Temporal Information Retrieval and Question Answering using Retrieval Augmented Generation

Sonith Bingi
UW-Madison
sbingi@wisc.edu

Manoj Arulmurugan
UW-Madison
arulmurugan@wisc.edu

Prasad Jawale
UW-Madison
pjawale@wisc.edu

Gauri Patki
UW-Madison
gpatki@wisc.edu

1 Introduction

Retrieval-Augmented Generation (RAG) models enhance knowledge-intensive tasks by combining a retriever, which extracts relevant external documents, with a generator that produces responses grounded in this evidence. While RAG frameworks provide fact-based answers, they typically depend on semantic similarity and often ignore temporal context. As facts evolve and events unfold over time, the validity of information becomes tied to its temporal relevance. Neglecting this dimension can lead to inconsistencies, such as retrieving outdated or temporally mismatched evidence. Temporal RAG addresses this by integrating time-aware retrieval and generation, ensuring that evidence aligns with the query's timeframe and that generated text reflects when facts are valid.

Building on this foundation, our work reimplements a temporal retrieval pipeline grounded in the MRAG framework. Using Contriever as the base dense retriever, we construct a fixed ATLAS-based corpus with gold-anchored positives and BM25-mined hard negatives to preserve retrieval difficulty. MRAG re-ranking then combines semantic and temporal signals through question-focused scoring and lightweight summarization, evaluated on the TempRAGEval benchmark.

To address these challenges, our work builds on the Multi-Stage Retrieval-Augmented Generation (MRAG) framework: a modular, interpretable architecture that couples dense retrieval with question-focused re-ranking. In MRAG, an initial retriever (Contriever) produces candidate passages from a fixed corpus, which are then refined through a semantic–temporal hybrid re-ranking stage. This re-ranking computes question-focused signals (QFS), performs lightweight summarization, and aggregates sentence-level scores to produce temporally coherent rankings. Using the ATLAS corpus and the TempRAGEval benchmark, our MRAG implementation achieves strong retrieval metrics such as Hit@K, MRR@K, and Recall@K, showing its effectiveness in surfacing time-aligned evidence.

2 Literature Survey

2.1 Temporal RAGs

The survey by (Piryani et al., 2025) identifies Temporal RAG (Retrieval-Augmented Generation) as a significant advancement in addressing the static knowledge limitations of temporal language models (TLMs) such as TempoT5 and BiTimeBERT, which are susceptible to outdated information and temporal hallucinations. Temporal RAG builds upon the standard RAG framework by incorporating time-aware retrieval and reasoning, thereby ensuring temporal consistency between queries and supporting evidence. This approach enhances retrievers, including TempRetriever and TsContriever, through timestamp-aware encoding that integrates explicit timestamps, temporal decay, or event timelines into dense vector representations. These improvements allow retrieval systems to prioritize documents that are relevant to the specific timeframe of a given query.

2.2 Modular RAG

Modular RAG(Siyue et al., 2025) is a training-free, modular framework that separates semantic relevance from temporal reasoning across three stages: Question Processing, Retrieval and Summarization, and Semantic–Temporal Hybrid Ranking. In the final ranking stage, semantic similarity is combined with a symbolic temporal alignment score using spline functions to assess how well the evidence aligns with the query's timeframe. The authors introduce TEMPRAGEVAL, a benchmark derived from TIMEQA and SITUATEDQA that includes temporal perturbations and human-annotated gold evidence to evaluate the model. These perturbations reveal that the performance of existing retrievers, such as GEMMA, drops significantly when temporal changes are introduced. Experimental results show MRAG achieves significant improvements in both gold evidence recall and question answering accuracy on TEMPRAGEVAL, establishing it as a robust framework for time-sensitive reasoning.

2.3 SituatedQA

SITUATEDQA (Zhang and Choi, 2021) highlights one of the main shortcomings of conventional open-retrieval Question Answering (QA) systems: the assumption that answers are fixed and do not vary based on contextual data coming from the external world, primarily time (when) and place (where). In the authors' view, for the purpose of providing helpful and accurate answers, QA systems ought to be able to consider the extra-linguistic context of the questioners. Being a diagnostic benchmark for Temporal RAG, the work evaluates models like Dense Passage Retrieval (DPR) and Closed Book approaches (BART) based on how flexible they can be under different situations. Their diagnostic performance detects that such models suffer heavily from performance declines upon reaching time-based queries, which indicates the need for context-aware retrieval and reasoning within QA systems.

2.4 Dense Passage Retrieval

Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) is a neural retrieval model that replaces keyword-based search with dense vector embeddings for open-domain question answering. It uses two BERT encoders, one for questions and one for passages, trained jointly through contrastive learning to align semantically related pairs. This design enables large-scale retrieval via maximum inner product search, retrieving passages by meaning rather than word overlap. Integrated into a two-stage QA pipeline, DPR significantly improves retrieval accuracy and overall performance on datasets like Natural Questions and TriviaQA, influencing later dense retrieval and retrieval-augmented generation systems that combine neural search with language models.

2.5 Temporal Hard Mining

Hard negative mining, introduced in Dense Passage Retrieval (DPR) (Karpukhin et al., 2020), trains models to distinguish relevant passages from misleading but similar ones. Hard negatives are passages that appear relevant under methods like BM25 yet lack the correct answer. Traditional retrievers trained on static data often fail on time-sensitive queries, returning outdated information (Zhang and Choi, 2021). Temporal hard negatives address this by including passages that fall outside a query's valid time frame. Models such as TS-Retriever (Wu et al., 2024) use these negatives to improve temporal grounding, achieving 13 to 27 percent performance gains on diachronic questions (Lau et al., 2025) and enhancing factual accuracy in dynamic knowledge contexts.

2.6 Contriever

Contriever is an unsupervised dense retriever that maps queries and documents into a shared semantic space using a single encoder trained with contrastive learning on unlabeled text like Wikipedia and CCNet. Its unified design simplifies training and achieves strong zero-shot retrieval performance, rivaling supervised models on benchmarks such as BEIR (Izacard et al., 2022). In retrieval-augmented generation (RAG), Contriever provides domain-general and generation-aligned retrieval. Recent work introduces reinforced contrastive learning that uses generation feedback to enhance relevance (Zhou and Chen, 2025) and temporal extensions that integrate time-aware signals to improve recall and precision on temporal datasets like TimeQA (Abdallah et al., 2025).

3 Implementation

The implementation of our MRAG re-creation centers on faithfully reproducing the modular design of the original framework while maintaining computational feasibility for evaluation on a fixed corpus. The overall architecture follows a two-stage retrieval pipeline in which dense semantic retrieval is first performed by the Contriever model, and the resulting top-K candidates are then refined through a hybrid semantic-temporal re-ranking module incorporating question-focused summarization (QFS). The design deliberately preserves MRAG's separation between retrieval, summarization, and ranking while optimizing the engineering workflow for reproducibility and efficiency.

3.1 Retrieval Corpus

Our experiments use the 2021 English Wikipedia dump referenced in the ATLAS: Few-Shot Learning paper (Izacard et al., 2022), which provides a comprehensive and standardized open-domain corpus widely adopted for retrieval-augmented generation research. Wikipedia remains the most suitable foundation for RAG experiments because of its factual accuracy, topical diversity, and stable versioning. Using a static snapshot ensures that all temporal comparisons and retrieval evaluations are consistent and unaffected by continuous content edits. Each article is pre-processed into smaller text segments, allowing retrieval at the passage level rather than at the entire article scale. This segmentation strategy balances context preservation with retrieval efficiency by keeping each passage long enough to convey meaning but short enough to remain discriminative during embedding-based similarity search.

By grounding retrieval in a fixed and publicly referenced corpus, our system inherits the interpretability and comparability of prior benchmarks such as Dense Passage Retrieval (DPR) and ATLAS. Since the corpus covers a broad time range of historical and contemporary topics, it naturally provides a strong base for testing temporal generalization, an essential aspect of MRAG's design philosophy. Importantly, we normalize metadata such as publication timestamps, titles, and unique identifiers to support time-aware retrieval in future iterations. This attention to corpus structuring not only supports the MRAG pipeline but also creates a reusable foundation for subsequent experiments on temporal reasoning.

3.2 MRAG Re-Implementation

Our MRAG reproduction is structured as a two-stage retrieval-and-re-ranking system designed for modularity and interpretability. The implementation can be divided into three key phases: first-stage dense retrieval using Contriever, question-focused signal extraction for semantic-temporal matching, and sentence-level hybrid re-ranking. Together, these stages operationalize the MRAG architecture within a reproducible and computationally lightweight experimental setup.

First-Stage Dense Retrieval: We employ the `facebook/contriever-msmarco` model as the backbone retriever. Contriever's unsupervised contrastive pretraining enables domain-agnostic semantic matching, making it ideal for our baseline retrieval layer. The process begins by encoding each passage from the Wikipedia corpus into high-dimensional dense embeddings using Contriever's document encoder. These vectors are L2-normalized and stored in a FAISS `IndexFlatIP` structure, which performs efficient inner-product search equivalent to cosine similarity. At query time, each question is embedded using the query encoder, and FAISS retrieves the top-K most similar passages. This design mirrors the first stage of MRAG but replaces heavy supervision with efficient dense embedding retrieval.

To improve scalability, we pre-compute and store all passage embeddings once. This preprocessing step eliminates redundant encoding during repeated evaluations, reducing latency while maintaining fidelity to MRAG's semantic retrieval stage. The system is designed to return the top-100 candidates (K = 100) per query, although this can be tuned based on available compute. The retrieved passages serve as input for the second-stage re-ranking module.

Question-Focused Re-Ranking: The second stage introduces MRAG's defining innovation: hybrid semantic-temporal reasoning through question-focused signal extraction. For each question, we extract both content-based and temporal cues. The question is tokenized, and a small keyword set is generated that includes temporal indicators (e.g., years, months, or phrases like "in 2023" or "during the pandemic"). For each candidate passage, we compute several sentence-level scores: token overlap with question keywords, semantic similarity using the underlying Contriever embeddings, and optional question-focused summary overlap. The latter is derived by prompting a lightweight language model (`Phi-3.5-mini-instruct`) to summarize each candidate passage in the context of the query, effectively producing condensed, question-specific representations. While this step is optional, it reflects MRAG's modular structure that fuses retrieval with summarization.

Each passage is then decomposed into sentences, and the system computes a hybrid score that aggregates keyword overlap, semantic similarity, and optional summary alignment. These sentence-level scores are combined via **max pooling**, capturing the best-matching sentence as a proxy for passage relevance. The resulting passage-level scores are used to re-rank the top-K candidates. This fine-grained scoring structure helps highlight temporally aligned evidence even when a passage contains multiple events or years. It also allows the system to differentiate between passages that mention similar facts at different times, a key challenge in temporal retrieval.

Efficiency and Modularity: To ensure the entire pipeline remains tractable under limited compute, we implement several optimizations while retaining MRAG's architectural principles. Candidate passages are pre-tokenized once and reused across all queries, reducing overhead during the QFS step. The summarization module can be toggled on or off depending on whether the evaluation is large-scale or qualitative. During full benchmark runs, summarization is disabled to improve speed; during targeted analysis, it is enabled to replicate MRAG's richer signal generation. This design reflects MRAG's emphasis on modularity, as components can be swapped or simplified without breaking the pipeline's structure. Overall, this re-implementation provides a faithful, computationally efficient reproduction of MRAG's hybrid retrieval mechanism.

3.3 Evaluation and Results

To assess retrieval performance, we evaluate both the Contriever baseline and MRAG's re-ranked outputs on **TempRAGEval**, a benchmark specifically designed to test time-sensitive question answering in retrieval-augmented systems. TempRAGEval integrates question sets from TimeQA and SituatedQA, covering a diverse range of temporal reasoning scenarios, from explicit date queries to implicit temporal shifts. Each question is associated with a known answer span, and we identify a passage as relevant if it contains this answer string after normalization and case folding. This provides an automatic but interpretable measure of corpus-level relevance.

We adopt three metrics to capture retrieval quality across both recall and ranking perspectives:

1. **Hit@K (Success@K):** the proportion of queries with at least one relevant passage among the top-K retrieved results. This measures whether the retriever can surface at least one correct passage.

2. **MRR@K (Mean Reciprocal Rank):** the average of reciprocal ranks of the first relevant passage per query, truncated at K. This emphasizes how early in the ranking relevant results appear.

3. **Recall@K:** the fraction of all relevant passages that are retrieved within the top-K results, reflecting overall coverage independent of rank.

These metrics are computed on a fixed evaluation pool restricted to queries with at least one relevant passage (the "evaluable" subset). The results of our experiments are summarized in Table 1. The MRAG-based re-ranking shows clear improvements over the initial Contriever baseline, particularly in Hit@K and Recall@K, indicating that hybrid re-ranking effectively prioritizes time-aligned passages. The modest MRR@K gain reflects that temporal disambiguation often improves coverage rather than dramatically shifting the top-ranked passage, consistent with MRAG's design goal of precision refinement rather than wholesale reranking.

Metric	**K=1**	**K=5**	**K=10**	**K=20**
Hit@K	0.2069	0.5268	0.7211	0.8199
MRR@K	0.2069	0.3109	0.3383	0.3451
Recall@K	0.0422	0.1228	0.1927	0.2236

Table 1: MRAG Re-ranking performance on TempRAGEval (fixed pool).

These findings suggest that MRAG's modular reranking pipeline improves retrieval breadth and stability across temporal queries. The hybrid approach particularly benefits questions involving evolving entities or recurrent events, where purely semantic retrieval often struggles. The observed Recall@20 of 0.2236 confirms that nearly one-quarter of all relevant passages are captured among the top-20 candidates, a strong result given the large corpus size and limited computational resources.

Faithfulness to MRAG: Our implementation mirrors the structural essence of MRAG: a dense retriever followed by a semantic-temporal refinement stage that integrates summarization cues. While we introduce engineering optimizations, such as pre-tokenization, FAISS-based indexing, and optional summary toggling, these choices preserve the intended modular design. Each module can be swapped with heavier models (e.g., larger LLMs for summarization or cross-encoders for reranking) without altering the overall flow. This modular integrity ensures that our system remains a faithful, efficient reproduction of MRAG's core principles while being suitable for constrained research environments.

4 Potential Methods

4.1 Metadata Aware Retrieval

Make the retriever use available document or chunk metadata to respect time windows, not just recency, by considering publication date plus other fields such as updated date and access/crawl time. We'll normalize these into a simple validity window per chunk and address known challenges like date ambiguity and the "publication vs event time" mismatch. Although under-explored and technically tricky, we will still ship a prototype as advised. (Publication timestamps are provided in ArchivalQA(Wang et al., 2022) and ChroniclingAmericaQA(Piryani et al., 2024), and modular time-aware retrieval frameworks separate temporal intent from semantic content.). As suggested by the Professor, we have no reason not to explore this approach, so we will try implementing a method using it.

4.2 Temporal Hard Mining

Our approach extends Contriever into a time-aware retriever through data generation, temporal mining, and continual learning to prevent catastrophic forgetting. The complete pipeline is implemented end-to-end, integrating temporal question generation, negative sampling, and contrastive fine-tuning, followed by in-domain and out-of-domain evaluations. The objective is to enable the retriever to differentiate between temporally valid and outdated evidence while maintaining its general retrieval ability across domains.

We begin by generating synthetic temporal supervision data. A regex-based extractor identifies four-digit years in Wikipedia passages from a cleaned subset of the ATLAS corpus. For each passage containing a year, a question-generation model (valhalla/t5-base-qg-hl) is prompted with "generate question about YYYY: passage" to produce a temporally anchored question. Each resulting question–passage pair explicitly ties the semantics of the question to a temporal reference within the passage. This process yields thousands of examples that require the retriever to align queries and documents within the same time frame. The resulting dataset is split 80/20 into training and testing partitions using a fixed random seed to ensure fair evaluation and prevent memorization.

To train the model to recognize time-specific information, we use the baseline Contriever (facebook/contriever-msmarco) to build a FAISS inner-product index over the synthetic passages. For each training question, the top-K most semantically similar passages are retrieved. Passages that share the same extracted year serve as positives, while those with different years act as hard negatives. This forms triplets that encourage the model to distinguish between semantically similar but temporally inconsistent passages.

The retriever is then fine-tuned using MarginRankingLoss with a margin of 1.0, employing mean-pooled and L2-normalized embeddings under cosine similarity. Training runs on GPU with mixed precision and a linear warm-up schedule. To preserve broad retrieval competence, MSMARCO triplets are interleaved 1:1 during fine-tuning. This combination yields a balanced retriever that learns temporal discrimination while retaining robust general-purpose retrieval performance.

5 Potential Datasets

5.1 TempRAGEval

Temporal Question Answering for RAG Evaluation is a benchmark dataset that can be used to rigorously evaluate the time-sensitive question answering on RAG systems. It was introduced in MRAG: Modular Retrieval Framework for Time Sensitive Questions" (Zhang and Choi, 2021). The dataset is a mix of existing datasets used for training temporal RAGs, and it maintains a natural human-written language style of question answering. It has about a thousand test instances, with each question having a manually annotated gold evidence sentence obtained from Wikipedia. The authors of this dataset added temporal perturbations by changing time constraints in the questions to check the robustness to temporal changes. The dataset intends to test both the retrieval performance and the question answering performance.

5.2 ChroniclingAmericaQA

Additionally, we will evaluate our models on ChroniclingAmericaQA (Piryani et al., 2024), a large-scale temporal QA benchmark based on historical American newspaper pages spanning 120 years. The dataset's explicit publication timestamps make it ideal for testing our metadata-aware retrieval method's ability to filter candidates by temporal validity and assess whether incorporating temporal constraints improves retrieval accuracy for time-sensitive questions.

5.3 Experimental Settings

Datasets: We will train and evaluate our models on the Motion and Torque (MT) datasets, as well as TempEval-3. These datasets are specifically chosen to assess a model's ability to handle fine-grained temporal relations and events. The data will be split into an 80/20 train/test set.
Hardware: Experiments will be conducted on an NVIDIA A40 GPU.
Hyperparameters: Both models will be trained for 3 epochs with a batch size of 8 and a learning rate of 2e-5. The proposed model's time embedding dimension will be 32.

5.4 Experiments and Evaluation

We will conduct two main experiments to validate our hypothesis:
Retrieval Performance: We will measure the models' ability to retrieve the correct context using Top-1 and Top-5 accuracy on the held-out test sets.
End-to-End Exact Match (EM): We will create a two-stage pipeline where the trained DPR models act as the retriever, and a separate, pre-trained reader model (e.g., DistilBERT) extracts the final answer. We will then calculate the Exact Match (EM) score to measure the pipeline's ability to produce the correct answer string.

6 Plan of Activities

Week 1: Building the New Temporal Retriever The first week focuses on designing and implementing a new retriever variant that extends the existing temporal hard mining framework. The objective is to incorporate richer temporal signals and improve how the model captures temporal distinctions in the embedding space. This involves integrating enhanced year-based negatives, improved temporal embeddings, and interval-aware positives as identified in the proposed methodological extensions.

The retriever will be trained end-to-end using the synthetic question–passage pairs generated by the T5 model, combined with the MSMARCO mix-in data for broader generalization. Small-scale experiments and sanity checks will be conducted to verify stable training dynamics and confirm that the learned embeddings meaningfully encode temporal separations. By the end of this phase, the project aims to have a fully functional and trainable temporal retriever ready for evaluation.

Week 2: Evaluation and Analysis of the Temporal Retriever The second week is dedicated to systematically evaluating the new retriever across both in-domain and out-of-domain datasets. Specifically, quantitative experiments will be performed on the T5-Split dataset for in-domain assessment, and on ChroniclingAmericaQA and TempRAGEval for out-of-domain testing. Standard retrieval metrics such as Hit@K, MRR@K, and Recall@K will be recorded to quantify the model's improvements over both the baseline Contriever and earlier fine-tuned variants.

A detailed error analysis will follow, focusing on retrieved passages to verify that the model successfully identifies correct temporal contexts while avoiding year mismatches. Insights from these analyses will guide the selection of the most effective retriever variant for integration into the MRAG pipeline. The goal for this phase is to establish a clear understanding of the model's standalone performance, its generalization capabilities, and its robustness in temporal reasoning.

Week 3: Integration with MRAG and Joint Evaluation In the third week, the fine-tuned temporal retriever will be integrated into the MRAG framework as the first-stage retriever. This integration will test whether time-aware retrieval improves the overall reasoning and question-answering performance of the MRAG pipeline. Comparative evaluations will be conducted to measure how MRAG's re-ranking module behaves when provided with temporally informed candidate passages versus those retrieved by the standard Contriever.

Comprehensive system-level metrics will be collected to quantify the compounded benefits of combining temporal retrieval with MRAG's semantic-temporal re-ranking. Additional diagnostic analyses will trace how temporal signals propagate through the pipeline, highlighting which stages benefit most from the inclusion of time-aware candidates. The overarching goal is to validate that the integration meaningfully enhances downstream RAG reasoning and boosts question-answering accuracy across time-sensitive benchmarks.

Week 4: Visualization, Result Consolidation, and Reporting The final week centers on consolidating results, visualizing performance trends, and preparing the final report. The team will generate tables and plots summarizing retrieval metrics, ablation comparisons, and improvements across datasets. Qualitative visualizations will illustrate how retrieved evidence changes over time-sensitive questions, providing an intuitive understanding of the model's temporal reasoning behavior.

The written report will synthesize experimental findings, emphasizing which configurations performed best, under what conditions, and why. Finally, the codebase, configuration files, and documentation will be organized for reproducibility and transparency. The goal for this phase is to deliver a polished, comprehensive experimental study with clearly articulated insights and verifiable results.

Work Division The project responsibilities are distributed among four team members to ensure balanced workload and specialized focus areas. Person 1 leads the fine-tuning pipeline, hyperparameter optimization, and FAISS infrastructure setup. Person 2 is responsible for developing and testing new temporal-mining mechanisms, including graded margin loss and overlap-based negative sampling logic. Person 3 handles the integration of the temporal retriever into the MRAG re-ranking pipeline and develops evaluation scripts for joint analysis. Person 4 focuses on result analysis, visualization, dataset diagnostics, and assembling the final report. This structured division of work ensures parallel progress across implementation, experimentation, integration, and analysis tasks.

References

Abdelrahman Abdallah, Bhawna Piryani, Ashmit Mishra, Franck Dernoncourt, Rodney Khan, and Avirup Sil. 2025. Tempretriever: Fusion-based temporal dense passage retrieval for time-sensitive questions. *Preprint*, arXiv:2502.21024.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Preprint*, arXiv:2004.04906.

Kwun Hang Lau, Ruiyuan Zhang, Weijie Shi, Xiaofang Zhou, and Xiaojun Cheng. 2025. Reading between the timelines: Rag for answering diachronic questions. *Preprint*, arXiv:2507.22917v1.

Bhawna Piryani, Abdelrahman Abdallah, Jamshid Mozafari, Avishek Anand, and Adam Jatowt. 2025. It's high time: A survey of temporal question answering. *Preprint*, arXiv:2505.20243.

Bhawna Piryani, Jamshid Mozafari, and Adam Jatowt. 2024. Chroniclingamericaqa: A large-scale question answering dataset based on historical american newspaper pages. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2024, page 2038–2048. ACM.

Zhang Siyue, Xue Yuxiang, Zhang Yiming, Wu Xiaobao, Luu Anh Tuan, and Zhao Chen. 2025. Mrag: A modular retrieval framework for time-sensitive question answering. *Preprint*, arXiv:2412.15540.

Jiexin Wang, Adam Jatowt, and Masatoshi Yoshikawa. 2022. Archivalqa: A large-scale benchmark dataset for open domain question answering over historical news collections. *Preprint*, arXiv:2109.03438.

Feifan Wu, Lingyuan Liu, Wentao He, Ziqi Liu, Zhiqiang Zhang, Haofen Wang, and Meng Wang. 2024. Time-sensitive retrieval-augmented generation for question answering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24)*, pages 2544–2553. ACM.

Michael J. Q. Zhang and Eunsol Choi. 2021. Situatedqa: Incorporating extra-linguistic contexts into qa. *Preprint*, arXiv:2109.06157.

Jiawei Zhou and Lei Chen. 2025. R3: Towards optimal retrieval for rag through trial-and-feedback reinforced contrastive learning. *Preprint*, arXiv:2510.24652.