

Efficient Temporal-aware Matryoshka Adaptation for Temporal Information Retrieval

Tuan-Luc Huynh¹ Weiqing Wang¹ Thuy-Trang Vu¹
Trung Le¹ Dragan Gašević² Yuan-Fang Li¹ Thanh-Toan Do¹

¹ Department of Data Science & AI, Monash University, Australia

² Department of Human Centred Computing, Monash University, Australia

{tuan.huynh1, teresa.wang, trang.vu1,

trunglm, dragan.gasevic, yuanfang.li, toan.do}@monash.edu

Abstract

Retrievers are a key bottleneck in temporal Retrieval-Augmented Generation (RAG) systems: failing to retrieve temporally relevant context can degrade downstream generation, regardless of LLM reasoning. We propose Temporal-aware Matryoshka Representation Learning (TMRL), an efficient method that equips retrievers with temporal-aware Matryoshka embeddings. TMRL leverages the nested structure of Matryoshka embeddings to introduce a temporal subspace, enhancing temporal encoding while preserving general semantic representations. Experiments show that TMRL efficiently adapts diverse standard encoder-only text embedding models, achieving competitive temporal retrieval and temporal RAG performance compared to existing Matryoshka-based adaptation and temporal retrieval methods, while enabling flexible accuracy-efficiency trade-offs.¹

1 Introduction

Retrieval-augmented Generation (RAG) (Lewis et al., 2020) is a paradigm in information retrieval where systems generate answers grounded in external knowledge (Li et al., 2025). By combining LLM reasoning with retrieved evidence, RAG tackles knowledge-intensive tasks like question answering (QA), mitigating outdated or incomplete knowledge. Temporal QA presents unique challenges that go beyond semantic relevance. Answering time-sensitive queries requires identifying relevant time spans or even reasoning over time-evolving facts (Pirayani et al., 2026). Driven by the strong reasoning capabilities of LLMs, recent research has proposed a myriad of datasets, analyses, and methods to benchmark Temporal QA (Wang and Zhao, 2024; Fatemi et al., 2025; Wallat et al., 2025; Song et al., 2025); however, they focus on LLM reasoning, overlooking the crucial role of retrieval.

¹Our code is available at <https://github.com/LouisDo2108/TMRL>.

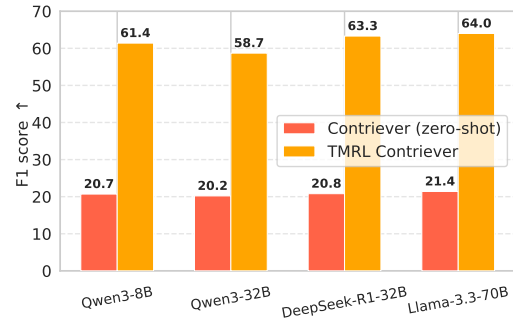


Figure 1: F1 score on Temporal Nobel Prize across LLMs of varying scales. Using larger LLMs yields negligible gains while replacing with our proposed Temporal-aware Matryoshka Representation Learning contriever consistently outperforms zero-shot contriever up to 3×. This demonstrates that the retriever is the primary bottleneck in temporal RAG performance.

We argue that in temporal RAG systems, the main bottleneck lies in the *retriever*, which is usually a text embedding model (TEM). As shown in Figure 1, popular retrievers like contriever (Izacard et al., 2022) struggle with temporal retrieval, and scaling LLMs from 8B to 70B parameters cannot compensate poor retrieval. With limited temporal encoding capability, semantic retrievers fail to distinguish outdated from current sources, constraining downstream generation to produce incomplete or hallucinated answers, regardless of the LLM’s reasoning capabilities (examples in Appendix A).

On the other hand, fine-tuning TEMs for temporal tasks often degrades semantic retrieval performance due to catastrophic forgetting (McCloskey and Cohen, 1989; Thrun, 1995; Han et al., 2025). Prior work (Wu et al., 2024; Abdallah et al., 2026) mitigates this by maintaining separate temporal and semantic retrievers with query routing, increasing both training and inference costs. Beyond effectiveness, temporal retrievers must also be *efficient* to scale with growing corpora. In

this regard, Matryoshka Representation Learning (MRL) (Kusupati et al., 2022) has emerged as a promising paradigm in modern TEMs (Lee et al., 2024; Nussbaum et al., 2025), producing hierarchically structured embeddings that support flexible truncation for accuracy-efficiency trade-offs. However, extending MRL to temporal retrieval is challenging, as temporal signals are often sparse and easily overshadowed by dominant semantic representation, particularly under aggressive embedding truncation. Moreover, naively injecting temporal supervision into Matryoshka embeddings can exacerbate the trade-off between preserving semantic retrieval quality and encoding fine-grained temporal constraints, motivating a unified and efficient temporal-aware MRL approach.

We propose *Temporal-aware Matryoshka Representation Learning (TMRL)*, an efficient adaptation method that equips TEMs with temporal-aware Matryoshka embeddings while preserving general semantic performance. Our approach is grounded in the insight that Matryoshka embeddings’ hierarchical structure can be explicitly partitioned to preserve sparse temporal signals across dimensionalities. By dedicating a compact prefix to temporal alignment and leveraging progressive contextual expansion, TMRL enables flexible index size, lower latency yet potentially near lossless temporal retrieval via funnel retrieval. To provide clean temporal supervision for TMRL, we refine the Temporal Nobel Prize (TNP) dataset (Wu et al., 2024) using a temporal data augmentation pipeline that generates paragraph-level temporal contrastive examples, reducing conflicts caused by multiple temporal expressions. We develop TMRL using our augmented TNP dataset and extend evaluation to the TimeQA dataset (Chen et al., 2021), demonstrating strong temporal retrieval performance, efficiency, and robustness. In summary, we make the following contributions:

- We propose Temporal-aware Matryoshka Representation Learning (TMRL), the first efficient, plug-and-play, temporal-aware Matryoshka adaptation method for text embedding models.
- We introduce a temporal contrastive learning augmentation pipeline that refines and augments existing temporal datasets with clean training examples for temporal information retrieval.
- Experiments on the TNP and TimeQA datasets across six popular TEMs demonstrate TMRL’s

effectiveness and efficiency for temporal retrieval and temporal RAG.

2 Preliminary

Dense Retrieval. TEMs learn an encoder $f_\theta(\cdot)$ that maps an input sequence of L tokens $q = (q_1, \dots, q_L)$ into a sequence of contextualized token representations: $f_\theta(q) = [\mathbf{h}_{[\text{CLS}]}, \mathbf{h}_1, \dots, \mathbf{h}_L] \in \mathbb{R}^{(L+1) \times d}$, where $\mathbf{h}_i \in \mathbb{R}^d$ denotes the hidden representation of the i -th token. A pooling function aggregates these token-level representations into a fixed-dimensional vector: $\mathbf{z}_q = \text{Pool}(f_\theta(q)) \in \mathbb{R}^d$.

Most TEMs are optimized with a fixed d -dimensional InfoNCE loss with in-batch negatives (Oord et al., 2018; Zhao et al., 2024). Given a query q , a positive passage p^+ , and a set of negative passages $\mathcal{N}_q = \{p_1^-, \dots, p_n^-\}$, the loss is:

$$\mathcal{L}_{\text{InfoNCE}}^{(d)}(q, p^+, \mathcal{N}_q) = -\log \frac{e^{\text{sim}^{(d)}(\mathbf{z}_q, \mathbf{z}_{p^+})/\tau}}{e^{\text{sim}^{(d)}(\mathbf{z}_q, \mathbf{z}_{p^+})/\tau} + \sum_{i=1}^n e^{\text{sim}^{(d)}(\mathbf{z}_q, \mathbf{z}_{p_i^-})/\tau}}, \quad (1)$$

where τ is a temperature hyperparameter. Let $\mathbf{x}_{\leq m}$ denote the slicing of the first m dimensions of a vector $\mathbf{x}$. The similarity function is typically cosine similarity:

$$\text{sim}^{(m)}(\mathbf{x}, \mathbf{y}) = \frac{\langle \mathbf{x}_{\leq m}, \mathbf{y}_{\leq m} \rangle}{\|\mathbf{x}_{\leq m}\| \|\mathbf{y}_{\leq m}\|}. \quad (2)$$

Matryoshka Representation Learning (MRL). MRL (Kusupati et al., 2022) endows TEMs with nested representations for adaptive accuracy-efficiency trade-offs. Formally, given a set of target dimensions $\mathcal{M} \subset [d]$, the truncated embeddings $(\mathbf{z}_q)_{\leq m}$ for all $m \in \mathcal{M}$ remain independently functional for retrieval.

3 Method

In this section, we perform a preliminary study to motivate Temporal-aware Matryoshka Representation Learning (TMRL), which introduces a temporal subspace to enhance temporal representations across Matryoshka embedding dimensions (Figure 2). Subsequently, we introduce the temporal contrastive learning augmentation pipeline, designed specifically for training TMRL (Figure 3).

3.1 Preliminary Study

We hypothesize that global pooling mechanisms in TEMs inadvertently suppress temporal signals,

Representation	contriever	semantic	gte	gte1.5	bge	bge-m3
	Mean	Mean	Mean	CLS	CLS	CLS
Joint (zero-shot) $\mathbf{z}_q$	21.50	78.01	80.47	52.17	72.63	59.62
Non-temporal $\mathbf{z}_q^S$	10.90	74.91	80.31	46.91	73.51	57.04
Temporal $\mathbf{z}_q^T$	24.47	71.66	79.60	52.77	69.60	61.40

Table 1: nDCG@10 on TNP using different TEMs with different representations. The average number of semantic and temporal tokens before mean pooling is 18.5 and 3.5, respectively.

as they are optimized for global semantic aggregation. To investigate this, we conduct a preliminary analysis on the contextualized token-level representations produced by existing TEMs.

Let $\mathcal{T}_q \subseteq \{1, \dots, L\}$ denote the set of token indices corresponding to temporal expressions in the input query. To maintain compatibility with both [CLS]-pooling and mean-pooling TEMs, we define a conditional set $\mathcal{C}$ where $\mathcal{C} = \{\text{[CLS]}\}$ for [CLS]-pooling TEMs and $\mathcal{C} = \emptyset$ otherwise. The contextualized temporal and semantic representations are defined as:

$$\mathbf{z}_q^T = \text{MeanPool}(\{\mathbf{h}_i \mid i \in \mathcal{T}_q \cup \mathcal{C}\}). \quad (3)$$

$$\mathbf{z}_q^S = \text{MeanPool}(\{\mathbf{h}_i \mid i \in (\{1, \dots, L\} \setminus \mathcal{T}_q) \cup \mathcal{C}\}). \quad (4)$$

We include the [CLS] token in both representations since it aggregates global information and may encode temporal signals. We evaluate retrieval performance on the TNP benchmark (Wu et al., 2024) using the proposed representations.

Results in Table 1 demonstrate that despite being sparse (i.e., comprising $5 \times$ fewer tokens), $\mathbf{z}_q^T$ closely matches or even outperforms $\mathbf{z}_q^S$ on TIR tasks. This reveals two key insights:

- Contextualized temporal tokens constitute a highly informative, compact signal for TIR.
- Existing TEM embeddings fail to explicitly preserve and leverage these signals.

While Standard Matryoshka adaptation (Yoon et al., 2024b) yields hierarchical embeddings, it does not strengthen temporal encoding. Conversely, naive temporal contrastive learning enhances temporal cues but does not introduce a nested structure required for Matryoshka retrieval. Therefore, ensuring consistent temporal alignment across all Matryoshka embedding dimensions is non-trivial. These observations motivate a research question:

How should we adapt TEMs with temporal-aware Matryoshka embeddings?

3.2 Temporal-aware Matryoshka Representation Learning (TMRL)

3.2.1 Matryoshka Temporal Subspace

Minimizing the InfoNCE loss maximizes a variational lower bound on mutual information (MI) between a pair of query and document representations (Poole et al., 2019). In standard MRL, this corresponds to maximizing a lower bound on the cumulative MI across all truncation scales $\sum_{m \in \mathcal{M}} I((\mathbf{z}_q)_{\leq m}; (\mathbf{z}_p)_{\leq m})$ (proof in Appendix C). However, static truncation in standard MRL degrades information (Zhang et al., 2025a). In TIR, sparse temporal cues are scattered by standard contrastive alignment, making them highly vulnerable to dimension reduction. Standard MRL does not explicitly guarantee these temporal cues are preserved in the prefix dimensions. Therefore, to explicitly preserve temporal signals across all scales, we maximize the temporal MI for query-to-passage alignment: $I(\mathbf{z}_q^T; (\mathbf{z}_p)_{\leq m})$. By the MI chain rule:

$$I(\mathbf{z}_q^T; (\mathbf{z}_p)_{\leq m}) = I(\mathbf{z}_q^T; (\mathbf{z}_p)_{\leq t}) + I(\mathbf{z}_q^T; (\mathbf{z}_p)_{(t+1):m} \mid (\mathbf{z}_p)_{\leq t}).$$

Based on this decomposition, we explicitly maximize its prefix term, $I(\mathbf{z}_q^T; (\mathbf{z}_p)_{\leq t})$, where $t \leq d$ denotes a compact temporal subspace dimension chosen to reflect the inherent sparsity of temporal cues. Setting $(\mathbf{z}_p)_{\leq t} \triangleq \mathbf{z}_p^T$ guarantees temporal fidelity at the prefix and natural inheritance at all $m > t$. As dimensionality increases, the unconstrained suffix progressively captures additional retrieval signals beyond this fixed temporal prefix. The above derivations and design symmetrically apply to the passage-to-query direction.

To operationalize this allocation, we introduce a temporal projector $\mathcal{P}: \mathbb{R}^d \rightarrow \mathbb{R}^t$ (a two-layer FFN) to compress temporal tokens $\{\mathbf{h}_i \mid i \in \mathcal{T}_q \cup \mathcal{C}\}$ into a dimensionally compatible surrogate. While inherently lossy, this projection preserves essential temporal signals while enabling efficient prefix allocation. The final projected temporal representation is obtained via mean-pooling:

$$\widehat{\mathbf{z}}_q^T = \text{MeanPool}(\{\mathcal{P}(\mathbf{h}_i) \mid i \in \mathcal{T}_q \cup \mathcal{C}\}). \quad (5)$$

This defines the temporal prefix, yielding a structured representation where each truncated embedding decomposes into a fixed temporal subspace

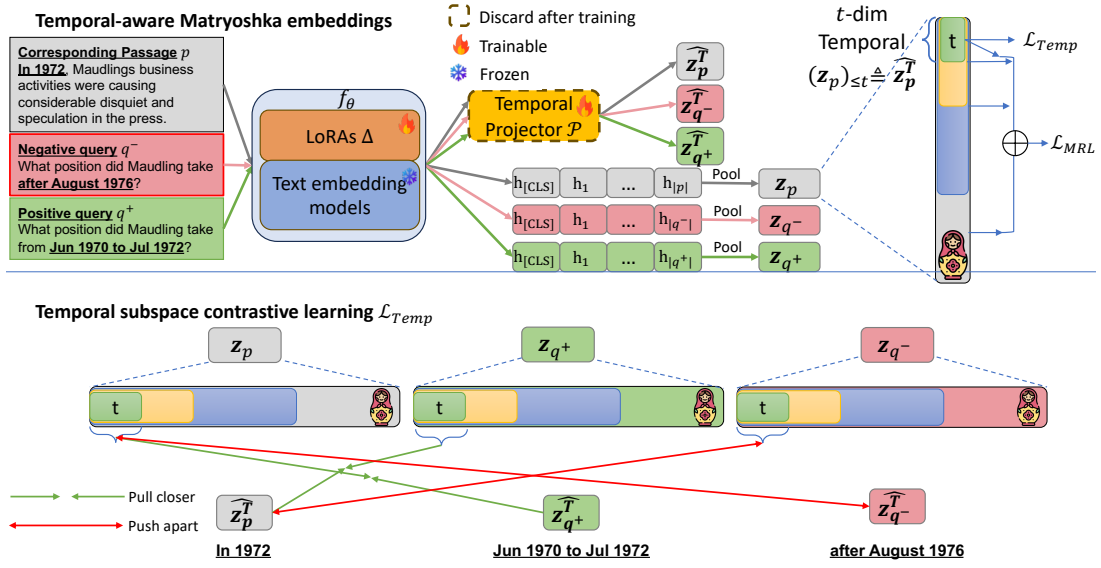


Figure 2: Overview of Temporal-aware Matryoshka Representation Learning. The temporal projector $\mathcal{P}$ learns to compress temporal tokens (Eq. 3) into a compact t -dimensional temporal representation (Eq. 5). These representations are used in the temporal subspace contrastive loss $\mathcal{L}_{\text{Temp}}$ (Eq. 8) to endow the Matryoshka embeddings with a dedicated temporal subspace, yielding the Temporal-aware Matryoshka embeddings (Eq. 7).

and a progressively expanding contextual suffix:

$$(\mathbf{z}_q)_{\leq t} := \widehat{\mathbf{z}}_q^T. \quad (6)$$

$$(\mathbf{z}_q)_{\leq m} := \left[\widehat{\mathbf{z}}_q^T \parallel (\mathbf{z}_q)_{(t+1):m} \right]. \quad (7)$$

3.2.2 Temporal Subspace Contrastive Learning

Following Wu et al. (2024), temporal contrastive learning operates on positive and negative queries rather than passages, with the negative set defined as $\mathcal{N}_p = \{q_1^-, \dots, q_n^-\}$. We optimize the temporal subspace via a dedicated contrastive loss (bottom of Figure 2). The InfoNCE loss for the t -dimensional temporal subspace is defined as:

$$\begin{aligned} \mathcal{L}_{\text{Temp}}^q(p, q^+, \mathcal{N}_p) &= -\log \frac{e^{\text{sim}^{(t)}(\widehat{\mathbf{z}}_p^T, \widehat{\mathbf{z}}_{q^+}^T)/\tau}}{e^{\text{sim}^{(t)}(\widehat{\mathbf{z}}_p^T, \widehat{\mathbf{z}}_{q^+}^T)/\tau} + \sum_{i=1}^n e^{\text{sim}^{(t)}(\widehat{\mathbf{z}}_p^T, \widehat{\mathbf{z}}_{q_i^-}^T)/\tau}}, \\ \mathcal{L}_{\text{Temp}}^p(p, q^+, \mathcal{N}_p) &= -\log \frac{e^{\text{sim}^{(t)}(\mathbf{z}_p, \widehat{\mathbf{z}}_{q^+}^T)/\tau}}{e^{\text{sim}^{(t)}(\mathbf{z}_p, \widehat{\mathbf{z}}_{q^+}^T)/\tau} + \sum_{i=1}^n e^{\text{sim}^{(t)}(\mathbf{z}_p, \widehat{\mathbf{z}}_{q_i^-}^T)/\tau}}, \\ \mathcal{L}_{\text{Temp}} &= \mathcal{L}_{\text{Temp}}^q + \mathcal{L}_{\text{Temp}}^p, \end{aligned} \quad (8)$$

where $\widehat{\mathbf{z}}_p^T$, $\widehat{\mathbf{z}}_{q^+}^T$, and $\widehat{\mathbf{z}}_{q_i^-}^T$ denote the passage, positive query, and negative query counterparts of Eq. (3).

3.2.3 Local and Structural Self-distillation

We frame MRL as self-distillation, where higher-dimensional embeddings supervise truncated counterparts (Yoon et al., 2024b). To enhance temporal-aware Matryoshka representations, we introduce an auxiliary objective that preserves local structure by aligning teacher-student nearest-neighbor similarities. Given a batch of embeddings $\mathbf{X}^{(d)} \in \mathbb{R}^{B \times d}$ and truncated counterparts $\mathbf{X}^{(m)} \in \mathbb{R}^{B \times m}$ ($d > m$), we apply a top- k pairwise similarity loss:

$$\mathcal{L}_{\text{Dist}} = \sum_{m \in \mathcal{M}} \sum_{i \neq d} \sum_{j \in \mathcal{N}} \left\| \text{sim}^{(d)}(\mathbf{x}_i, \mathbf{x}_j) - \text{sim}^{(m)}(\mathbf{x}_i, \mathbf{x}_j) \right\|_1, \quad (9)$$

where $\mathcal{N} = \arg \text{top}_k \text{sim}^{(d)}(\mathbf{x}_i, \mathbf{x}_j)$. To complement local pairwise consistency, we adopt linear Centered Kernel Alignment (CKA) (Kornblith et al., 2019) as a global structural regularizer, ensuring truncated embeddings preserve the teacher’s overall representational geometry. The linear CKA between two embedding sets can be expressed using covariance $\text{cov}(\cdot, \cdot)$ as:

$$\text{CKA}(\mathbf{X}, \mathbf{Y}) = \frac{\|\text{cov}(\mathbf{X}, \mathbf{Y})\|_F^2}{\|\text{cov}(\mathbf{X}, \mathbf{X})\|_F \|\text{cov}(\mathbf{Y}, \mathbf{Y})\|_F}. \quad (10)$$

Let us denote the centered variants of the d and m -dimensional embeddings as:

$$\begin{aligned} \tilde{\mathbf{X}}^{(d)} &= \mathbf{X}^{(d)} - \frac{1}{B} \mathbf{1} \mathbf{1}^\top \mathbf{X}^{(d)}, \\ \tilde{\mathbf{X}}^{(m)} &= \mathbf{X}^{(m)} - \frac{1}{B} \mathbf{1} \mathbf{1}^\top \mathbf{X}^{(m)}. \end{aligned} \quad (11)$$

The auxiliary Matryoshka CKA is defined as:

$$\mathcal{L}_{\text{CKA}} = \sum_{m \in \mathcal{M}, m \neq d} (1 - \text{CKA}(\tilde{\mathbf{X}}^{(d)}, \tilde{\mathbf{X}}^{(m)})) \quad (12)$$

3.2.4 Optimization Objective

TMRL-based TEMs are fine-tuned using LoRAs (Hu et al., 2022) to minimize catastrophic forgetting and enable zero-latency inference via adapter merging. The optimization objective is:

$$\mathcal{L}_{\text{TMRL}} = \mathcal{L}_{\text{MRL}} + \alpha \mathcal{L}_{\text{Temp}} + \beta \mathcal{L}_{\text{Dist}} + \gamma \mathcal{L}_{\text{CKA}}. \quad (13)$$

where α, β, γ are hyperparameters (ablation studies in Section 4.5) and $\mathcal{L}_{\text{MRL}} = \sum_{m \in \mathcal{M}} \mathcal{L}_{\text{InfoNCE}}^{(m)}$.

3.3 Temporal Contrastive Learning Augmentation

The temporal projector $\mathcal{P}$ in TMRL requires clean, unambiguous temporal expressions. However, the Temporal Nobel Prize (TNP) dataset (Wu et al., 2024) often contains passages with multiple temporal expressions (e.g., “1972” and “1966”), which introduce conflicting gradient signals during contrastive training (Figure 3, left). To address this, we refine the *train dataset* via two steps:

Passage Splitting. Using SUTime (Chang and Manning, 2012), we identify temporal expressions at the sentence level and iteratively merge adjacent sentences until a second expression would be introduced. Sentences containing multiple expressions are discarded. This yields passages with exactly one temporal expression while preserving maximal local context.

Positive/Negative Query Augmentation. We generate contrastive query-passage pairs by filtering and augmenting along three dimensions:

1. We target L2 (time–event) and L3 (event–event) understanding (Tan et al., 2023), excluding L1 (time–time) arithmetic and ordinal queries (Jia et al., 2021) that lack reliable corpus anchors.
2. We include temporal questions with *explicit*, *implicit*, and *temporal answer* (Jia et al., 2021).
3. We apply Allen’s interval relations (Allen, 1983) to explicitly control query–passage temporal alignment.

Details about the augmentation pipeline are provided in Appendix D.5. We validate the effectiveness of this pipeline in Section 4.2. It is worth noting that models are evaluated on *test datasets* consisting of unfiltered real-world documents, which

serve as stress tests for model’s robustness on passages with multiple temporal expressions.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate on Temporal Nobel Prize (TNP) (Wu et al., 2024) and extend to TimeQA (Chen et al., 2021). TNP decouples temporal retrieval from semantic matching, cleanly attributing gains to temporal modeling. In contrast, TimeQA leverages Wikipedia’s dynamic entity histories to provide a temporally realistic benchmark. Refer to Appendix E for further details about the datasets. Following BEIR (Thakur et al., 2021), we report nDCG@10 and Recall@100. To assess semantic retrieval performance, we report nDCG@10 on the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019) from BEIR using the original embedding dimension of each TEM. We further evaluate on downstream Temporal RAG, using F1 score as the evaluation metric. We focus on evaluating temporal retrieval, while multi-hop reasoning is handled downstream by an LLM. Refer to Appendix D.4 for further details about Temporal RAG setup.

Baselines. We categorize our baselines into two groups based on the evaluation objective:

- **Temporal Retrieval.** To assess temporal encoding in full-dimensional embeddings, we compare TMRL against: (i) zero-shot performance of original TEMs; (ii) BM25 (Robertson and Zaragoza, 2009); and (iii) prior dense temporal retrieval methods, including Ts-Retriever (Wu et al., 2024) and TempRetriever (Abdallah et al., 2026).
- **Matryoshka Adaptation.** To evaluate our proposed Matryoshka adaptation strategy, we compare against parameter-efficient fine-tuning adaptation approaches: Matryoshka Adaptor (M-Adaptor) (Yoon et al., 2024b) and a vanilla LoRA-based MRL baseline.

Backbones. We evaluate all methods across six open-source encoder-only TEMs (<600M parameters): *contriever* (Izacard et al., 2022), *gte-base* (gte) (Li et al., 2023), *nomic-embed-text-v1.5* (nomic) (Nussbaum et al., 2025), *gte-base-en-v1.5* (gte1.5) (Zhang et al., 2024), *bge-base-en-v1.5* (bge) (Xiao et al., 2024), *bge-m3* (Chen et al., 2024). Implementation details are provided in Appendix D.

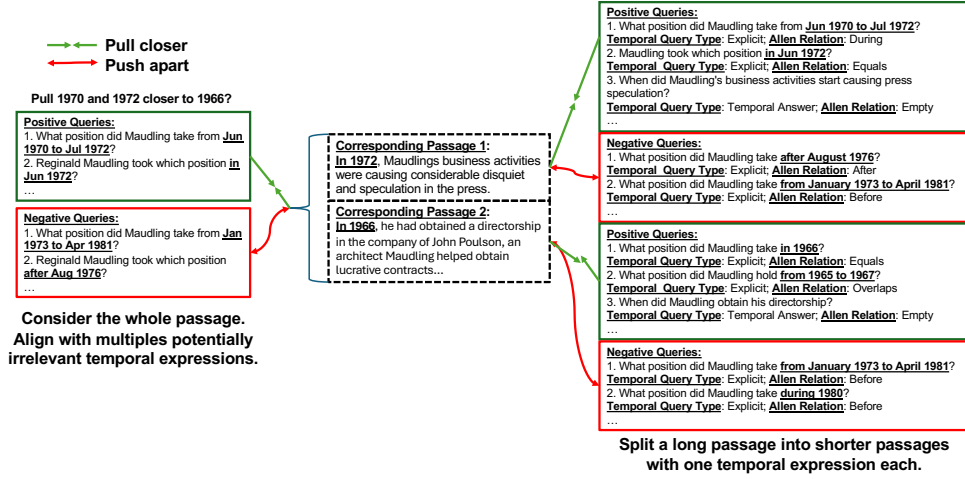


Figure 3: Comparison between the TNP’s training dataset (left) and our augmented version (right). We follow a simple principle: split a long passage into shorter passages, discard sentences with multiple temporal expressions, ensuring that each passage contains one temporal expression before generating augmented positive/negative queries.

Backbone	Train set	nDCG@10					Recall@100					BEIR NQ nDCG@10				
		0-shot	TsR	TempR	MA	LoRA	0-shot	TsR	TempR	MA	LoRA	0-shot	TsR	TempR	MA	LoRA
BM25	-	34.58	-	-	-	-	76.61	-	-	-	-	32.90	-	-	-	-
contriever (Mean-768)	Original Our	21.50	53.15 53.64	57.93 63.01	19.75 20.90	56.91 60.39	85.14	95.46 95.54	95.23 96.21	81.62 84.24	96.37 97.08	25.37	14.60 15.22	19.94 21.42	17.63 17.68	18.80 19.46
gte (Mean-768)	Original Our	80.47	-* -*	-* -*	78.65 78.72	76.04 80.34	97.56	-* -*	-* -*	97.50 97.52	96.24 97.13	52.84	-* -*	-* -*	52.44 52.53	44.58 50.78
nomic (Mean-768)	Original Our	78.01	56.25 60.53	-* -*	75.10 75.14	81.59 82.55	95.60	92.22 93.87	-* -*	95.79 95.75	96.89 97.58	59.73	36.66 41.13	-* -*	58.88 58.94	62.50 62.76
bge (CLS-768)	Original Our	72.63	44.75 59.87	-* -*	69.80 70.14	78.00 77.42	97.87	89.47 94.27	-* -*	97.38 97.46	97.49 97.58	54.15	37.22 34.30	-* -*	53.01 53.14	50.97 52.47
gte1.5 (CLS-768)	Original Our	52.17	53.14 67.10	-* -*	48.65 50.29	72.90 74.00	94.71	92.94 95.54	-* -*	93.84 94.20	96.42 96.09	52.96	41.81 40.86	-* -*	52.70 52.75	52.03 52.78
bge-m3 (CLS-1024)	Original Our	59.62	43.84 55.31	-* -*	57.65 57.83	78.51 80.23	96.99	88.57 95.36	-* -*	96.55 96.56	96.27 96.47	60.57	18.56 23.35	-* -*	59.92 59.96	57.81 59.54

Table 2: Retrieval performance on TNP (full-dimensional embeddings). “-” indicates N/A; * denotes unstable training. Abbreviations: 0-shot (Zero-shot), TsR (Ts-Retriever), TempR (TempRetriever), MA (M-Adaptor), LoRA. “Original” / “Our” denote original/augmented training sets. Green highlights the superior training set per method. Bold and underlined values indicate best and second-best results per TEM.

4.2 Validating The Augmented TNP Dataset

Table 2 demonstrates that our augmented TNP dataset yields consistent gains across methods and TEMs by providing cleaner temporal supervision. The results highlight limitations in prior approaches: TempRetriever peaks on contriever but lacks cross-TEM generalization, whereas Ts-Retriever shows stable improvements in nDCG@10 and Recall@100. Among parameter-efficient methods, M-Adaptor yields negligible gains, while LoRA best leverages the augmented data, consistently outperforming baselines with only marginal NQ degradation. Additional augmentation pipeline’s ablation studies are provided in Appendix D.5.

To understand how different temporal relations (e.g., Before/After, Contains/During) influence

TMRL’s performance, we perform a comprehensive ablation study that systematically removes specific Allen relations from the training data in Appendix B.3 This serves as a quantitative proxy diagnostic since both TNP and TimeQA lack explicit Allen relation annotations for test queries. Unless noted otherwise, all subsequent experiments use the augmented training datasets.

4.3 Main Results

Figures 4 and 8 report retrieval performance on TNP and TimeQA. Across both benchmarks, M-Adaptor’s performance is bounded by the TEM’s zero-shot capability, suggesting limitations in semantic-only Matryoshka adaptation. nomic’s zero-shot degradation further indicates that temporal retrieval may be more sensitive to dimension-

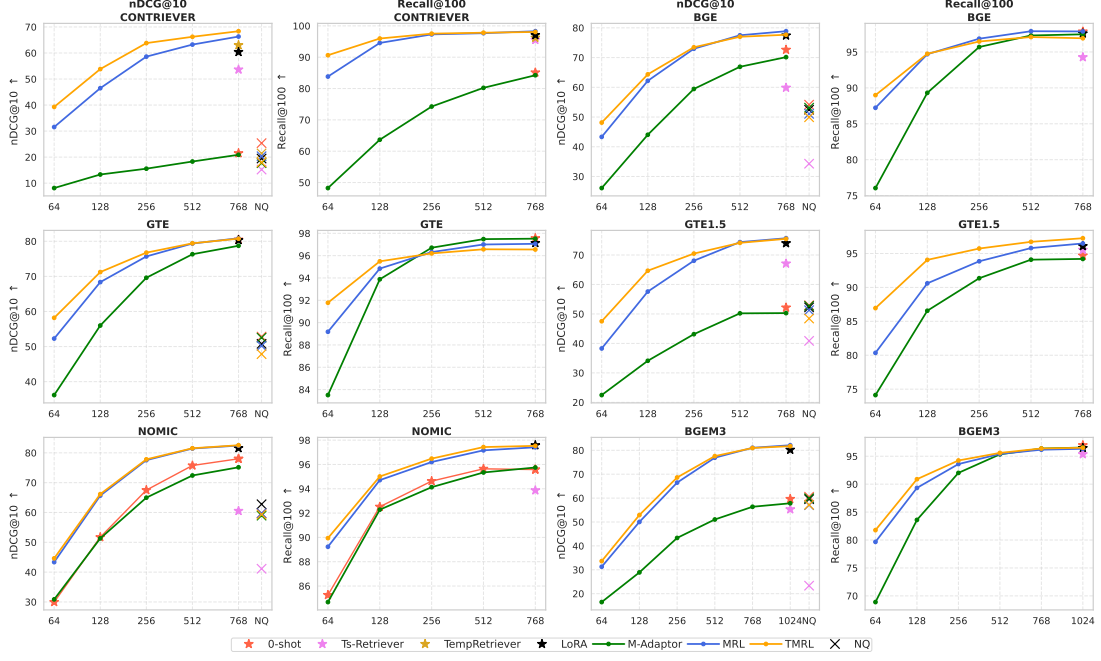


Figure 4: Retrieval performance on TNP. TMRL retrievers improves temporal retrieval, particularly nDCG@10 across embedding dimensionalities, while preserving general semantic performance on NQ.

ality reduction than semantic retrieval.

TNP. TMRL shows consistent improvements over M-Adaptor across dimensions and TEMs. Gains are most pronounced at lower dimensions, with nDCG@10 improvements suggesting that the dedicated temporal subspace helps preserve temporal signals under truncation. Unlike Ts-Retriever, TMRL maintains competitive semantic retrieval on NQ. We observe a slight Recall@100 drop at higher dimensions for gte and bge; however, since broader recall does not necessarily improve downstream RAG (Jin et al., 2025a), we prioritize nDCG@10, which better reflects top-ranked temporal relevance for RAG pipelines.

TimeQA. Due to space constraints, we place TimeQA results in Appendix B.1. On TimeQA, TMRL’s improvements emerge at higher dimensions (i.e., ≥ 256), suggesting this benchmark relies more heavily on semantic capacity. Beyond 256 dimensions, TMRL matches the MRL baseline’s Recall@100 while showing nDCG@10 gains, with NQ semantic performance remaining comparable. Our augmentation pipeline yields additional nDCG@10 improvements across most TEMs, with modest Recall@100 trade-offs at smaller scales. These results indicate that TMRL’s temporal-aware Matryoshka embeddings can improve temporal retrieval while supporting flexible

accuracy–efficiency trade-offs.

4.4 Temporal RAG Results

Figures 5 and 7 report temporal RAG F1 on TNP and TimeQA using Qwen3-8B (Yang et al., 2025) with contriever variants. RAG trends closely follow the corresponding retrieval results (Figures 4 and 8), suggesting retriever quality is a key factor in temporal RAG performance. Across both datasets, TMRL contriever achieves a strong accuracy–efficiency trade-off among Matryoshka baselines. With a dedicated temporal subspace, TMRL shows improvements over other adaptation methods across dimensions, with gains most pronounced under aggressive truncation. On TNP, for instance, 64-dimensional embeddings yield over $3\times$ the F1 score of M-Adaptor; 256-dimensional embeddings match TempRetriever’s performance while using only 33% of the index size.

4.5 Discussions

Efficiency Analysis. Table 3 compares TMRL against prior methods using the contriever backbone. TMRL matches Ts-Retriever in storage (0.8 GiB) and inference cost (single model) but offers three key advantages: (i) parameter-efficient LoRA adaptation, (ii) stronger temporal retrieval with competitive semantic performance on NQ, and (iii) native Matryoshka support for flexible truncation. Conversely, TempRetriever uses dual bi-encoders

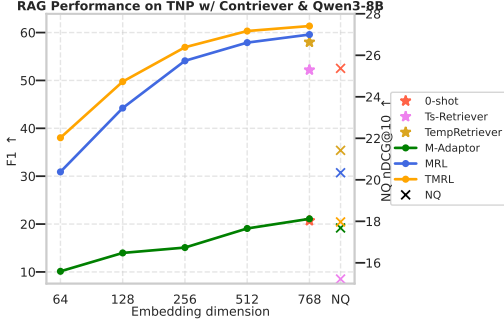


Figure 5: TMRL contriever enables the Temporal RAG system to achieve competitive TQA performance while enabling flexible accuracy-efficiency trade-off.

Method	Train ↓ (Model)	Inference ↓ (Model)	Storage ↓ (GiB)	Corpus Index Time ↓ (s)	TNP ↑ (nDCG@10)
TsR	1	1	0.8	< 1	53.6
TempR	2 [†]	2 [†]	1.6	82	63.0
TSM	7	1	5.7	< 1	64.6
TMRL	LoRA	1	0.8	< 1	68.4

Table 3: Efficiency comparison of TMRL and prior temporal retrieval methods using contriever on TNP. [†] denotes further requiring a temporal extractor. TsR and TempR denote Ts-Retriever and TempRetriever, respectively.

and an external extractor (SUTime (Chang and Manning, 2012)), doubling storage to 1.6 GiB, increasing latency, and requiring 82 seconds to index just 1000 documents. TSM (Han et al., 2025) trains seven time-specific models (5.7 GiB storage), yielding highly inefficient training despite inference-time parameter averaging.

Unlike these fixed-size baselines, TMRL enables dynamic index sizing. Table 4 shows that a 256-dimensional TMRL index uses only 33% of the full storage while retaining 92% of nDCG@10 and 84% of retrieval time. Since latency metrics cover search operations only, actual gains from reduced memory I/O and cache effects are likely higher.

Funnel Retrieval. Funnel retrieval leverages low-dimensional embeddings for initial candidate retrieval (e.g., $k=100, d=64$) followed by higher-dimensional re-ranking (e.g., $k=10, d=768$). Table 5 shows that TMRL contriever’s funnel pipeline on TNP maintains comparable nDCG@10 to full-dimensional retrieval and retains $\approx 95\%$ performance on TimeQA while reducing latency. In a scaled setting (e.g., TimeQA corpus duplicated $10\times$ to 1.1M passages), this approach cuts retrieval time to $\approx 36\%$ of the full-dimensional baseline.

Criterion	64	128	256	512	768
nDCG@10	26.9	39.8	45.0	48.1	49.0
Ratio	55%	81%	92%	98%	100%
Corpus Index (MiB)	28	54	107	214	321
Ratio	9%	17%	33%	67%	100%
Latency (ms)	183	187	192	210	230
Ratio	79%	81%	84%	92%	100%

Table 4: Performance, corpus index size, and retrieval latency (millisecond) of TMRL contriever’s Matryoshka embeddings on TimeQA. Retrieval latency reflects search operations only (index loading excluded).

Method	TNP		TimeQA	
	nDCG@10 ↑	Latency (s) ↓	nDCG@10 ↑	Latency (s) ↓
MRL	66.31	<u>0.101</u>	42.02	<u>0.226</u>
TMRL	68.36	<u>0.101</u>	<u>49.00</u>	<u>0.226</u>
Funnel MRL	66.19	0.083	39.59	0.185
Funnel TMRL	<u>68.35</u>	0.083	46.42	0.185
Latency (s) on $10\times$ larger TimeQA corpus ↓				
Standard	0.688			
Funnel	0.250			

Table 5: Funnel retrieval (top- $k=100, d=64$) $\rightarrow$ (top- $k=10, d=768$) vs. standard full-dimensional retrieval.

Effect of Temporal Subspace. Figure 6 examines the temporal subspace dimension t and loss weighting coefficient α , Eq. (13), with auxiliary losses disabled; corresponding semantic performance on NQ is reported in Table 6. For α , values above 0.5 notably degrade bge performance, while contriever shows improved temporal retrieval up to $\alpha = 0.5$ alongside a semantic trade-off. Regarding t , bge requires $t \geq 64$ to improve temporal retrieval without notable semantic loss. contriever follows a similar pattern, where increasing t yields consistent temporal gains with limited semantic impact. Based on these observations, we set $\alpha \in [0.1, 0.25]$ and $t \in [64, 128]$. Results on TimeQA are provided in Appendix B.2 and per-model configurations are detailed in Appendix D.

Auxiliary Losses Ablation Study. Tables 7 and 8 analyze the effect of the local (β) and structural (γ) self-distillation auxiliary losses in Eq. (13) on the TNP dataset. Setting both β and γ to a small value (e.g., 0.1) improves alignment at both local and global levels compared to using no auxiliary loss. This setting is robust across contriever and bge: the former trades semantic for temporal retrieval, while the latter shows only minor changes. However, larger values introduce competition with the

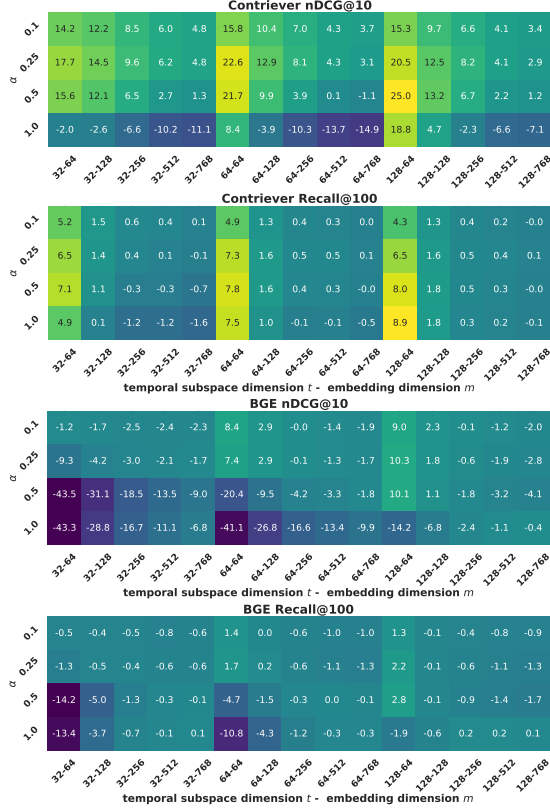


Figure 6: Ablation of the temporal subspace dimension t and the temporal subspace coefficient α . The results show the increase/decrease relative to the LoRA-based MRL baseline on TNP using contriever and bge.

	contriever				BGE			
α	0.1	0.25	0.5	1.0	0.1	0.25	0.5	1.0
$t=32$	19.67	18.72	17.04	15.00	48.81	47.21	48.71	45.82
$t=64$	19.64	18.56	16.45	13.42	53.76	49.18	<u>50.62</u>	49.33
$t=128$	<u>19.65</u>	18.92	17.62	14.97	50.29	49.05	47.36	50.30

Table 6: nDCG@10 on BEIR NQ.

contrastive objective. We set $\beta = \gamma = 0.1$ in all experiments. Results on TimeQA are provided in Appendix B.2 with similar observations.

5 Related Work

Temporal Information Retrieval (TIR) & Temporal Question Answering (TQA). TIR retrieves documents satisfying both semantic relevance and temporal constraints, while TQA extends this to time-aware reasoning (Pirayani et al., 2026). Despite advances in dense retrieval (Zhang et al., 2025c; Lee et al., 2025) and TQA (Fatemi et al., 2025; Bazaga et al., 2025), integrating temporal signals into dense retrievers remains underexplored. Recent efforts include supervised temporal contrastive learning (Wu et al., 2024), seman-

Dist. CKA		nDCG@10					Recall@100					NQ
β	γ	64	128	256	512	768	64	128	256	512	768	
0	0	36.5	51.3	62.7	66.0	68.8	87.9	95.8	97.7	97.9	98.3	19.6
0.1	0	39.2	53.8	63.6	66.3	68.6	90.5	95.9	97.7	97.9	98.1	18.2
0	0.1	37.3	52.3	63.5	66.4	68.8	88.6	96.0	97.8	98.0	98.4	19.4
0.1	0.1	39.3	53.8	63.8	66.2	68.4	90.6	96.0	97.5	97.8	98.1	18.0
0.25	0.25	37.4	48.7	55.9	58.4	60.4	91.0	95.0	96.9	97.2	97.4	14.2
0.5	0.5	11.4	14.0	17.7	21.0	22.8	53.7	70.6	82.7	87.4	89.6	18.1

Table 7: contriever auxiliary loss ablation on TNP.

Dist. CKA		nDCG@10					Recall@100					NQ
β	γ	64	128	256	512	768	64	128	256	512	768	
0	0	47.2	63.6	72.9	76.6	77.3	88.4	94.6	96.4	97.1	97.0	50.3
0.1	0	47.6	64.0	73.1	76.7	77.4	88.6	94.6	96.4	97.1	97.0	50.4
0	0.1	47.7	64.0	73.3	76.8	77.6	88.8	94.6	96.4	97.1	96.9	49.9
0.1	0.1	48.1	64.3	73.5	77.0	77.7	89.0	94.7	96.4	97.1	96.9	49.9
0.25	0.25	31.7	47.0	54.6	58.4	59.4	65.9	80.8	87.2	89.7	90.6	49.6
0.5	0.5	29.5	43.9	51.5	55.5	56.6	63.6	78.6	84.8	88.3	89.1	49.3

Table 8: bge auxiliary loss ablation on TNP.

tic-temporal fusion (Abdallah et al., 2026), and multi-retriever ensembles (Han et al., 2025), alongside emerging temporal RAG frameworks (Qian et al., 2024; Zhang et al., 2025b). In contrast, TMRL is a single-model, plug-and-play approach that equips TEMs with temporal-aware Matryoshka embeddings, improving both TIR and Temporal RAG efficiency.

Matryoshka Representation Learning (MRL). Modern TEMs increasingly adopt MRL (Kusupati et al., 2022) to produce nested embeddings for flexible compute-accuracy trade-offs (Lee et al., 2024; Nussbaum et al., 2025). While M-Adaptor (Yoon et al., 2024b) and subsequent multi-stage variants (Zhang et al., 2025a) adapt MRL for general retrieval efficiency. TMRL shares a similar motivation as M-Adaptor: both use parameter-efficient finetuning to equip TEMs with Matryoshka embeddings. However, TMRL emphasizes TIR, whereas M-Adaptor preserves general semantic retrieval.

6 Conclusion

We propose Temporal-aware Matryoshka Representation Learning (TMRL), an efficient Matryoshka adaptation method that equips TEMs with temporal-aware embeddings, improving temporal retrieval while preserving semantic performance. We also introduce a temporal contrastive data augmentation pipeline that provides cleaner temporal supervision. Experiments show that TMRL enhances temporal encoding in retrievers, improves downstream temporal RAG performance, and enables flexible accuracy-efficiency trade-offs under varying computational constraints.

Limitations

Despite evaluating TMRL across six widely used encoder-only text embedding models, we do not evaluate our method on LLM-based embedding models due to resource constraints. Evaluation on large-scale news, legal, or policy corpora, which often lack explicit temporal annotations, remains a promising direction for future work. However, it is worth noting that many real-world corpora are either access-restricted (e.g., copyrighted news) or lack time-sensitive retrieval annotations, making controlled evaluation of temporal retrieval difficult. Finally, to ensure clean training signals, our augmented temporal contrastive learning dataset follows a simple principle: splitting long passages into shorter passages and discarding sentences containing multiple temporal expressions. Extending TMRL’s training augmentation strategy to explicitly learn from passages containing multiple temporal expressions, which is common in real-world documents, remains an open direction for future research.

Acknowledgments

This material is based on research sponsored by Defense Advanced Research Projects Agency (DARPA) under agreement number HR0011-22-2-0047. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government. This work is supported by Monash eResearch capabilities, including M3. This work is partially supported by the Australian Research Council through Discovery Early Career Researcher Award (DE250100032). The authors are grateful to the anonymous reviewers for their helpful comments.

References

- Abdelrahman Abdallah, Bhawna Piryani, Jonas Wallat, Avishek Anand, and Adam Jatowt. 2026. [Tempretriever: Fusion-based temporal dense passage retrieval for time-sensitive questions](#). In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining, WSDM’26*, page 5–15, New York, NY, USA. Association for Computing Machinery.
- James F Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.
- Adrián Bazaga, Rexhina Blloshmi, Bill Byrne, and Adrià de Gispert. 2025. [Learning to reason over time: Timeline self-reflection for improved temporal reasoning in language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28014–28033, Vienna, Austria. Association for Computational Linguistics.
- Angel X. Chang and Christopher Manning. 2012. [SU-Time: A library for recognizing and normalizing time expressions](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 3735–3740, Istanbul, Turkey. European Language Resources Association (ELRA).
- Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. [M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.
- Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A Dataset for Answering Time-Sensitive Questions. In *NeurIPS*.
- Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. 2025. Test of Time: A Benchmark for Evaluating LLMs on Temporal Reasoning. In *ICLR*.
- Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. [Tevatron: An efficient and flexible toolkit for neural retrieval](#). In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’23*, page 3120–3124, New York, NY, USA. Association for Computing Machinery.
- SeungYoon Han, Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, Huije Lee, and Jong C Park. 2025. Temporal information retrieval via time-specifier model merging. In *Proceedings of the 3rd Workshop on Towards Knowledgeable Foundation Models (KnowFM)*, pages 1–13.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Duygu Sezen Islakoglu and Jan-Christoph Kalo. 2025. ChronoSense: Exploring temporal understanding in large language models with time intervals of events. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

- 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*.
- Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Complex temporal question answering on knowledge graphs. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 792–802.
- Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. 2025a. [Long-context LLMs meet RAG: Overcoming challenges for long inputs in RAG](#). In *The Thirteenth International Conference on Learning Representations*.
- Jiajie Jin, Yutao Zhu, Guanting Dong, Yuyao Zhang, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, Zhicheng Dou, and Ji-Rong Wen. 2025b. FlashRAG: A Modular Toolkit for Efficient Retrieval-Augmented Generation Research. In *WWW*.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR.
- Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2022. Matryoshka Representation Learning. In *NeurIPS*.
- Tom Kwiakowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoneybi, Bryan Catanzaro, and Wei Ping. 2025. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. In *ICLR*.
- Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, and 1 others. 2024. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. 2025. [From matching to generation: A survey on generative information retrieval](#). *ACM Trans. Inf. Syst.*, 43(3).
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. *arXiv preprint arXiv:2308.03281*.
- Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Xing Han Lù. 2024. [Bm25s: Orders of magnitude faster lexical search via eager sparse scoring](#). *Preprint*, arXiv:2407.03618.
- Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.
- Bhavnick Minhas and Shreyash Nigam. 2025. Chonkie: The lightweight ingestion library for fast, efficient and robust rag pipelines. <https://github.com/chonkie-inc/chonkie>.
- Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2025. Nomic Embed: Training a Reproducible Long Context Text Embedder. *TMLR*.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- Bhawna Piryani, Abdelrahman Abdallah, Jamshid Mozafari, Avishek Anand, and Adam Jatowt. 2026. [It’s high time: A survey of temporal question answering](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

- Long Papers*), pages 28852–28881, San Diego, California, United States. Association for Computational Linguistics.
- Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. 2019. On variational bounds of mutual information. In *International conference on machine learning*, pages 5171–5180. PMLR.
- Xinying Qian, Ying Zhang, Yu Zhao, Baohang Zhou, Xuhui Sui, Li Zhang, and Kehui Song. 2024. [TimeR⁴: Time-aware retrieval-augmented large language models for temporal knowledge graph question answering](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6942–6952, Miami, Florida, USA. Association for Computational Linguistics.
- Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *FNT in Information Retrieval*, 3.
- Xintong Song, Bin Liang, Yang Sun, Chenhua Zhang, Bingbing Wang, and Ruifeng Xu. 2025. [Bridging time gaps: Temporal logic relations for enhancing temporal reasoning in large language models](#). In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '25*, page 3040–3044, New York, NY, USA. Association for Computing Machinery.
- Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards Benchmarking and Improving the Temporal Reasoning Capability of Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 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)*.
- Sebastian Thrun. 1995. Is Learning The n-th Thing Any Easier Than Learning The First? In *Advances in Neural Information Processing Systems*, volume 8.
- Jonas Wallat, Abdelrahman Abdallah, Adam Jatowt, and Avishek Anand. 2025. [A study into investigating temporal robustness of LLMs](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15685–15705, Vienna, Austria. Association for Computational Linguistics.
- Yuqing Wang and Yun Zhao. 2024. [TRAM: Benchmarking temporal reasoning for large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6389–6415, Bangkok, Thailand. Association for Computational Linguistics.
- 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*.
- Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muenighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed Resources For General Chinese Embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Jinsung Yoon, Yanfei Chen, Sercan Arik, and Tomas Pfister. 2024a. [Search-adaptor: Embedding customization for information retrieval](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12230–12247, Bangkok, Thailand. Association for Computational Linguistics.
- Jinsung Yoon, Rajarishi Sinha, Sercan O Arik, and Tomas Pfister. 2024b. [Matryoshka-adaptor: Unsupervised and supervised tuning for smaller embedding dimensions](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10318–10336, Miami, Florida, USA. Association for Computational Linguistics.
- Biao Zhang, Lixin Chen, Tong Liu, and Bo Zheng. 2025a. [SMEC:rethinking matryoshka representation learning for retrieval embedding compression](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26220–26233, Suzhou, China. Association for Computational Linguistics.
- Siyue Zhang, Yuxiang Xue, Yiming Zhang, Xiaobao Wu, Anh Tuan Luu, and Chen Zhao. 2025b. [MRAG: A modular retrieval framework for time-sensitive question answering](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 3080–3118, Suzhou, China. Association for Computational Linguistics.
- Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. [mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412, Miami, Florida, US. Association for Computational Linguistics.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang,

Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025c. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. *Dense text retrieval based on pretrained language models: A survey*. *ACM Trans. Inf. Syst.*, 42(4).

A Examples for Figure 1

We use a question-answer pair from Temporal Nobel Prize (TNP) to demonstrate our argument in the introduction (Section 1).

Query: Who won the Nobel Prize in Chemistry between 1910 and 1911?

Answer: "Otto Wallach", "Marie Curie"

Zero-shot contriever retrieves temporally misaligned documents (e.g., Rutherford 1908, Richards 1914), leading all generators to fail:

- **Qwen3-8B:** None (F1=0)
- **Qwen3-32B:** No one won the Nobel Prize ... (F1=0)
- **DeepSeek-R1-Distill-Qwen-32B:** None of the provided documents list... (F1=0)
- **Llama-3.3-70B-Instruct:** E. Rutherford. (F1=0)

With TMRL contriever, we correctly retrieved one relevant document from the top-5 retrieved documents:

1. **Otto Wallach (1910 Chemistry) (Correct)**
2. Jacobus H. van't Hoff (1901 Chemistry)
3. E. Rutherford (1908 Chemistry)
4. W. Ostwald (1909 Chemistry)
5. A. Kossel (1910 Medicine)

As a result, stronger LLMs can leverage their strength to answer more correctly:

- **Qwen3-8B:** Otto Wallach and Jacobus H. van't Hoff (F1=0.33)
- **Qwen3-32B:** Otto Wallach, A. Kossel (F1=0.5)
- **DeepSeek-R1-Distill-Qwen-32B:** Otto Wallach (F1=0.67)
- **Llama-3.3-70B-Instruct:** Otto Wallach (F1=0.67)

B Additional Results

B.1 TimeQA Main Results

Due to limited space, we attach the TimeQA results in this Appendix section. The Matryoshka embedding results are displayed in Figure 8. The RAG result is presented in Figure 7.

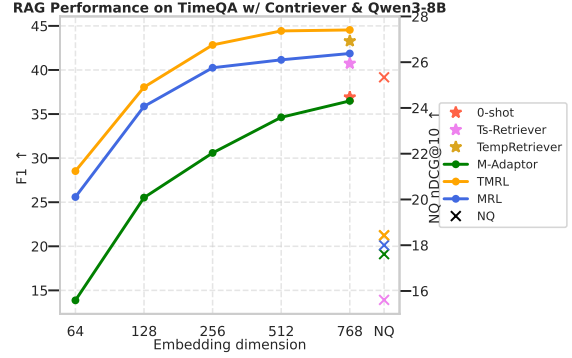


Figure 7: TimeQA RAG performance using Qwen3-8B with different contriever variants trained on our augmented data. Note that the respective semantic retrieval performance of TempRetriever and TMRL on NQ are 18.42 and 18.44, causing overlapping markers.

B.2 Additional Hyperparameters Ablation Studies on TimeQA

α	nDCG@10					Recall@100					NQ
	64	128	256	512	768	64	128	256	512	768	
0.1	64.8	80.8	87.8	90.2	90.6	26.9	39.8	45.0	48.1	49.0	18.4
0.25	60.0	78.4	86.5	89.5	90.1	23.0	35.3	40.9	44.5	45.4	18.4
0.5	57.6	76.1	84.8	88.0	88.9	21.3	33.1	39.3	42.9	44.2	18.2
1	52.3	71.3	79.8	84.1	85.0	18.1	28.4	33.6	37.8	38.6	17.7

Table 9: Ablation of the temporal subspace dimension t and the temporal subspace coefficient α on TimeQA using contriever. Auxiliary losses are disabled ($\beta = \gamma = 0$).

α	nDCG@10					Recall@100					NQ
	64	128	256	512	768	64	128	256	512	768	
0.1	70.5	84.2	90.5	92.4	93.5	34.6	49.9	57.8	60.6	61.1	52.3
0.25	71.2	84.4	90.1	92.4	93.3	35.0	49.7	58.3	61.3	61.9	51.7
0.5	62.3	78.1	87.4	90.5	91.6	25.9	40.7	51.3	55.4	55.7	51.8

Table 10: Ablation of the temporal subspace dimension t and the temporal subspace coefficient α on TimeQA using bge. Auxiliary losses are disabled ($\beta = \gamma = 0$).

We extend our hyperparameters ablation studies to the TimeQA dataset, which represents a fundamentally different domain distribution: while TNP focuses on isolated, fact-based temporal retrieval with explicit dates, TimeQA involves complex, evolving entity histories from Wikipedia where temporal signals are deeply entangled with dense semantic context.

Ablation studies on α . According to the results from Table 9 and 10, we conclude that the best for

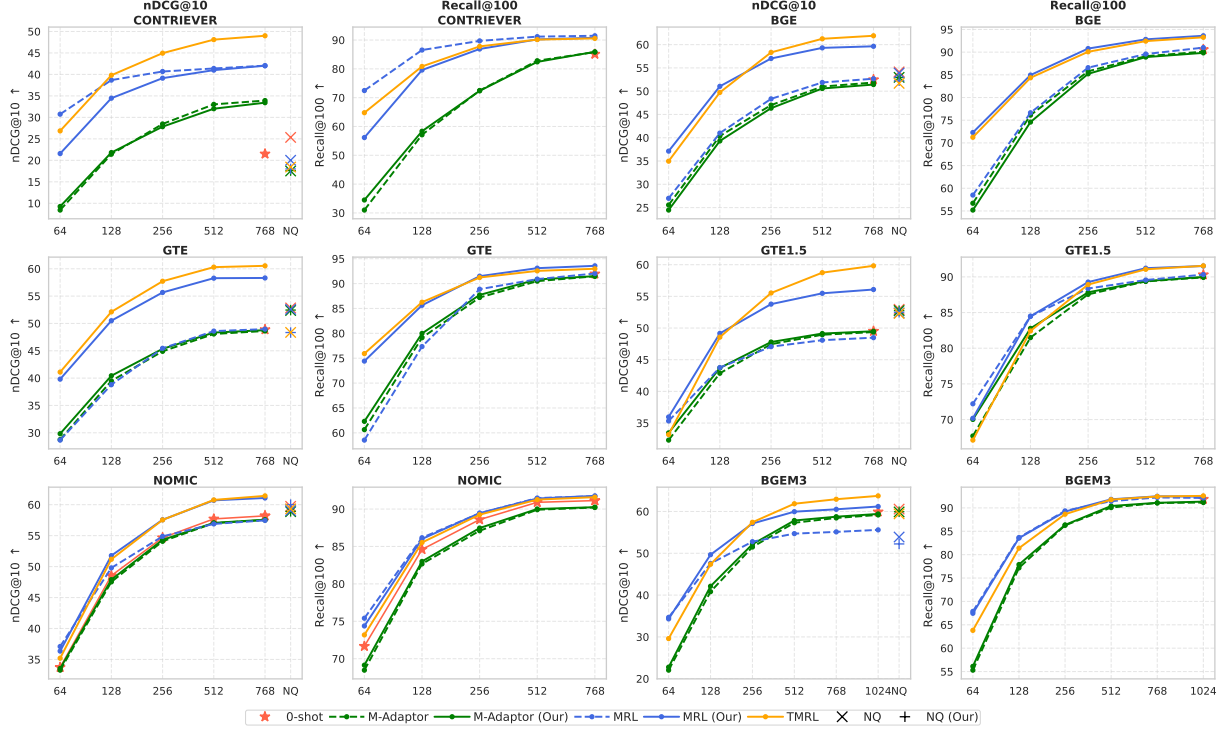


Figure 8: Matryoshka embedding results on TimeQA. (Our) denotes training using our augmented data.

Dist. CKA		nDCG@10					Recall@100					NQ
β	γ	64	128	256	512	768	64	128	256	512	768	
0	0	55.3	78.9	87.0	90.2	90.8	20.8	34.0	39.1	41.4	42.2	18.0
0.1	0.1	64.5	80.8	87.7	90.1	90.5	26.7	39.6	44.7	47.9	48.8	18.4
0.25	0.25	49.7	67.1	79.1	85.9	87.0	15.9	25.5	30.0	34.1	35.1	18.1
0.5	0.5	33.7	56.0	70.1	81.9	84.7	9.3	19.8	24.6	30.3	31.6	18.0

Table 11: contriever auxiliary loss ablation on TimeQA.

Dist. CKA		nDCG@10					Recall@100					NQ
β	γ	64	128	256	512	768	64	128	256	512	768	
0	0	62.4	79.0	88.2	91.1	91.9	27.2	42.7	52.2	55.7	56.0	52.3
0.1	0.1	70.5	84.2	90.5	92.4	93.5	34.6	49.9	57.8	60.6	61.1	52.3
0.25	0.25	61.8	78.1	87.7	90.6	91.5	26.2	41.3	51.0	54.9	55.1	52.2
0.5	0.5	65.5	80.7	88.6	91.3	92.4	29.1	43.9	53.4	57.3	57.6	52.2

Table 12: bge auxiliary loss ablation on TimeQA.

contriever and bge on TimeQA are 0.1 and 0.25, respectively.

Ablation studies on β, γ . According to the results from Table 11 and 12, we conclude that $\beta = \gamma = 0.1$ provides the best performance for both contriever and bge on TimeQA.

The above ablation studies demonstrate that default values $\alpha \in [0.1, 0.25]$ and $\beta = \gamma = 0.1$ in the main paper are robust enough for both contriever and bge on a different domain distribution - TimeQA, where they consistently perform

better compared to other settings.

B.3 Ablation Study on Allen Relations

To analyze how TMRL captures temporal reasoning patterns, we systematically remove each category of Allen relations from the augmented TimeQA dataset (both positives and negatives queries) and report the difference in retrieval performance on TimeQA, which contains diverse temporal query types. This controlled removal allows us to quantify how different temporal Allen relations contribute to the learned representations. We group symmetrical Allen relations together, according to [Islakoglu and Kalo \(2025\)](#).

From Table 13, we observe that Before/After relations dominate the training data and are critical, removing them leads to the largest performance drop. Relations involving exact timestamps (Equals) and temporal intervals (Contains/During, Overlaps/Overlapped-by) also substantially improve temporal retrieval, though with a mild trade-off against purely semantic retrieval. In contrast, Finishes/Finished-by, Meets/Met-by, and Starts/Started-by are rarely generated by our pipeline, and their removal does not harm (and slightly improves) performance on TimeQA.

Additionally, our augmentation includes ‘‘Temporal Answer’’ queries that ask for event times-

Data variants	64	128	256	512	768	BEIR NQ	# training samples
All (TMRL contriever)	26.9	39.8	45.0	48.1	49.0	18.4	8149
No Before/After	-15.2	-16.6	-15.5	-14.1	-13.6	-0.4	1849
No Equals	-0.5	-1.4	-1.3	-2.0	-2.1	0.6	8146
No Contains/During	-6.7	-6.1	-5.5	-4.8	-4.5	0.8	8124
No Overlaps/Overlapped-by	-7.4	-7.9	-7.7	-8.3	-8.1	-0.2	8149
No Finishes/Finished-by	0.5	0.6	0.5	0.4	0.5	0.1	8149
No Meets/Met-by	0.3	0.3	0.3	0.3	0.3	0.0	8149
No Starts/Started-by	0.3	0.5	0.4	0.4	0.6	0.2	8149
No "Temporal Answer" Queries	-4.9	-5.1	-5.1	-5.7	-5.7	-0.5	8114

Table 13: Ablation study on TimeQA and BEIR NQ using TMRL contriever. Values report difference in nDCG@10 relative to the model train with the fully augmented data (All) (positive/negative values indicate performance gain/drop). The final column shows the number of training samples containing at least one positive/negative query after filtering.

tamps without explicit temporal constraints, which further strengthen temporal representations while preserving semantic retrieval.

B.4 Qwen3-Embedding-0.6B’s Preliminary Results on TNP

We extended our evaluation on the TNP dataset to Qwen3-Embedding-0.6B (Zhang et al., 2025c), a representative instruction-tuned LLM-based embedder, as preliminary results beyond the encoder-only TEMs. Table 14 shows that there is a divergence in representation learning dynamics when applied TMRL to instruction-tuned architectures: setting the auxiliary structural loss coefficients to zero ($\beta = \gamma = 0$) consistently outperforms the standard setting ($\beta = \gamma = 0.1$) used for encoder-only TEMs. We hypothesize that while encoder-only bi-encoders (e.g., contriever) require structural mimicry via CKA and self-distillation to align truncated dimensions (as shown in Tables 7 and 8), instruction-tuned LLM-based embedders possess a highly optimized, instruction-aligned representational geometry. Forcing these models to structurally mimic the full-dimensional space causes over-regularization.

The funnel retrieval approach remains applicable to Qwen3-Embedding-0.6B. As shown in Table 15, it maintains comparable nDCG@10 performance while reducing latency compared to standard full-dimensional retrieval.

B.5 Semantic Retrieval Results on BEIR Datasets

Following TsRetriever (Wu et al., 2024) and TSM (Han et al., 2025), we use Natural Questions (NQ) as a benchmark to evaluate semantic retrieval

preservation. NQ contains 3452 test queries over 2.68M passages, of which only 1.54% are time-sensitive (Han et al., 2025). Another widely used benchmark is MS MARCO (7k test queries, 8.8M documents). However, as reported in TSM, TEMs that perform well on NQ generally exhibit similar trends on MS MARCO. Nevertheless, we additionally report results on additional datasets from the BEIR benchmark (Thakur et al., 2021), including SciFact and MS MARCO in terms of nDCG@10, using contriever (Table 16) and bge (Table 17). The results show that TMRL preserves semantic retrieval performance across datasets from the BEIR benchmark, with behavior varying slightly depending on the underlying TEM.

B.6 Qualitative Failure Analysis

We conducted manual case studies on TimeQA to identify specific failure modes of TMRL compared to baselines. Our analysis reveals that TMRL faces greater challenges in handling complex cases where temporal constraints are deeply intertwined with semantic inference. While baselines also struggle with these cases, they achieve higher Recall@k (for larger k) because they tend to retrieve a broader range of semantically related context. In contrast, TMRL excels at strictly ranking the most temporally relevant passages in the top positions, directly driving its superior nDCG@10 performance. Here are two representative cases:

Case 1: TMRL fails, MRL succeeds.

- **Query:** “Which political party did Walter Veltroni belong to between Jan 1999 and Aug 2006?”
- **Gold Passage:** “In 1998, he resigned, subsequent to his election as National Secretary of

Method	64	128	256	512	768	1024	NQ
Recall@100							
Zero-shot	78.0	86.2	91.6	94.1	94.9	95.3	-
MRL	84.0	89.6	93.2	95.2	95.8	95.7	-
TMRL ($\beta = \gamma = 0.1$)	84.6	89.7	92.6	94.6	95.1	95.3	-
TMRL ($\beta = \gamma = 0$)	86.2	90.6	93.1	94.8	95.3	95.5	-
nDCG@10							
Zero-shot	17.8	27.0	36.8	44.7	49.9	52.4	53.5
MRL	39.3	52.1	61.8	66.4	67.3	67.3	39.5
TMRL ($\beta = \gamma = 0.1$)	42.0	53.5	61.4	65.6	66.2	66.1	41.8
TMRL ($\beta = \gamma = 0$)	44.3	54.8	62.8	66.9	67.5	67.3	39.6

Table 14: Qwen3-Embedding-0.6B results on TNP.

Method	nDCG@10	Latency (s)	Latency on 10× larger corpus (s)
TMRL	67.34	0.090	0.154
Funnel TMRL	67.35	0.086	0.105

Table 15: Funnel retrieval (top-k=100, d=64) → (top-k=10, d=1024) vs. standard full-dimensional retrieval using Qwen3-Embedding-0.6B.

the Democrats of the Left (DS)... For example, in 2001, after the late night show “Satyri-con” aired an interview that discussed indictments on links between the right-wing leader and the mafia, Marco Travaglio reported that Veltroni dispatched a messenger menacing the closure of the show.”

- **Analysis:** The gold passage lacks an explicit temporal expression matching “1999–2006”. Retrieving it requires inferring that his role elected in 1998 continued through 2006, based on the given semantic context and corresponding temporal expression “in 2001”.
- **MRL:** Ranks the gold passage 10th. which indicates that MRL tried to retrieve everything related to politics (i.e., the semantic context).
- **TMRL:** Fails to retrieve it in the Top-10.

Case 2: TMRL succeeds, MRL fails

- **Query:** “Which team did the player Aman Verma belong to between Jul 2009 and Sep 2010?”
- **Gold Passage:** “Section: Leicester City... Verma had his first call up to the main Leicester squad as an unused substitute in a 2–1 win at Blackpool on 6 February 2010. He joined Conference National club Histon on loan for the remainder of the 2009–10 season on 26 March 2010, In May 2010, Verma was awarded another contract

extension by Leicester.”

- **Analysis:** The corpus contains over ten highly semantically similar passages about Aman Verma’s career across different years.
- **MRL:** Fails to retrieve the gold passage in the Top-10. It gets distracted by semantically identical but temporally mismatched passages (e.g., Top-1 discusses events in 2011 while the Top-2 provides an abstract overview of the player).
- **TMRL:** Successfully ranks the gold passage at Top-5. It correctly prioritizes the passage containing the 2010 temporal signals.

C Proof of Matryoshka Representation Learning as Multi-Scale Mutual Information Maximization

Consider the setup in Section 3.2.2, where the input is a passage representation $\mathbf{z}_p$ and the target is a query representation $\mathbf{z}_q$. Matryoshka Representation Learning (MRL) aggregates the InfoNCE loss across all target dimensions:

$$\mathcal{L}_{\text{MRL}} = \sum_{m \in \mathcal{M}} \mathcal{L}_{\text{InfoNCE}}^{(m)}. \quad (14)$$

Given a positive pair $(\mathbf{z}_p, \mathbf{z}_{q+}) \sim p_{\text{pos}}(\mathbf{z}_p, \mathbf{z}_q)$ and $K - 1$ negative samples $\{\mathbf{z}_{q_i^-}\}_{i=1}^{K-1} \sim p(\mathbf{z}_q)$ drawn

Dataset	contriever (zero-shot)	TMRL contriever	Semantic Preservation Ratio (%) $\uparrow$
NQ	25.4	18.0	71%
SciFact	57.1	53.2	93%
MS MARCO	20.6	15.9	77%

Table 16: Semantic retrieval preservation in terms of nDCG@10 between zero-shot contriever and TMRL contriever on NQ, SciFact, and MS MARCO from BEIR.

Dataset	bge (zero-shot)	TMRL bge	Semantic Preservation Ratio (%) $\uparrow$
NQ	54.2	50.0	92%
SciFact	74.3	72.0	97%
MS MARCO	35.0	33.8	97%

Table 17: Semantic retrieval preservation in terms of nDCG@10 between zero-shot bge and TMRL bge on NQ, SciFact, and MS MARCO from the BEIR benchmarks.

independently from the marginal distribution of the queries, the InfoNCE loss at scale m is:

$$\mathcal{L}_{\text{InfoNCE}}^{(m)} \triangleq -\mathbb{E} \left[\log \frac{e^{\text{sim}^{(m)}(\mathbf{z}_p, \mathbf{z}_{q+})/\tau}}{e^{\text{sim}^{(m)}(\mathbf{z}_p, \mathbf{z}_{q+})/\tau} + \sum_{i=1}^{K-1} e^{\text{sim}^{(m)}(\mathbf{z}_p, \mathbf{z}_{q_i^-})/\tau}} \right]. \quad (15)$$

Following the variational mutual information bound established in contrastive representation learning (Oord et al., 2018; Poole et al., 2019), minimizing $\mathcal{L}_{\text{InfoNCE}}^{(m)}$ maximizes a lower bound on the mutual information preserved in the truncated representation:

$$I((\mathbf{z}_p)_{\leq m}; (\mathbf{z}_q)_{\leq m}) \geq \log K - \mathcal{L}_{\text{InfoNCE}}^{(m)}. \quad (16)$$

Summing this inequality over all scales $m \in \mathcal{M}$ yields:

$$\sum_{m \in \mathcal{M}} I((\mathbf{z}_p)_{\leq m}; (\mathbf{z}_q)_{\leq m}) \geq |\mathcal{M}| \log K - \mathcal{L}_{\text{MRL}}. \quad (17)$$

Defining the tractable variational lower bound on cumulative mutual information as:

$$\mathcal{I}_{\text{MRL}} \triangleq |\mathcal{M}| \log K - \mathcal{L}_{\text{MRL}}, \quad (18)$$

we have $\sum_{m \in \mathcal{M}} I((\mathbf{z}_p)_{\leq m}; (\mathbf{z}_q)_{\leq m}) \geq \mathcal{I}_{\text{MRL}}$. Since $|\mathcal{M}| \log K$ is constant with respect to the encoder parameters θ , minimizing the MRL objective is equivalent to maximizing this lower bound:

$$\min_{\theta} \mathcal{L}_{\text{MRL}} \iff \max_{\theta} \mathcal{I}_{\text{MRL}} \leq \max_{\theta} \sum_{m \in \mathcal{M}} I((\mathbf{z}_p)_{\leq m}; (\mathbf{z}_q)_{\leq m}). \quad (19)$$

D Implementation Details

All experiments are conducted on a single NVIDIA A100 80GB GPU using the AdamW optimizer (Loshchilov and Hutter, 2019). All methods are fine-tuned for 5 epochs with in-batch negatives and additional hard negatives, except for nomic, which is only fine-tuned for 1 epoch, as noted in the original paper (Nussbaum et al., 2025). We summarize the general configurations for the six text embedding models (TEMs) used for evaluation in Table 18. We develop our methods and run experiments using the Tevatron framework (Gao et al., 2023). We use Pyserini (Lin et al., 2021) to perform retrieval evaluation. We use the vLLM framework (Kwon et al., 2023) for all LLMs related generation.

D.1 BM25 and full fine-tuning baselines

- **BM25:** We use the default implementation of BM25s (Lù, 2024).
- **Ts-Retriever:** We follow the original implementation.
- **TempRetriever:** TempRetriever employs two separate TEMs for semantic and temporal encoding and requires temporal expression extraction at inference time. We use SU-Time (Chang and Manning, 2012) to extract all temporal expressions, which are concatenated using comma separation. As a result, TempRetriever has a larger model footprint and significantly higher inference latency than other methods. As the code is not publicly available at the time of writing, we implement

TEM	HuggingFace Signature	Query’s prefix	Passage’s prefix	Temperature τ	Pooling
contriever	face-book/contriever			0.05	Mean
gte	thenlper/gte-base			0.02	Mean
nomic	nomic-ai/nomic-embed-text-v1.5	search_query:	search_document:	0.05	Mean
bge	BAAI/bge-base-en-v1.5	Represent this sentence for searching relevant passages:	Represent this sentence for searching relevant passages:	0.02	CLS
gte1.5	Alibaba-NLP/gte-base-en-v1.5			0.02	CLS
bgem3	BAAI/bge-m3			0.02	CLS
qwen3	Qwen/Qwen3-Embedding-0.6B	Instruct: Given a web search query, retrieve relevant passages that answer the query\nQuery:		0.02	EOS

Table 18: Training hyperparameters of TEMs.

TEM	TNP				TimeQA			
	t	α	β	γ	t	α	β	γ
contriever	64	0.1	0.1	0.1	128	0.1	0.1	0.1
gte	64	0.1	0.1	0.1	64	0.1	0.1	0.1
nomic	128	0.1	0.1	0.1	128	0.1	0.1	0.1
bge	128	0.1	0.1	0.1	128	0.25	0.1	0.1
gte1.5	64	0.25	0.1	0.1	128	0.1	0.1	0.1
bgem3	128	0.1	0.1	0.1	128	0.1	0.1	0.1
qwen3	128	0.1	0	0	-	-	-	-

Table 19: TMRL’s training hyperparameters for each TEM.

the Element-Wise Interaction (EWI) variant instead of Feature Stacking (FS) to ensure that all models using the same TEM have the same embedding dimensionality with the training setting reported in their paper.

For fair comparison, both Ts-Retriever and TempRetriever are evaluated without query routers, as our method does not employ one.

D.2 Parameter-efficient fine-tuning baselines

- **LoRA**: We apply LoRA to all linear layers of the TEM. The LoRA rank is set to $r = 4$, the scaling factor is 4, and the dropout rate is 0.1.
- **Matryoshka-Adaptor**: As the code is not

publicly available, we follow the original paper and a prior work from the same group (Yoon et al., 2024a). The adaptor is implemented as a linear layer with the same hidden dimension as the embedding and includes a skip connection. The adaptor is applied before the pooling operation.

- **MRL**: This method follows the same setup as LoRA and train with $\mathcal{L}_{\text{MRL}}$.

D.3 Temporal-aware Matryoshka Representation Learning (TMRL)

TMRL follows the same settings as the MRL baseline. The projector $\mathcal{P}$ is discarded after training.

We use $k = 10$ for Eq. (9). The hyperparameters in Eq. (13) are summarized in Table 19.

Regarding benchmarking retrieval latency, we take the average time of 100 retrieval process using precomputed query and corpus indices. Each iteration of retrieval only considers the approximate nearest neighbor search operation using FAISS (Johnson et al., 2019).

D.4 LLMs’ settings for Temporal RAG

We use the FlashRAG framework (Jin et al., 2025b) and mainly Qwen3-8B (Yang et al., 2025) as the LLM for generation to conduct RAG experiments. We use the top-5 retrieved documents from the retriever as context input to the LLM.

Qwen3-8B’s and Qwen3-32B’s hyperparameters are as follows: max_tokens=8192, temperature=0.7, top_p=0.8, top_k=20, min_p=0.

DeepSeek-R1-Distill-Qwen-32B’s hyperparameters are as follows: max_tokens=8192, temperature=0.6, top_p=0.95, top_k=20, min_p=0.

Llama-3.3-70B-Instruct-bnb-4bit’s hyperparameters are as follows: max_tokens=8192, temperature=0.6, top_p=0.9, top_k=20, min_p=0.

The system prompt is: “Answer the question based on the given document. Only give me the answer and do not output any other words. The following are given documents. {reference}.” The user prompt is: “Question: {question} Answer:”

D.5 Details about temporal contrastive learning data augmentation

We use Qwen/Qwen3-4B-Thinking-2507 (Yang et al., 2025) for generating the augmented positive/negative queries, instructed with a prompt template (Table 26). Within this prompt, we use the JSON schema and Allen relations defined in Tables 28 and 27, respectively. The prompt includes few-shot examples drawn from the original TNP dataset, with temporal expressions extracted using SUTime. We also provide the prompt used for constructing the TimeQA qrel files in Table 29.

The LLM also extracts temporal spans from queries and passages, refined with preliminary SUTime outputs, without introducing new facts to preserve dataset fidelity. The LLM hyperparameters are as follows: max_tokens=32768, temperature=0.7, top_p=0.8, top_k=20, min_p=0.

Post-processing filtering analysis. After generation, we apply a heuristic post-processing filtering step motivated by careful inspection of the raw

outputs. These filtering steps are designed to reduce temporally irrelevant or ambiguous supervision and prevent bias toward the LLM reference year. Specifically, we:

- Remove training samples with insufficient number of positive or negative query.
- Remove queries containing the current reference year used by the LLM (e.g., 2026), which may introduce artifacts in relative temporal expressions such as “this year”, “last year”, etc.
- Remove passages and queries containing ambiguous relative expressions such as “now”, “today”, “yesterday”, etc.

In Tables 20 and 21, we empirically verify the proposed post-processing filtering. After removing relative expressions and reference-year artifacts, TMRL contriever consistently achieves improved performance across embedding dimensions, supporting the effectiveness of the filtering process and indicating that any remaining noise does not dominate the training signal.

E Datasets Statistics

Temporal Nobel Prize (TNP). We show the detailed statistics of the TNP dataset (Wu et al., 2024) and our augmented version in Table 22. Table 23 further details the statistics of temporal query types and Allen relations as a by-product of our Temporal Contrastive Learning augmentation pipeline in Section 3.3.

Time-sensitive QA (TimeQA). We present detailed statistics of the TimeQA dataset (Chen et al., 2021) and our augmented version in Table 24. Table 25 further reports the statistics of temporal query types and Allen relations as a byproduct of our Temporal Contrastive Learning augmentation pipeline described in Section 3.3.

TimeQA is a popular TQA benchmark constructed from Wikipedia, where high semantic similarity between queries and documents makes evaluating temporal encoding challenging (Wu et al., 2024). To adapt TimeQA into a temporal information retrieval benchmark, we perform careful preprocessing to mitigate the issue of queries and passages being overly semantically aligned (Wu et al., 2024). We use the Wikipedia snapshot enwiki-20211220-pages-articles, which is the only publicly accessible snapshot with dates closest to those used in TimeQA. We use index_enwiki-20220820 Wikipedia index database. This snapshot is first processed follow-

Embedding dimension	Recall@100 (No filtering) ↑	Recall@100 (Filtered) ↑	Relative Gain (%) ↑
64	48.1	48.5	0.8
128	65.0	65.7	1.1
256	75.5	76.3	1
512	77.9	78.2	0.4
768	80.2	80.5	0.3

Table 20: Impact of data filtering on Recall@100 performance of TMRL Contriever on TNP

Embedding dimension	nDCG@10 (No filtering) ↑	nDCG@10 (Filtered) ↑	Relative Gain (%) ↑
64	38.8	39.3	1.2
128	53.2	53.8	1.1
256	63.1	63.8	1.1
512	65.9	66.2	0.5
768	68.0	68.4	0.5

Table 21: Impact of data filtering on nDCG@10 performance of TMRL Contriever on TNP

ing FlashRAG (Jin et al., 2025b)², and then filtered using SUTime to retain only pages that contain at least one temporal expression and include an answer from the TimeQA train, development, or test sets.

After splitting and generating our augmented dataset, we further filter the original training set to retain only passages whose Wikipedia IDs match those in the augmented dataset, resulting in the “Original (Modified)” column in Table 24. We use this training set as the baseline data for our augmentation pipeline. We do not modify the development set so that model selection is still evaluated under a semantically dominant retrieval setting.

To construct the test corpus, we combine all training, development, and test files from both the easy and hard splits. We then use the index database to map the corresponding passages’ Wikipedia URLs to Wikipedia IDs. All pages with non-null Wikipedia IDs from the selected snapshot are accumulated to form the test corpus. To create the test qrel and corpus files, we retain only queries that are answerable.

Additionally, for each page, we use chonkie (Minhas and Nigam, 2025) to further chunk the filtered corpus into smaller paragraphs using the o200k_base tokenizer with a chunk size of 128. Finally, for each test query, we prompt Qwen/Qwen3-4B-Instruct-2507 against all chunked paragraphs from the corresponding Wikipedia pages, assisted by SUTime filtering and manual inspection. This process ultimately

results in the test set summarized in Table 24. The prompts used for qrel construction are provided in Table 29.

²https://github.com/RUC-NLP/FlashRAG/blob/main/docs/original_docs/process-wiki.md

Temporal Nobel Prize	Train		Dev		Test
	Original	Our	Original	Our	
# Unique documents	8060	5922	165	306	989
# Passages	8060	11693	165	1630	989
# Positive queries	23388	64564	468	1520	3244
# Negative queries	23388	60563	468	146	-
# Qrel	-	-	-	-	13110

Table 22: Dataset statistics of the original Temporal Nobel Prize dataset and our augmented version.

Temporal Nobel Prize	Train			Dev		
Allen Relations	Explicit	Implicit	Temporal Answer	Explicit	Implicit	Temporal Answer
Empty	42	3	13298	0	0	304
After	36929	527	-	877	10	-
Before	38058	262	-	937	3	-
Contains	816	2	-	23	0	-
During	17463	230	-	451	7	-
Equals	9856	102	-	223	2	-
Finishes	134	11	-	7	0	-
FininshedBy	11	0	-	0	0	-
Meets	98	10	-	1	0	-
MetBy	61	29	-	4	0	-
Overlaps	6559	40	-	149	1	-
OverlappedBy	49	1	-	0	0	-
Starts	513	7	-	15	0	-
StartedBy	17	7	-	0	0	-
Total	110606	1223	13298	2687	23	304

Table 23: Augmented Temporal Nobel Prize extra annotations as byproduct of our augmentation pipeline.

Time Sensitive QA	Train			Dev	Test
	Original	Original (Modified)	Our		
# Unique Wikipedia documents	3401	2774	2774	622	4926
# Passages	9166	5143	8149	826	109182
# Positive queries	26936	15051	44015	3162	5167
# Negative queries	0	0	41926	0	-
# Qrel	-	-	-	-	5247

Table 24: Dataset statistics of the original, modified, and augmented TimeQA dataset.

TimeQA	Train		
Allen Relations	Explicit	Implicit	Temporal Answer
Empty	19	116	8514
After	24224	320	-
Before	28188	235	-
Contains	1024	1	-
During	12081	134	-
Equals	6551	54	-
Finishes	121	2	-
FininshedBy	2	0	-
Meets	52	5	-
MetBy	23	12	-
Overlaps	3922	21	-
OverlappedBy	52	0	-
Starts	250	7	-
StartedBy	10	1	-
Total	76519	908	8514

Table 25: Augmented TimeQA extra annotations as byproduct of our augmentation pipeline.

Prompt template for temporal annotation

You are a temporal annotation expert for information retrieval. We provide you with a query that may contain one or many temporal expressions and the corresponding SUTIME's output for your reference. We also provide you with previously annotated examples of positive and negative samples of the query; use them as references if they are semantically relevant to the query. Your job is to generate high-quality positive passages and negative passages for temporal contrastive learning. Your final output are annotated JSONs (no indentation) that strictly follow the provided JSON schema.

Your temporal annotations must be more precise and reliable than SUTIME or HeidelTime, with top-tier accuracy. You must follow the definitions and instructions below.

* Allen relations and descriptions: <Allen relations and descriptions>

* Instructions:

1. You must output one or multiple valid JSONs, delimited by a newline, strictly following this pydantic schema: <Pydantic JSON Schema>

2. "query_id": must be assigned with the provided "query_id".

3. "query": - A query is a provided sentence that may contain one or many temporal expressions. - Please note that SUTIME only provides explicit temporal expressions for the query and they are not perfect; therefore, you must double-check them and further detect additional explicit and implicit temporal expressions such as events. - You must extract all events that are anchored to specific datetime (e.g., "2000 FA Cup Final", "the 2007 election", etc.). - The "temporal" field should be extracted as written from the query's text and they must be concise, such as: "in April, 1906", "2 July 2010", "2004 general election", etc.

4. "positive_passages": - Natural, QA-style questions with diverse phrasing that seek information from the query. Each question must contain exactly one extractable temporal expression, logically align with one of the query's temporal expressions, yet be diverse in Allen relations. - Based on the query's temporal expressions, you must generate "positive_passages" that cover all "TemporalQueryType", including "Explicit", "Implicit", and "TemporalAnswer", when possible: - You MUST prioritise generating questions with explicit temporal constraints, like "in 2010", "from 2010 to 2015", etc. They must logically align with the query's temporal expression(s), such as being equals, overlapping with, or being contained within the query's temporal expression(s). If the query has multiple temporal expressions, the generated questions must logically align with at least one of them. - You should also prioritise generating questions with implicit temporal constraints, such as "after EVENT", "before EVENT". These must logically align with the query's temporal expression(s). - For both explicit and implicit "TemporalQueryType" questions, you must not use phrases like what/which date/day/month/year/time or when etc., that inquire about time. You must not confuse this with TemporalAnswer questions. - You can also generate "TemporalAnswer" (asking for datetime/duration/time-range of an EVENT, e.g., "When did EVENT happen?", "What time did he arrive?"). For this type of question, you must not add any temporal expression and you must set: "TemporalQueryType": "TemporalAnswer", "allen_relation": "Empty", and "temporal": []. Use sparingly (less than 2 per query, only when explicit/implicit cannot be formed). - "temporal" field: must extract all exact text spans of the temporal expressions that exist in the generated question (not normalized or paraphrased). Regarding explicit and implicit "TemporalQueryType", they must be concise temporal expressions as written, such as "after July 2010", "from 2012 to 2014", that are not just normalized dates. For instance, instead of "In 1906 he moved with his family to a farm", prefer the concise "In 1906" and keep prepositions or context words that anchor the time, e.g., 'from', 'in', 'after', etc. Regarding "TemporalAnswer" passages, the "temporal" field must be an empty list. - Allen relation: consider Passage = Event A and Query = Event B. Must be correct and align with the provided definition. - "docid": must remain the same as provided. - Quantity: For each split/rewritten query, you must generate a list of 5 high-quality positive passages, prioritizing quality over quantity.

5. "negative_passages": follow the same format as "positive_passages", but represent contexts that are temporally mismatched or semantically irrelevant to the query. Based on the query's temporal expressions, you must generate "negative_passages" that cover all "TemporalQueryType", including "Explicit", "Implicit", and "TemporalAnswer", when possible. They must be hard, temporally-confused, yet diverse in Allen relations questions that cover all following cases: - Case 1: Questions with temporal expressions that mismatch with the query's temporal expression(s): - If the query specifies a span (e.g., 2005–2007), use an interval that falls completely outside it (e.g., 2008, before 2005, after 2007). - Adjacent years or ranges are valid negatives (e.g., query = 2010, negative = 2009). - Use shifted but non-overlapping intervals (e.g., query = 2010, negative = 2012–2014). - Include misleading implicit cues (e.g., "shortly after 2011" vs. query "in 2010"). - You may reuse the same passage from "positive_passages" but replace its temporal expressions with mismatched ones. - Case 2: Questions with same temporal expression but irrelevant event/entity - Questions with overlapping or identical temporal expressions but targeting a different subject. - Example: query = "Ronaldo's career in 2010" vs. negative = "Messi's career in 2010". - This ensures negatives are temporally aligned but semantically irrelevant. - Case 3: "TemporalAnswer"-type questions that are either: - Irrelevant to the query. - Or ask for non-existent temporal information in the query context. - Use sparingly (less than 2 per query, only when explicit/implicit cannot be formed). - Allen relation: consider Passage = Event A and Query = Event B. Must be correct and align with the provided definition. - Quantity: For each split/rewritten query, you must generate a list of 5 high-quality negative passages, prioritizing quality over quantity. - Do not create trivial negatives (e.g., completely unrelated random text). - Ensure no accidental overlap with valid facts in the document.

6. "temporal_query_type": Must be "Explicit", "Implicit", or "TemporalAnswer". If the temporal contains a clear date/month/year, it is "Explicit", NOT "Implicit". If the passage asks for which date/month/year of an event, it is "TemporalAnswer".

7. Final output: Only output valid JSON(s). Do not explain, add comments, or include extra text, since your output will be parsed automatically.

Demonstration 1

<Demo 1>

End of Demonstration 1

Demonstration 2

<Demo 2>

End of Demonstration 2

Table 26: Detailed prompts for augmenting the positive/negative queries. The Allen relations and Pydantic JSON schemas are defined in Table 27 and Table 28, respectively.

Allen relations' descriptions

- Before: Did 'Event A' occur before 'Event B' without any overlap between the two events?
 - After: Did 'Event A' occur after 'Event B' without any overlap between the two events?
 - Meets: Did 'Event A' end in the same time as 'Event B' began? Answer True or False.
 - MetBy: Did 'Event B' end in the same time as 'Event A' began? Answer True or False.
 - Overlaps: Did 'Event A' begin before 'Event B' and end before 'Event B' ended, with some overlap between the two events?
 - OverlappedBy: Did 'Event B' begin before 'Event A' and end before 'Event A' ended, with some overlap between the two events?
 - Starts: 'Event A' begin in the same time as 'Event B', but end before 'Event B' ended?
 - StartedBy: Did 'Event B' begin in the same time as 'Event A', but end before 'Event A' ended?
 - During: Did 'Event A' begin after 'Event B' began and end before 'Event B' ended, being entirely contained within 'Event B'?
 - Contains: Did 'Event A' begin before 'Event B' began and end after 'Event B' ended, entirely containing 'Event B'?
 - Finishes: Did 'Event A' begin after 'Event B' began and end in the same time as 'Event B'?
 - FinishedBy: Did 'Event B' begin after 'Event A' began and end in the same time as 'Event A'?
 - Equals: Did 'Event A' begin in the same time as 'Event B' and end in the same time as 'Event B'?
 - Empty: a special case for TemporalAnswer where there is no temporal expression.
-

Table 27: Detailed prompts for Allen relations descriptions. Except for "Empty", the descriptions are provided by [Islakoglu and Kalo \(2025\)](#).

Pydantic JSON schema

```
('$defs': {'AllenRelation': {'enum': ['Before', 'After', 'Meets', 'MetBy', 'Overlaps', 'OverlappedBy', 'Starts', 'StartedBy', 'During', 'Contains', 'Finishes', 'FinishedBy', 'Equals', 'Empty'], 'title': 'AllenRelation', 'type': 'string'}, 'Passage': {'properties': {'docid': {'title': 'Docid', 'type': 'integer'}, 'text': {'title': 'Text', 'type': 'string'}, 'temporal': {'items': {'type': 'string'}, 'title': 'Temporal', 'type': 'array'}, 'temporal_query_type': {'$ref': '#/$defs/TemporalQueryType', 'allen_relation': {'$ref': '#/$defs/AllenRelation', 'required': ['docid', 'text', 'temporal', 'temporal_query_type', 'allen_relation'], 'title': 'Passage', 'type': 'object'}, 'TemporalQueryType': {'enum': ['Explicit', 'Implicit', 'TemporalAnswer'], 'title': 'TemporalQueryType', 'type': 'string'}, 'properties': {'query_id': {'title': 'Query Id', 'type': 'integer'}, 'query': {'title': 'Query', 'type': 'string'}, 'temporal': {'items': {'type': 'string'}, 'title': 'Temporal', 'type': 'array'}, 'positive_passages': {'items': {'$ref': '#/$defs/Passage', 'title': 'Positive Passages', 'type': 'array'}, 'negative_passages': {'items': {'$ref': '#/$defs/Passage', 'title': 'Negative Passages', 'type': 'array'}, 'required': ['query_id', 'query', 'temporal', 'positive_passages', 'negative_passages'], 'title': 'TemporalAnnotation', 'type': 'object'}}
```

Table 28: Detailed prompts for Pydantic JSON schema

Prompt template for temporal annotation

Input format (index, query, list of answer_terms, passage)

Task

Decide whether the passage is relevant to the query for temporal information retrieval (qrel construction). A passage is relevant if and only if ALL of the following conditions are satisfied:

1. Temporal correctness (mandatory): The passage must clearly satisfy the temporal constraint(s) explicitly or implicitly expressed in the query (e.g., in, before, after, between, from ... to ...). - Explicit dates or time spans in the passage are acceptable.

- Implicit temporal coverage is acceptable only if it unambiguously covers the queried time period (e.g., "served from 1989–1993" satisfies "in 1991").

- Vague, approximate, or underspecified temporal expressions (e.g., "around that time", "in the early days", "in the 1900s") do not satisfy temporal constraints.

- If the temporal alignment between the query and passage is unclear, the passage must be judged not relevant.

2. Answer-term validity within time: The passage must contain all answer_terms. - The temporal information must be directly associated with the answer_terms, meaning the answer_terms refer to the same entity, event, or state that is valid within the specified time frame. A passage is relevant if and only if ALL of the following conditions are satisfied:

- Merely mentioning the answer_terms elsewhere in the passage without temporal grounding is insufficient.

3. No external inference: Judge relevance only based on the information explicitly present in the passage.

- Do not rely on external knowledge, common facts, or assumptions.

- Do not infer missing dates or temporal relations.

4. Binary decision rule: If any of the above conditions is not satisfied, the passage is not relevant.

Output format

- Output the integer index only if the passage is relevant.

- Otherwise, output -1.

- Output only the index or -1.

- Do not include explanations, reasoning steps, or additional text.

- Do not guess or assign partial relevance.

Demonstration 1

Input:

(Index: 123, Question: Who was the head coach of the team 1. FC Köln from Nov 2019 to Nov 2020?, Answers: ['Markus Gisdol'], Passage: "Section: Decline and changes (2018–). FC Saarbrücken, the club decided to terminate Beierlorzer's contract on 9 November 2019. Sporting director Armin Veh, who weeks earlier had announced that he would not extend his contract with the club, was also dismissed from his position. On 18 November, former HSV manager Markus Gisdol was appointed to the club's head coaching position, while Horst Heldt was made sporting director. Both signed contracts until 2021. After avoiding relegation at the end of the season, Gisdol's contract was extended until 2023.)")

Output: 123

End of Demonstration 1

Demonstration 2

Input:

(Index: 2, Question: Who was the head coach of the team 1. FC Köln from Nov 2019 to Nov 2020?, Answers: ['Markus Gisdol'], Passage: "Section: Decline and changes (2018–). The club found itself in a renewed relegation during the 2020–21 season. On 11 April 2021, after losing to relegation rival Mainz 05, Gisdol was dismissed from his position as head coach. The next day, it was announced that Friedhelm Funkel would take over head coaching duties until the end of the season. On 11 May, it was reported that SC Paderborn manager Steffen Baumgart would succeed Funkel as head coach at the beginning of the 2021–22 season.)")

Output: -1

End of Demonstration 2

Demonstration 3

Input:

(Index: 3, Question: Who was the head coach of 1. FC Köln from Nov 2019 to Nov 2020?, Answers: ['Markus Gisdol'], Passage: "Gisdol was appointed head coach in November 2018 and left in October 2019 before the season ended."))

Output: -1

End of Demonstration 3

Table 29: Detailed prompts for generating the TimeQA's test qrel file.