Chronocept: Instilling a Sense of Time in Machines

Anonymous ACL submission

Abstract

Human cognition is deeply intertwined with a sense of time, known as *Chronoception*. This sense allows us to judge how long facts remain valid and when knowledge becomes outdated. Despite progress in vision, language, and motor control, AI still struggles to reason about temporal validity. We introduce Chronocept, the first benchmark to model temporal validity as a continuous probability distribution over time. Using skew-normal curves fitted along semantically decomposed temporal axes, Chronocept captures nuanced patterns of emergence, decay, and peak relevance. It includes two datasets: Benchmark I (atomic facts) and Benchmark II (multi-sentence passages). Annotations show strong inter-annotator agreement (84% and 89%). Our baselines predict curve parameters - location, scale, and skewness - enabling interpretable, generalizable learning and outperforming classification-based approaches. Chronocept fills a foundational gap in AI's temporal reasoning, supporting applications in knowledge grounding, fact-checking, retrieval-augmented generation (RAG), and proactive agents. Code and data are publicly available.

1 Introduction

Humans effortlessly track how information changes in relevance over time. We instinctively know when facts emerge, become useful, or fade into obsolescence - a cognitive ability known as Chronoception (Fontes et al., 2016; Zhou et al., 2019). This higher-order perception of time plays a crucial role in how we evaluate the persistence and usefulness of information in real-world contexts. Despite excelling in pattern recognition (He et al., 2016), language generation (Brown et al., 2020), and motor control (Levine et al., 2016), modern AI systems remain largely insensitive to the temporal validity of the information they process.

Prior work has advanced temporal understanding via event ordering (Allen, 1983; Ning et al., 2020; Wen and Ji, 2021), timestamp prediction (Kanhabua and Nørvåg, 2008; Kumar et al., 2012; Das et al., 2017), and temporal commonsense reasoning (Zhou et al., 2019). However, these approaches often reduce time to static labels or binary transitions. Even recent efforts in temporal validity change prediction (Wenzel and Jatowt, 2024) model shifts as discrete class changes, neglecting the gradual and asymmetric nature of temporal decay.

We introduce Chronocept, a benchmark that models temporal validity as a continuous probability distribution over time. Using a skewed-normal distribution over logarithmic time, parameterized by location (ξ), scale (ω), and skewness (α) (Azzalini, 1986; Schmidt et al., 2017), Chronocept captures subtle temporal patterns such as delayed peaks and asymmetric decay.

To support structured supervision, we decompose each sample along semantic temporal axes. We release two benchmarks: Benchmark I features atomic factual statements, and Benchmark II contains multi-sentence passages with temporally interdependent elements. High inter-annotator agreement across segmentation, axis labeling, and curve parameters validates annotation quality.

We benchmark a diverse set of models, including linear regression, SVMs, XGBoost, FFNNs, BiLSTMs, and BERT (Devlin et al., 2019). FFNNs perform best on the simpler Benchmark I, while BiLSTMs excel on the more complex Benchmark II. Surprisingly, fine-tuned BERTs do not outperform simpler architectures. To assess the role of temporal structure, we conduct ablations that remove or shuffle temporal axes during training - both lead to marked performance drops.

Chronocept enables several downstream applications. In Retrieval-Augmented Generation (RAG), temporal curves guide time-sensitive retrieval; in fact-checking, they help flag decaying or stale facts. Most importantly, Chronocept lays the foundation for proactive AI systems that reason not just about

what to do, but when to do it (Miksik et al., 2020).

All resources - dataset, and baseline implementations - are publicly available to support future research in machine time-awareness.

2 Related Work

2.1 Temporal Validity Prediction

In the earliest attempt to formalize the temporal validity of information, Takemura and Tajima (2012) proposed the concept of "content viability" by classifying tweets into "read now," "read later," and "expired" categories, to prioritize timeliness in information consumption. However, their approach assumed a rigid, monotonic decay of relevance, failing to model scenarios where validity peaks later rather than at publication. This restricted its applicability beyond real-time contexts such as Twitter streams.

Almquist and Jatowt (2019) extended this work by defining a "validity period" and effectively proposing a "content expiry date" for sentences, using linguistic and statistical features. However, their reliance on static time classes (e.g., hours, days, weeks) sacrificed granularity, and their approach required explicit feature engineering rather than leveraging more advanced, data-driven methods (Das et al., 2017).

Traditional approaches (Almquist and Jatowt, 2019; Lynden et al., 2023; Hosokawa et al., 2023) mostly treat validity as binary, where information is either valid or invalid at a given time, this can be formulated as:

$$\text{validity}_i(t) = \begin{cases} \text{True} & \text{if information } i \text{ is valid at } t, \\ \text{False} & \text{otherwise} \end{cases} \quad (1)$$

where i represents the information under consideration and t denotes the time at which its validity is evaluated. However, this model overlooks gradual decay, recurrence, and asymmetric relevance patterns.

More recently, Wenzel and Jatowt (2024) introduced Temporal Validity Change Prediction (TVCP), which models how context alters a statement's validity window. However, it does not quantify validity as a continuous probability distribution over time.

Chronocept advances this field by defining temporal validity as a continuous probability distribution, allowing a more precise and flexible representation of how information relevance evolves.

2.2 Temporal Reasoning and Commonsense

Temporal reasoning has largely focused on event ordering (Allen, 1983; Wen and Ji, 2021; Ning et al., 2020), predicting temporal context (Kanhabua and Nørvåg, 2008; Kumar et al., 2012; Das et al., 2017; Luu et al., 2021; Jatowt et al., 2013), and commonsense knowledge (Zhou et al., 2019). While these studies laid the groundwork for understanding event sequences, durations, and frequencies, recent work has expanded into implicit or commonsense dimensions of temporal reasoning.

TORQUE (Ning et al., 2020) is a benchmark designed for answering temporal ordering questions, while TRACIE, along with its associated model SYMTIME (Zhou et al., 2021), primarily ensures temporal-logical consistency rather than modeling truth probabilities.

McTACO (Zhou et al., 2019) evaluates temporal commonsense across five dimensions: event duration, ordering, frequency, stationarity, and typical time of occurrence. McTACO assesses whether a given statement aligns with general commonsense expectations, and does not quantify the likelihood of a statement being true over time.

Recent work Wenzel and Jatowt, 2023; Jain et al., 2023 has explored how LLMs handle temporal commonsense, exposing inconsistencies in event sequencing and continuity. However, these studies do not incorporate probabilistic modeling of temporal validity - a core focus of Chronocept, which models truthfulness as a dynamic, evolving probability distribution.

2.3 Dataset Structuring for Temporal Benchmarks

Temporal annotation frameworks like TimeML (Pustejovsky et al., 2003) and ISO-TimeML (Pustejovsky et al., 2010) focus on static event relationships, often suffering from low inter-annotator agreement due to event duration ambiguities. The TimeBank series (Pustejovsky, 2003; Cassidy et al., 2014) and TempEval challenges (Verhagen, 2007, 2010; UzZaman et al., 2012) expanded evaluations but remained limited in modeling evolving event validity.

In response, Ning et al. (2018) proposed a multi-axis annotation scheme that structures events into eight distinct categories - Main, Intention, Opinion, Hypothetical, Negation, Generic, Static, and Recurrent. Additionally, the scheme prioritizes event start points over full event intervals, reducing

ambiguity and significantly improving IAA scores. Chronocept builds on this by refining multi-axis annotation to model temporal validity, capturing how information relevance shifts over time through probabilistic distributions.

2.4 Statistical Modeling of Temporal Data Using Skewed Normal Distribution

Traditional normal distributions often fail to capture skewed temporal patterns. The skew-normal distribution (Azzalini, 1986, 1996) provides a more flexible alternative by incorporating asymmetry, improving accuracy in modeling time-dependent information relevance (Schmidt et al., 2017). Chronocept employs this distribution to capture various temporal behaviors, including gradual decay, peak relevance, and rapid obsolescence.

3 Chronocept: Task & Benchmark Design

3.1 Problem Definition

Temporal Validity Prediction (TVP) of Information seeks to model how long a factual statement remains true after it is published.

We formalize Temporal Validity Prediction as a probabilistic task of modeling information's relevance as a continuous probability distribution over time, rather than the binary-or-multiclass settings common in earlier work.

Let $T \subseteq \mathbb{R}_{\geq 0}$ denote the time domain, where $t \geq 0$ represents the elapsed time since publication of information i.

Then, we define a binary random variable,

$$\text{validity}_i(t) \in \{0, 1\} \tag{2}$$

where $\text{validity}_i(t) = 1$ indicates that the information i is valid at time t, and $\text{validity}_i(t) = 0$ otherwise.

Rather than predicting $\text{validity}_i(t)$ directly, TVP aims to learn a continuous probability density function $p_i(t)$

$$p_i(t) = P\big(\text{validity}_i(t) = 1\big),\ p_i : T \to [0, 1] \tag{3}$$

Accordingly, the probability that the statement remains valid throughout any interval $[a, b] \subseteq T$ is given by

$$P\Big(\forall\, t \in [a, b],\ \text{validity}_i(t) = 1\Big) = \int_a^b p_i(t)\, dt \tag{4}$$

Crucially, the model does not impose rigid boundary constraints - such as $p_i(0) = 1$ or monotonic decay - thereby permitting the learned distribution to capture complex temporal phenomena, including delayed onset, non-monotonic plateaus, and intermittent resurgences (Takemura and Tajima, 2012; Almquist and Jatowt, 2019)

3.2 Modeling Temporal Validity

We model the temporal validity of statements using a probability curve, with likelihood of being valid on the Y-axis and time since publication on the X-axis. To reduce ambiguity, sentences are decomposed along semantically distinct axes. A skew-normal distribution on a logarithmic time scale captures the validity dynamics.

Axes-Based Decomposition. We adopt the multi-axis annotation scheme of Ning et al. (2018) (MATRES), which partitions each sentence into eight semantically coherent axes (Main, Intention, Opinion, Hypothetical, Generic, Negation, Static, Recurrent). By isolating relation annotation within each axis, MATRES reduces cross-category ambiguity and better aligns with human temporal perception.

In our ablation Appendix F, removing axis features increases MSE by 4.57%, confirming that axis-level signals are essential for precise temporal modeling.

Skewed Normal Distribution. We model temporal validity using the skewed normal distribution, a generalization of the Gaussian with a shape parameter α that captures asymmetry. This enables representation of non-symmetric temporal patterns such as delayed onset, gradual decay, or skewed relevance, which symmetric (Gaussian) or memoryless (exponential) distributions fail to model.

The probability density function is:

$$f(x; \xi, \omega, \alpha) = \frac{2}{\omega}\, \phi\left(\frac{x - \xi}{\omega}\right)\, \Phi\left(\alpha\, \frac{x - \xi}{\omega}\right) \tag{5}$$

where:

- $\phi(z)$ is the standard normal PDF,
- $\Phi(z)$ is the standard normal CDF,
- ξ is the location parameter - determining the time at which an event is most likely valid,
- ω is the scale parameter - governing the duration of validity, and

- α is the shape parameter - controlling skewness (with positive values yielding right skew and negative values left skew).

Quantitative comparisons against Gaussian, log-normal, exponential and gamma distributions in Appendix D support this choice.

Logarithmic Time Scale. Linear time yields sparse coverage over key intervals, particularly at minute-level granularity. To address this, we compress the time axis using a monotonic logarithmic transformation:

$$t' = \log_{1.1}(t) \tag{6}$$

We default to a base of 1.1 for the near-linear spacing across canonical intervals (e.g., minutes, days, decades) while preserving granularity. Chronocept's target values remain compatible with alternative bases. See Appendix C for the base transformation framework, compression analysis, and the provided code implementation.

4 Dataset Creation

4.1 Benchmark Generation & Pre-Filtering

Chronocept comprises two benchmarks to facilitate evaluation across varying complexity levels. Benchmark I consists of 1,254 samples featuring simple, single-sentence texts with clear temporal relations - ideal for baseline reasoning - while Benchmark II includes 524 samples with complex, multi-sentence texts capturing nuanced, interdependent temporal phenomena.

Synthetic samples were generated using the GPT-o1[1] model (OpenAI, 2024) with tailored prompts to ensure temporal diversity across benchmarks. Full prompts for both benchmarks are disclosed in Appendix E for reproducibility. No real-world or personally-identifying data was used, ensuring complete privacy.

In the pre-annotation phase, SBERT[2] (Reimers and Gurevych, 2019) and TF-IDF embeddings were generated for all samples, and pairwise cosine similarities were calculated. Samples with SBERT or TF-IDF similarity exceeding 0.7 (70%) were removed to reduce semantic and lexical redundancy.

Annotation guidelines are disclosed in Appendix A and were continuously accessible during annotation.

[1] https://openai.com/o1

[2] all-MiniLM-L6-v2 available at https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

4.2 Annotation Workflow

Annotation Process. Our protocol consists of three steps: (i) *Temporal Segmentation* – partitioning text into coherent subtexts that preserve temporal markers; (ii) *Axis Categorization* – assigning each segment to one of eight temporal axes (Main, Intention, Opinion, Hypothetical, Generic, Negation, Static, Recurrent); and (iii) *Temporal Validity Distribution Plotting* – annotating a skewed normal distribution, parameterized by location (ξ), scale (ω), and skewness (α), over a logarithmic time axis.

To ensure interpretability and consistency, all parent texts are written in the present tense, distributions are anchored at $t = 0$, and multimodal curves are excluded. Additionally, any samples that did not exhibit a clearly assignable main timeline or violated these constraints were flagged and discarded during the annotation process.

4.3 Annotator Training & Quality Control

Eight third-year B.Tech. students with relevant coursework in Natural Language Processing, Machine Learning, and Information Retrieval participated. They underwent a 1-hour training session and a supervised warm-up on 50 samples. Agreement thresholds were set at ICC > 0.90 for numerical annotations, Jaccard Index > 0.75 for segment-level annotations, and $P_k < 0.15$ for segmentation consistency during this warm-up phase.

Each sample was annotated independently by two annotators. Quality control included daily reviews of 10% of annotations, a limit of 70 samples per annotator per day to mitigate fatigue, and automated flagging of samples with segmentation mismatches, target deviations $>2\sigma$, or $P_k > 0.2$. Discrepancies were adjudicated or, if unresolved, discarded.

No personal or identifying information was collected or stored during the annotation process.

Handling Edge Cases and Final Resolution. Ambiguous samples were flagged or discarded following the three-phase filtering scheme. For segmentation and axis labels, a union-based approach retained all plausible interpretations, recognizing that axis confusion may encode aspects of human temporal cognition useful for future modeling. For temporal validity targets (ξ, ω, α), annotator values were averaged to yield smooth probabilistic supervision rather than discrete target selection.

```
{
    "_id": "H0028",
    "parent_text": "They are discussing a philosophical concept, whereas an online forum simultaneously erupts in debate over similar ideas. They believe open dialogue fosters clarity, yet they recognize tensions may escalate. They intend to document their conclusions, hoping to contribute thoughtfully to the discussion."
    "axes": {
        "main_outcome_axis": "They are discussing a philosophical concept,",
        "intention_axis": "They intend to document their conclusions, hoping to contribute thoughtfully to the discussion.",
        "opinion_axis": "They believe open dialogue fosters clarity,",
        "hypothesis_axis": "",
        "generic_axis": "",
        "negation_axis": "",
        "static_axis": "whereas an online forum simultaneously erupts in debate over similar ideas. yet they recognize tensions may escalate.",
        "recurrent_axis": ""
    },
    "target_values": {
        "location": 39.865,
        "scale": 13.265,
        "skewness": 4.25
    }
}
```


Figure 1: Composition of samples in Chronocept benchmarks.

4.4 Inter-Annotator Agreement (IAA)

We evaluate Inter-Annotator Agreement (IAA) using stage-specific metrics aligned with each step of the annotation task. Segmentation quality is assessed using the P_k metric (Beeferman et al., 1997), axis categorization consistency is measured using the Jaccard Index, and agreement on the final temporal validity parameters (ξ, ω, α) is quantified using the Intraclass Correlation Coefficient (ICC).

We report only ICC as the benchmark-wide IAA, refraining from aggregating agreement across stages, as segmentation and axis categorization, while enriching the dataset structure, do not directly impact the core prediction task, which depends solely on the parent text and its annotated temporal validity distribution.

Agreement statistics across both benchmarks are summarized in Table 1. We observed notable confusion between the *Generic* and *Static* axes during the early stages of annotation, particularly in the warm-up phase. This source of disagreement is analyzed in detail in Appendix B.

IAA Metric	BI	BII
ICC	0.843	0.893
Jaccard Index	0.624	0.731
P_k **Metric**	0.233	0.009

Table 1: IAA metrics for segmentation, axis categorization, and temporal validity annotation across both benchmarks. For P_k, lower is better, with values ranging from 0 (perfect agreement) to 1 (chance-level).

4.5 Dataset Design

Each Chronocept sample captures the temporal dynamics of factual information through a structured annotation format, as illustrated in Figure 1.

Parent Text. A single sentence serving as the basis for annotation.

Temporal Axes. Each parent text is segmented into subtexts annotated along eight temporal axes:

- **Main:** Core verifiable events.
- **Intention:** Future plans or desires.
- **Opinion:** Subjective viewpoints.

- **Hypothetical:** Conditional or imagined events.
- **Negation:** Denied or unfulfilled events.
- **Generic:** Timeless truths or habitual patterns.
- **Static:** Unchanging states in context.
- **Recurrent:** Repeated temporal patterns.

Target Values. Temporal validity is quantified by three parameters:

- ξ **(Location):** The time point of peak validity.
- ω (Scale): The duration over which validity is maintained.
- α **(Skewness):** The asymmetry of the validity curve.

4.6 Dataset Statistics & Splits

Stratified sampling over the axes distribution was applied to partition the datasets into training (70%), validation (20%), and test (10%) splits, ensuring equitable axis coverage. Table 2 summarizes the splits for both benchmarks. The axes distribution, calculated based on non-null annotations for each sample, is detailed in Table 3.

Benchmark	Training	Validation	Test
Benchmark I	878	247	129
Benchmark II	365	104	55

Table 2: Dataset Composition and Splits.

Temporal Axis	Benchmark I	Benchmark II
Main Axis	1254	524
Static Axis	516	513
Generic Axis	228	116
Hypothetical Axis	136	182
Negation Axis	240	200
Intention Axis	165	522
Opinion Axis	328	519
Recurrent Axis	348	198

Table 3: Distribution of annotated temporal axes across Benchmark I and Benchmark II.

Token-level[3] and target parameter-level statistics for both benchmarks are summarized in Table 4 and Table 5.

[3]Tokenization performed using SpaCy's `en_core_web_sm` model: https://spacy.io/api/tokenizer

Benchmark	Mean Length (μ)	SD (σ)
Benchmark I	16.41 tokens	1.56 tokens
Benchmark II	56.21 tokens	6.21 tokens

Table 4: Sentence Length Statistics for Benchmarks.

4.7 Accessibility and Licensing

The Chronocept dataset is released under the Creative Commons Attribution 4.0 International (CC-BY 4.0)[4] license, allowing unrestricted use with proper attribution. It is publicly available on Hugging Face Datasets at: [redacted, disclosed as a zip file].

5 Baseline Model Performance

5.1 Task Scope and Evaluation Focus

Chronocept models temporal validity as a structured regression task over low-dimensional parameters: location (ξ), scale (ω), and skewness (α), predicted from annotated parent texts. Unlike prior work on event ordering (Pustejovsky, 2003), commonsense classification (Zhou et al., 2019), or temporal shift detection (Wenzel and Jatowt, 2024), segmentation and axis labels are treated as preprocessing and not modeled at inference.

Evaluation spans three dimensions: regression accuracy (MSE, MAE, R^2), calibration (Negative Log-Likelihood), and rank correlation (Spearman ρ). As the task involves parameter estimation rather than text generation, encoder-only models suffice. Decoder architectures are unnecessary, as Chronocept operates at the application layer, interfacing with downstream systems without altering core language models.

5.2 Baseline Models and Training Setup

We benchmark Chronocept against a representative set of baselines spanning statistical (LR, SVR), tree-based (XGB), and neural architectures (FFNN, Bi-LSTM, BERT Regressor). Each baseline is trained to jointly predict ξ, ω and α from BERT-based input embeddings of the parent text and temporal subtexts. Targets are Z-Score normalized to standardize learning across all models.

Hyperparameters for all baselines (except BERT) were tuned via grid search; final configurations are detailed in Appendix H. FFNN and Bi-LSTM models were trained for 100 epochs while BERT

[4]https://creativecommons.org/licenses/by/4.0

Parameter	Location (ξ)		Duration (ω)		Skewness (α)	
Benchmark	**Mean (μ)**	**SD (σ)**	**Mean (μ)**	**SD (σ)**	**Mean (μ)**	**SD (σ)**
Benchmark I	54.2803	20.4169	11.5474	3.7725	-0.0158	1.3858
Benchmark II	46.1511	13.3839	9.5553	2.5725	0.0275	1.1773

Table 5: Temporal Parameter Distribution Statistics for Benchmarks.

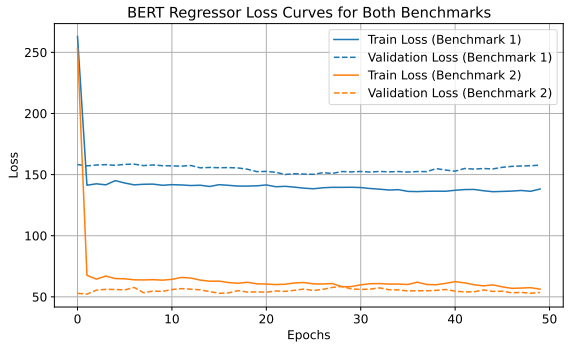


Figure 2: BERT training loss curves for Benchmark I and Benchmark II. The loss flatlined after 2 epochs for both benchmarks.

was trained for 50 epochs. BERT training loss plateaued after approximately 2 epochs across both benchmarks, as shown in Figure 2, suggesting early stopping could be beneficial for future experiments.

All training and inference experiments were conducted on a machine with an Intel Core i9-14900K CPU, 16GB DDR5 RAM, and an NVIDIA RTX 4060 GPU.

Baseline implementations and training scripts are publicly available at: https://anonymous.4open.science/r/chronocept-baseline-models-1EE1.

5.3 Quantitative Evaluation

Table 6 summarizes the performance of baseline models across both benchmarks. Each reported metric reflects the mean score across the three predicted parameters.

Feedforward Neural Networks (FFNN) outperform all other models overall, achieving the lowest MSE, MAE, NLL, and the highest Spearman Correlation on Benchmark I. This supports prior findings that simpler architectures, when paired with high-quality pretrained embeddings, can match or exceed deeper models in accuracy and efficiency (Saphra and Lopez, 2019; Wei et al., 2021).

Bi-LSTM trails FFNN on Benchmark I but outperforms it on Benchmark II in four of five metrics - MSE, R^2, NLL and Spearman ρ - on Benchmark II, which provides longer textual context. This is consistent with prior findings on sequence modeling (Meng and Rumshisky, 2018; Dligach et al., 2017), and may stem from Bi-LSTM's ability to better model long-range dependencies, while FFNNs rely on the BERT [CLS] token, which can struggle to encode longer contexts into a single vector (Li et al., 2020).

BERT Regression improves significantly from Benchmark I to II, with MSE dropping by over 50%, suggesting longer inputs help stabilize fine-tuning. However, BERT still underperforms across all metrics, consistent with its known sensitivity to overfitting and gradient noise on small regression datasets (Mosbach et al., 2021; Peters et al., 2019; Lee et al., 2020).

Among classical models, SVR and XGBoost perform reasonably but are outpaced by neural approaches. SVR achieves relatively strong R^2 and NLL scores on Benchmark I, while XGBoost and LR lag across all metrics. Their interpretability and training efficiency still make them useful reference baselines (Drucker et al., 1996; Rogers et al., 2020).

Together, these results affirm that pretrained embeddings paired with compact neural regressors like FFNN yield state-of-the-art performance. Additionally, they highlight how models with sequence-awareness, such as Bi-LSTM and BERT, benefit disproportionately from longer contexts.

5.4 Impact of Temporal Axes: Ablation Studies

To assess the utility of explicit temporal axes in Chronocept, we conduct two ablation studies on Benchmark 1 using the Bi-LSTM and FFNN baselines.

The first study evaluates the impact of removing all axis-level information, and the second examines the impact of randomly shuffling axis order during training. This setup parallels prior work on robustness testing via perturbed input labels (Moradi and

Metric	MSE		MAE		R^2		NLL		Spearman	
Baseline	BI	BII	BI	BII	BI	BII	BI	BII	BI	BII
LR	1.3610	1.1009	0.9179	0.8361	-0.3610	-0.1009	1.5730	1.4670	0.2338	0.3279
XGB	0.8884	0.9580	0.7424	0.8011	0.1116	0.0420	1.3598	1.3975	0.2940	0.2331
SVR	0.9067	0.8889	0.7529	0.7740	0.0933	0.1111	1.3700	1.3601	0.3281	0.3293
FFNN	**0.8763**	0.8715	**0.7284**	**0.7583**	**0.1237**	0.1285	**1.3529**	1.3502	**0.3543**	0.3437
Bi-LSTM	0.9203	**0.8702**	0.7571	0.7646	0.0797	**0.1298**	1.3774	**1.3494**	0.2367	**0.3535**
BERT	145.8611	68.1507	6.7570	4.6741	-0.0090	-0.1122	3.9103	3.5299	-0.0485	-0.2407

Table 6: Test set performance of baseline models for Benchmark I (BI) and Benchmark II (BII). Lower values for MSE, MAE, and NLL indicate better performance; higher R^2 and Spearman Correlation ρ denote improved fit.

Samwald, 2021).

Both the axis-removal and axis-shuffle setups lead to substantial performance degradation, indicating that both - the presence and consistent ordering of temporal axes - play a key role in accurately modeling temporal validity.

Table 7 summarizes the increase in MSE for the Bi-LSTM baseline. Experimental design and complete results for both baselines are detailed in Appendix F (excluded axes) and Appendix G (shuffled axes).

Ablation Type	Ablated MSE	Increase
Exclusion of Axes	0.9625	4.59%
Erroneous Labeling	1.0107	9.83%

Table 7: Ablation results for the Bi-LSTM baseline. Relative increases are computed over the original MSE of 0.9203.

6 Conclusion & Applications

We introduced Chronocept, a framework that models temporal validity as a continuous probability distribution using a unified, parameterized representation. By encoding validity through location (ξ), scale (ω), and skewness (α), Chronocept provides a generalizable mathematical scheme for temporal reasoning in language.

Through structured annotations and explicit temporal axes, Chronocept enables models to capture not just if, but when and for how long information remains valid - advancing beyond binary truth labels to a richer temporal understanding.

Empirical results highlight the effectiveness of simple neural models paired with pretrained embeddings, and ablation studies underscore the importance of structural consistency and axis-level decomposition.

Chronocept opens pathways for temporally aware applications, including retrieval-augmented generation (RAG), fact verification, knowledge lifecycle modeling, and proactive AI agents that act based on temporal salience (Miksik et al., 2020). All datasets, annotations, and baselines are publicly released to support continued research in this space.

7 Limitations

In this section, we highlight key limitations of Chronocept and suggest directions for future refinement and broader applicability.

Unimodal Temporal Representation. Chronocept models temporal validity with a unimodal, single-peaked distribution. While this ensures interpretability and efficient annotation, it cannot represent events with multiple distinct periods of relevance, such as seasonal or recurring phenomena.

Sentence-Level Context Only. The dataset consists of short, self-contained sentences without document-level or historical context. This limits the modeling of long-range temporal dependencies and evolving narratives, constraining discourse-level temporal reasoning.

No Atemporality Indicators. Chronocept lacks explicit labels for atemporal or universally valid facts, introducing ambiguity between permanently valid and time-sensitive information.

Minimum Validity Constraint from Log Time Scale. The logarithmic time scale imposes a lower bound of one minute, making it unsuitable for modeling events that become instantly obsolete, such as flash updates or ephemeral statements.

References

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

Axel Almquist and Adam Jatowt. 2019. Towards content expiry date determination: Predicting validity periods of sentences. pages 86–101.

A Azzalini. 1996. The multivariate skew-normal distribution. *Biometrika*, 83(4):715–726.

Adelchi Azzalini. 1986. A class of distributions which includes the normal ones. *Scandinavian Journal of Statistics*.

Doug Beeferman, Adam Berger, and John Lafferty. 1997. Text segmentation using exponential models. In *Second Conference on Empirical Methods in Natural Language Processing*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Gaurav Sastry, Amanda Askell, Ariel Agarwal, Shelly Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. 33:1877–1901.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering.

Supratim Das, Arunav Mishra, Klaus Berberich, and Vinay Setty. 2017. Estimating event focus time using neural word embeddings. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, New York, NY, USA. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186.

Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural temporal relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 746–751, Valencia, Spain. Association for Computational Linguistics.

Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. 1996. Support vector regression machines. 9.

Rhailana Fontes, Jéssica Ribeiro, Daya S Gupta, Dionis Machado, Fernando Lopes-Júnior, Francisco Magalhães, Victor Hugo Bastos, Kaline Rocha, Victor Marinho, Gildário Lima, Bruna Velasques, Pedro Ribeiro, Marco Orsini, Bruno Pessoa, Marco Antonio Araujo Leite, and Silmar Teixeira. 2016. Time perception mechanisms at central nervous system. *Neurol. Int.*, 8(1):5939.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. pages 770–778.

Taishi Hosokawa, Adam Jatowt, and Kazunari Sugiyama. 2023. Temporal natural language inference: Evidence-based evaluation of temporal text validity. In *Lecture Notes in Computer Science*, Lecture notes in computer science, pages 441–458. Springer Nature Switzerland, Cham.

Raghav Jain, Daivik Sojitra, Arkadeep Acharya, Sriparna Saha, Adam Jatowt, and Sandipan Dandapat. 2023. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6750–6774, Stroudsburg, PA, USA. Association for Computational Linguistics.

Adam Jatowt, Ching-Man Au Yeung, and Katsumi Tanaka. 2013. Estimating document focus time. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*, New York, New York, USA. ACM Press.

Nattiya Kanhabua and Kjetil Nørvåg. 2008. Improving temporal language models for determining time of non-timestamped documents. In *Research and Advanced Technology for Digital Libraries*, Lecture notes in computer science, pages 358–370. Springer Berlin Heidelberg, Berlin, Heidelberg.

Abhimanu Kumar, Jason Baldridge, Matthew Lease, and Joydeep Ghosh. 2012. Dating texts without explicit temporal cues. *arXiv [cs.CL]*.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. 2016. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A Smith. 2021. Time waits for no one! analysis and challenges of temporal misalignment. *arXiv [cs.CL]*.

Steven Lynden, Mehari Heilemariam, Kyoung-Sook Kim, Adam Jatowt, Akiyoshi Matono, Hai-Tao Yu, Xin Liu, and Yijun Duan. 2023. Commonsense temporal action knowledge (cotak) dataset. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM 2023*.

Yuanliang Meng and Anna Rumshisky. 2018. Context-aware neural model for temporal information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 527–536, Melbourne, Australia. Association for Computational Linguistics.

Ondrej Miksik, I Munasinghe, J Asensio-Cubero, S Reddy Bethi, ST Huang, S Zylfo, Xuechen Liu, T Nica, A Mitrocsak, S Mezza, et al. 2020. Building proactive voice assistants: When and how (not) to interact. *arXiv preprint arXiv:2005.01322*.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1558–1570, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. TORQUE: A reading comprehension dataset of temporal ordering questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328. Association for Computational Linguistics.

OpenAI. 2024. Openai o1 system card. *arXiv*.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

J Pustejovsky, Kiyong Lee, H Bunt, and Laurent Romary. 2010. ISO-TimeML: An international standard for semantic annotation. *LREC*, pages 394–397.

James Pustejovsky. 2003. The timebank corpus. *Corpus linguistics*.

James Pustejovsky, José M Castaño, Robert Ingria, and Graham Katz. 2003. TimeML: A specification language for temporal and event expressions.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *arXiv [cs.CL]*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexandra M Schmidt, Kelly C M Gonçalves, and Patrícia L Velozo. 2017. Spatiotemporal models for skewed processes. *Environmetrics*, 28(6):e2411.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. pages 231–235.

Hikaru Takemura and Keishi Tajima. 2012. Tweet classification based on their lifetime duration.

Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. 2012. TempEval-3: Evaluating events, time expressions, and temporal relations. *arXiv [cs.CL]*.

Marc Verhagen. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the fourth international workshop on semantic evaluations*.

Marc Verhagen. 2010. SemEval-2010 task 13: TempEval-2. In *Proceedings of the 5th international workshop on semantic evaluation*.

Colin Wei, Sang Michael Xie, and Tengyu Ma. 2021. Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. *Advances in Neural Information Processing Systems*, 34:16158–16170.

Haoyang Wen and Heng Ji. 2021. Utilizing relative event time to enhance event-event temporal relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Georg Wenzel and Adam Jatowt. 2023. An overview of temporal commonsense reasoning and acquisition. *arXiv [cs.AI]*.

Georg Wenzel and Adam Jatowt. 2024. Temporal validity change prediction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1424–1446, Bangkok, Thailand. Association for Computational Linguistics.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. 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 3363–3369, Stroudsburg, PA, USA. Association for Computational Linguistics.

Appendix

A Annotation Guidelines

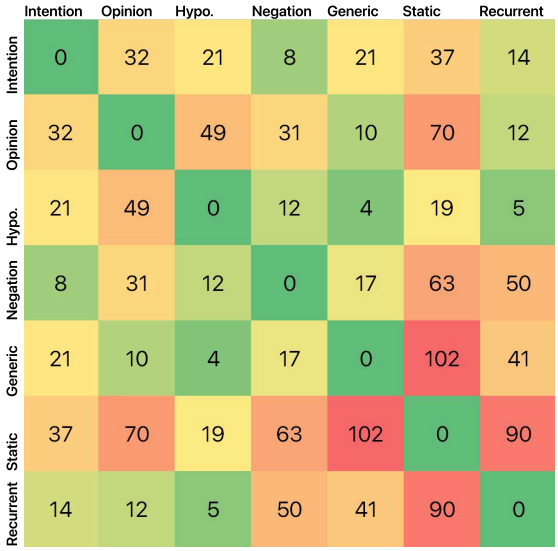
This section outlines the annotation guidelines used in the Chronocept dataset. These were introduced through an in-person training session and remained accessible throughout the annotation phase via a custom Streamlit-based interface for annotations[5]. The guidelines provide precise instructions for temporal segmentation, axis categorization, and temporal validity distribution plotting, supplemented with definitions, examples, and coverage of edge cases for all eight temporal axes.

During the initial warm-up phase, annotators exhibited substantial confusion between the Generic and Static axes. To mitigate this, the guidelines were revised to incorporate clearer contextual definitions and axis-specific "key questions" designed to improve disambiguation. These revisions led to a marked improvement in inter annotator agreement.

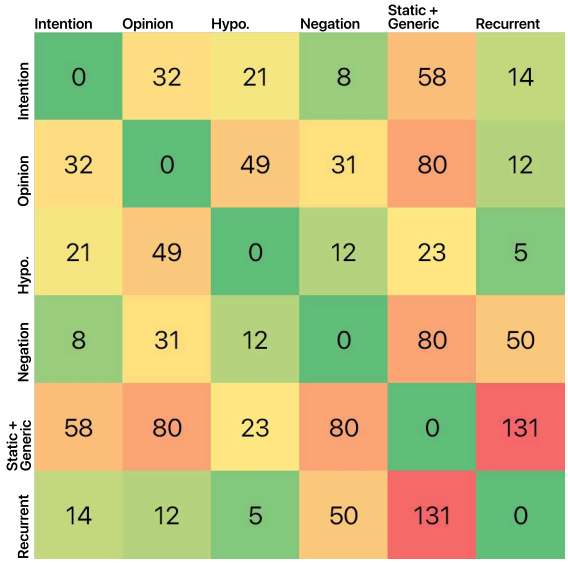
The complete guidelines are shown in Figure 3.

[5] https://streamlit.io

B Axis Confusion Analysis: Generic and Static



(a) Axis assignment co-occurrence matrix with Generic and Static treated as distinct classes



(b) Axis assignment co-occurrence matrix after merging Generic and Static into a unified class

Figure 4: Comparison of co-occurrence matrices before and after merging the Generic and Static axes, used to assess annotation consistency.

This appendix investigates a key source of annotator disagreement in the Chronocept annotation process: the difficulty in consistently distinguishing between the Generic and Static temporal axes.

Generic segments typically express habitual or timeless statements, while Static segments describe ongoing but context-specific states. Their semantic similarity led to frequent disagreement in axis

assignment.

To address this, the annotation guidelines were refined during the warm-up phase with axis-specific clarifications and diagnostic questions. The guideline clarification led to reduced confusion, as shown in the co-occurrence matrices in Figure 4.

While co-occurrence matrices are traditionally used to analyze disagreement patterns between annotators, we treat them here as confusion matrices by including agreement counts along the diagonal, enabling standard metric computation.

To quantify the benefit of merging these axes, we computed micro-averaged inter-annotator precision. Treating this as a multi-class classification task, we additionally calculate Cohen's Kappa to assess inter-annotator agreement beyond chance. As shown in Table 8, merging resulted in a consistent improvement across both metrics: precision improved by 18.0% and Cohen's Kappa by 17.47%.

Axis Setting	Precision	Cohen's Kappa
Original	0.4443	0.3291
Merged	0.5243	0.3866

Table 8: Improvement in annotator alignment metrics after merging Generic and Static into a single class.

C Time Scale Logarithm Base Conversion

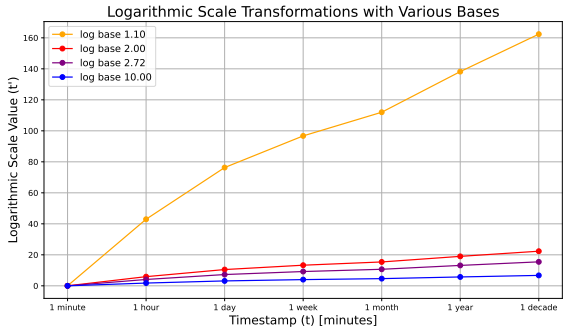


Figure 5: Effect of logarithmic base choice on time axis representation. Base 1.1 preserves quasi-linear spacing; larger bases induce stronger compression.

Chronocept represents time on a logarithmic axis to unify short- and long-term temporal dynamics in a compact space. The transformation is defined over a configurable base b; all released datasets use base 1.1. A reusable `DataLoader` with log conversion is available in the official baselines repository[6].

Log Transformation. Given time t in minutes, the log-space representation is:

$$t' = \frac{\ln(t)}{\ln(b)}.$$

Base 1.1 yields quasi-linear spacing across intervals like hours, days, and years, preserving interpretability. Figure 5 shows that higher bases increasingly compress longer intervals, while base 1.1 maintains resolution across scales.

Compression Analysis. Table 9 summarizes the compression effect across bases 1.1, 2, and 10. For each timestamp, we report the log value t', compression ratio $\text{CR} = t'/t$, and percentage compression.

To convert values between log bases m and b:

$$t'^{(b)} = \frac{\ln(m)}{\ln(b)} \cdot t'^{(m)}.$$

Skew-Normal Parameter Adjustment. Chronocept models temporal validity using a skew-normal distribution:

$$f(x;\, \xi, \omega, \alpha) = \frac{2}{\omega}\, \phi\left(\frac{x-\xi}{\omega}\right)\, \Phi\left(\alpha\, \frac{x-\xi}{\omega}\right),$$

where ξ and ω denote location and scale. When converting between bases:

$$\xi^{(b)} = \frac{\ln(m)}{\ln(b)} \cdot \xi^{(m)}, \quad \omega^{(b)} = \frac{\ln(m)}{\ln(b)} \cdot \omega^{(m)}.$$

Skewness α remains invariant.

D Comparison of Distributions for Modeling Temporal Validity and Curve Fitting Methodology

This section evaluates candidate distributions for modeling temporal validity and outlines the curve fitting methodology. We consider six synthetic, unimodal scenarios varying along three axes: *offset* (peak position), *duration* (span of validity), and *asymmetry* (skew in rise and decay). Table 10 lists a representative sentence and five annotation points per scenario, placed on a base-1.1 logarithmic time axis.

Each temporal profile is defined by a smooth freehand curve from which five points are sampled—one at the peak, two mid-validity, and two

[6] https://anonymous.4open.science/r/chronocept-baseline-models-1EE1

		log base 1.1			log base 2			log base 10		
Timestamp	**Linear (t)**	t'	**CR**	**%**	t'	**CR**	**%**	t'	**CR**	**%**
1 minute	1	0.0	0.000	100	0.0	0.000	100	0.0	0.000	100
1 hour	60	42.96	0.716	28.4	5.91	0.099	90.1	1.78	0.030	97.0
1 day	1440	76.30	0.053	94.7	10.47	0.007	99.3	3.16	0.002	99.8
1 week	10080	96.73	0.010	99.0	13.30	0.001	99.9	4.00	3.968e-4	99.9
1 month	43200	111.97	0.003	99.7	15.39	3.563e-4	99.9	4.63	1.072e-4	~100
1 year	525600	138.23	2.623e-4	~100	19.00	3.615e-5	~100	5.72	1.088e-5	~100
1 decade	5256000	162.25	3.087e-5	~100	22.33	4.249e-6	~100	6.72	1.279e-6	~100

Table 9: Compression analysis across logarithmic bases. CR = t'/t, Compression % = $100 \times (1 - \text{CR})$.

low-validity points. These define a proportional shape used for fitting.

Since these curves represent relative probabilities, their area under the curve (AUC) is unconstrained. During optimization, a scaling factor is applied to fit freely, followed by Trapezoidal Rule normalization to enforce AUC = 1 while preserving shape.

To reduce computational overhead over long-tailed domains, we recommend rescaling the fitted curve by its maximum value to constrain it to $[0, 1]$. This avoids instability from very small values in AUC-normalized densities. The result, while no longer a true probability distribution, retains shape and relative comparisons. We refer to it as a *proportional validity curve*, useful in applications prioritizing ranking or visualization over strict probabilistic semantics.

Candidate distributions include:

Gaussian Normal:

$$f_{Gaussian}(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Exponential:

$$f_{Exp}(x; \lambda) = \lambda \exp(-\lambda x), \text{where } x \geq 0$$

Log-normal:

$$f_{LN}(x; \mu, \sigma) = \frac{1}{x\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right),$$

where $x > 0$

Gamma:

$$f_{\Gamma}(x; k, \theta) = \frac{1}{\Gamma(k)\,\theta^k} x^{k-1} \exp\left(-\frac{x}{\theta}\right),$$

where $x > 0$

Skewed Normal:

$$f_{SN}(x; \xi, \omega, \alpha) = \frac{2}{\omega}\,\phi\left(\frac{x-\xi}{\omega}\right)\,\Phi\left(\alpha\,\frac{x-\xi}{\omega}\right)$$

where $\phi(z)$ is the standard normal PDF and $\Phi(z)$ is the standard normal CDF.

Optimization: Parameter estimation is performed using the Trust Region Reflective (TRF) algorithm by minimizing the sum of squared residuals:

$$SSR(\theta) = \sum_{i=1}^{N} (y_i - f(x_i; \theta))^2$$

This is implemented via `scipy.optimize.curve_fit`[7]. After optimization, we compute:

$$N = \int_{x_{\min}}^{x_{\max}} f_{\text{fit}}(x)\,dx,$$

$$f_{\text{norm}}(x) = \frac{f_{\text{fit}}(x)}{N}, \quad f_{\max} = \max_x f_{\text{norm}}(x),$$

$$S_{\text{final}} = \frac{S_{\text{fit}}}{N \cdot f_{\max}}$$

Evaluation: RMSE is used as the primary goodness-of-fit metric. As a scale-sensitive measure that penalizes large deviations, a lower RMSE indicates superior fit quality.

Table 10 and Figure 6 present the six scenarios, annotation points, and corresponding fitted curves. Table 11 reports RMSE for each candidate distribution across scenarios. The skew-normal consistently yields the lowest RMSE, confirming its suitability for modeling asymmetric and variable-duration temporal profiles.

[7] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html

Temporal Scenario	Sample Sentence	Annotation Points (x, y)
S1: Early Onset	"He is making coffee for himself right now."	(14.91, 0.19), (21.64, 0.41), (27.64, 0.77), (31.64, 0.41), (34.91, 0.20)
S2: Late Onset	"The movie is going to hit the theaters in a few weeks."	(93.75, 0.21), (100.67, 0.80), (106.57, 0.42), (112.73, 0.20), (98.0, 0.63)
S3: Short Duration	"The site has been crashing for a few minutes as there is some server maintenance work going on."	(12.73, 0.21), (28.19, 0.80), (41.28, 0.20), (32.19, 0.60), (18.91, 0.40)
S4: Long Duration	"The ruling government brings growth and progress."	(1, 0.05), (130.38, 0.81), (147.84, 0.21), (111.29, 0.42), (138.38, 0.60)
S5: Rapid Rise, Slow Decay	"The advertisement's impact peaks immediately and lingers."	(42.73, 0.21), (46.91, 0.40), (53.10, 0.80), (63.46, 0.56), (81.83, 0.27)
S6: Slow Rise, Rapid Decay	"The news slowly gains attention but quickly becomes outdated."	(43.28, 0.20), (58.01, 0.40), (76.92, 0.79), (84.92, 0.40), (88.92, 0.17)

Table 10: Six temporal scenarios illustrating the effects of offset, duration, and asymmetry. Each scenario is represented by 5 annotation points on a log-transformed time axis with base 1.1.

E Synthetic Generation of Samples

This section presents the plaintext markdown prompts used for synthetic dataset generation in Chronocept via the GPT-o1 model (OpenAI, 2024). The prompts are designed to yield syntactically coherent text with explicit temporal structure. Generation was performed in batches of 50 samples per prompt.

The prompts are shown in Figure 7 for Benchmark-I and Figure 8 for Benchmark-II.

F Ablation Study: Impact of Structured Temporal Axes on Model Performance

To evaluate the contribution of multi-axis temporal annotations in modeling temporal validity, we conduct an ablation study on the Bi-LSTM and FFNN baselines. Specifically, we assess the effect of removing structured temporal axes from the model input.

Input Construction. Each example in Chronocept is annotated along multiple temporal axes. In the standard setup, axis-specific embeddings are concatenated in a fixed order to the embedding of the parent text, forming a structured input representation. The ablation removes these axis embeddings, retaining only the parent text embedding.

Setup. We compare the two configurations (with and without axis embeddings) using Bi-LSTM and FFNN models on Benchmark I. Both models are trained to predict the parameters ξ, ω, and α of the skew-normal temporal validity distribution. Evaluation is performed using MSE, MAE, R^2, NLL, and CRPS.

Results. Table 12 reports the results for both models. Including axis embeddings reduces Bi-LSTM MSE by 4.6% and boosts R^2 by 112%, confirming that structured cues matter more for goodness-of-fit than for absolute error. FFNN sees a 6.9% MSE drop and a 95.7% gain in R^2, exhibiting a similar trend with even greater error reduction across all metrics.

These findings are consistent with prior work showing that compositional and auxiliary structure improves model generalization and fit across tasks (Lake and Baroni, 2018; Søgaard and Goldberg, 2016).

Distribution	S1	S2	S3	S4	S5	S6	Parameters
Gaussian	0.0709	0.0673	0.0424	0.0273	0.1193	0.0806	$(\mu,\ \sigma)$
Exponential	0.2103	0.2291	0.2312	0.2704	0.2126	0.2212	(λ)
Log-normal	0.0844	0.0597	0.0804	0.0325	0.0872	0.0919	$(\mu,\ \sigma)$
Gamma	0.0827	0.0623	0.0668	0.0307	0.0968	0.0899	$(k,\ \theta)$
Skewed Normal	**0.0514**	**0.0357**	**0.0407**	**0.0224**	**0.0505**	**0.0247**	$(\xi,\ \omega,\ \alpha)$

Table 11: Average RMSE values for candidate distributions across six temporal scenarios. All distributions were fitted using a scaling factor S to enforce AUC $= 1$. A lower RMSE indicates a better fit, as RMSE heavily penalizes large errors due to squaring, is scale-dependent, and more sensitive to outliers.

Model	Setting	MSE	MAE	R^2	NLL	CRPS
Bi-LSTM	Without Axes	0.9625	0.7659	0.0375	1.3998	0.7659
	Absolute Change (Δ)	0.0422	0.0088	0.0422	0.0224	0.0088
	Improvement	4.59%	1.16%	112.53%	1.63%	1.16%
FFNN	Without Axes	0.9368	0.7531	0.0632	1.3863	0.7531
	Absolute Change (Δ)	0.0605	0.0247	0.0605	0.0334	0.0247
	Improvement	6.91%	3.39%	95.71%	2.47%	3.39%

Table 12: Ablation results on Benchmark I for Bi-LSTM and FFNN with axis embeddings removed. "Absolute Change" rows show differences from the original metrics in Table 6.

Conclusion. Structured axis embeddings improve performance across both architectures, particularly in R^2, which nearly doubles, indicating better distributional alignment. These results validate Chronocept's use of explicit temporal structure and are consistent with prior work on structured auxiliary signals.

G Ablation Study: Impact of Incorrect Temporal Axes Labeling

We evaluate the sensitivity of temporal validity modeling to erroneous axis labelling by conducting an ablation on FFNN and Bi-LSTM baselines. Specifically, we shuffle the order of temporal axis embeddings during training while preserving correct ordering in the test set.

Setup. In Chronocept, input representations are formed by concatenating temporal axis embeddings in a fixed sequence with the parent text embedding. This ablation introduces erroneous axis labelling by disrupting the axis order during training, thereby breaking the structural alignment. The evaluation set remains unperturbed. Models are trained to predict skew-normal parameters ξ, ω, and α, and evaluated on Benchmark I using MSE, MAE, R^2, NLL, and CRPS.

Results. Table 13 shows that misaligned axis ordering during training degrades performance significantly. Bi-LSTM MSE increases by 9.81% and R^2 decreases by 113.43%; FFNN sees a 13.36% MSE increase and 94.58% R^2 decrease. These results suggest that disrupting structural alignment introduces inductive noise, echoing prior findings on the role of compositional structure (Lake and Baroni, 2018) and input robustness (Moradi and Samwald, 2021). The pronounced drop in R^2 highlights that axis ordering is critical for fit quality.

Conclusion. Erroneous axis labelling during training leads to statistically significant drops in performance, particularly in R^2, highlighting the importance of Chronocept's structured multi-axis representation for accurate temporal modeling.

H Hyperparameter Search and Final Baseline Configurations

All baseline models were tuned via grid search on the validation split of each benchmark. All neural models except BERT were trained for 100 epochs, with early stopping applied based on validation loss when applicable. BERT was trained for 50 epochs. Final hyperparameters are summarized below.

Support Vector Regression (SVR). We searched over $C \in \{0.1, 1, 10\}$, $\varepsilon \in \{0.01, 0.1, 1\}$,

Model	Setting	MSE	MAE	R^2	NLL	CRPS
Bi-LSTM	Erroneous Axes	1.0107	0.7984	-0.0107	1.4243	0.7984
	Absolute Change (Δ)	0.0904	0.0413	-0.0904	0.0469	0.0413
	Performance Drop	9.81%	5.46%	113.43%	3.40%	5.46%
FFNN	Erroneous Axes	0.9933	0.7591	0.0067	1.4156	0.7591
	Absolute Change (Δ)	0.1170	0.0307	-0.1170	0.0627	0.0307
	Performance Drop	13.36%	4.21%	94.58%	4.63%	4.21%

Table 13: Ablation results on Benchmark I for Bi-LSTM and FFNN under erroneous temporal axis labelling during training. "Absolute Change" rows show differences from the original metrics in Table 6.

and kernel type $\in \{linear, rbf\}$. The optimal setting used an RBF kernel with $C = 1$ and $\varepsilon = 1$ (see Table 14).

Benchmark	C	ε	Kernel
Benchmark I	1	1	rbf
Benchmark II	1	1	rbf

Table 14: Final SVR hyperparameters.

Linear Regression (LR). The grid search over `fit_intercept` $\in \{True, False\}$ selected *False* in both cases (see Table 15).

Benchmark	Fit Intercept
Benchmark I	False
Benchmark II	False

Table 15: Final Linear Regression setting.

XGBoost (XGB). We tuned $n\_estimators \in \{50, 100\}$, $max\_depth \in \{3, 5\}$, and learning rate $\in \{0.1, 0.01\}$. The best configuration used 50 estimators, depth 3, and learning rate 0.1 (see Table 16).

Benchmark	n	Depth	Learning Rate
Benchmark I	50	3	0.1
Benchmark II	50	3	0.1

Table 16: Final XGBoost hyperparameters.

Feedforward Neural Network (FFNN). We searched over hidden size $\in \{64, 128, 256\}$, dropout $\in \{0.0, 0.2, 0.5\}$, learning rate $\in \{0.01, 0.001, 0.0001\}$, L1 regularization $\in \{0.0, 0.0001, 0.001\}$, and weight decay $\in \{0.0, 0.001, 0.01\}$. Final settings differed between benchmarks (see Table 17).

Benchmark	Hidden Dim	Learning Rate
Benchmark I	64	0.001
Benchmark II	256	0.01

Table 17: Final FFNN hyperparameters. Other parameters were fixed at: dropout = 0.0, L1 = 0.001, weight decay = 0.0.

Bidirectional LSTM (Bi-LSTM). Search space included hidden size $\in \{64, 128, 256\}$ and learning rate $\in \{0.01, 0.001, 0.0001\}$. The final configuration used hidden size 64 and learning rate 0.0001 (see Table 18).

Benchmark	Hidden Dim	Learning Rate
Benchmark I	64	0.0001
Benchmark II	64	0.0001

Table 18: Final Bi-LSTM hyperparameters.

BERT Regression. We tuned dropout $\in \{0.0, 0.2, 0.4\}$ and learning rate $\in \{0.0001\}$. The best setting used no dropout and learning rate 0.0001. Training loss converged within 2 epochs on both benchmarks (see Figure 2).

All scripts used for hyperparameter search and training are disclosed (see footnote 6).

Annotation Guidelines for Chronocept

This document provides instructions for annotating temporal validity using a **three-step process**: **Text Splitting**, **Axis Assignment**, and **Temporal Validity Distribution Plotting**.
These guidelines are tailored to the nature of this benchmark, which typically involves one **Main Axis** segment and one additional axis segment from the seven auxiliary axes.

Step 1: Text Splitting

Objective:
Divide the input sentence into grammatically correct segments, ensuring semantic and temporal integrity is preserved.

Guidelines:
1. **Identify Splitting Points:**
   - Divide the sentence into meaningful subtexts. Most samples will include one **Main Axis** segment and one from the other seven axes.
   - Use punctuation and conjunctions as natural delimiters but ensure that each subtext is self-contained.
2. **Preserve Temporal Context:**
   - Retain essential markers (e.g., *continuously*, *in 2023*, *every month*).
   - Avoid removing or altering any text.
3. **Avoid Over-Splitting:**
   - Ensure each subtext conveys clear, standalone meaning.
   - Over-splitting may lead to fragments that lose context or temporal clarity.
4. **Text Copying Convention:**
   - Copy text exactly as it appears in the sample, including punctuation.
5. **Example:**
   - Input: *"The company is expanding its operations in Asia, and the CEO is leading the efforts, planning a significant increase in market share."*
   - Split:
     - Subtext 1: "The company is expanding its operations in Asia," (Main Axis)
     - Subtext 2: "and the CEO is leading the efforts, planning a significant increase in market share." (Intention Axis)
6. **Ambiguity Handling:**
   - If a sample seems to violate the condition of one Main Axis plus one other axis, document the **Sample ID** and consult **[redacted]**.
   - If a sample does not carry a Main Axis with a clearly definable temporal cue, document the **Sample ID** and consult **[redacted]**.
   - Incorrect samples will be discarded.

Step 2: Axis Assignment

Objective:
Classify each subtext into one of the **seven temporal axes** based on its primary temporal characteristic.
Temporal Axes:
1. **Main Axis (Factual Events):**
   - **Definition**: Verifiable events along a timeline, representing objective truths.
   - **Purpose**: Captures the primary narrative and establishes a concrete temporal sequence.
   - **Example**: "The company is expanding its operations in Asia."
   - **Key Question**: Does this event occur within the primary timeline of the narrative?
2. **Intention Axis:**
   - **Definition**: Captures someone's intention, desire, or plan, even if unfulfilled.
   - **Purpose**: Highlights future-directed actions or goals tied to the narrative but not necessarily realized.
   - **Example**: "The CEO is leading the efforts, planning a significant increase in market share."
   - **Key Question**: Is this event stated as an intended action or goal, regardless of its realization?
3. **Opinion Axis:**
   - **Definition**: Represents subjective viewpoints, expectations, or beliefs about events.
   - **Purpose**: Differentiates opinions or speculations from factual occurrences.
   - **Example**: "Experts believe the market will grow rapidly."
   - **Key Question**: Does this event represent a belief or expectation rather than a verified fact?
4. **Hypothetical Axis:**
   - **Definition**: Includes conditional or hypothetical events dependent on certain conditions.
   - **Purpose**: Tracks scenarios that are imagined or conditional, often using "if" statements.
   - **Example**: "If the company secures funding, it will expand globally."
   - **Key Question**: Is this event presented as dependent on another event or condition?
5. **Negation Axis:**
   - **Definition**: Identifies events explicitly stated as not occurring.
   - **Purpose**: Tracks denied actions or outcomes to separate them from realized events.
   - **Example**: "The company did not expand its operations in 2020."
   - **Key Question**: Is this event explicitly stated as unfulfilled or negated?
6. **Generic Axis:**
   - **Definition**: Represents universal truths or habitual occurrences, not tied to a specific timeline.
   - **Purpose**: Highlights timeless facts or generalizations applicable broadly.
   - **Example**: "Lions eat meat."
   - **Key Question**: Is this event a universal truth or a habitual occurrence that transcends specific contexts?
7. **Static Axis:**
   - **Definition**: Captures unchanging states or conditions **within a specific context or timeframe**.
   - **Purpose**: Tracks context-dependent facts or conditions relevant to the narrative.
   - **Example**: "The room is cold."
   - **Key Question**: Is this event context-specific and static within the described situation?
8. **Recurrent Axis:**
   - **Definition**: Describes events or states that happen repeatedly over time.
   - **Purpose**: Tracks patterns or cycles of actions/events relevant to the narrative.
   - **Example**: "The train arrives every morning at 8 AM."
   - **Key Question**: Does this event represent a recurring action or pattern?

Guidelines:
1. **Assign to the Closest Axis:**
   - Carefully analyze the temporal and semantic meaning of the subtext.
   - Decide if the event can be anchored to a specific axis based on its nature.
   - Most samples will have one **Main Axis** subtext and one auxiliary axis subtext.
2. **Handle Ambiguities:**
   - Focus on the start-points of events to reduce ambiguity related to durations.
   - Only compare events on the same axis; cross-axis relations require separate investigation.
   - If unsure about the axis, document the **Sample ID** and consult **[redacted]**.
   - Incorrect samples will be removed from the dataset.
3. **Use Context:**
   - Assess the broader context to distinguish between axes like Static and Generic.
4. **Example Annotation:**
   - Subtext: "The CEO is leading the efforts, planning a significant increase in market share."
   - Assigned Axis: **Intention Axis**
5. **Advisory for Complex Cases:**
   - Consider the following example: "The printer is making strange noises while the IT technician tries to fix it."
     - "The IT technician is trying to fix the printer" can be treated as the **Main Axis**, while "the printer is making strange noises" can be assigned to the **Generic Axis**.
     - This requires thoughtful analysis, as the roles of subtexts may not be apparent immediately. Annotators should carefully consider such cases, akin to transposing the segments for clarity.

Step 3: Temporal Validity Distribution Plotting

Objective:
Plot a skewed probability distribution over a **time graph** to represent the temporal validity of each subtext.

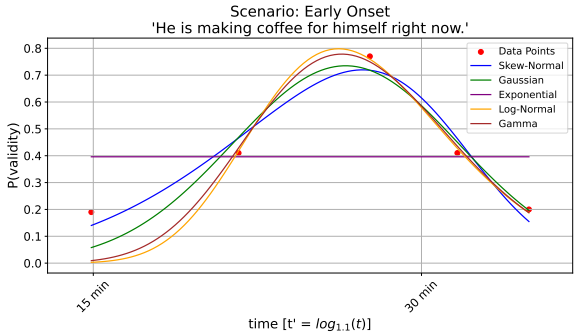
Guidelines:
1. **Temporal Cue Assignment:**
   - For samples with clear temporal cues (e.g., "solving for 1 hour"), assign a time interval to that cue. As an advisory, consider that a vernacularly used "1 hour" can range from 45 minutes to 90 minutes.
2. **Graph Axes:**
   - **X-Axis (Time):**
     - Labeled with intervals: 1 minute, 15 minutes, 30 minutes, 1 hour, 12 hours, 1 day, 1 week, 1 month, 1 year, 1 decade, and infinite validity.
   - **Y-Axis (Probability):**
     - Range: 0 (not valid) to 1 (fully valid).
3. **Plotting Points:**
   - Place 3–5 points on the timeline to indicate the probability of validity at specific times.
   - The user need not worry about making an ideal probability distribution with **AUC = 1**. Instead, plot proportions relative to the temporal "point" with the highest probability (Maximum Likelihood Estimate, MLE).
4. **Fit a Skewed Probability Distribution:**
   - A skewed curve will be automatically fitted through the plotted points to represent the temporal validity distribution.
5. **Consistency:**
   - Maintain consistency in plotting for similar subtexts.
6. **Ambiguity Handling:**
   - If the sample is technically correct but you are highly unsure about the temporal interval, annotate to the best of your ability. Low inter-annotator agreement (IAA) samples will be flagged and eliminated during post-processing.
   - If unsure about the distribution, document the **Sample ID** and consult **[redacted]**.
   - Incorrect samples will be removed.

The result of this step is a skewed probability distribution reflecting the temporal validity over time.

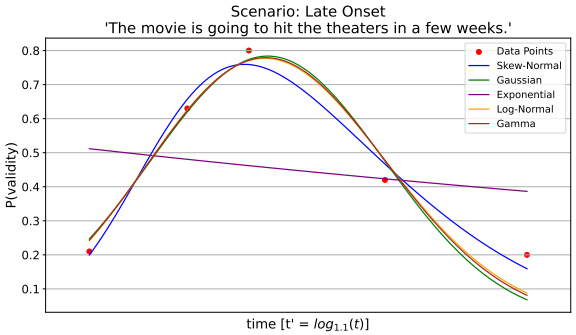
General Notes for Annotators

1. **Ambiguities:**
   - For unclear splits, axis assignments, or validity distributions, contact **[redacted]** with the **Sample ID** for resolution.
2. **Discarding Samples:**
   - Multimodal samples or those with excessive ambiguity should be flagged for review and potential removal.
3. **Temporal Objectivity:**
   - Avoid consulting peers during annotation to maintain objectivity and ensure consistency across annotators.
4. **Quality Control:**
   - Ensure all annotations are thorough, consistent, and adhere to these guidelines.

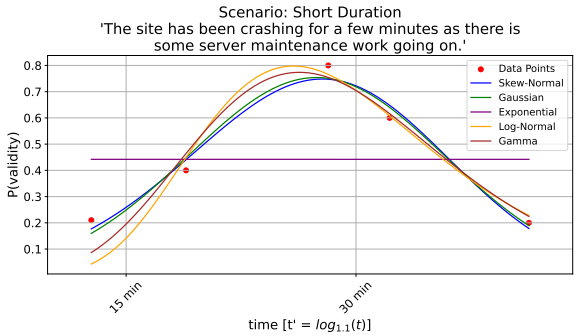
Figure 3: Annotation guidelines for Chronocept.



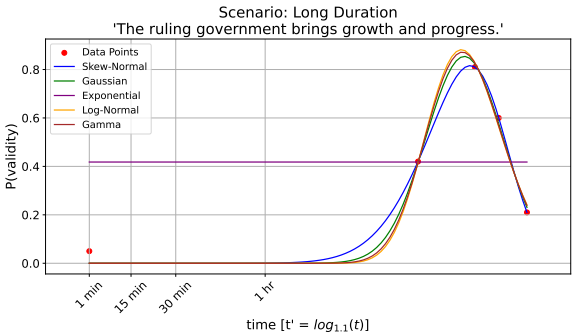
(a) Early Onset: Peak validity occurs soon after publication.



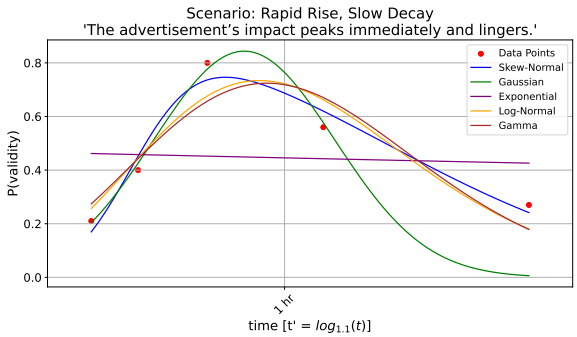
(b) Late Onset: Validity emerges gradually and peaks later.



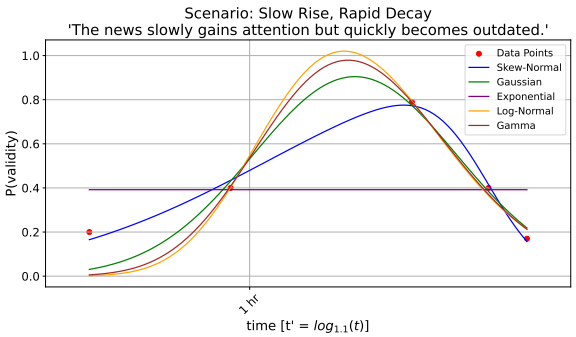
(c) Short Duration: A narrow window of high relevance.



(d) Long Duration: Validity persists over time.



(e) Rapid Rise, Slow Decay: Sudden onset, gradual decline.



(f) Slow Rise, Rapid Decay: Gradual onset, sharp drop.

Figure 6: Visual fit comparison of candidate distributions across six temporal scenarios. The skew-normal consistently provides the best fit, modeling varied validity patterns in onset, duration, and asymmetry.

Synthetic Data Generation for a Temporal Validity Benchmark

Objective
This task involves creating synthetic sentences that will form the basis of a benchmark for temporal validity research. Your role as a text generation model is to produce *high-quality sentences only*, without accompanying explanations or axis descriptions. These sentences should describe occurrences or events that happen simultaneously or contrastively, incorporating various actions, states, or processes.

Key Definition: Axis
An axis represents a semantic dimension or characteristic used to classify and analyze the relationships between events in a sentence. Axes are categorized into two types:
1. **Event-Related Axes**: Describe the relationship between events or states in a sentence, focusing on interactions or dependencies.
2. **Annotation Axes**: Provide supplementary semantic information about the events, enhancing interpretability.

Event-Related Axes
Specify the relationship between events in the sentence:
1. **Temporal Overlap**: Events occur simultaneously or in parallel.
2. **Causality**: One event causes or results from the other.
3. **Subordination**: One event depends on or occurs due to the other.
4. **Unrelated**: Events are independent of each other.

Annotation Axes
Provide semantic context and additional dimensions of meaning:
1. **Main Axis (Factual Events)**: Verifiable, objective events tied to a specific timeline.
2. **Intention**: Future-directed plans, desires, or actions.
3. **Opinion**: Subjective beliefs or expectations about events.
4. **Hypothetical**: Conditional or imagined scenarios.
5. **Negation**: Explicitly unfulfilled or denied actions or outcomes.
6. **Generic**: Universal truths or habitual actions that apply broadly across contexts and are not tied to specific timelines.
7. **Static**: Unchanging states or conditions that are specific to a particular context or timeframe.
8. **Recurrent**: Events or states that recur over time, forming patterns or cycles.

Guidelines for Sentence Generation

Sentence Structure
- Sentences should be written in the *present tense*. Use ***all forms of present tense*** - Simple Present Tense, Present Continuous Tense, Present Perfect Tense and Present Perfect Continuous Tense.
- Each sentence should incorporate:
  - *At least one Event-Related Axis* to define the relationship between events.
  - *Two Annotation Axes, one of which must be the ***Main Axis (Factual Events)****.

Neutrality and Diversity
- Sentences must span *diverse domains*, including daily life, technology, abstract concepts, and nature.
- Use a mix of *pronouns* ("he," "she," "they"), *generic entities* (e.g., "a person," "a machine"), and *articles* ("the," "a"). Ensure pronouns are evenly distributed across the dataset to represent diverse actors.

Task Output
1. Generate *50 sentences* adhering strictly to the above structure and requirements.
2. Ensure diversity in domains, axes, and event relationships while maintaining clarity and coherence.
3. Each sentence must explicitly include:
   - ***At least one Event-Related Axis***.
   - ***Two Annotation Axes***, with the *Main Axis (Factual Events)* included.

Examples of Correct Sentences
1. "She is cooking dinner, but the oven keeps malfunctioning."
2. "He is driving to work, while the traffic jam is worsening."
3. "They are reviewing documents, as the deadline approaches."
4. "A researcher is designing an experiment, while the technician prepares the equipment."
5. "The sky is darkening, but the lake remains calm and still."
6. "A student is reading the manual to understand how the device might operate."
7. "She is negotiating a contract, while her team finalizes the presentation."
8. "The clouds are gathering, and the wind is picking up speed."
9. "The robot is performing a task, while the operator monitors its efficiency."
10. "He is practicing the piano, but the audience remains silent."

Figure 7: Plaintext markdown prompt for Benchmark I.

Synthetic Data Generation for a Temporal Validity Benchmark

Objective
Your role as a text generation model is to produce *high-quality, coherent, and naturally flowing sentences or short paragraphs*, without accompanying explanations or axis descriptions. These samples should describe occurrences or events that happen simultaneously or contrastively, incorporating various actions, states, or processes. Avoid unnatural, overly formal, or stilted constructions.

Key Definition: Axis
An axis represents a semantic dimension or characteristic used to classify and analyze the relationships between events in a sentence. Axes are categorized into two types:
1. **Event-Related Axes**: Describe the relationship between events or states in a sentence, focusing on interactions or dependencies.
2. **Annotation Axes**: Provide supplementary semantic information about the events, enhancing interpretability.

Event-Related Axes
Specify the relationship between events in the sentence:
1. **Temporal Overlap**: Events occur simultaneously or in parallel.
2. **Causality**: One event causes or results from the other.
3. **Subordination**: One event depends on or occurs due to the other.
4. **Unrelated**: Events are independent of each other.

Annotation Axes
Provide semantic context and additional dimensions of meaning:
1. **Main Axis (Factual Events)**: Verifiable, objective events tied to a specific timeline.
2. **Intention**: Future-directed plans, desires, or actions.
3. **Opinion**: Subjective beliefs or expectations about events.
4. **Hypothetical**: Conditional or imagined scenarios.
5. **Negation**: Explicitly unfulfilled or denied actions or outcomes.
6. **Generic**: Universal truths or habitual actions that apply broadly across contexts and are not tied to specific timelines.
7. **Static**: Unchanging states or conditions that are specific to a particular context or timeframe.
8. **Recurrent**: Events or states that recur over time, forming patterns or cycles.

Guidelines for Sentence Generation

Sentence Structure
- Sentences should be written in the *present tense*. Use **all forms of present tense** - Simple Present Tense, Present Continuous Tense, Present Perfect Tense and Present Perfect Continuous Tense.
- Each sentence should incorporate:
  - *At least two Event-Related Axes* to define the relationship between events.
  - *Four or more Annotation Axes*, one of which must be the **Main Axis (Factual Events)**.
- Avoid overusing commas. Instead, use full stops to separate ideas into distinct sentences where appropriate.

Neutrality and Diversity
- Sentences must span *diverse domains*, including daily life, technology, abstract concepts, and nature.
- Use a mix of *pronouns* ("he," "she," "they"), *generic entities* (e.g., "a person," "a machine"), and *articles* ("the," "a"). Ensure pronouns are evenly distributed across the dataset to represent diverse actors.

Task Output
1. Generate *50 sentences* adhering strictly to the above structure and requirements.
2. Ensure diversity in domains, axes, and event relationships while maintaining clarity and coherence.
3. Each sentence must explicitly include:
   - **At least two Event-Related Axis**.
   - **Four or more Annotation Axes**, with the *Main Axis (Factual Events)* included.

Examples of Correct Sentences
1. "She is cooking dinner. At the same time, the oven is malfunctioning, which causes delays in her preparation. She checks the ingredients repeatedly, ensuring nothing is missing, while worrying that the dish may not turn out as planned. Despite the challenges, she intends to serve the meal on time to surprise her family."
2. "He is driving to work, navigating through dense traffic as the morning rush intensifies. Meanwhile, the traffic jam worsens due to a nearby accident, forcing him to rethink his route while calculating the estimated delay. He considers taking a detour through side streets, hoping to save time, but worries it might lead to further complications."
3. "She is watering the garden while the sun remains hidden behind the clouds, leading to slower evaporation. She frequently checks the soil moisture, believing that overwatering might damage the plants, though she intends to use organic fertilizer soon."

Figure 8: Plaintext markdown prompt for Benchmark II.