Time to Revisit Exact Match

Anonymous ACL submission

Abstract

Temporal question-answer (QA) is an established method to assess temporal reasoning in large language models (LLMs). Expected answers are often numeric (e.g., dates or durations), yet the model responses are evaluated like regular text with exact match (EM), unable to distinguish small from large errors. In this investigative work, we frame temporal QA as a numerical estimation task to assess the shortcomings of EM. We introduce *TempAnswerQA*, a benchmark distilled from *Test of Time* and *TempTabQA*, where all questions require a numerical temporal answer, allowing us to evaluate models beyond EM. We used the forecasting metrics symmetric mean absolute percentage error (sMAPE) and mean absolute scaled error (MASE). With sMAPE, we found that error size and EM are decoupled. Models with low EM still had low sMAPE (both 20%), and some models had high sMAPE despite high EM. Scaling errors by the deviance of the ground truth data with MASE reshuffles model rankings compared to EM, revealing gaps in models' understanding of temporal domain knowledge, especially when trained with synthetic data. Lastly, the models' most frequent error was to deviate only ± 1 from the ground truth. sMAPE and MASE, unlike EM, adequately weight these errors. Our findings underscore the need for specialised metrics for temporal QA tasks [1].

1 Introduction

Time is an inherent part of the real world, and reasoning about it is essential for intelligent behaviour (Xiong et al., 2024). As such, temporal reasoning is crucial in many domains, including high-stakes areas such as logistics (Li et al., 2023), finance (Wu et al., 2023), and medicine (Blease et al., 2024), which increases the stakes for adequate evaluation. Question-answering (QA) benchmarks are a well-established method to perform this evaluation, and the binary string-matching metric exact match (EM) is widely used to assess their effectiveness (Wang and Zhao, 2024; Wei et al., 2023).

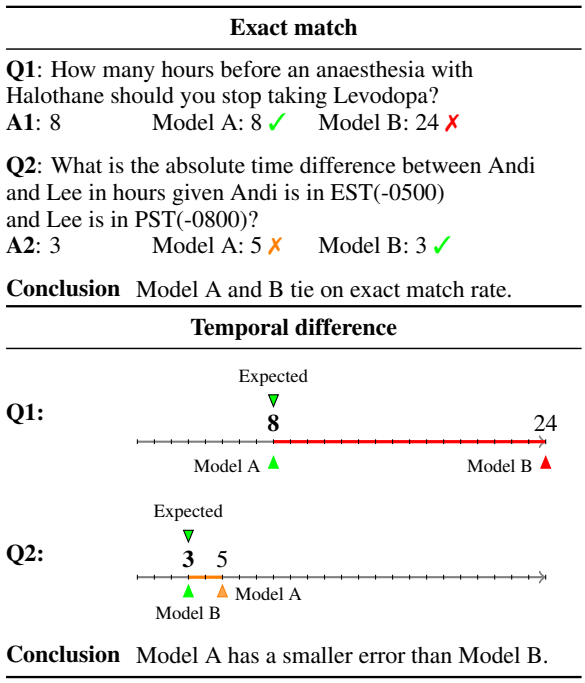


Figure 1: Exemplary performance evaluation of two models comparing exact match and temporal difference. Both models have an exact match of 50%, but Model B has a greater temporal difference than Model A.

However, while it is prevalent, EM does not consider the continuous nature of time. As illustrated in Fig. 1, EM considers Model A and Model B to be tied, despite Model A's error being much smaller (Δ2h) than Model B's (Δ8h). Although there are popular continuous alternatives, such as ROUGE, METEOR (Gupta et al., 2023) and F1 (Gruber et al., 2024), they collapse to binary scores when temporal answers consist solely of digits. Therefore, current benchmarks suffer from a mismatch between evaluation and deployment risk (Fig 1). This work aims to solve this mismatch by exploring metrics

[1] Code and data will be made publicly available

better suited to the temporal nature of the task.

Exploring continuous alternatives to EM allows us to differentiate between small and large errors. Beyond that, continuous metrics are more adequate to assess temporal reasoning for two more reasons. First, studies by Jack Lindsey et al. (2025) and Khodja et al. (2025) demonstrate that LLMs tend to approximate the answer to a (temporal) arithmetic task. Relying solely on EM risks undervaluing models that approximate correct answers well. Second, answers to temporal questions can be ambiguous, such as calculating a person's age using only their birth year, where two answers with a difference of 1 year could be true(Khodja et al., 2025). This ambiguity is called transition times. With EM alone, we cannot distinguish relevant errors from transition time ambiguities.

We frame temporal QA as a numerical estimation task and borrow two scale-free error metrics from forecasting to evaluate LLMs beyond EM. The first is the symmetric mean absolute percentage error (sMAPE) (Tofallis, 2014), which measures the percentage error of the model predictions. The second is the mean absolute scaled error (MASE) (Hyndman and Koehler, 2006). This metric scales errors by a sensible baseline derived from the benchmark data and thus aims to measure the models' temporal domain knowledge.

Our contributions can be summarised as follows:

1. We sample QA pairs from recent temporal benchmarks composed solely of questions requiring temporal answers to explore the limitations of EM. Augmenting questions with metadata allows us to transform model responses into time-aware objects, enabling evaluation using regression-based metrics.

2. Our evaluation with the regression-based metric sMAPE reveals that relative errors do not increase much even for very low EM (both $\sim$20 %). At the same time, it reveals outliers, that is, models with large relative errors despite a high EM. EM and sMAPE produced similar but not identical model rankings, making it a crucial addition to identifying robust models that made more minor errors.

3. Scaling errors by the deviance of the ground truth data using MASE assesses the temporal domain knowledge of the models. MASE produces different model rankings than EM, lowering the ranking of models trained on synthetic data. MASE reveals that models can achieve high EM and sMAPE and still make errors that exceed what we expect, given sufficient temporal domain knowledge.

4. Finally, by treating errors numerically, we show that many model predictions are off by only ± 1, caused by transition times (e.g., determining someone's age based only on the birth year). Furthermore, MASE shows that the error magnitude was not symmetric to the sign and that errors with a positive sign are much larger (> 0). Our findings underscore the need for a specialised evaluation procedure for temporal QA tasks and the inadequacy of using EM alone.

2 Related work

2.1 Temporal QA benchmarks

Generally speaking, Temporal QA aims to evaluate a model's understanding of time. Prior work often thematises the numeric nature of this task. The seminal QA benchmark *TempQuestions* by Jia et al. (2018) defined temporal questions as those that have a temporal expression (e.g. "three weeks"), a temporal signal (e.g. "before") or expect a temporal answer ("When..."). The latter indicates that the expected answer needs to be a measure of time. Tan et al. (2023) proposed categorising temporal questions into increasingly more complex levels of temporal understanding, namely those with a time-time, time-event, and event-event relation between question and answer. Again, this highlights how the numeric properties of time play a central role in temporal QA. Furthermore, temporal reasoning capabilities were often linked to mathematical reasoning skills (Su et al., 2024b; Yuan et al., 2023; Fatemi et al., 2024; Wang and Zhao, 2024). While there is consensus on the numeric properties of time, there exist.

2.2 Evaluation challenges in temporal QA

All benchmarks mentioned above either use token-level binary metrics or EM for evaluation. In one instance, ROGUE and METEOR were also used (Gupta et al., 2023).

Non-binary evaluations were conducted in some instances. Tan et al. (2023) and Wang et al. (2025) measured the mean absolute error for a selection of temporal arithmetic tasks. However, this measure cannot be compared across temporal resolutions (days vs. years). Tan et al. (2023) also measured

trend accuracy, recognising that temporal errors are directional. Since this metric is binary, it does not detect directional biases.

Evaluations of models that prevent errors in medication directions are less informative because the metrics do not consider the numeric nature of time (Pais et al., 2024). In another application setting, Zhang et al. (2025) mitigated this issue using a temporal version of the F1 score, which considers only temporal entities. This score is valid for longer texts but not if answers consist of only digits, as in our setting.

A review by Su et al. (2024a) shows that a growing body of work in temporal QA focuses on knowledge graphs. They often aim to retrieve the correct answer from graphs. Retrieval is evaluated differently from free text, so the concerns raised in this work do not apply here.

2.3 Transitional times

The necessity of investigating error magnitudes has been shown before. Khodja et al. (2025) showed that the models have a significantly higher log-likelihood for answers constituting transitional times (errors of ± 1) than for the correct answer. They hypothesised that transitional dates are more prevalent in the models' training data since events tend to be mentioned more often around their start and end. However, the log-likelihood of answers is not available for closed-source models.

Fatemi et al. (2024) also observed a higher proportion of errors equal to ± 1 in duration questions and speculated that models may approximate the answers well but fail in the final arithmetic computation. Despite these findings, no alternative to EM has been proposed. We, therefore, see an urgent need to investigate model errors on a continuous scale.

3 Methods and data

3.1 Dataset creation

Existing temporal QA benchmarks expect a mix of free text and temporal answers. "Who won the Oscar for best actor in 2024?" is a temporal question, but its answer is not. "When was Oppenheimer released?", on the other hand, expects a temporal answer. Currently, no QA dataset expects only (numeric) temporal answers. We classify an answer as a **temporal answer** if it is a date or a duration (including age). To fill this gap, we sampled a QA dataset that expects only temporal answers, which we will refer to as *TempAnswerQA*. Tab. 1 contains an example of the dataset.

Question	Answer	*Temporal answer*	*Answer format*
How many years did Art Carney work as an actor starting from 1939?	54	✓	# years
Who was the spouse of Art Carney in 1970?	Barbara Isaac	✗	–

Table 1: Example of labelling results for *TempAnswerQA*. Questions from TTQA and ToT expecting a temporal answer (date or duration) were retained. The expected answer format was added to facilitate parsing answers as numeric objects. Newly created columns are in italics.

The dataset should reflect current benchmarks and, therefore, should include stand-alone questions, questions that require context, real-world questions, and synthetic ones. The latter has become increasingly relevant for combating leakage into LLMs' training data. Two datasets, Test of Time (ToT) and TempTabQA (TTQA), meet these requirements [2].

ToT (Fatemi et al., 2024) is a synthetic QA benchmark for temporal reasoning. It consists of two subsets. One is the *arithmetic* subset, which has a real-world focus and contains questions that require time-related computations. The other is the *semantic* subset, which asks questions related to a randomly generated graph that assesses the model's understanding of temporal semantic and logical reasoning.

The enhanced version of TTQA evaluates a model's ability to answer temporal questions over semi-structured Wikipedia tables (Deng et al., 2024). The authors split the dataset to mitigate data leakage problems into a head and tail dataset, where the latter consists of less frequented tables.

We manually extracted questions that require a temporal answer, which leaves us with 1103 QA pairs for the head subset of TTQA and 634 for the tail subset. For ToT, we extracted 1016 QA pairs for the arithmetic subset and 681 for the semantic subset. In total, we have **3434 QA pairs**. Additionally, we annotated the required temporal unit for each question, i.e. if the answer is a date or a temporal measure in years, months, days, minutes, or seconds. We picked the higher temporal resolution

[2] Both have CC-BY-4.0

ToT		TTQA	
Temporal unit	Count	Temporal unit	Count
# seconds	411	# years	1194
Date	328	yyyy	305
# years	229	# days	94
# days	100	# months	85
# months	50	Date	59
# minutes	38		

Table 2: The number of questions per temporal unit of the answer. Answers can be either a duration measured as a number of <temporal unit>, a full date or a date with only the year (yyyy).

if the answer contained a mix of units, for example, seconds if the answer was formatted as HH:SS. Lastly, we annotated the expected answer format to allow parsing the answer numerically as integers, `timedelta`, or `datetime` objects in Python. Tab. 2 lists the number of temporal answer units per dataset.

3.2 Regression-based metrics for temporal QA

Metrics used to evaluate QA benchmarks are designed for text and, therefore, do not capture the size and direction of the error for temporal answers. Specifically, minor errors due to transitional times are indistinguishable from significant errors. *TempAnswerQA*'s expected answers all have numerical representation, which allows us to use regression-based metrics for evaluation.

There are a few considerations that we need to make before selecting metrics. We need to select (1) an aggregation technique that avoids errors of different signs cancelling each other out, (2) decide whether we want to weight errors, (3) how to summarise errors, and lastly, ensure that (4) errors will be comparable across different units, e.g., years and seconds.

We selected metrics using absolute errors to avoid the cancellation of errors of different signs. We decided against weighting errors by squaring or taking their logarithm since this impedes interpretation, and we lack justifications. Errors can be summarised using the mean or median. However, the median resulted in many scores of 0s or 100s for EM and sMAPE. Therefore, we pick the mean. Lastly, we must select a scale-free metric to compare errors across units (e.g. relative errors).

sMAPE is scale-free and uses absolute errors. It is bound between 0 and 100 and has a higher symmetry between negative and positive errors than its precursor, the mean absolute percentage error. sMAPE cannot be easily compared between experiments as its denominator contains model predictions and expected values. It is defined as:

$$sMAPE = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{|y_i| + |\hat{y}|},$$

where n is the number of QA pairs, y is the expected temporal answer, and $\hat{y}$ is the predicted temporal answer. If an answer is not parsable, sMAPE is defined as 100%, and if the numerator and denominator are 0, we define it as 0%.

A subset of answers are dates whose percentage error is not defined (Tab. 2). Therefore, we also consider MASE. It fulfils our requirements and is defined for dates. MASE measures the absolute errors scaled by the mean absolute deviation of the dataset. MASE is also considered superior to most forecasting metrics and is used in the well-known Makridakis forecasting challenge (Makridakis et al., 2022). It is defined as:

$$MASE = \frac{1}{n} \sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{|y_i - \bar{Y}_u|},$$

where $\bar{Y}$ is the average of the expected values. Instead of using all data to calculate $\bar{Y}$, we use a temporal unit-specific $\bar{Y}_u$. Some of the answers have a bimodal distribution. The answers with the temporal unit years in ToT have a peak for the answers < 100 and a peak around 2000. The mean is not representative in this case. To resolve this issue, we performed clustering with a setting that allowed our model also to return one cluster (unimodally distributed). The results and model settings are in the Appendix E.

The intuition behind MASE is that the answers in this dataset are not uniformly distributed. With sufficient domain knowledge, the correct answer can be estimated. When answering how old someone is, for example, we expect the answer to not exceed 100 by much. Without human annotation, MASE can capture these expected values from the dataset.

Another class of metrics measures semantic similarity. BERTScore (Zhang et al., 2020) is a widespread implementation of such a metric. However, it cannot distinguish between small and large

differences between integers (Appendix F), so we did not consider it.

3.3 Models and prompts

Similarly to previous work, we used a selection of open-source models for our experiments, namely Phi-4-mini, Phi-4 (Abdin et al., 2024), Llama-3.1-8B, Llama-3.3-70B (Grattafiori et al., 2024), Qwen2.5-7B, and Qwen2.5-14B (Qwen et al., 2025). The model settings are in the Appendix (B). Since evaluation relied on parsing answers into time-aware objects, we selected instruction-tuned models for better instruction-following capabilities. We considered using Timo, a temporal Llama 2 model by Su et al., but its context window was too small for some questions.

TTQA and ToT come with their own set of (user) prompts, which we adopted to use chat templates without any other modifications. Our selection of small models had difficulties following instructions otherwise. We moved the formatting instructions to the system prompt. These were especially important for ToT, where answers needed to be JSONs. Examples were presented as turns between the assistant and user in the case of few-shot prompting. Both adjustments improved instruction following. Furthermore, models produced valid JSONs more often when ending the prompt with an assistant turn, appending the beginning of the required JSON and removing generation prompts (see Appendices C and D for prompts and B for small experiments justifying chat templates and different generation strategies).

4 Experiments and results

We conducted our experiments based on these six selected models of different sizes with and without few-shot prompting on the sampled dataset *TempAnswerQA*. Its questions expect temporal answers that can be assessed in a regression-like fashion. Our experiments aim to answer the following questions:

RQ1: Is the binary metric EM enough to evaluate LLMs on temporal QA benchmarks expecting temporal answers?

RQ2: Can regression-based metrics help improve our understanding of LLMs' performance on QA tasks expecting a temporal answer?

RQ3: What advantages do we have in using regression-based metrics compared to EM?

4.1 Exact match does not capture error magnitudes

EM does not differentiate between small and large errors, that is, their error magnitudes. Wrong predictions (EM $= 0$) can have vastly different values for sMAPE. For example, two models with EM of 80% could have sMAPE of 1% and 20%, respectively. The lower the EM rate, the wider the range of values that sMAPE can assume. Appendix A contains an illustration of this relationship.

Model predictions on the *TempAnswerQA* dataset evaluated by EM and sMAPE are shown in Fig. 2. According to EM, Llama-3.3-70B is the best model. Phi-4 and then Qwen2.5-14B closely follow it. Smaller models follow thereafter. The range of EM is large with values as low as 20% for Llama-3.1-8B, Qwen2.5-7B, and Phi-4-mini. sMAPE values, on the other hand, span a shorter range between models and data splits (up to 40%) than in the EM dimension (15-80%). sMAPE changes the model ranking, placing Qwen2.5-14B in the first place. It is also the model with the narrowest 95% confidence interval. All models, except Qwen2.5-14B, have outliers hovering around 40%. For example, Llama-3.3-70B failed severely in answering how many days the Ingenuity took to get to Mars. Due to an arithmetic mistake, it answered 0.057 days. The expected answer is 418 days.

Qwen2.5-14B, which has increased mathematical capabilities and improved understanding of structured data, overtakes Llama-3.3-70B when evaluated with sMAPE. The findings also show that larger models performed better, equivalent to the EM results. If errors produce non-linear costs and low errors are more desired than a high EM, Qwen2.5-14B should be preferred over Llama-3.3-70B. The results in tabular form are in Appendix G.

4.2 Tolerable error magnitudes depend on the task difficulty

MASE was introduced as a metric superior to other regression-based metrics and is the gold standard in forecasting. Unlike sMAPE, it can also be applied to dates. Its main property is that it scales the prediction errors by the difficulty of the problem, which is relevant because answers in the *TempAnswerQA* are not arbitrarily distributed and benefit from temporal domain knowledge. For example, a subset of questions is related to the time zone. The maximum time difference between time zones is 26 hours. Models with this knowledge should

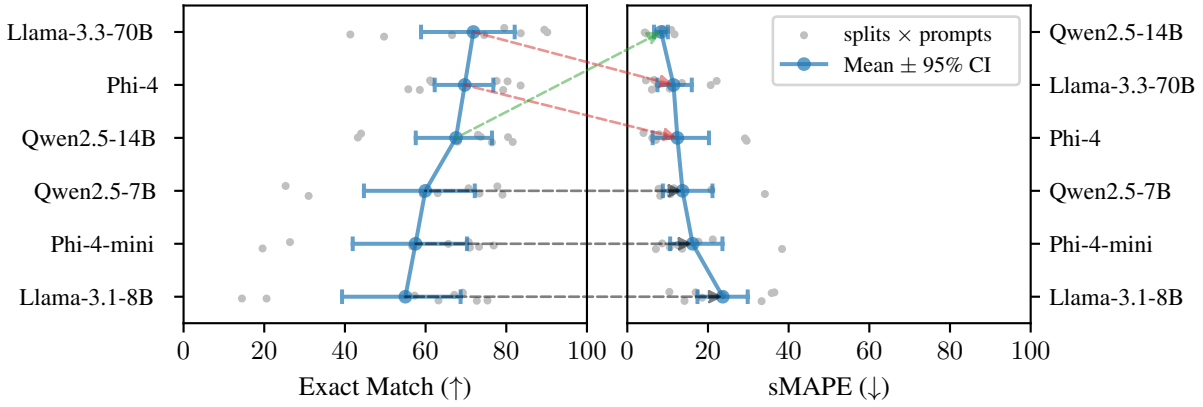


Figure 2: Model ranking by sMAPE and EM. Blue dots represent the mean score, and bars around it the 95% confidence interval. Grey dots are individual runs with and without few-shot prompting on all splits of ToT and TTQA. Arrows indicate a rank change from EM to sMAPE. It is green if it improves, red if it decreases, and black if it stays the same.

not produce errors larger than that. Without human annotation for acceptable error ranges, MASE can extract it from the data instead.

Model predictions on the *TempAnswerQA* dataset evaluated by EM and MASE are shown in Fig. 3. All models had a MASE greater than 1, meaning their mean absolute error was higher than the mean absolute deviance of the dataset stratified by the temporal answer unit per data split. Although experiments with sMAPE and EM ranked larger models higher and sMAPE showed the advantage of Qwen2.5-14B being trained in mathematical tasks and structured data, MASE draws a different picture. Llama-3.1-8B's rank dramatically increases from the last to the second place. Scaling errors is relevant, as the relative size does not reveal if an error is significant. Qwen2.5-7B, for example, when asked which year Jenson Button (racing driver) won his first championship, answered with 2018. The expected answer is 2009, which leads to a scaled error of 5.12. Athlete's careers are relatively short-lived, making a 9-year error striking.

Llama-3.1-8B is the only model that was not trained on synthetic data. MASE captures domain knowledge, and we hypothesise that the synthetic data on which Qwen and Phi were trained distort the models' temporal domain knowledge. The tabular results are in the Appendix G.

4.3 Scaled errors produce different rankings, percentage errors do not

EM is a gold standard metric for evaluating LLMs on QA benchmarks. Therefore, it is necessary to compare sMAPE and MASE with EM. We used Spearman's rank correlation coefficient to compare model rankings across metrics, and the results are shown in Fig. 4a and Fig. 4b.

EM had a high rank correlation with sMAPE for both datasets, ToT (-0.82) and TTQA (-0.92). It was negative because a higher EM is better, while a lower sMAPE is better. The correlation was much lower between MASE and EM, with values around 0.4 for both datasets.

Considering the high agreement in the ranking between both metrics, but knowing that sMAPE is more affected by outliers by definition, which is also observable in Fig. 2, we find that sMAPE is a crucial addition to EM for model evaluation if error magnitude matters. Since it does not produce significantly different model ranks, interpreting EM and sMAPE in tandem is easier.

MASE produced different model ranks. This is unsurprising since, unlike sMAPE, the same error magnitude will scale differently depending on the task. MASE is, therefore, more strict if data deviance is low. Scaling errors for time-zone or age-related questions are such instances. Datasets are most likely designed to span reasonable time periods. If not, clustering should help make MASE more reliable. However, further verification, ideally by humans, is required.

4.4 Transitional times and error directions

Casting answers into time-aware objects allows us to investigate raw errors, helping us identify tiny errors (+/-1) due to transitional times and their

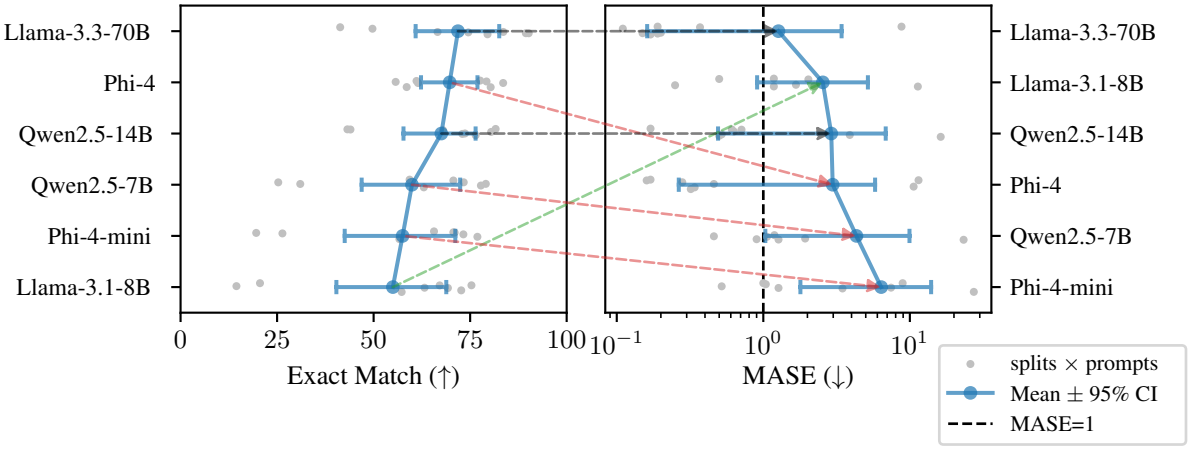
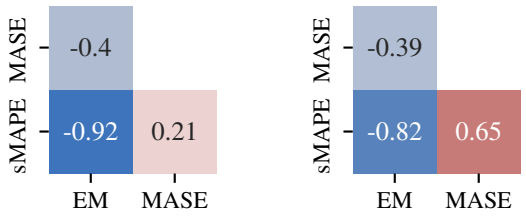


Figure 3: Model ranking by MASE and EM. Blue dots represent the mean score, and bars around it the 95% confidence interval. Grey dots are individual runs with and without few-shot prompting on all splits of ToT and TTQA. Arrows indicate a rank change from EM to MASE. It is green if it improves, red if it decreases, and black if it stays the same.



(a) TTQA correlations (b) ToT correlations

Figure 4: Spearman rank correlation between metrics on all experiments (models × prompts) per dataset.

ToT			TTQA		
$\|e\|$	Count	Share (%)	$\|e\|$	Count	Share (%)
1	1002	11.62	1	1853	49.49
2	446	5.17	2	250	6.68
4	344	3.99	3	159	4.25
3	258	2.99	4	128	3.42
5	208	2.41	6	117	3.12

Table 3: Five most frequent absolute errors per dataset over all experiments (models × prompts) with number of occurrences and relative share in percent. Note that models performed better on TTQA, therefore, the share of errors with $|e| = 1$ is so high.

direction.

Transitional times most often involve questions asking for durations. In the following, we will measure whether an error $|e| = 1$ appeared more often than others and if it appeared more often in questions asking for durations.

Indeed, our analysis reveals that an absolute error of 1 is the most common in both datasets (Tab. 3). For ToT, the share of absolute errors equal to 1 is 11.62%. For TTQA, it is 49.49%. This finding is striking because the number of unique errors is infinite.

Next, we verified whether errors equal to $|e| = 1$ occurred more often for duration-related questions. We divide the dataset by the type of question defined by the authors of ToT and the answer format for the dataset TTQA. Tab. 4 shows that the types of questions are evenly distributed within ToT. The share of question types where $|e| = 1$ is vastly different. RelationDuration and Duration questions tremendously increased their share. Trick questions doubled as well. The Trick setup confused LLMs to decide whether to ex- or include either the start and end dates for a duration calculation.

Due to a lack of question-type labels in TTQA, we used the expected answer format instead. Tab. 5 compares the share of questions by answer format for all data and when the errors are equal to $|e| = 1$. The TTQA dataset contains many more duration-related questions than ToT. Therefore, the increase in share is not as prominent as in ToT, but it is striking that all non-duration answers have a significantly smaller share among the questions where

Question Type	Share (%)	
	all data	where $\|e\| = 1$
MultiOP	20.57	4.99
EventAtWhatTime	20.15	4.59
RelationDuration	19.98	32.14
AddSubtract	14.73	16.57
Duration	11.79	18.96
Trick	6.89	22.36
Timezone	5.89	0.40

Table 4: Share of question types in ToT dataset compared by share of question types where prediction error is 1 ($|e| = 1$) over all experiments (models $\times$ prompts).

Answer Format	Share (%)	
	all data	where $\|e\| = 1$
# years	68.74	81.27
yyyy	17.56	5.56
# days	5.41	7.34
# months	4.89	5.40
%B %d, %Y	3.40	0.43

Table 5: Share of answer formats in TTQA dataset compared by share of answer formats where prediction error is 1 ($|e| = 1$) over all experiments (models $\times$ prompts).

the error is $|e| = 1$.

Finally, we investigate whether model errors have a directional bias. In Tab 6, we see that sMAPE is similar for positive and negative errors. This is not the case for MASE. Positive errors produced much higher MASE. This difference is pronounced for the TTQA dataset. In the ToT dataset, the difference in the standard deviation is more noticeable. This insight is relevant to applications where the cost of errors is not symmetric with respect to direction.

5 Conclusion

This work extended existing benchmarks by evaluating LLMs on questions expecting a temporal answer. Furthermore, we showed that the gold-standard metric EM does not capture all relevant information, namely error magnitude and direction.

To this end, we used sMAPE and MASE, two regression-based metrics that captured properties in the prediction errors of the models that EM did not. sMAPE was relatively low, even if EM was low. This suggests that models may have an understanding of the correct answer. Qwen2.5-14B, which was trained on structured data and mathematical reasoning, performed best according to sMAPE, overtaking Llama-3.3-70B, the best model according to EM. Both Llama models performed the best according to MASE; they were the only models not trained on synthetic data, suggesting that their temporal domain knowledge is higher and synthetic data distorts this knowledge.

Answers to duration-related questions can be ambiguous due to transitional times, leading to two answers being correct with a difference of just 1. This leads to an inflation of errors equal to ± 1. sMAPE and MASE are continuous metrics and thus provide a more balanced evaluation than EM.

Lastly, we could show that MASE and sMAPE are valuable additions to EM. sMAPE ranks models similar to EM, making it a good choice to use in tandem with EM while offering more insight into the model's robustness in producing small errors. MASE ranks models significantly differently. It attempts to scale errors by prior domain knowledge about the correct answer and, unlike sMAPE, will identify outliers not only by its relative error but relative to the difficulty of the task. Without human-annotated data, MASE is a viable alternative to measure prior temporal knowledge.

Dataset	Error	sMAPE (±std)	MASE (±std)
ToT	neg.	24.73 (31.21)	1.40 (7.09)
	pos.	21.60 (29.26)	3.98 (40.82)
TTQA	neg.	22.83 (30.42)	0.55 (1.09)
	pos.	29.32 (31.72)	48.09 (334.80)

Table 6: sMAPE and MASE including their standard deviation where error is either strictly positive or negative per dataset.

6 Outlook

More work is required to verify the benefit of regression-based metrics. This can be achieved through either a separate dataset or by human preference. Specifically, other approaches for scaling errors for MASE should be considered. Small sMAPE suggests that models have a good understanding of the problem but have problems dealing with numbers. Tool calling is an interesting next step in assessing whether the low performance is due to arithmetic miscalculations rather than insufficient temporal reasoning capabilities.

7 Limitations

There are answers in both datasets that are temporal, but we did not include them because they are not trivially evaluable. These include date and time

ranges, multiple answers, or frequencies such as "every first Monday of the month". The latter is related to absolute times and dates, which have a bounded error. For example, if we ask in which month Christmas is celebrated, the maximal error is 11 months, while for other answers in the dataset, the error can be arbitrarily large in theory.

Both regression-based metrics are not defined for all answers. Either because the answer is not parsable in the case of MASE, or because the answer is a date or a time (sMAPE). This shortcoming needs to be addressed.

MASE was scaled by each subset of both datasets and the expected temporal unit of the answer. Although this approach makes the reasonable assumption that the authors of the paper produced problems that are similar within a subset and that the expected temporal unit is enough indication to capture similar kind of problems, this may not always hold. Clustering could potentially unravel such questions into more representative clusters, but this approach does not hold up to a hand-crafted dataset where the mean absolute deviance is neatly justified for each question.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 Technical Report. *arXiv preprint*. ArXiv:2412.08905 [cs].

Charlotte R Blease, Cosima Locher, Jens Gaab, Maria Hägglund, and Kenneth D Mandl. 2024. Generative artificial intelligence in primary care: an online survey of UK general practitioners. *BMJ Health & Care Informatics*, 31(1):e101102.

Irwin Deng, Kushagra Dixit, Vivek Gupta, and Dan Roth. 2024. Enhancing Temporal Understanding in LLMs for Semi-structured Tables. *arXiv preprint*. ArXiv:2407.16030 [cs].

Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. 2024. Test of Time: A Benchmark for Evaluating LLMs on Temporal Reasoning. *arXiv preprint*. ArXiv:2406.09170 [cs].

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *arXiv preprint*. ArXiv:2407.21783 [cs].

Raphael Gruber, Abdelrahman Abdallah, Michael Färber, and Adam Jatowt. 2024. ComplexTempQA: A Large-Scale Dataset for Complex Temporal Question Answering. *arXiv preprint*. ArXiv:2406.04866 [cs].

Vivek Gupta, Pranshu Kandoi, Mahek Vora, Shuo Zhang, Yujie He, Ridho Reinanda, and Vivek Srikumar. 2023. TempTabQA: Temporal Question Answering for Semi-Structured Tables. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2431–2453, Singapore. Association for Computational Linguistics.

Rob J. Hyndman and Anne B. Koehler. 2006. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 8 others. 2025. On the Biology of a Large Language Model.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. TempQuestions: A Benchmark for Temporal Question Answering. In *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18*, pages 1057–1062, Lyon, France. ACM Press.

Hichem Ammar Khodja, Frédéric Béchet, Quentin Brabant, Alexis Nasr, and Gwénolé Lecorvé. 2025. Language Models Struggle to Achieve a Consistent Temporal Representation of Facts. *arXiv preprint*. ArXiv:2502.01220 [cs].

Beibin Li, Konstantina Mellou, Bo Zhang, Jeevan Pathuri, and Ishai Menache. 2023. Large Language Models for Supply Chain Optimization. *arXiv preprint*. ArXiv:2307.03875 [cs].

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2022. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4):1346–1364.

Cristobal Pais, Jianfeng Liu, Robert Voigt, Vin Gupta, Elizabeth Wade, and Mohsen Bayati. 2024. Large language models for preventing medication direction errors in online pharmacies. *Nature Medicine*, 30(6):1574–1582. Publisher: Nature Publishing Group.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos,

David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 Technical Report. *arXiv preprint*. ArXiv:2412.15115 [cs].

Miao Su, Zixuan Li, Zhuo Chen, Long Bai, Xiaolong Jin, and Jiafeng Guo. 2024a. Temporal Knowledge Graph Question Answering: A Survey. *arXiv preprint*. ArXiv:2406.14191 [cs].

Zhaochen Su, Jun Zhang, Tong Zhu, Xiaoye Qu, Juntao Li, Min Zhang, and Yu Cheng. 2024b. Timo: Towards Better Temporal Reasoning for Language Models. *arXiv preprint*. ArXiv:2406.14192.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards Benchmarking and Improving the Temporal Reasoning Capability of Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada. Association for Computational Linguistics.

Chris Tofallis. 2014. A Better Measure of Relative Prediction Accuracy for Model Selection and Model Estimation.

Jiexin Wang, Adam Jatowt, and Yi Cai. 2025. Towards Effective Time-Aware Language Representation: Exploring Enhanced Temporal Understanding in Language Models. *arXiv preprint*. ArXiv:2406.01863 [cs].

Yuqing Wang and Yun Zhao. 2024. TRAM: Benchmarking Temporal Reasoning for Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6389–6415, Bangkok, Thailand. Association for Computational Linguistics.

Yifan Wei, Yisong Su, Huanhuan Ma, Xiaoyan Yu, Fangyu Lei, Yuanzhe Zhang, Jun Zhao, and Kang Liu. 2023. MenatQA: A New Dataset for Testing the Temporal Comprehension and Reasoning Abilities of Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1434–1447, Singapore. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint*. ArXiv:1910.03771 [cs].

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. *arXiv preprint*. ArXiv:2303.17564 [cs].

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large Language Models Can Learn Temporal Reasoning. *arXiv preprint*. ArXiv:2401.06853 [cs].

Chenhan Yuan, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2023. Back to the Future: Towards Explainable Temporal Reasoning with Large Language Models. *arXiv preprint*. ArXiv:2310.01074 [cs].

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *Conference on Learning Representations*.

Xi Zhang, Zaiqiao Meng, Jake Lever, and Edmond S. L. Ho. 2025. Libra: Leveraging Temporal Images for Biomedical Radiology Analysis. *arXiv preprint*. ArXiv:2411.19378 [cs].

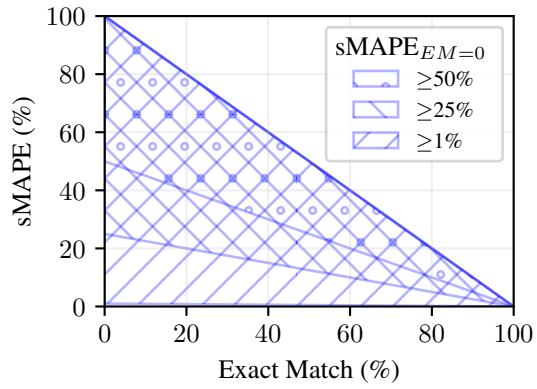


Figure 5: EM does not capture error magnitude. The possible variance in error magnitude (measured by sMAPE) is higher the lower EM is.

Dataset	Max. new token	End of sequence tokens
ToT	512	No
TTQA	512	Yes

Table 7: Generation settings.

Model	# Parsing rrrors	
	add generation prompt	continue final message
Llama-3.1-8B	0/50	2/50
Qwen2.5-7B	4/50	0/50
Phi-4-mini	13/50	1/50

Table 8: Number of parsable JSONs per model for different generation strategies tested on 50 randomly selected questions from the semantic split of ToT.

A sMAPE intuition

If a model's answer is wrong, the answer's error can range from a tiny relative error up to an infinitely large one. The lower EM, the higher the sMAPE can be in a model. In other words, models with the same EM can spread wider, and the lower EM is, the better sMAPE is at discriminating model performance. Fig. 5 illustrates this. If we assume that for all wrong predictions that a model makes, the minimum error measured by sMAPE is 1%, 25%, and 50%, the figure shows the range of values that a model can still score with respect to sMAPE for all possible values of EM.

B Model settings

All our models were accessed via Hugging Face using the `transformers` Python library at version 4.49.0 (Wolf et al., 2020). We used the default settings for each model in our experiments. For text generation, we used the settings in Tab. 7. We used a mix of GPUs to run our experiments, including GeForce 3090s and 4090s, and two A100s in parallel to run inference for Llama-3.3-70B. GPU hours required to run inference on Llama-3.3-70B required approximately 24 hours. Experiments with smaller models took 1-3 hours per run. At least as many GPU hours across GPUs were used to run small experiments or test code.

The evaluation of ToT depends on the models that produce parsable JSONs. Therefore, we experimented with setting either `add_generation_prompt` or `continue_final_message` to true in Hugging Face. The first appends an assistant token to our messages, if available, indicating an answer. The latter does not do this, prompting the models to continue their messages. The resulting prompts can be seen in the Appendix D. To test when JSON formatting was more successful, we randomly sampled 50 questions from the semantic split of ToT and compared the number of correctly parsed JSONs. The results are in Tab. 8. Setting `continue_final_message` produced less parsing errors (3 over three models) than `add_generation_prompt` (17 over three models).

The evaluation of TTQA also depended on the correct format of the output. Specifically, models needed to place their answer after the string "Final Answer:". We observed a low amount of correct formatting and thus experimented with transferring prompts into a chat template. The correct output formatting was compared between the original and the prompts translated into chat templates. We tested the models' instruction following on the tail split of the TTQA dataset. The results are shown in Tab 9. Qwen and Phi improved their instruction following, with Qwen almost doubling it from 44.52 to 99.56%. Llama has a slight decrease when using chat templates from 81.98% to 74.40%.

C TTQA prompts

In the following, we list the TTQA prompts used for this work. We compare the prompts originally used by Deng et al. (2024) and our adaption to make use of chat templates. For brevity, we replaced some turns in the few-shot example with "...". Furthermore, we did not use the original

Model	Correct output format (%)	
	Original prompt	With chat template
Llama-3.1-8B	81.98	74.40
Qwen2.5-7B	44.52	99.56
Phi-4-mini	94.76	99.38

Table 9: Number of answers containing expected string "Final Answer:" in their response for each model on the head split of the TTQA dataset. Percentage was calculated based on slightly varying numbers of questions as experiments were conducted at different steps in the labelling process.

questions, tables, or answers below but replaced them with placeholders enclosed by "<>".

C.1 TTQA zero-shot prompt

User prompt: Given an entity-centric table and corresponding question, answer the question by providing step-by-step reasoning and then clearly and concisely stating the final answer using "Final Answer:".
Each table-question pair is presented as a table (identified by "Table:") followed by a question (identified by "Q:"). Tables are presented in a linear format, with columns separated by tabs, rows separated by newlines, and subsections separated by double newlines. If necessary, assume the current date is December, 2022.

==========
Table:

<TABLE>

<QUESTION>

A: Let's think step by step.
Assistant:

C.2 TTQA zero-shot prompt as chat template

System prompt: Given an entity-centric table and corresponding question, answer the question by providing step-by-step reasoning and then clearly and concisely stating the final answer using "Final Answer:".
Each table-question pair is presented as a table (identified by "Table:") followed by a question (identified by "Q:"). Tables are presented in a linear format, with columns separated by tabs, rows separated by newlines, and subsections separated by double newlines. If necessary, assume the current date is December, 2022.
User prompt:
Table:

<TABLE>

<QUESTION>

A: Let's think step by step.
Assistant:

C.3 TTQA few-shot prompt

User prompt: Given an entity-centric table and corresponding question, answer the question by providing step-by-step reasoning and then clearly and concisely stating the final answer using "Final Answer:".
Each table-question pair is presented as a table (identified by "Table:") followed by a question (identified by "Q:"). Tables are presented in a linear format, with columns separated by tabs, rows separated by newlines, and subsections separated by double newlines. If necessary, assume the current date is December, 2022.

Here is an example that follows these instructions. Answer the provided questions in a similar format:

==========
Table:

<TABLE, SHOT 1>

Q: <QUESTION, SHOT 1>

A: <ANSWER, SHOT 1>

==========
...
<TABLE, SHOT 3>

Q: <QUESTION, SHOT 3>

A: <QUESTION, SHOT 3>

==========

Table:

<TABLE>

<QUESTION>

A: Let's think step by step.
Assistant:

C.4 TTQA few-shot prompt as chat template

System prompt: Given an entity-centric table and corresponding question, answer the question by providing step-by-step reasoning and then clearly and concisely stating the final answer using "Final Answer:".
Each table-question pair is presented as a table (identified by "Table:") followed by a question (identified by "Q:"). Tables are presented in a linear format, with columns separated by tabs, rows separated by newlines, and subsections separated by double newlines. If necessary, assume the current date is December, 2022.

Here is an example that follows these instructions. Answer the provided questions in a similar format:
User prompt
Table:

<TABLE, SHOT 1>

Q: <QUESTION, SHOT 1>

A:
Assistant prompt: <ANSWER, SHOT 1>
...
User prompt:
Table:
<TABLE, SHOT 3>

Q: <QUESTION, SHOT 3>

A:
Assistant prompt: <ANSWER, SHOT 3>
User prompt:
Table:

<TABLE>

<QUESTION>

A: Let's think step by step.
Assistant:

D ToT prompts

In the following, we list the ToT prompts used for this work. We compare the prompts originally used by Fatemi et al. (2024) and our adaption to make use of chat templates.

A few-shot version of the prompts was constructed by modifying existing questions. The chat template was filled as in C.4, where examples were presented as turns between the user and the assistant. In the case of the semantic subset, the graph information was included in the system prompt. The generation prompt was removed, and the assistant prompt was pre-filled.

D.1 ToT zero-shot prompt

User prompt: Natalie and Chris were born on 2004-Feb-18 and 2004-Dec-30 respectively. When Chris was 991 days old, how old was Natalie in days? Return your answer as a JSON like: JSON = {""explanation"": <your step by step solution>, ""answer"": <num_days>}
Assistant:

D.2 ToT zero-shot prompt as chat template

System prompt: Return your answer as a JSON like: JSON = {"explanation": <your step by step solution>, "answer": <num_days>}
User prompt: Natalie and Chris were born on 2004-Feb-18 and 2004-Dec-30 respectively. When Chris was 991 days old, how old was Natalie in days?
Assistant: JSON = {"explanation":

E Cluster results

MASE required the mean answer per temporal unit of the answer and the split of each dataset. Clus-

Expected	Predicted	BERTScore
1	1	1.0000
1	2	0.9998
1	10	0.9992
1	100	0.9987

Table 10: BERTScore for some predictions. Scores were rounded to the last four digits.

tering did not affect the TTQA data. ToT, however, exhibited some bimodality, which was identified by the clustering algorithm. The distribution of the answers per split and temporal unit for TTQA is shown in Fig. 6 and Fig.7. ToT's answer distribution for the arithmetic split before and after clustering can be found in Fig. 8 and Fig. 9 respectively and in Fig. 10 for the semantic split.

Clustering was performed using sklearn's HDBSCAN (hierarchical density-based spatial clustering of applications with noise) model. The minimum cluster size was set to 30% to avoid too small clusters. The model was allowed to produce single clusters. All other settings were set to default. We used version 1.6.1 of `scikit-learn` (Pedregosa et al., 2011).

F BERTScore

We did not consider similarity-based metrics as they tend to return high similarity for digits, regardless of how close they are to each other, as can be seen in Tab. 10.

G Results extended

Results in tabular form are listed in Tab. 12 for ToT and in Tab. 11 for TTQA. Fig. 11 contains a scatter plot with the results comparing EM and sMAPE and Fig. 12.

EM is defined for all pairs QA, sMAPE and MASE not. sMAPE is not defined for dates or times. Since it has a maximum value, namely 100%, it is defined even if the answer of the model is not parsable. MASE does not have this property as it has no upper bound. Instead, it is defined for dates and times. Tab. 14 lists the number of QA pairs in the ToT dataset for which either metric is defined, and Tab. 13 does the same for the TTQA dataset.

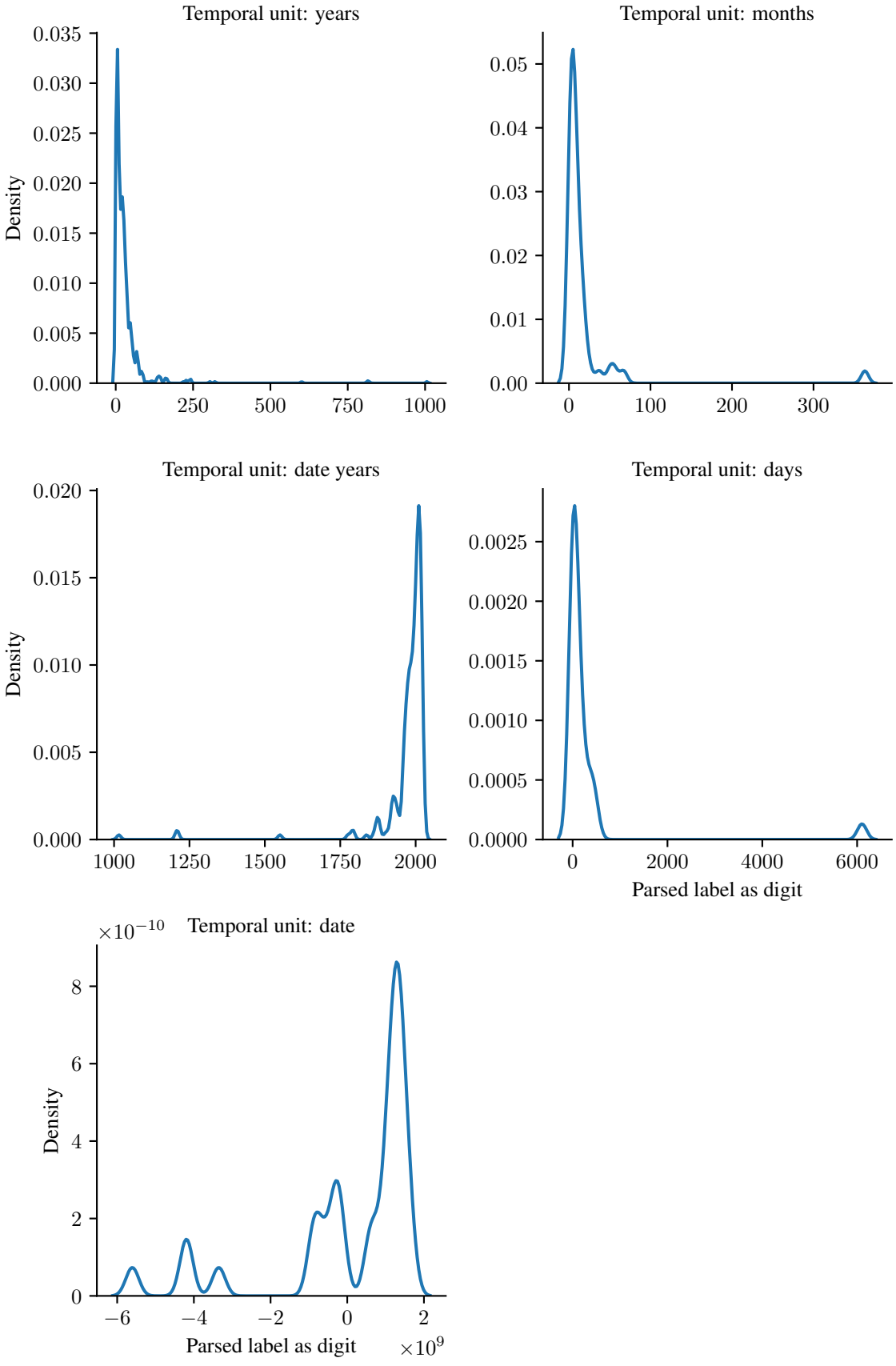


Figure 6: Distribution of the expected answers by temporal unit of the answer for the head split of the TTQA dataset. Answers were transformed into numeric form. In the case of dates, they were converted into timestamps.

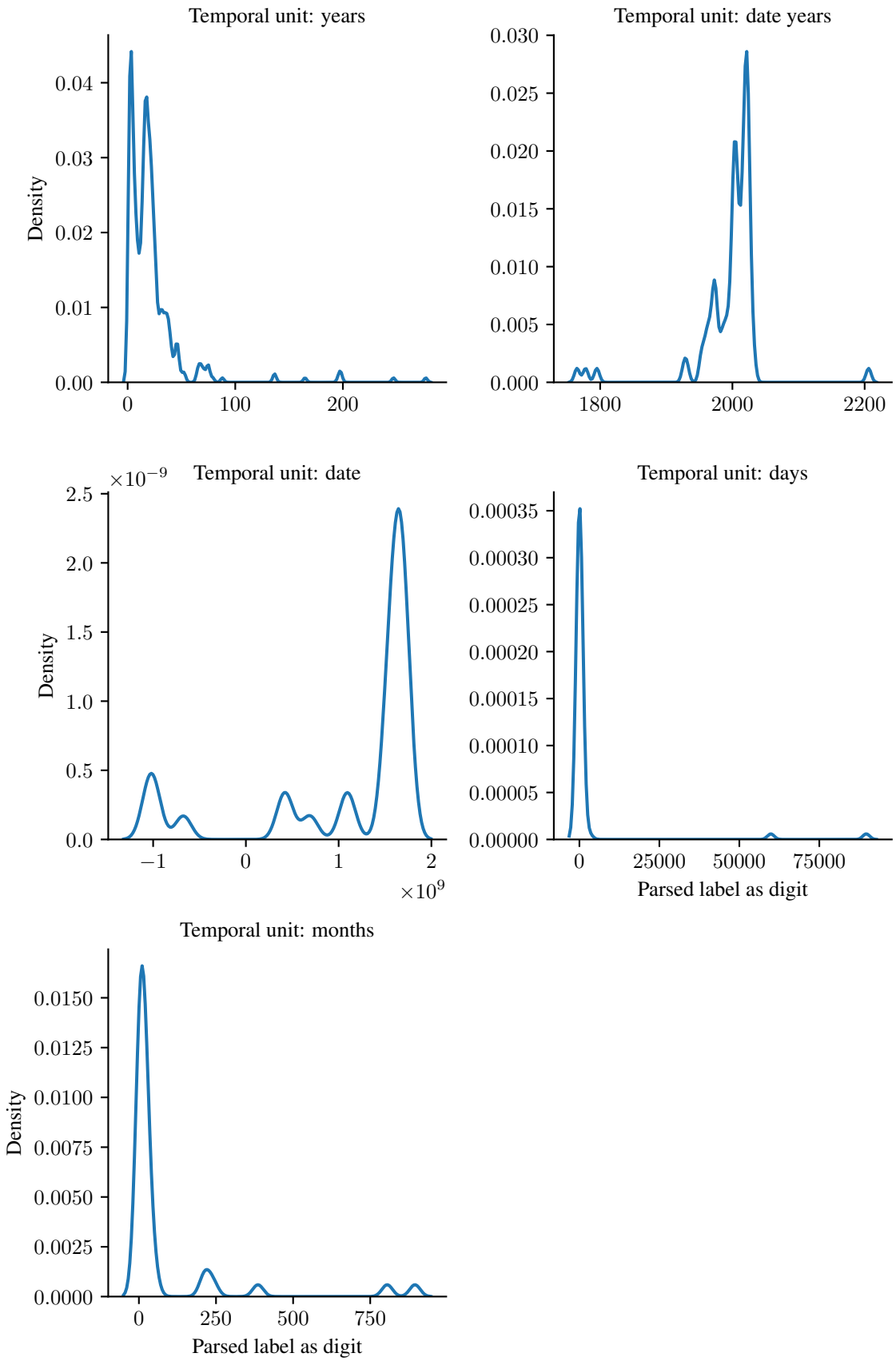


Figure 7: Distribution of the expected answers by temporal unit of the answer for the tail split of the TTQA dataset. Answers were transformed into numeric form. In the case of dates, they were converted into timestamps.

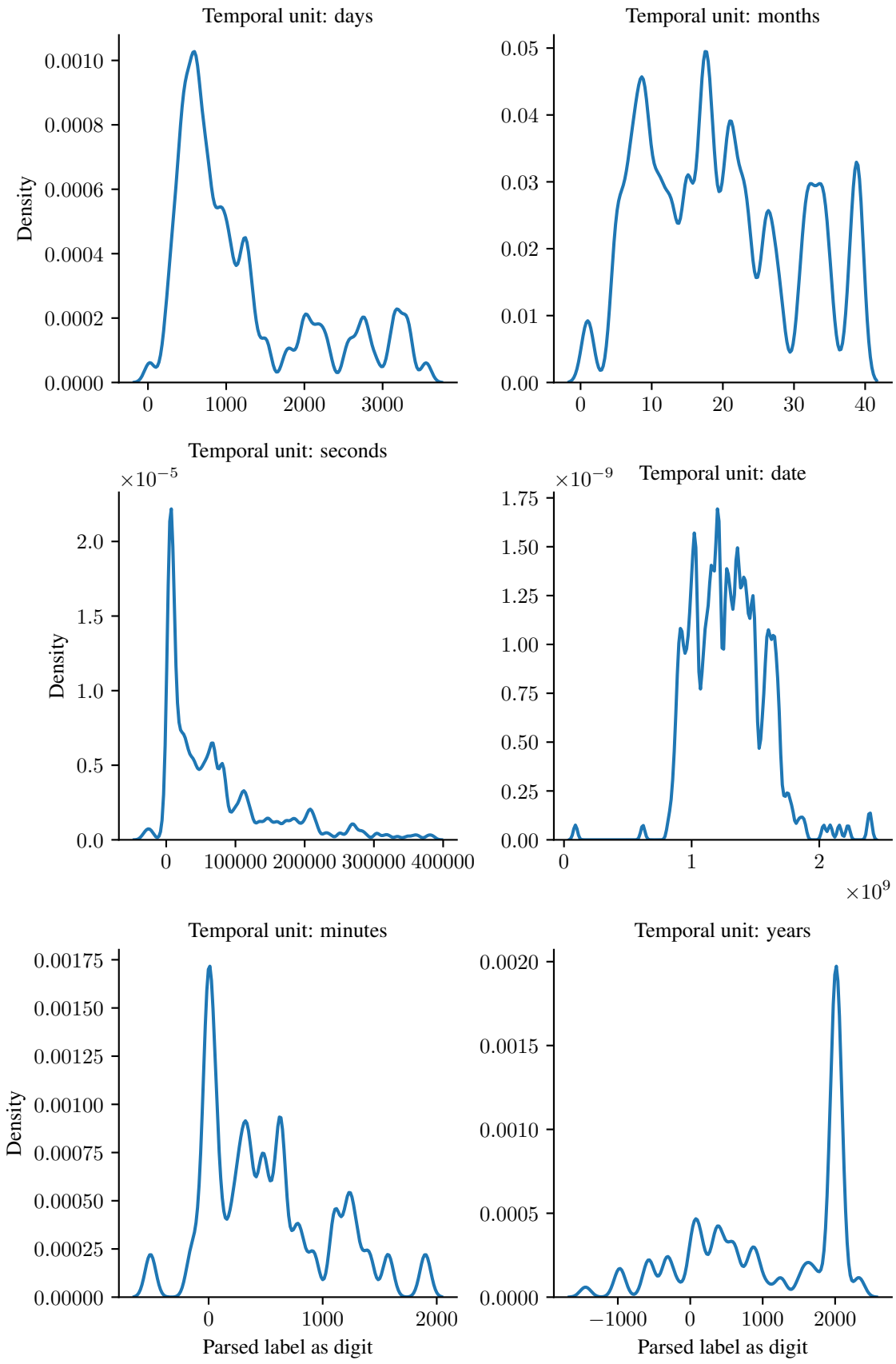


Figure 8: Distribution of the expected answers by temporal unit of the answer for the arithmetic split of the ToT dataset. Answers were transformed into numeric form. In the case of dates, they were converted into timestamps.

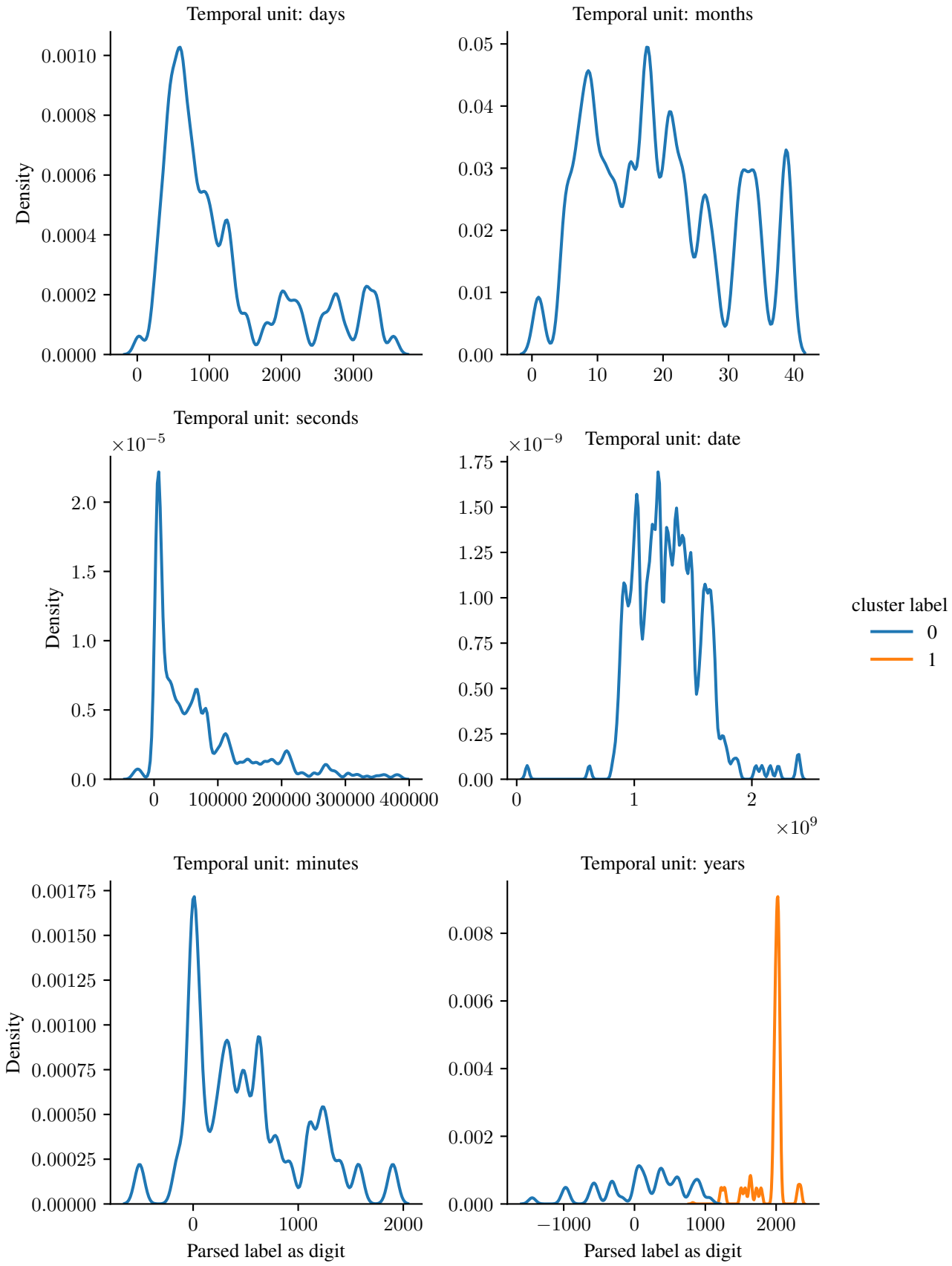


Figure 9: Distribution of the expected answers by temporal unit of the answer for the arithmetic split of the ToT dataset. If answers were clustered, clusters are highlighted by different colours. Answers were transformed into numeric form. In the case of dates, they were converted into timestamps.

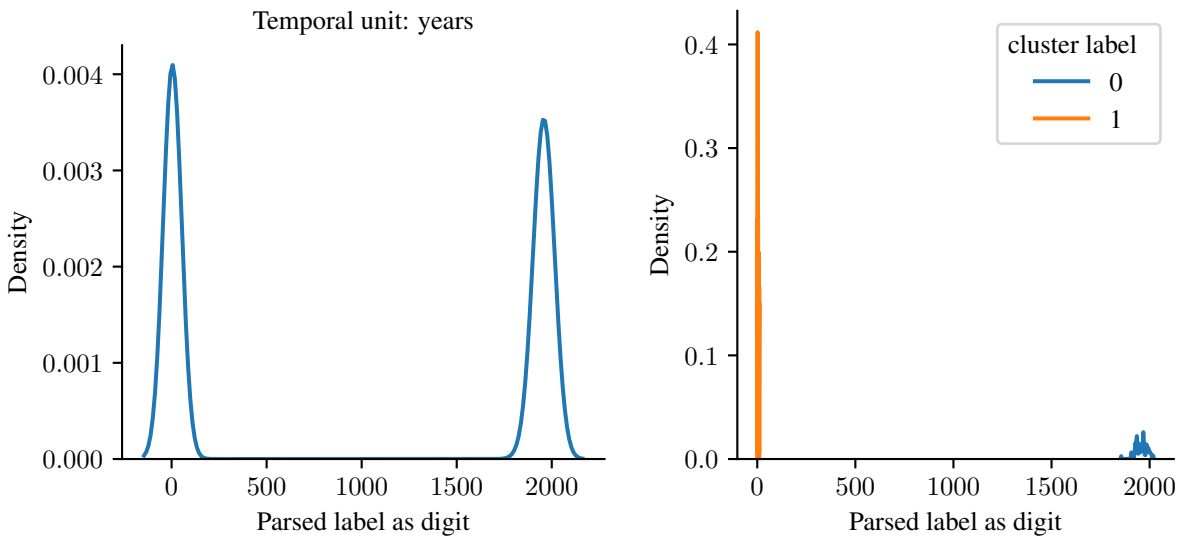


Figure 10: Distribution of the expected answers by temporal unit of the answer for the semantic split of the ToT dataset. Left is the raw distribution, and on the right is the distribution after clustering. Answers were transformed into numeric form. In the case of dates, they were converted into timestamps. If

Split	Model	Prompting	EM (↑)	sMAPE (↓)	MASE (↓)
head	Llama-3.1-8B	few shot	75.34	17.02	0.50
		zero shot	63.37	33.11	0.25
	Llama-3.3-70B	few shot	**83.71**	6.58	0.20
		zero shot	74.62	20.59	**0.17**
	Phi-4-mini	few shot	77.05	7.03	0.52
		zero shot	73.09	12.65	7.45
	Phi-4	few shot	79.39	6.13	0.34
		zero shot	61.30	29.36	0.29
	Qwen2.5-7B	few shot	79.12	7.77	4.47
		zero shot	77.77	8.05	0.91
	Qwen2.5-14B	few shot	81.73	**4.28**	0.63
		zero shot	76.33	9.21	3.91
tail	Llama-3.1-8B	few shot	69.10	18.52	1.17
		zero shot	57.14	36.29	1.66
	Llama-3.3-70B	few shot	79.19	7.50	0.19
		zero shot	66.30	22.51	8.72
	Phi-4-mini	few shot	70.81	8.89	8.87
		zero shot	70.34	14.04	27.06
	Phi-4	few shot	77.48	7.96	10.53
		zero shot	61.18	29.58	11.41
	Qwen2.5-7B	few shot	73.29	8.20	2.03
		zero shot	70.81	10.62	23.05
	Qwen2.5-14B	few shot	**80.12**	**4.89**	**0.17**
		zero shot	73.60	8.60	16.09

Table 11: Model performance on the TTQA subset. The best performance per metric and split is bold.

Split	Model	Prompting	EM (↑)	sMAPE (↓)	MASE (↓)
arithmetic	Llama-3.1-8B	few shot	20.57	23.66	2.23
		zero shot	14.47	35.80	2.03
	Llama-3.3-70B	few shot	49.70	10.54	**0.15**
		zero shot	41.34	13.53	0.37
	Phi-4-mini	few shot	26.38	21.19	1.00
		zero shot	19.59	38.37	3.47
	Phi-4	few shot	55.71	**7.12**	0.17
		zero shot	**58.56**	9.03	0.16
	Qwen2.5-7B	few shot	31.00	20.85	1.06
		zero shot	25.30	34.11	0.46
	Qwen2.5-14B	few shot	43.21	9.21	0.71
		zero shot	44.00	10.88	0.52
semantic	Llama-3.1-8B	few shot	72.69	10.43	1.18
		zero shot	67.11	14.19	11.34
	Llama-3.3-70B	few shot	89.43	4.61	**0.11**
		zero shot	**90.16**	6.09	0.19
	Phi-4-mini	few shot	65.64	10.54	1.04
		zero shot	56.68	17.64	1.27
	Phi-4	few shot	83.55	**4.01**	0.32
		zero shot	80.32	6.39	0.46
	Qwen2.5-7B	few shot	63.00	8.47	1.19
		zero shot	59.32	11.57	1.27
	Qwen2.5-14B	few shot	72.98	8.79	0.50
		zero shot	67.99	11.72	0.61

Table 12: Model performance on the ToT subset. The best performance per metric and split is bold.

		# of defined errors		
Model	Prompting	EM	sMAPE	MASE
Llama-3.1-8B	Few shot	1737	1373	1530
	Zero shot	1737	1373	1225
Llama-3.3-70B	Few shot	1737	1373	1667
	Zero shot	1737	1373	1417
Phi-4-mini	Few shot	1737	1373	1722
	Zero shot	1737	1373	1668
Phi-4	Few shot	1737	1373	1680
	Zero shot	1737	1373	1341
Qwen2.5-7B	Few shot	1737	1373	1706
	Zero shot	1737	1373	1708
Qwen2.5-14B	Few shot	1737	1373	1727
	Zero shot	1737	1373	1700

Table 13: Number of QA-pairs of the TTQA dataset for which each metric is defined. EM is defined for each question. sMAPE is not defined for dates and is set to 100% if errors are not parsable. MASE is defined for all questions but is not defined if the model's answer is not parsable.

Model	Prompting	# of defined errors		
		EM	sMAPE	MASE
Llama-3.1-8B	Few shot	1697	1369	1575
	Zero shot	1697	1369	1527
Llama-3.3.70B	Few shot	1697	1369	1581
	Zero shot	1697	1369	1524
Phi-4-mini	Few shot	1697	1369	1618
	Zero shot	1697	1369	1438
Phi-4	Few shot	1697	1369	1640
	Zero shot	1697	1369	1595
Qwen2.5-7B	Few shot	1697	1369	1600
	Zero shot	1697	1369	1499
Qwen2.5-14B	Few shot	1697	1369	1660
	Zero shot	1697	1369	1648

Table 14: Number of QA-pairs of the ToT dataset for which each metric is defined. EM is defined for each question. sMAPE is not defined for dates and is set to 100% if errors are not parsable. MASE is defined for all questions but not defined if the model's answer is not parsable.

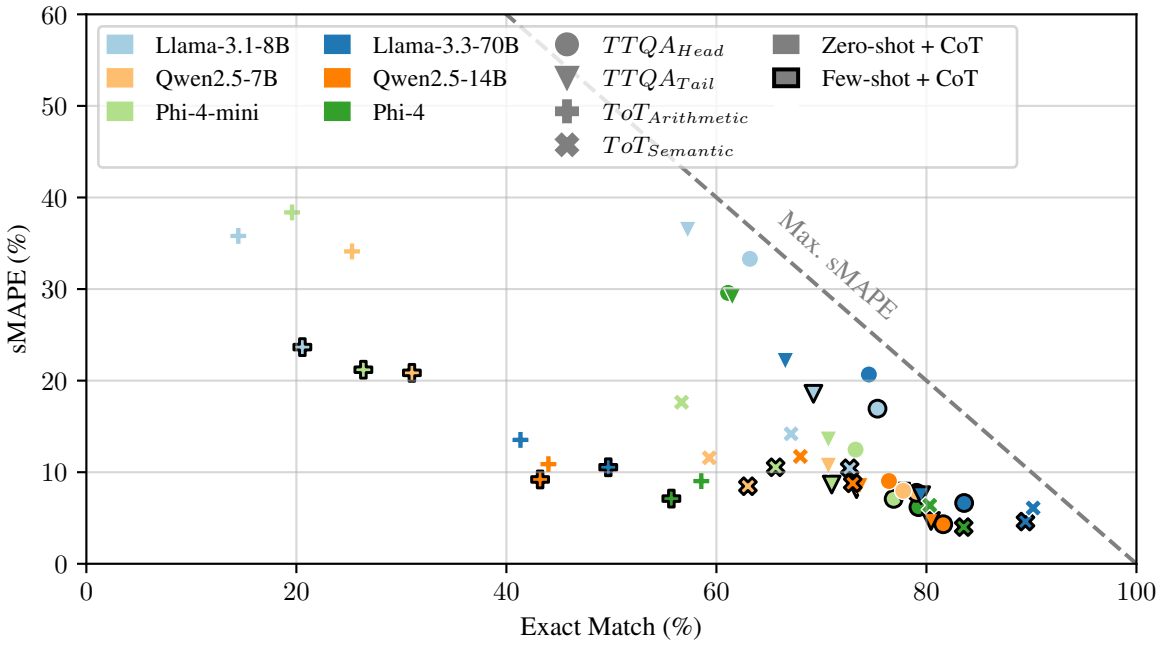


Figure 11: Comparison of performance measured by sMAPE and EM.

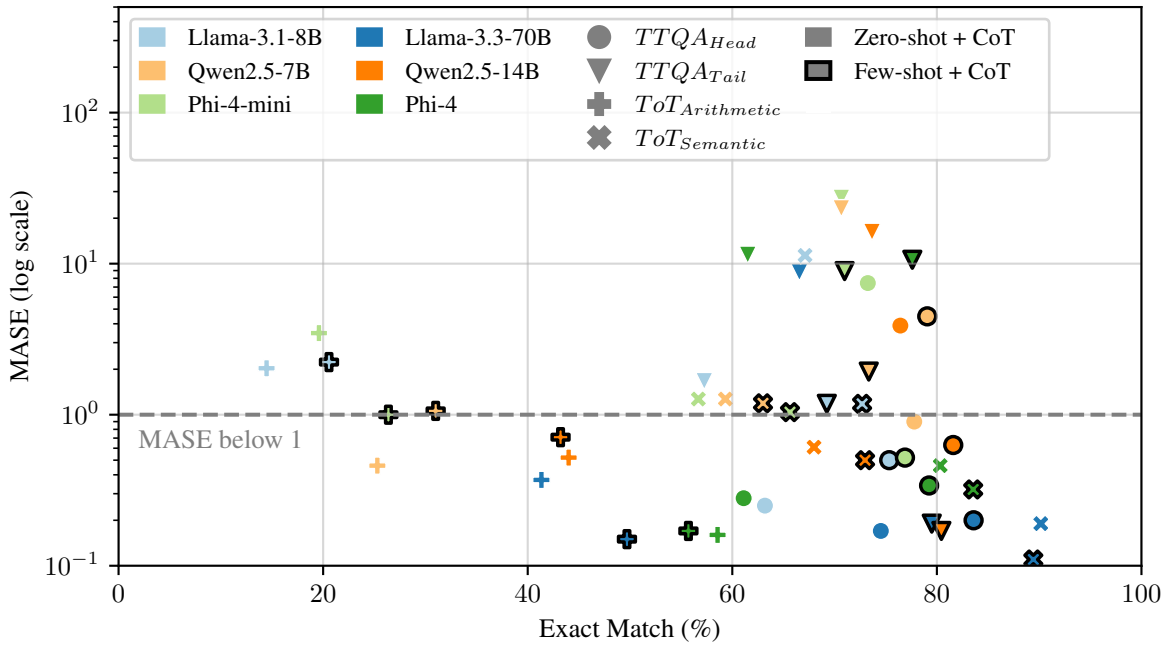


Figure 12: Comparison of performance measured by MASE and EM.