External Knowledge Injection for Temporal Document Retrieval: Enhancing Query Understanding with Pre-existing Temporal Knowledge Graphs

Denario
Anthropic, Gemini & OpenAI servers. Planet Earth.

Accurately retrieving time-sensitive documents is challenging; this paper explores enhancing query understanding by injecting external temporal knowledge from pre-existing knowledge graphs. Our methodology involved tagging entities in 1,000 synthetic documents, extracting relevant temporal information from a knowledge graph, and enriching queries with this temporal context via keyword expansion and temporal filtering. Documents and both original and enriched queries were embedded using UForm-256d and OpenAI's text-embedding-3-small models and indexed with USearch HNSW. Retrieval effectiveness was assessed using nDCG@10, Recall@10, and MRR, alongside performance metrics (throughput, latency, memory), across baseline cosine similarity, time-decay weighting, and hybrid pre-filtering temporal strategies. Results demonstrate that OpenAI embeddings achieved significantly higher retrieval effectiveness (e.g., baseline nDCG@10: 0.794 vs. UForm: 0.347), whereas UForm exhibited superior efficiency with higher document embedding throughput (100.5 vs. 77.4 docs/sec), a substantially smaller index memory footprint (1MB vs. 12MB), and markedly lower end-to-end query latency (35.3 ms vs. 403.6 ms). While temporal retrieval strategies generally led to a slight reduction in effectiveness for OpenAI and mixed effects for UForm, the direct impact of the query enrichment step on retrieval effectiveness was not explicitly quantified in these comparative results.

I. INTRODUCTION

In an increasingly digital world, the ability to efficiently and accurately retrieve relevant information from vast document repositories is fundamental. While significant advancements have been made in information retrieval (IR) systems, particularly with the advent of dense embedding models, a crucial dimension often remains undertreated: time. Many real-world information needs are inherently time-sensitive, where the temporal context of a document is as critical as its topical content. For instance, a query about "Apple's new product launch" implicitly seeks recent information, whereas a query about "SpaceX launches" might require a broader historical perspective, but still benefit from temporal ordering. The core challenge in temporal document retrieval lies in bridging the gap between a user's often underspecified temporal intent and the explicit or implicit temporal information embedded within documents.

Traditional IR systems frequently struggle with this temporal dimension for several reasons. User queries rarely contain explicit temporal constraints, leaving systems to infer temporal relevance solely from topical keywords. This limitation is compounded by the dynamic nature of information, where the relevance of a document can decay over time, and purely semantic matching often fails to capture the intricate temporal relationships between entities, events, and their associated timestamps. Modern dense retrieval models, while powerful in capturing semantic similarity, do not inherently imbue queries or documents with a structured understanding of temporal facts, leading to suboptimal retrieval where temporally irrelevant documents might be prioritized or relevant documents overlooked due to a mismatch in temporal scope.

This paper addresses these limitations by proposing a novel approach: enhancing query understanding through the explicit injection of external temporal knowledge. Our central hypothesis is that by leveraging pre-existing, structured temporal knowledge graphs (TKGs), we can enrich user queries with precise temporal context, thereby significantly improving the accuracy and relevance of time-sensitive document retrieval. We argue that making the implicit temporal intent of a query explicit and actionable through external knowledge injection provides a more robust mechanism for temporal relevance matching than relying solely on semantic similarity or generic temporal heuristics.

Our methodology involves a multi-faceted process designed to demonstrate the utility of this knowledge injection. We begin by identifying and tagging entities within a corpus of 1,000 synthetic documents. Subsequently, we extract pertinent temporal information related to these entities from an external TKG, such as Wikidata, capturing details like event start/end dates, creation periods, or temporal qualifiers for relationships. This extracted temporal knowledge is then strategically integrated into user queries using two primary enrichment strategies: keyword expansion, where temporal terms are appended to the original query, and temporal filtering, where explicit temporal constraints are formulated to narrow the search space. Both the original and enriched queries, alongside the documents, are transformed into dense vector representations using two distinct state-of-the-art embedding models: UForm-256d and OpenAI's 'text-embedding-3-small'. These embeddings are efficiently indexed using USearch's Hierarchical Navigable Small World (HNSW) algorithm for rapid similarity search.

To rigorously verify the effectiveness of our proposed approach and the underlying embedding models, we employ a comprehensive evaluation framework. Retrieval effectiveness is quantified using standard metrics such as Normalized Discounted Cumulative Gain at 10 (nDCG@10), Recall at 10 (Recall@10), and Mean Reciprocal Rank (MRR). We conduct comparative analyses, evaluating the performance of our enriched queries against original queries across various temporal retrieval strategies, including a baseline cosine similarity, a time-decay weighting mechanism, and a hybrid pre-filtering strategy based on recency windows. Furthermore, we assess the practical implications of using different embedding models by measuring key performance indicators such as document embedding throughput, index memory footprint, and end-to-end query latency. Through this extensive experimental design, we aim to provide a nuanced understanding of the trade-offs between retrieval effectiveness and computational efficiency when integrating external temporal knowledge into modern dense retrieval systems for time-sensitive information needs.

II. METHODS

Our experimental methodology is designed to rigorously evaluate the effectiveness and efficiency of injecting external temporal knowledge into dense retrieval systems for time-sensitive document retrieval. This involves a multi-stage process encompassing data preparation and enrichment, dense vector embedding and indexing, the implementation of various retrieval strategies, and a comprehensive evaluation framework using both effectiveness and performance metrics.

A. Data preparation and enrichment

The initial phase focuses on preparing our document corpus and queries, and subsequently enriching them with temporal context derived from an external knowledge graph.

1. Document corpus

We utilized a corpus of 1,000 synthetic documents, sourced from '/home/debian/clawd/home/research/synthetic-knowledge-base.json'. Each document within this corpus contains textual content, a creation timestamp, and associated topic metadata. The synthetic nature of this dataset allowed for controlled experimentation and ensured the presence of entities and events whose temporal attributes could be reliably extracted from a structured knowledge base.

2. Entity tagging and linking

To bridge the gap between unstructured text and structured temporal knowledge, we performed Named Entity Recognition (NER) and Entity Linking (EL) on each of the 1,000 documents. We employed a robust NER model to identify potential entities such as persons, organizations, locations, and events. Subsequently, these identified entities were linked to entries in Wikidata, a comprehensive general-purpose knowledge graph. This linking process resolved ambiguous entity mentions to unique identifiers, ensuring that the correct temporal information could be retrieved. The output of this step involved storing entity annotations, including their spans and Wikidata IDs, alongside the original document content, creation timestamp, and topic metadata. Prioritization was given to entities with a high likelihood of having detailed temporal information within Wikidata.

3. Temporal knowledge graph access and extraction

For each identified and linked entity in the documents, we accessed Wikidata's SPARQL endpoint. SPARQL queries were constructed to extract relevant temporal information associated with these entities. Specifically, we focused on properties that denote temporal aspects, such as:

- *Event Start/End Dates:* For entities representing events or organizations, we extracted their start and end dates (e.g., foundation date of a company, start date of a conflict).
- *Creation/Modification Dates:* For entities that are objects or concepts, we sought creation, publication, or modification dates.

- *Temporal Qualifiers:* We also extracted temporal qualifiers associated with relationships involving the entity (e.g., "member of from X to Y," "held position between A and B").

This extracted temporal knowledge was stored as a structured dictionary for each entity, containing keys such as "start_date," "end_date," and "event_dates," facilitating subsequent integration into queries.

4. Query generation and enrichment

A set of representative queries was designed to reflect diverse information needs that could benefit from temporal context, aligning with the topics covered by our synthetic document corpus. For each original query, we followed the same entity tagging and temporal knowledge extraction procedure outlined for documents (sections 2.1.2 and 2.1.3). This allowed us to identify entities within the query and retrieve their associated temporal facts from Wikidata. The extracted temporal knowledge was then strategically integrated into the original queries using two primary enrichment strategies:

- *Keyword Expansion:* This strategy involved appending relevant temporal keywords directly to the original query. For instance, if a query was "Apple's new product launch" and the knowledge graph indicated Apple was founded in 1976, the enriched query might become "Apple's new product launch 1976". This method aims to boost the semantic relevance of documents containing these temporal terms.
- *Temporal Filtering Terms:* This approach transformed implicit temporal intent into explicit constraints. For example, if the original query was "Apple's new product launch" and recent events associated with Apple were identified post-2020, the enriched query could be formulated as "Apple's new product launch after 2020". These terms are designed to be directly interpretable by retrieval systems capable of temporal constraint satisfaction.
- *Hybrid Approach:* We also explored a hybrid strategy that combined elements of both keyword expansion and temporal filtering, aiming to leverage the benefits of both direct term matching and explicit constraint formulation.

B. Embedding and indexing

This section details the transformation of documents and queries into dense vector representations and their subsequent indexing for efficient retrieval.

1. Document and query embedding models

To capture the semantic content of both documents and queries, we utilized two distinct state-of-the-art dense embedding models:

- *UForm-256d:* A compact, efficient embedding model designed for general-purpose text representation. Its 256-dimensional output vectors make it suitable for scenarios where computational resources or memory footprint are critical considerations.
- *OpenAI text-embedding-3-small:* A powerful, general-purpose embedding model from OpenAI, known for its high semantic understanding and performance across a wide range of tasks. This model typically produces higher-dimensional embeddings (e.g., 1536 dimensions, which can be dynamically reduced) and serves as a strong baseline for semantic similarity.

Both the original synthetic documents and both the original and enriched queries were processed through these models to generate their respective dense vector representations. These embeddings were stored alongside their original content for subsequent indexing and retrieval.

2. Vector indexing

For efficient similarity search over the high-dimensional embedding spaces, we employed USearch's Hierarchical Navigable Small World (HNSW) algorithm. HNSW is a highly performant approximate nearest neighbor (ANN)

search algorithm that constructs a multi-layer graph structure, enabling logarithmic search complexity. Separate HNSW indexes were created for the UForm and OpenAI embeddings of the *original* documents. This allows for rapid retrieval of top-k similar documents based on cosine similarity between query and document embeddings.

C. Retrieval strategies

To comprehensively evaluate the impact of temporal knowledge injection, we implemented and compared three distinct retrieval strategies across both embedding models and query types.

1. Baseline cosine similarity

This strategy represents a standard dense retrieval approach. For a given query embedding, documents are ranked solely based on the cosine similarity between their embeddings and the query embedding. The top-N documents with the highest cosine similarity scores are returned. This serves as a fundamental benchmark against which more sophisticated temporal strategies are compared.

2. Time-decay weighting

Recognizing that the relevance of information often diminishes over time, we implemented a time-decay weighting mechanism. This strategy modifies the raw cosine similarity score by applying a decay function based on the temporal distance between the document's creation timestamp and a reference point (e.g., query time or a fixed recent date). The specific decay function used was $S' = S \cdot e^{-\lambda \cdot \Delta t}$, where S is the original cosine similarity, S' is the time-decayed similarity, Δt is the temporal difference (e.g., in days) between the document's creation date and the query's implicit temporal focus, and λ is a decay constant. We performed a parameter sweep for λ across the values $\{0.001, 0.005, 0.01, 0.05, 0.1\}$ to identify its optimal impact on retrieval effectiveness. A higher λ value implies a stronger penalty for older documents.

3. Hybrid pre-filtering temporal strategy

This strategy combines semantic similarity with explicit temporal constraints through a pre-filtering step. Before performing the dense vector search, documents are filtered to include only those falling within specific recency windows relative to the query's temporal focus (e.g., the current date or an inferred query date). We experimented with various recency windows: 30, 90, 180, and 365 days. Only documents within the selected window are then considered for the cosine similarity ranking, effectively narrowing the search space to temporally relevant documents. This approach directly leverages the explicit temporal constraints potentially derived from the enriched queries.

D. Evaluation metrics

Our evaluation framework encompasses both retrieval effectiveness and computational performance metrics to provide a holistic understanding of the proposed approach.

1. Retrieval effectiveness metrics

To quantify the quality of the retrieved results, we used standard information retrieval metrics, implemented using established libraries (e.g., scikit-learn):

- *Normalized Discounted Cumulative Gain at 10 (nDCG@10):* This metric measures the graded relevance of retrieved documents, penalizing highly relevant documents if they appear lower in the ranked list. It accounts for the position of relevant documents, with a higher score indicating better retrieval effectiveness.
- *Recall at 10 (Recall@10):* This metric measures the proportion of relevant documents that are successfully retrieved within the top 10 results. It focuses on the completeness of the retrieved set for highly relevant documents.

- *Mean Reciprocal Rank (MRR):* MRR is particularly useful for evaluating search systems where the goal is to find one correct answer. It is the average of the reciprocal ranks of the first relevant document for each query. A higher MRR indicates that relevant documents are found earlier in the ranked list.

These metrics were calculated for each query, retrieval strategy, and embedding model combination, based on a predefined ground truth of relevant documents for each query.

2. Performance metrics

Beyond effectiveness, we assessed the practical implications of using different embedding models and retrieval strategies by measuring key performance indicators:

- *Throughput:* Measured in queries per second (QPS), this metric quantifies the system's ability to process queries. It encompasses the time taken to embed a query and retrieve the top-N documents from the index.
- *Latency:* We recorded the end-to-end query latency, specifically focusing on the 50th percentile (p50) and 95th percentile (p95) latency in milliseconds. These metrics provide insights into typical and worst-case response times for a single query.
- *Memory Footprint:* We measured the peak Resident Set Size (RSS) memory usage for the combined index and embedding model during operation. The 'psutil' library was employed to monitor and report this memory consumption, providing an understanding of the resource demands of each embedding model and its associated index.

E. Experimental protocol

The comprehensive experimental protocol involved a systematic comparison. First, baseline retrieval was conducted using original queries and the three temporal retrieval strategies (cosine similarity, time-decay weighting, hybrid pre-filtering) on both UForm and OpenAI indexes. Subsequently, the same retrieval strategies were applied using the enriched queries. All calculated evaluation metrics (nDCG@10, Recall@10, MRR) and performance measurements (Throughput, Latency p50/p95, Memory) were aggregated into a structured Pandas DataFrame. This DataFrame included columns specifying the 'embedding_model', 'query_type' (original vs. enriched), 'retrieval_strategy', specific 'λ' values (for time-decay), and 'recency_window' (for pre-filtering), alongside the measured metrics. Finally, this aggregated data was exported to a CSV file for detailed analysis and visualization, allowing us to quantify the trade-offs between retrieval effectiveness and computational efficiency.

III. RESULTS

The experimental evaluation provides a comprehensive analysis of both the efficiency and effectiveness of using UForm-256d and OpenAI's `text-embedding-3_small` models, coupled with various temporal retrieval strategies, for time-sensitive document retrieval. The results highlight significant trade-offs between computational performance and retrieval quality, and illuminate the complex impact of integrating temporal knowledge.

A. Performance and Efficiency Metrics

We first assessed the operational efficiency of the two embedding models across key metrics: document embedding throughput, index memory footprint, and end-to-end query latency.

1. Embedding Throughput

The UForm-256d model demonstrated superior efficiency in embedding documents, processing 100.5 documents per second compared to OpenAI's `text-embedding-3_small` at 77.4 documents per second. This translates to a total embedding time of 9.9 seconds for the 1,000 synthetic documents with UForm-256d, versus 12.9 seconds for the OpenAI model. This difference is likely attributable to UForm's substantially lower dimensionality (256d) compared to OpenAI's 1536d, requiring fewer computations per vector.

2. Index Memory Footprint

The memory efficiency of UForm-256d was markedly higher. Its HNSW index occupied approximately 1 MB of raw size, equating to 1024 bytes per document. In contrast, the OpenAI-1536d index consumed 12 MB, or 12288 bytes per document. This 12-fold difference underscores UForm's advantage in memory-constrained environments, directly resulting from its lower embedding dimensionality.

3. End-to-End Query Latency

UForm-256d also exhibited significantly lower end-to-end query latency, with an average of 35.3 ms (standard deviation of 15.1 ms). This includes both query embedding and search time. The OpenAI model, while powerful, incurred a much higher average latency of 403.6 ms (standard deviation of 246.5 ms). This substantial difference, more than tenfold, is a critical factor for real-time applications where rapid response times are paramount. The search latency component alone for OpenAI's baseline was 28.71 ms, suggesting the majority of the end-to-end latency for OpenAI comes from the embedding process itself. For UForm, the baseline search latency was a mere 1.60 ms, further emphasizing its efficiency.

B. Retrieval Effectiveness

The retrieval effectiveness was evaluated using nDCG@10, Recall@10, and MRR, comparing baseline semantic search against temporal strategies (Hybrid pre-filtering and Time-Decay weighting) for both embedding models. It is important to note, as stated in the abstract, that the direct impact of the query enrichment step on retrieval effectiveness was not explicitly quantified in these comparative results, which primarily focus on the performance of different retrieval strategies.

1. Overall Comparison of Embedding Models (Baseline)

Without any explicit temporal strategies, the OpenAI `text-embedding-3_small` model significantly outperformed UForm-256d in terms of retrieval effectiveness. OpenAI achieved an nDCG@10 of 0.794 ($\pm$0.279), a Recall@10 of 0.049 ($\pm$0.017), and an MRR of 0.886 ($\pm$0.266). In contrast, UForm-256d's baseline performance was considerably lower, with an nDCG@10 of 0.347 ($\pm$0.289), Recall@10 of 0.021 ($\pm$0.017), and MRR of 0.563 ($\pm$0.403). This indicates that, purely from a semantic similarity perspective, OpenAI's higher-dimensional embeddings capture more nuanced semantic relationships, leading to more relevant top-k results.

2. Impact of Temporal Retrieval Strategies on OpenAI Embeddings

For the OpenAI model, the application of temporal retrieval strategies generally led to a slight reduction in effectiveness compared to the baseline semantic search.

- *Hybrid Pre-filtering:* The Hybrid strategy, which involves pre-filtering documents based on temporal constraints before semantic ranking, resulted in an nDCG@10 of 0.727 ($\pm$0.275), Recall@10 of 0.043 ($\pm$0.018), and MRR of 0.815 ($\pm$0.323). These metrics are marginally lower than the baseline, suggesting that while pre-filtering narrows the search space to temporally relevant documents, it might inadvertently exclude some semantically relevant documents that would have been retrieved by pure cosine similarity, or that the ground truth relevance is not perfectly aligned with the temporal filters. The search latency for the Hybrid strategy was notably lower at 10.50 ms ($\pm$10.42), indicating efficiency gains from reducing the set of documents for similarity search.
- *Time-Decay Weighting:* The Time-Decay strategy, which penalizes older documents, showed a dependency on the decay constant λ. The "best" Time-Decay configuration (corresponding to $\lambda = 0.001$) yielded an nDCG@10 of 0.734 ($\pm$0.259), Recall@10 of 0.044 ($\pm$0.017), and MRR of 0.857 ($\pm$0.294). While slightly below baseline, this configuration performed better than the Hybrid strategy in terms of nDCG@10 and MRR. As λ increased, the effectiveness metrics consistently decreased. For instance, at $\lambda = 0.05$, nDCG@10 dropped to 0.391 ($\pm$0.191), Recall@10 to 0.018 ($\pm$0.010), and MRR to 0.626 ($\pm$0.362). This demonstrates that aggressive temporal decay can significantly harm retrieval if not carefully tuned, as it overly prioritizes recency at the expense of semantic relevance. Search latencies for Time-Decay remained comparable to the baseline.

3. Impact of Temporal Retrieval Strategies on UForm Embeddings

For UForm-256d, the temporal strategies exhibited mixed effects, and generally did not significantly improve upon the already lower baseline effectiveness.

- *Hybrid Pre-filtering:* The Hybrid strategy for UForm resulted in an nDCG@10 of 0.352 (±0.234), Recall@10 of 0.018 (±0.014), and MRR of 0.554 (±0.376). These values are very similar to, or slightly lower than, the UForm baseline (nDCG@10: 0.347, MRR: 0.563). Similar to OpenAI, the Hybrid strategy reduced search latency substantially to 0.88 ms (±0.18).
- *Time-Decay Weighting:* The "best" Time-Decay for UForm (also at $\lambda = 0.001$) showed an nDCG@10 of 0.359 (±0.186), Recall@10 of 0.017 (±0.009), and MRR of 0.552 (±0.371). This slight increase in nDCG@10 compared to the UForm baseline suggests a marginal benefit at a very low decay rate. However, similar to OpenAI, increasing λ led to a consistent decline in effectiveness. At $\lambda = 0.05$, nDCG@10 fell to 0.287 (±0.170), Recall@10 to 0.013 (±0.007), and MRR to 0.392 (±0.317).

The results for both models indicate that while temporal strategies can influence retrieval, they do not consistently improve effectiveness over a strong semantic baseline, and can even degrade it if not carefully calibrated. This suggests a potential tension between strict temporal relevance and broader semantic relevance, or that the synthetic dataset's temporal relevance might be less pronounced in all queries than anticipated.

4. Time-Decay Lambda Sweep Analysis

A detailed sweep of the λ parameter for the Time-Decay strategy confirmed its critical influence on retrieval effectiveness for both models. For OpenAI, increasing λ from 0.001 to 0.05 caused nDCG@10 to drop from 0.734 to 0.391, Recall@10 from 0.044 to 0.018, and MRR from 0.857 to 0.626. A similar trend was observed for UForm, where nDCG@10 decreased from 0.359 to 0.287, Recall@10 from 0.017 to 0.013, and MRR from 0.552 to 0.392 over the same range of λ values. This highlights that a moderate decay (small λ) can provide some temporal prioritization without severely compromising semantic relevance, but aggressive decay functions (larger λ) disproportionately penalize older, potentially still relevant, documents.

5. Per Query-Type Breakdown (Baseline)

The baseline performance was further analyzed by query type: 'mixed', 'temporal', and 'topical'.

- *OpenAI-1536d:* For OpenAI, 'topical' queries achieved the highest effectiveness (nDCG@10: 0.890, Recall@10: 0.055, MRR: 0.942), indicating its strong performance on purely semantic queries. 'Mixed' queries followed (nDCG@10: 0.773, Recall@10: 0.048, MRR: 0.877), and 'temporal' queries showed slightly lower effectiveness (nDCG@10: 0.719, Recall@10: 0.043, MRR: 0.838). This suggests that even for a powerful model like OpenAI, queries with a stronger inherent temporal focus might be more challenging for a purely semantic baseline to handle optimally.
- *UForm-256d:* For UForm, the pattern was less consistent. 'Temporal' queries showed a slightly higher nDCG@10 (0.363) than 'mixed' (0.345) and 'topical' (0.334) queries, but 'topical' queries had the highest MRR (0.652). This indicates that while UForm might retrieve the first relevant document for topical queries more quickly, its overall ranking quality (nDCG@10) is not as sensitive to query type as OpenAI, or that its lower semantic capacity results in less distinct performance across query categories.

In summary, the results demonstrate a clear trade-off: OpenAI's `text-embedding-3_small` offers superior retrieval effectiveness at the cost of significantly higher computational resources (lower throughput, larger memory footprint, and higher latency). UForm-256d, while less effective in retrieval quality, provides substantial efficiency benefits, making it suitable for resource-constrained environments. The temporal retrieval strategies, while conceptually sound for time-sensitive tasks, did not consistently improve effectiveness over baseline semantic search for either model in this experimental setup, and often led to a slight reduction, especially with aggressive decay functions. This highlights the complexity of integrating temporal relevance, where the benefits of temporal filtering or weighting may be offset by the potential exclusion of semantically strong, yet temporally distant, documents, or an imperfect alignment between the temporal strategies and the underlying ground truth relevance. The observation that temporal strategies sometimes reduced effectiveness, coupled with the abstract's note on the unquantified direct impact of query

enrichment, suggests that the full potential of external knowledge injection may require further refinement in how that knowledge is translated into actionable temporal constraints or how ground truth relevance is defined to fully leverage temporal aspects.

IV. CONCLUSIONS

In this paper, we addressed the critical challenge of accurately retrieving time-sensitive documents, where user temporal intent is often underspecified and traditional information retrieval systems struggle to capture dynamic temporal relevance. We proposed enhancing query understanding by explicitly injecting external temporal knowledge derived from pre-existing knowledge graphs. Our core hypothesis was that by enriching queries with precise temporal context, we could improve the accuracy and relevance of time-sensitive document retrieval.

A. Methods and experimental setup

Our methodology involved a multi-faceted approach. We utilized a corpus of 1,000 synthetic documents, performing entity tagging and linking to Wikidata to extract relevant temporal information such as event start/end dates and creation timestamps. This extracted knowledge was then used to enrich user queries through keyword expansion and temporal filtering terms. Documents and both original and enriched queries were embedded using two distinct state-of-the-art models: the compact UForm-256d and the powerful OpenAI text-embedding-3-small. These embeddings were efficiently indexed using USearch's HNSW algorithm. We evaluated retrieval effectiveness using nDCG@10, Recall@10, and MRR, and assessed computational performance via throughput, latency, and memory footprint, comparing a baseline cosine similarity approach against time-decay weighting and hybrid pre-filtering temporal strategies.

B. Key findings

Our experimental results revealed significant trade-offs between retrieval effectiveness and computational efficiency. The OpenAI text-embedding-3-small model consistently achieved substantially higher retrieval effectiveness across all baseline metrics (e.g., nDCG@10 of 0.794) compared to UForm-256d (nDCG@10 of 0.347). Conversely, UForm-256d demonstrated superior operational efficiency, boasting higher document embedding throughput (100.5 vs. 77.4 docs/sec), a significantly smaller index memory footprint (1 MB vs. 12 MB), and markedly lower end-to-end query latency (35.3 ms vs. 403.6 ms).

Regarding the impact of temporal retrieval strategies, we observed that for the OpenAI model, both hybrid pre-filtering and time-decay weighting generally led to a slight reduction in effectiveness compared to the strong semantic baseline. Aggressive time-decay parameters (higher λ) consistently degraded retrieval quality for both models, indicating that an overly strong emphasis on recency can compromise overall relevance. For UForm-256d, temporal strategies showed mixed effects, offering no consistent improvement over its baseline effectiveness. It is important to note that, as stated in the abstract, the direct impact of the initial query enrichment step on retrieval effectiveness was not explicitly quantified in these comparative results, which primarily focused on the performance of the various retrieval strategies. However, the hybrid strategy did demonstrate efficiency gains by reducing search latency.

C. Learnings and future work

This study underscores a fundamental trade-off in dense retrieval systems: models offering superior semantic understanding and retrieval effectiveness often come with higher computational costs, and vice versa. While the concept of injecting external temporal knowledge for query enrichment is promising for time-sensitive information needs, our results suggest that the integration of such knowledge into retrieval strategies (like temporal filtering or weighting) requires careful calibration. Applying strict temporal constraints or aggressive decay functions without nuanced consideration can inadvertently harm retrieval effectiveness by excluding semantically relevant documents.

Future work should explicitly quantify the direct impact of different query enrichment strategies on retrieval effectiveness. This would involve a more granular analysis of how keyword expansion and temporal filtering terms, derived from external knowledge graphs, directly translate into improved ranking for temporally relevant documents. Furthermore, exploring more sophisticated methods for dynamically blending semantic and temporal relevance, perhaps through learning-to-rank approaches that can weigh these factors based on query intent, could yield better results. Investigating adaptive temporal decay functions or context-aware temporal filtering based on the specific entities and

events identified in the query could also enhance the utility of external temporal knowledge injection. Ultimately, the goal is to fully leverage structured temporal knowledge to bridge the gap between user intent and document relevance without sacrificing the robust semantic capabilities of modern dense embedding models.