Temporal Information Retrieval via Time-Specifier Model Merging

SeungYoon Han[1] **Taeho Hwang**[1] **Sukmin Cho**[1] **Soyeong Jeong**[2]
Hoyun Song[1] **Huije Lee**[1] **Jong C. Park**[1*]
[1]School of Computing, [2]Graduate School of AI
Korea Advanced Institute of Science and Technology (KAIST)
{seungyoonee,doubleyyh,nelllpic,starsuzi,
hysong,huijelee,jongpark}@kaist.ac.kr

Abstract

The rapid expansion of digital information and knowledge across structured and unstructured sources has heightened the importance of Information Retrieval (IR). While dense retrieval methods have substantially improved semantic matching for general queries, they consistently underperform on queries with explicit temporal constraints–often those containing numerical expressions and time specifiers such as "in 2015." Existing approaches to Temporal Information Retrieval (TIR) improve temporal reasoning but often suffer from catastrophic forgetting, leading to reduced performance on non-temporal queries. To address this, we propose Time-Specifier Model Merging (TSM), a novel method that enhances temporal retrieval while preserving accuracy on non-temporal queries. TSM trains specialized retrievers for individual time specifiers and merges them into a unified model, enabling precise handling of temporal constraints without compromising non-temporal retrieval. Extensive experiments on both temporal and non-temporal datasets demonstrate that TSM significantly improves performance on temporally constrained queries while maintaining strong results on non-temporal queries, consistently outperforming other baseline methods. Our code is available at `https://github.com/seungyoonee/TSM`.

1 Introduction

In the contemporary era of digital information, Information Retrieval (IR)–the process of finding and ranking documents from a large collection that are most relevant to a search query–has become increasingly important as information and knowledge rapidly expand across both structured sources (e.g., knowledge bases) (Lan et al., 2021; Dhingra et al., 2022) and unstructured sources (e.g., Wikipedia, web documents) (Vrandečić and Krötzsch, 2014). This significance is more amplified in the era of Large Language Models (LLMs), where IR is a crucial component of Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Khandelwal et al., 2020) pipelines.

As the importance of IR continues to grow, there have been significant advances in retrieval methods, notably the development of dense retrieval methods (Karpukhin et al., 2020; Izacard et al., 2022). Dense retrieval leverages neural models to encode both queries and documents into dense embeddings to capture semantic similarity, substantially improving retrieval effectiveness for general-domain queries. However, these models exhibit *attention bias*, where their embeddings are optimized primarily for semantic similarity and topical relevance, making them less effective at capturing temporal expressions in queries (Wu et al., 2024). As a result, dense retrievers struggle with queries containing temporal expressions (e.g., "in 2015," "between 2010 and 2012") (Chen et al., 2021).

To address these challenges, the field of Temporal Information Retrieval (TIR) has emerged, focusing on improving retrieval accuracy for temporal queries by enhancing temporal understanding capabilities of retrievers (Allen, 1983; Alonso et al., 2011). Recent research has attempted to increase the time-awareness of dense models from the pre-training process using different temporal information masking (Rosin et al., 2021; Wang et al., 2023; Cole et al., 2023), fine-tuning process (Chen et al., 2021; Dhingra et al., 2022; Wu et al., 2024). By incorporating temporal awareness, TIR aims to enhance the accuracy and relevance of retrieved documents for temporal queries.

Previous studies have primarily focused on improving retrieval performance for temporal queries, often overlooking the resulting performance drop on non-temporal queries. However, while enhancing temporal retrieval capabilities is important, it is equally crucial to maintain robust performance on non-temporal queries. This is because both tem-

[*] Corresponding author

poral and non-temporal queries are fundamentally part of general-domain information retrieval and do not require domain-specific knowledge.

Unlike domain-specific retrieval tasks that target specialized topics, temporal queries remain general in scope, with their distinction based solely on the presence of explicit time constraints–typically signaled by time specifiers such as “in,” “after,” or “between.” Accordingly, this paper treats temporal queries as a subset of general queries with explicit time constraints, while non-temporal queries lack such time specifiers. This distinction highlights the need for retrieval models that can flexibly and effectively handle both query types without sacrificing overall performance.

Despite this need for balanced retrieval capabilities, fine-tuning dense models to improve accuracy on temporal queries often comes at a significant cost: a noticeable decline in performance on general, non-temporal queries, primarily due to catastrophic forgetting (Goodfellow et al., 2014; Luo et al., 2023). For instance, as illustrated in Figure 1, fine-tuning Contriever (Izacard et al., 2022) on TimeQA (Chen et al., 2021) enhances temporal retrieval but substantially reduces performance on the general-domain dataset Natural Questions (NQ) (Kwiatkowski et al., 2019).

To address this issue, Wu et al. (2024) and Abdallah et al. (2025) proposed a routing-based method that directs temporal queries to a temporally fine-tuned retriever and non-temporal queries to a vanilla retriever, which helps mitigate catastrophic forgetting. However, this approach requires maintaining and operating two separate dense retrievers models, resulting in an increased memory usage, which can be resource-intensive in practical deployments. Furthermore, while this method helps preserve performance across both query types, it heavily relies on accurate classification of queries as temporal or non-temporal, which can result in suboptimal retrieval accuracy, as shown in Table 2.

To address the challenge of handling both temporal and non-temporal queries, we propose Time-Specifier Model Merging (TSM), a novel temporal fine-tuning method. TSM involves separately training specialized retrievers on data subsets corresponding to specific time specifiers (e.g., “in,” “after,” “between”) for temporal queries with explicit expressions. Each retriever develops expertise in a particular temporal constraint. We then merge these specialized models by simply averaging their parameters, allowing the unified retriever to inherit the specialized performance of each time-specifier-specific model.

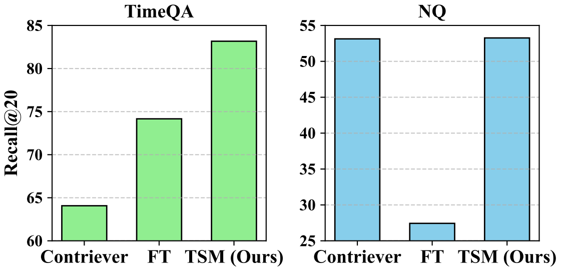

Figure 1: Recall@20 performance of vanilla Contriever, full-parameter fine-tuning (FT), and TSM (Ours) on the temporal dataset TimeQA (green) and the non-temporal general-domain dataset Natural Questions (blue).

This merging process is effective at mitigating catastrophic forgetting because it results in lower-magnitude weight changes–preserving knowledge from both temporal and non-temporal data, rather than overwriting it as in standard fine-tuning (Alexandrov et al., 2024; Yang et al., 2024). As a result, the merged model can more effectively encode temporal relevance associated with each time specifier while still maintaining strong performance on non-temporal queries. Extensive experiments on both temporal and non-temporal datasets demonstrate that TSM significantly improves performance on temporal queries while preserving performance on non-temporal datasets. TSM consistently outperforms alternative temporally-aware training methods, including full fine-tuning, regularization, LoRA, routing, and ensembling.

To summarize, our contributions are threefold:

- We identify and address the critical challenge of improving temporal retrieval performance without compromising non-temporal (general-domain) retrieval accuracy, emphasizing the need for retrieval models that can flexibly handle both query types.
- We propose a novel **Time-Specifier Model Merging (TSM)** method, which fine-tunes separate, specialized retrievers for individual time specifiers and then merges them into a unified model. This method enables precise handling of temporal constraints while effectively preserving general retrieval capabilities.
- Through extensive experiments on both temporal and non-temporal datasets, we demonstrate that TSM significantly improves performance on temporal queries without sacrificing non-temporal retrieval accuracy, consistently outperforming other fine-tuning strategies.

2 Related Work

Temporal Information Retrieval Temporal Information Retrieval (TIR) is a specialized subfield of Information Retrieval (IR) focused on accurately interpreting temporal information in both user queries and documents (Allen, 1983; Alonso et al., 2011). Temporal information refers to specific points in time (e.g., "in 2015"), intervals (e.g., "between 2010 and 2012"), and can be expressed in various forms: *explicit* (e.g., "January 2010"), *relative* (e.g., "tomorrow"), or *implicit* (e.g., "Labor Day") (Kanhabua and Anand, 2016). Temporal queries typically involve time specifiers such as "after" or "between" to define temporal constraints. TIR research addresses challenges such as temporal query analysis, time-aware embedding, and the extraction of temporal expressions to improve temporal retrieval effectiveness. Our work builds on these developments, aiming to enhance retrieval performance for temporally relevant information, with a focus on *explicit* temporal expressions.

Semantic vs. Temporal Focus in Dense Models Dense retrieval models (Karpukhin et al., 2020; Izacard et al., 2022) have advanced Information Retrieval (IR) but still struggle with temporal information retrieval (TIR). This is because their embeddings are primarily optimized for semantic similarity and topical relevance, rather than explicit temporal expressions–a limitation known as *attention bias* (Wu et al., 2024). To address this, recent studies have introduced temporal information masking strategies during pre-training, enabling models to better encode explicit temporal expressions, which leads to improved temporal representations (Rosin et al., 2021; Dhingra et al., 2022; Wang et al., 2023; Cole et al., 2023). Other approaches, such as TempRALM, enhance retrievers with temporal scoring mechanisms to more accurately rank documents based on temporal relevance (Gade and Jetcheva, 2024). While these methods improve retrieval performance for temporal queries, they often overlook the resulting decline in performance on non-temporal queries.

Among the approaches addressing both temporal and non-temporal retrieval using off-the-shelf dense models, Wu et al. (2024) and Abdallah et al. (2025) proposed a routing-based method that directs temporal queries to a retriever fine-tuned on temporal datasets and non-temporal queries to a vanilla retriever, mitigating catastrophic forgetting. While this preserves performance across query types, it heavily relies on accurate query classification, which can result in suboptimal performance. In this study, we focus on fine-tuning off-the-shelf dense retriever models to handle both temporal and non-temporal queries within a single model, eliminating the dependence on additional modules for query classification.

Mitigating Catastrophic Forgetting Catastrophic forgetting occurs when a model, after being fine-tuned on a new task or domain, loses performance or knowledge on previously learned tasks (Goodfellow et al., 2014; Luo et al., 2023). Regularization is a fundamental technique to address this, constraining parameter updates during fine-tuning to preserve pre-trained knowledge (Kirkpatrick et al., 2016; Li and Hoiem, 2016; Triki et al., 2017). Low-Rank Adaptation (LoRA) is another effective approach, which introduces a small number of trainable low-rank matrices while keeping most weights frozen (Hu et al., 2021). LoRA and its variants have shown strong performance in continual and out-of-domain learning by isolating task-specific updates and preserving prior knowledge, helping to reduce catastrophic forgetting (Lee et al., 2023).

Another approach is ensemble learning, which combines the predictions of multiple models–each specialized for different tasks or domains – to achieve balanced performance (Ganaie et al., 2021; Ibomoiye and Sun, 2022; Mohammed and Kora, 2023). However, this approach requires running multiple models simultaneously, increasing both memory usage and inference costs. Routing-based methods have also been proposed, dynamically directing queries to either a fine-tuned or the vanilla model based on the query type (Wu et al., 2024; Abdallah et al., 2025). While routing leverages the strengths of both specialized and general models, its effectiveness depends on accurate query classification and still requires maintaining multiple models, making it resource-intensive in practice.

Model merging has recently emerged as a simple and effective approach to mitigating catastrophic forgetting by flattening high-magnitude weight changes during adaptation, resulting in more stable and higher-quality parameter updates (Alexandrov et al., 2024; Yang et al., 2024). Motivated by these findings, we adopt model merging in this study and propose a novel temporal fine-tuning method. Our method fine-tunes specialized retrievers for individual time specifiers and merges them into a unified model, enabling effective retrieval for both temporal and non-temporal queries.

3 Method

We define the temporal and non-temporal retrieval problem and introduce out method, Time-Specifier Model Merging (TSM).

3.1 Problem Formulation and Preliminaries

We begin by defining the information retrieval task, distinguishing between temporal and non-temporal, and introducing key concepts and notations used throughout our method.

Information Retrieval (IR). IR identifies a subset of documents $D = \{d_1, d_2, \ldots, d_k\}$ from a corpus C that are most relevant to a given user query q. Formally, the retrieval process can be defined as:

$$D = \{d_1, \ldots, d_k\} = \texttt{Retriever}(q, \mathcal{C}), \tag{1}$$

where the `Retriever` function returns the top-k documents from $\mathcal{C}$ ranked by their relevance to q.

Dense Retrieval. Dense retrieval encodes queries and documents into dense vector representations using neural encoders. Let f_θ denote an encoder parameterized by θ, which maps q and d_i to dense vectors:

$$\mathbf{q} = f_\theta(q),\ \mathbf{d}_i = f_\theta(d_i),\ \forall d_i \in \mathcal{C} \tag{2}$$

The relevance score between a query and a document is computed via the dot product of their vector representations:

$$sim(\mathbf{q}, \mathbf{d}_i) = \mathbf{q}^\top \mathbf{d}_i \tag{3}$$

and the retriever selects documents with the highest similarity scores.

Temporal and Non-Temporal Queries. Let Q denote the set of all general-domain queries. The subset of temporal queries $Q_T \subseteq Q$ is defined as:

$$Q_T = \{q_T \in Q \mid q_T = (s, t), s \in \mathcal{S}, t \in \mathcal{T}\} \tag{4}$$

where $\mathcal{S}$ is the set of time specifiers: $\mathcal{S} = \{\text{before}, \text{between}, \ldots\}$, and $\mathcal{T}$ is the set of specific temporal point or period: $\mathcal{T} = \{\text{Apr 2020}, [1990{,}2000], \ldots\}$. The subset of non-temporal queries $Q_N \subseteq Q$ is given by:

$$Q_N = Q \setminus Q_T \tag{5}$$

such that $Q = Q_T \cup Q_N$ and $Q_T \cap Q_N = \emptyset$.

Objective of Our Method. The objective of our method is to address the newly defined problem of balancing effective temporal retrieval for temporal queries (Q_T) with robust performance on non-temporal queries (Q_N), ensuring that improvements in one do not come at the expense of the other.

Time Specifier	Train	Dev
from $[time_1]$ to $[time_2]$	11,676	2,486
in $[time]$	5,759	1,233
between $[time_1]$ and $[time_2]$	4,888	1,054
after $[time]$	2,741	587
before $[time]$	2,867	609
in early $[time]$s	1,885	438
in late $[time]$s	2,392	474
Total	32,208	6,881

Table 1: Statistics of the augmented TimeQA dataset showing the number of queries containing each time specifier in the training and development sets.

3.2 Time-Specifier Model Merging (TSM)

Now, we introduce our method, TSM, for improving temporal retrieval performance while maintaining strong non-temporal retrieval capabilities. TSM first fine-tunes dense retrieval models on data sampled according to each time specifier, and then merges their parameters to create a unified retriever.

3.2.1 Data Sampling

We utilize TimeQA (Chen et al., 2021) for fine-tuning dense retrievers. Following the TimeQA taxonomy of seven time specifiers–in $[time]$, after $[time]$, before $[time]$, in early $[time]$s, in late $[time]$s, between $[time_1]$ and $[time_2]$, and from $[time_1]$ to $[time_2]$– we categorize the dataset into seven groups based on these specifiers. Each $[time]$ refers to a specific year or a year with a month. However, the original TimeQA training set is imbalanced across the time specifiers. To address this, we use the official TimeQA data processing scrips and annotation labels to augment the comparatively less frequent time specifiers: *after*, *before*, *in early*, and *in late*. As a result, we increase the training set from 25,064 to 32,208 instances and the dev set from 5,348 to 6,881. Detailed statistics for the original dataset are provided in Appendix A.2. Note that we only use answerable questions with gold answers, as non-answerable questions do not have gold answers and therefore cannot be used for contrastive learning, since there would be no positive passages available. Table 1 summarizes the statistics of the augmented dataset for each time specifier.

3.2.2 Specifier-Specific Fine-Tuning

For each time specifier s, we fine-tune a separate dense retriever on the corresponding subset of sampled data. We employ a contrastive learning objective with the InfoNCE loss (Izacard et al., 2022).

For a given temporal query q_T, the loss is defined as:

$$L(q_T, p^+) = -log \frac{e^{sim(q_T, p^+)/\tau}}{e^{sim(q_T, p^+)/\tau} + \sum_{i=1}^{n} e^{sim(q_T, p_i^-)/\tau}},$$

where p^+ is the positive passage (containing the gold answer), $\{p_i^-\}_{i=1}^n$ are n in-batch negative (Izacard et al., 2022) passages, $sim(q_T, p)$ is the dot-product similarity between the temporal query q_T and passages $p = \{p^+, p^-\}$, and τ is a temperature hyperparameter that controls the smoothness of the probability distribution.

3.2.3 Parameter Merging

After fine-tuning specifier-specific models with parameters $\theta_1, ..., \theta_k$, we merge them by simply averaging the parameters (Xiao et al., 2024):

$$\theta_{merged} = \frac{1}{k} \sum_{i=1}^{k} \theta_i. \tag{6}$$

The merged retriever is then used to encode both temporal queries and general, non-temporal queries.

This two-stage approach enables our method to leverage the fine-tuned representations learned from time specifier-specific data while maintaining a merged model for non-temporal retrieval tasks.

4 Experimental Setups

4.1 Datasets

We evaluate on four QA datasets: two that emphasize *temporal* retrieval–TimeQA (Chen et al., 2021) and Nobel Prize (Wu et al., 2024)–and two representing *non-temporal* retrieval tasks–Natural Questions (NQ) (Kwiatkowski et al., 2019) and MS MARCO (Nguyen et al., 2016). Below, we briefly describe each dataset and clarify our usage protocol.

TimeQA (Chen et al., 2021) consists of around 25K time-sensitive questions derived from WikiData (Vrandečić and Krötzsch, 2014). These queries focus on facts that evolve over time, requiring models to perform temporal understanding and reasoning. We evaluate on the original TimeQA test set in a closed-domain scenario, using the official document collection chunked by 100-word segments following Wang et al. (2019) and Karpukhin et al. (2020). **Nobel Prize** (Wu et al., 2024) dataset is a template-based corpus created from structured data on Nobel laureates. It includes about 3.2K time-sensitive queries, and we use the provided corpus and test set. **Natural Questions** (Kwiatkowski et al., 2019) is a benchmark for general QA tasks. We employ the test set from the BEIR benchmark (Thakur et al., 2021) to evaluate retrieval performance on general queries. **MS MARCO** (Nguyen et al., 2016) is a widely used benchmark for open-domain question answering. For evaluation, we use its validation set provided through the BEIR benchmark (Thakur et al., 2021).

4.2 Models

We employ **Contriever** (Izacard et al., 2022) for an *unsupervised* dense retriever, and **Dense Passage Retriever (DPR)** (Karpukhin et al., 2020) for a *supervised* dense retriever, allowing us to assess the effectiveness of baseline methods and our method on both unsupervised and supervised retrievers.

4.3 Baselines

We compare our method, TSM, against the following approaches:

Vanilla Dense Retrievers. Contriever and DPR, using their off-the-shelf checkpoints without any additional fine-tuning.

Full-Parameter Fine-Tuning (FT). Fine-tuning full parameters of Contriever and DPR on the entire TimeQA training set, without any sampling based on time specifier.

FT with Regularization. Full-parameter fine-tuning on the entire TimeQA training set with regularization (Kirkpatrick et al., 2016). Specifically, we use a dropout rate of 0.1 and a weight decay of 0.01 during training. Note that all other methods are trained with the same regularization as it is now fundamental in modern model training.

Low-Rank Adaptation (LoRA). LoRA fine-tuning (Hu et al., 2021) of Contriever and DPR on the entire TimeQA training set.

Routing. A query router that directs temporal queries to the retriever fully fine-tuned on TimeQA and sends general queries to the vanilla retriever, using the router checkpoint provided by Wu et al. (2024). The router is a two-layer feedforward neural network trained on TimeQA and Natural Questions (NQ) to perform binary classification of queries as either temporal or non-temporal.

Ensembling. We combine the outputs of multiple dense retrievers, each trained on a different time specifier. Similarity scores from each retriever are first normalized using min-max normalization for

Method	**TimeQA**				**Nobel Prize**				**NQ**				**MS MARCO**				**Average**			
	Recall		**nDCG**		**Recall**		**nDCG**		**Recall**		**nDCG**		**Recall**		**nDCG**		**Recall**		**nDCG**	
	@5	@20	@5	@20	@5	@20	@5	@20	@5	@20	@5	@20	@5	@20	@5	@20	@5	@20	@5	@20
Unsupervised Dense Retriever																				
Contriever	35.29	64.07	22.98	31.49	21.20	51.40	22.34	33.58	29.28	53.13	21.27	28.51	25.24	**45.99**	17.14	**23.20**	27.75	53.65	20.93	29.20
FT	57.40	71.12	45.20	49.25	14.94	39.31	14.05	23.46	11.32	22.69	7.75	11.10	13.80	24.97	9.58	12.80	24.37	39.52	19.15	24.15
FT + Reg	60.30	74.38	46.93	51.10	20.21	51.21	18.67	30.65	13.60	27.43	9.44	13.52	15.87	28.88	10.96	14.68	27.50	45.48	21.50	27.49
LoRA	65.20	80.20	49.63	54.13	11.04	27.52	11.54	17.52	27.06	44.69	20.09	25.47	20.40	37.17	14.14	18.98	30.93	47.40	23.85	29.03
Routing	50.15	74.35	35.36	42.54	25.96	62.42	26.47	40.22	29.28	53.13	21.27	28.51	25.09	45.71	17.04	23.08	32.62	58.90	25.04	33.59
Ensembling	63.46	77.31	48.94	53.04	34.39	71.47	35.13	49.12	25.49	45.65	18.04	24.14	22.36	39.97	15.39	20.51	36.43	58.60	29.38	36.70
TSM (Ours)	**68.73**	**83.49**	**53.45**	**57.89**	**35.33**	**75.58**	**35.73**	**50.83**	**32.58**	**53.26**	**23.66**	**29.95**	**25.26**	44.28	**17.36**	22.92	**40.48**	**64.15**	**32.55**	**40.40**
Supervised Dense Retriever																				
DPR	29.98	48.08	21.08	26.39	22.58	46.52	22.91	31.69	**58.20**	**76.55**	**46.95**	**52.67**	21.97	35.54	15.73	19.66	33.18	51.67	26.67	32.60
FT	52.17	66.20	41.13	45.30	13.75	34.92	13.32	21.35	18.55	30.69	13.83	17.51	6.64	12.70	4.54	6.28	22.78	36.13	18.21	22.61
FT + Reg	49.03	64.39	38.34	42.83	16.33	37.86	15.67	23.72	17.75	31.54	12.86	17.01	7.72	13.99	5.41	7.23	22.71	36.95	18.07	22.70
LoRA	65.64	78.40	51.31	55.12	24.56	50.26	23.98	33.62	47.25	62.95	38.02	42.83	17.87	30.97	12.85	16.62	38.83	55.65	31.54	37.05
Routing	35.25	52.34	25.21	30.24	19.22	42.21	19.68	28.10	**58.20**	**76.55**	**46.95**	**52.67**	21.97	35.53	15.74	19.67	33.66	51.66	26.90	32.67
Ensembling	64.11	76.67	50.38	54.11	30.71	58.02	**30.48**	40.80	43.70	60.87	34.63	39.90	19.91	33.61	14.05	18.01	39.61	57.29	32.39	38.21
TSM (Ours)	**66.61**	**79.21**	**52.53**	**56.30**	**30.78**	**60.63**	30.34	**41.72**	48.07	66.03	38.33	43.85	**23.26**	**37.80**	**16.64**	**20.84**	**42.18**	**60.92**	**34.46**	**40.68**

Table 2: Main results across all datasets and methods, evaluated using Recall and nDCG at top-$\{5, 20\}$ documents, with averages reported for each metric. Results are grouped by base retrievers: *Contriever-based* (unsupervised) and *DPR-based* (supervised). The best performance for each metric is shown in **bold**, and the second-best is underlined.

a given query. The normalized scores for each candidate passage are then averaged across retrievers to produce an ensemble score, and passages are ranked accordingly (Li et al., 2024).

Further implementation details are provided in Appendix A.4.

4.4 Evaluation Metrics

We report our main results evaluating retrieval performance using two standard metrics: **Recall** and **nDCG** at top-$\{5, 20\}$ documents. Recall measures the proportion of relevant documents successfully retrieved, while nDCG evaluates the quality of ranking by considering both relevance and position.

5 Main Results

Table 2 shows our results across four QA datasets: TimeQA and Nobel Prize as temporal datasets, and NQ and MS MARCO as non-temporal datasets. We evaluate both unsupervised (Contriever) and supervised (DPR) dense retrievers and compare our proposed method, TSM, against several baselines, including vanilla retrievers, full fine-tuning (FT), FT with regularization (FT + Reg), LoRA, routing, and ensembling.

On temporal datasets, TSM achieves the strongest performance across all metrics for both Contriever and DPR. For example, On TimeQA, TSM with Contriever achieves substantial improvements over the vanilla retriever. Similarly, on the Nobel Prize dataset–which serves as an out-of-domain temporal test set–TSM achieves the best performance for both unsupervised and supervised retrievers, confirming its strong generalization to unseen temporal data. Although ensembling yields a marginally higher nDCG@5 on Nobel Prize, TSM remains the most robust performer overall.

On non-temporal datasets, TSM also maintains competitive performance, achieving the strongest results across most metrics with both Contriever and DPR. On NQ, where DPR is trained in-domain, vanilla DPR achieves the highest Recall and nDCG. However, DPR-based TSM performs most closely to vanilla DPR on Recall@5/20 and nDCG@5/20, while outperforming FT, LoRA, and ensembling. On Contriever, which is not trained in-domain, TSM significantly improves retrieval effectiveness. On MS MARCO, which is out-of-domain for both Contriever and DPR, TSM achieves highly competitive performance. For Contriever, it matches or exceeds other baselines on Recall@5 and nDCG@5, and trails slightly behind vanilla and Router on Recall@20 and nDCG@20. Similarly, for DPR, TSM outperforms all other methods across all retrieval metrics. This competitive performance on non-temporal datasets can be attributed to TSM's model merging approach, which reduces the magnitude of weight changes during fine-tuning and helps to preserve non-temporal retrieval capabilities while integrating temporal expertise.

Overall, the average results for both Contriever-based and DPR-based TSM show that TSM consistently outperforms other baselines. These results demonstrate that TSM significantly improves temporal retrieval performance without sacrificing effectiveness on non-temporal queries.

6 Analyses

In this section, we systematically examine the effectiveness and underlying mechanisms of our proposed approach.

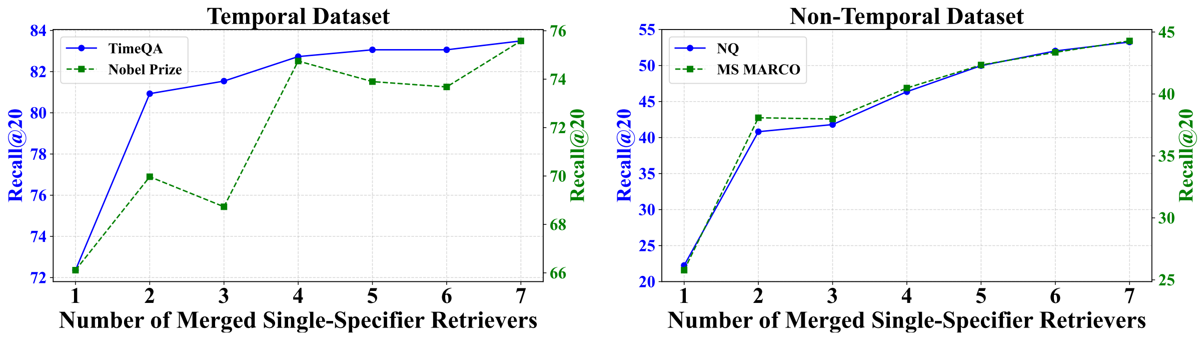

Figure 2: Recall@20 on temporal datasets (TimeQA, Nobel Prize; left) and non-temporal (NQ, MS MARCO; right) datasets as the number of merged single-specifier retrievers increases.

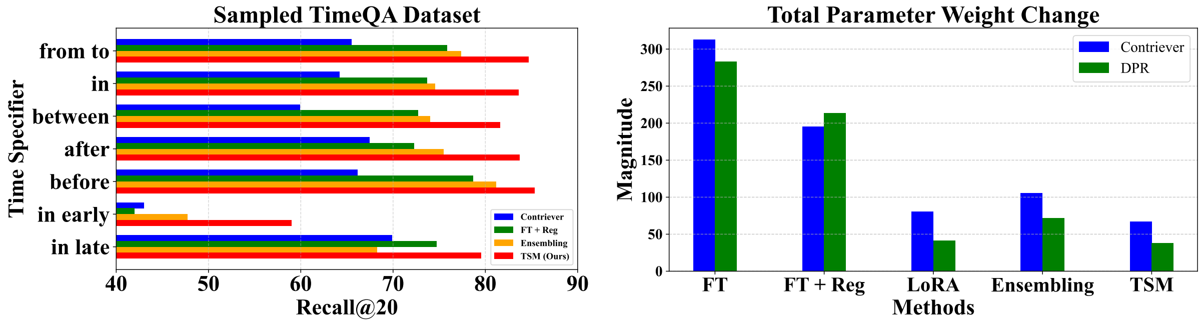

Figure 3: Left: Recall@20 for each time specifier on the TimeQA test set, comparing vanilla Contriever, FT + Reg, Ensembling, and TSM (Ours). Right: Total parameter weight change after fine-tuning for each method, showing how much all network weights are updated. Lower values indicate more stable parameter adaptation.

6.1 Impact of Merging Specifier-Specific Retrievers

Figure 2 shows how retrieval performance changes as the number of merged single-specifier retrievers increases, for temporal (TimeQA and Nobel Prize) and non-temporal (NQ and MS MARCO) datasets. Single-specifier retrievers are merged sequentially in order of data frequency, from most to least frequent, as shown in Table 1.

For the temporal datasets, Recall@20 improves steadily as more single-specifier retrievers are merged. Specifically, for TimeQA (blue line), Recall@20 starts at approximately 72 with a single retriever and rises to about 83 when all seven retrievers are merged (TSM). The Nobel Prize dataset (green line) shows a similar upward trend, increasing from 66 to 76 as more retrievers are merged.

A comparable trend is observed for the non-temporal datasets. For NQ (blue line), Recall@20 increases consistently from about 22 with one retriever to roughly 53 with all seven merged. MS MARCO (green line) also shows a steady improvement, rising from approximately 26 to 45 as the number of merged retrievers increases.

These results demonstrate that merging multiple retrievers, each trained on a specific time specifier, consistently enhances retrieval performance for both temporal and non-temporal queries.

6.2 Coverage Analysis Across Specifiers

Figure 3 (left) compares Contriever, FT + Reg, Ensembling, and TSM on queries grouped by individual time specifiers within the TimeQA test set, reporting Recall@20 for each subset. Across all time specifier categories, TSM achieves the highest recall. For example, on "between [$time_1$] and [$time_2$]" queries, TSM outperforms Contriever, FT + Reg, and Ensembling by a significant margin.

Ensembling, which averages the outputs of retrievers fine-tuned on each time specifier, consistently improves performance over single retrievers for every specifier. However, while Ensembling enhances the overall recall, it does not match the level of specialization achieved by model merging. By merging retrievers individually trained on each time specifier, TSM inherits the strengths of each specialist model and more precisely captures the nuances of temporal constraints. This approach avoids the narrow focus of single-specifier retrievers and achieves a more robust temporal understanding than fine-tuning or ensembling.

In summary, while Ensembling provides notable gains by leveraging the diversity of multiple retrievers, model merging (TSM) delivers superior coverage and specialization across all time specifiers, resulting in the best balance between specialization and generality for temporally constrained queries.

	Contriever	**FT + Reg**	**TSM (Ours)**
Query	Which position did Charles Clarke hold from May 1997 to May 2001?		
Answer	Member of Parliament		
Top-1 Retrieved Passage	Guardian Unlimited Politics – Ask Aristotle: Charles Clarke MP - TheyWorkForYou.com – Charles Clarke MP - BBC News – Charles Clarke profile 17 October 2002 - Interview on Meet The Writers, Monocle 24 with Georgina Godwin - Charles Clarke takes a leading role in promoting animal protection. - Charles Clarke interviewed on Blair, Europe and what Gordon Brown must do next. - The Role of Courts in a Democracy: A Debate Video of Charles Clarke in a Public Debate for the Foundation for Law, Justice and Society, Oxford, 2011	He was a member of the Socialist Campaign Group, Secretary of the All-Party Parliamentary Group for Vietnam, a member of the All-Party Group on Tibet and Chair of the All-Party Parliamentary Group for Cambodia, Member of the Home Affairs Select committee (1992**–97**), and Chairman of the Home Affairs Select Committee from **1997** to 1999 and again from **2001** to 2003.	Charles Rodway Clarke (born 21 September 1950) is a British Labour Party politician, who was the Member of Parliament (MP) for Norwich South **from 1997 until 2010**, and served as Home Secretary from December 2004 until May 2006.
Gold Passage	No	No	**Yes**

Table 3: Case study comparing retrieved passages using Contriever-based methods: vanilla Contriever, FT + Reg, and TSM (Ours). General, non-temporal information is highlighted in blue, temporal information is highlighted in green, and the gold answer that the gold passage should include is highlighted in yellow. Related information, such as correct temporal information, is in **bold**.

6.3 Parameter Weight Change Magnitude

Figure 3 (right) shows the total parameter weight change after fine-tuning for each method. Full fine-tuning (FT) and FT with regularization (FT + Reg) result in the biggest weight changes, indicating extensive updates that improve temporal retrieval but also increase the risk of catastrophic forgetting, leading to significant performance drops on non-temporal queries. By contrast, LoRA and Ensembling exhibit much smaller parameter weight changes, reflecting more stable adaptation and a better balance between temporal and non-temporal retrieval. Notably, TSM achieves the smallest parameter changes for both Contriever and DPR, highlighting its effectiveness at integrating temporal expertise while preserving non-temporal retrieval capabilities. The minimal weight change in TSM underscores its ability to mitigate catastrophic forgetting and maintain robust performance across both temporal and non-temporal queries.

6.4 Case Study: Qualitative Comparison

Table 3 presents a case study from the TimeQA test set: "*Which position did Charles Clarke hold from May 1997 to May 2001?*" Only TSM successfully retrieved the correct gold passage at top-1, while vanilla Contriever and FT + Reg did not. This qualitative analysis examines the types of information each method prioritizes within the retrieved passages. For clarity, information types are color-coded: temporal features (green), non-temporal features (blue), and the gold answer (yellow).

Vanilla Contriever retrieved a passage with non-temporal information about *Charles Clarke* but lacked explicit temporal details matching the required period. This highlights a tendency to focus on non-temporal content, overlooking crucial temporal context. **FT + Reg** retrieved a passage containing relevant temporal markers ("*1997*" and "*2001*") but failed to associate them with *Charles Clarke*'s positions, demonstrating a bias toward temporal information at the expense of non-temporal context. **TSM** retrieved a passage explicitly stating that Charles Clarke was "*Member of Parliament*" from *1997* to *2010*, directly addressing both the temporal and non-temporal requirements of the query and fully covering the specified time frame.

This case illustrates three key insights: (1) dense retrievers often overlook temporal information; (2) naïve fine-tuning can shift attention too far toward temporal cues, missing essential context; and (3) TSM's approach of merging time-specifier-specialized retrievers effectively balances temporal and non-temporal information, mitigating attention bias.

7 Conclusion

This work addresses the challenge of balancing temporal and non-temporal information retrieval by introducing Time-Specifier Model Merging (TSM), a method designed to address attention bias and catastrophic forgetting. TSM trains specialized retrievers for each time specifier and merges them into a unified model. Experiments on both temporal and non-temporal datasets demonstrate that TSM substantially improves performance on temporally constrained queries while maintaining strong performance on non-temporal queries. Our analysis further show that TSM effectively integrates temporal and non-temporal information, mitigating attention bias and outperforming other baselines. These results establish TSM as a robust and efficient solution for diverse information retrieval tasks.

Limitations

While Time-Specifier Model Merging (TSM) demonstrates strong performance in balancing temporal and non-temporal information retrieval, several limitations remain. First, TSM relies on the availability of labeled data for each time specifier; underrepresented or ambiguous temporal expressions may limit the effectiveness of specialized retrievers and the merged model. Second, the current approach focuses on explicit temporal constraints and may not generalize as well to queries with implicit, relative, or underspecified temporal information. Third, our method currently utilizes only seven time specifiers, which may not capture the full range of temporal constraint nuances present in real-world queries. Extending the number and diversity of time specifiers is an important direction for future work to improve coverage and robustness. Fourth, this study merged retrievers solely using simple parameter merging. Alternative approaches leveraging other model merging techniques, such as layer-wise weight averaging (Jang et al., 2024) and spherical linear interpolation (Goddard et al., 2024) can be further explored. Finally, while our experiments cover several benchmark datasets, further evaluation on more diverse domains and real-world temporal retrieval scenarios is needed to fully assess the generalizability and robustness of TSM.

Ethics Statement

This research advances temporal information retrieval by introducing and evaluating the Time-Specifier Model Merging (TSM) method on publicly available benchmark datasets, including TimeQA, Nobel Prize, Natural Questions, and MS MARCO.

We recognize that improved retrieval models, especially those sensitive to temporal constraints, could potentially be misused to surface misleading, outdated, or biased information. To mitigate these risks, we encourage responsible deployment of TSM and recommend incorporating safeguards such as fact-checking and bias detection when applying this technology in real-world systems.

No human subjects, private data, or proprietary information were involved in this research. All model training and evaluation were conducted in accordance with the terms of use of the respective datasets.

Acknowledgements

This work was supported by the Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00010, Development of Korean sign language translation service technology for the deaf in medical environment).

References

Abdelrahman Abdallah, Bhawna Piryani, Jonas Wallat, Avishek Anand, and Adam Jatowt. 2025. Tempretriever: Fusion-based temporal dense passage retrieval for time-sensitive questions. *Preprint*, arXiv:2502.21024.

Anton Alexandrov, Veselin Raychev, Mark Mueller, Ce Zhang, Martin T. Vechev, and Kristina Toutanova. 2024. Mitigating catastrophic forgetting in language transfer via model merging. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 17167–17186. Association for Computational Linguistics.

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843.

Omar Alonso, Jannik Strötgen, Ricardo Baeza-Yates, and Michael Gertz. 2011. Temporal information retrieval: Challenges and opportunities. In *WWW2011 Workshop on Linked Data on the Web, Hyderabad, India, March 29, 2011*, volume 813 of *CEUR Workshop Proceedings*, pages 1–8. CEUR-WS.org.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Jeremy R. Cole, Aditi Chaudhary, Bhuwan Dhingra, and Partha Talukdar. 2023. Salient span masking for temporal understanding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3052–3060, Dubrovnik, Croatia. Association for Computational Linguistics.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Trans. Assoc. Comput. Linguistics*, 10:257–273.

Anoushka Gade and Jorjeta G. Jetcheva. 2024. It's about time: Incorporating temporality in retrieval augmented language models. *CoRR*, abs/2401.13222.

Mudasir Ahmad Ganaie, Minghui Hu, Mohammad Tanveer, and Ponnuthurai N. Suganthan. 2021. Ensemble deep learning: A review. *CoRR*, abs/2104.02395.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's mergekit: A toolkit for merging large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 477–485.

Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. 2014. An empirical investigation of catastrophic forgeting in gradient-based neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Domor Mienye Ibomoiye and Yanxia Sun. 2022. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, 10:99129–99149.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Dong-Hwan Jang, Sangdoo Yun, and Dongyoon Han. 2024. Model stock: All we need is just a few fine-tuned models. *Preprint*, arXiv:2403.19522.

Nattiya Kanhabua and Avishek Anand. 2016. Temporal information retrieval. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 1235–1238, New York, NY, USA. Association for Computing Machinery.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. 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. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. *CoRR*, abs/2105.11644.

Hyunji Lee, Luca Soldaini, Arman Cohan, Minjoon Seo, and Kyle Lo. 2023. Back to basics: A simple recipe for improving out-of-domain retrieval in dense encoders. *CoRR*, abs/2311.09765.

Patrick S. H. 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Mingda Li, Xinyu Li, Yifan Chen, Wenfeng Xuan, and Weinan Zhang. 2024. Unraveling and mitigating retriever inconsistencies in retrieval-augmented large language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 4833–4850. Association for Computational Linguistics.

Zhizhong Li and Derek Hoiem. 2016. Learning without forgetting. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pages 614–629. Springer.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *CoRR*, abs/2308.08747.

Ammar Mohammed and Rania Kora. 2023. A comprehensive review on ensemble deep learning: Opportunities and challenges. *J. King Saud Univ. Comput. Inf. Sci.*, 35(2):757–774.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.

Guy D. Rosin, Ido Guy, and Kira Radinsky. 2021. Time masking for temporal language models. *CoRR*, abs/2110.06366.

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 *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Amal Rannen Triki, Rahaf Aljundi, Matthew B. Blaschko, and Tinne Tuytelaars. 2017. Encoder based lifelong learning. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1329–1337. IEEE Computer Society.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Jiexin Wang, Adam Jatowt, Masatoshi Yoshikawa, and Yi Cai. 2023. Bitimebert: Extending pre-trained language representations with bi-temporal information. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 812–821.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5878–5882, Hong Kong, China. Association for Computational Linguistics.

Feifan Wu, Lingyuan Liu, Wentao He, Ziqi Liu, Zhiqiang Zhang, Haofen Wang, and Meng Wang. 2024. Time-sensitve retrieval-augmented generation for question answering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, pages 2544–2553. ACM.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Xingrun Xing. 2024. Lm-cocktail: Resilient tuning of language models via model merging. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 2474–2488. Association for Computational Linguistics.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *CoRR*, abs/2408.07666.

Appendix

A Additional Experimental Setups

A.1 Model Weights

All model weights used for both the vanilla model and training were obtained from Hugging Face as off-the-shelf checkpoints, without any additional training. Below, we provide the exact Hugging Face model names for the weights used in our experiments:

Contriever:

- facebook/contriever

DPR:

- facebook/dpr-question_encoder-multiset-base
- facebook/dpr-ctx_encoder-multiset-base

A.2 TimeQA Dataset Statistics

Time Specifier	Original		Augmented	
	Train	Dev	Train	Dev
from $[time_1]$ to $[time_2]$	11,676	2,486	-	-
in $[time]$	5,759	1,233	-	-
between $[time_1]$ and $[time_2]$	4,888	1,054	-	-
after $[time]$	903	201	2,741	587
before $[time]$	973	181	2,867	609
in early $[time]$s	309	82	1,885	438
in late $[time]$s	473	91	2,392	474
Total	24,981	5,238	32,208	6,881

Table 4: Statistics for the original and augmented TimeQA datasets illustrate the number of queries containing each time specifier in the training and development sets. To mitigate bias, only the data for the comparatively less frequent time specifiers–after, before, in early, and in late–were augmented.

A.3 Temporal Queries in Non-Temporal Datasets

Dataset	Split	Total Queries	Temporal Queries	Temporal Query (%)
NQ	Test	3,452	53	1.54%
MS MARCO	Dev	509,962	232	0.05%

Table 5: Statistics of *explicit* temporal queries within the test splits of two non-temporal datasets, NQ (Kwiatkowski et al., 2019) and MS MARCO (Nguyen et al., 2016). The table reports the total number of queries, the count of temporal queries, and their proportion in each dataset.

A.4 Implementation Details

For all fine-tuning experiments, each method is trained for five epochs and per-GPU batch size of 64 using on an NVIDIA A100 80GB. We use the publicly available code from Izacard et al. (2022) and follow their hyperparameter settings: a learning rate of 1e-4, the AdamW optimizer (Loshchilov and Hutter, 2017) with a linear learning rate scheduler, and a temperature parameter τ set to 1.0. Model evaluation is performed every 50 steps based on top-1 accuracy, and the best-performing model is selected accordingly. Additionally, five in-batch negative passages are incorporated in the contrastive learning objective.

B Additional Experimental Results

B.1 Coverage Analysis Across Specifiers

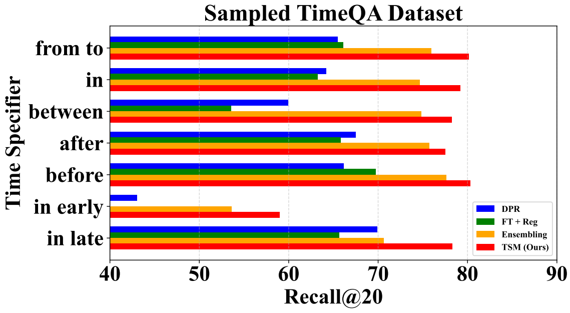

Figure 4: Recall@20 for each time specifier on the TimeQA test set, comparing retrieval performance of vanilla DPR, FT + Reg, Ensembling, and TSM (Ours)

Figure 4 compares DPR, FT + Reg, Ensembling, and TSM on queries grouped by individual time specifiers within the TimeQA test set, reporting Recall@20 for each subset. Across all time specifier categories, TSM achieves the highest recall. For example, on "between $[time_1]$ and $[time_2]$" queries, TSM outperforms DPR, FT + Reg, and ensembling by a significant margin.

Ensembling, which averages the outputs of retrievers fine-tuned on each time specifier, consistently improves performance over single retrievers for every specifier. However, while ensembling enhances the overall recall, it does not match the level of specialization achieved by model merging. By merging retrievers individually trained on each time specifier, TSM inherits the strengths of each specialist model and more precisely captures the nuances of temporal constraints. This approach avoids the narrow focus of single-specifier retrievers and achieves a more robust temporal understanding than simply fine-tuning or ensembling.

In summary, while ensembling provides notable gains by leveraging the diversity of multiple retrievers, model merging (TSM) delivers superior coverage and specialization across all time specifiers, resulting in the best balance between specialization and generality for temporally constrained queries.

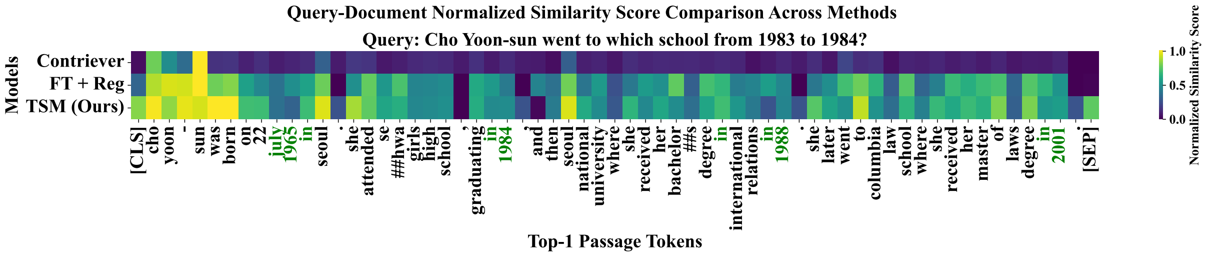

Figure 5: Heatmap of normalized query document similarity scores for the query "*Cho Yoon-sun went to which school from 1983 to 1984?*" comparing vanilla Contriever, FT + Reg, and TSM (Ours). Passage tokens in green represent temporal information.

B.2 Parameter Weight Change Magnitude of Each Single-Specifier Model

Time Specifier	Training Set Size	Weight Change Magnitude
from $[time_1]$ to $[time_2]$	11,676	98.41
in $[time]$	5,759	88.17
between $[time_1]$ and $[time_2]$	4,888	98.87
after $[time]$	2,741	55.60
before $[time]$	2,867	55.98
in early $[time]$s	1,885	72.16
in late $[time]$s	2,392	56.61
Ensembling	-	75.11
TSM (Ours)	-	67.41

Table 6: Parameter weight change magnitude for models fine-tuned on individual time specifiers, compared to Ensembling and TSM. The Ensembling value represents the average weight change magnitude across all single-specifier retrievers. Lower values indicate more stable adaptation.

B.3 Case Study: Query-Document Similarity Score Analysis

Figure 5 shows a heatmap of normalized similarity scores between the TimeQA query "*Cho Yoon-sun went to which school from 1983 to 1984?*" and the same top-1 retrieved passage, comparing Contriever, FT + Reg, and TSM. The x-axis represents the tokenized passage.

Vanilla Contriever mainly highlights non-temporal tokens, such as the person ("*Cho Yoon-sun*") and location ("*Seoul*"), while largely ignoring temporal tokens such as "*1984*." This indicates that without temporal-specific training, Contriever overlooks time constraints and focuses on general keywords. **FT + Reg** increases attention to temporal information, especially the correct year "*1984*," while still attending to non-temporal tokens, though less effectively than TSM. This demonstrates that temporal fine-tuning helps the model better align temporal aspects of queries and passages. **TSM** further sharpens this focus, concentrating on the relevant temporal token "*1984*" and reducing attention to irrelevant years, while also maintaining strong attention to non-temporal features. This indicates a more balanced integration of temporal and non-temporal information.

Overall, these results show that while Contriever neglects temporal cues, FT + Reg improves temporal sensitivity, and TSM achieves the best balance, accurately attending both temporal spans and key non-temporal details. This balanced attention enables TSM to deliver robust retrieval performance for both temporal and non-temporal queries.