JRITM
JOURNAL OF RESEARCH AND INNOVATION IN TECHNOLOGY, COMMERCE AND MANAGEMENT

Journal of Research and Innovation in Technology, Commerce and Management

Vol. 3 Issue 9, September 2026, pp. 39066-39074

ISSN: 3049-3129 (Online)

What Actually Fits: Context Selection for Local-First Retrieval on Memory-Constrained Hardware

Mohammed Wasif Ul Haq
Lords Institute of Engineering and Technology, Telangana
Email: wasif.haq434701@gmail.com

Cite as: Mohammed Wasif Ul Haq. (2026). What Actually Fits: Context Selection for Local-First Retrieval on Memory-Constrained Hardware. Journal of Research and Innovation in Technology, Commerce and Management, Vol. 3(Issue 9), 39066–39074. https://doi.org/10.5281/zenodo.22705201

DOI: https://doi.org/10.5281/zenodo.22705201

***Abstract*—**Running a language model on your own machine keeps your files private, but consumer hardware imposes a context budget that server deployments never feel. We report measurements from a deployed local-first system that indexes a user's files and answers questions over them. Early versions injected whole-project content into every turn: roughly 59K tokens, prompt evaluation in the minutes, mitigated by truncating context to the trailing 8,000 characters. Three findings. First, retrieval-based selection recovers most of the available answer coverage at a small fraction of the prompt: on real repositories a 2,500-character budget reaches 0.79 topic recall against whole-project injection's 1.00, using 23 times fewer tokens with 14.9 times faster prompt evaluation. Second, the trailing-window truncation that several local tools ship is close to worthless on real corpora, 0.12 recall on one repository and 0.00 on another, against 1.00 on a synthetic benchmark, where its apparent success is an artifact of file ordering. Third, and the reason for this paper's framing: a diversity-aware selection rule that improved recall from 0.62 to 0.93 on our synthetic benchmark is corpus-dependent. It gains 0.64 to 0.78 on a document corpus but slightly loses on both code repositories. We give a predictor computable from retrieval alone: where a flat top-k already spans more than about 1.5 clusters, forcing further spread costs relevance; where it concentrates below that, diversifying pays. It separates all four corpora. We also report two measurement faults that each produced a confident false positive in our own results before real corpora exposed them: prompt-prefix caching, which inverts latency comparisons by more than 500 times, and substring-based recall scoring, which under-reports on any corpus whose files exceed one chunk.

***Keywords*—**Retrieval-Augmented Generation; On-device Inference; Local-first Software; Context Selection; Benchmark Validity

I. INTRODUCTION

Vol. 3 Issue 9, September 2026, pp. 39066-39074

ISSN: 3049-3129 (Online)

Sending a codebase or a folder of personal documents to a hosted model means handing that material to a third party. Local inference removes the exposure, and quantised models make it practical on ordinary laptops. What does not transfer from the server setting is headroom. A datacentre deployment can place tens of thousands of tokens in front of a model; a laptop sharing memory between an operating system, an editor and a 12B model cannot.

This paper starts from a failure in a deployed system rather than a hypothetical. The system is a local-first desktop application that indexes a user's files, source code, PDFs, Office documents, spreadsheets and SQLite databases, into a per-project vector store and answers questions over them with a local or cloud model. Early versions injected whole-project content into every turn: roughly 59K tokens, and prompt evaluation measured in minutes. The stopgap was a blunt truncation to the trailing 8,000 characters, cheap, and blind to the question.

We set out to show that a diversity-aware selection rule, reusing *k*-means cluster identifiers already computed at index time, would spend a small budget better than a flat nearest-neighbour lookup. On a synthetic benchmark built for the purpose it did exactly that, lifting topic recall from 0.62 to 0.93. On real repositories it does not. Establishing that took two corrections to our own measurement apparatus, both of which had produced results we believed.

On a document corpus it gains again. The mechanism is real but conditional, and the condition is measurable.

We think the resulting paper is more useful than the one we intended to write. What survives is a quantified case for retrieval-based context reduction on constrained hardware, a warning about a truncation strategy that is widely shipped and demonstrably poor, and a measured account of when diversity-aware selection helps and when it does not.

Contributions. We contribute a measured characterisation of context reduction for local-first retrieval across two inference backends and three budget tiers; evidence that trailing-window truncation collapses on real corpora and that a synthetic benchmark hides this; a corpus-dependent result with a predictor, namely that diversity-aware selection gains on documents and on a synthetic corpus, loses on real code, and that the cluster spread of a flat top-*k* separates the cases; two benchmark-validity faults, prefix caching and substring recall scoring, each of which produced a false positive in our own data; and an open harness with pre-registered ground truth.

II. Background

Retrieval-augmented generation. Grounding generation in retrieved passages is well established [1], usually as a flat top-*k* nearest-neighbour lookup over a dense index of sentence embeddings [2]. HyDE-style query expansion improves recall when query and target use different vocabulary [3]. These improve *which* neighbourhood is found; our concern is how a small budget is distributed once one is located.

Result diversification. That relevance alone is a poor selection criterion is long established in information retrieval; Maximal Marginal Relevance and its descendants trade relevance against novelty [4]. Our rule pursues the same end using a clustering computed at index time rather than pairwise similarities at query time, coarser, but effectively free, which matters on a machine chosen because it lacks compute.

On-device inference. Work on post-training quantisation [5], sparse expert routing [6], paged key-value caches [7] and flash offloading [8] targets making a model *fit*; open weights [9] are what make local deployment possible at all. We take fitting as given and ask what to put in front of a model that fits but has little room to spare.

III. System

ISSN: 3049-3129 (Online)

The measurements come from a shipping application, and its design shapes the constraints. The system extracts text from source files across roughly twenty languages, PDFs, DOCX, PPTX, XLSX, CSV, SQLite databases and images via OCR, chunks it, and embeds it with a MiniLM sentence encoder into a per-project vector collection. Content hashing means only changed files are reprocessed. Chunk embeddings are additionally projected to three dimensions and grouped with *k*-means to drive an interactive graph of the project.

Around that sit an editor with an integrated terminal, a code runner and a debug-adapter client; a document editor with grammar and completion support; a research agent; generation of PDF, PPTX, DOCX and CSV artifacts; and a local memory ledger. Model routing covers local inference, Apple MLX and cloud providers behind one key field, with a hardware-aware catalogue that recommends models which will fit.

The system ships as a one-click installer for macOS, Windows and Linux, bundling the backend and native shell. This is not incidental: the intended user has documents and questions, not a toolchain. It also constrains the work, mechanisms must degrade to a clear in-application message rather than a traceback, and must not assume a GPU.

Context budget. The budget follows the model's footprint relative to available memory. For a model with p billion parameters we estimate a 4-bit resident footprint m of about $0.6p$ GB and take usable memory a as device VRAM, or a conservative fraction of system RAM. The ratio $r = m/a$ selects a character budget: 2500 for $r < 0.35$, 1600 for $0.35 \le r < 0.7$, and 900 above. The budget is split across three chunks: on a real repository, splitting a fixed 900-character ceiling across 1, 2, 3 and 4 chunks yields recall 0.53, 0.64, 0.80 and 0.82, more chunks reach more files.

IV. Experimental Setup

Corpora. Three, deliberately different. A *synthetic* corpus of 72 chunks built with one dominant redundant topic (45 near-duplicate session modules) plus three minority topics. Two real repositories: requests (72 chunks) and fastapi (202 chunks), chosen by size regime before any result was seen. And a *document* corpus of six CC-licensed ML-systems papers (161 chunks), deliberately same-domain so that topics genuinely overlap; licences were verified per paper and papers under arXiv's default non-exclusive licence were excluded from redistribution even where cited here.

Ground truth is pre-registered. Topics, markers and queries for each repository were written from its own module structure and committed to version control before any strategy was run. Each repository has 13 queries, of which 10 require two or more modules; the remainder are single-topic controls that would expose a strategy that harms focused queries.

Metric. *Topic recall*: the fraction of the topics a correct answer requires that are present in the assembled prompt, scored by the source file each injected chunk came from. Section VII explains why scoring by source file rather than by marker text is not a detail. Topic recall measures whether the needed material reached the model, not whether the model used it.

Strategies. full injects every chunk; truncate keeps the trailing 8,000 characters; flat is top-k at a fixed budget; adaptive sizes the budget by hardware; cluster adds diversity-aware selection. Each adds one mechanism to the previous, so differences isolate that mechanism.

Cost measurement uses deepseek-r1:14b through llama.cpp and gemma4:12b through Apple MLX, on a machine with 21.6 GB usable memory. The model is warmed before measurement and every call is cache-busted. Medians over queries; dispersion is inter-quartile range.

Vol. 3 Issue 9, September 2026, pp. 39066-39074

ISSN: 3049-3129 (Online)

V. Context Reduction

TABLE I

Constrained tier, cache-busted, warm model, synthetic corpus. Medians over seven queries; IQR in parentheses.

	recall	tokens	p-eval (ms)	total (ms)
full	1.00	6624	32,975 (10,191)	41,389
truncate	1.00	2222	18,106 (5,878)	29,351
adaptive	0.62	288	2,103 (263)	11,505
cluster	0.93	291	2,207 (313)	11,829

Table I gives the cost side. Retrieval reduces the prompt by 23 times, prompt evaluation by 14.9 times and end-to-end latency by 3.5 times (Figs. 1 and 2). Repeating the measurement on a second backend and budget tier gives 13.0 times on prompt evaluation for comparable throughput (291 against 201 tokens/second cache-busted), so the reduction is not specific to one configuration.

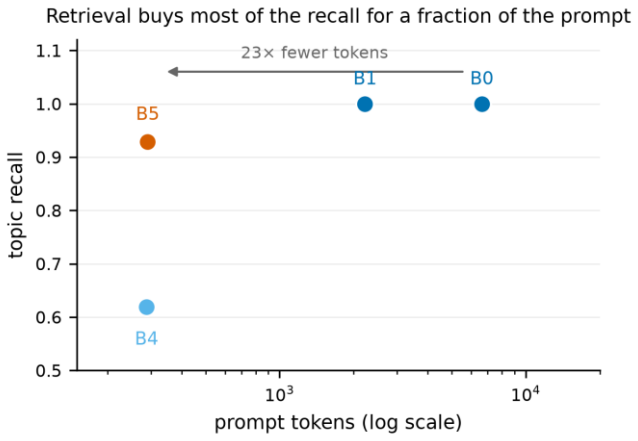

Fig. 1. Recall against prompt size. The gap between the retrieval strategies and whole-project injection is two orders of magnitude in tokens.

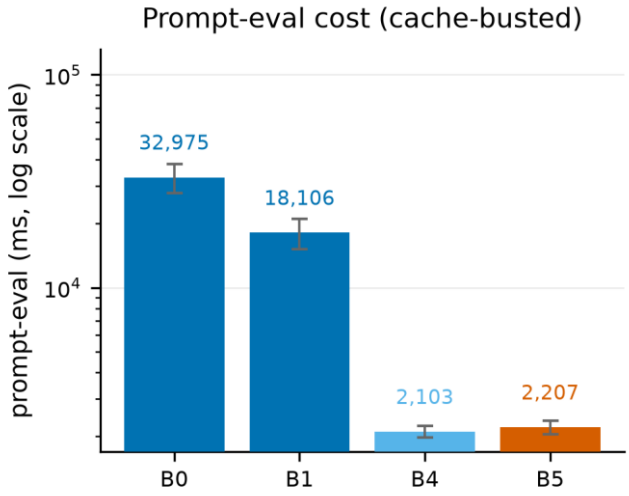

Fig. 2. Prompt-evaluation cost, log scale, IQR error bars.

TABLE II

Recall across four corpora. Whole-project injection is the ceiling; the retrieval budget is 2,500 characters throughout.

	synth.	docs	requests	fastapi
full (chars)	23,044	399,513	182,593	605,212
full	1.00	1.00	1.00	1.00
truncate	1.00	0.12	0.12	0.00
flat	0.62	0.64	0.79	0.53
adaptive	0.62	0.64	0.79	0.53
cluster	0.93	0.78	0.74	0.51

Table II carries the more important result. On requests, retrieval reaches 0.79 of the achievable recall from 2,500 characters against a 182K-

Vol. 3 Issue 9, September 2026, pp. 39066-39074

ISSN: 3049-3129 (Online)

character corpus, 1.4% of the material, for four fifths of the coverage.

Trailing-window truncation collapses. It scores 1.00 on the synthetic corpus and 0.12 and 0.00 on the two real ones. The synthetic result is an artifact: that corpus is written in topic order, so its final 8,000 characters happen to span topics. On a real repository the cut point is arbitrary, and the strategy is close to worthless. This matters beyond our system, trailing-window truncation is a common default in local RAG tooling, and a synthetic evaluation will not reveal the problem.

VI. When Diversity-Aware Selection Helps

Our original hypothesis was that a flat top-*k* over a redundant corpus wastes a small budget inside one semantic neighbourhood, and that spreading selection across index-time clusters would fix it. On the synthetic corpus this holds emphatically. Splitting a fixed 900-character ceiling across more chunks, flat top-*k* barely moves, 0.62 through three chunks, 0.67 at four, while diversified selection climbs to 0.93 (Fig. 3, left): additional retrieval depth converts into coverage only when selection spreads.

On requests the same experiment reverses (Fig. 3, right). Flat top-*k* is *not* flat in the budget, it climbs 0.53, 0.64, 0.80, 0.82, and beats diversified selection at every point.

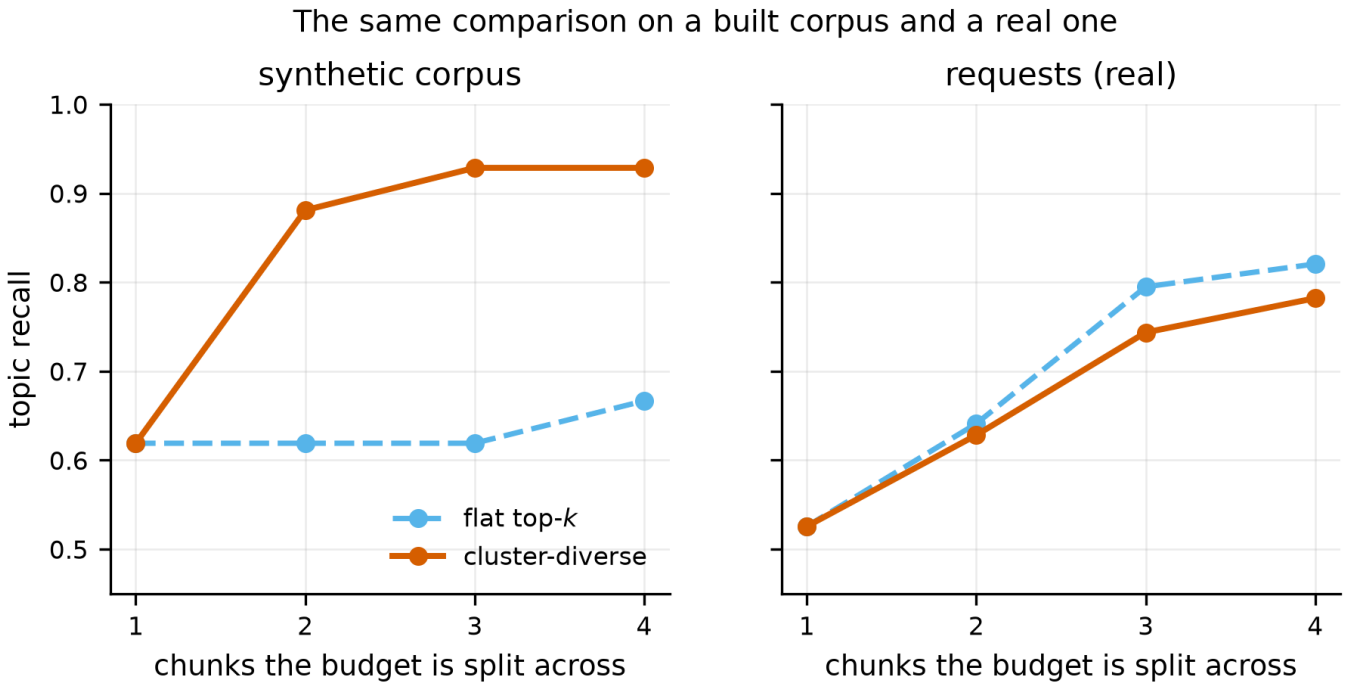

Fig. 3. The same ablation on the corpus we built and on a real repository. The ordering of the two strategies inverts.

This is a property of the corpora, not the measurement: re-scoring the synthetic corpus by the same source-file rule used for the repositories reproduces its original result exactly.

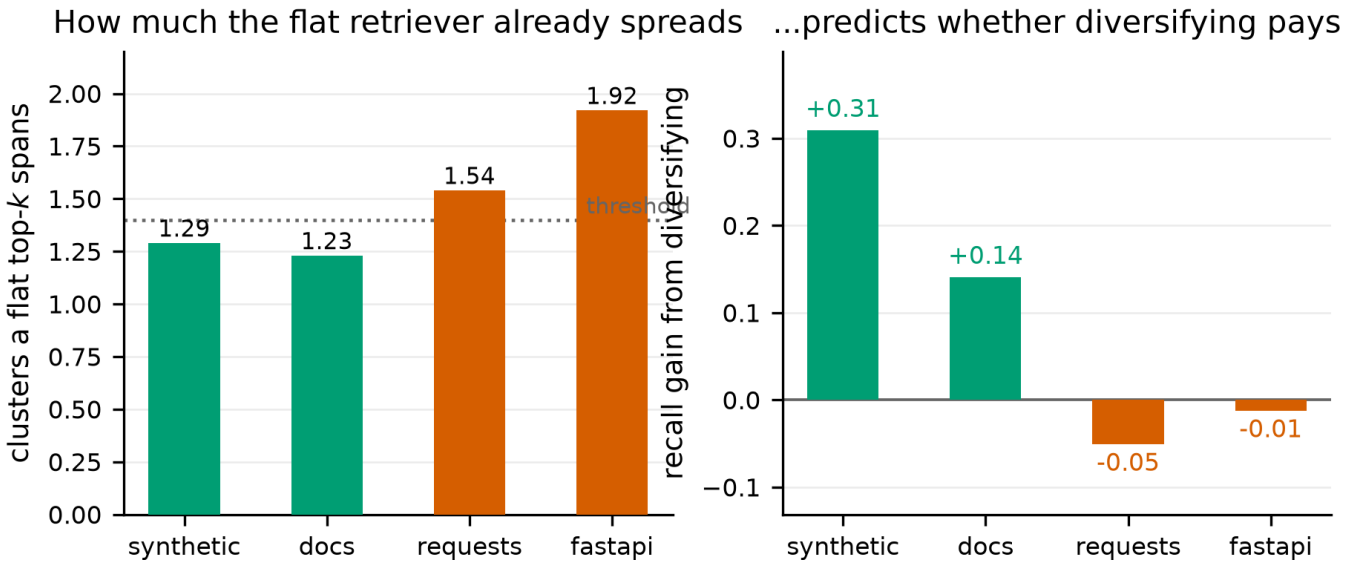

Fig. 4. Left: what the retriever returns. Right: the recall change from diversifying. Diversity pays where duplicates span files, and costs where they do not.

Fig. 4 measures the difference. On the synthetic corpus the top-12 retrieved chunks have mean pairwise cosine similarity 0.80 and *no* two come from the same file: it was built from 45 near-duplicate modules, so its redundancy is exactly the across-file kind diversification removes. On both real repositories similarity is 0.60 and 59% of the retrieved set is multiple chunks of the same file. That is not redundancy. Adjacent chunks of the relevant module are what answers the question, and evicting them to reach a different cluster costs recall.

Documents behave like the synthetic corpus, not like code. On the six-paper document corpus, diversified selection gains: 0.78 against 0.64 (Table III). So this is not simply an artifact of a corpus we built, there is a real, non-synthetic case where the mechanism pays.

What predicts the sign. Our first hypothesis was that retrieved-set redundancy across distinct files was the discriminator. It is not: the document corpus has the *highest* same-file rate of the four (0.78) and still benefits, and pairwise similarity does not separate them either. What does separate them is how much the flat retriever already spreads.

TABLE III

Cluster spread of a flat top-k predicts whether diversifying helps. Threshold is descriptive over four corpora, not validated.

	clusters	flat recall	gain
synthetic	1.29	0.62	+0.31
documents	1.23	0.64	+0.14
requests	1.54	0.79	−0.05
fastapi	1.92	0.53	−0.01

Where a flat top-*k* already spans more than about 1.5 clusters, it is finding distinct material unaided and forcing further spread trades

Vol. 3 Issue 9, September 2026, pp. 39066-39074

ISSN: 3049-3129 (Online)

relevance for nothing. Where it concentrates below that, the budget is being spent inside one neighbourhood and diversifying recovers coverage. The statistic needs no labels and is available at query time, so a system can choose per corpus. We ship flat by default and expose the diversified rule.

We note the scope: four corpora, and a threshold read off four points rather than validated on held-out ones.

VII. Benchmark Validity

Two faults in our own apparatus each produced a result we believed. Both are easy to reproduce accidentally.

Prompt-prefix caching inverts latency comparisons. Re-sending an identical prompt drops prompt evaluation from 21,204 ms to 39 ms, a 544 times drop, while three distinct prompts of comparable length cost 32,106, 30,980 and 31,604 ms (Fig. 5). Repeating a measurement therefore measures the cache. Worse, whole-project injection is the *most* cacheable strategy in any such comparison: its context is byte-identical across queries, where retrieval produces a different context each time. Our uncorrected figure for it was 190 ms against a true 32,975 ms, a 173 times error in the direction that made the naive baseline look fastest. Every call now carries a unique prefix.

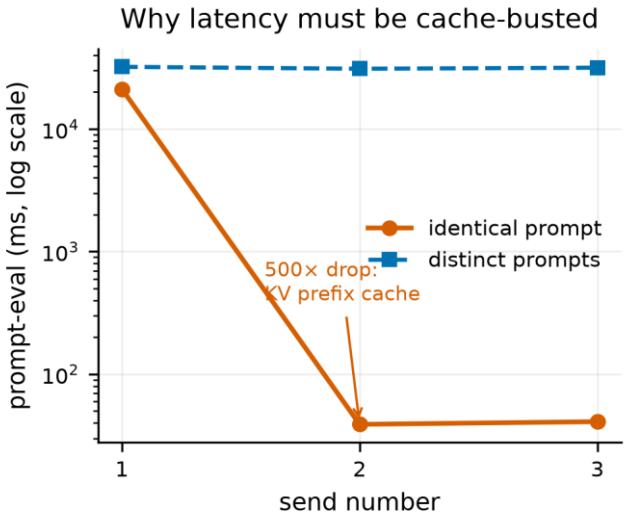

Fig. 5. Prompt evaluation for a repeated prompt against distinct prompts of similar length.

Substring recall scoring under-reports on real corpora. Our first metric scored a topic as present when one of its marker identifiers appeared in the prompt. On the synthetic corpus, where every file is a single chunk, this is equivalent to scoring by source file. On a real repository a module spans many chunks and its markers sit in one of them, so retrieving a *different* chunk of exactly the right file scores zero. On fastapi this counted a query that retrieved routing.py three times as a complete miss, and produced an apparent +54% for diversified selection that disappeared entirely once scoring moved to source files.

Marker leakage. Relatedly, markers must be checked for uniqueness. Our first hand-written manifest had 15 markers of 27 that also appeared in other topics' files, and two topics with none that were unambiguous. A leaking marker credits its topic whenever any file containing it is retrieved, inflating recall for every strategy equally and compressing the differences under test. The harness now rejects such manifests.

Vol. 3 Issue 9, September 2026, pp. 39066-39074

ISSN: 3049-3129 (Online)

The common thread is that all three faults produce *plausible* numbers. None throws an error, and none looks anomalous without a second corpus or a second metric to disagree with.

VIII. Threats to Validity

Two repositories, one language. Both are Python libraries of similar character. Broader coverage is the obvious next step.

The budget discards most of what retrieval returns. We expected prose to be truncated where code passed through whole: chunking splits at 400 words, roughly 2,400 characters, against a per-chunk allowance of 600. Measured, the first half holds and the second does not. Prose chunks have median 2,529 characters and 24% of each survives; real code chunks have median 2,802 and also 24% survives. The cap discards about three quarters of every retrieved chunk regardless of modality. Our earlier expectation that code would be unaffected came from the synthetic corpus, whose files are around 280 characters and fit whole. Whether the discarded remainder carries answer-relevant material is untested, and is the most concrete open question these results leave.

Recall is a proxy. It measures whether required material reached the prompt, not whether the model used it correctly.

Beyond the memory cliff. A 27B model requiring 22.3 GB on a machine with 21.6 GB usable drove 8.7 GB of swap and over five minutes for a single-token generation. Past that point prompt length is not the bottleneck and context selection rescues nothing. Our claims concern models that fit on machines with little headroom.

Cluster granularity. The index requests six clusters regardless of corpus size. Raising it to 16 and 48 on fastapi changed nothing, but the parameter is otherwise unexplored.

IX. Conclusion

For local-first assistants on constrained hardware, retrieval-based context selection is a large and reliable win: on a real repository, 1.4% of the material yields four fifths of the achievable answer coverage, with prompt evaluation an order of magnitude faster. The trailing-window truncation that several local tools ship instead is close to worthless on real corpora, and a synthetic benchmark will not reveal it.

Our intended contribution, diversity-aware selection using index-time clusters, does not generalise. It is a substantial gain on a corpus whose redundancy spans distinct files, and a small loss where the retrieved set concentrates on one genuinely relevant module, which is what real code does. We report the statistic that distinguishes the two, and default to flat selection.

The broader lesson is about benchmark construction. A corpus built to exhibit the phenomenon a method addresses will confirm that method, and a metric that looks reasonable on that corpus can fail silently on any other. Both of our false positives were visible only from a second corpus.

Artifact Availability

The system, the benchmark harness, the pre-registered ground truth and every results file are available in a public repository, withheld here for double-blind review. A verification script recomputes every numeric claim in this paper from those results files and exits non-zero on any mismatch. The document corpus is not redistributed; the repository records the URL, licence and retrieval date of each source.

AI Disclosure

AI-assisted software tools were used while developing the system described here and while

drafting and editing this manuscript. The experimental design, the benchmark corpora and ground truth, the measurements and their interpretation are the author's own. Every numeric claim in this paper is recomputed from the recorded results files by a verification script distributed with the artifact, and the author takes full responsibility for the accuracy and originality of the work.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W. Yih, T. Rocktaschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," arXiv preprint arXiv:2005.11401, 2020.

[2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," arXiv preprint arXiv:1908.10084, 2019.

[3] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," arXiv preprint arXiv:2212.10496, 2022.

[4] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in Proc. 21st Annu. Int. ACM SIGIR Conf. Research and Development in Information Retrieval, 1998, pp. 335–336.

[5] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "GPTQ: Accurate post-training quantization for generative pre-trained transformers," arXiv preprint arXiv:2210.17323, 2022.

[6] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de las Casas, E. B. Hanna, F. Bressand, et al., "Mixtral of experts," arXiv preprint arXiv:2401.04088, 2024.

[7] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with PagedAttention," arXiv preprint arXiv:2309.06180, 2023.

[8] K. Alizadeh, I. Mirzadeh, D. Belenko, K. Khatamifard, M. Cho, C. C. Del Mundo, M. Rastegari, and M. Farajtabar, "LLM in a flash: Efficient large language model inference with limited memory," arXiv preprint arXiv:2312.11514, 2023.

[9] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Roziere, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," arXiv preprint arXiv:2302.13971, 2023.