Temporal-Aware Search Strategies for Personal Knowledge Bases: A Comparative Study Using USearch

T117 Research Experiment

January 2026

Abstract

Personal knowledge bases (PKBs) accumulate timestamped documents that users query with both semantic and temporal intent. Standard vector similarity search ignores temporal context, degrading performance on recency-dependent queries. We compare three temporal-aware search strategies—time-decay scoring, temporal feature injection, and hybrid date-range pre-filtering—against a pure cosine similarity baseline, using USearch HNSW indices over an 810-document corpus. Our results show that hybrid pre-filtering achieves the best overall nDCG@10 (0.769 vs 0.629 baseline), with the largest gains on temporal queries (+64%). Time-decay scoring with $\lambda = 0.02$ offers the best single-parameter trade-off. All approaches maintain sub-3ms latency.

1 Introduction

Personal knowledge bases—daily logs, meeting notes, research entries—are inherently temporal. Users frequently query with time-dependent intent: "What did I work on last week?" or "Recent decisions about project X." Standard dense retrieval using cosine similarity over learned embeddings treats all documents equally regardless of age, producing suboptimal results for such queries.

We evaluate three approaches to inject temporal awareness into vector search built on USearch's HNSW implementation, comparing their effectiveness on temporal vs. non-temporal queries while monitoring computational overhead.

2 Methods

2.1 Corpus and Queries

We constructed a corpus of 810 documents: 10 real daily memory logs and 800 synthetic documents spanning 200 days across 10 topic categories. Documents were embedded using TF-IDF+SVD (256 dimensions) and indexed with USearch HNSW (cosine metric).

We created 50 benchmark queries: 25 temporal-sensitive (referencing time windows) and 25 temporal-neutral (topical/factual). Ground truth relevance (0–3 scale) combines topic overlap with temporal window matching.

2.2 Approaches

Baseline. Pure cosine similarity search via USearch.

Time-Decay Scoring. Re-rank results using:

$$\text{score} = \text{cosine\_sim} \times e^{-\lambda \cdot \text{age\_days}} \tag{1}$$

We sweep $\lambda \in \{0.001, 0.005, 0.01, 0.02, 0.05, 0.1\}$.

Temporal Feature Injection. Append 5 normalized temporal features (recency score, day-of-week sin/cos, month sin/cos) to each embedding vector, creating 261-dimensional vectors. Query embeddings are augmented with target temporal features based on detected time references.

Hybrid Pre-Filter. For temporal queries: detect time reference, filter documents to the relevant date range (with margin), then cosine search within the filtered set. For neutral queries: standard cosine search.

3 Results

Table 1: Overall performance comparison

Approach	nDCG@10	MRR	Recall@10	Latency (ms)
Baseline	0.629	0.910	0.031	0.26
Time Decay (λ=0.01)	0.747	0.952	0.034	2.27
Temporal Injection	0.727	0.973	0.034	0.67
Hybrid Pre-filter	**0.769**	**1.000**	**0.036**	0.69

Table 2: nDCG@10 by query type

Approach	Temporal	Neutral
Baseline	0.439	0.818
Time Decay (λ=0.01)	0.671	0.824
Temporal Injection	0.649	0.805
Hybrid Pre-filter	**0.720**	**0.818**

3.1 Lambda Sweep

The optimal λ depends on query mix: $\lambda = 0.02$ maximizes overall nDCG@10 (0.759), while $\lambda = 0.1$ maximizes temporal query performance (0.717) at the cost of neutral query degradation (0.770 vs 0.818 baseline).

Table 3: Time-decay λ sweep

λ	Overall	Temporal	Neutral
0.001	0.683	0.557	0.809
0.005	0.722	0.628	0.816
0.01	0.747	0.671	0.824
0.02	**0.759**	0.692	0.826
0.05	0.753	0.712	0.793
0.1	0.743	**0.717**	0.770

4 Discussion

Hybrid pre-filtering is the clear winner for mixed workloads. It achieves the highest overall nDCG@10 (0.769) by selectively applying temporal filtering only when temporal intent

is detected, leaving neutral queries unaffected. Its perfect MRR (1.000) indicates the most relevant document always appears first.

Time-decay scoring offers a simpler implementation with good results. The λ parameter provides a tunable knob, but the optimal value depends on the query distribution. The re-ranking step adds $\sim$2ms latency.

Temporal feature injection is conceptually elegant but underperforms both alternatives. Modifying the embedding space affects all queries, slightly degrading neutral query performance (0.805 vs 0.818 baseline). However, it requires no query intent classification.

Key trade-off: Hybrid requires a temporal intent classifier (even a simple keyword-based one suffices), while time-decay and injection are classifier-free.

5 Implications for QMD Backend

For the USearch-based QMD backend (T92):

1. **Default**: Use hybrid pre-filter with a lightweight temporal intent detector
2. **Fallback**: Time-decay with $\lambda = 0.02$ when intent detection is unavailable
3. **Index overhead**: Temporal injection adds only 5 dimensions (2% increase for 256d embeddings) but provides persistent temporal awareness without runtime overhead

6 Conclusion

Temporal awareness significantly improves search quality for personal knowledge bases. The hybrid pre-filter approach improves nDCG@10 by 22% overall and 64% on temporal queries compared to pure cosine similarity, with negligible latency impact. We recommend hybrid pre-filtering as the default strategy for the QMD backend.