

Datasets and Sample Selection. We evaluated our approach on four widely adopted QA benchmarks, covering both single-document and multi-document retrieval scenarios:

- Natural Questions (NQ) [9]: A single-hop reading comprehension dataset comprising real anonymized queries issued to the Google search engine, where the answers are localized within a specific Wikipedia page.
- TriviaQA [7]: A collection of trivia questions authored by enthusiasts, paired with evidence documents from Wikipedia and the broader web. This benchmark tests robust single-hop reasoning.
- HotpotQA [22]: A multi-hop QA dataset where questions require the model to find and reason over multiple Wikipedia documents, explicitly evaluating the ability to chain intermediate facts.
- 2WikiMultihopQA [3]: Another multi-hop benchmark designed to require multi-step logical reasoning over Wikipedia articles, heavily relying on entity linking and coreference resolution.

Due to the computational cost of long-context generation and cross-encoder reranking, we performed evaluation on a fixed subset of 200 examples from the validation split of each benchmark.

The same subsets were consistently used across all compared methods to ensure fair and reproducible comparisons.

Models: For the retrieval stage, we utilized all-MiniLM-L6-v2 (a highly efficient 384-dimensional bi-encoder)[18] to generate initial

Table 1. Experimental results on TriviaQA (single-hop, 200 samples), showing the effect of sentence extraction (k) and local context window size (w) on QA performance and compression ratio

Top-K (k)	Window Size (w)	F1 Score	Exact Match	Compression Ratio
3	0	0.489	0.440	120.3x
3	1	0.548	0.495	44.7x
3	2	0.594	0.530	28.1x
5	0	0.531	0.465	73.1x
5	1	0.615	0.545	27.5x
5	2	0.630	0.555	17.5x
15	0	0.625	0.555	23.9x
15	1	0.640	0.570	9.6x
15	2	0.678	0.595	6.4x

candidate pools. For our ablation study on multi-hop tasks, we substituted this with the state-of-the-art BAAI/bge-base-en-v1.5 (768 dimensions)[19]. The generative component of our RAG pipeline was powered by Qwen3.5-9B[1], an open-weights Large Language Model known for its robust instruction-following and reasoning capabilities.

Generation Settings: The LLM was configured with a temperature of 0.1 and a top_p of 0.9 to encourage deterministic, fact-based answering. The maximum generation length was capped at 100 tokens, as the datasets expect concise, entity-centric answers.

Evaluation Metrics: We assessed generation quality using two standard QA metrics: *Exact Match (EM)*, which strictly measures if the predicted answer perfectly matches the ground truth, and *F1 Score*, which measures token-level overlap. Additionally, we tracked the *Compression Ratio (CR)*, calculated as the total token length of the original documents divided by the token length of the compressed prompt.

Hardware: All experiments were conducted on a single NVIDIA A100 GPU with 40 GB of memory. This setup was used for retrieval, cross-encoder reranking, and generative evaluation under a unified experimental configuration.

4.2 Single-hop QA Performance

Table 1 demonstrates the impact of hyperparameters on TriviaQA, while Table 2 presents the comprehensive results across all single-hop and multi-hop datasets.



Fig. 2. Trade-off between compression ratio and F1-score on the TriviaQA dataset. Adding a local context window ($w \geq 1$) consistently shifts the Pareto frontier upward, achieving higher F1 scores at comparable compression levels

In single-hop scenarios, the answer is typically localized within a specific sentence. Our findings demonstrate that aggressive extraction without local context ($k \leq 5, w = 0$) yields extreme compression ratios (up to 67.8x on NQ) but severely penalizes F1 scores due to semantic isolation and coreference loss.

However, the introduction of a minimal sentence window ($w = 1$) dynamically restores local coherence. As visualized in Figure 2, expanding the extracted focal sentences consistently shifts the Pareto frontier upward across all compression levels on the TriviaQA dataset. For example, on the NQ dataset, using $w = 1$ with $k = 5$ sentences still yields a 25x compression ratio but increases the F1 score from 0.437 to 0.464.

TriviaQA shows the same trend: adding a one-sentence window at $k = 5$ raises the F1 score by 15% compared to the zero-window baseline. This confirms that in single-hop RAG, up to 96% of the retrieved tokens are just noise and can be dropped without hurting the model's accuracy.

4.3 Multi-hop QA Performance

Multi-hop QA is inherently more difficult than single-hop tasks because the LLM needs to connect

facts from several documents instead of finding one specific text span. If we compress the text too aggressively, we might delete bridging facts, making it impossible for the model to deduce the final answer.

As the results show, extracting isolated sentences ($k = 15$ and $w = 0$) performs worse than passing the full text on both multi-hop datasets. This implies that keeping only the top-scoring sentences is not enough to capture all the supporting details. However, adding a small local window helps recover the missing context.

On 2WikiMultiHopQA, changing the setup from $k = 15, w = 0$ to $k = 15, w = 1$ pushes the F1 score from 0.519 to 0.539, and Exact Match from 0.440 to 0.455. On HotpotQA, the same change improves F1 from 0.608 to 0.668 and Exact Match from 0.500 to 0.550.

These results suggest that local context restoration is especially important for multi-hop reasoning, where adjacent sentences often contain the entity links, coreference cues, and intermediate facts required to connect evidence across documents.

5 Discussion

5.1 Mitigating "Lost in the Middle"

A critical finding of our experiments is the failure of the Full Context baseline. When provided with uncompressed, lengthy documents, the generative LLM (Qwen3.5-9B) struggled to locate relevant facts, yielding a poor F1 score of 0.283 on the Natural Questions dataset. This is consistent with the Lost in the Middle phenomenon, where excessive context overwhelms the model's attention mechanism.

The exceptionally poor performance and high latency of the full-context baseline can be partially attributed to tokenization constraints, specifically zero-padding effects up to a fixed maximum context length of 8192 tokens. Due to the quadratic computational cost of the attention mechanism, processing thousands of padding tokens severely slows down inference and further degrades the model's ability to focus on relevant semantic signals, exacerbating the "Lost in the Middle" effect.

Table 2. Comprehensive performance comparison across four distinct QA benchmarks (200 samples each). In single-hop tasks (top), our windowed extraction pipeline substantially outperforms the uncompressed baseline, mitigating the “Lost in the Middle” phenomenon while reducing token consumption by up to 27.5x. In multi-hop tasks (bottom), our method provides an effective trade-off between compression and answer quality, recovering significant F1 points previously lost to extreme isolated extraction ($w = 0$)

Dataset (Task Type)	Method Config	Top-K (k)	Window (w)	F1 Score	Exact Match	Comp. Ratio (C_R)
TriviaQA (Single-hop)	Baseline (Full Context)	All	-	0.262	0.235	1.0x
	Ours (Extreme Comp.)	5	0	0.531	0.465	73.1x
	Ours (Windowed)	5	1	0.615	0.545	27.5x
Nat. Questions (Single-hop)	Baseline (Full Context)	All	-	0.283	0.170	1.0x
	Ours (Extreme Comp.)	5	0	0.437	0.315	67.8x
	Ours (Windowed)	5	1	0.464	0.335	25.0x
2WikiMultihop (Multi-hop)	Baseline (Full Context)	All	-	0.576	0.485	1.0x
	Ours (Extreme Comp.)	15	0	0.519	0.440	1.7x
	Ours (Windowed)	15	1	0.539	0.455	1.2x
HotpotQA (Multi-hop)	Baseline (Full Context)	All	-	0.732	0.605	1.0x
	Ours (Extreme Comp.)	15	0	0.608	0.500	2.7x
	Ours (Windowed)	15	1	0.668	0.550	1.7x

Conversely, our compressed prompt ($k = 5, w = 1$) not only reduced the input token length by a factor of 25 but also *significantly increased* downstream accuracy, achieving an F1 score of 0.464 (a 64% relative improvement over the baseline). This demonstrates that our cross-encoder and sentence windowing pipeline does not merely act as a cost-saving compression tool, but serves as an effective semantic noise filter that improves answer generation in long-context RAG settings.

5.2 Limitations

This work has several limitations. First, all experiments were conducted on fixed subsets of 200 examples per dataset rather than the full evaluation splits. Although the same subset was used for all compared methods to ensure fairness, the reported results may not fully capture the variance of model behavior on the complete benchmarks. Second, our evaluation focuses on a zero-shot sentence-level compression pipeline under a single generative setup.

Different generators, retrieval backbones, or prompting strategies may lead to different compression–performance trade-offs. Third, while the proposed method performs strongly on single-hop QA and provides useful trade-offs

on multi-hop QA, it does not consistently outperform the full-context baseline on all multi-hop benchmarks. This suggests that aggressive compression remains challenging in settings where reasoning depends on connecting multiple pieces of evidence across documents.

Finally, we evaluate answer quality mainly through Exact Match and F1. These metrics are standard for QA, but they do not fully capture aspects such as reasoning faithfulness, calibration, or latency-quality trade-offs at deployment time.

5.3 Comparison with Existing Compression Methods

Compared to existing context compression methods, our approach occupies a distinct position in the design space. Token-level compression methods such as LLMingua achieve high compression ratios by aggressively removing tokens, but often at the cost of grammatical coherence and interpretability. In contrast, our method preserves full sentences, ensuring that the compressed context remains structurally intact.

Similarly, learned compression frameworks such as RECOMP rely on additional models to summarize or filter retrieved content, which introduces extra computational overhead and

requires training data. Our approach, being fully zero-shot and deterministic, avoids these limitations while remaining competitive in terms of compression and downstream QA performance.

These differences suggest that our method is particularly suitable for real-time or resource-constrained RAG systems, where simplicity, efficiency, and robustness are critical.

While the theoretical and architectural advantages of our zero-shot approach are clear, conducting a rigorous quantitative benchmarking against these state-of-the-art compression baselines (such as LLMLingua, RECOMP, and EXIT) on identical data splits remains a priority for our future work.

6 Conclusion

In this paper, we tackled the computational bottleneck of long-context Retrieval-Augmented Generation (RAG). We introduced a lightweight, zero-shot sentence compression pipeline that restores local context dynamically. By chaining a fast bi-encoder, a precise cross-encoder, and a deterministic sentence window, our approach filters out irrelevant text while keeping the logical flow intact.

Our evaluations across four QA benchmarks highlight a clear trade-off between prompt size and reasoning accuracy. On single-hop datasets, the pipeline aggressively cuts down prompt length and actually outperforms the full-context baseline, proving that most retrieved text is just noise. For multi-hop reasoning, the uncompressed full documents still achieve the highest absolute accuracy. However, our method provides a highly efficient alternative: by adding just a single-sentence window ($w = 1$), we recover the context lost by isolated extraction and massively reduce token consumption without needing to process thousands of tokens.

Ultimately, sentence-level extraction combined with local windowing is a practical way to run RAG under strict memory and latency budgets. The pipeline requires no additional training, works with any retrieval backbone, and easily plugs into existing systems.

Future Work. Moving forward, we plan to benchmark our pipeline against other established compression techniques, such as LLMLingua, RECOMP, and EXIT. We also aim to scale our evaluation to full dataset splits to verify statistical significance across a broader range of generative models.

Acknowledgments

This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP23489782).

References

1. Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., others (2023). Qwen technical report. arXiv preprint arXiv:2309.16609.
2. Erkan, G., Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, Vol. 22, pp. 457–479.
3. Ho, X., Nguyen, A.-K. D., Sugiyama, S., Aizawa, A. (2020). Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pp. 6609–6625.
4. Hwang, T., others (2024). EXIT: Context-aware extractive compression for enhancing retrieval-augmented generation. arXiv preprint arXiv:2412.12559.
5. Jiang, H., Wu, Q., Lin, C.-Y., Yang, Y., Qiu, L. (2023). LLMLingua: Compressing prompts for accelerated inference of large language models. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

6. Jiang, H., Wu, Q., Luo, X., Li, D., Lin, C.-Y., Yang, Y., Qiu, L. (2024). Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL).
7. Joshi, M., Choi, E., Weld, D. S., Zettlemoyer, L. (2017). Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1601–1611.
8. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6769–6781.
9. Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., others (2019). Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics, Vol. 7, pp. 453–466.
10. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., others (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems (NeurIPS), Vol. 33, pp. 9459–9474.
11. Li, Z., Hu, X., Liu, A., Zheng, K., Huang, S., Xiong, H. (2024). Refiner: Restructure retrieval content efficiently to advance question-answering capabilities. arXiv preprint arXiv:2406.11357.
12. Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., Liang, P. (2024). Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics, Vol. 12, pp. 157–173.
13. Liu, Y., Lapata, M. (2019). Text summarization with pretrained encoders. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3730–3740.
14. Mihalcea, R., Tarau, P. (2004). Textrank: Bringing order into text. Proceedings of the 2004 conference on empirical methods in natural language processing (EMNLP), pp. 404–411.
15. Reimers, N., Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3982–3992.
16. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., others (2023). Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.
17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS), Vol. 30.
18. Wang, W., Wei, F., Dong, L., Bao, H., Yang, M., Zhou, M. (2020). Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. Advances in Neural Information Processing Systems (NeurIPS), Vol. 33, pp. 5776–5788.
19. Xiao, S., Liu, Z., Zhang, P., Mu, N. (2023). C-pack: Packaged resources to advance general chinese embedding. arXiv preprint arXiv:2309.07597.
20. Xiao, W., Izacard, G., Lewis, M., Carenini, G. (2022). Primera: Pyramid-based masked sentence pre-training for multi-document summarization. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 5245–5263.
21. Xu, F., Shi, W., Choi, E. (2024). RECOMP: Improving retrieval-augmented LMs with

compression and selective augmentation. The Twelfth International Conference on Learning Representations (ICLR).

22. **Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., Manning, C. D. (2018).** Hotpotqa: A dataset for diverse, explainable multi-hop question answering. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2369–2380.
23. **Yoon, C., Lee, T., Hwang, H., Jeong, M., Kang, J. (2024).** Compact: Compressing retrieved documents actively for question answering. Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP).
24. **Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., Huang, X. (2020).** Extractive summarization as text matching. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 6197–6208.

Article received on 05/04/2026; accepted on 29/07/2026.

**Corresponding author is Alexander Gelbukh.*