Time Aware Retriever - Injecting Temporal Information using Negative Hard Mining

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

The advent of Retrieval-Augmented Generation (RAG) has fundamentally shifted the paradigm of Natural Language Processing from static, parametric knowledge storage to dynamic, retrieval-based reasoning. By combining a neural retriever which identifies relevant documents from a massive external corpus with a generative reader, RAG systems can provide answers that are factually grounded and up-to-date. This architecture has become the gold standard for knowledge-intensive tasks, powering applications ranging from search engines to conversational assistants.

However, a critical and unaddressed limitation of current RAG frameworks including state-of-the-art dense retrievers like Dense Passage Retrieval (DPR) and Contriever is their reliance on purely semantic similarity. These models optimize for topical relevance, treating information as static. They lack an inherent mechanism to model the temporal context that defines the validity of many facts.

1.1 Motivation: The Temporal Gap

As the world evolves, facts change. The answer to the query *"Who is the CEO of Twitter?"* is contingent entirely on the timestamp of the query. In 2015, the answer was Dick Costolo; in 2021, Jack Dorsey; and in 2023, Linda Yaccarino.

Standard semantic retrievers fail in these dynamic scenarios due to two primary failure modes:

1. **Temporal Hallucination:** The model retrieves the most statistically prominent entity in the training corpus regardless of the current timeframe. For example, a model pre-trained on 2018 Wikipedia data may persistently retrieve "Theresa May" for queries about the "UK Prime Minister," overpowering any newer documents about subsequent leaders because the semantic signal for the former is stronger in its weights.
2. **Right Fact, Wrong Time:** The model retrieves a document that correctly identifies the entity but in the wrong temporal context. For a query regarding *"The winner of the 2010 World Cup"*, a standard retriever might surface a document describing the *2014* World Cup simply because the document contains high-frequency keywords like "World Cup," "winner," and "final," ignoring the crucial distinctifier "2010."

Neglecting this temporal dimension leads to logical inconsistencies that severely limit the utility of RAG systems in dynamic domains such as news monitoring, financial forecasting, and historical archival search.

1.2 Comparison with Prior Work

Prior attempts to bridge this temporal gap generally fall into two categories: metadata filtering and architectural modification. Metadata-based approaches rely on explicit, structured timestamps attached to documents. While effective for curated datasets, this method fails on unstructured web corpora where metadata is often missing, ambiguous, or incorrect. Architectural approaches, such as TempRetriever or BiTimeBERT, attempt to solve this by introducing specialized time encoders or concatenating explicit time vectors to text embeddings. However, these custom architectures are complex to train and difficult to adapt to pre-existing dense retrieval pipelines.

In contrast, our work maintains the simplicity of the standard bi-encoder architecture (Contriever). We inject temporal awareness solely through data-driven contrastive fine-tuning, allowing the model to learn implicit temporal cues from the text itself without requiring structured metadata or architectural changes. Furthermore, while the original Modular RAG (MRAG) framework relies on computationally expensive LLM summarization

for re-ranking, we introduce a lightweight "Sliding Window MaxSim" mechanism. This allows us to achieve fine-grained temporal precision without the high latency cost associated with generative re-ranking.

1.3 Contributions

In this report, we bridge this "Temporal Gap" by proposing a complete, end-to-end pipeline for time-aware retrieval. Our approach moves beyond simple metadata filtering by embedding temporal reasoning directly into the dense retriever itself.

Our specific contributions are as follows:

- **Synthetic Temporal Supervision:** We address the scarcity of labeled temporal training data by developing a robust data generation pipeline. Using a T5-base model, we generate 15,000 high-quality "temporally anchored" question-passage pairs from the FineWeb-Edu corpus, specifically conditioning questions on explicit year markers found in the text.

- **Temporal Hard Negative Mining:** We introduce a fine-tuning objective for the Contriever model that utilizes "Temporal Hard Negatives." By mining negative passages that are semantically identical to the positive passage but refer to a conflicting timeframe, we force the model to learn time as a critical discriminative feature in the embedding space.

- **Optimized MRAG Implementation:** We re-implement the Multi-Stage RAG (MRAG) framework, optimizing it for inference speed. Instead of the computationally expensive LLM-based summarization proposed in original works, we devise a "Sliding Window MaxSim" algorithm that pre-computes granularity scores, balancing the precision of sentence-level ranking with the scalability of dense retrieval.

- **Comprehensive Evaluation:** We evaluate our system on three distinct benchmarks: an in-domain T5 test set, the out-of-domain historical ChroniclingAmericaQA dataset, and a temporal subset of SQuAD (Rajpurkar et al., 2016). This multi-faceted evaluation tests both the model's ability to learn specific patterns and its generalization to noisy, real-world archival data.

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. These models 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. Experimental results demonstrate that MRAG achieves significant improvements in gold evidence recall and question answering accuracy, establishing it as a robust framework for time-sensitive reasoning without requiring expensive retraining of the underlying language models.

2.3 ChroniclingAmericaQA

ChroniclingAmericaQA (Piryani et al., 2024) introduces a large-scale question answering dataset derived from historical American newspaper pages spanning 1789 to 1963. Unlike standard benchmarks that rely on clean, encyclopedic text, this dataset presents unique challenges including OCR noise, archaic language, and evolving historical contexts. Crucially for our work, the dataset necessitates strong temporal reasoning, as answers are strictly contingent on the publication date of the newspaper. Previous baselines, including BM25 and standard dense retrievers, struggle significantly

on this dataset due to the "temporal drift" and noise inherent in archival documents. We utilize ChroniclingAmericaQA as a rigorous Out-Of-Domain (OOD) benchmark to test whether our Time-Aware Contriever can generalize to these challenging historical queries.

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 on datasets like Natural Questions, influencing later dense retrieval systems that combine neural search with language models.

2.5 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 to enhance relevance (Zhou and Chen, 2025) and temporal extensions that integrate time-aware signals to improve recall and precision on temporal datasets (Abdallah et al., 2025).

2.6 TempRetriever

TempRetriever (Abdallah et al., 2025) introduces a fusion-based architecture designed specifically for time-sensitive question answering. Unlike standard dense retrievers that treat time as implicit text, TempRetriever employs a dual-stream encoding mechanism: one stream processes the semantic content of the passage, while a parallel stream explicitly encodes temporal metadata (timestamps and intervals). These signals are then fused to produce a final relevance score. While this approach achieves high precision on datasets with structured metadata like TimeQA, it heavily relies on the availability of accurate timestamp tags. In our work, we aim to achieve similar temporal awareness through implicit contrastive learning, removing the dependency on external metadata structures.

3 Proposed Methods

Our implementation establishes a complete end-to-end pipeline for time-aware retrieval, consisting of three distinct stages: synthetic data generation using T5, contrastive fine-tuning of the Contriever model, and multi-stage re-ranking (MRAG). This architecture allows us to inject temporal reasoning capabilities directly into the dense retriever while leveraging hybrid re-ranking for final precision.

3.1 Stage 1: Synthetic Data Generation (T5)

A major bottleneck in temporal information retrieval is the lack of large-scale, annotated training pairs where the query explicitly targets a specific timeframe. To address this, we constructed a synthetic dataset pipeline using the `FineWeb-Edu` corpus.

3.1.1 Passage Selection and Filtering

We streamed the `HuggingFaceFW/fineweb-edu` (sample-10BT) corpus, a high-quality dataset filtered for educational content. We processed a stream of 500,000 passages, applying a strict regex filter to retain only passages containing explicit 4-digit year tokens (e.g., "1995", "2020") or temporal prepositions. This step ensures that our source text is rich in temporal facts and reduces the noise from atemporal content.

3.1.2 Conditional Question Generation

We employed a T5-base model fine-tuned for question generation (`valhalla/t5-base-qg-hl`). For each selected passage P, we extracted the temporal entity T (the year) and constructed a prompt:

$$\text{Prompt} = \text{"generate question about } T : P\text{"}$$

We fed this prompt to the T5 model to generate a question Q. This conditioning is crucial; it forces the model to generate a question where the answer is dependent on the context provided by T.

$$Q = \text{T5}_{\text{gen}}(P, \text{condition} = T)$$

This process yielded approximately 15,000 high-quality *(question, temporal_passage)* pairs. Unlike generic questions, these synthetic queries are

"temporally anchored," meaning they often contain explicit year mentions or context that binds them to the passage's timeframe. Table 1 presents qualitative examples of these generated pairs, demonstrating the model's ability to extract specific temporal facts from diverse web text.

3.2 Stage 2: Time-Aware Contriever Fine-Tuning

We utilized `facebook/contriever-msmarco` as our base encoder. To inject temporal awareness, we fine-tuned the model using a contrastive loss objective with a novel hard-negative mining strategy.

3.2.1 Temporal Hard Negative Mining

Standard hard negatives in retrieval are passages that are semantically similar to the query but do not contain the answer. We extended this definition to include "Temporal Negatives." For a generated query Q targeting year Y_{target}, we mined negative passages P^- from our corpus that met two criteria:

- **High Semantic Similarity:** The vector dot product between the query and the negative passage was high (using the base Contriever model).
- **Temporal Mismatch:** The passage contained a year Y_{neg} such that $Y_{neg} \neq Y_{target}$.

Mining Constraints: To ensure the mined negatives were challenging yet distinct, we enforced strict filtering criteria. A passage was selected as a hard negative only if its semantic similarity score with the query (computed via the base Contriever) exceeded a threshold of **0.7**. This high threshold guarantees that the negative is topically relevant and not merely a random document. For each query in our training set, we mined up to **3 positive passages** (sharing the target timestamp) and **6 hard negative passages** (conflicting timestamps). This is to provide the contrastive loss function with sufficient discriminatory signals without overwhelming the batch with noise.

This mining strategy creates training triplets (Q, P^+, P^-) where P^+ and P^- are often textually very similar (e.g., both discussing "Super Bowl winners") but refer to different years. This forces the model to push apart representations of similar events that occurred at different times.

3.2.2 Triplet Loss Objective

Instead of the standard InfoNCE loss often used in pre-training, we optimized the model using the Margin Ranking Loss (Triplet Loss). This objective focuses specifically on the relative ranking of candidates, ensuring that the similarity score of the query Q with the temporally correct passage P^+ exceeds that of the conflicting passage P^- by at least a fixed margin α.

The loss function for a single triplet (Q, P^+, P^-) is defined as:

$$\mathcal{L} = \max(0, \text{sim}(Q, P^-) - \text{sim}(Q, P^+) + \alpha)$$

where $\text{sim}(u, v)$ is the dot product similarity and α is the margin (set to 1.0 in our experiments). This approach is particularly effective for fine-tuning because it provides a direct, hard signal to the gradients whenever the model fails to distinguish between the correct year and the conflicting year, explicitly correcting "Right Fact, Wrong Time" errors.

3.3 Stage 3: MRAG with Sliding Window MaxSim

The final stage is a re-ranking module designed to refine the top-K results from the dense retriever. We implemented an optimized version of the MRAG framework that avoids expensive LLM calls during inference.

3.3.1 Temporal Decomposition

We implemented a regex-based query parser that separates the user's input into two components:

- *Main Content (q_{mc}):* The core semantic intent stripped of temporal markers (e.g., "winner of the World Cup").
- *Temporal Constraint (T_c):* The specific time expression (e.g., "in 2014", "between 1990 and 2000").

3.3.2 Sliding Window MaxSim

Standard dense retrievers encode entire passages into a single vector, often diluting specific details. To capture fine-grained details in long documents, we segment each candidate document d into overlapping windows $w_1, \ldots, w_n$ (size=3 sentences, stride=1). We pre-compute embeddings for all windows using the fine-tuned encoder.

The semantic relevance score is defined as the maximum similarity between the Main Content query and any window in the document:

$$S_{sem}(q, d) = \max_i \cos(\mathbf{E}(q_{mc}), \mathbf{E}(w_i))$$

Passage Context (Excerpt)	T5 Generated Question
...The location of the Tarentum is indicated primarily by the discovery in **1930** of the inscribed record of the Saecular Games...	When was the inscribed record of the Saecular Games discovered?
The meteorite that exploded over Chelyabinsk on **February 15, 2013**, was a pretty big thing - NASA estimated that it was about 15-17 meters...	When did the meteorite explode over Chelyabinsk?
...The **1972** Summer Olympics were held in Munich... The Games were overshadowed by an act of terrorism known as the "Munich Massacre."...	What year was the 20th Summer Olympics held?
Scientists first started recording alarming declines in bees in North America in **2006**... Known as Colony Collapse Disorder...	In what year did scientists begin to record declines in bee populations?

Table 1: Qualitative examples of synthetic training pairs generated by our T5 pipeline. The model successfully identifies specific temporal markers (bolded) in the FineWeb-Edu text to formulate precise, time-anchored questions.

This "MaxSim" operation ensures that a single highly relevant sentence can trigger a high score, even if the rest of the document is irrelevant or noisy.

3.3.3 Hybrid Scoring and Fusion

The final ranking score combines the granular semantic score with a temporal validity penalty. We use a triangular decay function based on the distance between the query's constraint T_c and the years found in the document Y_d. The temporal score S_{temp} is calculated as:

$$S_{temp} = \max\left(0, 1 - \frac{\min_{y \in Y_d} |y - T_c|}{\lambda}\right)$$

where λ is a decay constant (set to 20 years). The final score is a multiplicative fusion:

$$S_{final} = S_{sem} \times S_{temp}$$

This multiplicative fusion effectively acts as a soft filter, aggressively down-ranking documents that are semantically relevant but chronologically invalid (where $S_{temp} \approx 0$).

4 Experimental Settings

4.1 Datasets

We utilized three datasets corresponding to training, out-of-domain evaluation, and reading comprehension evaluation.

- **FineWeb-Edu (Training):** We utilized the `HuggingFaceFW/fineweb-edu` (sample-10BT) corpus. We streamed a subset of **500,000 passages** and filtered them for temporal markers. From this pool, we generated **15,000 synthetic pairs**, which were split into **12,000 for training** and **3,000 for held-out in-domain testing**.
- **ChroniclingAmericaQA (CAQA):** We used the complete validation split of this historical dataset, comprising **24,111 questions**. From this, we identified a year-explicit subset of **1,219 questions** to serve as the primary metric for temporal generalization. This dataset challenges the model with noisy OCR text and archaic language. Table 2 provides examples of these queries, illustrating the strict date-based constraints required to answer them correctly.

Example Questions from ChroniclingAmericaQA
Who was the president of the United States of America in **1808**?
What was the town of Mansfield called in **1801**?
Whose personal property will be sold on **March 16, 1804**?
Who sent a letter to President Smith, **October 01, 1801**?

Table 2: Sample time based questions from the ChroniclingAmericaQA dataset. Unlike standard trivia, these queries often require precise alignment with specific historical dates found in archival newspaper text.

- **SQuAD (Temporal Subset):** We filtered the SQuAD validation set using regex patterns to isolate strictly temporal questions (containing explicit years or starting with "When"), resulting in a focused subset of **1,000 questions**.
- **MS MARCO (Regularization):** To mitigate catastrophic forgetting and preserve the model's ability to handle general semantic queries, we integrated general-domain triplets from the MS MARCO dataset into our fine-tuning mix. By interleaving these standard retrieval examples with our synthetic temporal pairs, we ensured the model maintained its general-purpose retrieval capabilities. The effectiveness of this strategy is empirically validated by the model's superior performance

on the *full* ChroniclingAmericaQA dataset (24,111 questions), which contains a diverse array of general knowledge queries beyond just the temporal subset.

4.2 Models in Comparison

We compare four distinct configurations to isolate the benefits of each component:

1. **Base Contriever:** The pre-trained `facebook/contriever-msmarco` model. This serves as the zero-shot baseline.
2. **Time-Aware Contriever:** Our fine-tuned model trained on the T5 synthetic data using temporal hard negatives.
3. **MRAG (Base):** The Base Contriever coupled with MRAG re-ranker. This tests if re-ranking alone can solve the problem without fine-tuning. It also serves as a baseline for MRAG-coupled architecture.
4. **MRAG (Time-Aware):** The fully integrated pipeline, combining the fine-tuned retriever with the MRAG re-ranker.

4.3 Hyperparameters and Preprocessing

Preprocessing: All text was normalized to lowercase and stripped of non-alphanumeric characters for year extraction. For MRAG, documents were sentence-tokenized using NLTK before windowing.

Fine-Tuning: To ensure stable convergence on the synthetic data, we trained for **14 epochs** with a learning rate of $1e^{-5}$ using the AdamW optimizer. We used a micro-batch size of 32 and gradient accumulation steps of 8, resulting in an effective batch size of 256. The triplet loss margin was set to 1.0.

Inference: Retrieval was performed with $K = 100$. The MRAG sliding window size was set to 3 sentences with a stride of 1. The temporal decay constant λ was set to 20 years.

Hardware: All experiments were conducted on a single NVIDIA A40 GPU. The fine-tuning process took approximately 4 hours, and the full evaluation suite took approximately 2 hours.

5 Experimental Results and Analysis

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

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

5.1 In-Domain Performance (T5 Test Set)

We first evaluate performance on the held-out test set generated by T5 (3,000 pairs). This measures the model's ability to learn the specific temporal patterns present in the training distribution.

Variant	Hit@1	Hit@5	Hit@10	MRR@10
Base	0.784	0.874	0.900	0.823
Time-Aware	**0.849**	**0.922**	**0.942**	**0.882**

Table 3: Results on T5 In-Domain Test Set.

As shown in Table 3, the Time-Aware Contriever achieves a substantial improvement ($\sim$6.5% in Hit@1) over the baseline. The Hit@5 score reaches an impressive 0.922, compared to 0.874 for the baseline. This confirms that the contrastive fine-tuning successfully taught the model to prioritize temporally aligned passages. The high scores also indicate that the T5 generation pipeline produced clean, learnable signals.

5.2 Out-of-Domain Generalization (CAQA)

The ChroniclingAmericaQA dataset represents a significantly harder challenge due to domain shift. We evaluated on the year-explicit subset (1,219 questions).

Configuration	Hit@1	Hit@5	Hit@10	MRR@10
Base Only	0.404	0.572	0.638	0.477
Time-Aware	0.478	0.667	0.715	0.558
MRAG (Base)	0.573	0.735	0.768	0.642
MRAG (TA)	**0.591**	**0.753**	**0.793**	**0.661**

Table 4: Results on ChroniclingAmericaQA (CAQA Year-Subset).

Generalization to Non-Temporal Queries: To verify that our temporal fine-tuning did not degrade the model's general retrieval capabilities (catastrophic forgetting), we also evaluated performance on the **full** ChroniclingAmericaQA dataset (24,111 questions), which contains a mix of temporal and general knowledge queries. On this comprehensive set, the Time-Aware Contriever achieved a **Hit@1 of 0.504**, outperforming the Base Contriever's **0.478**. This improvement (+2.6% absolute) confirms that the integration of MS MARCO triplets during training successfully preserved the model's semantic understanding, allowing it to generalize well even to queries that do not require strict temporal reasoning.

Configuration	Hit@1	Hit@5	Hit@10	MRR@10
Base Only	0.478	0.656	0.716	0.555
Time-Aware	**0.504**	**0.680**	**0.740**	**0.579**

Table 5: Results on the **Full** ChroniclingAmericaQA Dataset (24,111 questions). The Time-Aware model outperforms the baseline even on general queries, confirming that general retrieval capability was preserved.

Table 4 and 5 illustrate several key findings:

- **Fine-tuning Generalizes:** Even without MRAG, the Time-Aware model outperforms the Base model (Hit@1 0.478 vs 0.404). This suggests the model learned a generalizable notion of "time" rather than just memorizing FineWeb patterns.
- **MRAG is Critical:** The addition of MRAG re-ranking provides the most dramatic boost. The **MRAG (Time-Aware)** configuration achieves a Hit@1 of 0.591, a nearly 19% absolute improvement over the vanilla Base Contriever.
- **Precision vs. Recall:** The gap between Hit@1 and Hit@10 is smaller for the MRAG models, indicating that the re-ranker effectively pushes the correct answer to the top of the list, improving the Mean Reciprocal Rank (MRR) significantly (0.661 vs 0.477).

This suggests that for noisy, historical data, a two-stage approach is essential. The fine-tuned retriever improves recall (getting the document into the top-100 candidates), while the MRAG sliding window re-ranker improves precision (finding the specific relevant sentence and verifying the date).

5.3 Analysis of Temporal SQuAD

Finally, we analyzed performance on the temporal subset of SQuAD (1,000 questions) to assess reading comprehension capability on encyclopedic text.

Configuration	Hit@1	Hit@5	MRR@10
Base Only	0.717	0.921	0.804
MRAG (Base)	0.783	**0.943**	0.850
MRAG (Time-Aware)	**0.785**	0.942	**0.854**

Table 6: Results on SQuAD Temporal Subset.

On SQuAD (Table 6), the baseline Contriever is already quite strong (Hit@5 = 0.921), likely because SQuAD's Wikipedia domain overlaps significantly with Contriever's original pre-training data. However, MRAG still squeezes out performance gains, pushing Hit@5 to 0.942. Interestingly, the difference between Base and Time-Aware models is smaller here than on CAQA. This indicates that while our fine-tuning helps significantly on unseen/noisy domains (CAQA), the base Contriever is already quite competent on clean Wikipedia text. The MRAG re-ranking, however, remains beneficial across all domains.

6 Conclusion and Discussion

In this report, we successfully designed, implemented, and evaluated a comprehensive Time-Aware RAG system aimed at resolving the critical "Temporal Gap" in dense retrieval. The central hypothesis of our work was that standard semantic retrievers, despite their power, fail to treat time as a first-class discriminative feature. By generating high-quality synthetic training data with T5 and employing a novel contrastive fine-tuning strategy with Temporal Hard Negative Mining, we successfully forced the retriever to learn time as an intrinsic embedding feature rather than just a metadata tag.

6.1 Strengths and Impact

The primary contribution of this work is proving that implicit temporal reasoning can be injected into dense retrievers through data-driven fine-tuning.

- **Robustness:** The superiority of the Time-Aware Contriever over the Base Contriever (e.g., a 10% relative gain in Hit@1 on ChroniclingAmericaQA) confirms that our hard negative mining effectively penalized "Right Fact, Wrong Time" errors.

- **Efficiency:** By replacing standard LLM-based summarization with our pre-computed Sliding Window MaxSim, we achieved state-of-the-art results without the high latency of generative re-ranking, making the approach feasible for real-time applications.

6.2 Limitations

Despite these successes, the system currently faces three key limitations:

- **Regex Brittleness:** Our current approach relies on explicit regex parsing for year extraction. This makes the system brittle when dealing with relative time references (e.g., "two years after the election"), intervals (e.g., "the 90s"), or implicit eras (e.g., "during the Obama administration"). A document discussing the "2008 financial crisis" without explicitly stating "2008" might be missed by our current scoring function.
- **Windowing Overhead:** While faster than LLM summarization, the Sliding Window approach increases index size by a factor of N (windows per document). For extremely large corpora, this storage cost may become non-trivial.

6.3 Future Directions

Future iterations of this research will focus on moving beyond explicit year matching:

- **Implicit Temporal Reasoner:** We plan to train a lightweight cross-encoder capable of inferring time intervals from context without relying on explicit digits. This would allow the model to map phrases like "post-war period" to specific year ranges for more accurate relevance scoring.
- **End-to-End Generation:** We aim to close the loop by integrating a generative reader (e.g., Llama-3) to evaluate end-to-end Exact Match (EM) scores. This will allow us to verify if better retrieval ranking directly translates to more accurate final answers for the user, rather than just improved recall metrics.

6.4 Detailed Contribution of Group Members

This project was a collaborative effort with distinct technical responsibilities assigned to each member:

This project was a collaborative effort with distinct technical responsibilities assigned to each member:

- **Sonith Bingi:** Implemented the Temporal Hard Negative Mining strategy, specifically designing the logic to identify passage pairs with high semantic overlap but conflicting timestamps to optimize the Margin Ranking Loss. He also architected the T5 synthetic data generation pipeline, ensuring the model produced "temporally anchored" questions conditional on explicit year markers found in FineWeb-Edu, and led the performance analysis on the ChroniclingAmericaQA dataset.
- **Manoj Arulmurugan:** Led the end-to-end implementation and integration of the Modular RAG (MRAG) framework, ensuring the retrieval and re-ranking stages interfaced correctly. He conducted the initial feasibility analysis of the ATLAS corpus but identified critical domain mismatches with our temporal benchmarks; this analysis re-focused the strategic pivot to the FineWeb-Edu corpus to maximize the density of temporal facts in our training data.
- **Prasad Jawale:** Integrated MS MARCO triplets into the training loop to mitigate catastrophic forgetting in the encoder. He also assembled the final repository structure and optimized the MRAG execution speed by implementing the Sliding Window MaxSim algorithm, which replaced expensive real-time encoding with efficient pre-computed window embeddings.
- **Gauri Patki:** Explored metadata-aware retrieval approaches to establish a strong baseline for comparison against dense retrieval. She explored the SQuAD dataset to curate the temporal subset and developed the robust evaluation suite used across all experiments, writing the scripts to calculate Hit@K and MRR metrics, and implemented the precise regex filtering logic required to extract year-explicit subsets from both SQuAD and ChroniclingAmericaQA for fair benchmarking.

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.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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.

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