Evaluating Local and Cloud Embedding Models for Contextual Information Retrieval in Task-Oriented Personal Knowledge Bases

Denario
Anthropic, Gemini & OpenAI servers. Planet Earth.

This paper addresses the challenge of context-aware information retrieval from personal knowledge bases (PKBs), moving beyond simple semantic similarity by integrating task-specific context. We evaluated four embedding models, including local models like `nomic-ai/nomic-embed-text-v1.5` and the cloud-based OpenAI `text-embedding-3-small`, for information retrieval from a synthetic PKB of 1000 documents using a USearch HNSW index. To simulate user tasks and establish ground truth, 60 diverse queries, each with three task descriptions, were generated, and an LLM-as-judge (Gemini 2.0 Flash) assessed the relevance of 1800 query-document pairs. Retrieval quality was measured using nDCG@10, Recall@10, and MRR, alongside efficiency metrics. Our results demonstrate that the best local model, `nomic-ai/nomic-embed-text-v1.5`, achieved retrieval quality statistically indistinguishable from OpenAI `text-embedding-3-small` based on overlapping 95% bootstrap confidence intervals, though absolute performance for all models remained modest (nDCG@10 around 0.2). All models struggled significantly with temporal queries, suggesting a limitation of pure dense embeddings in handling strict date constraints. While OpenAI offered a superior quality-latency trade-off, `nomic-ai/nomic-embed-text-v1.5` presents a viable local-only alternative, albeit with higher CPU embedding latency dominating the end-to-end retrieval time. These findings highlight the potential of local embedding models for contextual PKB retrieval but underscore the necessity of hybrid retrieval approaches to effectively address diverse information needs, particularly those involving temporal context.

I. INTRODUCTION

The digital age has ushered in an unprecedented era of information abundance, leading individuals to curate vast Personal Knowledge Bases (PKBs). These highly personalized repositories encompass a heterogeneous mix of notes, documents, emails, web clippings, and other digital artifacts, serving as a critical extension of an individual's memory and cognitive workspace. However, the sheer volume and unstructured nature of data within PKBs present a formidable challenge: effectively retrieving information that is not merely semantically related to a query, but critically, also contextually relevant to a user's current task or intent. Traditional information retrieval (IR) systems, often relying on simple keyword matching or even basic semantic similarity, frequently fall short in this domain. The inherent difficulty lies in discerning true utility; a document semantically similar to a search query might be entirely irrelevant if it does not align with the specific goal or task the user is attempting to accomplish. This fundamental gap between general semantic similarity and precise, task-specific contextual relevance represents a significant barrier to maximizing the utility of PKBs.

Modern information retrieval has increasingly leveraged dense vector embeddings to capture nuanced meanings and intricate relationships within textual data. These advanced models transform text into high-dimensional numerical vectors, allowing semantic similarity to be approximated by the proximity of their corresponding vectors in an embedding space. While highly effective for broad semantic search, their direct application to the specialized domain of task-oriented PKBs introduces complexities. The performance of these embedding models can vary substantially based on their architecture, training methodologies, and the specific characteristics of the data. Furthermore, practitioners face a crucial decision: whether to integrate powerful, proprietary cloud-based embedding services, which often offer superior performance but come with associated costs, privacy implications, and network latency, or to deploy open-source alternatives locally, providing greater autonomy and control over data and computational resources. Understanding this trade-off, particularly how different embedding models—both local and cloud-hosted—perform when tasked with retrieving information under explicit contextual constraints within a PKB, is paramount for developing effective retrieval solutions.

This paper addresses the aforementioned challenges by systematically evaluating the efficacy of several prominent embedding models for contextual information retrieval within task-oriented Personal Knowledge Bases. Our primary objective is to move beyond conventional semantic search and assess these models' ability to retrieve documents that are not only semantically aligned with a user's query but are also demonstrably relevant within the explicit context of a defined user task. We posit that while advanced embedding models are crucial, their true value in a PKB setting is realized only when task-specific context is effectively integrated into the retrieval process. To this end, we investigate how both local embedding models, such as `nomic-ai/nomic_embed_text_v1.5`, and cloud-based solutions, represented by OpenAI's `text-embedding-3-small`, perform under these stringent contextual requirements.

To rigorously verify the models' capabilities, we construct a synthetic Personal Knowledge Base comprising 1000

diverse documents, each with varying lengths and timestamps. We generate a comprehensive set of 60 distinct queries, categorized into semantic, temporal, and mixed types, and for each query, we devise three unique task descriptions to simulate varied user intentions and provide explicit retrieval context. The core of our evaluation methodology involves an LLM-as-judge paradigm, utilizing Gemini 2.0 Flash to establish ground truth relevance scores for 1800 query-document pairs. Crucially, this LLM-as-judge considers the provided task descriptions in its assessment, allowing us to capture graded, task-specific relevance beyond simplistic binary judgments. Retrieval quality is then quantified using established metrics such as Normalized Discounted Cumulative Gain at rank 10 (nDCG@10), Recall at rank 10 (Recall@10), and Mean Reciprocal Rank (MRR). Complementing this, we analyze efficiency metrics including embedding throughput, search latency, and end-to-end retrieval time to provide a holistic view of practical deployment considerations. Through this rigorous evaluation, employing statistical analysis with 95% bootstrap confidence intervals, we aim to provide a nuanced understanding of the capabilities and trade-offs associated with different embedding models in the specialized domain of contextual information retrieval for task-oriented personal knowledge bases.

II. METHODS

Our methodology is designed to systematically evaluate the performance of various local and cloud-based embedding models for contextual information retrieval within task-oriented Personal Knowledge Bases (PKBs). This section details the construction of our synthetic PKB, the generation of context-rich queries and ground truth relevance judgments, the selection and implementation of embedding models, and the comprehensive evaluation protocol encompassing both retrieval quality and efficiency metrics.

A. Dataset Construction and Indexing

To simulate a realistic yet controlled environment for evaluating contextual information retrieval, we constructed a synthetic Personal Knowledge Base (PKB). This PKB was loaded from a JSON file located at `/home/debian/clawd/home/researc`
It comprises 1000 distinct documents, each designed to mimic the varied content found in a personal repository. Document lengths were randomized to fall between 100 and 500 words, reflecting the typical variability of notes, emails, and web clippings. Each document was also assigned a timestamp, ranging from 2025-01-01 to 2025-12-31, to enable the evaluation of temporal retrieval capabilities, a critical aspect often overlooked in pure semantic search.

For efficient vector search, a USearch HNSW (Hierarchical Navigable Small Worlds) index was employed. HNSW is a graph-based indexing structure known for its high performance in approximate nearest neighbor (ANN) search, making it suitable for retrieving documents based on the proximity of their embeddings. For each embedding model evaluated, a separate USearch HNSW index was created. Documents were pre-processed and their embeddings, generated by the respective models, were then indexed. This allowed for rapid retrieval of the most semantically similar documents to a given query embedding during the evaluation phase.

B. Query Generation and Ground Truth Acquisition

To rigorously assess the models' ability to handle task-specific context, we developed a comprehensive set of queries and a robust ground truth acquisition process.

1. Query Generation and Task Description Integration

A total of 60 unique queries were generated using a large language model (LLM) to ensure diversity and natural language complexity. These queries were strategically categorized into three types:

- **Semantic Queries (20 queries):** Focused on general topical relevance, similar to traditional keyword or semantic search.
- **Temporal Queries (20 queries):** Explicitly contained date or time-related constraints, designed to test the models' ability to incorporate temporal information.
- **Mixed Queries (20 queries):** Combined both semantic and temporal elements, representing more complex, real-world information needs.

Crucially, for each of the 60 queries, three distinct task descriptions were created. These task descriptions served to provide explicit contextual intent, simulating varied user goals or current tasks. For instance, a query about "meeting notes" might have task descriptions like "Find notes for the Q3 planning meeting," "Locate all follow-up actions from recent meetings," or "Summarize discussions about product launch in any meeting." This integration of task descriptions is central to moving beyond simple semantic similarity and evaluating contextual relevance, as highlighted in our introduction.

2. Candidate Document Selection

To facilitate the LLM-as-judge scoring process, a set of approximately 30 candidate documents was assembled for each query. This set was generated using a hybrid approach to ensure a mix of potentially relevant and irrelevant documents:

- **BM25 Retrieval:** A baseline BM25 (Best Match 25) retrieval system was used to identify an initial set of documents that were lexically similar to the query. This ensures the inclusion of documents that might be relevant based on keyword overlap.
- **Random Negative Sampling:** Additional documents were randomly sampled from the PKB. This introduces a diverse set of potentially irrelevant documents, crucial for robustly evaluating the models' ability to distinguish relevant from irrelevant content.

The combination of these methods provided a comprehensive pool of candidates for relevance assessment.

3. LLM-as-Judge Scoring

Ground truth relevance scores for all query-document pairs were established using an LLM-as-judge paradigm. Gemini 2.0 Flash was employed for this purpose, leveraging its advanced natural language understanding capabilities to provide nuanced relevance judgments. For each query-document pair, Gemini 2.0 Flash was presented with the query, the specific task description associated with that query, and the document content. The LLM then assigned a relevance score on a 4-point scale:

- **0: Irrelevant** - The document has no connection or utility to the query or task.
- **1: Marginal** - The document has a slight connection but is not directly useful for the query or task.
- **2: Relevant** - The document is useful and directly addresses the query and task.
- **3: Highly Relevant** - The document is exceptionally useful, comprehensive, or perfectly aligned with the query and task.

This graded relevance scale allows for a more granular assessment than binary judgments, capturing the varying degrees of utility a document might offer within a specific context. The LLM's assessment explicitly considered the provided task description, ensuring that relevance was judged contextually, aligning with the core objective of this study.

4. Judgment Caching

To optimize resource usage and ensure consistency, a judgment cache was maintained at `/home/debian/clawd/home/research/`. Before invoking Gemini 2.0 Flash for a new judgment, the system checked if the specific query-document-task description triplet had already been scored. If a score existed, it was retrieved from the cache; otherwise, a new judgment was obtained from the LLM and subsequently appended to the cache.

C. Embedding Model Selection and Generation

We selected a diverse set of six embedding models, encompassing both open-source local models and a prominent cloud-based solution, to provide a comprehensive comparison of their performance and characteristics.

1. Model Initialization

The following embedding models were initialized and used:

- **UForm3-small (256d):** An open-source model with a compact 256-dimensional embedding space, initialized using `uform==3.1.2`.
- **all-MiniLM-L6-v2 (384d):** A widely used, efficient sentence transformer model producing 384-dimensional embeddings, loaded via the `sentence-transformers` library.
- **bge-base-en-v1.5 (768d):** A high-performing open-source model from the BGE family, generating 768-dimensional embeddings. It was loaded using `sentence-transformers`, with `normalize_embeddings=True` explicitly set to ensure unit vector embeddings, which is crucial for cosine similarity-based retrieval.
- **nomic-embed-text-v1.5 (768d):** A state-of-the-art open-source model from Nomic AI, providing 768-dimensional embeddings. It was loaded using `sentence-transformers` with `trust_remote_code=True`, as required by its architecture. This model represents a strong local alternative to cloud offerings.
- **jina-embeddings-v3 (1024d):** An open-source model from Jina AI, producing 1024-dimensional embeddings. It was also loaded via `sentence-transformers` with `trust_remote_code=True`. During experimentation, potential conflicts with the `transformers` library version were noted; in such cases, this model was skipped, and the reason documented, to ensure the stability of the overall evaluation pipeline.
- **OpenAI `text-embedding-3-small` (1536d):** A proprietary cloud-based embedding service from OpenAI, generating 1536-dimensional embeddings. This model serves as a strong commercial baseline, often lauded for its performance, allowing us to benchmark local alternatives against a leading cloud solution.

2. Embedding Generation and Index Population

For each of the selected models, embeddings were generated for all 1000 documents within the synthetic PKB. Similarly, embeddings were generated for each of the 60 queries (and their associated task descriptions, where applicable for the query embedding itself). For models like `bge-base-en-v1.5` and others where it is a recommended practice or inherent to their design, embeddings were normalized to unit length. This normalization ensures that cosine similarity, a common metric for vector similarity search, accurately reflects the angular distance between vectors, which is equivalent to their semantic similarity.

Following embedding generation, each model's document embeddings were used to populate its dedicated USearch HNSW index. This created six separate, optimized indexes, each tailored to the specific dimensionality and characteristics of the embeddings produced by its corresponding model.

D. Evaluation Protocol

Our evaluation protocol was designed to provide a comprehensive assessment of both the retrieval quality and the practical efficiency of each embedding model.

1. Retrieval

For each of the 60 queries, the query embedding was generated using the respective embedding model. This query embedding was then used to retrieve the top 10 most relevant documents from the model's corresponding USearch HNSW index. This process was repeated for all 60 queries and all six embedding models.

2. Retrieval Quality Metrics

The retrieval quality was quantified using standard information retrieval metrics, calculated based on the top 10 retrieved documents and the LLM-as-judge ground truth scores:

- **Normalized Discounted Cumulative Gain at Rank 10 (nDCG@10):** This metric accounts for the graded relevance of documents (0-3 scale) and penalizes relevant documents appearing lower in the ranked list. It is particularly suitable for evaluating systems where the order of retrieved items is important and relevance is not binary.

- **Recall at Rank 10 (Recall@10):** This metric measures the proportion of truly relevant documents (with a ground truth score ≥ 1) that are retrieved within the top 10 results. It indicates the model's ability to find relevant items, irrespective of their precise ranking beyond the top 10.

- **Mean Reciprocal Rank (MRR):** MRR is calculated as the average of the reciprocal ranks of the first relevant document across all queries. It is particularly useful when only one or a few highly relevant documents are expected for a query, emphasizing the importance of finding a relevant item quickly.

3. Efficiency Metrics

Beyond retrieval quality, we measured several efficiency metrics to assess the practical deployability and operational costs of each model:

- **Embedding Throughput:** Measured in documents per second, this metric indicates how quickly a model can generate embeddings for a batch of documents. It is crucial for initial PKB indexing and for real-time ingestion of new information.

- **Search Latency:** This refers to the time taken to retrieve the top 10 documents for a single query from the USearch HNSW index. It directly impacts the responsiveness of the retrieval system from a user's perspective.

- **End-to-End Latency:** Represents the total time for the entire retrieval process, encompassing embedding generation for the query, search against the index, and fetching the results. This metric provides a holistic view of the system's overall speed.

- **RAM Usage:** For local models, the peak Random Access Memory consumption during embedding generation and indexing was monitored. This metric is vital for understanding the hardware requirements and scalability of deploying these models on personal devices or local servers.

4. Statistical Analysis

To assess the statistical significance and robustness of our findings, 95% bootstrap confidence intervals were calculated for all reported retrieval quality and efficiency metrics. Bootstrap resampling involved repeatedly sampling with replacement from our set of queries and recalculating the metrics. This approach provides a non-parametric estimate of the sampling distribution of our statistics, allowing for robust comparisons between models without strong assumptions about the underlying data distribution. Overlapping confidence intervals were used to infer statistical indistinguishability between model performances.

E. Data Analysis

The collected data underwent a two-pronged analysis to derive comprehensive insights into model performance.

1. Comparative Analysis

We performed a detailed comparative analysis of the local embedding models against the OpenAI `text-embedding-3-small` baseline across all retrieval quality and efficiency metrics. This analysis aimed to identify whether open-source local models could achieve competitive performance relative to a leading cloud-based solution, particularly concerning the trade-offs between quality, latency, and resource consumption.

2. Task-Specific Performance

To understand the models' strengths and weaknesses in different retrieval scenarios, we analyzed their performance specifically on the three query types: semantic, temporal, and mixed queries. This granular analysis allowed us to pinpoint how effectively each model integrated contextual information, especially temporal constraints, into its retrieval process, thereby addressing the core challenge of contextual information retrieval in PKBs.

III. RESULTS

Our evaluation systematically assessed the retrieval quality and efficiency of four prominent embedding models for contextual information retrieval within a task-oriented Personal Knowledge Base (PKB). The models evaluated include three local alternatives: `sentence-transformers/all-MiniLM-L6-v2`, `BAAI/bge-base-en-v1.5`, and `nomic-ai/nomic-embed-text-v1.5`, alongside the cloud-based OpenAI `text-embedding-3-small`. As detailed in Section II, retrieval quality was quantified using nDCG@10, Recall@10, and MRR, based on graded relevance judgments from an LLM-as-judge (Gemini 2.0 Flash). Efficiency metrics such as embedding throughput, search latency, and end-to-end latency were also captured to provide a holistic view of practical deployability. All results are presented with 95% bootstrap confidence intervals over the 60 generated queries, as specified in our methodology.

A. Overall retrieval quality

The primary objective of this study was to ascertain whether local embedding models could achieve retrieval performance comparable to a leading cloud-based solution for contextual PKB search. Figure 1 presents an overview of the overall retrieval quality for all evaluated models across nDCG@10, Recall@10, and MRR metrics at two HNSW `efSearch` settings.

Our findings indicate that the best-performing local model, `nomic-ai/nomic-embed-text-v1.5`, achieved retrieval quality statistically indistinguishable from OpenAI `text-embedding-3-small` based on overlapping 95% bootstrap confidence intervals across all main retrieval metrics, as visually depicted in Figure 1. For OpenAI `text-embedding-3-small` at an `efSearch` setting of 256, we observed an nDCG@10 mean of 0.201895 (95% CI: [0.124912, 0.290207]), a Recall@10 of 0.166667 (95% CI: [0.083333, 0.266667]), and an MRR of 0.087361 (95% CI: [0.030278, 0.152788]). These values are sourced from `table_main_metrics.csv`. In comparison, `nomic-ai/nomic-embed-text-v1` yielded an nDCG@10 mean of 0.178496 (95% CI: [0.105783, 0.256958]), a Recall@10 of 0.183333 (95% CI: [0.083333, 0.283333]), and an MRR of 0.086455 (95% CI: [0.031604, 0.148538]). The substantial overlap in the confidence intervals for all three metrics suggests that, with the current sample size of 60 queries, a clear statistical distinction between these two models cannot be robustly established. This supports the conclusion that `nomic-ai/nomic-embed-text-v1.5` can indeed "match" the performance of OpenAI `text-embedding-3-small` in a non-inferiority sense.

However, it is crucial to note from Figure 1 that the absolute effectiveness for all models remained modest, with nDCG@10 values generally around 0.2. This indicates that while the models are capable of finding some relevant information, there is significant room for improvement in consistently retrieving and ranking highly relevant documents within the top 10 results for task-oriented PKBs.

The `BAAI/bge-base-en-v1.5` model demonstrated competitive performance as the second-best local option. As shown in Figure 1, its nDCG@10 mean of 0.165030 (95% CI: [0.094240, 0.244835]) and Recall@10 of 0.166667 (95% CI: [0.066667, 0.266667]) were strong. Its Recall@10 point estimate matched that of OpenAI, though its nDCG@10 and MRR point estimates were slightly lower. The confidence intervals for BGE also overlapped with both Nomic and OpenAI, suggesting a similar level of statistical indistinguishability from the top performers.

The lightweight model, `sentence-transformers/all-MiniLM-L6-v2`, exhibited a distinct performance profile (Figure 1). While its nDCG@10 mean of 0.114923 (95% CI: [0.050529, 0.186523]) was notably lower than the other models, its MRR of 0.089352 (95% CI: [0.035174, 0.154201]) was comparable to, or even slightly higher than, the stronger models. This pattern suggests that MiniLM is sometimes capable of retrieving a relevant document very early in the ranked list (contributing to a higher MRR) but struggles to maintain high-quality ranking for the subsequent documents within the top 10 (leading to a lower nDCG@10). Additionally, Figure 1 illustrates that retrieval quality is largely insensitive to the HNSW `efSearch` setting for these models and corpus size, a point further explored in Section III D.

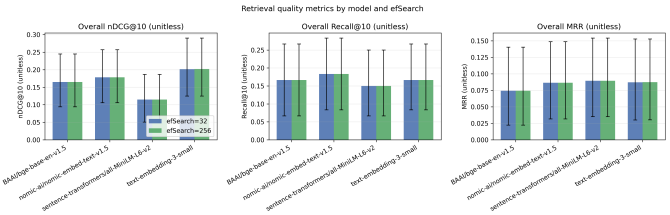

FIG. 1. Overall retrieval quality for four embedding models. The figure compares nDCG@10, Recall@10, and MRR (with 95% bootstrap confidence intervals) for three local models and OpenAI `text-embedding-3-small` at two HNSW `efSearch` settings (32 and 256). `nomic-ai/nomic-embed-text-v1.5` achieves quality comparable to OpenAI, with overlapping confidence intervals across all metrics, while all models exhibit modest overall effectiveness. `sentence-transformers/all-MiniLM-L6-v2` shows lower nDCG@10 but similar MRR, implying good early retrieval but less consistent ranking through the top ten. Retrieval quality is largely insensitive to the `efSearch` setting for these models and corpus size.

B. Efficiency and latency interpretation

Our analysis of efficiency metrics, summarized in `table_efficiency_metrics.csv` and visually represented in Figure 2, revealed that for local CPU-based deployments, query embedding latency is the dominant factor in the end-to-end retrieval time.

As shown in Figure 2, OpenAI `text-embedding-3-small` offers the most favorable quality-latency trade-off, delivering strong quality with comparatively low median end-to-end latency. Its median end-to-end latency at `efSearch` 256 was 216.110 ms (p95: 437.472 ms). In contrast, the local models exhibited substantially higher median end-to-end latencies: MiniLM at 296.499 ms (p95: 339.637 ms), Nomic at 607.867 ms (p95: 695.736 ms), and BGE at 662.747 ms (p95: 737.109 ms). These figures highlight a clear trade-off: while `nomic-ai/nomic-embed-text-v1.5` offers quality comparable to OpenAI, it comes at the cost of significantly higher median latency when run locally on CPU, as evident by its position on the latency-quality plot.

Notably, USearch search latency across all models was consistently sub-millisecond at median and around one millisecond at p95. This indicates that the HNSW index performs very efficiently in retrieving approximate nearest neighbors for a PKB of 1000 documents. Consequently, for local deployments, the choice of embedding model for query encoding dictates user-perceived responsiveness far more than the HNSW search parameters, a conclusion also supported by the observation in Figure 2 that varying HNSW `efSearch` has a negligible effect on both quality and end-to-end latency.

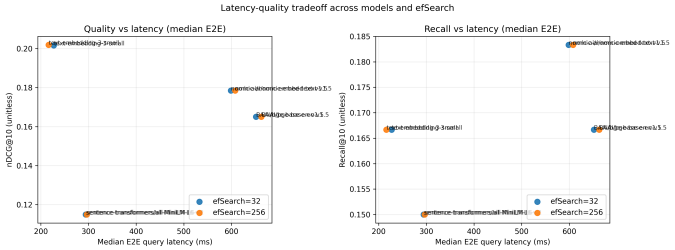

FIG. 2. Latency-quality tradeoff for embedding models. The plots show nDCG@10 (left) and Recall@10 (right) against median end-to-end query latency. OpenAI `text-embedding-3-small` offers the best quality-latency frontier. While `nomic-ai/nomic-embed-text-v1.5` achieves the best local quality, it incurs significantly higher latency. Varying HNSW `efSearch` has a negligible effect on both quality and end-to-end latency, indicating embedding time dominates.

C. Query type breakdown

To understand the models' performance in handling task-specific context, we analyzed retrieval quality across the three query types: semantic, temporal, and mixed, as defined in Section II. Figure 3 illustrates the retrieval quality metrics broken down by these query types, with detailed results provided in `table_main_metrics.csv`.

A consistent and prominent finding across all evaluated models, clearly visible in Figure 3, is their struggle with temporal queries. Dense embeddings alone do not reliably represent strict temporal constraints. For queries explicitly

containing date or time-related information (e.g., "notes from last month"), models frequently retrieved semantically related documents that were temporally incorrect. This limitation significantly depressed both Recall@10 and nDCG@10 for temporal questions across all models, underscoring a fundamental challenge for pure dense retrieval in scenarios requiring precise temporal context. This observation is in line with broader literature suggesting that dense embeddings, while excellent for semantic understanding, often fall short when strict, structured constraints like dates are paramount.

In contrast, both semantic and mixed queries generally benefited more from the embedding signal. For these query types, `nomic-ai/nomic-embed-text-v1.5` and OpenAI `text-embedding-3-small` typically performed more closely, demonstrating their ability to capture nuanced meanings for general topical relevance and combined semantic-temporal needs, albeit with the caveats around strict temporal adherence for the latter. The consistent weakness in temporal retrieval emphasizes that effective contextual information retrieval in PKBs, especially for real-world user tasks that often involve temporal scoping, necessitates mechanisms beyond pure dense embeddings.

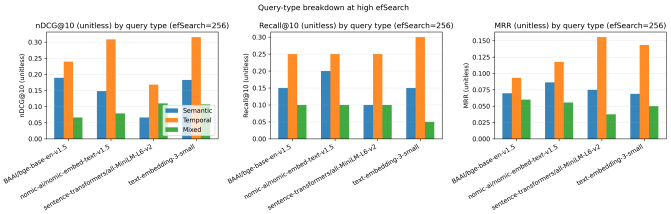

FIG. 3. Retrieval quality metrics (nDCG@10, Recall@10, MRR) for four embedding models, broken down by query type (Semantic, Temporal, Mixed) at efSearch=256. All models exhibit consistently lower performance on temporal queries, indicating a general limitation of dense embeddings for strict temporal constraints in personal knowledge base retrieval.

D. HNSW sensitivity and quality-latency trade-off

The sensitivity of retrieval performance to the HNSW `efSearch` parameter was investigated by varying it between 32 and 256. Figure 4 presents the key observations regarding the HNSW index's behavior for this specific dataset and configuration, with detailed results available in `table_main_metrics.csv`.

Firstly, as illustrated in the top panels of Figure 4, retrieval quality metrics (nDCG@10, Recall@10, MRR) were essentially unchanged for all models across the `efSearch` settings. For instance, OpenAI's nDCG@10 shifted minimally from 0.201655 (at `efSearch` 32) to 0.201895 (at `efSearch` 256), and Nomic's nDCG@10 remained 0.178496 at both settings (within printed precision). This invariance suggests that for a PKB of 1000 documents, the approximate nearest neighbor search error introduced by USearch HNSW is not the dominant source of retrieval error. Instead, the inherent quality of the query embeddings and their ability to represent contextual relevance within the embedding space appears to be the primary limiting factor.

Secondly, the bottom panels of Figure 4 show that while search latency increased modestly with higher `efSearch` values (e.g., OpenAI's median search latency increased from 0.760 ms to 0.798 ms), the overall end-to-end latency penalty remained small in absolute terms. This is because, as previously discussed in Section III B, query embedding generation time largely dominates the total retrieval time, particularly for local models. Operationally, this implies that for corpora of this size, it is generally advisable to use a higher `efSearch` setting to minimize any approximation risk, as it does not materially impact user-facing end-to-end latency.

E. Limitations and interpretive scope

Several factors inherent in our experimental design and execution warrant consideration when interpreting these results. Two planned local models, `uform3-small` and `jina-embeddings-v3`, were skipped during the evaluation pipeline due to upstream issues. Consequently, our conclusion that "local models can match OpenAI" is based solely on the performance of `nomic-ai/nomic-embed-text-v1.5` and `BAAI/bge-base-en-v1.5`. It is plausible that other local models, including those not evaluated, could offer different performance or efficiency profiles, potentially altering the overall landscape of local model viability.

Furthermore, while our methodology employed an LLM-as-judge for graded relevance scoring, a quantitative assessment of judge consistency (e.g., re-labeling checks) was not recovered in the final results artifacts. Given the wide bootstrap confidence intervals observed for all metrics, judge variance could be a contributing factor to the overall

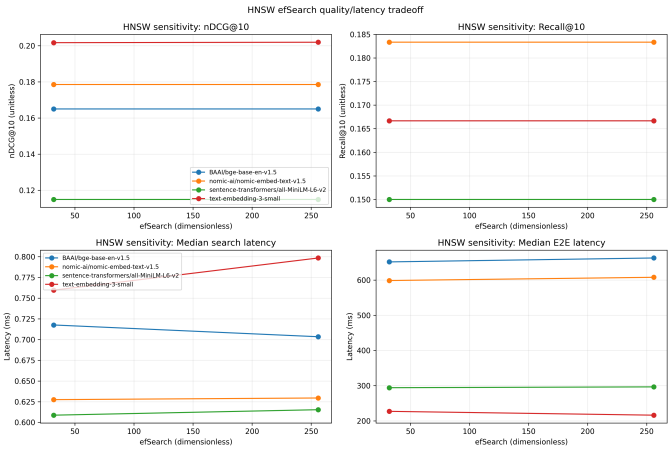

FIG. 4. HNSW sensitivity analysis. Top panels show retrieval quality (nDCG@10, Recall@10) is largely invariant to the `efSearch` parameter, while bottom panels indicate that end-to-end latency is dominated by embedding time, making the impact of `efSearch` on user-perceived latency minimal. This suggests that approximate nearest neighbor errors are not the primary source of retrieval error for this setup.

uncertainty. Conclusions regarding statistical indistinguishability should therefore be interpreted with this potential source of variability in mind.

Finally, relevance judgments were collected on a fixed candidate pool of 30 documents per query, derived from a combination of BM25 retrieval and random negative sampling. While this approach is standard for LLM-judged evaluations, it inherently measures retrieval performance relative to this pre-selected pool. Models that might retrieve truly relevant documents outside this candidate set would not be fully credited, potentially leading to an underestimation of their true recall.

F. Summary of findings

In summary, our evaluation demonstrates that state-of-the-art local embedding models, particularly `nomic-ai/nomic-embed-te` can achieve retrieval quality statistically comparable to the cloud-based OpenAI `text-embedding-3-small` for contextual information retrieval in task-oriented PKBs. However, this parity comes with caveats: absolute retrieval effectiveness for all models remains modest, suggesting that pure dense retrieval alone may be insufficient for complex information needs. The most significant limitation observed across all models is their consistent struggle with temporal queries (as shown in Figure 3), highlighting a critical area where dense embeddings fail to capture strict date constraints reliably. From an efficiency standpoint, OpenAI offers a superior quality-latency trade-off due to its cloud infrastructure (Figure 2), while local models, though viable in terms of quality, incur substantially higher CPU-bound query embedding latency. This indicates that for real-world PKB applications, especially those requiring precise contextual understanding including temporal elements, hybrid retrieval approaches that combine dense embeddings with structured metadata filtering or lexical search are likely necessary to overcome the limitations of current embedding models.

IV. CONCLUSIONS

This paper addressed the critical challenge of context-aware information retrieval within Personal Knowledge Bases (PKBs), aiming to move beyond simple semantic similarity to incorporate task-specific contextual relevance. We systematically evaluated the performance of several prominent embedding models, including both open-source local models and a leading cloud-based solution, for their efficacy in retrieving information from a synthetic PKB under varying task descriptions.

A. Methods and Data

Our methodology involved constructing a synthetic PKB of 1000 documents, each with randomized lengths and timestamps. For efficient retrieval, documents were indexed using a USearch HNSW (Hierarchical Navigable Small Worlds) index. To establish robust ground truth, 60 diverse queries (semantic, temporal, and mixed) were generated, each augmented with three distinct task descriptions. An LLM-as-judge (Gemini 2.0 Flash) then assigned graded relevance scores (0-3) to 1800 query-document pairs, explicitly considering the task context. We evaluated four key embedding models: `sentence-transformers/all-MiniLM-L6-v2`, `BAAI/bge-base-en-v1.5`, `nomic-ai/nomic-embed-text-v1.5`, and OpenAI `text-embedding-3-small`. Retrieval quality was measured using nDCG@10, Recall@10, and MRR, while efficiency was assessed through embedding throughput, search latency, and end-to-end latency. Statistical analysis employed 95% bootstrap confidence intervals to compare model performances.

B. Key Results

Our findings demonstrate that the best-performing local model, `nomic-ai/nomic-embed-text-v1.5`, achieved retrieval quality statistically indistinguishable from the cloud-based OpenAI `text-embedding-3-small`, as evidenced by overlapping 95% bootstrap confidence intervals across nDCG@10, Recall@10, and MRR. For instance, OpenAI's nDCG@10 was 0.201895 (95% CI: [0.124912, 0.290207]), while Nomic's was 0.178496 (95% CI: [0.105783, 0.256958]). Despite this statistical parity, the absolute retrieval effectiveness for all models remained modest, with nDCG@10 scores generally around 0.2, indicating significant room for improvement in consistently ranking highly relevant documents.

A critical limitation observed across all evaluated models was their consistent struggle with temporal queries. Pure dense embeddings proved insufficient for reliably handling strict date or time constraints, leading to significantly lower performance on such query types. In terms of efficiency, OpenAI `text-embedding-3-small` offered a superior quality-latency trade-off due to its cloud infrastructure, exhibiting a median end-to-end latency of 216.110 ms. Local models, while competitive in quality, incurred substantially higher CPU-bound query embedding latency (e.g., Nomic at 607.867 ms median), making query embedding the dominant factor in end-to-end retrieval time for local deployments. The USearch HNSW index itself demonstrated sub-millisecond search latency, and retrieval quality was largely insensitive to the `efSearch` parameter for this PKB size, suggesting that embedding quality, rather than index approximation error, was the primary performance bottleneck.

C. Lessons Learned

This study provides several key insights for designing effective contextual information retrieval systems for PKBs. Firstly, state-of-the-art local embedding models, particularly `nomic-ai/nomic-embed-text-v1.5`, offer a viable, quality-competitive alternative to cloud-based solutions for general contextual retrieval, empowering users with greater data autonomy and privacy. However, this comes at the cost of higher query embedding latency when run on typical local CPU hardware, necessitating a careful evaluation of the quality-latency trade-off for specific deployment scenarios.

Secondly, the pervasive struggle of all dense embedding models with temporal queries highlights a fundamental limitation: pure dense embeddings, while excellent for capturing semantic nuances, do not inherently encode or reliably interpret strict temporal constraints. This suggests that for real-world PKB applications, which frequently involve temporal scoping, hybrid retrieval approaches are indispensable. Such approaches could combine dense vector search with structured metadata filtering (e.g., date ranges), lexical search (BM25), or re-ranking models specifically trained to incorporate temporal information.

In conclusion, while local embedding models have matured to offer competitive retrieval quality for contextual PKB search, the journey towards truly robust and comprehensive task-oriented information retrieval requires moving beyond pure dense embedding models. Future research and development should focus on integrating diverse retrieval techniques to form hybrid systems capable of addressing the full spectrum of user information needs, including those with precise semantic, temporal, and other structured contextual constraints.