When Do Answers Change? Estimating Question Recency Demands in QA

Anonymous ACL submission

Abstract

Language Large language models (LLMs) often rely on outdated knowledge for time-sensitive questions, producing confident but incorrect answers. Without a way to check if newer information is needed, the systems struggle with deciding when to retrieve evidence, how to handle outdated facts, and how to rank answers based on freshness. Existing benchmarks either refresh answers or use fixed templates, but they do not provide a means to track how often answers change or whether a question requires up-to-date information. To address this, we introduce a recency-stationarity taxonomy that categorizes questions based on how frequently their answers are expected to change. We then present RECENCYQA, a dataset of 6,115 open-domain questions, each annotated with a recency label and a stationarity label. The recency label indicates how often the answer changes, while stationarity defines whether this change frequency remains stable over time (time-invariance) or depends on context such as "turning points" events (context-dependence). Additionally, each question includes a temporal context that explains when the recency label applies. We evaluate RECENCYQA through human assessment and empirical experiments. Our results show that questions labeled as non-stationary, those needing fresh information, are more challenging for LLMs, especially as update frequency increases.

1 Introduction

Large language models (LLMs) have demonstrated strong performance on a wide range of question answering (QA) tasks (Brown et al., 2020; Chang et al., 2024). However, their ability to handle *temporal questions*, i.e., questions whose answers change over time, remains limited. Most existing QA benchmarks treat answers as static facts (Joshi et al., 2017; Kwiatkowski et al., 2019), ignoring the reality that many real-world questions are inherently dynamic. For instance, *What is the interest rate set by the European Central Bank?* or *Which companies are currently included in the CSI 300 Index?* require temporally grounded answers that reflect the current state of the world rather than snapshots fixed at a models training cutoff.

Recent research in temporal QA has sought to address these challenges but leaves crucial gaps. Benchmarks such as TimeQA (Chen et al., 2021), TEMPLAMA (Dhingra et al., 2022), RealTimeQA (Kasai et al., 2023), FreshQA (Vu et al., 2024), and PATQA (Meem et al., 2024) introduce time-sensitive questions or periodically refresh their answers. While these efforts focus on temporal validity, they generally treat *recency* as a binary property distinguishing between current and outdated information rather than modeling how frequently an answer is expected to change. Moreover, datasets like TempReason (Tan et al., 2023) and CRON-QUESTIONS (Chen et al., 2023) focus on temporal reasoning chains or event sequences but do not quantify the rate at which information becomes outdated. As a result, existing work lacks a principled framework for estimating the *recency demand* of a question or for distinguishing between questions whose temporal sensitivity is fixed versus context-dependent.

To address these limitations, we introduce a new framework for *recency-aware question answering*. At its core is a **recency-stationarity taxonomy** that characterizes questions along two dimensions: (i) the expected update frequency of their answers, ranging from hourly to permanent facts, and (ii) whether their temporal sensitivity remains constant (**stationarity**) or changes with time or context (**non-stationarity**). This distinction captures not only how fast information evolves but also if the rate of change of a question's temporal relevance is stable.

Understanding recency and stationarity has important implications for modern LLM systems.

Dataset	Creation	KC	Multi-hop	Recency-Label	# Ques.
TimeQA (Chen et al., 2021)	Templ.-Wikidata	Wikipedia	✗	✗	20k
SituatedQA (Zhang and Choi, 2021)	Man.-Filt.	Wikipedia	✗	✗	12k
TempLama (Dhingra et al., 2022)	Templ./Cloze	Custom-News	✗	✗	50k
StreamingQA (Liska et al., 2022)	Man.+Gen.	WMT News	✓	✗	410k
ArchivalQA (Wang et al., 2022)	Gen.	NYT Articles	✗	✗	532k
ChroniclingAmericaQA (Piryani et al., 2024)	Gen.	Chronicling America Newpaers	✗	✗	485k
RealTimeQA (Kasai et al., 2023)	News websites	News Articles	✓	✗	~5k
PATQA (Meem et al., 2024)	Templ.-wikidata	Wikipedia	✓	✗	6,172
FreshQA (Vu et al., 2024)	Man.	Google Search	✓	✗	600
RecencyQA (ours)	Man.-Filt.+Gen	Wikipedia/Wikidata	✓	✓	6,115

Table 1: Comparison of temporal question answering datasets. Abbreviations: Man.=created manually, Gen.=Automatically generated, Man.-Filt.=filtered from other datasets, Man.+Gen.=created by crowdsourcing and LLM generation, Templ.=created using templates, Man.-Filt.+Gen=filtered from other datasets and LLM generation, KC=Knowledge Corpus.

First, retrieval-augmented generation (RAG) models lack principled criteria for deciding when to rely on internal parametric knowledge versus external retrieval (Lewis et al., 2020; Mallen et al., 2023). Second, models often exhibit misplaced confidence, expressing equal certainty about stable facts and fast-changing events (Kadavath et al., 2022), leading to *temporal hallucinations* (Wallat et al., 2025). Third, retrieval systems must balance topical relevance against temporal freshness, as outdated yet relevant documents can be more misleading than current but less relevant ones (Campos et al., 2014; Piryani et al., 2025). Finally, temporal reasoning often requires combining multiple pieces of evidence across time (multi-hop) rather than relying on a single fact (single-hop) (Tan et al., 2024; Meem et al., 2024), which further complicates when and how models should update their answers.

We evaluate several state-of-the-art LLMs under different prompting strategies, zero-shot, few-shot, and chain-of-thought, and analyze how recency and stationarity affect their performance. Our results show systematic drops in accuracy for questions with high recency demands and reveal that standard prompting strategies can amplify temporal hallucinations when models rely on outdated knowledge.

Contributions. To our knowledge, this paper is the first to formalize recency in QA as an explicit, operational taxonomy. Specifically:

- **Recency-stationarity taxonomy.** We propose a two-dimensional taxonomy that (i) categorizes questions by the *expected update frequency* of their answers and (ii) distinguishes *stationary* from *non-stationary* questions based on whether their recency needs are time-invariant or context-dependent. We operationalize this framework through a scalable LLM-assisted annotation pipeline with human validation.
- **RecencyQA Dataset.** We introduce RecencyQA, the first open-domain dataset annotated for expected answer update frequency and (non-)stationarity, spanning both single-hop and multi-hop questions.
- **Empirical Evaluation.** We benchmark several large language models under zero-shot, few-shot, and chain-of-thought prompting, demonstrating how recency and stationarity jointly influence model accuracy and temporal reliability.

2 Related Work

Early QA benchmarks such as TriviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), WebQ (Berant et al., 2013) assume static factual knowledge, overlooking that many real-world answers evolve over time. Temporal QA datasets such as TimeQA (Chen et al., 2021), ArchivalQA (Wang et al., 2022), ChroniclingAmericaQA (Piryani et al., 2024) and SituatedQA (Zhang and Choi, 2021) introduced timestamped or context-anchored questions to assess models temporal reasoning. Later efforts, such as StreamingQA (Liska et al., 2022), RealTimeQA (Kasai et al., 2023), PATQA (Meem et al., 2024), and FreshQA (Vu et al., 2024), have emphasized answer freshness by continually updating data or sourcing recent news. However, these benchmarks usually frame recency as a binary distinction between current and outdated information, without modeling the expected rate at which answers change.

Work on temporal validity and drift examines how information and model behavior change over time. Zhao et al. (2022) studied explanation degradation under evolving data, while Wenzel and Jatowt (2024) predicted when factual statements become outdated. Dhingra et al. (2022) modeled LMs as temporal knowledge bases, and Yang et al. (2020) predicted fact duration. These studies focus on fact-level temporal dynamics, not on characterizing question types by their expected update frequency, a distinction our taxonomy explicitly models.

The IR community has long explored freshness and temporal intent. The Temporalia benchmark series (Joho et al., 2014, 2016) and related works (Li and Croft, 2003; Berberich et al., 2010; Styskin et al., 2011) classify queries as Past, Recency, Future, or Atemporal and apply recency-sensitive ranking to time-critical information needs. Similarly, RAG methods (Lewis et al., 2020; Izacard and Grave, 2021) mitigate factual staleness by augmenting LLMs with external evidence.

Across temporal QA, drift prediction, and recency-aware retrieval, existing efforts advance temporal evaluation but treat time sensitivity as discrete or reactive. Our work introduces the Recency-Stationarity Taxonomy, which models (i) how frequently an answer is expected to change and (ii) whether that pattern is context-dependent. This fine-grained, question-centered and proactive perspective offers a scalable foundation for analyzing temporal reliability and recency awareness in LLMs. While prior temporal QA datasets emphasize when information is valid, they do not capture how often the answer changes. `RECENCYQA` complements these efforts by providing fine-grained recency and stationarity annotations, allowing temporal sensitivity to be measured as a continuous property rather than a binary label. Table 1 summarizes key differences between `RECENCYQA` and existing temporal QA benchmarks.

3 Taxonomy

Temporal dynamics play a crucial role in question answering (QA), as the validity of an answer often depends on when the question is asked. Prior studies have explored temporal aspects of information and reasoning, such as temporal drift in facts (Zhao et al., 2022), temporal validity prediction (Wenzel and Jatowt, 2024), and the challenges of evaluating models on time-sensitive questions (Tan et al., 2024; Liska et al., 2022). However, most existing QA benchmarks still assume that answers are static and remain correct indefinitely, overlooking the fact that many real-world questions are inherently time-dependent.

Recency Class	**Expected Time Until Answer Change**
An-Hour	Within an hour
A-Few-Hours	Within a few hours
A-Day	Within a day
A-Few-Days	Within a few days
A-Week	Within a week
A-Few-Weeks	Within a few weeks
A-Month	Within a month
A-Few-Months	Within a few months
A-Year	Within a year
A-Few-Years	Within a few years
Many-Years	After many years
Never	Not expected to change

Table 2: The proposed Recency Taxonomy consists of twelve classes, ordered from highly volatile (top) to temporally stable (bottom). Each class reflects the expected time until a questions answer first changes.

We introduce the notion of recency to describe the expected temporal stability of a questions answer that is, how soon the answer is anticipated to change for the first time. For example, the answer to *"Who is the president of the United States?"* may change within a few years, while the answer to *"What is the chemical symbol for gold?"* is unlikely to ever change.

Closely related to recency is the concept of stationarity, which we define in terms of the stability of the recency label itself, rather than the answer. A question is stationary if its expected rate of change (i.e., its recency label) remains consistent regardless of when the question is asked. For instance, *"How can individuals register to attend the World Technology Summit?"* is stationary, since it always belongs to the *"A-Year"* class - the summit is held regularly every year, and the recency expectation remains consistent across time. Conversely, a non-stationary question exhibits variability in its recency label depending on the temporal context or framing. For example, *"Who is currently leading the medal table at the Olympic Games?"* is non-stationary, because its recency label changes depending on when the question is asked, during the event - it may require hourly or daily updates as competitions progress, and after the event, the updates are not required until the next games.

In this view, recency captures how quickly an answer changes, whereas stationarity captures how consistently that rate of change is expected to hold.

To operationalize this concept, we propose

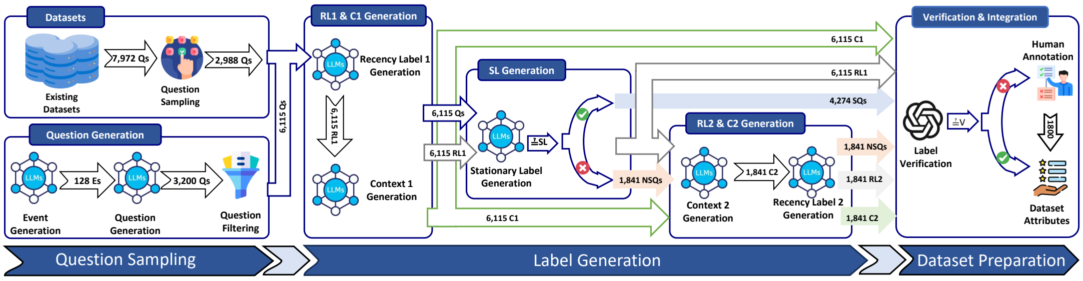


Figure 1: **The pipeline of RECENCYQA Generation.** (1) Question Sampling: First, 2,988 questions are sampled from existing benchmarks; second, 3,200 synthetic questions are generated based on LLM-produced events. (2) Label Generation: For each question, an initial Recency Label (RL_1) is assigned. Based on RL_1, a corresponding Context (C_1) is generated. Each question-recency label pair (Q, RL_1) is then assigned a stationarity label. Stationary questions (SQs) are sent for Verification and Integration. Non-stationary questions (NQs) proceed to a second round of context and recency labeling. For each non-stationary question, a second context (C_2) is generated based on RL_1 and C_1. A second recency label (RL_2) is then assigned for non-stationary question and C_2. The generated contexts (C_1, C_2) and corresponding recency labels (RL_1, RL_2) are sent for verification. (3) Dataset Preparation: During verification, each recency label is validated against its associated context. Incorrect labels are further reviewed and corrected by human annotators. Finally, all verified attributes are stored in the RECENCYQA dataset. *Note:* Qs represents Questions, SL represents Stationary Label, SQs represents Stationary Questions, and NQs represents Non-stationary Questions.

a Recency-Stationarity Taxonomy that describes questions along two levels. The first level defines twelve recency classes, representing the expected time until a questions answer changes (listed in Table 2). The second level, stationarity, characterizes whether this recency label itself remains stable over time. In this view, Table 2 presents the recency labels forming one axis of the taxonomy, while stationarity determines whether a questions temporal behavior is consistent or context-dependent.

This taxonomy provides a principled framework to quantify temporal volatility, distinguishing between stationary and non-stationary questions along a continuous spectrum. It serves as the foundation for our subsequent analyses of recency awareness, temporal reliability, and factual stability in LLMs.

4 Dataset Construction

This section describes the pipeline we used to construct the RECENCYQA dataset. The process involves three main stages: (1) question sampling, (2) label generation, and (3) dataset preparation, as illustrated in the Figure 1. We use LLaMA 3.3 70B (Grattafiori et al., 2024) as the generation backbone, as it is a highly capable open-source LLM available on HuggingFace[1] that allows users to freely reproduce RECENCYQA pipeline. While GPT-4o (Achiam et al., 2023) served as the final verification model.

4.1 Question Sampling

To construct a diverse and temporally grounded set of questions for the RECENCYQA dataset, we combined two complementary sources: Existing temporal QA datasets and synthetic event-based questions generated by an LLM.

Sampling from Existing QA Datasets. We first sampled 2,988 questions from three publicly available QA datasets: FreshQA (Vu et al., 2024), PATQA (Meem et al., 2024), and SituatedQA (Zhang and Choi, 2021). These datasets were chosen because they explicitly target temporal reasoning and align closely with our task.

FreshQA is designed to test LLMs on both fast-changing and slow-changing information and is supposed to be updated every few months. From this dataset, we selected all questions labeled with a "true premise," yielding a total of 453 questions.

PATQA anchors questions to the present time, enabling automatic answer updates through structured knowledge bases. We sampled 50 questions from each of its relation types, obtaining 1,151 multi-hop and 453 single-hop questions.

SituatedQA evaluates context-dependent temporal reasoning; we used its full test set of 931 questions. Together, these sources contributed 2,988 questions representing a wide range of temporal dynamics and reasoning types.

[1] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct

Event-Based Question Generation To increase coverage of temporal events beyond existing datasets, we prompted LLAMA 3.3 (70B) to generate 128 short textual event descriptions. Each event served as a seed for producing 25 time-sensitive questions, resulting in approximately 3,200 synthetic questions. After generation, we filtered out duplicate questions, removing 73 questions. The prompts for event and question generation are shown in Figure 5 and 7 in the AppendixA.

Combining sampled and generated questions resulted in a total of 6,115 questions.

4.2 Label Generation

The second stage corresponds to label generation, where recency and stationarity labels are assigned to each question and contextual information is generated.

Recency Label Annotation. Each question was first annotated with a **recency label** (RL_1) indicating how frequently its answer is expected to change, following the taxonomy defined in Section 3. We used LLAMA 3.3 (70B) in a zero-shot setup to assign one of twelve predefined recency classes based on the expected rate of change of answer. The full prompt is provided in Figure 4 (Appendix A).

Context Generation. For each question and its assigned recency label, LLAMA 3.3 (70B) generated a short contextual description (C_1) that justifies the label and situates the question in time. This ensures that each assigned label is semantically grounded in a plausible temporal scenario.

Stationarity Annotation. Next, we determined whether each questions recency label remains constant over time. Using LLAMA 3.3 (70B), questions were classified as *stationary* (the same recency label applies regardless of when the question is asked) or *non-stationary* (the recency label varies across contexts). Out of 6,115 questions, 4,274 were labeled as stationary and 1,841 were labeled as non-stationary. The prompt for this step is shown in Figure 8 (Appendix A).

Non-Stationary Re-Labeling For questions annotated non-stationary, the model was prompted to generate an alternative context (C_2) based on context (C_2) and Recency Label (RL_1) where the recency expectation differs, followed by a second recency label (RL_2) assigned to that new context (C_2). This procedure allows the dataset to capture how recency demands shift under changing temporal frames. Figure 12 in Appendix B illustrates the recency label transitions from RL_1 to RL_2 for all non-stationary questions, revealing how recency labels get altered along with context change. The prompts for second context and recency label generation are provided in Figures 10 and 11 in Appendix A.

4.3 Data Preparation

The final stage involves verification, human review, and dataset consolidation.

Label Verification and Human Review. All generated contexts (C_1, C_2) and labels (RL_1, RL_2) were first evaluated automatically using GPT-4o (Achiam et al., 2023). The verification prompt is shown in Figure 6. Out of 6,115 questions, the model flagged 1,800 as having potentially incorrect labels. These cases were manually reviewed by human annotators, who verified and corrected the recency labels to ensure the high quality of the dataset. Additionally, a random sample of 100 questions from correctly verified questions was evaluated by humans to ensure the quality of the generated dataset.

Final Compilation After verification, all validated questions, contexts, and labels were consolidated into the final RECENCYQA dataset, which contains 6,115 questions with corresponding recency labels, contexts, and stationarity annotations. This dataset serves as the foundation for the analyses and experiments presented in subsequent sections.

5 Dataset Analysis

In this section, we analyze the RECENCYQA dataset from two perspectives: *Dataset Statistics* and *Human Evaluation*.

5.1 Dataset Statistics

The final RECENCYQA dataset contains 6,115 temporally grounded questions generated through the hybrid pipeline in Figure 1. Each question is annotated with a *recency label* (how often its answer changes), a *stationarity label* (whether that change pattern is context-dependent), and one or more *context sentences* grounding the assigned labels. The dataset includes 4,849 *multi-hop* and 1,266 *single-hop* questions, reflecting both simple and compositional temporal reasoning. Table 3 presents key statistics of the dataset, and Figure 2 illustrates the distribution of recency classes.

Metric	Value
Total Recency classes	12
Total Questions	6,115
Total Recency Labels	7,956
Avg. Question Length (words)	14.26
Avg. Context 1 Length (words)	22.22
Avg. Context 2 Length (words)	22.35
Total Stationary Question	4,274
Total Non-Stationary Question	1,841
Total Singlehop Questions	4,849
Total Multihop Questions	1,266

Table 3: Statistical Summary of the RECENCYQA Dataset

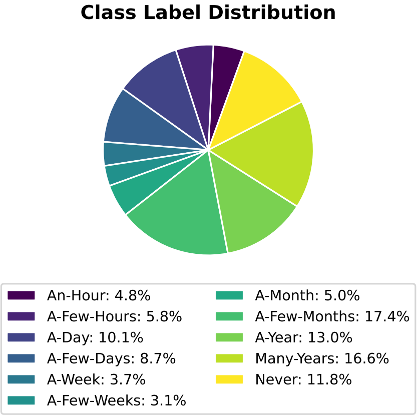


Figure 2: The distribution of class labels in RECENCYQA.

	Accuracy	Tolerant Accuracy
Recency	0.62	0.81
Stationarity	0.74	–
Human Rating Averages (1- 5 scale)		
Clarity of Question	4.66	
Difficulty to Label	2.26	
Difficulty to Answer	2.41	

Table 4: Human evaluation results for the RECENCYQA dataset. The upper section reports labeling accuracies, and the lower section reports mean human ratings (1-5 scale). Recency accuracy is shown as strict and tolerant (within ±1 bin).

5.2 Human Evaluation

We conducted a human evaluation of the dataset to assess whether the labels generated by LLMs, given contextual information, align with human judgments. Six graduate students (3 male, 3 female) served as annotators for this evaluation.

We selected a balanced sample of 360 questions from the dataset, ensuring an even distribution across all recency labels. The selected questions were divided into six sets of 60 questions each, with three annotators independently evaluating every set. For each question, annotators were asked to (i) assign a *recency* label based on the provided context, (ii) assign a *stationarity* label, (iii) rate the *clarity* of the question, (iv) rate the *difficulty of assigning a recency label*, and (v) rate the *difficulty of answering* the question itself. Annotators were asked to provide ratings on a 5-point scale. For clarity, a score of 1 indicates that the question was not clear, and 5 indicates it was very clear. For the difficulty ratings, 1 denotes that the task was not difficult, and 5 denotes it was very difficult.

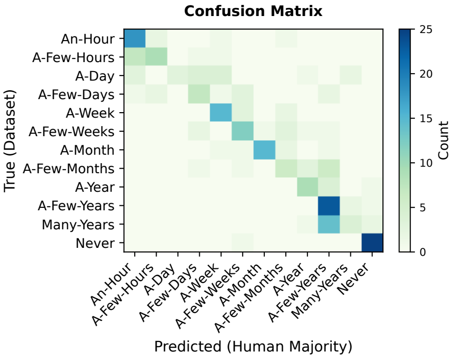


Figure 3: Confusion matrix for Recency labels comparing dataset labels vs human majority labels (predicted).

To analyze the results, we used the majority vote among annotators as the final human label for both recency and stationarity. These aggregated labels were compared with the corresponding dataset labels to compute accuracy. Table 4 summarizes the quantitative results, while Figure 3 shows the confusion matrix comparing dataset recency labels with the human majority labels. The dataset labels show strong alignment with human judgments for both recency and stationarity, demonstrating the reliability of our annotation pipeline. The clarity ratings are generally high, indicating that most questions were well-posed. The moderate difficulty scores suggest that annotators found the tasks reasonably straightforward to complete.

6 Experiments

In this section, we present a comprehensive analysis of the RECENCYQA dataset. Our goal is to evaluate how well large language models (LLMs) can classify the recency level of a question under different prompting paradigms. Specifically, we compare performance across several LLMs using zero-shot, few-shot, and chain-of-thought (CoT) prompting.

Model	# of Parameters	Accuracy	Tolerant Accuracy	F1 Score	Tolerant F1
Zero-shot					
Qwen 2.5	7b	0.28	0.63	0.23	0.65
Mistral	7b	0.26	0.58	0.22	0.56
Gemma 2	27b	**0.38**	0.68	**0.37**	0.69
Mistral	8x7B	0.34	0.71	0.32	0.72
Qwen 2.5	72b	0.37	**0.75**	0.32	**0.77**
Few-shot					
Qwen 2.5	7b	0.42	0.69	0.40	0.70
Mistral	7b	0.37	0.66	0.35	0.65
Gemma 2	27b	0.49	0.76	0.46	0.77
Mixtral	8x7b	0.37	0.72	0.31	0.79
Qwen 2.5	72b	**0.52**	**0.77**	**0.51**	**0.80**
Chain-of-Thought					
Qwen 2.5	7b	0.35	0.61	0.33	0.61
Mistral	7b	0.34	0.66	0.32	0.70
Gemma 2	27b	0.28	0.56	0.27	0.55
Mixtral	8x7b	0.40	0.73	0.39	0.74
Qwen 2.5	72b	**0.42**	**0.75**	**0.42**	**0.76**
Fine-tuned Model					
RoBerta-large	330m	**0.64**	**0.81**	**0.63**	**0.80**

Table 5: Performance comparison of different LLMs and fine-tuned baselines on recency classification across different prompting paradigms (zero-shot, few-shot, and chain-of-thought). Bold values indicate the best results within each prompting strategy, while underlined values represent the overall best performance across all approaches for each metric.

6.1 Experimental Setup

We conduct experiments using a diverse set of open-source LLMs spanning different model families and parameter scales. The models include **Qwen2.5-7B** and **Qwen2.5-72B** (Qwen et al., 2024), **Gemma-2-27B** (Team et al., 2024), **Mixtral-8x22B** (Jiang et al., 2024) and **Mixtral-7B** (Jiang et al., 2023). This selection enables us to evaluate both smaller and larger models.

To ensure computational feasibility, we sample a balanced subset of **1,200 questions** from RECENCYQA. This subset covers all 12 recency classes uniformly and preserves the original ratio of *stationary* and *non-stationary* questions. The remaining portion of the dataset is reserved for fine-tuning experiment. This split enables us to (i) assess the zero-shot, few-shot, and chain-of-thought reasoning ability of pretrained models on unseen questions, and (ii) later measure how much targeted supervision on recency classification improves temporal sensitivity.

6.2 Evaluation Metrics

We evaluate model performance using standard classification metrics, including **Accuracy (ACC)** and **F1 Score (F1)**. In addition, we report **Tolerant Accuracy** and **Tolerant F1**, which consider predictions as partially correct when the predicted recency class is adjacent to the ground-truth label. This tolerant evaluation provides a more realistic measure of model performance, accounting for the gradual and often overlapping nature of temporal boundaries between recency levels.

7 Results

7.1 Impact of Prompting Paradigms and Finetuning

Table 5 presents the results of our first experiment, where we evaluate different prompting strategies on the full evaluation subset of 1,200 questions from RECENCYQA. Each model predicts one of the 12 recency classes for every question under three prompting paradigms: zero-shot, few-shot, and chain-of-thought (CoT).

As shown in Table 5, larger models generally perform better across all prompting setups, confirming that model scale improves temporal reasoning and understanding of recency patterns. In the zero-shot setup, **Gemma-2-27B** achieves the highest strict accuracy, while **Qwen2.5-72B** obtains the best tolerant results.

When few-shot examples are introduced, all models show substantial gains, with **Qwen2.5-72B** again achieving the best overall performance. *This demonstrates that providing a small number of examples helps models better align with the notion of recency frequency.*

Interestingly, chain-of-thought prompting does not consistently improve results. While some models such as **Mixtral-8x7B** and **Qwen2.5-72B** benefit slightly, others like **Gemma-2-27B** experience performance drops. This suggests that explicit reasoning chains are not always beneficial for categorical temporal classification, which relies more on recognizing time-related cues than step-by-step logical reasoning.

Finally, fine-tuned baselines significantly outperform all prompting setups. The **RoBERTa-large** model achieves 0.64 Accuracy and 0.81 Tolerant Accuracy, showing that direct supervision on recency labels leads to much more reliable temporal understanding than in-context prompting alone.

Additionally, we evaluate all three prompting strategies both with and without context across all models. The confusion matrices for all approaches are shown in Figures 13, 14, and 15 in Appendix B.

7.2 Effect of Context

To evaluate how context affects recency classification, we tested all models on stationary and non-stationary questions both **with** and **without** additional temporal context. In the *context-free* con-

Model	# of Parameters	Stationary (Accuracy)			Stationary (F1-score)			Non-stationary (Accuracy)			Non-stationary (F1-score)		
		w/ C	w/o C	Δ%	w/ C	w/o C	Δ%	w/ C	w/o C	Δ%	w/ C	w/o C	Δ%
					Zero-shot Prompting								
Qwen 2.5	7b	0.225	0.248	**9.3%↓**	0.206	0.220	**6.4%↓**	0.293	0.352	**16.8%↓**	0.164	0.221	**25.8%↓**
Mistral	7b	0.189	0.217	**12.9%↓**	0.200	0.221	**9.5%↓**	0.340	0.330	**+3.0%↑**	0.229	0.179	**+27.9%↑**
Gemma 2	27b	0.324	0.347	**6.6%↓**	0.317	0.357	**11.2%↓**	0.373	0.452	**17.5%↓**	0.260	0.310	**16.1%↓**
Mistral	8×7b	0.249	0.335	**25.7%↓**	0.235	0.333	**29.4%↓**	0.410	0.355	**+15.5%↑**	0.290	0.251	**+15.5%↑**
Qwen 2.5	72b	0.296	0.319	**7.2%↓**	0.294	0.281	**+4.6%↑**	0.468	0.449	**+4.2%↑**	0.368	0.338	**+8.9%↑**
					Few-shot Prompting								
Qwen 2.5	7b	0.468	0.452	**+3.5%↑**	0.454	0.422	**+7.6%↑**	0.422	0.362	**+16.6%↑**	0.339	0.309	**+9.7%↑**
Mistral	7b	0.380	0.414	**8.2%↓**	0.346	0.381	**9.2%↓**	0.339	0.290	**+16.9%↑**	0.214	0.257	**16.7%↓**
Gemma 2	27b	0.573	0.521	**+10.0%↑**	0.526	0.474	**+11.0%↑**	0.516	0.426	**+21.1%↑**	0.472	0.361	**+30.8%↑**
Mistral	8×7b	0.352	0.373	**5.6%↓**	0.300	0.326	**8.0%↓**	0.406	0.355	**+14.4%↑**	0.243	0.222	**+9.5%↑**
Qwen 2.5	72b	0.528	0.523	**+1.0%↑**	0.519	0.494	**+5.1%↑**	0.581	0.518	**+12.2%↑**	0.488	0.449	**+8.7%↑**
					Chain-of-Thought (CoT) Prompting								
Qwen 2.5	7b	0.33	0.35	**5.7%↓**	0.30	0.34	**11.8%↓**	0.32	0.33	**3.0%↓**	0.20	0.25	**20.0%↓**
Mistral	7b	0.34	0.32	**+6.3%↑**	0.34	0.33	**+3.0%↑**	0.40	0.36	**+11.1%↑**	0.31	0.25	**+24.0%↑**
Gemma 2	27b	0.33	0.29	**+13.8%↑**	0.33	0.29	**+13.8%↑**	0.32	0.25	**+28.0%↑**	0.25	0.16	**+56.3%↑**
Mistral	8×7b	0.39	0.41	**4.9%↓**	0.40	0.39	**+2.6%↑**	0.45	0.38	**+18.4%↑**	0.32	0.28	**+14.3%↑**
Qwen 2.5	72b	0.41	0.42	**2.4%↓**	0.41	0.43	**4.7%↓**	0.47	0.42	**+11.9%↑**	0.36	0.31	**+16.1%↑**

Table 6: Effect of context on stationary and non-stationary question performance across prompting strategies. Arrows (↑ / ↓) and colored percentages indicate improvement or degradation.

dition, models receive only the question; in the *context-aware* condition, we additionally provide a short snippet clarifying the temporal background (as described in Section 4).

Table 6 compares performance with (**w/ C**) and without (**w/o C**) context across stationary and non-stationary questions for all prompting strategies. Context generally *reduces* accuracy for stationary questions but *improves* it for non-stationary ones. For example, **Mixtral-8×7B** drops by -25.7% on stationary accuracy but gains $+15.5\%$ on non-stationary, while **Qwen2.5-72B** shows smaller stationary declines yet consistent improvements on dynamic items. This indicates that for stationary questions, where recency is independent of query time, providing context can confuse the model by introducing unnecessary temporal signals.

Adding few-shot examples stabilizes model behavior: context becomes broadly beneficial, especially for non-stationary questions. For instance, **Gemma-2-27B** improves by $+21.1\%$ in accuracy and $+30.8\%$ in F1 on non-stationary cases, suggesting that limited supervision helps models learn when temporal cues should be trusted.

CoT prompting yields mixed effects: context often lowers stationary performance but can strongly boost non-stationary results (e.g., **Gemma-2-27B**: $+56.3\%$ non-stationary F1). Reasoning chains thus amplify relevant temporal cues when change is expected but introduce noise when no temporal evolution is needed.

Analysis. The results reveal consistent asymmetries in how LLMs interpret temporal context and reasoning cues. For stationary questions, additional temporal context often reduces accuracy because models over-attend to explicit time expressions (e.g., as of now, currently), leading them to reinterpret stable facts as time-dependent. In contrast, non-stationary questions benefit from temporal grounding: contextual snippets help anchor interpretation to the correct timeframe, mitigating overreliance on outdated parametric knowledge.

Chain-of-thought prompting amplifies whichever temporal bias dominates the models internal knowledge. For stable facts, CoT reinforces outdated reasoning and reduces accuracy; for dynamic cases, it helps reason through expected changes, yielding large gains.

Overall, these findings suggest that current LLMs lack *temporal selectivity*-the ability to determine when time matters. They treat every temporal cue as equally informative rather than learning to ignore or exploit it based on question stability. Developing mechanisms for *context gating* or *recency calibration* may therefore be critical for future recency-aware QA and RAG systems.

8 Conclusion

We introduced RECENCYQA, a benchmark and taxonomy for estimating how often answers change and whether their temporal behavior is stable. Our analyses show that LLMs perform well on temporally stable questions but falter when recency demands increase. Context helps in dynamic settings yet often harms stationary ones, revealing that models lack temporal selectivity. This work provides a foundation for recency-aware QA, encouraging future systems that can decide when to trust memory and when to seek new information.

Limitations

Our study has several limitations. First, the RECENCYQA dataset focuses on open-domain textual QA and does not include multimodal or domain-specific sources, where temporal dynamics may differ substantially. Second, the labeling pipeline relies on LLM-assisted annotation followed by human validation, which, while scalable, may inherit biases from the underlying models temporal priors and knowledge cutoff. Third, our work is limited to English-language questions, and the generalizability of our taxonomy and findings to multilingual or low-resource settings remains untested. Finally, the pipeline depends on current LLMs, whose behavior and biases can affect both labeling quality and model evaluation outcomes, potentially influencing fairness and reproducibility.

Ethical Considerations

This study uses GPT models licensed under OpenAI and Apache 2.0, as well as LLaMA models governed by Metas LLaMA Community License Agreement. All model and data usage complies with the respective licensing terms. The datasets employed are derived from publicly available sources that permit academic research use. To promote transparency and reproducibility, we release our code and processed data under the MIT license. Throughout the project, we adhered to institutional ethical standards and applicable legal requirements regarding data handling, model usage, and dissemination of results.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Klaus Berberich, Srikanta Bedathur, Omar Alonso, and Gerhard Weikum. 2010. A language modeling approach for temporal information needs. In *Proceedings of the 32nd European Conference on Advances in Information Retrieval*, ECIR'2010, page 1325, Berlin, Heidelberg. Springer-Verlag.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

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

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

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

Ziyang Chen, Jinzhi Liao, and Xiang Zhao. 2023. Multi-granularity temporal question answering over knowledge graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11378–11392, Toronto, Canada. Association for Computational Linguistics.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

AQ Jiang, A Sablayrolles, A Mensch, C Bamford, DS Chaplot, D de Las Casas, F Bressand, G Lengyel, G Lample, L Saulnier, and 1 others. 2023. Mistral 7b. corr, abs/2310.06825, 2023. doi: 10.48550. *arXiv preprint ARXIV.2310.06825*, 10.

Hideo Joho, Adam Jatowt, Roi Blanco, Hajime Naka, and Shuhei Yamamoto. 2014. Overview of NTCIR-11 temporal information access (temporalia) task. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-11, National Center of Sciences, Tokyo, Japan, December 9-12, 2014*. National Institute of Informatics (NII).

Hideo Joho, Adam Jatowt, Roi Blanco, Haitao Yu, and Shuhei Yamamoto. 2016. Overview of NTCIR-12 temporal information access (temporalia-2) task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016*. National Institute of Informatics (NII).

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, and 1 others. 2023. Realtime qa: What's the answer right now? *Advances in neural information processing systems*, 36:49025–49043.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Xiaoyan Li and W. Bruce Croft. 2003. Time-based language models. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, page 469475, New York, NY, USA. Association for Computing Machinery.

Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien De Masson DAutume, Tim Scholtes, Manzil Zaheer, Susannah Young, and 1 others. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pages 13604–13622. PMLR.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Jannat Meem, Muhammad Rashid, Yue Dong, and Vagelis Hristidis. 2024. PAT-questions: A self-updating benchmark for present-anchored temporal question-answering. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13129–13148, Bangkok, Thailand. Association for Computational Linguistics.

Bhawna Piryani, Abdelrahman Abdullah, Jamshid Mozafari, Avishek Anand, and Adam Jatowt. 2025. It's high time: A survey of temporal information retrieval and question answering. *arXiv preprint arXiv:2505.20243*.

Bhawna Piryani, Jamshid Mozafari, and Adam Jatowt. 2024. Chroniclingamericaqa: A large-scale question answering dataset based on historical american newspaper pages. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 20382048, New York, NY, USA. Association for Computing Machinery.

A Yang Qwen, Baosong Yang, B Zhang, B Hui, B Zheng, B Yu, Chengpeng Li, D Liu, F Huang, H Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint*.

Andrey Styskin, Fedor Romanenko, Fedor Vorobyev, and Pavel Serdyukov. 2011. Recency ranking by diversification of result set. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, page 19491952, New York, NY, USA. Association for Computing Machinery.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada. Association for Computational Linguistics.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2024. Towards robust temporal reasoning of large language models via a multi-hop QA dataset and pseudo-instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6272–6286, Bangkok, Thailand. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. 2024. FreshLLMs: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13697–13720, Bangkok, Thailand. Association for Computational Linguistics.

Jonas Wallat, Abdelrahman Abdallah, Adam Jatowt, and Avishek Anand. 2025. A study into investigating temporal robustness of LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15685–15705, Vienna, Austria. Association for Computational Linguistics.

Jiexin Wang, Adam Jatowt, and Masatoshi Yoshikawa. 2022. Archivalqa: A large-scale benchmark dataset for open-domain question answering over historical news collections. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 30253035, New York, NY, USA. Association for Computing Machinery.

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.

Zonglin Yang, Xinya Du, Alexander Rush, and Claire Cardie. 2020. Improving event duration prediction via time-aware pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3370–3378, Online. Association for Computational Linguistics.

Michael Zhang and Eunsol Choi. 2021. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhixue Zhao, George Chrysostomou, Kalina Bontcheva, and Nikolaos Aletras. 2022. On the impact of temporal concept drift on model explanations. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4039–4054, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Prompts

B Additional Results

System Prompt: You are a temporal analyst. Your task is to assign a label to a question. The label should reflect when you expect the answer to this question to change for the first time, based on the nature of the information it requires. ***User Prompt:*** Based on the question, provided below Question:

<question>

1. Assign a label to the question based on when you expect the first change in the answer to occur.

   Consider **how soon** you expect the answer to change **for the first time.**

   - An-Hour The answer is likely to change within an hour
   - A-Few-Hours The answer is likely to change within a few hours
   - A-Day The answer is likely to change within a day
   - A-Few-Days The answer is likely to change within a few days
   - A-Week The answer is likely to change within a week
   - A-Few-Weeks The answer is likely to change within a few weeks
   - A-Month The answer is likely to change within a month
   - A-Few-Months The answer is likely to change within a few months
   - A-Year The answer is likely to change within a year
   - A-Few-Years The answer is likely to change within a few years
   - Many-Years The answer is likely to change within many years
   - Never The answer is not likely to ever change.

2. Provide your reasoning for the label you assigned. Explain when and why you expect this information to change, or state why you believe it will never change. Please limit your explanation to 2-3 concise sentences.

Format your response as a JSON list, where each output is represented as:

```
[
  {
    "Label": "<label>",
    "Justification": "<justification>"
  }
]
```

The output must be a valid JSON list only.

Figure 4: Prompt for assigning recency label. `<question>` represents the given question. recency label is represented by `<label>` and a justification (`<justification>`) justifying the assigned recency label.

System Prompt: You are an event generation expert. You deeply understand real-world domains and can identify events that naturally evolve over time. You think creatively, avoid redundancy, and ensure the output can be used flexibly for machine learning and question-answer datasets. ***User Prompt:*** You

are building a comprehensive dataset of generic events that can later be turned into time-sensitive questions. The events must be diverse, realistic, and grouped by how soon they are likely to change.
Follow these steps to generate the event list:

1. Identify 8 primary temporal labels:

   - An-Hour
   - A-Few-Hours
   - A-Day
   - A-Few-Days
   - A-Week
   - A-Few-Weeks
   - A-Month
   - A-Few-Months

2. For each label, generate diverse, generic events that naturally evolve within that timeframe.

3. Cover multiple domains.

4. Make events concrete enough to be used for dynamic questions.

Format your response as a JSON list, where each output is represented as:

```
[
  {
    "Event": "<the generated event>",
    "Label": "<one of: An-Hour, A-Few-Hours, A-Day, A-Few-Days, A-Week, A-Few-Weeks, A-Month, A-Few-Months>"
  }
]
```

The output must be a valid JSON list only.

Figure 5: Prompt for Event Generation.

System Prompt: You are an analyst tasked with verifying labels for questions. You will be given:

- A question
- Its assigned temporal label
- Additional context

Your job is to assess whether the assigned label correctly reflects when the answer to a question is most likely to change, based on the combined meaning of the question and the context, typical patterns, and reasonable expectations.
The possible labels are:

- An-Hour
- A-Few-Hours
- A-Day
- A-Few-Days
- A-Week
- A-Few-Weeks
- A-Month
- A-Few-Months
- A-Year
- A-Few-Years
- Many-Years
- Never

Decision rules:

- First, determine your `Ideal Label` from the list above.
- If `Ideal Label` equals the `Assigned Label` (case-insensitive, ignoring formatting) → always answer `"YES"`.
- Answer `"YES"` if the `Assigned Label` is exactly correct or close to `Ideal Label`.
- Answer `"NO"` only if the `Assigned Label` is clearly incorrect.
- Never answer `"NO"` if you recommend the same label as the `Assigned Label`.

User Prompt: Based on the given question: `{question}`
Assigned label: `{label}`
And the context: `{context}`

1. **Verification Decision:**
   - Answer `"YES"` if the label accurately reflects when this answer is most likely to change for the first time, considering both the question and the context.
   - Answer `"NO"` if you believe the label is clearly incorrect or far from the expected timeframe, considering both the question and the context.
2. **Justification:** Provide your assessment in 2–3 sentences explaining when you expect this answer to most likely change and why.

Please limit your explanation to 2–3 concise sentences.
Format your response as a JSON list, where each output is represented as:

```
[
  {
    "Label": "<YES or NO>",
    "Justification": "<justification>"
  }
]
```

Figure 6: Prompt for Label Verification

System Prompt: You are an expert in question generation. You will be given events describing real-world situations that can change over time. Your task is to generate exactly 10 diverse, time-sensitive questions for each event.

CRITICAL: You must generate exactly 10 questions. No more, no less.

Follow the instructions exactly as given, output only in the requested JSON format, and ensure all questions are fact-based, unambiguous.

User Prompt: Based on the event provided below:
Event: <event>
First, make this event specific and verifiable by adding concrete details, such as specific names and locations. Avoid specific calendar dates or years that become outdated. Then, generate exactly 10 diverse, realistic, and fact-based questions using this event.
Guidelines:

1. Generate EXACTLY 10 questions - this is mandatory.
2. The questions must rely on information that changes over time.
3. If the event lacks specifics, add realistic details to anchor the questions

Your output must be a single JSON list with exactly 10 questions:

```
[
  {
    "Question": "<question>",
    "Question": "<question>",
    "Question": "<question>",
    .....

  }
]
```

Generate exactly 10 unique questions.

Figure 7: Prompt for Question Generation from Events.

System Prompt: You are an analyst embedded within a state-of-the-art large language model. You will be given a natural language question and a label. The label indicates how soon the answer to the question is expected to change at the time the question is asked.

Your core responsibility is to assess whether a question is **stationary** meaning its temporal label remains stable no matter when the question is asked or **non-stationary**, meaning its temporal label will change depending on the time of year, world events, or other context.

You are expected to provide clear, consistent, and justifiable classifications of stationarity to support the creation of a temporally robust QA dataset.

User Prompt:
You are given a question and its label:

Question: <question>
Label: <label>

The label represents how soon the answer to that question is likely to change.

1. Determine whether the question is stationary or non-stationary.

   Answer **YES (stationary)** if:

   - The label would remain the same regardless of when the question is asked.
   - Even if the answer changes regularly, the time frame of change remains consistent so the label is stable.

   Answer **NO (non-stationary)** only if:

   - The label would definitely change depending on when the question is asked.
   - The question refers to a one-time event or incident, rather than a recurring cycle.
   - The question is only relevant during a short window of time making its temporal behavior unstable.

   Important: Time references like "this week," "since event X," or "following announcement Y" do NOT automatically make a question non-stationary if the underlying temporal pattern remains consistent.

   Critical distinction: Questions mentioning specific events are stationary if they ask about metrics that change at the same frequency regardless of the event.

   Focus only on whether the **label would change**, not the answer itself.

   Here are some examples:

```
[
  {
    "Question": "What is the current air quality index in Beijing?",
    "Label": "A-Few-Hours",
    "Stationary": "YES",
   "Justification": "The question seeks real-time environmental data that updates multiple times a day. The label remains stable."
  },
  {
    "Question": "Which films are nominated for Best Picture at the Cannes Film Festival?",
    "Label": "A-Year",
    "Stationary": "YES",
    "Justification": "The event is annual, so the temporal label is always 'A-Year'."
  },
  {
    "Question": "How many days are left until the U.S. presidential election?",
    "Label": "A-Day",
    "Stationary": "YES",
    "Justification": "The answer changes daily, but the cadence is stable, so the label remains 'A-Day'."
  },
  {
    "Question": "What is the weather forecast for the New Year's Eve celebration in Times Square?",
    "Label": "A-Few-Days",
    "Stationary": "NO",
    "Justification": "Event-specific and only relevant in a narrow time window."
  },
  {
    "Question": "What is the current road closure status due to the flooding in downtown Houston?",
    "Label": "A-Few-Hours",
    "Stationary": "NO",
    "Justification": "One-time incident; label changes once the flooding ends."
  }
]
```

2. Provide your reasoning for the label you assigned. Please limit your explanation to 2-3 concise sentences.

Format your response as a JSON list, where each output is represented as:

```
[
  {
    "Stationary": "<label>",
    "Justification": "<justification>",

  }
]
```

Figure 8: Prompt for assigning Stationary Label.

System Prompt: You are helping construct a temporal QA dataset where each question has a label indicating how soon its answer is expected to change. However, some questions only make sense in a particular moment or situation. To make the label appropriate, we need to add a brief, realistic context for each question.

User Prompt: You are given a question and its label:

> Question: `{question}`
> Label: `{label}`

Label represents how soon the answer to that question is likely to change.
Generate a very brief but clear contextual sentence that gives information about when this question is asked and why this label makes sense. The context should ground the question in an event.
Important: Do *not* include specific years simply create a moment where the question arises and the label is appropriate.

Format your response as a JSON list, where each output is represented as:

```
[
  {
    "Context": "<context>"
  }
]
```

Figure 9: Prompt for Context Generation for Recency Label1

System Prompt: You are assisting in constructing a temporal QA dataset where each question is assigned a label indicating how soon its answer is expected to change. However, for some questions, the appropriate label depends on the specific context or timing in which the question is asked.
Your task is to generate a brief and realistic context for such questions a moment, event, or situation that would cause the label to differ from its default. This context should ground the question in a plausible scenario where the expected answer volatility changes.

User Prompt: You are given a question, its default label, and context as follows:

> Question: `{question}`
> Label: `{label}`
> Context: `{context}`

Label represents how soon the answer to that question is likely to change, and the context gives information about when this question is asked and why this label makes sense.
Generate another very brief but clear contextual sentence a specific moment or event that would result in a **different label** if the question were asked then. The context should ground the question. **Do not** include specific years simply create a moment when the question is asked and the expected time for the answer to change would differ.

Format your response as a JSON list, where each output is represented as:

```
[
  {
    "Context.2": "<context>"
  }
]
```

Figure 10: Prompt for Generating Alternative Context (Context.2) for Recency Label

System Prompt: You are an analyst. Your task is to assign a label to a question. The label should reflect when you expect the answer to this question to change for the first time, based on the nature of the information it requires.

User Prompt: Based on the question and the context provided below:

> Question: `{question}`
> Context: `{context}`

1. **Assign a label** to the question based on when you expect the answer to change for the first time, given the context. Consider **how soon** you expect the answer to change **for the first time**:
   - `An-Hour` The answer is likely to change within an hour
   - `A-Few-Hours` The answer is likely to change within a few hours
   - `A-Day` The answer is likely to change within a day
   - `A-Few-Days` The answer is likely to change within a few days
   - `A-Week` The answer is likely to change within a week
   - `A-Few-Weeks` The answer is likely to change within a few weeks
   - `A-Month` The answer is likely to change within a month
   - `A-Few-Months` The answer is likely to change within a few months
   - `A-Year` The answer is likely to change within a year
   - `A-Few-Years` The answer is likely to change within a few years
   - `Many-Years` The answer is likely to change within many years
   - `Never` The answer is not likely to ever change
2. **Provide your justification** for the label you assigned. Explain when and why you expect this information to change, or state why you believe it will never change. Please limit your explanation to 2–3 concise sentences.

Format your response as a JSON list, where each output is represented as:

```
[
  {
    "Label.2": "<label>",
    "Justification.2": "<justification>"
  }
]
```

Figure 11: Prompt for Recency Label 2 Generation

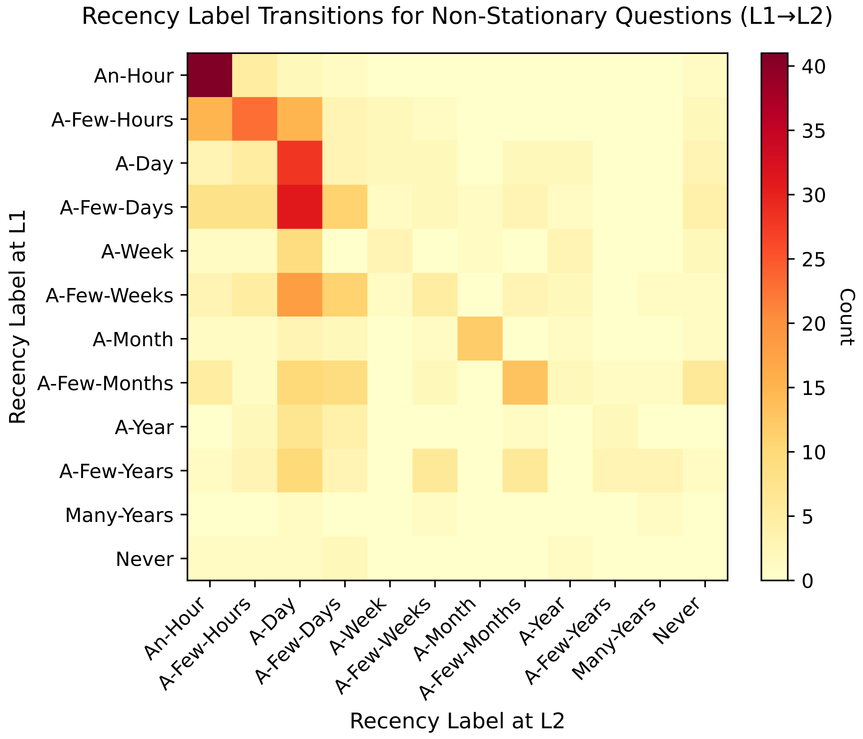
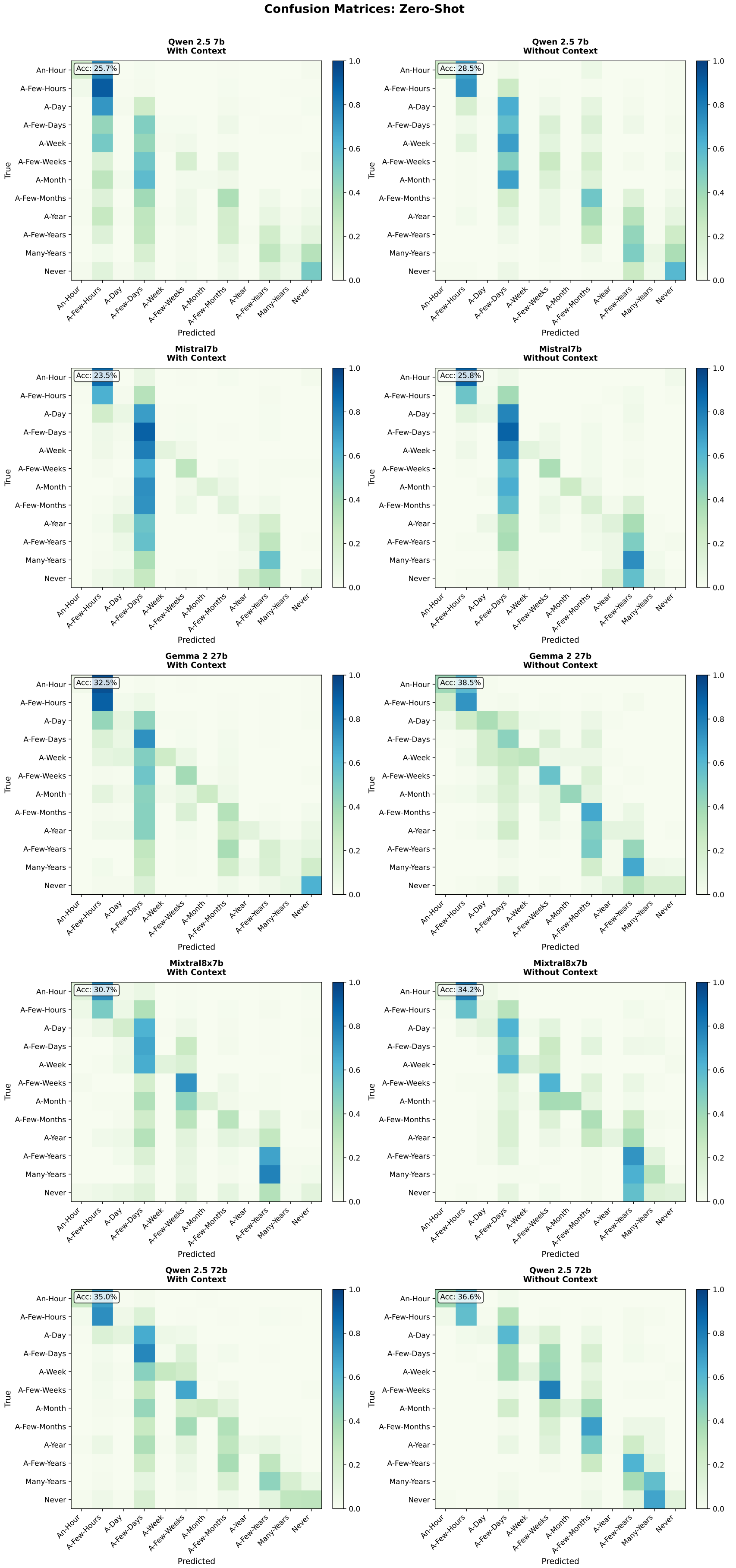


Figure 12: Recency label transitions for non-stationary questions from RL1 to RL2. The heatmap shows how the ground truth recency labels change between the two temporal stages for questions where the answer's recency naturally evolves over time. The y-axis represents the recency label at stage L1, while the x-axis shows the label at stage L2. Diagonal entries indicate questions where the recency label remains the same despite being non-stationary, while off-diagonal entries reveal the temporal progression patterns inherent in the dataset.


Confusion Matrices: Zero-Shot

Qwen 2.5 7b With Context

Qwen 2.5 7b Without Context

Mistral7b With Context

Mistral7b Without Context

Gemma 2 27b With Context

Gemma 2 27b Without Context

Mixtral8x7b With Context

Mixtral8x7b Without Context

Qwen 2.5 72b With Context

Qwen 2.5 72b Without Context

Figure 13: Confusion matrices for zero-shot prompting across all evaluated models with and without context. Each subplot shows the distribution of predicted recency classes (x-axis) against ground truth labels (y-axis), with color intensity indicating the proportion of predictions. Models are organized by rows and context availability by columns. Accuracy percentages are displayed in the top-left corner of each matrix.

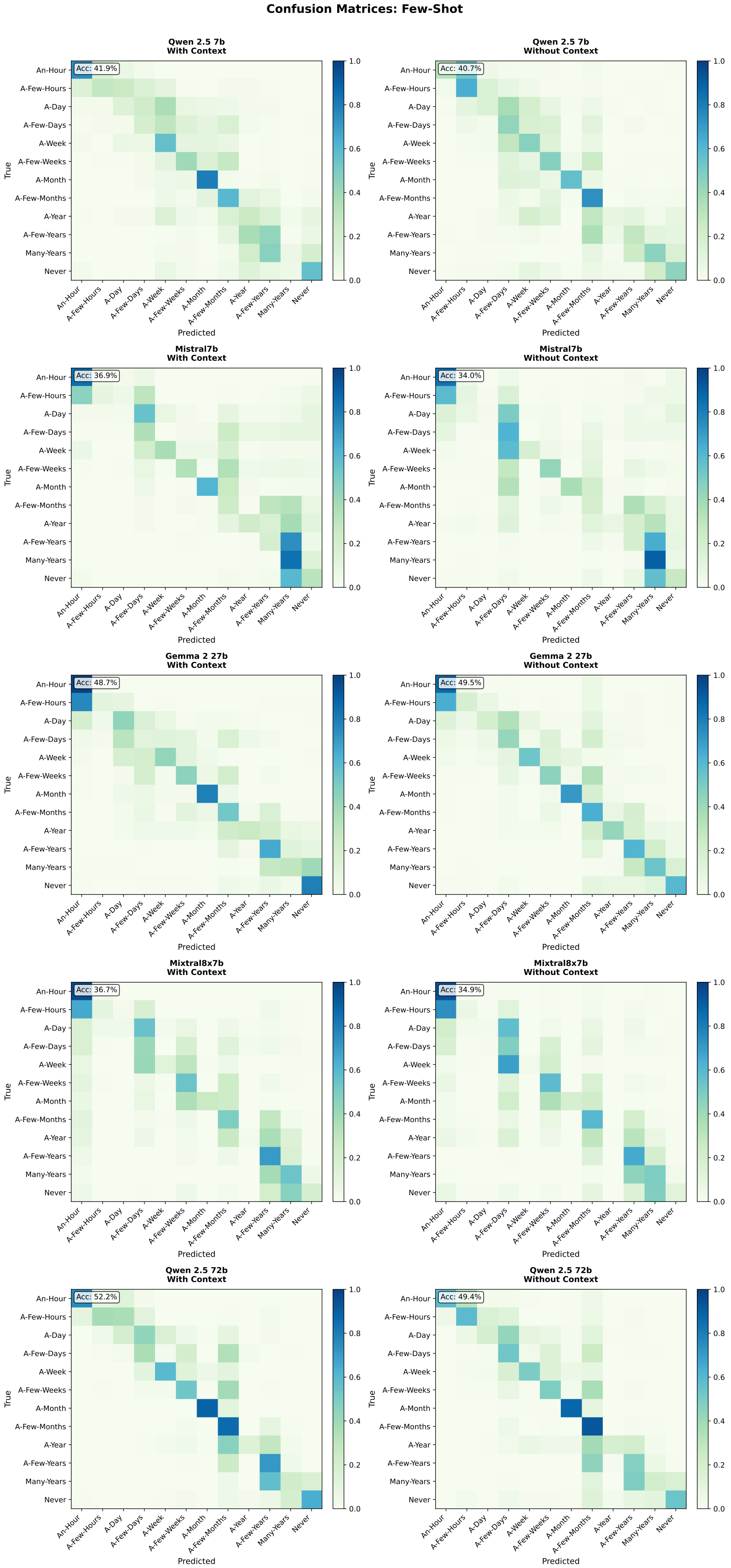


Figure 14: Confusion matrices for few-shot prompting across all evaluated models with and without context. Each subplot shows the distribution of predicted recency classes (x-axis) against ground truth labels (y-axis), with color intensity indicating the proportion of predictions. Models are organized by rows and context availability by columns. Accuracy percentages are displayed in the top-left corner of each matrix

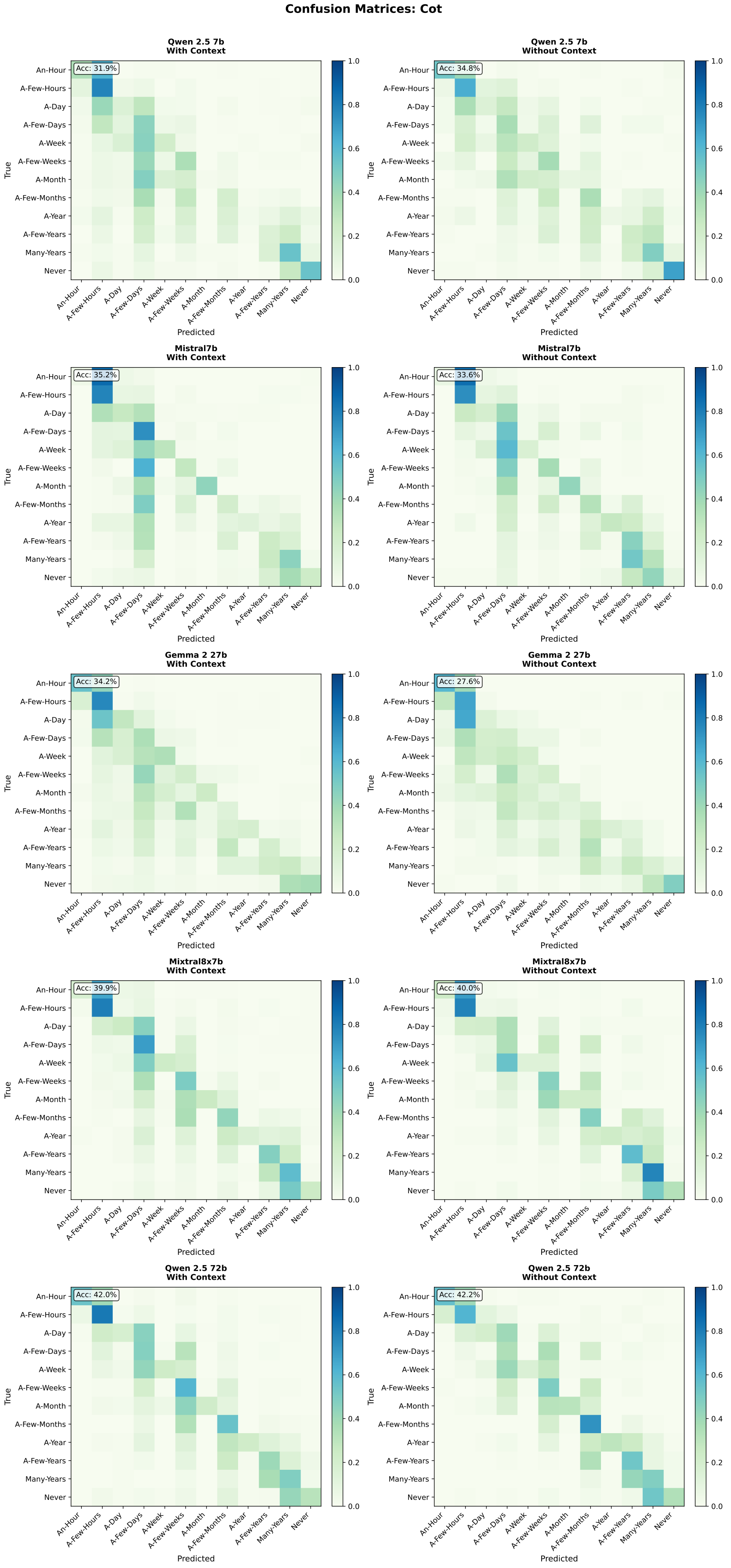


Figure 15: Confusion matrices for chain-of-thought(CoT) prompting across all evaluated models with and without context. Each subplot shows the distribution of predicted recency classes (x-axis) against ground truth labels (y-axis), with color intensity indicating the proportion of predictions. Models are organized by rows and context availability by columns. Accuracy percentages are displayed in the top-left corner of each matrix