Temporal Retrieval-Augmented Generation for Reasoning over Dynamic Knowledge Bases

Renzo Pavanello
Department of Computer Science, George Mason University, Fairfax, VA, USA.
renzowork@gmu.edu

Domingo Gilabert
Department of Computer Science, University of Central Florida, Orlando, FL, USA.
gilabert384@ucf.edu

Fryderyk Chochol
School of Information Technology, University of Cincinnati, Cincinnati, OH, USA.
fryderyk456@uc.edu

Abstract

Temporal retrieval-augmented generation (TRAG) extends conventional retrieval-augmented generation from relatively static document collections to dynamic knowledge bases whose contents, validity, authority, and relevance change over time. Conventional systems often treat retrieval as a spatial similarity problem, ranking passages by semantic proximity while giving limited attention to temporal order, provenance, versioning, or the lifecycle of facts. This paper develops a system-level account of TRAG for reasoning over dynamic knowledge bases. It examines architectural choices across ingestion, temporal normalization, indexing, retrieval, fusion, generation, and verification, with emphasis on how time-aware representations alter infrastructure requirements and governance obligations. The discussion considers structural trade-offs among freshness, consistency, completeness, latency, cost, and fairness. It also analyzes temporal reasoning as a socio-technical capability rather than a purely algorithmic one, because validity intervals, source authority, and revision histories are shaped by institutions, policies, and record-keeping practices. The paper further explores robustness and evaluation challenges, including temporal drift, contradictory evidence, adversarial updates, and retrospective correction. Deployment and sustainability concerns are addressed through the lens of storage growth, index maintenance, compute intensity, and long-term maintainability. Governance, privacy, fairness, and policy implications are discussed, particularly when dynamic retrieval systems are used in journalism, medicine, finance, law, and public administration. The paper concludes that TRAG requires integrated design principles that treat time, provenance, and accountability as first-class infrastructure concerns rather than post hoc metadata.

Keywords

temporal retrieval-augmented generation; dynamic knowledge bases; temporal indexing; reasoning; governance; robustness; fairness; sustainability.

1. Introduction

Retrieval-augmented generation has become a central strategy for grounding language models in external knowledge, reducing reliance on purely parametric memory, and improving factual specificity in knowledge-intensive tasks [1]. Early systems demonstrated that dense

retrieval over large corpora can support open-domain question answering and generation by selecting relevant passages and conditioning output on retrieved evidence [2]. Subsequent work extended this idea to pretraining, joint retrieval and generation, and very large-scale retrieval from trillions of tokens [3], [4], [5]. These advances established a powerful architecture in which a retriever narrows a massive search space and a generator synthesizes an answer from selected evidence. However, most foundational formulations assume that the underlying knowledge base is stable enough that similarity ranking, not temporal reasoning, is the primary retrieval problem.

Dynamic knowledge bases violate that assumption. News archives, clinical guidelines, financial disclosures, legal codes, product documentation, public health advisories, and enterprise records all change continuously. Facts may be superseded, corrected, retracted, clarified, or reinterpreted. A passage that was authoritative last year may be misleading today, while an older document may remain legally binding even after newer commentary appears. Retrieval systems that optimize only semantic relevance can therefore surface outdated or decontextualized evidence, producing fluent but temporally invalid responses. The problem is not merely that models lack access to current information. It is that they often lack a structured understanding of when retrieved claims were valid, under what conditions, for which populations, and according to which institutional authority.

Temporal retrieval-augmented generation addresses this gap by incorporating time into every stage of the retrieval and reasoning pipeline. Time becomes a dimension of indexing, a criterion for ranking, a constraint on evidence selection, and a subject of explanation. The system must decide not only what is relevant but also what is current, what was true at a specified time, what changed, and how conflicting versions should be reconciled. This shift has architectural consequences. Temporal metadata increases storage and indexing complexity, raises questions about clock synchronization and timestamp provenance, and complicates evaluation because correctness may be time-indexed. It also has governance consequences, because temporal records can expose personal data histories, institutional decisions, and contested narratives.

This paper argues that TRAG should be understood as a socio-technical infrastructure rather than a single model modification. A robust temporal retrieval system integrates ingestion pipelines, temporal normalization, versioned storage, hybrid retrieval, evidence fusion, generation-time constraints, and post hoc auditing. Each layer introduces trade-offs among freshness, consistency, cost, fairness, and explainability. The discussion proceeds from conceptual foundations to architecture, indexing, reasoning, robustness, governance, deployment, and cross-domain comparison. The central claim is that time-aware retrieval is not simply a feature added to existing systems; it changes the epistemic contract between users, institutions, and machine-generated answers.

2. From Static Retrieval to Temporal Knowledge Infrastructure

Conventional retrieval-augmented generation treats knowledge as a corpus of passages that can be embedded and compared. Dense passage retrieval and related methods show that learned representations can outperform sparse lexical matching in many open-domain settings [2]. Retrieval-augmented pretraining further demonstrates that models can learn to use retrieved evidence during training rather than only at inference [3]. Fusion-in-decoder approaches show that multiple retrieved passages can be combined effectively by a generator [4]. Large-scale retrieval systems extend this logic to massive token collections, suggesting that scale itself can improve coverage [5]. These systems are powerful, but they generally

treat documents as simultaneous and equally available. Time is often represented, if at all, as a simple recency feature or a filter on publication date.

Dynamic knowledge bases require a richer temporal ontology. Records may have publication time, event time, validity start, validity end, ingestion time, revision time, and authority status. A clinical guideline may be published in one year, updated in another, and applied retrospectively to earlier cases. A financial statement may describe a period that ended before its publication. A legal decision may overturn an earlier interpretation while preserving the earlier decision as part of the record. A news article may be corrected after publication, leaving both the original and the correction relevant for different purposes. Static retrieval flattens these distinctions. Temporal retrieval must preserve them.

The movement toward self-reflective and adaptive retrieval systems provides partial foundations. Self-RAG explores learning when to retrieve and how to critique generated evidence [6]. Retrieval-augmented black-box language models show that external retrieval can improve systems without retraining the generator [7]. Surveys of retrieval-augmented generation map a growing design space that includes query rewriting, multi-hop retrieval, and evidence selection [8]. In-context retrieval-augmented language models demonstrate that retrieval can be integrated into prompting workflows [9]. These advances are necessary but insufficient for dynamic knowledge bases because they rarely specify how evidence should be ordered, versioned, or invalidated over time.

The historical trajectory from reading comprehension over Wikipedia to knowledge-base reasoning illustrates the challenge. Reading Wikipedia to answer open-domain questions established a strong benchmark for retrieval and reading [10]. Work on language models as knowledge bases questioned whether parametric memory alone could reliably store factual knowledge [11]. Datasets such as Quasar and narrative-alignment corpora showed that evidence can be distributed across search results, books, and media [12], [13]. Heterogeneous benchmarks such as BEIR demonstrated that retrieval models must generalize across domains and tasks [14]. Dynamic knowledge bases intensify these challenges because the target of retrieval is not a fixed answer but a temporally situated evidence set. The system must retrieve not only what is semantically related but also what is temporally legitimate for the user's question.

3. System Architecture for Temporal Retrieval-Augmented Generation

A TRAG architecture must coordinate several subsystems. The ingestion layer collects documents, database records, APIs, streams, and sensor outputs. It assigns timestamps, extracts validity intervals, identifies authors or institutions, and records provenance. The normalization layer resolves date formats, time zones, calendar systems, and ambiguous temporal expressions. The indexing layer stores content representations alongside temporal metadata and source authority. The retrieval layer combines semantic similarity, lexical matching, temporal filtering, and authority weighting. The fusion layer reconciles multiple evidence sources, including conflicting versions. The generation layer conditions output on selected evidence and temporal constraints. The verification layer checks whether claims are supported, current, and appropriately qualified.

A key architectural distinction is between retrieval for recency and retrieval for temporal reasoning. Recency-oriented systems prioritize the newest evidence, which is appropriate for rapidly changing domains such as breaking news or operational monitoring. Temporal reasoning systems must also support as-of queries, interval queries, change detection, and

counterfactual historical analysis. A user may ask what policy was in effect during a past period, how a recommendation changed after new evidence, or whether a current claim is consistent with earlier statements. These tasks require the system to reason over time rather than simply prefer newer documents. A recent multi-source framework illustrates the importance of unified semantic spaces and attention-based fusion when heterogeneous retrieval sources must be combined [16]. Such fusion is especially relevant in TRAG because evidence may arrive from structured databases, unstructured reports, and real-time feeds with different temporal granularities.

Embedding models also shape temporal architecture. Multi-lingual and multi-granularity embeddings can improve retrieval across languages and document types [15]. Weakly supervised contrastive pretraining for text embeddings has advanced general-purpose retrieval quality [17]. However, embeddings trained on static snapshots may not capture temporal relations such as before, after, during, superseded by, or valid until. TRAG systems may need temporal adapters, date-aware encoders, or hybrid representations that separate semantic content from temporal context. The architecture must also decide whether time is encoded into the vector space, applied as a filter after retrieval, or handled by a separate temporal reasoner. Each choice affects recall, interpretability, latency, and maintainability.

4. Indexing, Storage, and Temporal Representation

Indexing dynamic knowledge bases requires storage designs that support versioning, deletion, correction, and retrospective queries. A simple document store with a single current version is inadequate when users need historical answers or when audits require reconstruction of past system behavior. Temporal databases, append-only logs, and immutable object stores offer different trade-offs. Append-only designs improve auditability and reproducibility but increase storage volume. Mutable designs save space but can obscure how a fact changed. Many systems adopt hybrid approaches in which current versions are optimized for low-latency retrieval while historical versions are archived in cheaper storage and activated for as-of queries.

Approximate nearest neighbor indexes are central to dense retrieval at scale. Hierarchical navigable small world graphs provide efficient and robust similarity search [18], while GPU-accelerated similarity search enables billion-scale retrieval [19]. Temporal indexing complicates these structures because filters on time and authority must be applied without destroying recall. A system may partition indexes by time period, maintain multiple snapshots, or attach temporal attributes to vectors. Partitioning by time can improve freshness and simplify deletion, but it fragments semantic neighborhoods and may require merging results across partitions. Snapshot indexes support reproducibility but become expensive as snapshots accumulate. Attribute-aware indexing can balance these concerns, but it requires careful query planning and metadata governance.

Temporal representation also affects evaluation. Heterogeneous retrieval benchmarks such as BEIR test zero-shot generalization across domains [14]. Embedding models such as M3-Embedding support multi-lingual and multi-granularity retrieval [15]. Yet most benchmarks do not systematically test whether a system retrieves the right version of a document for a given as-of date. Future benchmarks must include temporal validity, contradiction, revision history, and provenance. Without such evaluation, systems may appear accurate on current questions while failing on historical, legal, or clinical queries where temporal context is decisive.

5. Temporal Reasoning and Consistency

Reasoning over dynamic knowledge bases involves more than filtering by date. The system must determine whether retrieved evidence is mutually consistent, whether one source supersedes another, and whether a claim was valid at the time referenced by the query. Temporal consistency can be modeled through validity intervals, event ordering, and source authority. For example, a clinical recommendation may be valid for a specific period and population. A financial disclosure may describe a fiscal period that differs from its publication date. A legal statute may remain in force even after later cases interpret it differently. TRAG systems must represent these distinctions to avoid conflating publication time with event time.

Contradiction handling is particularly difficult. When two sources conflict, the system may prefer the newer source, the more authoritative source, the source closer to the event, or a synthesis that explains the disagreement. A simple recency heuristic can fail when newer sources are less reliable or when older sources remain legally controlling. Authority-aware fusion is therefore necessary. The fusion layer should consider provenance, institutional standing, methodological quality, and domain norms. Attention-based fusion can help integrate heterogeneous evidence, but it must be constrained by explicit temporal and authority signals rather than relying only on learned relevance [16].

Multi-hop reasoning adds further complexity. A query may require connecting an event, a policy change, and a later outcome across different time periods. In-context retrieval-augmented models can support such chains by incorporating retrieved evidence into prompts [9], while self-reflective systems can critique whether the evidence supports the answer [6]. However, temporal multi-hop reasoning requires the system to maintain a coherent timeline. It must distinguish between causal order and narrative order, between retrospective interpretation and contemporaneous evidence, and between stable facts and changing circumstances. These requirements push TRAG toward structured knowledge representation, even when the generator remains a large language model.

6. Robustness, Evaluation, and Operational Risk

TRAG systems face robustness challenges that static systems often underplay. Temporal drift occurs when the distribution of facts, language, or source availability changes over time. A model trained on historical data may perform poorly on current queries, while a model tuned for current events may misinterpret historical language. Adversarial updates can inject false or misleading documents into dynamic sources. Coordinated disinformation campaigns may exploit recency biases by flooding retrieval indexes with timely but false content. Robustness therefore depends on source verification, anomaly detection, and provenance tracking.

Evaluation must move beyond single-point accuracy. A TRAG system should be assessed on temporal accuracy, version correctness, contradiction resolution, calibration of uncertainty, and auditability. Heterogeneous retrieval benchmarks provide a foundation for cross-domain evaluation [14], but they must be extended with temporal annotations. Surveys of retrieval-augmented generation highlight the need for standardized metrics and failure analysis [8]. Black-box retrieval systems show that external evidence can improve performance without retraining [7], but they also raise questions about how evidence is selected and updated. Large-scale retrieval systems demonstrate coverage benefits [5], yet coverage without temporal validity can amplify outdated or conflicting claims.

Operational risk also includes silent failure. A system may retrieve an outdated guideline but present it fluently, without indicating that a newer version exists. It may merge evidence from

different periods into a single coherent but historically inaccurate answer. It may fail to recognize that a source has been retracted. Mitigation requires explicit uncertainty communication, version warnings, and answer-level provenance. The system should be able to say when evidence is contested, when a source has been superseded, and when the available record is incomplete. These capabilities are as much infrastructural as they are model-centric.

7. Governance, Fairness, Privacy, and Policy

Temporal retrieval systems raise governance questions because they preserve and expose histories. Model cards and dataset documentation provide precedents for transparency and accountability [20]. Dynamic knowledge bases require similar documentation for temporal provenance, update policies, and authority criteria. Who decides which source is authoritative? How are corrections propagated? How long are deleted or retracted items retained for audit? These questions are institutional as much as technical. A hospital, newsroom, financial regulator, or court may have different retention rules, evidentiary standards, and privacy obligations.

Fairness concerns arise when temporal data reflects historical inequities. A retrieval system trained on historical records may reproduce discriminatory patterns from the past. A credit or hiring system that uses temporal records may penalize individuals for old information that is no longer relevant. A medical system may underrepresent populations that were excluded from earlier studies. TRAG must therefore support fairness-aware temporal filtering, including expiration of certain data types, correction of biased historical labels, and consideration of disparate impact across groups. The system should not treat all historical data as equally usable merely because it is retrievable.

Privacy is equally challenging. Temporal records can reveal personal trajectories, changing addresses, employment histories, health conditions, and political activities. Differential privacy and related techniques offer formal protections, but dynamic retrieval complicates their application because repeated queries over time can enable inference. Data minimization, purpose limitation, and retention limits become architectural requirements. The system must be able to forget, redact, or restrict access without breaking auditability. This tension between memory and forgetting is central to the governance of temporal knowledge infrastructures.

8. Deployment, Sustainability, and Cost

Deploying TRAG at scale requires significant infrastructure. Ingestion pipelines must handle continuous streams, batch updates, and schema changes. Indexes must be rebuilt, merged, or incrementally updated. Storage must accommodate multiple versions, provenance records, and audit logs. Compute must support embedding generation, retrieval, fusion, and generation. These costs are not incidental. They determine whether a system can remain fresh, reproducible, and equitable across users and regions.

Sustainability concerns follow from these requirements. Large-scale retrieval systems already consume substantial compute [5]. GPU-accelerated similarity search improves efficiency but still requires energy and hardware [19]. Graph-based indexes such as HNSW offer efficient search [18], yet maintaining them over time adds operational overhead. Multi-lingual and multi-granularity embeddings increase coverage but may require larger models and more storage [15]. A sustainable TRAG design must consider lifecycle costs, including hardware refresh, data transfer, cold storage, and the energy intensity of continuous indexing.

Architectural choices such as tiered storage, incremental updates, and query-aware retrieval can reduce waste, but they must be balanced against latency and correctness.

Cost also affects access. If only well-resourced institutions can maintain high-quality temporal indexes, the benefits of TRAG may concentrate in wealthy organizations. Public knowledge institutions, small newsrooms, community clinics, and legal aid organizations may lack the infrastructure to keep pace. Policy interventions such as shared temporal indexes, open provenance standards, and public auditing infrastructure could reduce inequality. The design of TRAG is therefore not only a technical optimization problem but also a question of equitable knowledge infrastructure.

9. Case Illustrations and Cross-Domain Comparisons

In journalism, TRAG can help reporters trace how a claim evolved, identify corrections, and distinguish contemporaneous reporting from later analysis. A system that retrieves only current articles may miss the historical record. A system that retrieves only historical articles may fail to reflect corrections. Temporal retrieval must preserve both the original publication and its revision history, allowing users to see how narratives changed. Reading Wikipedia to answer questions provides a baseline for evidence-based synthesis [10], but journalism requires stronger provenance and editorial context.

In medicine, clinical guidelines change as evidence accumulates. A treatment recommendation may be valid for one period but contraindicated later. A TRAG system must support as-of queries for clinicians reviewing historical cases and current queries for active care. It must distinguish between publication date, evidence date, and implementation date. It must also handle conflicting guidelines from different professional bodies. These requirements demand authority-aware fusion and transparent uncertainty. The system cannot simply average sources or prefer the newest document without considering the quality of evidence.

In finance, temporal reasoning is central to regulatory compliance, risk assessment, and fraud detection. Financial disclosures describe periods that may not align with publication dates. Market conditions change rapidly, and a strategy that was appropriate under one regime may be unsafe under another. A TRAG system must retrieve time-indexed evidence, detect regime changes, and avoid look-ahead bias in backtesting. In law, the problem is even more acute because older statutes and cases may remain binding. A legal retrieval system must distinguish overruled, distinguished, and still-valid precedents. These cross-domain comparisons show that temporal reasoning is not a single capability but a family of domain-specific requirements.

10. Future Directions

Future research should develop temporal benchmarks that test version correctness, as-of retrieval, contradiction resolution, and provenance explanation. It should integrate temporal knowledge graphs with dense retrieval, allowing structured and unstructured evidence to be combined [16]. It should explore continual learning and index updating without catastrophic forgetting. It should design human-in-the-loop workflows for correcting temporal metadata and adjudicating conflicts. It should also advance privacy-preserving retrieval, including methods that allow deletion, redaction, and purpose-limited access without sacrificing auditability.

Another direction is the development of temporal governance standards. Model cards and dataset documentation offer starting points [20], but dynamic systems need update cards, provenance standards, and expiration policies. These artifacts should specify who can modify records, how corrections propagate, how long versions are retained, and how users are informed when evidence changes. Standards should also address fairness across time, ensuring that historical inequities are not automatically reproduced. Technical mechanisms such as HNSW indexes [18] and self-reflective retrieval [6] will matter, but they must be embedded in institutional processes that assign responsibility for temporal accuracy.

Finally, TRAG should be evaluated as an infrastructure rather than a standalone model. Its success depends on ingestion reliability, metadata quality, index maintainability, governance clarity, and user trust. In-context retrieval and black-box augmentation provide flexibility [7], [9], but they do not absolve designers of the need for temporal accountability. The most promising systems will combine semantic retrieval, temporal constraints, authority-aware fusion, and transparent provenance. They will treat time not as an auxiliary feature but as a constitutive dimension of knowledge.

11. Conclusion

Temporal retrieval-augmented generation addresses a fundamental limitation of static retrieval systems: the assumption that knowledge is stable enough that relevance can be separated from time. Dynamic knowledge bases require systems that can represent validity, versioning, provenance, and change. This paper has argued that TRAG is a socio-technical infrastructure spanning ingestion, indexing, retrieval, fusion, generation, verification, governance, and deployment. Each layer introduces trade-offs among freshness, consistency, cost, fairness, and sustainability. The architecture must support as-of queries, contradiction handling, authority-aware fusion, and transparent uncertainty. It must also be accountable to policy, privacy, and equity concerns.

The transition from static to temporal retrieval is not merely an incremental improvement. It changes the epistemic contract between users and machine-generated answers. A TRAG system must not only retrieve relevant evidence but also explain when that evidence was valid, who authorized it, and how it relates to other versions. It must be robust to drift and adversarial updates, sustainable over long periods, and fair across populations and institutions. These requirements cannot be met by model scaling alone. They demand integrated design, standards, and governance. Temporal retrieval-augmented generation therefore represents both a technical research agenda and an infrastructural commitment to reasoning responsibly over changing knowledge.

References

1. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.

2. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense passage retrieval for open-domain question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (pp. 6769-6781). Association for Computational Linguistics.

3. Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. (2020). Retrieval augmented language model pre-training. In Proceedings of the 37th International Conference on Machine Learning (pp. 3929-3938). PMLR.

4. Izacard, G., & Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (pp. 874-880). Association for Computational Linguistics.

5. Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., van den Driessche, G., Lespiau, J. B., Damoc, B., Clark, A., de Las Casas, D., Guy, A., Menick, J., Ring, R., Hennigan, T., Huang, S., Maggiore, L., Jones, C., Cassirer, A., & Sifre, L. (2022). Improving language models by retrieving from trillions of tokens. In Proceedings of the 39th International Conference on Machine Learning (pp. 2206-2240). PMLR.

6. Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In The Twelfth International Conference on Learning Representations. ICLR.

7. Shi, W., Min, S., Yasunaga, M., Seo, M., James, R., Lewis, M., Zettlemoyer, L., & Yih, W. (2023). REPLUG: Retrieval-augmented black-box language models. arXiv preprint arXiv:2301.12652.

8. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.

9. Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., & Shoham, Y. (2023). In-context retrieval-augmented language models. Transactions of the Association for Computational Linguistics, 11, 1316-1331.

10. Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). Reading Wikipedia to answer open-domain questions. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (pp. 1870-1879). Association for Computational Linguistics.

11. Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., & Miller, A. (2019). Language models as knowledge bases? In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (pp. 2463-2473). Association for Computational Linguistics.

12. Dhingra, B., Mazaitis, K., & Cohen, W. W. (2017). Quasar: Datasets for question answering by search and reading. arXiv preprint arXiv:1707.03904.

13. Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE International Conference on Computer Vision (pp. 19-27). IEEE.

14. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., & Gurevych, I. (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 (Round 2). NeurIPS.

15. Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., & Liu, K. (2024). M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Findings of the Association for Computational Linguistics: ACL 2024 (pp. 2318-2335). Association for Computational Linguistics.

16. Deng, Q., Liu, S., Qiu, H., Long, N., Lin, W., & Chen, S. (2026, April). Unified Semantic Space and Attention-Based Fusion Framework for Multi-Source Heterogeneous Retrieval-Augmented Generation. In 2026 6th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA) (pp. 591-597). IEEE.

17. Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., & Wei, F. (2022). Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint arXiv:2212.03533.

18. Malkov, Y. A., & Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(4), 824-836.

19. Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 535-547.

20. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. In Proceedings of the Conference on Fairness, Accountability, and Transparency (pp. 220-229). ACM.