WCAR: Weighted Contextual Accumulative Retrieval for Context Propagation in RAG Systems

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) systems commonly rely on cosine similarity to retrieve relevant document chunks. However, this approach evaluates each chunk independently, ignoring relationships between neighboring chunks. As a result, semantically distant yet contextually important information is often overlooked.

In this work, we propose **Weighted Contextual Accumulative Retrieval (WCAR)**, a lightweight retrieval framework that extends cosine-based retrieval through progressive context accumulation. WCAR introduces a similarity-weighted linear combination mechanism over sequentially retrieved embedding vectors, wherein each accumulated context state is expressed as a weighted sum of previously selected chunk embeddings governed by their respective similarity scores. We further present a mathematical formulation of WCAR and demonstrate how it emerges from two intermediate strategies: sequential addition retrieval and weighted contextual sequential retrieval.

We evaluate WCAR on multiple datasets from the BEIR benchmark, including SciFact, NFCorpus, and FiQA, as well as a refined version of HotpotQA, using the lightweight `all-MiniLM-L6-v2` embedding model. On SciFact, compared to standard cosine-based retrieval, WCAR achieves up to **+5.8% improvement in NDCG@10** and **+4.0% improvement in Recall@10**. When combined with BM25, WCAR further improves performance with **+7.3% gain in NDCG@10** and **+6.0% improvement in Recall@10** over cosine-based retrieval. Across datasets, WCAR consistently demonstrates improvements in ranking quality while maintaining low computational overhead.

1 Introduction

Information retrieval is a fundamental component of modern AI systems, particularly in Retrieval-Augmented Generation (RAG), where the quality of retrieved context directly impacts downstream generation. While recent advances in embedding models have improved semantic matching, achieving strong retrieval performance often relies on computationally expensive pipelines, including large language models or multi-stage reranking systems. In contrast, lightweight retrieval approaches typically depend on cosine similarity, which evaluates each document chunk independently.

This independence assumption introduces a critical limitation: it fails to capture contextual relationships between chunks. In many real-world scenarios, relevant information is distributed across multiple segments, where individual chunks may appear weakly relevant in isolation but become important when considered in relation to previously retrieved content.

To illustrate this limitation, consider the following example:

- **A** = "The 2008 financial crisis was largely caused by the collapse of the housing market and the widespread issuance of subprime mortgages."
- **B** = "Financial institutions created complex derivatives such as mortgage-backed securities and collateralized debt obligations to redistribute risk."
- **Q** = "What caused the 2008 financial crisis?"

Using cosine similarity, $\cos(A, Q) = 0.7043$ leads to the retrieval of **A**, while $\cos(B, Q) = 0.3905$ results in **B** being discarded. However, **B** provides crucial contextual information that complements **A**. This example highlights a broader challenge: a chunk can be *semantically distant* from a query while remaining *contextually relevant* through its relationship with other retrieved chunks.

We formalise this observation as a contextual dependency chain, where information propagates

across chunks (e.g., $B \rightarrow A \rightarrow Q$). Existing approaches attempt to capture such dependencies through complex architectures or reranking strategies (Nogueira and Cho, 2019), often at the cost of increased computational overhead. In contrast, we focus on a lightweight alternative that explicitly models context accumulation within the embedding space.

In this work, we propose **Weighted Contextual Accumulative Retrieval (WCAR)** framework that progressively accumulates contextual information by representing each candidate chunk as a similarity-weighted linear combination of previously selected vectors. Instead of treating retrieval as a set of independent scoring decisions, WCAR enables contextual signals to propagate across sequentially selected chunks. This is achieved by iteratively updating the query representation as a weighted linear combination of vectors, effectively transferring and refining relevance within the embedding space. This formulation allows the retrieval process to capture inter-chunk relationships directly within the embedding space of lightweight models such as *all-MiniLM-L6-v2*, without requiring complex architectures or expensive reranking pipelines. We evaluate WCAR on multiple datasets, including SciFact (Wadden et al., 2020), NFCorpus (Boteva et al., 2016), and FiQA (Maia et al., 2018) from the BEIR benchmark (Thakur et al., 2021), as well as a refined version of HotpotQA (Yang et al., 2018). Across these datasets, WCAR demonstrates consistent improvements over cosine-based retrieval, particularly in ranking quality, while maintaining low computational overhead. The main contributions of this work are as follows:

- We introduce a progressive retrieval framework consisting of Sequential Addition Retrieval (SAR), Weighted Contextual Sequential Retrieval (WCSR), and Weighted Contextual Accumulative Retrieval (WCAR), along with their mathematical formulation.

- We propose a similarity-aware contextual accumulation mechanism that represents each retrieval state as a weighted *linear combination of sequentially selected vectors*, enabling systematic context transfer and progressive refinement across successive retrieval steps.

- We design an *adaptive thresholding mechanism* that dynamically adjusts acceptance criteria as contextual evidence accumulates.

- We perform comprehensive evaluation on multiple datasets (SciFact (Wadden et al., 2020), NFCorpus (Boteva et al., 2016), FiQA (Maia et al., 2018), and HotpotQA (Yang et al., 2018)), demonstrating consistent improvements over cosine-based retrieval while maintaining efficiency.

- We introduce a hybrid WCAR + BM25 (Robertson and Zaragoza, 2009) variant that further improves retrieval performance, particularly in recall.

2 Related Work

2.1 Lexical and Dense Retrieval

Classical lexical retrieval methods such as BM25 (Robertson and Zaragoza, 2009) remain strong baselines due to their efficiency and effectiveness in term matching. More recently, dense retrieval approaches such as DPR (Karpukhin et al., 2020) have improved semantic matching by encoding queries and documents into continuous vector spaces. Despite these advances, both lexical and dense retrieval methods typically evaluate document chunks independently, limiting their ability to capture contextual relationships across multiple segments.

2.2 Retrieval-Augmented Generation Frameworks

Retrieval-Augmented Generation (RAG) frameworks (Lewis et al., 2020), such as LlamaIndex (Liu, 2023), integrate retrieval with downstream generation to improve response quality. While effective, their performance is highly dependent on the quality of retrieved context and the underlying retrieval strategy. In particular, these systems inherit the limitations of independent chunk scoring, making them sensitive to incomplete or fragmented retrieval.

2.3 Chunking Strategies

A range of chunking strategies has been proposed to improve retrieval quality by preserving contextual information.

Context-aware chunking (Anthropic, 2024) introduces overlapping tokens between adjacent segments to maintain local continuity. While this improves retrieval in some cases, it relies on heuristic

overlap windows and may introduce redundancy or fail to capture longer-range dependencies.

Late chunking (Günther et al., 2024) delays segmentation until after encoding full documents, allowing models to process broader context before splitting into chunks. However, this approach is constrained by input token limits and increased computational cost, particularly for long documents.

Semantic chunking (Qu et al., 2024) aims to partition text into semantically coherent units. Although this can improve chunk quality, it typically yields only modest gains while requiring additional preprocessing and computation.

2.4 Iterative and Agent-based Retrieval

More advanced approaches, such as agent-based (Zhangdutao and Liao, 2026) or iterative retrieval methods, attempt to refine retrieval through multiple steps or adaptive search strategies. These methods can capture complex dependencies and achieve strong performance, but often introduce significant system complexity, latency, and computational overhead.

2.5 Model Scaling and Fine-tuning

Another line of work focuses on improving retrieval through larger or fine-tuned embedding models such as Voyage-Large-2 (Voyage AI, 2023). While such approaches can enhance retrieval quality, they require increased computational resources and may not be suitable for lightweight or cost-sensitive applications.

2.6 Positioning of WCAR

In contrast to aforementioned approaches, WCAR is specifically designed to model contextual relationships intrinsically within the retrieval process. The proposed framework introduces a retrieval mechanism grounded in the linear combination of vectors, through which contextual information is systematically accumulated and propagated across sequentially retrieved chunks. This mathematically principled formulation empowers the retrieval process to effectively capture inter-chunk semantic dependencies, while rigorously preserving the computational efficiency inherent to lightweight embedding architectures. Furthermore, WCAR is architected as a model-agnostic, zero-overhead augmentation layer, designed for seamless integration into any standard cosine similarity-based retrieval pipeline, thereby significantly enhancing contextual recall without introducing additional architectural complexity or computational burden to the underlying system (Reimers and Gurevych, 2019).

3 Methodology

Our approach is developed through a sequence of progressively refined retrieval formulations, each addressing a specific limitation of the previous method. Starting from a simple additive framework, we incrementally introduce weighting and contextual accumulation, ultimately arriving at the proposed Weighted Contextual Accumulative Retrieval (WCAR) formulation.

3.1 Motivation: The Contextual Gap Problem

Let $Q \in \mathbb{R}^d$ denote the query embedding and $A, B \in \mathbb{R}^d$ denote embeddings of document chunks. Standard retrieval evaluates each chunk independently:

$$\text{retrieve}(X) \iff \cos(X, Q) \geq \tau$$

where τ is a fixed acceptance threshold. A fundamental limitation arises when:

$$\cos(A, Q) \geq \tau, \cos(B, Q) < \tau, \text{yet } \cos(A, B) \gg 0$$

In this setting, A is strongly relevant to Q, and B is moderately related to A, implying that B may contain complementary contextual information necessary for answering Q. However, B is discarded due to its low direct similarity with the query. We formalize this as a *contextual dependency chain*:

$$B \to A \to Q$$

This motivates a retrieval paradigm that evaluates chunks not independently, but in the context of previously accepted content.

3.2 Phase 1: Sequential Addition Retrieval (SAR)

To capture contextual dependencies, we introduce a sequential retrieval process. Once a chunk A is accepted, subsequent chunks are evaluated in the context of previously accepted content rather than independently. We define an accumulated representation:

$$R \leftarrow A$$

and update it as each new chunk B is considered:

$$R \leftarrow R + B$$

The retrieval condition becomes:

$$\text{retrieve}(B) \iff \cos(R, Q) \geq \tau$$

In the motivating example, while $\cos(B, Q)$ is low, the combined representation improves alignment with the query:

$$\cos(A + B, Q) = 0.832 > \cos(B, Q) = 0.496$$

allowing B to be retrieved through contextual reinforcement.

3.2.1 Adaptive Anchor Reset

As R grows with accumulated chunks, a fixed threshold τ becomes inadequate. We define an adapted threshold:

$$\tau_{\text{adapted}} = \tau^* \cdot \alpha$$

where $\alpha \in (0, 1)$ is a decay factor. If $\cos(R, Q) < \tau_{\text{adapted}}$, the accumulation is reset to the next high-confidence chunk C:

$$R \leftarrow C$$

and the threshold is restored to τ^*.

3.2.2 Limitation of SAR

Simple vector addition introduces a *dominance bias*: earlier chunks can overshadow later ones, leading to:

$$\cos(A, Q) > \cos(R, Q) > \cos(B, Q)$$

This masks the individual contribution of newly added chunks and prevents accurate contextual evaluation.

3.3 Phase 2: Weighted Contextual Sequential Retrieval (WCSR)

To mitigate the dominance bias of SAR, we introduce a weighting factor K that controls the degree of influence exerted by prior context:

$$R \leftarrow A, \qquad R \leftarrow K \cdot A + B$$

We define $K = \cos(A, B)$, so that the contribution of A is proportional to its similarity to B:

$$R \leftarrow \cos(A, B) \cdot A + B$$

This generalizes across multiple chunks:

$$R \leftarrow \cos(A, C) \cdot A + \cos(B, C) \cdot B + C$$

From the motivating example, this yields $\cos(R, Q) = 0.757$, sufficient for retrieval of B.

3.3.1 Limitation of WCSR

This formulation requires pairwise similarity computations between all prior chunks and each new chunk, resulting in $O(n^2)$ complexity — prohibitive as the number of retrieved chunks grows.

3.4 Phase 3: Weighted Contextual Accumulative Retrieval (WCAR)

To preserve the contextual weighting behavior of WCSR while reducing computational cost, we collapse the chunk history into a single accumulated representation R:

$$R \leftarrow A, R \leftarrow \cos(R, B) \cdot R + B, R \leftarrow \cos(R, C) \cdot R + C,$$

Formally, the update rule at each step t is:

$$R_{t+1} = C + \cos(R_t, C) \cdot R_t$$

where C denotes a newly considered chunk. This reduces complexity to $O(n)$ per query while preserving the weighting behavior of WCSR.

3.4.1 Deviation-Based Acceptance Criterion

Instead of a strict threshold $\cos(R, Q) \geq \tau$, WCAR employs a deviation-based criterion:

$$\left|\cos(R, Q) - \tau^*\right| < \delta$$

where δ is a tolerance parameter, improving robustness under varying levels of context accumulation.

3.4.2 Adaptive Thresholding

The acceptance threshold is dynamically adjusted based on the highest observed relevance of retrieved chunks:

$$\tau_n = \left(s_{\text{best}} \cdot \frac{n}{n+1} \right)$$

where n is the number of accepted chunks, s_{best} is the maximum similarity score observed so far, and $\lambda = 0.8$ in our experiments. This formulation provides:

- **Progressive tightening:** $\frac{n}{n+1}$ increases with n, making the threshold stricter as more context is accumulated.

Table 1 illustrates the evolution of the threshold under representative values.

Table 1: Adaptive threshold values as chunks accumulate ($\tau^* = 0.68$)

Accepted Chunks (n)	$\tau_n = \tau^* \cdot \left(1 - \frac{1}{1+n}\right)$	Value
0	$\tau^* \cdot 0.000$	0.000
1	$\tau^* \cdot 0.500$	0.340
2	$\tau^* \cdot 0.667$	0.454
3	$\tau^* \cdot 0.750$	0.510
4	$\tau^* \cdot 0.800$	0.544
5	$\tau^* \cdot 0.833$	0.567
7	$\tau^* \cdot 0.875$	0.595
9	$\tau^* \cdot 0.900$	0.612
11	$\tau^* \cdot 0.917$	0.623

3.4.3 Lightweight Reranking

We incorporate a lightweight reranking step (Nogueira and Cho, 2019) to refine the final set of selected chunks. The reranker operates only on the small subset of top candidates produced by WCAR and is used to improve ordering rather than alter retrieval behavior. The primary performance gains of WCAR arise from the contextual accumulation mechanism itself; reranking provides only marginal additional improvements.

3.5 Evaluation Metrics

We evaluate retrieval performance using standard information retrieval metrics (Thakur et al., 2021):

- **MRR@10** (Mean Reciprocal Rank):

$$\text{MRR@10} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}_q}$$

- **NDCG@10** (Normalized Discounted Cumulative Gain):

$$\text{NDCG@10} = \frac{1}{\text{IDCG@10}} \sum_{i=1}^{10} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)}$$

- **Recall@10**:

$$\text{Recall@10} = \frac{\text{\# relevant documents retrieved in top 10}}{\text{total relevant documents}}$$

4 Results

We evaluate WCAR on the SciFact dataset (Wadden et al., 2020) from the BEIR benchmark (Thakur et al., 2021) as the primary testbed, and additionally on FiQA (Maia et al., 2018), NFCorpus (Boteva et al., 2016), and HotpotQA (Yang et al., 2018) to assess generalization. All our experiments use the lightweight `all-MiniLM-L6-v2` embedding model (Reimers and Gurevych, 2019).

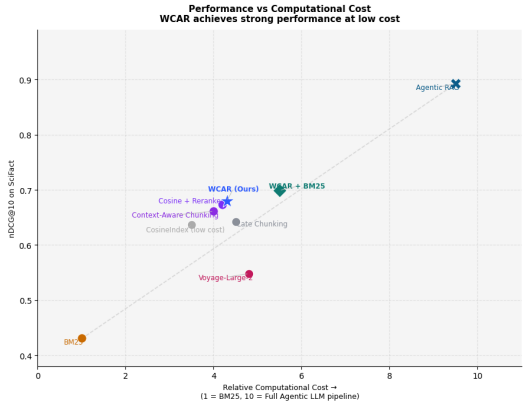

Figure 1: nDCG@10 vs. relative computational cost on SciFact. WCAR achieves strong retrieval performance without requiring LLM calls or fine-tuned models.

4.1 Main Results on SciFact

Table 2 presents the full comparison of WCAR against retrieval baselines on SciFact. WCAR achieves MRR@10 of 0.6445, NDCG@10 of 0.6824, and Recall@10 of 0.8196 using the same lightweight embedding model as Vanilla RAG. It achieves a +5.8% improvement in NDCG@10 and +4.0% improvement in Recall@10 over the cosine-only baseline at a relative computational cost of 3.

WCAR achieves strong retrieval performance even when compared to approaches that leverage substantially heavier infrastructure such as Voyage-Large-2 (Voyage AI, 2023) (cost 6), Late Chunking (Günther et al., 2024) (cost 5), and Context-Aware Chunking (Anthropic, 2024) (cost 4), while operating within a significantly lower computational budget. This highlights WCAR's potential as a practical, lightweight enhancement that integrates seamlessly into any existing cosine-based retrieval pipeline without requiring architectural modifications.

4.2 Performance vs. Computational Cost

Figure 1 plots nDCG@10 against relative computational cost. State-of-the-art retrieval models consistently achieve superior performance at the expense of substantial computational overhead. In contrast, WCAR occupies a favorable region by delivering retrieval quality approaching that of more expensive systems while remaining firmly within the low-cost tier. The accumulation step introduces only $\mathcal{O}(n)$ overhead per query using lightweight cosine similarity operations, with no additional model calls, retraining, or indexing changes required.

Table 2: Performance comparison on SciFact (BEIR Benchmark (Thakur et al., 2021)). Best results in bold; second best underlined.

Model / Method	MRR@10	NDCG@10	Recall@10	Rel. Cost	Models Required
BM25 (Robertson and Zaragoza, 2009) (avg BEIR)	—	0.434	—	1	No
Vanilla RAG (Cosine) (Reimers and Gurevych, 2019)	0.5979	0.6402	0.7867	2	Basic embedding model
Voyage-Large-2 (Voyage AI, 2023) (dense)	—	0.548	—	6	voyage-large-2
Late Chunking (Günther et al., 2024)	—	0.661	—	5	Large embedding model
Context-Aware Chunking (Anthropic, 2024)	—	0.661	—	4	Large language model + Basic embedding model
LlamaIndex RAG (Liu, 2023)	0.6053	0.6497	0.7867	3	Basic embedding model
Cosine + Reranker (Nogueira and Cho, 2019)	0.6354	0.6730	0.8112	4	Basic embedding model
WCAR (Ours)	0.6418	0.6803	0.8196	**3**	Basic embedding model
WCAR + Improved Scoring (Ours)	**0.6445**	**0.6824**	**0.8196**	**3**	Basic embedding model
WCAR + BM25 (Ours)	**0.6646**	**0.6990**	**0.8372**	4	Basic embedding model
Agentic RAG (SOTA) (Zhangdutao and Liao, 2026)	—	0.8924	—	10	Large language model + Basic embedding model

Table 3: Detailed WCAR results across all cutoffs on SciFact (Wadden et al., 2020) (BEIR (Thakur et al., 2021)).

Metric	@Cutoff	Score
NDCG	@1	0.54667
	@3	0.62892
	@5	0.65296
	@10	**0.68028**
Recall	@1	0.52567
	@3	0.68272
	@5	0.74183
	@10	**0.81956**

Table 4: WCAR + BM25 (Robertson and Zaragoza, 2009) hybrid results on SciFact (Wadden et al., 2020).

Metric	Score
MRR@10	0.6646
NDCG@1	0.5767
NDCG@3	0.6465
NDCG@5	0.6746
NDCG@10	**0.6990**
Recall@10	**0.8372**
Precision@10	0.0940

Table 5: WCAR vs. chunking methods (nDCG@10, SciFact (Wadden et al., 2020)).

Method	nDCG@10	Gain vs. Naive
Naive Chunking (Reimers and Gurevych, 2019)	0.6420	—
Semantic Chunking (Qu et al., 2024)	0.648–0.656	+1–3%
Context-Aware (Anthropic, 2024)	0.6610	+3.0%
Late Chunking (Günther et al., 2024)	0.6610	+3.0%
WCAR (Ours)	**0.6803**	**+5.8%**
WCAR + BM25 (Ours)	**0.6990**	**+7.33%**

4.3 Detailed Performance Across Cutoffs

Table 3 reports WCAR performance across multiple retrieval cutoffs. NDCG and Recall show steady improvement as the cutoff increases, reflecting the benefit of contextual accumulation. Table 1 shows how the adaptive threshold tightens progressively as chunks accumulate, managing the precision-recall trade-off.

4.4 Effect of BM25 Hybridization

Combining WCAR with BM25 (Robertson and Zaragoza, 2009) yields the strongest overall results (Table 4): NDCG@10 of 0.6990 and Recall@10 of 0.8372, a +7.33% and +6.0% respective gain over naive chunking. Lexical matching via BM25 captures exact term overlap that dense embeddings may underweight, while WCAR recovers semantically related content that BM25 would miss. The combination requires no additional training or architectural modifications.

4.5 Comparison with Chunking Strategies

Table 5 compares WCAR against chunking methods on nDCG@10. Context-Aware (Anthropic, 2024) and Late Chunking (Günther et al., 2024) both achieve +3.0% over naive chunking by modifying how documents are segmented or encoded. WCAR achieves +5.8% without modifying the chunking strategy or encoder at all operating entirely at retrieval time over the same chunk embeddings any standard pipeline would produce. This is a key distinction: WCAR improves retrieval by changing *"how embeddings are used*, not *how they are produced*".

4.6 Three-Metric Summary

Figure 2 and Table 2 show WCAR across all three metrics. The gains in Recall@10 are particularly significant: moving from 0.7867 to 0.8196 means WCAR surfaces contextually relevant documents that independent similarity scoring discards, directly validating the core motivation of this work.

4.7 Generalization Across Datasets

Table 6 presents results on three additional datasets. On HotpotQA (Yang et al., 2018), a multi-hop reasoning benchmark, WCAR achieves its strongest results (NDCG@10: 0.8392, Recall@10: 0.8076),

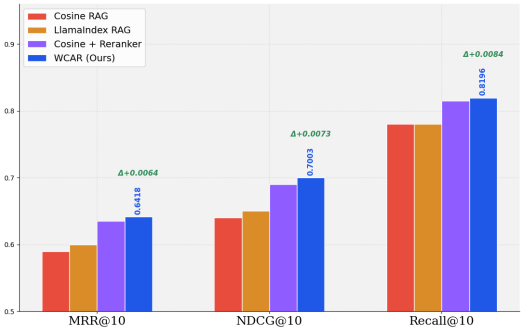

Figure 2: MRR@10, NDCG@10, and Recall@10 comparison across all measured systems on SciFact (Wadden et al., 2020).

Table 6: WCAR performance across multiple BEIR (Thakur et al., 2021) datasets.

Dataset	MRR@10	NDCG@10	Recall@10
NFCorpus (Boteva et al., 2016)	0.5361	0.3518	0.1726
FiQA (Maia et al., 2018)	0.4578	0.3870	0.4612
HotpotQA (Yang et al., 2018)	0.9735	0.8392	0.8076

confirming that contextual accumulation is most beneficial when answers require connecting information across multiple document segments. On FiQA (Maia et al., 2018), WCAR achieves competitive performance in moderately structured financial retrieval. On NFCorpus (Boteva et al., 2016), performance is lower, consistent with the heterogeneous and noisy nature of that corpus where inter-chunk contextual structure is weak. Collectively, these results confirm that WCAR's effectiveness scales with the degree of contextual structure in the retrieval corpus.

4.8 Comparison with Strong Retrieval Baselines

To better contextualize the performance of WCAR, we compare it against a range of strong lexical and dense retrieval baselines commonly reported in the BEIR benchmark (Thakur et al., 2021), including BM25 (Robertson and Zaragoza, 2009), Contriever, ColBERTv2 (Khattab and Zaharia, 2020), and SPLADE v2. These methods represent competitive retrieval paradigms spanning sparse, dense, and late-interaction approaches.

Table 7 summarizes the comparison across multiple datasets. Reported baseline values are approximate ranges based on prior literature, while WCAR results correspond to our experimental setup using the `all-MiniLM-L6-v2` embedding model.

SciFact. On SciFact (Wadden et al., 2020), WCAR achieves **MRR@10: 0.6418**, **NDCG@10: 0.6803**, and **Recall@10: 0.8196** using a lightweight embedding model. These results demonstrate that progressive context accumulation can meaningfully enhance retrieval quality without reliance on large-scale encoders or complex architectures, affirming that effective contextual reasoning can be achieved through efficient, model-agnostic retrieval mechanisms.

Table 7: Comparison with strong retrieval baselines (NDCG@10).

Method	SciFact	FiQA	HotpotQA	NFCorpus
BM25	∼0.43	∼0.30	∼0.62	∼0.33
Contriever	∼0.69	∼0.40	∼0.82	∼0.36
ColBERTv2	∼0.73	∼0.43	∼0.85	∼0.39
SPLADE v2	∼0.74	∼0.44	∼0.84	∼0.40
WCAR (Ours)	**0.6803**	**0.3870**	**0.8392**	**0.3518**

FiQA. On FiQA (Maia et al., 2018), WCAR achieves **MRR@10: 0.4578**, **NDCG@10: 0.3870**, and **Recall@10: 0.4612**. Performance is comparable to dense retrieval approaches such as Contriever, while remaining below late-interaction and sparse expansion methods. This behavior is expected, as financial queries often exhibit lower inter-chunk contextual dependency.

HotpotQA. On the multi-hop dataset HotpotQA (Yang et al., 2018), WCAR achieves **MRR@10: 0.9735**, **NDCG@10: 0.8392**, and **Recall@10: 0.8076**. This represents the strongest performance among the evaluated datasets, reaching levels comparable to advance baseline retrieval models such as ColBERTv2. The improvement highlights the effectiveness of context accumulation in multi-hop scenarios, where relevant information is distributed across multiple interconnected chunks. Results are reported on a filtered subset due to computational constraints.

NFCorpus. On NFCorpus (Boteva et al., 2016), WCAR achieves **MRR@10: 0.5361**, **NDCG@10: 0.3518**, and **Recall@10: 0.1726**. Performance is comparable to dense retrieval methods such as Contriever. The relatively smaller gains on this dataset are consistent with its weaker inter-chunk structure, where contextual dependencies are less pronounced.

Discussion. Overall, WCAR demonstrates competitive performance with strong dense retrievers across multiple datasets, despite operating as a lightweight, model-agnostic retrieval layer. The largest gains are observed in datasets with

stronger contextual dependencies (e.g., HotpotQA), while performance remains stable and competitive in more heterogeneous corpora (e.g., NFCorpus). These results support the central hypothesis of this work: that context accumulation in embedding space can complement traditional similarity-based retrieval without requiring complex architectures or additional training.

5 Discussion

5.1 Why WCAR Works

Standard retrieval assumes that semantic similarity and contextual relevance are equivalent. In practice, this assumption often fails: a chunk B that directly answers a query may score low in cosine similarity due to lexical mismatch, while another chunk A scores high simply by sharing surface-level vocabulary. WCAR addresses this by introducing *contextual propagation* — if a retrieved chunk A is strongly related to chunk B, then A acts as a contextual bridge, allowing B to be retrieved even when $\cos(B, Q)$ is low. This shifts retrieval from independent chunk scoring to a context-aware accumulation process, recovering information that isolated similarity evaluation would discard.

5.2 Interaction with Query Expansion

Incorporating a query expansion module leads to a slight degradation in precision (approximately 2% in our experiments). WCAR already performs implicit context expansion through accumulation across related chunks; external query expansion introduces redundant or noisy signals that overlap with what accumulation already captures. Designing expansion strategies that are aware of downstream accumulation remains an open direction for future work.

5.3 Interaction with Reranking

Because WCAR improves recall by surfacing contextually relevant but semantically distant chunks, the retrieved set may include borderline candidates. A lightweight reranking stage (Nogueira and Cho, 2019) can refine ordering as a post-processing step. In our experiments, reranking yields only marginal precision gains; the primary performance improvements arise from the contextual accumulation mechanism itself.

6 Conclusion

We presented **WCAR (Weighted Contextual Accumulative Retrieval)**, a lightweight retrieval framework that extends cosine-based retrieval by incorporating contextual relationships directly within the embedding space. WCAR is developed through a progressive formulation. Sequential Addition Retrieval (SAR) introduces context accumulation, Weighted Contextual Sequential Retrieval (WCSR) refines it through similarity-aware weighting, and WCAR unifies these ideas into an efficient $O(n)$ update rule. This formulation transforms retrieval from an independent similarity evaluation process into a *context-aware accumulation mechanism*, where relevance is iteratively propagated and refined.

Across multiple datasets from the BEIR benchmark (Thakur et al., 2021), including SciFact (Wadden et al., 2020), NFCorpus (Boteva et al., 2016), and FiQA (Maia et al., 2018), as well as a refined version of HotpotQA (Yang et al., 2018), WCAR demonstrates consistent improvements in ranking quality over cosine-based retrieval. On SciFact, WCAR achieves up to **+5.8%** improvement in NDCG@10, with further gains when combined with BM25 (Robertson and Zaragoza, 2009). The strongest improvements are observed in settings where information is distributed across multiple segments, highlighting the role of contextual accumulation in such scenarios. These gains are achieved without additional model training, LLM calls, or changes to the indexing pipeline. WCAR operates entirely at retrieval time as a model-agnostic layer, making it directly applicable to existing systems and suitable for cost-sensitive or latency-constrained environments.

Limitations

WCAR's performance gains are most pronounced in datasets with strong inter-chunk contextual structure (e.g., HotpotQA) and more modest in heterogeneous corpora (e.g., NFCorpus). Results on HotpotQA are reported on a filtered subset due to computational constraints, and comprehensive benchmarking across the full BEIR suite remains future work. Additionally, the sequential nature of context accumulation may introduce sensitivity to chunk ordering.

Acknowledgments

The authors thank the maintainers of the BEIR benchmark (Thakur et al., 2021) and the Sentence Transformers library (Reimers and Gurevych, 2019) for providing the evaluation infrastructure and embedding models used in this work.

References

Anthropic. 2024. Introducing contextual retrieval.

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *Advances in Information Retrieval: 38th European Conference on IR Research*. Springer.

Michael Günther, Georgios Mastrapas, Isabella Mohr, Bo Wang, and Saba Sturua. 2024. Late chunking: Contextual chunk embeddings using long-context embedding models. *arXiv preprint arXiv:2409.04701*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48. ACM.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Jerry Liu. 2023. LlamaIndex: Data framework for LLM applications.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. WWW'18 open challenge: Financial opinion mining and question answering. In *Companion Proceedings of The Web Conference 2018*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. In *arXiv preprint arXiv:1901.04085*.

Renyi Qu, Ruixuan Tu, and Forrest Bao. 2024. Is semantic chunking worth the computational cost? *arXiv preprint arXiv:2410.13070*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Voyage AI. 2023. Voyage large 2: Superior embedding model for retrieval.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Zhangdutao and Tian Liao. 2026. An agent-oriented pluggable experience-rag skill for experience-driven retrieval strategy orchestration. *arXiv preprint arXiv:2605.03989*.