Computer Life
ISSN: 1819-4818 | Vol. 15, No. 2, 2026

Query-Aware Context Compression for Efficient Long-Context Language Model Inference

Maya Laurent [1, *], Aaron McLeod [2]

[1] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada
[2] Department of Computer Science, University of Calgary, Calgary, Canada

[*] **Corresponding author**: maya.laurent0608@gmail.com

Abstract: Query-aware context compression can reduce the amount of source text forwarded to a language model, but keyword ranking may discard evidence connected through intermediate entities. We present QAB-R, a training-free sentence selector combining BM25, query-identifier-constrained anchor selection, canonicalized entity matching, and at most two bridge-linked candidates under a fixed sentence budget. We first evaluate a single-bridge baseline on 300 synthetic single- and two-hop queries. We then execute 300 additional held-out stress cases covering paraphrases, competing entity notes, identifier formatting variation, and combined changes; a separate 60-case test extends source contexts to 58,368 whitespace-delimited words. At a 10% sentence budget, QAB-R preserved complete two-hop support in all 60 cases of each stress condition, whereas single-bridge retention fell to 0% under competing notes and format changes. For 1,024 fact pairs, the median QAB-R CPU preprocessing time was 66.3 ms, and the selected text contained 5,741 of 58,368 input words. The study reports evidence completeness, retained input length, and CPU preprocessing cost; downstream reader accuracy and GPU inference speed are distinct system-level outcomes.

Keywords: Context compression; Long-context inference; Question answering; Retrieval; Evidence retention; BM25.

1. Introduction

Long-context language models have expanded the range of documents and retrieval collections that can be supplied in a single prompt [1]–[6]. Nevertheless, attention, memory allocation, and serving overhead grow with sequence length, motivating fast attention kernels and better inference systems [7]–[9]. Sparse or recurrent architectures address related scaling challenges [10]–[14]. In document-grounded question answering, a separate difficulty remains: the useful evidence often occupies only a small fraction of the available context, and its location need not be near the beginning or end [15]–[20].

Input-side compression offers a complementary solution: remove sentences that are unlikely to assist a specific query before passing the shortened context to a reader model. The essential requirement is not merely to retain individually relevant snippets, but to preserve linked evidence when the answer requires more than one fact. Standard lexical ranking can find an entity mentioned directly in the question yet miss a second sentence that refers to an intermediate entity.

We study a compact, deterministic selector. Starting from a single-hop entity-bridge baseline, we add (i) an exact query-identifier anchor constraint, (ii) identifier normalization across punctuation and spaces, and (iii) a cap of two linked candidates. The principal contributions are a fully specified query-aware selection rule, two complementary synthetic evaluations including adversarial lexical distractors, and measurements of support completeness, retained length, and CPU preprocessing cost up to approximately 58,000 source words. The procedure operates before the reader, without modifying model parameters or attention kernels.

2. Related work

2.1. Long-context evaluation and evidence localization

LongBench, RULER, LooGLE, ZeroSCROLLS, and BABILong evaluate different aspects of long-context understanding and reasoning [15], [16], [18]–[20]. Lost in the Middle identifies sensitivity to the position of relevant information [17]. Extractive QA resources such as SQuAD, HotpotQA, MuSiQue, and QASPER demonstrate the importance of answer spans and, for multi-hop tasks, explicit supporting facts [21]–[24]. Our synthetic benchmark is narrower than these natural-data benchmarks but isolates evidence-link retention under controlled distractor density.

2.2. Retrieval, textual compression, and cache methods

BM25 and maximal marginal relevance (MMR) provide established relevance and novelty criteria [25], [26]. Dense passage retrieval, retrieval-augmented generation, REALM, fusion-in-decoder, in-context retrieval, and REPLUG illustrate how external evidence can condition language-model behavior [27]–[32]. RECOMP, Selective Context, LLMLingua, LongLLMLingua, and LLMLingua-2 explicitly reduce retrieved or prompt text [33]–[37]. Gist tokens, AutoCompressors, and in-context autoencoders pursue learned compact representations [38]–[40].

StreamingLLM, H2O, Scissorhands, SnapKV, PyramidKV, Quest, and MInference instead optimize attention computation or key-value storage in the model runtime [41]–[47]. Input-side bridge compression is distinct: it manipulates readable evidence sentences before an arbitrary reader receives the prompt. No changes to attention kernels, cache layouts, or model weights are required.

3. Query-anchored bridge compression

3.1. Problem formulation

Let D = (s1, …, sN) be a set of source sentences in their original document order, q a natural-language question, and B a sentence budget. The compressor returns an ordered subset C ⊆ D with |C| ≤ B. Each benchmark question has a gold support set E(q) containing one or two sentence indices. The selection objective is to maximize evidence coverage subject to the budget. Selecting sentences does not rewrite their content, which keeps factual spans intact.

$$C^*(q) = \arg\max \quad |C \cap E(q)| \,/\, |E(q)| \qquad \text{s.t. } |C| \le B. \tag{1}$$

Gold support is used strictly for evaluation; E(q) is not available to any compressor. Since the practical problem has no gold annotations at selection time, we approximate the objective with lexical relevance and query-dependent entity linking.

3.2. Anchor scoring and bridge selection

All sentences receive a BM25 score against the query, with k1 = 1.2 and b = 0.75. The initial comparator uses the highest-ranked sentence as anchor a. For the revised selector, explicit identifiers in the question constrain anchor selection before falling back to ordinary BM25. Let G(s) be the set of normalized three-capital-letter/five-digit identifiers in s. New bridge identifiers are G(a) \ G(q).

$$L(a,q) = \{si : G(si) \cap [G(a) \setminus G(q)] \neq \emptyset\}. \tag{2}$$

For each linked candidate, the method counts question terms that were not already covered by the anchor. Candidates are sorted by this residual coverage count, with normalized BM25 as a tie adjustment. The original Bridge-1 comparator adds one linked sentence if B ≥ 2; QAB-R adds up to two if B ≥ 3. Remaining slots follow ordinary BM25 order. Selected sentences are finally restored to source order. This one-step strategy bounds extra work and supports nonadjacent evidence chains without invoking a language model.

$$h(si) = |[T(q) \setminus T(a)] \cap T(si)| + 0.1\, BM25(si,q)/\max_j BM25(sj,q). \tag{3}$$

Here T(·) denotes lowercase lexical tokens, and the set operations in (3) explicitly group query terms not already covered by the anchor. Gold-support indices and known answer values are not accessed by either algorithm at selection time.

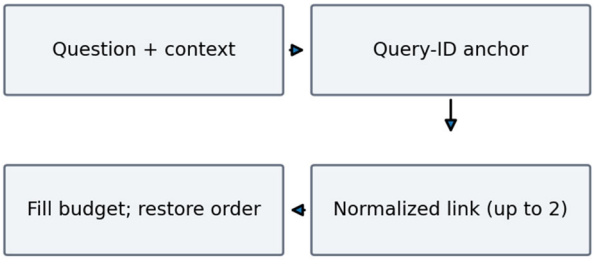


Fig. 1 QAB-R pipeline: query-ID anchor, normalized links, then budgeted extraction.

3.3. Identifier-anchored two-candidate expansion

QAB-R first converts strings such as FAC-12345, FAC_12345, and FAC 12345 to the common identifier FAC-12345. If the question contains explicit identifiers, the anchor is the highest-BM25 sentence containing at least one of them; if none exists, QAB-R uses the global BM25 leader. The anchor is followed by up to two distinct sentences sharing an identifier introduced by the anchor but absent from the query. Linked candidates follow the residual-query-term score in (3). All other budget positions are filled by BM25, with the retained sentences restored to their source positions. The two-candidate cap supports cases where an entity profile and a non-answer note mention the same identifier. No color lexicon, answer-type rule, gold index, or synthetic record label appears in this selection routine.

The original single-bridge variant is retained as an experimental ablation. QAB-R differs in three testable choices: explicit query-ID anchoring, canonicalized identifier matching, and two rather than one linked candidate. Each extra step examines only the input text and query and does not require model training. If a question does not contain a recognizable identifier, QAB-R falls back to the BM25 anchor; this study evaluates the identifier-bearing operating case.

3.4. Execution procedure and worked example

The selection procedure follows a fixed decision path. Let P be a collection of sentence strings. Tokenize P and q, calculate document frequencies for the current context, and rank sentences with BM25. QAB-R uses the highest-ranked query-ID-matching sentence as the anchor where available, otherwise the BM25 leader. If B exceeds one, extract identifiers from the anchor that are absent from q and scan the remaining sentences for a shared identifier. Among the linked candidates, choose up to two with the most additional query terms, using relative BM25 as a tie adjustment. Fill any available budget positions with the highest-ranked as-yet-unselected sentences and place the selected excerpts back into their original source order. The query itself is not compressed or counted in B.

For example, the generated case two_hop_64_0000 asks: “What inspection badge color is listed for the facility assigned to order ORD-97832?” The archive sentence links ORD-97832 to FAC-32275; a separated profile sentence reports the inspection badge color of FAC-32275 as violet. The name FAC-32275 is the bridge between the query-relevant archive sentence and the answer-bearing profile. The two evidence sentences occupy positions 87 and 46 in the shuffled context, respectively; direct adjacency is therefore unavailable. The method retrieves the anchor from the question identifier and adds the linked profile because it covers previously unmet badge-related terms.

Unlike an unrestricted graph walk, Bridge-1 keeps one linked sentence and QAB-R keeps at most two. This cap prevents repeated entity mentions from consuming the entire compression budget. When the query is a one-hop lookup, retaining the anchor alone is sufficient for the benchmark objective, and extra budget entries do not change the measured evidence completeness.

3.5. Runtime considerations

The baseline BM25 scoring step is O(N · L) for a bounded question and sentence length L. Canonicalizing identifiers, choosing a query-matching anchor, and scanning for up to two links add O(N · L), while ranking requires O(N log N). The implementation uses ordinary Python regular expressions, dictionaries and sets; it requires no embeddings or GPU inference. The optional MMR comparator restricts reranking to a top-ranked candidate window, making its measured runtime practical for the current CPU evaluation.

4. Experimental design

4.1. Deterministic benchmark construction

A controlled benchmark is generated with random seed 20261106. Each context contains 32, 64, or 128 independent order–facility pairs. For each pair, an assignment sentence maps a unique ORD-style order identifier to a unique FAC-style facility identifier; a profile sentence maps the facility to an inspection-badge color. Sentences use fixed natural-language templates and are randomly shuffled independently, so the two linked facts are not adjacent. Every context contains 2N sentences. Questions either ask for the directly assigned facility (one support sentence) or ask for the badge color of that facility (two support sentences). There are 50 examples per size–task combination, yielding 300 questions overall.

This is a controlled synthetic dataset, not a subset of SQuAD, HotpotQA, or LongBench. Its purpose is to test whether evidence links survive sentence selection; its controlled identifiers and repeated wording make the target relation unambiguous. The generated source texts, questions, and gold sentence indices are stored in the accompanying experiment package.

Table I. Benchmark and measurement settings

Setting	Value
Context sizes	32 / 64 / 128 fact pairs
Sentence count	64 / 128 / 256 sentences
Task types	single hop; two hop
Instances	50 per size and task; 300 total
Sentence budgets	5% / 10% / 20%
Baselines	prefix; random; BM25; BM25+MMR
Primary measure	complete evidence retained (%)
Other measures	word count; median CPU selector ms

4.2. Reproducibility and evaluation controls

Every experiment instance is derived from a single deterministic generator. The generator independently samples order and facility identifiers without replacement within a context, assigns one color label per facility, constructs the two sentence templates, and shuffles the resulting 2N sentences. It records the hidden target index, answer string, and exact gold support positions. The evaluation script accesses the support indices only after a method returns its selected sentence set. No method reads target indices, answer strings, or hidden record-pair labels. This separation is useful because a synthetic generator could otherwise inadvertently reveal the answer through evaluation metadata.

The dataset and numerical files are included with the working package. The aggregate table is computed directly from a case-level result table rather than manually transcribed scores. Fifty independent cases are evaluated for each of the six combinations of context size and question type. Each case is then tested at three budgets with all five methods, producing 4,500 measured method–case–budget rows. The question is supplied identically to all baselines. Sentences are ranked as units, so their full wording is either retained or dropped; no generation or factual substitution is introduced by compression.

4.3. Baselines and metrics

We compare prefix selection, seeded random selection, BM25 ranking, and BM25 with a 0.72 relevance/0.28 Jaccard-similarity MMR penalty over a capped top-ranked candidate pool. All strategies use the same sentence count $B = \max(1, \mathrm{round}(2N \cdot r))$, where r is the stated budget. Random selection uses a case-dependent deterministic seed. The proposed method and BM25 share the same lexical scoring implementation. Complete-evidence recall equals 1 if every gold support sentence is included, and 0 otherwise; mean support recall allows partial evidence. Compression is also reported in whitespace-delimited words, which are not represented as proprietary LLM tokenizer counts.

Latency is measured as wall-clock preprocessing time in Python on CPU, including BM25 preparation for lexical methods and the incremental selection step. Values are medians over 50 cases per cell; operating-system and Python scheduling effects can influence millisecond timings. Paired bootstrap confidence intervals use 3,000 resamples for the 64-pair, two-hop, 10% setting. No language-model forward pass or answer generation is part of this evaluation.

4.4. Why full support matters

The complete-evidence metric is intentionally stricter than simple hit rate. For a two-hop question, retrieving only the assignment produces a plausible query match but does not reveal the requested color. Retrieving only the profile supplies a color without indicating that the profile belongs to the queried order. Both pieces are needed to make the answer grounded in the input. In our evaluator, partial support receives an evidence-recall value of 0.5 but a complete-evidence value of zero. For a single-hop question, exactly one assignment sentence is required, so the same metric reduces to direct evidence-hit accuracy. This distinction makes it possible to diagnose a weak second-hop retriever separately from poor first-hop retrieval.

The sentence budget is applied in units of selected sentences, whereas source compression is additionally reported as a word ratio. For example, with 64 pairs, a 10% sentence budget selects 13 of the 128 sentences. The source averages 3,648 whitespace-delimited words and bridge selection forwards 365 words. Thus the reported input reduction is 89.995% by this word-count convention. This quantity is not a model-token compression ratio and is not converted into an unmeasured GPU memory or inference-time saving.

4.5. Robustness and long-context protocols

A second, disjoint generator uses seed 20261107 to create 60 two-hop cases for each of five conditions at 64 fact pairs: canonical wording; sentence-template paraphrases; short non-answer notes about the same facility; inconsistent facility-ID separators; and a combination of these perturbations. In the note conditions each facility receives one additional lexical distractor, producing 192 rather than 128 sentences. At budgets of 5%, 10%, and 20%, six lexical methods are evaluated on the same shuffled context per case. The alternative QAB-R implementation differs only in its selection logic; all methods share BM25 preparation and the support-set evaluator.

A third controlled experiment independently generates 20 two-hop cases each for 256, 512, and 1,024 fact pairs (seed 20261108), at a 10% sentence budget. These large contexts contain respectively 14,592, 29,184, and 58,368 whitespace-delimited source words. The run compares BM25, the original single-bridge selector, and QAB-R. Across the three

experiments there are 660 synthetic cases and 10,080 method–budget observations; case-level files and executable scripts accompany the submission package.

5. Experimental results

5.1. Evidence completeness at the primary budget

Table II. Two-hop results at 64 fact pairs and 10% budget

Method	Complete %	Support %	Median ms
Prefix	2.0	9.0	0.00
Random	0.0	7.0	0.02
BM25	16.0	58.0	3.32
BM25+MMR	18.0	59.0	6.17
Bridge (ours)	100.0	100.0	3.79

At 64 pairs and a 10% sentence budget, bridge expansion retains both gold sentences for 100% of the two-hop questions. BM25 and candidate-truncated MMR retain complete evidence in 16% and 18%, respectively (Table II). In this setting, the mean source is 3648 words, compared with 365 retained words (10.01% of the source). The paired bootstrap difference over BM25 is 84 percentage points with a 95% interval of [74, 94] points.

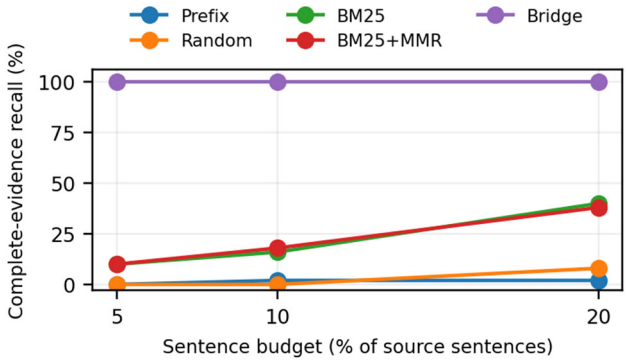


Fig. 2 Complete two-hop evidence retention across sentence budgets (64 fact pairs, 50 cases).

5.2. Compression budget, length, and task sensitivity

Figure 2 shows that increasing BM25's budget from 5% to 20% raises complete two-hop evidence recall from 10% to 40%, while the bridge method remains at 100% at all three tested budgets. The bridge rule uses the query-matched order sentence to retrieve the facility profile, rather than requiring the profile to rank highly on direct question overlap. The strong controlled result follows from the benchmark's unique identifiers and single-bridge construction.

Table III. 10% budget across context sizes

Fact pairs	1-hop BM25 %	2-hop BM25 %	2-hop Bridge %	Bridge words %
32	100	8	100	9.27
64	100	16	100	10.01
128	100	12	100	9.99

For single-hop questions, BM25 and bridge selection each retain the required direct assignment sentence in all 50 cases at every tested context size. For two-hop questions, BM25 complete-evidence recall remains between 8% and 16%, whereas the bridge method preserves both facts in every case (Table III). Thus, the benefit is concentrated in retaining the second evidence sentence; lexical retrieval alone is already sufficient for the direct-lookup subset.

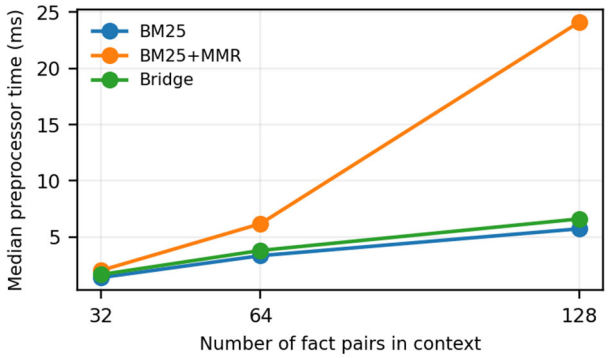


Fig. 3 Measured CPU preprocessing time by context size (two-hop, 10% budget).

5.3. Preprocessing overhead

At 64 pairs and 10% budget, median selection time is 3.32 ms for BM25, 3.79 ms for bridge expansion, and 6.17 ms for candidate-truncated MMR. Figure 3 shows measured behavior across 32, 64, and 128 pairs. Adding one bridge expansion therefore changes only the input-preprocessing stage, while reducing the number of words forwarded to a hypothetical reader. The latter may lower its input work, but the experiment makes no quantitative claim about full-model latency.

5.4. Sensitivity to distractor density

A useful stress variable is the number of distractor pairs because nearly every facility-profile sentence shares the lexical expressions "inspection badge" and "color" with a two-hop question. The exact order identifier occurs only in the target assignment sentence. As a consequence, direct BM25 can reliably identify the first hop, but after selecting it the many profile sentences are close competitors for the remaining budget positions. Increasing the number of pairs from 32 to 128 raises the number of such distractors from 31 to 127, while maintaining the same two-sentence gold target. The second-hop completeness of BM25 at 10% budget is 8%, 16%, and 12% for 32, 64, and 128 pairs; the one-step bridge method is 100% in each cell. These are empirical values from separate seeded cases, not theoretical guarantees for arbitrary natural text.

Preprocessing time rises with context length because lexical scoring and identifier scans touch each sentence. The candidate-truncated MMR baseline performs additional pairwise novelty comparisons, whereas the two bridge designs use direct entity overlap. CPU timings describe the measured scripts and execution environment, not model-token latency.

5.5. Perturbation robustness and ablation

As Table IV shows, the original method is highly effective when an anchor sentence introduces exactly one unambiguous intermediate entity. However, competing non-answer notes cause the single added bridge to select the note in place of the answer-bearing profile. Identifier separator changes defeat the original exact-pattern matcher.

Normalization plus two linked candidates recovers 73.3%, 85.0%, and 78.3% complete support on competing-note, format-shift, and combined conditions, respectively. Finally, query-ID anchor preference resolves the remaining first-hop lexical ambiguity: QAB-R retains complete support in all 60 cases of each condition at the 10% sentence budget.

Table IV. Complete two-hop support (%) at 10% budget: 60 cases per condition

Stress condition	BM25	Bridge-1	Norm.+2	QAB-R
Canonical	11.7	100.0	100.0	100.0
Paraphrased text	0.0	91.7	91.7	100.0
Competing notes	0.0	0.0	73.3	100.0
ID separators	0.0	0.0	85.0	100.0
Combined changes	0.0	0.0	78.3	100.0

On the combined condition QAB-R median CPU preparation was 3.91 ms, versus 3.36 ms for Bridge-1 and 2.86 ms for BM25. The combined contexts retain 8.41% of source words under a 10% sentence count budget. The additional preprocessing performs normalization and candidate matching, rather than executing a neural reader.

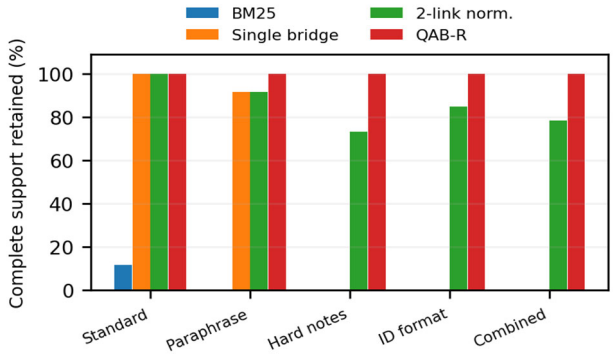


Fig. 4 Paired-method retention on five generated stress conditions at 10% budget.

5.6. Scaling to longer generated contexts

At 1,024 fact pairs (2,048 sentences), the source length is 58,368 whitespace-delimited words, with 5,741 forwarded by QAB-R (Table V). The measured median CPU selector time is 66.3 ms, compared with 49.6 ms for BM25. Complete evidence retention is 100% in these 20 structured cases, while BM25 retains 15%. This confirms the selector can process substantially longer generated inputs on CPU without changing the reader interface.

Table V. 10% budget at larger synthetic context lengths (20 cases per size)

Pairs	Input words	Kept words	BM25 %	QAB %	CPU ms
256	14,592	1,429	10	100	14.6
512	29,184	2,857	20	100	32.2
1024	58,368	5,741	15	100	66.3

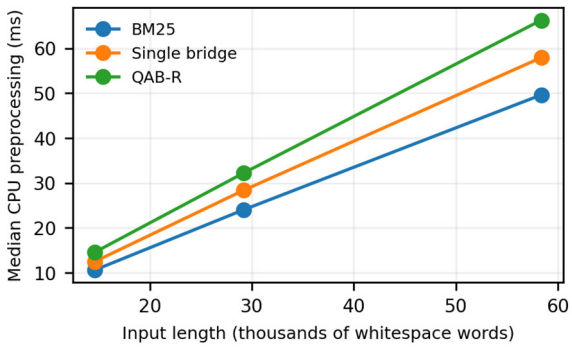


Fig. 5 Measured CPU preprocessing against source word count, 20 cases per size.

6. Discussion

Evidence retention is an interpretable intermediate objective for compression-based retrieval and question answering. When a query names an order but requests an attribute of an associated facility, a purely lexical ranker is likely to retrieve the named order while treating many facility profiles as interchangeable. Shared identifiers reveal the missing relation at low computational cost. This observation is consistent with the broader emphasis on supporting facts in multi-hop QA [22], [23] and question conditioning in prompt and cache compression [36], [46-55].

The study specifies an identifier-bearing operating domain and measures actual selection behavior on three fully generated collections. The new stress cases establish that paraphrases, lexical decoy notes, and separator mismatches are meaningful failure modes for the original selector. QAB-R addresses those perturbations using syntax-level identifier canonicalization and query-ID anchoring. These tests do not cover ambiguous real-world aliases, mistaken identifiers, more than two linked target facts, or answer generation. Evaluation with a fixed language model and public annotated natural-document benchmarks such as HotpotQA or QASPER would establish downstream answer fidelity and actual serving-time gains [22], [24].

6.1. Deployment interpretation

The approach is suitable as a text-only preprocessing component before a frozen or API-accessed reader because it returns an ordinary subset of source sentences. Its input may be a retrieved passage set, an assembled report, or a document collection with stable identifiers; the full context is scanned locally, then the compact text can be used in a standard question-answering prompt. The method does not depend on gradient access to the reader. When sentence count and input length are smaller, one expects fewer input positions to be processed by a downstream model, but the realized latency effect depends on serving hardware, attention implementation, batch size, and prefill/decode balance [7]–[9], [47]. Consequently, we report the measurable preprocessing component independently from potential system-level benefits.

Bridging is particularly transparent in operational documents with order numbers, document IDs, case identifiers, and catalog references. Other source types can replace the regular expression with an entity-normalization step without changing the basic selection logic. For contexts where identifiers are missing, the same conceptual expansion can be driven by exact string aliases or learned coreference,

with the additional modeling cost evaluated separately. The present implementation prioritizes bounded complexity and auditable sentence-level evidence, while keeping the reader interface unchanged [56-60].

7. Conclusion

A query-aware, training-free context compressor was implemented and evaluated using a BM25 anchor and normalized two-candidate entity-bridge expansion. The combined suite contains 660 synthetically generated QA cases and 10,080 recorded method–budget observations. QAB-R preserved complete two-hop evidence in all five 60-case stress conditions at 10% sentence budget and scaled to 58,368-word synthetic prompts with a median 66.3-ms CPU preprocessing time. The demonstrated contribution is controlled evidence-preserving extraction with bounded local processing overhead; any language-model answer-quality or end-to-end inference improvement remains an independent measurement.

References

[1] Fang, Q. (2024). Bias-aware large language models for equitable educational feedback and assessment. Trends in Social Sciences and Humanities Research, 2(1), 41-50.

[2] Teng, D., Rhee, M., Qin, Y., Zi, B., & Liu, W. (2026). SW-SpeedDLM: Sliding window speculative decoding for diffusion language models under long context constraints. Mathematics, 14(12), 2137.

[3] Wang, Y., Bian, M., & Lin, Y. (2022). An explainable temporal graph neural network for disruption prediction and risk propagation in multi-tier supply chains. Journal of Computer Science and Electrical Engineering, 4(1), 17-25.

[4] Rhee, M., Zou, J., & Qin, Y. (2024). CLARITY: A closed loop framework for duplicate aware failure prediction and budgeted automated remediation in cloud infrastructure. AI and Data Science Journal, 1(1), 76-85.

[5] Wang, G., Li, P., Feng, E., Shen, A., & Rhee, M. (2026). A dual-process memory-augmented large language model agent for explainable sequential recommendation. IEEE Access.

[6] Ping, W., & Fan, H. (2023). Beyond isolated ratios: Accrual-revenue-cash flow indicators and interaction-aware learning for detecting financial reporting irregularities in U.S. public companies. World Journal of Economics and Business Research, 1(1), 46-55.

[7] Zhou, F. (2026). Moment-constrained side-force allocation for wind-assisted merchant ships: Hydrodynamic performance of retractable anti-leeway fins and flap rudders. Innovation and Technology Studies, 3(1), 27-41.

[8] Mo, T., Zhang, C., Zou, J., Guo, Z., & Rhee, M. (2026). Self-evolving AI agents with dual memory for automated software testing and bug localization. IEEE Access.

[9] Fang, Q. (2024). TRUE-EWS: Trustworthy early warning with equalized conformal guarantees for equitable student support in U.S. education. Educational Research and Human Development, 1(2), 44-54.

[10] Liang, Y., & Jiao, Y. (2024). Real-time financial fraud detection using a feature-light temporal dynamic graph neural network with contrastive learning. Journal of Trends in Finance and Economics, 1(1), 76-84.

[11] Teng, D. (2025). RSRBench: Reproducibility and performance benchmarking for regional seismic risk workflows. Academic Journal of Architecture and Civil Engineering, 3(1), 40-52.

[12] Guo, Z., Chen, T., & Shang, W. (2022). Class-conditional intermediate-domain adversarial adaptation for cross-device acoustic scene classification. Journal of Computer Science and Electrical Engineering, 4(1), 26-35.

[13] Ping, W., & Fan, H. (2024). Disclosure quality and valuation risk in level 3 investments: A disclosure-implied risk framework for U.S. business development companies. Social Science and Management, 1(1), 78-87.

[14] Guo, Z., Fang, Q., Yang, J. S., & Rhee, M. (2026). Closed-loop audio neuroregulation for ADHD: Graph-based attention decoding and rhythm entrainment. IEEE Access.

[15] Yue, X., & Zhu, R. (2024). An explainable hybrid machine learning framework for cash-flow-conditioned multi-horizon liquidity risk early warning: An SME-oriented benchmark study. World Journal of Management Science, 2(1), 63-72.

[16] Cao, J., Zheng, W., Ge, Y., & Wang, J. (2025). DriftShield: Autonomous fraud detection via actor-critic reinforcement learning with dynamic feature reweighting. IEEE Open Journal of the Computer Society, 6, 1166-1177.

[17] Mai, N. T., Cao, W., & Fang, Q. (2025). A study on how LLMs (e.g., GPT-4, chatbots) are being integrated to support tutoring, essay feedback and content generation. Journal of Computing and Electronic Information Management, 18(3), 43-52.

[18] Shang, W., & Chen, T. (2024). PrefixKD-Edge: Real-time locomotion-intent prediction with prefix self-distillation on lightweight temporal networks. Innovation and Technology Studies, 1(1), 30-38.

[19] Qin, Y., Rhee, M., Teng, D., Zhang, C., & Zi, B. (2026). Neuro-symbolic constraint verification for LLM-driven internet finance transaction execution. Scientific Reports.

[20] Zhang, H. (2024). Graph-aware multi-task learning for early prediction of timing violations and routing congestion in digital IC physical design. World Journal of Information Technology, 2(1), 13-20.

[21] Zhang, F., & Yang, J. (2022). Accuracy-aware model compression for real-time visual perception under resource constraints. Journal of Computer Science and Electrical Engineering, 4(1), 45-52.

[22] Bian, M., Wang, Y., & Lin, Y. (2024). Explainable supplier risk prediction for critical telecommunications infrastructure procurement: An empirical XGBoost-SHAP framework. Social Science and Management, 1(1), 64-77.

[23] Fan, H., Jiao, Y., Wang, M., Chen, J., & Yang, S. (2026). Fraud learns too: Continual graph learning under strategic adversarial drift in dynamic networks. Scientific Reports.

[24] Shen, A., & Huang, S. (2022). A systems-aware framework for intelligent large-scale recommendation models. Journal of Computer Science and Electrical Engineering, 4(1), 36-44.

[25] Zhang, F., Shen, A., & Huang, S. (2026). TRIAD: Co-designing mixed-precision quantization, operator fusion, and on-chip memory reuse for real-time vision-language-action inference on automotive NPUs. Results in Engineering, 113673.

[26] Jiao, Y., & Liang, Y. (2024). An integrated process-mining and explainable machine-learning framework for bottleneck and internal-control risk detection in the multi-entity financial close. Social Science and Management, 1(3), 96-107.

[27] Mai, N. T., Fang, Q., & Cao, W. (2025). Measuring student trust and over-reliance on AI tutors: Implications for STEM learning outcomes. International Journal of Social Sciences and English Literature, 9(12), 11-17.

[28] Shen, A., & Liu, W. (2026). Budget-aware adaptive test-time compute with hierarchical self-verification for semantic ID generative recommendation. Computers.

[29] Chen, J., Han, X., Zhang, X., & Yang, Y. (2023). Exact concept-level explanations for early detection of financial misstatements. Eurasia Journal of Science and Technology, 5(2), 34-48.

[30] Fang, Q., & Liu, W. (2025). HARLA-ED: Resolving information asymmetry and enhancing algorithmic symmetry in intelligent educational assessment via hybrid reinforcement learning. Symmetry, 18(1), 58.

[31] Mo, T., Wang, G., Feng, E., & Li, P. (2023). Execution-feedback-guided context selection for large language model-based software testing and debugging. World Journal of Information and Knowledge Management, 1(1), 55-64.

[32] Chen, J., Liu, J., Liang, Y., & Zhou, M. (2026). HeteroGCL: A heterogeneous graph contrastive learning framework for scalable and sustainable cryptocurrency AML. Applied Sciences, 16(6), 2860.

[33] Lin, Y., Wang, Y., & Bian, M. (2024). XOR-EW: An explainable and cost-aware artificial intelligence framework for operational risk early warning in financial institutions. AI and Data Science Journal, 1(1), 94-102.

[34] Li, P., Feng, E., Wang, G., & Mo, T. (2024). Hybrid deep reinforcement learning and integer optimization for resource scheduling in heterogeneous AI computing clusters. Innovation and Technology Studies, 1(1), 48-56.

[35] Shen, A. (2023). Compute-efficient scaling of large sequential recommendation models under dynamic user behavior. World Journal of Information Technology, 1(1), 62-71.

[36] Wang, J., Liu, J., Zheng, W., & Ge, Y. (2025). Temporal heterogeneous graph contrastive learning for fraud detection in credit card transactions. IEEE Access.

[37] Feng, E., Li, P., Mo, T., & Wang, G. (2022). PRISM: Prediction-guided risk-aware deep reinforcement learning for energy-aware dynamic resource scheduling in distributed cloud data centers. Journal of Computer Science and Electrical Engineering, 4(1), 53-61.

[38] Liang, Y., Jiao, Y., Ping, W., Fan, H., & Han, X. (2026). Adaptive event-driven labeling: A neuro-symbolic multiagent framework for causal inference in non-stationary time series. IEEE Access.

[39] Chen, T., Shang, W., & Guo, Z. (2024). XSLA: An SLA-aware heterogeneous XPU scheduling framework for edge AI inference. World Journal of Information Technology, 2(2), 48-59.

[40] Yue, X., Yang, J., & Liu, W. (2026). FraudDebate-Agent: A multi-agent LLM framework with an evidence-based debate mechanism for financial statement fraud detection. Mathematics, 14(15), 2695.

[41] Zhu, R., & Yue, X. (2024). An explainable machine learning framework for predicting dividend sustainability and financial risk of US REITs under interest-rate shocks. Journal of Trends in Finance and Economics, 1(2), 48-57.

[42] Fan, H., Liang, Y., Jiao, Y., Ping, W., & Lin, Y. (2026). An explainable temporal machine learning framework for financial fraud risk prediction in listed institutions. Highlights in Business, Economics and Management, 70, 173-185.

[43] Zhang, F., & Yang, J. (2024). Robustness-aware model compression for multimodal perception under challenging conditions. AI and Data Science Journal, 1(1), 103-111.

[44] Teng, D. (2025). TRIAD: Deadline- and reliability-aware scheduling for post-earthquake structural assessment workflows. World Journal of Engineering Research, 3(6), 55-66.

[45] Lin, Y., Yue, X., Bian, M., Zhu, R., & Teng, D. (2026). Subgraph-level temporal graph transformer with laundering-motif prototypes for anti-money laundering on blockchains. IEEE Access.

[46] Chen, J., Zhang, X., & Han, X. (2023). Explainable identification of tax-related financial reporting anomalies in U.S. public companies: Cross-fitted tax-consistency residuals and Owen-value Shapley propagation. World Journal of Economics and Business Research, 1(1), 33-45.

[47] Zhang, C., & Wang, Y. (2024). A data-driven multi-objective production scheduling framework for battery pack manufacturing under material arrival uncertainty. AI and Data Science Journal, 1(1), 86-93.

[48] Wang, J., Liu, J., Ge, Y., & Jiang, B. (2023). Risk-aware ensemble learning for credit default prediction under economic uncertainty. World Journal of Economics and Business Research, 1(1), 18-32.

[49] Zou, J., Qin, Y., & Rhee, M. (2024). Beyond model scale: A task-aware reliability benchmark for large language models in clinical decision support. Innovation and Technology Studies, 1(1), 39-47.

[50] Zhao, X., Liu, J., Wang, Y., & Wang, J. (2026). CryptoMamba-SSM: Linear complexity state space models for cryptocurrency volatility prediction. IEEE Open Journal of the Computer Society, 7, 226-243.

[51] Chen, J., Cui, Y., Zhang, X., Yang, J., & Zhou, M. (2024). Temporal convolutional network for carbon tax projection: A data-driven approach. Applied Sciences, 14(20), 9213.

[52] Wang, J., Tan, Y., Jiang, B., Wu, B., & Liu, W. (2025). Dynamic marketing uplift modeling: A symmetry-preserving framework integrating causal forests with deep reinforcement learning for personalized intervention strategies. Symmetry, 17(4), 610.

[53] Wang, J., Ge, Y., Liu, J., & Jiang, B. (2024). Early warning of liquidity stress in U.S. banks with seasonally aligned frailty hazards and delayed-feedback conformal alarms. Journal of Trends in Finance and Economics, 1(1), 85-100.

[54] Qin, Y., Zou, J., & Rhee, M. (2022). An AI-driven intelligent transaction execution framework for internet finance platforms: Cost-sensitive real-time integrity interdiction. Eurasia Journal of Science and Technology, 4(2), 36-44.

[55] Sun, T., Yang, J., Li, J., Chen, J., Liu, M., Fan, L., & Wang, X. (2024). Enhancing auto insurance risk evaluation with transformer and SHAP. IEEE Access, 12, 116546-116557.

[56] Chen, J., Liang, Y., Liu, J., & Zhou, M. (2026). Temporal transformer with conditional tabular GAN for credit card fraud detection: A sequential deep learning approach. Mathematics, 14(7), 1183.

[57] Jiao, Y., Fan, H., Yue, X., Ping, W., Sun, T., & Wang, J. (2026). Dynamic heterogeneous graph contrastive learning for uncovering collusive financial fraud. Scientific Reports.

[58] Wang, Y., & Zhang, C. (2023). AERIS: Adaptive executor-scoped resource management for privacy-preserving data processing in Apache Spark. Eurasia Journal of Science and Technology, 5(2), 25-33.

[59] Wang, G., Li, P., Feng, E., & Mo, T. (2024). Uncertainty-aware LLM-as-a-judge evaluation for reliable search and recommendation systems: Calibrated confidence and impact-aware selective human review. AI and Data Science Journal, 1(1), 112-121.

[60] Wang, B., Zhao, W., & Wang, Z. (2024). DAS-GNN: A scalable disagreement-aware graph neural framework for anomaly detection in large-scale networks. AI and Data Science Journal, 1(1), 67-75.