TEMPS: Temporal Sentence Embeddings for Temporal Information Retrieval

Anonymous ACL submission

Abstract

Modern information retrieval (IR) systems rarely represent time, yet many information needs depend on it: in clinical, journalistic, and legal search, *when* an event occurred can decide whether a document is relevant. Dense retrievers and Retrieval-Augmented Generation (RAG) pipelines match queries to documents well on topic but poorly on time, so they surface content that is on-topic yet temporally wrong. We introduce **Temporal Textual Similarity (TTS)**, a task that measures how well two anchored sentences align in time, independent of their topical similarity. We then present **TEMPS** (Temporal Embedding Model for Precise Search), a modular temporal branch that attaches to a frozen semantic retriever. TEMPS grounds anchored temporal expressions to intervals and, by moment matching, turns them into directional Gaussian supervision; an anchor-date-conditioned encoder learns to reproduce this signal and is combined with the semantic score at inference. Because the supervision is derived automatically, training requires no manually labeled temporal data. For the temporal score we reuse the Gaussian-KL inclusion machinery from prior distributional embedding work, so the new component is the temporal grounding and moment-matched supervision rather than a new similarity function. On three temporal retrieval benchmarks, TEMPS improves alignment between time-sensitive queries and documents and, on TS-Retriever, lifts R@1 from 19.92 to 25.85 over the prior temporal state of the art.

1 Introduction

Information Retrieval (IR) underpins many Natural Language Processing (NLP) applications, including information extraction, topic detection, and summarization (Alonso et al., 2011). Traditional IR has largely focused on *semantic* relevance, matching documents to queries by topical similarity (Campos et al., 2014a; Rizzo et al., 2022), yet many information needs are also inherently *temporal*: *when* something happened can matter as much as *what* happened. Documents are almost always anchored in time, whether explicitly (e.g., "on January 15th, 2008") or implicitly (e.g., "last summer"), and interpreting these contexts is essential for results that are both topically and temporally relevant (Campos et al., 2014a; Derczynski et al., 2015; Han et al., 2025).

Failing to account for temporal constraints can lead to factually correct but contextually misleading answers. Figure 1 illustrates this: given the query "Who was the Director-General of the World Health Organization between 2006 and 2017?", two passages appear semantically relevant, yet only one aligns with the correct time frame. Without temporal reasoning, IR systems risk returning outdated, premature, or otherwise temporally mismatched information, undermining trust in downstream applications such as search, question answering, and decision support.

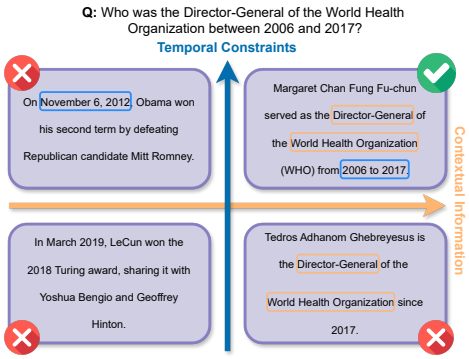

Figure 1: Importance of temporal constraints in IR

This need spans many domains: clinical narratives, where diagnoses and treatments unfold over time (Moharasar and Ho, 2016); journalism, where ignoring a publication date recirculates outdated stories as misinformation (Musi and Rocci, 2022;

Yoon et al., 2023); and legal contexts, where a document's validity depends on its date of enactment. Even consumer search for reviews or events relies on temporal alignment to stay relevant.

Historically, retrieval models such as Term Frequency–Inverse Document Frequency (TF-IDF) (Salton and Buckley, 1988) and BM25 (Robertson et al., 2009) operated solely on lexical overlap, with no mechanism for temporal reasoning (Qian et al., 2024). Dense retrieval models (Karpukhin et al., 2020; Khattab and Zaharia, 2020; Gao et al., 2021), widely used for Retrieval-Augmented Generation (RAG) (Lewis et al., 2020), improved semantic matching by encoding queries and documents into high-dimensional vector spaces with Large Language Models (LLMs), then passing the retrieved documents to a generator to synthesize an answer.

However, the encoders behind dense retrieval, such as BERT (Devlin, 2018), ModernBERT (Warner et al., 2024), and ELECTRA (Clark et al., 2020), struggle with temporal reasoning (Chen et al., 2021; Song et al., 2025). This stems from two factors. First, large-scale pretraining corpora (Rajpurkar et al., 2018; Kwiatkowski et al., 2019) underrepresent explicit temporal expressions and rely on coarse-grained timestamps, leaving models ill-equipped for questions involving sequence, overlap, and duration (Wang et al., 2023). Second, the embedding spaces of dense retrievers are static snapshots: they encode knowledge available at training time but cannot adapt to evolving timelines (Sojitra et al., 2024) or reason about time-dependent constraints. The result is *temporally brittle* retrieval that matches on semantics while missing time-sensitive relationships, returning temporally incorrect documents.

Temporal cues compound this problem, since they are often implicit or ambiguous (Nye et al., 2021). Resolving "earlier this year" requires the document's publication date, while "five years from now" requires arithmetic over an anchor; without dedicated modeling, such cases yield irrelevant results.

A central difficulty is that temporal compatibility is often *asymmetric*. A passage about "in 2020" is compatible with a query about a specific date that year, but a passage about a single day should not be equally compatible with a query about the whole year. Symmetric functions such as cosine similarity or interval overlap cannot express this directionality, scoring "day contained in year" and "year contains day" identically. This motivates modeling temporal expressions as distributions, where the mean captures location, the variance captures granularity or uncertainty, and a directional divergence distinguishes fine-to-coarse compatibility from its reverse.

We address these limitations with a framework that models temporal information explicitly and folds it into dense retrieval without retraining the underlying retriever. The core idea is to ground anchored temporal language to intervals and derive supervision directly from those intervals, then train an anchor-date-conditioned module to reproduce it and fuse with any semantic retriever. To represent and compare intervals we adopt the Gaussian-KL inclusion machinery from prior distributional embedding work (Vilnis and McCallum, 2014; He et al., 2015; Yoda et al., 2023), whose directional divergence already captures the fine-to-coarse asymmetry that temporal inclusion needs; what is new is the temporal grounding and the moment-matched interval-to-Gaussian supervision, not the divergence.

Our contributions are threefold:

- We formulate **Temporal Textual Similarity (TTS)**, a task that scores *when* two anchored sentences are temporally compatible, independent of topical similarity, giving STS a temporal counterpart that existing similarity resources do not capture.
- We show this compatibility can be **supervised at scale without manual labeling**: a pipeline grounds temporal language to intervals and moment-matches them into directional Gaussian targets, yielding 60M training tuples.
- We instantiate the signal as **TEMPS**, an anchor-date-conditioned temporal module that attaches to *frozen* semantic retrievers and adds fine-grained temporal reasoning at roughly 27M extra parameters. It repurposes established Gaussian-KL inclusion machinery for temporally grounded text rather than introducing a new divergence.

We call our approach **TEMPS** (Temporal Embedding Model for Precise Search).

2 Preliminaries

Temporal Information & Expressions Temporal information facilitates time-aware retrieval

through computable relations (Allen, 1983), hierarchical granularity, and normalizable expressions (Pustejovsky et al., 2003). These expressions are categorized by their grounding (Derczynski et al., 2012): *absolute* (explicit timestamps like "January 15th"), *deictic* (speaker-relative, e.g., "yesterday"), *anaphoric* (relative to a prior reference), and *duration* (defined spans).

TimeML Specification TimeML (Pustejovsky et al., 2003, 2005, 2010) is the standard markup for annotating these expressions in natural language (Jia et al., 2018), forming the foundation for many temporal taggers (Llorens et al., 2010; Strötgen and Gertz, 2010; Lee et al., 2014). Instead of a standalone table, we outline our synthetic dataset's TimeML annotations here: Dates (e.g., `2024-09` for September 2024), Offsets (`THIS P1D` for the present day), References (`PRESENT_REF` for now), and Intervals (`2023-SU, 2024-FA` for summer 2023 to fall 2024).

3 Related Work

Temporal Information Retrieval. Temporal Information Retrieval (TIR) incorporates time into search by exploiting document timestamps, temporal expressions, and explicit or implicit temporal intent in queries (Campos et al., 2014b; Kanhabua and Anand, 2016). This allows systems to rank documents by both topical and temporal relevance, which is essential for dynamic collections where the same entity or event may have different correct interpretations at different times.

Semantic Textual Similarity. Semantic Textual Similarity (STS) measures the degree of semantic equivalence between sentence pairs and supports tasks such as machine translation, summarization, QA, and semantic search (Agirre et al., 2012, 2016; Cer et al., 2017). Benchmarks such as STS-B and SICK provide human similarity or relatedness annotations (Cer et al., 2017; Marelli et al., 2014), but they primarily target semantic relatedness rather than temporal alignment. As a result, high STS scores do not necessarily imply that two sentences express compatible temporal information.

Time-Sensitive Question Answering. Time-sensitive QA requires models to answer under temporal constraints, e.g., distinguishing facts that hold before or after a given date. Benchmarks such as TimeQA (Chen et al., 2021) test whether models can identify, normalize, and reason over temporal expressions in order to retrieve or generate answers valid for the specified time context.

Retrieval-Augmented Generation. RAG improves factuality in knowledge-intensive tasks by retrieving external evidence before generation (Lewis et al., 2020; Kandpal et al., 2023). Modern systems commonly rely on sparse retrievers such as BM25 or dense embedding models, including instruction-tuned retrievers such as AngIE and BGE (Li and Li, 2023; Chen et al., 2024). Efficiency-oriented RAG methods such as COCOM and FlexRAG compress retrieved context to reduce long-context costs (Rau et al., 2025; Wu et al., 2025). However, most RAG retrievers optimize semantic relevance and do not explicitly model temporal compatibility between queries and evidence.

Gaussian Embeddings and Directional Similarity. Gaussian embeddings represent items as distributions rather than points, allowing the mean to encode location and the covariance to encode uncertainty or generality. Vilnis and McCallum (Vilnis and McCallum, 2014) use KL divergence between Gaussian word embeddings to model asymmetric inclusion in lexical semantics; KG2E (He et al., 2015) applies Gaussian embeddings and asymmetric KL to directional knowledge-graph relations; and GaussCSE (Yoda et al., 2023) uses a bounded inverse-KL score for entailment-oriented sentence similarity. TEMPS follows this lineage rather than introducing a new Gaussian divergence or similarity head. Its novelty is temporal: grounded intervals are converted into Gaussian supervision by moment matching, and an anchor-date-conditioned branch learns to reproduce this temporal compatibility signal for retrieval.

Temporal Retrieval for RAG. TS-Retriever and TSContriever (Wu et al., 2024) directly address time-sensitive retrieval by constructing semantically similar but temporally distinct documents and training Contriever (Izacard et al., 2021) to separate them. This line of work shows the value of temporal supervision, but existing approaches remain tied to specific retriever training setups. In contrast, our work introduces TTS as a temporal similarity objective and TEMPS as a modular temporal component that can be combined with different semantic retrievers.

Gap. Despite progress in STS, TIR, and RAG, current embedding models remain limited on tem-

porally rich text. Standard similarity functions such as cosine similarity are symmetric and cannot naturally represent asymmetric relations such as a specific date being contained in a broader interval. Moreover, existing STS resources do not teach models to ignore topical similarity when temporal information conflicts. These gaps motivate temporally aware representations and datasets that explicitly model temporal alignment.

4 Problem Definition

We propose the **Temporal Textual Similarity (TTS)** task to improve time-aware **IR** models. Designed as a temporal analogue of the STS task, TTS trains embedding models to measure the alignment of temporal information expressed in sentences.

However, unlike STS, which assesses semantic or topical relatedness, **TTS disregards semantic similarity and focuses solely on the temporal dimension**.

Formally:

> *Given a quadruple (s_1, d_1, s_2, d_2), where d_1 and d_2 are anchor dates for interpreting temporal expressions in s_1 and s_2, estimate the degree to which the sentences express similar temporal information, independent of semantic or topical similarity.*

This task is intended as a basis for developing retrieval systems sensitive to temporal context.

5 Methodology

TEMPS learns a temporal signal that complements semantic similarity. The method has two parts: (i) weakly supervised temporal labels for pairs of anchored expressions, and (ii) a lightweight temporal embedding module that can be combined with arbitrary dense retrievers. Both the labels and the inference-time score use the same Gaussian-KL form, so training reduces to teaching the module to reproduce a closed-form temporal target. Detailed dataset construction and model/training specifications are provided in Appendices A and B.

5.1 Temporal Supervision

Each training instance is a tuple (s_1, d_1, s_2, d_2, ts), where s_i is a text containing a temporal expression, d_i is its anchor date at day granularity, and ts is a temporal similarity value. We generate these tuples from synthetic TimeML-style expressions spanning the years 1000-2030 and the English portion of the Multilingual MLM Temporal Tagging Resources dataset (Lange et al., 2022), yielding roughly 60M tuples in total.

Temporal labels are computed by grounding expressions to intervals and applying a directional temporal similarity function. Given intervals $I_1 = [a_1, b_1]$ and $I_2 = [a_2, b_2]$ for the temporal expressions in s_1 and s_2, respectively, we first represent each interval as a Gaussian by moment matching the uniform distribution over its inclusive day range:

$$\begin{aligned} p_I &= \mathcal{N}(\mu_I, \sigma_I^2), \\ \mu_I &= \frac{a+b+1}{2}, \\ \sigma_I^2 &= \frac{(b-a+1)^2}{12}. \end{aligned} \tag{1}$$

The mean captures temporal location, and the variance captures the granularity or uncertainty of the expression: a single day has low variance, whereas a month or year has higher variance. The distributional view matters because temporal relevance is not symmetric. A specific date can be judged compatible with a broader containing interval, while the reverse direction should be penalized.

We therefore define the temporal similarity function as an asymmetric KL divergence from the first sentence interval to the second:

$$\begin{aligned} \mathrm{TSF}(I_1 \to I_2) &= D_{\mathrm{KL}}(p_{I_1} \| p_{I_2}) \\ &= \frac{1}{2} \log \frac{\sigma_2^2}{\sigma_1^2} \\ &\quad + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}. \end{aligned} \tag{2}$$

Smaller values indicate stronger temporal compatibility in the $s_1 \to s_2$ direction. For training we use the bounded value $ts = 1/(1 + \mathrm{TSF}(I_1 \to I_2))$, matching the inverse-KL form used in prior Gaussian sentence embeddings (Yoda et al., 2023). This gives smooth, directional supervision and keeps the training labels in the same form as the Gaussian embedding head used at inference. Additional derivation and examples are given in Appendix A.1.1.

5.2 Temporal Model

Architecture. Figure 2 shows TEMPS, which consists of a semantic branch and a temporal branch. The semantic branch supplies the standard dense representation used for topical relevance. The temporal branch receives a sentence and an

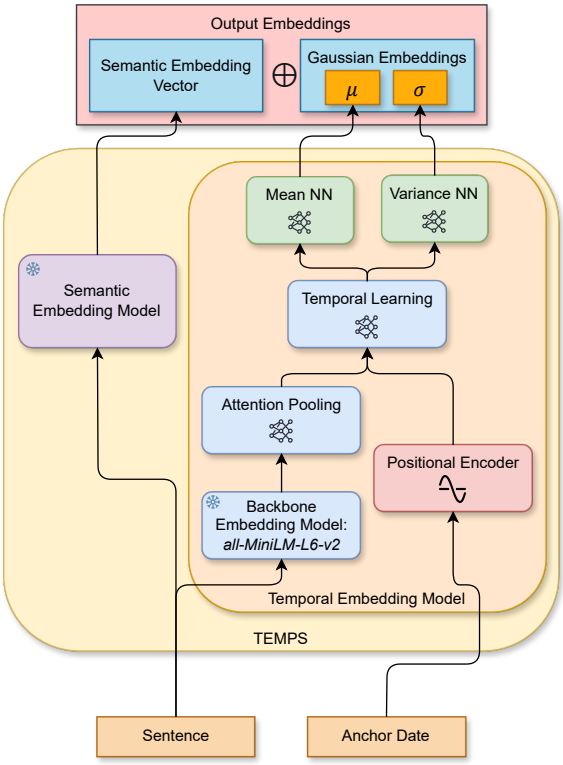

Figure 2: TEMPS' Model Architecture

anchor date, encodes the sentence with a frozen all-MiniLM-L6-v2 backbone, encodes the date with a sinusoidal positional representation, and maps the combined representation to a Gaussian embedding. Following prior Gaussian embedding models (Vilnis and McCallum, 2014; Yoda et al., 2023), the mean captures temporal location, while the diagonal variance captures temporal granularity and uncertainty.

Temporal Similarity. For an anchored text (s, d), TEMPS outputs a semantic vector E_S and temporal Gaussian parameters (μ, σ^2). Given two inputs, we combine semantic cosine similarity with an asymmetric temporal score $s_{1,2}$:

$$\begin{aligned} s_{1,2} &= (1-\alpha)\,\cos(E_{S,1}, E_{S,2}) + \alpha\,\tau_{1,2}, \\ \tau_{1,2} &= \frac{1}{1 + D_{\mathrm{KL}}(p_1 \| p_2)}, \\ p_j &= \mathcal{N}(\mu_j, \mathrm{diag}(\sigma_j^2)), \quad j \in \{1, 2\}. \end{aligned} \tag{3}$$

Here, $\alpha \in [0, 1]$ controls the temporal contribution. Because D_{KL} is directional, the score can distinguish relations such as a date being contained in a broader interval from the reverse direction.

Training and Integration. The temporal branch is trained on TTS supervision with a CoSENT-style ranking loss, while the semantic branch can be replaced by any external embedding model. This modular design lets TEMPS add temporal awareness to existing retrievers without retraining their semantic encoders. Architectural, loss, and hyperparameter details are provided in Appendix B.

6 Experiments

6.1 Experimental Setup

We evaluate TEMPS on three temporal retrieval benchmarks: **TimeQA** (Chen et al., 2021), which tests evolving facts over long documents; **TempReason** (Tan et al., 2023), which evaluates date arithmetic and event-time grounding; and **TS-Retriever** (Wu et al., 2024), which stresses retrieval among semantically similar but temporally mismatched documents. We compare against lexical, rule-based, dense, LLM-based, and temporal retrieval baselines, including BM25, SUTime (Chang and Manning, 2012), MPNet$_{\text{base}}$-v2 (Song et al., 2020), E5$_{\text{base}}$-v2 (Wang et al., 2022), BGE$_{\text{large}}$-v1.5 (Xiao et al., 2024), Mistral, and TSContriever (Wu et al., 2024). For each dense retriever, we also report the corresponding +TEMPS variant where applicable. To test whether temporal retrieval gains transfer to an end task, we additionally evaluate a TimeQA RAG pipeline that retrieves the top-5 passages and uses Qwen2.5-7B-Instruct (Yang et al., 2024) as the generator.

For retrieval, we report MRR, NDCG@5, Recall@5, and Precision@5. For downstream RAG, we report only answer-level metrics: exact match (EM), token-level F1, and answer containment. The temporal-semantic interpolation parameter α is treated as a benchmark-level calibration parameter, not a trained model parameter. Because the benchmarks intentionally emphasize different mixtures of topical matching and temporal discrimination, we sweep a common set of candidate values and report the best value for each benchmark. The same α is then used for all +TEMPS variants within that benchmark, so individual semantic backbones are not tuned separately. This protocol is intended to characterize the appropriate temporal-semantic trade-off for each evaluation setting; in deployment, α should be chosen on held-out validation data. Additional benchmark statistics, baseline descriptions, and computational details are provided in Appendix C.

6.2 Results

To explore our results, we structure our findings into several research questions.

RQ1: Does temporal modeling help with temporal IR? Adding TEMPS consistently improves retrieval performance across datasets and base models. The results, obtained with the temporal similarity function defined in Section 5.2, show that the temporal component improves both lightweight and large semantic encoders rather than benefiting only a single backbone.

On **TimeQA**, every model augmented with TEMPS shows a large MRR improvement. *Mistral + TEMPS* achieves the highest MRR, while $E5_{base}$-v2 + TEMPS obtains the best NDCG@5, Recall@5, and Precision@5. TimeQA probes sensitivity to *temporal changes in factual knowledge*, embedding explicit time references (e.g., "in 2007", "before 2010") whose modification changes the correct answer. The gains indicate that TEMPS enables models to go beyond static retrieval or memorization and to make finer temporal distinctions among otherwise relevant passages.

On **TempReason**, which combines pure temporal reasoning and event temporal grounding, we again observe gains across all semantic models. *Mistral + TEMPS* achieves the highest MRR and NDCG@5, while $E5_{base}$-v2 + TEMPS obtains the highest Recall@5 and Precision@5. The temporal-only variant is also strong, suggesting that the task is especially sensitive to temporal structure. This benchmark requires both *abstract logical operations* (such as temporal arithmetic) and the ability to ground events to valid time ranges. The results therefore show that TEMPS captures generalizable temporal logic, with the semantic model mainly refining the final ranking among temporally plausible candidates.

On **TS-Retriever**, improvements remain consistent even for high-performing baselines. *Mistral + TEMPS* reaches the highest MRR in Table 1, while $E5_{base}$-v2 + TEMPS achieves the best NDCG@5, Recall@5, and Precision@5. TEMPS improves both early precision and broad temporal evidence coverage.

Table 2 compares $E5_{base}$-v2 + TEMPS with the state-of-the-art *TSContriever* models. This configuration surpasses *TSContriever* not only at deeper ranks but also at rank 1, improving R@1 from 19.92 to 25.85 and P@1 from 58.63 to 70.50. It also maintains stronger recall across the ranking window, reaching R@100 of 99.23 compared to 94.57 for *TSContriever*, while improving deeper precision such as P@20 (17.91 vs. 15.72).

To address the fact that *TSContriever* is purely temporal and trained on TS-Retriever, Wu et al. (Wu et al., 2024) proposed a hybrid query-router architecture that sends time-sensitive queries to *TSContriever* and other queries to the standard semantic *Contriever*. Our unified approach outperforms this composite system on every metric, suggesting that TEMPS can combine temporal sensitivity with semantic matching without requiring a separate routing mechanism.

These results demonstrate that TEMPS addresses subtle weaknesses in handling *semantically similar but temporally mismatched distractors*.

RQ2: Do temporal retrieval gains improve downstream RAG? Table 3 evaluates whether the retrieval improvements translate into end-to-end answer quality. Using the same TimeQA queries and Qwen2.5-7B-Instruct generator, every +TEMPS retriever improves EM, F1, and containment over its unaugmented counterpart. The strongest overall RAG performance comes from $E5_{base}$-v2 + TEMPS, with similar gains across the other retrievers.

These RAG results provide a downstream validation of the TTS objective: better temporal alignment in retrieval leads to better generated answers, not only better ranking scores. They also address the practical concern that improvements at deeper retrieval ranks may not affect top-sensitive pipelines. Although the generator sees only the top-5 retrieved passages, temporalized retrievers consistently produce more correct final answers, indicating that TEMPS improves the evidence available to the generator in the part of the ranking that directly matters for RAG.

RQ3: Is it better to increase the size of the model or to better model time in temporal IR? Our results show that correctly modeling temporal information yields larger gains than simply scaling model size. Across all three benchmarks, $E5_{base}$-v2 + TEMPS outperforms the unaugmented 7B-parameter *Mistral* model despite having around 50 times fewer parameters. Scaling remains useful when the temporal module is also added, as *Mistral + TEMPS* obtains the best MRR on all three datasets. However, the unaugmented 7B model is consistently weaker than much smaller temporalized encoders. These results highlight that tempo-

Table 1: Performance of Models on Different Datasets (All improvements are statistically significant with p-value < 0.05; detailed p-values are reported in Appendix C.6)

Benchmark	Model	MRR	NDCG@5	Recall@5	Precision@5	MRR Gain	# params
TimeQA ($\alpha = 0.6$)	BM25	0.2398	0.2472	0.4059	0.0812	-	-
	SUTime	0.4704	0.4966	0.6471	0.1294	-	-
	TEMPS w/o Semantic Model	0.4655	0.4938	0.6528	0.1306	-	27M
	$MPNet_{base}$-v2	0.2985	0.3087	0.4746	0.0949	-	109M
	$MPNet_{base}$-v2 + TEMPS	0.5071	0.5360	0.6967	0.1393	+69.88%	136M
	$E5_{base}$-v2	0.3283	0.3664	0.5998	0.1200	-	109M
	$E5_{base}$-v2 + TEMPS	0.5398	**0.5726**	**0.7317**	**0.1463**	+64.42%	136M
	BGE_{large}-v1.5	0.2942	0.3238	0.5412	0.1082	-	335M
	BGE_{large}-v1.5 + TEMPS	0.5346	0.5632	0.7159	0.1432	+81.71%	362M
	Mistral	0.4085	0.4478	0.6561	0.1312	-	7111M
	Mistral + TEMPS	**0.5427**	0.5710	0.7215	0.1443	+32.83%	7138M
TempReason ($\alpha = 0.7$)	BM25	0.3778	0.3512	0.5100	0.1143	-	-
	SUTime	0.3188	0.2781	0.4296	0.0976	-	-
	TEMPS w/o Semantic Model	0.9081	0.9276	0.9926	0.2167	-	27M
	$MPNet_{base}$-v2	0.7269	0.7700	0.9356	0.2010	-	109M
	$MPNet_{base}$-v2 + TEMPS	0.9026	0.9229	0.9917	0.2162	+24.17%	136M
	$E5_{base}$-v2	0.8174	0.8565	0.9846	0.2147	-	109M
	$E5_{base}$-v2 + TEMPS	0.9161	0.9342	**0.9936**	**0.2169**	+12.07%	136M
	BGE_{large}-v1.5	0.7972	0.8377	0.9768	0.2123	-	335M
	BGE_{large}-v1.5 + TEMPS	0.9170	0.9345	0.9931	0.2167	+15.03%	362M
	Mistral	0.8161	0.8518	0.9769	0.2120	-	7111M
	Mistral + TEMPS	**0.9183**	**0.9353**	0.9925	0.2166	+12.53%	7138M
TS-Retriever ($\alpha = 0.2$)	BM25	0.0359	0.0100	0.0133	0.0090	-	-
	SUTime	0.3109	0.1915	0.2347	0.1329	-	-
	TEMPS w/o Semantic Model	0.1900	0.0974	0.1098	0.0707	-	27M
	$MPNet_{base}$-v2	0.3849	0.2381	0.2378	0.1632	-	109M
	$MPNet_{base}$-v2 + TEMPS	0.5092	0.3437	0.3328	0.2357	+32.30%	136M
	$E5_{base}$-v2	0.7840	0.6405	0.6110	0.4221	-	109M
	$E5_{base}$-v2 + TEMPS	0.8052	**0.6820**	**0.6532**	**0.4527**	+2.70%	136M
	BGE_{large}-v1.5	0.7136	0.5450	0.5157	0.3598	-	335M
	BGE_{large}-v1.5 + TEMPS	0.7896	0.6362	0.5987	0.4166	+10.65%	362M
	Mistral	0.7505	0.6103	0.5856	0.4058	-	7111M
	Mistral + TEMPS	**0.8114**	0.6816	0.6448	0.4482	+8.11%	7138M

ral modeling can outweigh raw capacity increases, especially on tasks with strong temporal dependencies, which is important for resource-constrained retrieval systems.

RQ4: What is the importance of time in the model's scoring function? An α sweep (Appendix C.3) confirms performance peaks at non-zero values across all datasets, proving temporal information is critical. We tune α per benchmark to balance semantic relevance and temporal compatibility, as a single global value would underperform on diverse tasks. For Table 1, we use: TimeQA ($\alpha = 0.6$), TempReason ($\alpha = 0.7$), and TS-Retriever ($\alpha = 0.2$). TimeQA and TempReason benefit from higher temporal weights due to their focus on evolving facts and date arithmetic. Conversely, TS-Retriever requires a lower weight to prevent time from suppressing the semantic signals needed to distinguish topically close distractors.

RQ5: Generalization of Temporal Modeling Across Reasoning Types Our method demonstrates consistent improvements across distinct reasoning scenarios: (i) **Fact retrieval over evolving knowledge** (*TimeQA*), (ii) **Distractor filtering with temporal mismatch** (*TS-Retriever*), (iii) **Abstract date arithmetic and Event-Time grounding** (*TempReason*).

The performance gains of TEMPS across these diverse tasks suggest that it captures both surface-level temporal cues and deeper reasoning patterns. Specifically, it can accurately align documents with explicit temporal constraints, infer ambiguous temporal references relative to anchor dates, and perform purely symbolic date operations without relying on topical context. This ability to generalize indicates that the learned temporal representations are not tied to a single task but encode transferable reasoning skills applicable across multiple temporal reasoning challenges.

Table 2: Comparison of baselines on TS-Retriever (* = metrics from (Wu et al., 2024))

Model	Recall						Precision					
	R@1	R@5	R@10	R@20	R@50	R@100	P@1	P@5	P@10	P@20	P@50	P@100
Contriever(Izacard et al., 2021)*	4.72	16.89	28.14	43.78	68.40	85.17	17.69	13.14	11.12	8.90	5.50	3.43
TAS-B(Hofstätter et al., 2021)*	2.62	10.17	18.40	31.02	55.18	75.56	9.49	7.27	6.56	5.56	4.03	2.85
JinaAI(Günther et al., 2023)*	12.77	35.58	50.10	65.97	84.15	94.46	38.47	24.68	18.19	12.38	6.57	3.77
OpenAI*	9.50	29.56	40.93	55.33	77.53	92.84	29.93	20.38	14.75	10.31	6.01	3.68
TSContriever*	19.92	52.70	69.13	81.52	91.53	94.57	58.63	36.49	25.34	15.72	7.30	3.81
TSContriever*$_{\text{only-q}}$	11.62	38.96	56.38	74.44	89.88	94.82	41.27	29.15	21.60	14.54	7.18	3.81
+ query-router*	10.94	36.71	52.95	71.01	88.94	95.11	39.49	27.67	20.40	14.02	7.15	3.83
$E5_{base} - v2$ + TEMPS ($\alpha = 0.2$)	**25.85**	**65.32**	**81.08**	**91.21**	**97.42**	**99.23**	**70.50**	**45.27**	**30.49**	**17.91**	**7.80**	**3.99**

Table 3: Downstream RAG performance on TimeQA using Qwen2.5-7B-Instruct as the generator.

Retriever	α	EM	F1	Containment
TEMPS w/o Semantic Model	–	0.3641	0.4394	0.4081
$E5_{base}$-v2	–	0.2830	0.3563	0.3326
$E5_{base}$-v2 + TEMPS	0.60	**0.3878**	**0.4742**	**0.4498**
BGE_{large}-v1.5	–	0.2627	0.3258	0.3112
BGE_{large}-v1.5 + TEMPS	0.60	0.3743	0.4602	0.4397
$MPNet_{base}$-v2	–	0.2187	0.2729	0.2604
$MPNet_{base}$-v2 + TEMPS	0.60	0.3687	0.4471	0.4205
Mistral	–	0.3112	0.3844	0.3709
Mistral + TEMPS	0.60	0.3811	0.4668	0.4453

RQ6: Can temporal modeling be integrated without huge computational cost? TEMPS adds only a lightweight temporal embedding module to an existing model, increasing parameter count by approximately 27M in our experiments. Even when applied to small embedding models like $E5_{base}$-v2, it delivers performance competitive with, and often exceeding, the unaugmented 7B-parameter Mistral model. Furthermore, TEMPS can be attached to different semantic embedding models without retraining the semantic component, making it a modular addition to existing retrieval pipelines. This makes the approach suitable for real-world systems where both latency and memory footprint are critical constraints.

7 Conclusion

While large language models and dense retrieval methods have advanced the field of information retrieval, they often fall short in addressing temporal reasoning, an essential factor for accurately interpreting time-sensitive user queries. To tackle this limitation, we proposed a novel framework called **TEMPS** based on the Temporal Textual Similarity task, leveraging weak supervision and synthetic data to train a dedicated embedding model that captures temporal information in documents. Our evaluations showed improvements over existing baselines, with temporal embeddings emerging as the key contributor to this success. These results highlight the critical role of temporal modeling in retrieval systems, paving the way for more accurate performance in dynamic domains and setting the stage for further research in temporal retrieval.

Future Work While our explicit temporal modeling proves effective, several promising directions remain open. First, tighter integration of semantic and temporal embeddings, via joint training or attention-based fusion, could improve robustness against semantically similar yet temporally incongruent distractors. Second, extending TEMPS to *implicit* temporal references and event-based reasoning would broaden its applicability to real-world narrative and conversational domains. Third, scaling data generation to diverse large-scale open corpora could enhance coverage of rare temporal expressions. Finally, we aim to explore multilingual retrieval and lightweight modules for low-latency, resource-constrained environments.

Limitations

Our evaluation focuses on benchmarks with explicit temporal annotations and reliable timestamps, enabling controlled comparison but not covering all real-world variability in temporal language or metadata quality. All experiments are conducted in English; we have not evaluated TEMPS on other languages, and its effectiveness in multilingual or cross-lingual settings remains an open question. Our temporal supervision is built from synthetic templates and a rule-based, HeidelTime-tagged resource, and it grounds anchored, resolvable temporal expressions; implicit or highly ambiguous references and event–event relations (e.g., TempReason Level 3) are outside our current scope, and the weak supervision inherits any noise from the underlying tagger. The temporal–semantic weight α is selected

per benchmark from a sweep over the same evaluation sets rather than a separate held-out split; in deployment it should be calibrated on held-out validation data. Reported results come from a single training run (seed 42), with statistical significance measured across queries rather than across training seeds. TEMPS is also model-agnostic and thus inherits the biases and factual gaps of the semantic backbone it augments, a trade-off we accept to ensure modularity and ease of integration. Finally, while the added temporal encoding introduces a small computational overhead, it remains far lower than the cost of scaling to much larger models for comparable gains.

Ethical Considerations

TEMPS is foundational retrieval research and is not tied to a specific deployment. Because it is model-agnostic, it inherits any biases or factual gaps of the semantic backbone it augments; in time-sensitive, high-stakes domains such as clinical or legal search, temporally miscalibrated retrieval could surface misleading evidence, and downstream applications should therefore validate temporal alignment before relying on it. We use only synthetic and publicly available research data and release our artifacts for research use only.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

Eneko Agirre, Johan Bos, Mona Diab, Suresh Manandhar, Yuval Marton, and Deniz Yuret. 2012. * sem 2012: The first joint conference on lexical and computational semantics–volume 1: Proceedings of the main conference and the shared task, and volume 2: Proceedings of the sixth international workshop on semantic evaluation (semeval 2012). In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*.

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

Omar Alonso, Jannik Strötgen, Ricardo Baeza-Yates, and Michael Gertz. 2011. Temporal information retrieval: Challenges and opportunities. *Twaw*, 11:1–8.

Klaus Berberich, Srikanta Bedathur, Omar Alonso, and Gerhard Weikum. 2010. A language modeling approach for temporal information needs. In *Proceedings of the European Conference on Information Retrieval (ECIR)*.

Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *International Conference on Learning Representations (ICLR)*.

Ricardo Campos, Gaël Dias, Alípio M. Jorge, and Adam Jatowt. 2014a. Survey of temporal information retrieval and related applications. *ACM Comput. Surv.*, 47(2).

Ricardo Campos, Gaël Dias, Alípio M Jorge, and Adam Jatowt. 2014b. Survey of temporal information retrieval and related applications. *ACM Computing Surveys (CSUR)*, 47(2):1–41.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Angel X Chang and Christopher D Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In *Lrec*, volume 12, pages 3735–3740.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. *arXiv preprint arXiv:2108.06314*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Leon Derczynski, Héctor Llorens, and Estela Saquete. 2012. Massively increasing timex3 resources: a transduction approach. *arXiv preprint arXiv:1203.5076*.

Leon Derczynski, Jannik Strotgen, Ricardo Campos, and Omar Alonso. 2015. Time and information retrieval: Introduction to the special issue. *Inf. Process. Manag.*, 51:786–790.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, and 1 others. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *arXiv preprint arXiv:2310.19923*.

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, Vienna, Austria. Association for Computational Linguistics.

Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, pages 623–632.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 113–122.

Xiang Huang, Hao Peng, Dongcheng Zou, Zhiwei Liu, Jianxin Li, Kay Liu, Jia Wu, Jianlin Su, and Philip S Yu. 2024. Cosent: consistent sentence embedding via similarity ranking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2800–2813.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. Tempquestions: A benchmark for temporal question answering. In *Companion Proceedings of the The Web Conference 2018*, pages 1057–1062.

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.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.

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*, pages 1235–1238.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Lukas Lange, Jannik Strötgen, Heike Adel, and Dietrich Klakow. 2022. Multilingual normalization of temporal expressions with masked language models. *arXiv preprint arXiv:2205.10399*.

Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. 2014. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*.

Hector Llorens, Estela Saquete, and Borja Navarro. 2010. Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 284–291.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. The sick (sentences involving compositional knowledge) dataset for relatedness and entailment. *URL: https://doi. org/10.5281/zenodo*, 2787612.

Gandhimathi Moharasar and Tu Bao Ho. 2016. A semi-supervised approach for temporal information extraction from clinical text. In *2016 IEEE RIVF international conference on computing & communication technologies, research, innovation, and vision for the future (RIVF)*, pages 7–12. IEEE.

Elene Musi and Andrea Rocci. 2022. Staying up to date with fact and reason checking: An argumentative analysis of outdated news. *The Pandemic of Argumentation. Cham: Springer*, pages 311–330.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, and 1 others. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

James Pustejovsky, Robert Knippen, Jessica Littman, and Roser Saurí. 2005. Temporal and event information in natural language text. *Language resources and evaluation*, 39:123–164.

James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. Iso-timeml: An international standard for semantic annotation. In *LREC*, volume 10, pages 394–397.

Xinying Qian, Ying Zhang, Yu Zhao, Baohang Zhou, Xuhui Sui, Li Zhang, and Kehui Song. 2024. TimeR[4] : 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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

David Rau, Shuai Wang, Hervé Déjean, Stéphane Clinchant, and Jaap Kamps. 2025. Context embeddings for efficient answer generation in retrieval-augmented generation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, WSDM '25, page 493–502, New York, NY, USA. Association for Computing Machinery.

Stefano Giovanni Rizzo, Matteo Brucato, and Danilo Montesi. 2022. Ranking models for the temporal dimension of text. *ACM Transactions on Information Systems*, 41(2):1–34.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Daivik Sojitra, Raghav Jain, Sriparna Saha, Adam Jatowt, and Manish Gupta. 2024. Timeline summarization in the era of llms. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2657–2661.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

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*, pages 3040–3044.

Jannik Strötgen and Michael Gertz. 2010. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 321–324.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. *arXiv preprint arXiv:2306.08952*.

Luke Vilnis and Andrew McCallum. 2014. Word representations via gaussian embedding. *arXiv preprint arXiv:1412.6623*.

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.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, and 1 others. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*.

Chenyuan Wu, Ninglu Shao, Zheng Liu, Shitao Xiao, Chaozhuo Li, Chen Zhang, Senzhang Wang, and Defu Lian. 2025. Lighter and better: Towards flexible context adaptation for retrieval augmented generation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, WSDM '25, page 271–280, New York, NY, USA. Association for Computing Machinery.

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*, pages 2544–2553.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, 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*, pages 641–649.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Shohei Yoda, Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. 2023. Sentence representations via gaussian embedding. *arXiv preprint arXiv:2305.12990*.

Susik Yoon, Dongha Lee, Yunyi Zhang, and Jiawei Han. 2023. Unsupervised story discovery from continuous news streams via scalable thematic embedding. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 802–811.

A Dataset Construction Details

A.1 Training Dataset Construction

Figure 3 describes our training dataset generation pipeline. The construction is based on two main components: generating synthetic data based on predefined patterns and adapting an existing dataset for our task. The pipeline produces a dataset composed of tuples (s_1, d_1, s_2, d_2, ts), where s_1 and s_2 are texts containing temporal expressions, d_1 and d_2 are anchor dates at a daily granularity (YYYY-MM-DD), and ts is the temporal label between (s_1, d_1) and (s_2, d_2) as defined by our temporal similarity function (see Section A.1.1).

A.1.1 Temporal Similarity Function

Setup Given two TimeML annotations grounded as closed integer day intervals $I_q = [a_q, b_q]$ and $I_d = [a_d, b_d]$, where a single date is the degenerate case $a = b$ and year or month granularities map to the full corresponding range, we define a function $\mathrm{TSF}(I_q \to I_d) \in \mathbb{R}_{\geq 0}$ that quantifies how well the temporal extent of the document interval I_d accommodates that of the query interval I_q. The function serves as a soft supervision signal for contrastive training of a temporal embedding model whose inference-time similarity adopts the asymmetric Gaussian-KL scoring used in prior distributional embedding work (Vilnis and McCallum, 2014; He et al., 2015; Yoda et al., 2023).

We require four properties. First, the function should be **asymmetric**, in order to encode directional relations from Allen's interval algebra (Allen, 1983), such as *during* (I_q is contained in I_d) and *contains*, as distinct from their inverses. Second, it should be **non-degenerate on disjoint pairs**, ranking far-apart pairs differently from near-disjoint pairs rather than collapsing both to zero. Finally, it should be **continuous and smooth** in the four endpoints, with no discontinuity at the overlap/disjoint boundary.

Construction Following temporal IR work (Berberich et al., 2010; Kanhabua and Anand, 2016), we treat each grounded interval as a uniform probability distribution over its temporal extent, $T_I \sim \mathrm{U}[a, b+1)$, where the $+1$ maps the discrete inclusive interval to a continuous half-open interval. Two annotations are then compared as distributions rather than as sets.

For analytical tractability and to match the embedding head described in Section 5.2, we project each uniform distribution onto the Gaussian family by moment matching:

$$\begin{aligned} p_I &= \mathcal{N}(\mu_I, \sigma_I^2), \\ \mu_I &= \frac{a+b+1}{2}, \\ \sigma_I^2 &= \frac{(b-a+1)^2}{12}. \end{aligned} \tag{4}$$

These are the exact first and second central moments of T_I. The projection preserves temporal location through the mean and the uncertainty implied by granularity through the variance, while admitting closed-form divergences. Both parameters depend only on (a, b); no global constants enter the construction.

Definition We define the temporal similarity function as the asymmetric Kullback-Leibler divergence from the query distribution to the document distribution:

$$\begin{aligned} \mathrm{TSF}(I_q \to I_d) &:= D_{\mathrm{KL}}(p_{I_q} \,\|\, p_{I_d}) \\ &= \frac{1}{2} \log \frac{\sigma_d^2}{\sigma_q^2} \\ &\quad + \frac{\sigma_q^2 + (\mu_q - \mu_d)^2}{2\,\sigma_d^2} - \frac{1}{2}. \end{aligned} \tag{5}$$

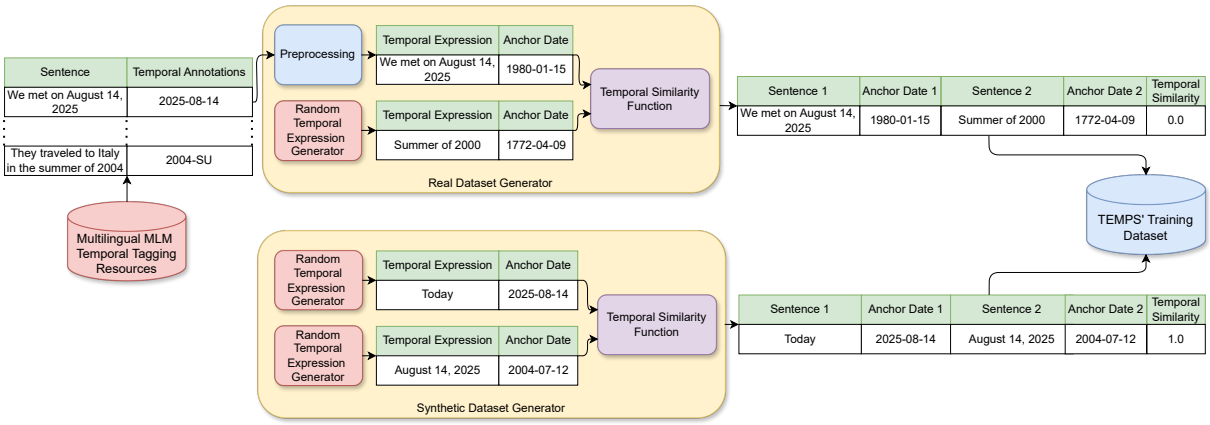

Figure 3: TEMPS' Training Dataset Creation Pipeline

Smaller values indicate that I_q is more temporally consistent with I_d. The function is non-negative and is zero iff $p_{I_q} = p_{I_d}$, i.e., iff I_q and I_d are identical intervals.

When a bounded similarity in $(0, 1]$ is required, such as for losses that expect similarity rather than distance, we use the monotone-decreasing transform

$$\sigma(I_q \to I_d) = \frac{1}{1 + \mathrm{TSF}(I_q \to I_d)},$$

mirroring the form of the model's inference-time similarity head.

Properties **Asymmetry.** When $\sigma_q \neq \sigma_d$, the function is asymmetric in its arguments. The variance term $\sigma_q^2/(2\sigma_d^2)$ is small when the query is sharper than the document and large in the reverse direction. For a day d inside its containing year Y, an instance of Allen's *during* relation, $\mathrm{TSF}(d \to Y)$ is small because the day is well explained by the year, whereas $\mathrm{TSF}(Y \to d)$ is large because the year's broad distribution is poorly explained by a single day. This is the kind of inclusion asymmetry that Gaussian-KL embeddings have previously modeled in lexical, graph, and knowledge-graph settings (Vilnis and McCallum, 2014; He et al., 2015; Bojchevski and Günnemann, 2018).

Non-degeneracy. The mean-squared term $(\mu_q - \mu_d)^2/(2\sigma_d^2)$ is unbounded above and grows quadratically with temporal separation. Distant disjoint pairs therefore receive larger labels than near-disjoint pairs, providing a well-ordered ranking signal across the full input space.

Smoothness. TSF is C^∞ in $(\mu_q, \sigma_q, \mu_d, \sigma_d)$ on $\sigma_q, \sigma_d > 0$, and therefore C^∞ in the four endpoint parameters. The overlap/disjoint boundary is invisible to the function.

Pair	$q \to d$	$d \to q$
Same interval	0.0	0.0
Year vs. day-in-middle	6.8×10^4	5.4
Year vs. day-at-boundary	2.7×10^5	6.9
Decade vs. year-inside	48.8	1.8
Adjacent years	6.0	6.0
Far-disjoint years (1990, 2025)	1.56×10^6	1.56×10^6
1-day overlap	4.86	4.86
1-day gap	6.00	6.00

Table 4: TSF values on canonical interval pairs, in both directions. Containment relations exhibit the targeted asymmetry, same-width pairs are symmetric, and far-disjoint pairs receive labels distinct from near-disjoint pairs.

Granularity invariance. The same closed form applies regardless of whether I_q and I_d have matched granularity. Coarse-fine pairs, such as a year and a day, and same-granularity pairs, such as two years, are handled uniformly, with the resulting asymmetry direction aligned with the inclusion relation. No special-case branching is required.

Compatibility with the embedding head. Our model emits a Gaussian $\mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ per text-date input and scores candidate pairs with $1/(1 + D_{\mathrm{KL}}(\hat{p}_q \| \hat{p}_d))$ (Section 5.2), mirroring the bounded inverse-KL score used by GaussCSE (Yoda et al., 2023). With TSF as the label function, training reduces to same-family distillation: the model learns to map anchored textual surface forms to Gaussian parameters that reproduce the closed-form temporal KL of the underlying intervals. The temporal-specific contribution is the interval grounding and moment-matched supervision, not the Gaussian-KL score itself.

Numerical illustration Table 4 reports TSF on canonical pairs. For a day in the middle of a

containing year, $\mathrm{TSF}(\text{day} \to \text{year}) = 5.4$ while $\mathrm{TSF}(\text{year} \to \text{day}) = 6.8 \times 10^4$. For two adjacent same-width years, the function is symmetric, with both directions equal to 6.0. For far-disjoint years it grows quadratically with separation, reaching 1.56×10^6 for 1990 vs. 2025. The transition from overlapping to gapped intervals is smooth: a one-day overlap gives 4.86 and a one-day gap gives 6.00, unlike IoU-based schemes that exhibit a discontinuity at the same boundary.

A.1.2 Data Sources

Synthetic Dataset To enable scalable and diverse temporal understanding, we construct a synthetic dataset spanning the years 1000 to 2030. TimeML annotations are randomly generated and converted into natural language using predefined templates (Table 5). Each expression is paired with a randomly assigned anchor date (daily granularity) to resolve context-dependent expressions.

Table 5: Sample TimeML Annotations with Variable Offsets

TimeML Annotation	Text Template
OFFSET P{n}D	"{n} days ago/later"
OFFSET P{n}W	"{n} weeks ago/later"
OFFSET P{n}M	"{n} months ago/later"
OFFSET P{n}Y	"{n} years ago/later"
THIS P{n}D	"these {n} days"
THIS P{n}W	"these {n} weeks"
THIS P{n}M	"these {n} months"
THIS P{n}Y	"these {n} years"
{date1},{date2}	"spanning from {date1} to {date2}"
{date1},{date2}	"between {date1} and {date2}"

Multilingual MLM Temporal Tagging Dataset We incorporate the English portion of the Multilingual MLM Temporal Tagging Resources dataset (Lange et al., 2022). This dataset provides text from Wikipedia with temporal expressions automatically annotated using a modified HeidelTime (Strötgen and Gertz, 2010) tagger. This serves as a source of weakly supervised, real-world linguistic patterns to complement our synthetic data.

A.1.3 Training Instance Generation

We apply a unified processing pipeline to both data sources to create a contrastive learning dataset. For every temporal expression t found in either source, we generate one positive sample (s_0) and four negative samples ($s_1 ... s_4$).

The positive sample is created by slightly modifying the boundaries of t while maintaining a bounded temporal similarity > 0.9, and generating a corresponding textual representation. Negative samples are chosen at random. Random anchor dates are assigned to all negative samples, while the positive sample retains the original or logically derived anchor.

The final dataset consists of 60M tuples (s_1, d_1, s_2, d_2, ts), obtained by concatenating 55M synthetic tuples and 5M real-world tuples. Here, s_1 and s_2 are temporal expression strings, d_1 and d_2 are their respective anchor dates (YYYY-MM-DD), and ts denotes the temporal label computed via our temporal similarity function or its bounded monotone transform, depending on the training objective.

B Temporal Model Details

Input Encoding. The temporal branch operates on a sentence and an anchor date. The sentence is processed by a frozen all-MiniLM-L6-v2 backbone to produce contextualized hidden states $\mathbf{H}$. The anchor date is encoded using a sinusoidal positional encoder following (Jia et al., 2021). Let $k \in \mathbb{N}$ be the number of days between the input date and 1 January 1000. The encoder is defined as:

$$PE_{2i}(t) = \sin\left(\frac{k}{n^{2i/d}}\right),$$
$$PE_{2i+1}(t) = \cos\left(\frac{k}{n^{2i/d}}\right),$$

where d is the output embedding dimension, set to 32 in our experiments, $i \in [0, d/2]$, and n is a frequency scaling factor.

Gaussian Embedding Head. The hidden states are aggregated with attention pooling to emphasize temporal tokens:

$$\mathbf{e}_{\text{sent}} = \mathrm{AttentionPooling}(\mathbf{H}).$$

The pooled sentence vector is concatenated with the date representation $\mathbf{p}_{\text{date}}$ and passed through a temporal learning network:

$$\mathbf{z} = \mathrm{TemporalLearning}(\mathbf{e}_{\text{sent}} \oplus \mathbf{p}_{\text{date}}).$$

Two output heads then estimate the Gaussian parameters:

$$\mu = \text{MeanNN}(\mathbf{z}), \qquad \sigma = \text{VarianceNN}(\mathbf{z}).$$

We use only the diagonal elements of the covariance matrix for computational efficiency (Vilnis and McCallum, 2014; Yoda et al., 2023). Gaussian embeddings are useful because the mean represents temporal location and the variance represents uncertainty or granularity, enabling asymmetric inclusion behavior through KL divergence.

Training Objective. Let $(s_1^i, d_1^i, s_2^i, d_2^i, ts_i)$ be a training point. The model produces Gaussian embeddings for both anchored expressions, and their bounded temporal score is:

$$\begin{aligned} s_i &= \frac{1}{1 + D_{\text{KL}}(p_1^i \,\|\, p_2^i)}, \\ p_j^i &= \mathcal{N}\big(\mu_j^i,\, \text{diag}((\sigma_j^i)^2)\big). \end{aligned} \tag{6}$$

We adapt the CoSENT loss (Huang et al., 2024) to enforce the ordering induced by the ground-truth temporal labels:

$$\mathcal{L} = \log\left(1 + \sum_{i,j:\, ts_i < ts_j} \exp\left(\lambda \cdot (s_i - s_j)\right)\right), \tag{7}$$

where $\lambda = 20$ in our experiments.

Training Configuration. The model was trained for three epochs over the full training corpus on a single HPC `gpu_p5` node with exclusive allocation. The node provides two AMD EPYC 7543 CPUs, 512 GiB of DDR4 memory, and eight NVIDIA A100-SXM4 GPUs with 80 GB of HBM2 memory each. Training used PyTorch's distributed launcher (`torchrun`, one process per GPU) with a global batch size of 2,048, a learning rate of 3×10^{-4}, weight decay of 0.01, a linear warmup over the first 5% of optimization steps, a maximum input length of 512 tokens, 16-bit floating-point precision, and random seed 42. Intermediate validation is performed every 500 optimization steps.

C Experimental Details

C.1 Baselines

To evaluate the effectiveness of our proposed temporal embedding model, we benchmark its performance against a set of baselines stratified by model size. This allows us to assess whether simply scaling up model parameters captures temporal nuances or if specialized architecture is required regardless of size.

We categorize our dense retrieval baselines into three distinct tiers based on parameter count:

- **Base-Scale Models:** We employ **MPNet$_{\text{base}}$-v2**[1] (Song et al., 2020) and **E5$_{\text{base}}$-v2**[2] (Wang et al., 2022). These models represent the standard for efficient, low-latency deployment. We include them to determine if our method provides significant gains in resource-constrained environments where larger models are impractical.

- **Large-Scale Models:** We include **BGE$_{\text{large}}$-v1.5**[3] (Xiao et al., 2024) to represent the upper bound of traditional BERT-based encoder architectures. This baseline tests whether increasing the capacity of pure encoder models is sufficient to resolve complex temporal queries.

- **LLM-Based Models:** We use **Mistral**[4], a state-of-the-art retrieval model finetuned from Mistral-7B. By comparing against a model orders of magnitude larger, we investigate whether the emergent reasoning capabilities of Large Language Models render explicit temporal modeling obsolete.

- **State-of-the-Art Temporal Baseline**: We compare against TSContriever (Wu et al., 2024), a dense retrieval model explicitly optimized for time-sensitive questioning. We also include its variant, TSContriever + Query Router, which selectively employs temporal reasoning modules. Including these ensures we benchmark TEMPS against the current specialized state-of-the-art, not just general-purpose embeddings.

In addition to these dense retrievers, we include **BM25** (lexical baseline) and **SUTime** (Chang and Manning, 2012) (rule-based baseline). Finally, we compare all dense models against their augmented versions (e.g., *Mistral + TEMPS*) to isolate the specific performance contributions of our temporal module.

[1] `sentence-transformers/all-mpnet-base-v2`
[2] `intfloat/e5-base-v2`
[3] `BAAI/bge-large-en-v1.5`
[4] `Salesforce/SFR-Embedding-Mistral`

C.2 Benchmarks and Metrics

We evaluate our system using three temporal benchmarks: **TimeQA** (Chen et al., 2021) assesses the capability to understand the time scope of evolving facts within long documents, specifically evaluating **temporal understanding** of explicit and implicit time mentions. **TempReason** (Tan et al., 2023) evaluates specific reasoning levels, combining abstract logic rules (Level 1) and grounding events to valid time ranges (Level 2). We omit Level 3 from this study, as it focuses on event-event relations, whereas our research targets temporal relations covered by the first two levels. **TS-Retriever** (Wu et al., 2024) focuses on the retrieval stage by testing the distinction of semantically similar documents based on temporal constraints via **explicit** (specific years), **implicit** (vague periods), and **hybrid** (ranges) query types. Together, these benchmarks provide a comprehensive assessment of factual, retrieval-based, and arithmetic temporal reasoning, sharing the common feature of evaluating time-sensitive retrieval from a **given set of documents**. We additionally use TimeQA for downstream RAG evaluation by passing the top-5 retrieved passages to Qwen2.5-7B-Instruct (Yang et al., 2024) and evaluating the generated answer.

Dataset	Number of Entries
TimeQA	887
TS-Retriever	3244
TempReason	9397

Table 6: Overview of evaluation datasets used in our experiments.

Retrieval effectiveness is evaluated using four ranking metrics: MRR, NDCG@5, Recall@5, and Precision@5. Downstream RAG effectiveness is evaluated separately using exact match (EM), token-level F1, and answer containment, where containment measures whether a normalized gold answer string appears in the generated response.

C.3 Effect of Alpha Values

The interpolation weight α controls how much the final retrieval score relies on the temporal branch relative to the semantic branch. We therefore evaluate a sweep over candidate α values for each benchmark, using the same value for all +TEMPS semantic backbones within a benchmark. Figure 4 visualizes the resulting sensitivity curves.

The sweep shows that the best fusion weight is not universal across temporal IR settings. TimeQA and TempReason peak at larger α values, consistent with their emphasis on evolving facts, date arithmetic, and event-time grounding. In these benchmarks, temporal compatibility is often the main cue that separates correct from incorrect evidence. TS-Retriever peaks at a smaller α, reflecting a different trade-off: candidates are deliberately semantically similar, so the temporal branch is useful as a corrective signal but should not dominate the semantic score. This supports our use of benchmark-level calibration values rather than a single global α for all tasks.

C.4 Computational Resources

All experiments were conducted on an HPC partition. Each compute node provides two AMD EPYC 7543 CPUs (64 physical cores, 128 threads in total), 512 GiB of DDR4 memory, and eight NVIDIA A100-SXM4 GPUs with 80 GB of HBM2 memory each, interconnected via NVLink and NVSwitch. Training was performed on a single node with exclusive allocation, distributing the workload across all eight A100 GPUs through PyTorch's distributed launcher (`torchrun`, one process per GPU) for a batch size of 2,048. Training the temporal module for three epochs over the full corpus took approximately 4 wall-clock hours on the eight A100 GPUs. Training and evaluation scripts are available in the accompanying code repository at https://anonymous.4open.science/r/temporal-embeddings-7E49/README.md.

C.5 Artifacts, Licenses, and Intended Use

All datasets (TimeQA (Chen et al., 2021), TempReason (Tan et al., 2023), TS-Retriever (Wu et al., 2024), and the Multilingual MLM Temporal Tagging Resources (Lange et al., 2022)) and pretrained models (all-MiniLM-L6-v2, $\text{MPNet}_{\text{base}}$-v2 (Song et al., 2020), E5_{base}-v2 (Wang et al., 2022), $\text{BGE}_{\text{large}}$-v1.5 (Xiao et al., 2024), SFR-Embedding-Mistral, and Qwen2.5-7B-Instruct (Yang et al., 2024)) used in this work are publicly released for research, and we use them strictly within that intended research scope. Our synthetic temporal data is generated from templates and the publicly available Wikipedia-derived Multilingual MLM Temporal Tagging Resources (Lange et al., 2022). We release our code and the TEMPS temporal module under the MIT license, intended solely for research

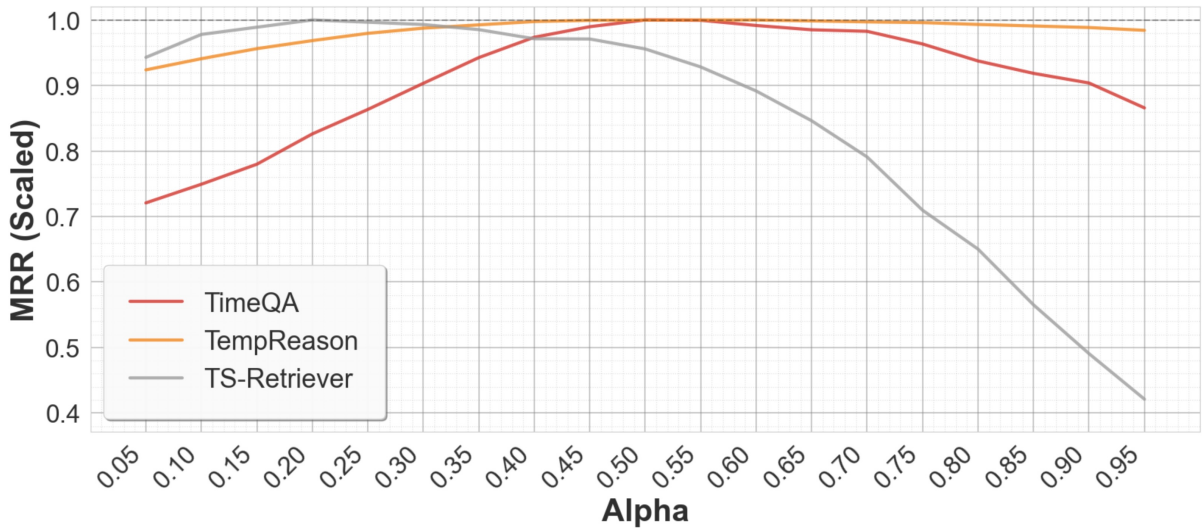

Figure 4: Effect of varying the temporal-semantic interpolation weight α across benchmarks. Low values rely mostly on semantic similarity, while high values rely increasingly on the temporal branch. The selected benchmark-level values balance these two signals: TimeQA and TempReason favor larger temporal weights, whereas TS-Retriever favors a smaller temporal weight because semantic discrimination among topically similar candidates remains important.

on temporal retrieval.

C.6 Statistical Significance

For every (benchmark, baseline) cell of the main results table we compare the semantic baseline alone against the same baseline fused with the trained TEMPS temporal model at the per-benchmark α reported in the body (TimeQA $\alpha = 0.6$, TempReason $\alpha = 0.7$, TS-Retriever $\alpha = 0.2$). These values are treated as benchmark-level calibration constants from the same candidate sweep and are not varied by semantic backbone or by metric in the significance analysis. Per-query metric scores are computed from the cached retrieval similarities; the hybrid follows the per-query min-max-normalized merge $\alpha \cdot s_{\text{temporal}} + (1 - \alpha) \cdot s_{\text{semantic}}$. For each (baseline, metric) pair we run a two-sided paired t-test on the per-query metric differences and report a bootstrap-resampled 95% confidence interval on the mean per-query gain ($B = 10{,}000$ resamples, seed 42).

All 48 comparisons (3 benchmarks $\times$ 4 baselines $\times$ 4 metrics) are significant at $p < 5 \times 10^{-5}$. After Bonferroni correction across all comparisons ($\alpha = 0.05/48 \approx 1.04 \times 10^{-3}$) every comparison remains significant by at least an order of magnitude. A literal entry of $< 2.2 \times 10^{-308}$ in the tables below indicates that the analytical p-value returned by SciPy underflowed the IEEE-754 double-precision floor; the actual value is smaller still and is reported verbatim as the underflow bound.

Per-metric results. Full per-metric statistics for TimeQA, TempReason, and TS-Retriever are in Tables 7-9.

Table 7: TimeQA per-cell statistics ($\alpha = 0.6$, $n = 887$ queries). Two-sided paired t-test; 95% bootstrap CI on the mean per-query gain.

Baseline	Metric	Base	+TEMPS	Δ	95% CI	t-test p
MPNet$_{\text{base}}$-v2	MRR	0.2985	0.5071	$+0.2086$	$[0.1810, 0.2365]$	1.10×10^{-44}
MPNet$_{\text{base}}$-v2	NDCG@5	0.3087	0.5360	$+0.2274$	$[0.1987, 0.2561]$	2.83×10^{-46}
MPNet$_{\text{base}}$-v2	Recall@5	0.4746	0.6967	$+0.2221$	$[0.1849, 0.2593]$	7.17×10^{-30}
MPNet$_{\text{base}}$-v2	Precision@5	0.0949	0.1393	$+0.0444$	$[0.0370, 0.0519]$	7.17×10^{-30}
E5$_{\text{base}}$-v2	MRR	0.3283	0.5398	$+0.2115$	$[0.1868, 0.2360]$	3.82×10^{-56}
E5$_{\text{base}}$-v2	NDCG@5	0.3664	0.5726	$+0.2061$	$[0.1807, 0.2310]$	3.17×10^{-51}
E5$_{\text{base}}$-v2	Recall@5	0.5998	0.7317	$+0.1319$	$[0.0992, 0.1646]$	8.79×10^{-15}
E5$_{\text{base}}$-v2	Precision@5	0.1200	0.1463	$+0.0264$	$[0.0198, 0.0329]$	8.79×10^{-15}
BGE$_{\text{large}}$-v1.5	MRR	0.2942	0.5346	$+0.2404$	$[0.2148, 0.2655]$	2.20×10^{-66}
BGE$_{\text{large}}$-v1.5	NDCG@5	0.3238	0.5632	$+0.2394$	$[0.2121, 0.2664]$	4.15×10^{-58}
BGE$_{\text{large}}$-v1.5	Recall@5	0.5412	0.7159	$+0.1747$	$[0.1387, 0.2108]$	2.59×10^{-20}
BGE$_{\text{large}}$-v1.5	Precision@5	0.1082	0.1432	$+0.0349$	$[0.0277, 0.0422]$	2.59×10^{-20}
Mistral	MRR	0.4085	0.5427	$+0.1341$	$[0.1066, 0.1607]$	9.25×10^{-22}
Mistral	NDCG@5	0.4478	0.5710	$+0.1232$	$[0.0961, 0.1497]$	2.49×10^{-18}
Mistral	Recall@5	0.6561	0.7215	$+0.0654$	$[0.0327, 0.0958]$	4.98×10^{-5}
Mistral	Precision@5	0.1312	0.1443	$+0.0131$	$[0.0068, 0.0194]$	4.98×10^{-5}

Table 8: TempReason per-cell statistics ($\alpha = 0.7$, $n = 9{,}397$ queries). Two-sided paired t-test; 95% bootstrap CI on the mean per-query gain. Entries of $< 2.2\times10^{-308}$ indicate SciPy's float64 underflow.

Baseline	Metric	Base	+TEMPS	Δ	95% CI	t-test p
MPNet$_{\text{base}}$-v2	MRR	0.7269	0.9026	$+0.1757$	$[0.1696, 0.1819]$	$<2.2\times10^{-308}$
MPNet$_{\text{base}}$-v2	NDCG@5	0.7700	0.9229	$+0.1529$	$[0.1477, 0.1581]$	$<2.2\times10^{-308}$
MPNet$_{\text{base}}$-v2	Recall@5	0.9356	0.9917	$+0.0561$	$[0.0517, 0.0605]$	2.25×10^{-132}
MPNet$_{\text{base}}$-v2	Precision@5	0.2010	0.2162	$+0.0151$	$[0.0140, 0.0163]$	7.58×10^{-134}
E5$_{\text{base}}$-v2	MRR	0.8174	0.9161	$+0.0988$	$[0.0931, 0.1042]$	4.29×10^{-257}
E5$_{\text{base}}$-v2	NDCG@5	0.8565	0.9342	$+0.0777$	$[0.0735, 0.0819]$	8.03×10^{-267}
E5$_{\text{base}}$-v2	Recall@5	0.9846	0.9936	$+0.0090$	$[0.0071, 0.0110]$	5.95×10^{-20}
E5$_{\text{base}}$-v2	Precision@5	0.2147	0.2169	$+0.0022$	$[0.0017, 0.0027]$	2.24×10^{-18}
BGE$_{\text{large}}$-v1.5	MRR	0.7972	0.9170	$+0.1198$	$[0.1141, 0.1253]$	$<2.2\times10^{-308}$
BGE$_{\text{large}}$-v1.5	NDCG@5	0.8377	0.9345	$+0.0968$	$[0.0925, 0.1012]$	$<2.2\times10^{-308}$
BGE$_{\text{large}}$-v1.5	Recall@5	0.9768	0.9931	$+0.0163$	$[0.0139, 0.0187]$	1.12×10^{-39}
BGE$_{\text{large}}$-v1.5	Precision@5	0.2123	0.2167	$+0.0044$	$[0.0038, 0.0051]$	8.11×10^{-41}
Mistral	MRR	0.8161	0.9183	$+0.1022$	$[0.0968, 0.1074]$	1.59×10^{-289}
Mistral	NDCG@5	0.8518	0.9353	$+0.0835$	$[0.0793, 0.0877]$	$<2.2\times10^{-308}$
Mistral	Recall@5	0.9769	0.9925	$+0.0157$	$[0.0133, 0.0181]$	1.69×10^{-38}
Mistral	Precision@5	0.2120	0.2166	$+0.0045$	$[0.0039, 0.0052]$	1.09×10^{-42}

Table 9: TS-Retriever per-cell statistics ($\alpha = 0.2$, $n = 3{,}244$ queries). Two-sided paired t-test; 95% bootstrap CI on the mean per-query gain.

Baseline	Metric	Base	+TEMPS	Δ	95% CI	t-test p
MPNet$_{\text{base}}$-v2	MRR	0.3849	0.5092	$+0.1243$	$[0.1157, 0.1330]$	3.22×10^{-161}
MPNet$_{\text{base}}$-v2	NDCG@5	0.2381	0.3437	$+0.1056$	$[0.0998, 0.1115]$	3.49×10^{-233}
MPNet$_{\text{base}}$-v2	Recall@5	0.2378	0.3328	$+0.0950$	$[0.0880, 0.1023]$	1.04×10^{-137}
MPNet$_{\text{base}}$-v2	Precision@5	0.1632	0.2357	$+0.0725$	$[0.0679, 0.0769]$	9.76×10^{-190}
E5$_{\text{base}}$-v2	MRR	0.7840	0.8052	$+0.0212$	$[0.0136, 0.0288]$	4.72×10^{-8}
E5$_{\text{base}}$-v2	NDCG@5	0.6405	0.6820	$+0.0415$	$[0.0362, 0.0469]$	6.05×10^{-50}
E5$_{\text{base}}$-v2	Recall@5	0.6110	0.6532	$+0.0422$	$[0.0363, 0.0482]$	4.23×10^{-43}
E5$_{\text{base}}$-v2	Precision@5	0.4221	0.4527	$+0.0306$	$[0.0265, 0.0346]$	2.51×10^{-47}
BGE$_{\text{large}}$-v1.5	MRR	0.7136	0.7896	$+0.0760$	$[0.0685, 0.0839]$	4.64×10^{-78}
BGE$_{\text{large}}$-v1.5	NDCG@5	0.5450	0.6362	$+0.0912$	$[0.0853, 0.0972]$	3.51×10^{-176}
BGE$_{\text{large}}$-v1.5	Recall@5	0.5157	0.5987	$+0.0831$	$[0.0759, 0.0904]$	3.82×10^{-105}
BGE$_{\text{large}}$-v1.5	Precision@5	0.3598	0.4166	$+0.0568$	$[0.0524, 0.0612]$	9.94×10^{-126}
Mistral	MRR	0.7505	0.8114	$+0.0609$	$[0.0538, 0.0680]$	2.63×10^{-61}
Mistral	NDCG@5	0.6103	0.6816	$+0.0713$	$[0.0661, 0.0767]$	6.39×10^{-140}
Mistral	Recall@5	0.5856	0.6448	$+0.0592$	$[0.0530, 0.0655]$	1.55×10^{-73}
Mistral	Precision@5	0.4058	0.4482	$+0.0424$	$[0.0384, 0.0462]$	4.28×10^{-92}