Do Language Models Have a Common Sense regarding Time? Revisiting Temporal Commonsense Reasoning in the Era of Large Language Models

Raghav Jain[1], **Daivik Sojitra**[1], **Arkadeep Acharya**[1], **Sriparna Saha**[1],
Adam Jatowt[2], and **Sandipan Dandapat**[3]
[1]Department of Computer Science and Engineering, Indian Institute of Technology Patna
[2] University of Innsbruck, Austria
[3] Microsoft, India
raghavjain106@gmail.com

Abstract

Temporal reasoning represents a vital component of human communication and understanding, yet remains an underexplored area within the context of Large Language Models (LLMs). Despite LLMs demonstrating significant proficiency in a range of tasks, a comprehensive, large-scale analysis of their temporal reasoning capabilities is missing. Our paper addresses this gap, presenting the first extensive benchmarking of LLMs on temporal reasoning tasks. We critically evaluate 8 different LLMs across 6 datasets using 3 distinct prompting strategies. Additionally, we broaden the scope of our evaluation by including in our analysis 2 Code Generation LMs. Beyond broad benchmarking of models and prompts, we also conduct a fine-grained investigation of performance across different categories of temporal tasks. We further analyze the LLMs on varying temporal aspects, offering insights into their proficiency in understanding and predicting the continuity, sequence, and progression of events over time. Our findings reveal a nuanced depiction of the capabilities and limitations of the models within temporal reasoning, offering a comprehensive reference for future research in this pivotal domain.

1 Introduction

Temporal reasoning (Allen, 1983) stands as a fundamental pillar of human communication and understanding, acting as a guiding force in our interpretation of events and narratives. Comprehending natural language involves a deep understanding of time and its facets, which include the duration, sequence, and frequency of events. Within the realm of Natural Language Understanding (NLU), the ability to reason with temporal information has emerged as a significant area of research. This exploration has witnessed substantial strides in recent years, with numerous researchers contributing to this growing body of knowledge (Zhou et al., 2019; Qin et al., 2021). Temporal reasoning extends beyond the mere awareness of time — it requires a nuanced understanding of time's relation to events and actions. For instance, humans inherently know that a vacation usually lasts longer than a walk and occurs less frequently. Therefore, the quest to enhance temporal reasoning within AI systems is of paramount importance. A language model with a robust understanding of temporal context is primed to perform better on downstream Natural Language Processing (NLP) tasks such as storytelling (Mostafazadeh et al., 2016), natural language inference (Hosokawa et al., 2023), timeline understanding (Steen and Markert, 2019), and user status tracking (Xia and Qi, 2022). The integration of temporal reasoning not only enhances the nuances of these applications but also holds the potential to significantly improve the overall performance of AI systems.

Large Language Models (LLMs) (Zhao et al., 2023) have demonstrated remarkable capabilities in a variety of tasks, ranging from commonsense reasoning (Li et al., 2022) to arithmetic problem-solving (Imani et al., 2023). Despite the abundance of studies benchmarking LLMs on these tasks, there is a conspicuous absence of comprehensive, large-scale analysis focusing on benchmarking the models on temporal reasoning. Since temporal reasoning represents a crucial aspect of human comprehension, influencing our interpretation and response to a myriad of scenarios, hence the lack of large-scale benchmarking for LLMs on temporal reasoning tasks is a significant gap in our understanding of these models' capabilities. Acknowledging this considerable gap, we have undertaken the first extensive benchmarking of LLMs on temporal reasoning tasks. (1) Our comprehensive analysis encompasses 6 datasets, leveraging 8 different language models. The language models have been tested through 3 different prompting strategies, aiming to explore the breadth and depth of their temporal reasoning proficiency. Moreover, we have also included 2 Code Generation LMs, further broadening the spectrum of our analysis. (2) In addition to our broad benchmarking efforts, we have conducted a fine-grained analysis of the performance of these models across different categories of temporal reasoning including estimation of event duration, event order, event frequency, stationarity of events and typical time of events. (3) We further analyze the LLMs on varying temporal aspects, offering insight into their proficiency in understanding and predicting the continuity, sequence, and progression of events over time.

In particular, we investigate the following research questions:

- *What is the general performance of LLMs in Temporal Commonsense Reasoning?*
- *Are the models proficient across all different temporal tasks?*
- *Which temporal commonsense tasks present the greatest challenges?*
- *Does the ambiguity in temporal expressions affect model performance?*
- *How do models perform when they need to reason about long time frames, multiple events, or over past and future events?*

2 Related Works

Temporal Reasoning and Understanding: Recent years have witnessed a significant rise in interest in evaluating models' temporal understanding. Key contributions to this field have been made through the introduction of datasets explicitly designed to assess and improve the temporal understanding of models. Recent work by Thukral et al. (2021) and Hosokawa et al. (2023) have created Natural Language Inference (NLI) datasets to assess pretrained models' understanding of typical common-sense temporal expressions, encompassing concepts such as containment and verification of the state of events. To probe models' common sense, researchers formulated *TimeDial* (Qin et al., 2021) and *MC-TACO* (Zhou et al., 2019), which contain a diverse array of situations and temporal expressions. The recent past has also seen the proposal of several QA datasets that are sensitive to time (Chen et al., 2021). Furthermore, the recent research in temporal reasoning has focused on developing time-aware training and representation strategies for language models (Wang et al., 2023; Cole et al., 2023; Kimura et al., 2021; Zhou et al., 2020; Kimura et al., 2022; Saxena et al., 2021). The Temporal Knowledge Graph Completion (TKGC) domain has been exploring temporal reasoning within knowledge graphs (Dhingra et al., 2022; Jang et al., 2023). Overall, contemporary research has exhibited a notable expansion in temporal reasoning studies in natural language understanding.

Benchmarking LLMs: The proficiency of LLMs has been notably illustrated across various tasks, yet their exact potential and constraints remain somewhat ambiguous. Recent studies have made strides in scrutinizing the performance of LLMs in diverse scenarios and tasks. For instance, Asai et al. (2023) and Ahuja et al. (2023) have conducted extensive benchmarking of three LLMs on cross-lingual and multilingual tasks, respectively. In addition, Wadhwa et al. (2023) performed an assessment of two LLMs' capabilities on relation extraction tasks. Yang et al. (2023) carried out benchmarking of ChatGPT in the context of mental health issues. Furthermore, Nay et al. (2023), conducted comparative analyses of ChatGPT and GPT-4 regarding their performances on legal tax problems. In essence, the latest research showcases an escalating trend in probing the potential applications of LLMs across a variety of domains, languages, and tasks.

In conclusion, despite comprehensive research on LLMs benchmarking in diverse contexts, their proficiency in a temporal common sense remains largely unexplored. This area reveals a necessity for a systematic evaluation of LLMs' understanding and reasoning within the temporal domain.

3 Benchmark Setup

In the following sections, we provide the details of the datasets, tasks, prompting techniques, and language models used in our research study. The datasets and tasks primarily pertain to temporal reasoning tasks, requiring the models to display an understanding and reasoning in time-sensitive contexts and situations. In terms of language models, we examine a diverse set of models pretrained with different strategies, including both standard and Code Generation LMs.

3.1 Datasets and Tasks

We have employed the following datasets that are related to temporal reasoning tasks:

MC-TACO (Zhou et al., 2019): Given a context, a question, and a candidate response, the objective is to determine whether the candidate answer is "yes" (plausible) or "no" (implausible). The dataset fo-

Dataset	Task Description	Output	Evaluation Metric	Temporal Reasoning
MC-TACO	Binary Classification	Yes/No	Acc and weighted F1	ED, EO, F, S, TT
TimeDial	Binary Classification	Yes/No	Acc and weighted F1	TT
TNLI	Natural Language Inference	Support/Invalidate/Neutral	Acc and weighted F1	S
WikiHow	Binary Classification	Yes/No	Acc and weighted F1	EO
BIG-bench	Multi-Class Classification	Correct Option number	Acc and weighted F1	EO
TimeQA	Question Answering	Answer string	EM and F1	TT

Table 1: Datasets Summary (ED: Event Duration, EO: Event Ordering, F: Frequency, S: Stationarity, TT: Typical Time, Acc: Accuracy, EM: Exact Match)

cuses on assessing the plausibility of the answer within the temporal context provided.

TimeDial (Qin et al., 2021): Dataset of a multiple-choice cloze task featuring over 1.1K carefully curated dialogues. The dialogues require an understanding of temporal commonsense concepts interwoven with the presented events.

TNLI (Hosokawa et al., 2023): Dataset for a novel task known as Temporal Natural Language Inference (*TNLI*). In this task, the model has to ascertain the validity of textual content by using additional associated content as corroborating evidence.

WikiHow (Zhang et al., 2020): Given a goal and a number of steps, a system has to determine if the steps are in the correct temporal order.

BIG-bench (Srivastava et al., 2023): Provided with a sequence of finished events, each with its defined timeframe, the model needs to determine when an individual might have been available for an unscheduled activity. While both *BIG-bench* and *WikiHow* encompass various other reasoning tasks, we specifically focused only on temporal reasoning subtasks.

TimeQA (Chen et al., 2021): This dataset comprises a series of time-sensitive question-answer pairs. Answering these questions involves understanding and reasoning within a longer context that requires temporal comprehension.

The above datasets (refer to Appendix B.4 for examples of each dataset) cover most of the temporal commonsense reasoning styles according to the categorization proposed by Zhou et al. (2019) (refer to Appendix B.1 for detailed description and examples of each task):

Event Duration (ED): reasoning about event durations.

Event Ordering (EO): reasoning about the typical sequence of events.

Frequency (F): reasoning about the frequency of event occurrences.

Stationarity (S): reasoning about the length of state persistence.

Typical Time (TT): reasoning about the specific timing of events.

Table 1 summarizes the datasets we use and gives information on the types of their temporal commonsense reasoning (cf. the last column), and the characteristics of their tasks, the format of the output, and the evaluation metrics applied.

3.2 Prompting Techniques

In the context of our research, we undertake a comprehensive examination of the following in-context learning methods across various models:

Zero-shot Prompting: Zero-shot prompting is the most basic form of prompting. It is simply showing the model a prompt without examples and asking it to generate a response. The zero-shot prompt can be represented as $P = f_{prompt}(TD; x_{test})$ where TD corresponds to the task description, x_{test} refers to the test example, and f_{prompt} is a function transforming the data into a natural language prompt.

Few-shot Prompting: This technique involves presenting the model with two or more instances, known as few-shot prompting. For each label in the dataset, examples are selected randomly; in our case, a single example is chosen for each label. Few-shot prompt can be represented as $P = f_{prompt}(TD; (x_i, y_i)_n; x_{test})$ where (x_i,y_i) symbolize randomly picked samples from the dataset, and n denotes the number of examples[1].

Chain-of-Thought (CoT) Prompting (Wei et al., 2023): This is a recently introduced prompting technique that facilitates the LLMs to elucidate their thought process. The core idea of CoT lies in presenting the LLM with few-shot examples that incorporate explanations of the reasoning process. The CoT prompt can be represented as $P = f_{prompt}(TD; (x_i, y_i, R_i)_n; x_{test})$ where R_i stands for the rationale associated with each few-shot example (x_i, y_i).

Code Prompts (Zhang et al., 2023): The technique of using code-like structures (for example, Python) to prompt Code Generation LM for natural language tasks has been found to enhance performance. The code prompt can be represented

[1] Please note that we didn't perform Few-shot on *TimeQA* dataset because of the context length limit of LLMs.

Language Model	Params	Architecture	Type	Few-Shot	Zero-Shot	CoT	Code-Prompt
GPT-J	6B	Autoregressive Decoder only	Base	✓	✓		
GPT Neo	1.3B	Autoregressive Decoder only	Base	✓	✓		
LLaMA	7B	Autoregressive Decoder only	Base	✓	✓		
OPT	350M	Autoregressive Decoder only	Base	✓	✓		
BLOOMZ	560M	Autoregressive Decoder only	SIFT	✓	✓	✓	
Dolly	3B	Autoregressive Decoder only	SIFT	✓	✓	✓	
FLAN-T5	780M	Encoder-Decoder	SIFT	✓	✓	✓	
SantaCoder	1.1B	Autoregressive Decoder only	Base				✓
CodeGen2	2B	Encoder-Decoder	Base				✓
GPT-3.5	-	-	RLHF	✓	✓	✓	✓

Table 2: Characteristics of Different LLMs employed in this study. Base denotes standard pre-training strategies, SIFT means Supervised Instruction Fine Tuning and RLHF means Reinforcement Learning from Human Feedback

as $P = f_{code}(TD; x_{test})$ where f_{code} denotes the function that translates a natural language prompt into a code representation, wherein instructions and input samples are given as variables with relevant and meaningful names, enriched by comments that describe their purpose and provide an overarching task description. We refer readers to Appendix B.3 for the exact prompts used. Note that scope of the paper does not include many possible advances of the aforementioned prompt like dynamic prompting (Liu et al., 2021), Auto-CoT (Zhang et al., 2022b) which can be explored in future research.

3.3 Language Models

In this work, we evaluate a set of diverse models pre-trained with different strategies (cf. Table 2): **(1) Models for In-Context Learning:** We experiment with a set of diverse models; **Large autoregressive models** - GPT-J,[2] GPT Neo,[3] LLaMA (Touvron et al., 2023), and OPT (Zhang et al., 2022a), **Supervised Instruction Finetuned models** - FLAN-T5 (Chung et al., 2022), BLOOMZ (Muennighoff et al., 2022), Dolly,[4] and **RLHF model** - GPT-3.5. **(2) Code Generation LMs:** Additionally, we explore the use of Code Generation LM to gauge their effectiveness in handling temporal reasoning tasks. For this purpose, we have utilized the following models: (a) SantaCoder (Allal et al., 2023) and (b) CodeGen2 (Nijkamp et al., 2023).

4 Results

In the following subsection, we initiate our discussion by providing a succinct overview of the performance of various LLMs, code generation LMs, and diverse prompting techniques across multiple datasets. We then advance to a more in-depth exploration in Sec. 4.2, scrutinizing the performance of different LLMs in various temporal tasks. Finally, we turn our attention to analyzing different temporal characteristics of the best-performing LLMs from our previous assessments (Sec. 4.3).

4.1 Model and Prompt-based Analysis

Strong performance of GPT-3.5 and FLAN-T5.

Table 3 illustrates the superior performance of GPT-3.5, especially in few-shot and zero-shot learning tasks in datasets such as *MC-TACO* and *TNLI*, showcasing its good generalization ability and vast intrinsic knowledge. This is further improved by the CoT prompting strategy and the RLHF training strategy, highlighting GPT-3.5's robust in-context learning. On the other hand, FLAN-T5, despite being an older model when compared to the other instruction-tuned models, delivers strong performance closely following GPT-3.5 in few-shot learning tasks and even surpasses it in zero-shot learning on the *TimeQA* dataset. This can be attributed to the inherent strength of base T5 models which have been trained with 1 trillion tokens and leverage the extensive C4 dataset (Raffel et al., 2019). BLOOMZ also performs better than the other instruction-tuned decoder model(Dolly) as it was was trained on a cross-lingual mixture of tasks (xP3[5]) spanning dozens of languages. This exposes the model to far higher diversity during pretraining compared to Dolly, which was trained only on English data from a single company's employees. Previous studies also shown (Tanwar et al., 2023) that multilingual LLMs are better at instruction following and in-context learning. LLaMA performs better than other base autoregressive models as LLaMa has been trained on much more larger and diverse dataset as well as training objectives, compared to other base autoregressive models. Moreover, the influence of the CoT prompting strategy (Table 4)

[2] https://github.com/kingoflolz/mesh-transformer-jax

[3] https://github.com/EleutherAI/gpt-neo

[4] https://huggingface.co/databricks/dolly-v2-3b

[5] https://huggingface.co/datasets/bigscience/xP3

	MC-TACO		*TNLI*		*TimeDial*		*WikiHow*		*BIG-bench*		*TimeQA*
	Few-shot	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Zero-shot**
Model	**Acc**	**Acc**	**Acc**	**Acc**	**Acc**	**Acc**	**Acc**	**Acc**	**Acc**	**Acc**	**EM**
`GPT-3.5`	**0.8**	**0.75** (⇓5%)	**0.62**	**0.5** (⇓12%)	**0.65**	**0.65** (=0%)	**0.55**	0.49 (⇓6%)	0.25	0.26 (⇑1%)	0.1
`FLAN-T5`	0.7	0.74 (⇑4%)	0.39	0.37 (⇓2%)	0.54	0.54 (=0%)	0.45	0.45 (=0%)	0.16	0.14 (⇓2%)	**0.4**
`BLOOMZ`	0.59	0.63 (⇑4%)	0.34	0.31 (⇓3%)	0.49	0.46 (⇓3%)	0.53	0.46 (⇓7%)	**0.28**	**0.28** (=0%)	-
`Dolly`	0.45	0.52 (⇑7%)	0.42	0.32 (⇓10%)	0.5	0.49 (⇓1%)	0.45	0.44 (⇓1%)	0.23	0.27 (⇑4%)	0
`GPT-J`	0.59	0.37 (⇓22%)	0.34	0.33 (⇓1%)	0.45	0.33 (⇓12%)	0.46	0.45 (⇓1%)	0.26	0.26 (=0%)	-
`GPT Neo`	0.59	0.37 (⇓22%)	0.32	0.33 (⇑1%)	0.49	0.48 (⇓1%)	0.47	0.44 (⇓3%)	0.26	0.27 (⇑1%)	-
`LLaMA`	0.65	0.65 (=0%)	0.32	0.32 (=0%)	0.5	0.35 (⇓15%)	**0.55**	**0.54** (⇓1%)	**0.28**	0.18 (⇓10%)	0.1
`OPT`	0.65	0.45 (⇓20%)	0.33	0.35 (⇑2%)	0.5	0.49 (⇓1%)	0.46	0.49 (⇑3%)	0.24	0.23 (⇓1%)	-

Table 3: Performance of eight Large Language Models on six datasets, analyzed under two different prompting strategies. "Acc" stands for Accuracy and "EM" corresponds to Exact Match. The percentage changes in accuracy performance between Zero-shot and Few-shot prompting are indicated in parentheses. (- suggests the data instances exceed the LLM's context limit; hence results cannot be determined.)

	MC-Taco	*TNLI*	*TimeDial*	*WikiHow*	*BIG-bench*	*TimeQA*
Model	**Acc**	**Acc**	**Acc**	**Acc**	**Acc**	**EM**
`GPT-3.5`	**0.82** (⇑2%)	**0.46** (⇓16%)	**0.7** (⇑5%)	**0.54** (⇓1%)	**0.67** (⇑42%)	**0.15**
`FLAN-T5`	0.73 (⇑3%)	0.38 (⇓1%)	0.54 (0%)	0.48 (⇑3%)	0.21 (⇑5%)	0.05
`BLOOMZ`	0.45 (⇓14%)	0.36 (⇑2%)	0.53 (⇑4%)	0.5 (⇓3%)	0.3 (⇑2%)	-
`Dolly`	0.48 (⇑3%)	0.34 (⇓8%)	0.5 (0%)	0.47 (⇑2%)	0.29 (⇑6%)	0

Table 4: Performance of Instruction Tuned LLMs with CoT prompting strategy. The percentage changes in accuracy performance between CoT and Few-shot prompting (from Table 3) are indicated in parentheses.

	MC-TACO	*TNLI*	*TimeDial*	*WikiHow*	*BIG-bench*	*TimeQA*
Model	**Acc**	**Acc**	**Acc**	**Acc**	**Acc**	**EM**
`GPT-3.5`	0.5	**0.45**	0.49	**0.45**	**0.29**	**0.1**
`SantaCoder`	0.5	0.36	**0.5**	**0.45**	0.27	-
`CodeGen2`	**0.61**	0.35	**0.5**	**0.45**	0.25	**0.1**

Table 5: Performance of Code Generation LMs with Code prompts in Zero-shot setting. (- suggests the data instances exceed the LLM's context limit; hence results cannot be determined.)

over different models varies; it significantly improves complex temporal reasoning tasks like *BIG-bench*, while the enhancement in others, such as *WikiHow*, is similar across other settings. However, *TNLI* is characterized by inconsistent performance under the CoT setup.

Code Generation LMs are not temporal commonsense reasoners.

Table 5 shows a performance analysis of various models across different tasks, specifically focusing on code generation language models and their ability to reason with temporal commonsense. Previous studies (Madaan et al., 2022) have highlighted the superiority of Code Generation LMs in reasoning and commonsense tasks compared to general-purpose LMs. However, upon reviewing the results in Table 5, it becomes evident that the code generation LMs, `SantaCoder` and `CodeGen2`, struggle as temporal commonsense reasoners. They encounter difficulties across multiple datasets, including *MC-TACO*, *TNLI*, *TimeDial*, and *WikiHow*, where understanding and reasoning about temporal aspects are crucial. Similarly, `GPT-3.5` with code prompts also exhibits limited performance in temporal commonsense reasoning, as reflected by its relatively lower scores in tasks like *TimeDial* and *BIG-bench*. Although `GPT-3.5` outperforms the other code generation LMs, its performance still falls short compared to normal text prompts on all datasets and tasks as shown in Table 3. Our experiments align with the findings of Zhang et al. (2023), who demonstrate that code prompts do not surpass the performance of text prompts. Appendix B.5 provides detailed results with F1 scores.

4.2 Temporal Task-based Analysis

Strong performance of LLMs on event frequency, and duration tasks.

The heatmap illustrated in Figure 1 provides a visualization of the performance of various LLMs across distinct prompting settings when applied to the *MC-TACO* datasets and their respective fine-grained temporal task categories (ED, EO, F, S, TT). Models predominantly perform well on tasks associated with event duration, with `GPT-3.5` taking the lead in accuracy and F1 scores, followed closely by `FLAN-T5`. Other models like `Dolly`, `GPT-J`, `GPT Neo`, `LLaMA`, and `OPT` demonstrate mixed results. However, `BLOOMZ`'s zero-shot capabilities align well with `FLAN-T5` and `GPT-3.5`. On tasks related to event frequency, `GPT-3.5` maintains strong performance, while `FLAN-T5` experiences a slight drop but still presents impressive performance.

Mixed performance on event ordering tasks.

Performance varies across models on the 'Event Ordering' task in the *MC-TACO* dataset, with `GPT-3.5` and `FLAN-T5` leading and others like `LLaMA` and `OPT` showing declines, indicating challenges with event ordering (Figure 1). However, comparison with other datasets like *WikiHow* and *BIG-bench* reveals lower performance, likely due to the increased complexity of these tasks. The improvement in results when models are combined with the CoT prompting technique suggests that more complex event-ordering tasks demand greater reasoning abilities from LLMs.

Performance drop on understanding event temporal states.

There is a notable decrease in performance across all models in the 'Stationarity' task, highlighting the difficulty in assessing the temporal states of events, such as whether events or situations remain constant over time (Figure 1). Remarkably, BLOOMZ demonstrates strong performance in the zero-shot configuration, almost on par with GPT-3.5, suggesting that it has a specific strength in identifying event stationarity. Furthermore, referring to Table 4, it is evident that even when using the CoT prompting strategy, Language Models struggle with other tasks related to Stationarity, such as ones in *TNLI* dataset (*TNLI* dataset requires understanding the states of events). This points towards an inherent challenge faced by these models in grasping and reasoning over concepts of the temporal stability of events.

LLMs struggle with specific event timings.

The 'Typical Time' task stands out as the most demanding for all the models, emphasizing the intricate nature of predicting and reasoning over typical event timings (Figure 1). While GPT-3.5 continues to perform best in this category, its lead over other models is significantly reduced compared to other tasks. The performance gap between GPT-3.5 and other models such as FLAN-T5 and BLOOMZ is notably narrower in the zero-shot configuration, indicating less dominance by GPT-3.5 on this task. A substantial performance decline is also observed in other models, further underlining the task's complexity. A reference to Table 3 reinforces this conclusion of LLMs struggling with exact timings, as it reveals that all Language Models struggle not only with the 'Typical Time' task on *MC-TACO*, but also with other time-related tasks, such as *TimeQA*. This indicates a broader challenge for LLMs in reasoning over specific time periods. Readers can refer to Appendix B.7 for both F1 and accuracy results.

4.3 Temporal Aspect-based Analysis

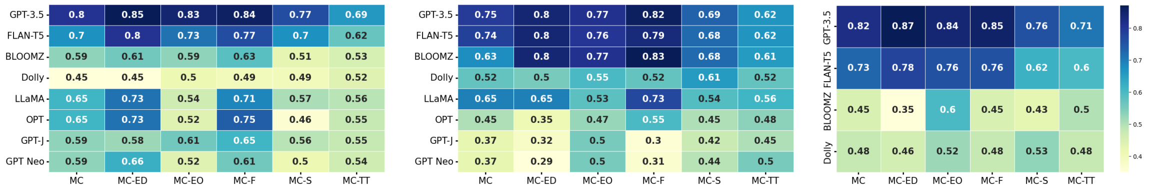
Reasoning about future events is more difficult than about past events.

We used the *TimeDial* dataset and identified 'Past' and 'Future' events based on verb tenses using SpaCy toolkit[6]. Manual verification ensured the accuracy post automatic classification. We retained 200 instances each for both categories to enable an effective performance comparison (an example of such an instance is shown in Figure 2(a). Figure 3(a) highlights a consistent trend among models, indicating a slight drop in performance when reasoning about the Future events compared to the Past events. All models show a trend of higher accuracy in reasoning about the past compared to the future. This disparity may be attributed to the model's extensive training on past events or scenarios, which is more likely to be found in the training data, providing it with a richer database to draw from when making predictions about the past. Joho et al. (2015) also showed that users struggle more with finding information about the future than one about the past when using search engines. In summary, while these models are adept at temporal reasoning, their performance slightly drops when dealing with future events.

LLMs perform better on temporal reasoning over longer timeframes.

To construct the dataset for this experiment, we scrutinized the *TimeDial* dataset. Each instance was classified based on the duration specified by the correct label associated with its context. This led to the creation of categories: seconds, minutes, hours, a day, and durations exceeding a day. After manual verification for accuracy, we sampled 200 instances from each category (Figure 2(b)). Figure 3(b) offers a comparative analysis of models' performance when reasoning about events that occur over varying time frames. The emerging pattern highlights that the models tend to perform better when dealing with longer time frames compared to shorter ones. GPT-3.5 provides the clearest representation of this trend, starting with a relatively lower accuracy for events that transpire over seconds. However, its accuracy increases as the duration of events extends to minutes, hours, and days. Impressively,

[6] https://spacy.io/



(a) Heatmap for Accuracy of LLMs in Few-shot setting

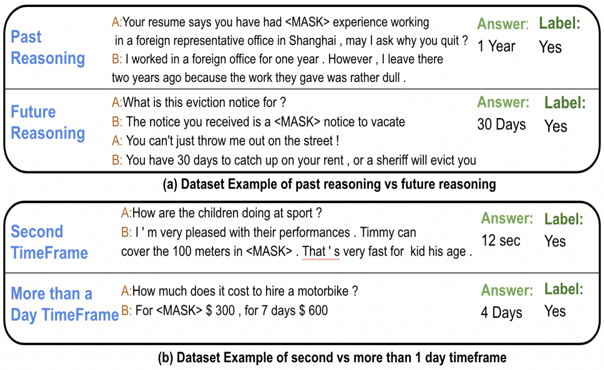
(b) Heatmap for Accuracy of LLMs in Zero-shot setting

(c) Heatmap for Accuracy of LLMs in CoT setting

Figure 1: Performance of LLMs on MC-TACO dataset and it's fine-grained temporal task categories on Few-shot settings. Y-axis represents different models and X represent different temporal tasks (MC: *MC-TACO*)

Dataset	*MC-TACO*	*TNLI*	*TimeDial*	*WikiHow*	*BIG-bench*	*TimeQA*
	Acc	**Acc**	**Acc**	**Acc**	**Acc**	**EM**
`Human Baselines`	0.75	0.82	0.97	0.97	1.0	0.9
`Baseline Fine-Tuned Models`	0.64	0.878	0.748	0.801	-	0.55

Table 6: Performance comparison of Fine-Tuned Models and Human Baselines across Various Datasets. - indicates that the BIG-bench dataset does not have a training set, so no model can be fine-tuned.



A:Your resume says you have had <MASK> experience working in a foreign representative office in Shanghai , may I ask why you quit ?
B: I worked in a foreign office for one year . However , I leave there two years ago because the work they gave was rather dull .

A:What is this eviction notice for ?
B: The notice you received is a <MASK> notice to vacate
A: You can't just throw me out on the street !
B: You have 30 days to catch up on your rent , or a sheriff will evict you

(a) Dataset Example of past reasoning vs future reasoning

A:How are the children doing at sport ?
B: I ' m very pleased with their performances . Timmy can cover the 100 meters in <MASK> . That ' s very fast for kid his age .

A:How much does it cost to hire a motorbike ?
B: For <MASK> $ 300 , for 7 days $ 600

(b) Dataset Example of second vs more than 1 day timeframe

Figure 2: Dataset example for temporal aspect analysis

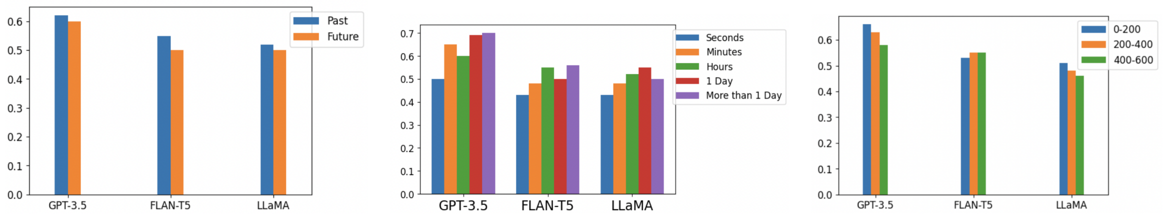
`GPT-3.5` reaches its peak performance when reasoning about events that last more than a day. This upward trajectory suggests that `GPT-3.5` is more adept at handling the intricacies involved in reasoning about longer-time frame events. Similarly, `FLAN-T5` shows an improvement in performance when moving from shorter to longer time frames, although its trajectory is not that consistent. `LLaMA`, despite some variation, also seems to perform better with longer-duration events. These findings could suggest that models are better equipped to handle the complexities and nuances involved in reasoning about longer timeframe events.

LLMs have difficulty with temporal reasoning over longer context.

We divided the *TimeDial* dataset into three categories based on context length: 0-200 words, 200-400 words, and 400-600 words. We then selected 200 instances randomly from each category, effectively creating a dataset with diverse context lengths. In Figure 3(c), we observe that as the context length increases the performance of the models tends to decrease (with the exception of `FLAN-T5`). `GPT-3.5` shows a decline in performance as the context length increases from 0-200 to 400-600, indicating a possible difficulty in handling long contextual information.

LLMs struggle with exact temporal expressions compared to ambiguous ones.

We created a specialized dataset from *MC-TACO* leveraging the combination of HeidelTime (Strötgen and Gertz, 2010) and manual extraction. This allowed us to distinguish between instances containing 'Exact Timings', such as specific numerical expressions of time, day names, and month names, and 'Ambiguous Temporal Expressions', like *in the meantime*, *after a few days*, and *meanwhile* (examples shown in Figure 4(a)). We then performed a manual verification of these instances, maintaining a balanced collection of 200 instances each from the categories of exact and ambiguous temporal expressions. The comparative performance analysis (Figure 5(a)) across diverse temporal reasoning tasks highlights a persistent challenge the models face when dealing with 'Exact Timings' such as numerical values, day names, or month names, as opposed to 'ambiguous temporal expressions'. The models may encounter difficulties in associating specific timings with their implications for the occurrence or sequence of events, indicating a potential challenge in comprehending and connecting exact temporal information. It underscores a key area of struggle for these models - effective rea-

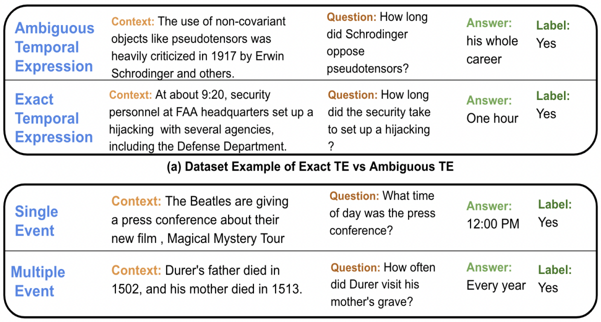


(a) Performance of LLMs across Past and Future events

(b) Performance of LLMs across different timeframes

(c) Performance of LLMs across different context length

Figure 3: Bar Plots indicating the performance of LLMs across different temporal aspects in few-shot setting. X-axis represents the different LLMs, and Y-axis represents performance in accuracy



Ambiguous Temporal Expression	Context: The use of non-covariant objects like pseudotensors was heavily criticized in 1917 by Erwin Schrodinger and others.	Question: How long did Schrodinger oppose pseudotensors?	Answer: his whole career	Label: Yes
Exact Temporal Expression	Context: At about 9:20, security personnel at FAA headquarters set up a hijacking with several agencies, including the Defense Department.	Question: How long did the security take to set up a hijacking ?	Answer: One hour	Label: Yes

(a) Dataset Example of Exact TE vs Ambiguous TE

Single Event	Context: The Beatles are giving a press conference about their new film , Magical Mystery Tour	Question: What time of day was the press conference?	Answer: 12:00 PM	Label: Yes
Multiple Event	Context: Durer's father died in 1502, and his mother died in 1513.	Question: How often did Durer visit his mother's grave?	Answer: Every year	Label: Yes

(b) Dataset Example of Single Event vs Multiple Event

Figure 4: Dataset example for temporal aspect analysis

soning over specific time periods across different tasks. This may be attributed to the higher prevalence of ambiguous events as compared to precise ones, largely because they are utilized more frequently in a variety of scenarios. Readers can refer to Appendix B.6 for detailed results of this section.

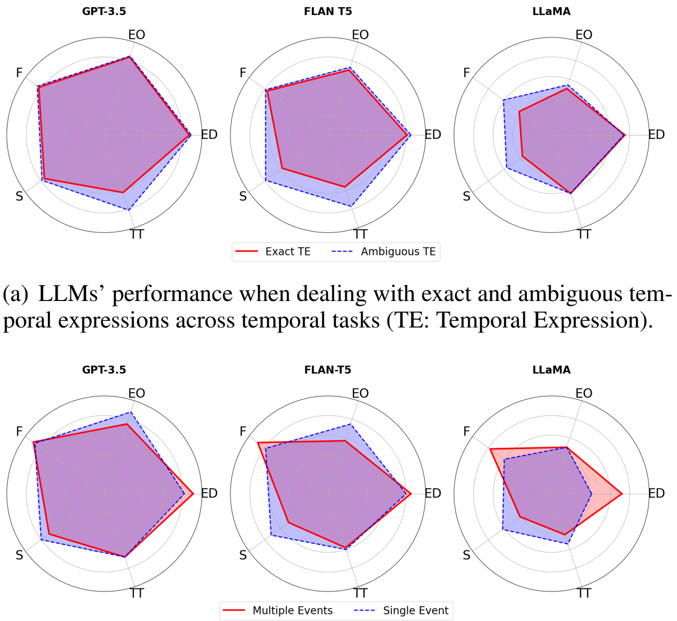
LLMs struggle with understanding the states and orders of multiple events.

We manually dissected the *MC-TACO* dataset due to the inherent complexity of accurately identifying single and multiple temporal events within instances. We systematically curated two groups from each category: instances with single and multiple temporal events. To maintain uniformity, we selected 200 instances from each group, as shown in Figure 4(b). Figure 5(b) compares model performance across tasks for single versus multiple temporal events. In the context of 'Event Duration' and 'Frequency', all models perform better when reasoning about multiple events as compared to single events. On the other hand, the 'Stationarity' and 'Event Ordering' tasks display a different trend, where all models perform better when reasoning about single events as opposed to multiple events. This could indicate that these tasks, which require understanding the persistence of states and the typical sequence of events, can become more complex and challenging when multiple events are involved. In the 'Typical Time' task, we observe that all models generally perform better or at least equally well when reasoning about single events, perhaps indicative of their difficulties when attempting to comprehend specific timings associated with multiple events.

5 Comparison with Fine-Tuned models and Human Performance

Human Evaluation: Human evaluations were carried out using 100 random samples from the TNLI, BIG-bench, and TimeQA datasets, assessed by three in-house annotators. For other datasets, we relied on human baselines provided in the original papers. Annotators' accuracy was gauged by comparing their responses to the ground truth, establishing human benchmarks for performance (Table 6). For TimeDial, WikiHow, BIG-bench, and TimeQA tasks, human proficiency substantially outperforms all LLMs. For example, humans scored 0.97 on TimeDial, whereas the best LLM, GPT-3.5, only achieved 0.65. The gap is further widened in tasks like WikiHow, where humans scored 0.975. In the TNLI dataset, even the top-performing LLM, GPT-3.5, falls short of the human baseline score (0.82 vs. 0.62), indicating lingering challenges for LLMs in this task. Contrarily, GPT-3.5 matches human performance on the MC-TACO dataset in a few-shot setting, both scoring 0.8, suggesting that under certain conditions, latest LLMs can achieve human-level capabilities.

Fine Tuned Model Baselines: Table 6 also includes evaluation results for the best-performing fine-tuned models from the papers introducing each dataset. Key findings are as follows: In the MC-TACO dataset, the baseline model scored 0.64.



(a) LLMs' performance when dealing with exact and ambiguous temporal expressions across temporal tasks (TE: Temporal Expression).

(b) Performance of LLMs when processing both single-event and multiple-event scenarios across temporal tasks.

Figure 5: Radar plots comparing accuracy of LLMs across different temporal aspects in few-shot setting

However, LLMs like GPT-3.5 and FLAN-T5 significantly outperformed it in both few-shot and zero-shot scenarios. For TNLI, the fine-tuned baseline model achieved 0.878, outperforming all LLMs. Notably, GPT-3.5, the top LLM, scored only 0.62 in few-shot settings. This is because the baseline was fine-tuned on TNLI data and leveraged external commonsense knowledge. On TimeDial and WikiHow, the baseline models scored 0.748 and 0.801, respectively. GPT-3.5 led among LLMs but did not surpass the baseline. The fine-tuned models excel due to task-specific optimizations. For TimeQA, with a baseline score of 0.55, FLAN-T5 was the closest among LLMs with an EM score of 0.4. The baseline's higher performance is attributed to its use of Retrieval Augmented Generation (FiD), allowing it to handle the dataset's long context.

6 Conclusion

In this paper, we aimed to bridge a critical knowledge gap by conducting a comprehensive benchmarking of LLMs on temporal reasoning tasks. Our thorough analysis has shed light on certain limitations in the ability of LLMs to reason temporally. Specifically, we have identified areas where LLMs struggle, such as comprehending the temporal states of events, accurately reasoning over precise time timings, managing multiple temporal events, and predicting future events. By highlighting these challenges, our study contributes to a better understanding of the capabilities and limitations of LLMs in temporal reasoning tasks.

Limitations

There are several limitations of our study that should be acknowledged. Firstly, in terms of prompt selection, our testing was limited to a few prompting strategies, and there exist numerous other techniques and variations that could be explored. Therefore, the generalizability of our findings to different prompt settings may be constrained. Secondly, our evaluation of models was primarily focused on open-source models and covered only one closed model (`GPT-3.5`). We did not include closed models like `PaLM`. In this study, we covered only temporal commonsense reasoning. Yet, we acknowledge that there are various other temporal tasks that were not covered, such as timeline summarization and temporal information retrieval. There also exists a certain risk that closed-source models, like GPT-3.5, will adapt during our testing with one dataset, leading to improved performance in subsequent rounds on another dataset.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Maxamed

Axmed, Kalika Bali, and Sunayana Sitaram. 2023. Mega: Multilingual evaluation of generative ai.

Loubna Ben Allal, Raymond Li, Denis Kocetkov, Chenghao Mou, Christopher Akiki, Carlos Munoz Ferrandis, Niklas Muennighoff, Mayank Mishra, Alex Gu, Manan Dey, et al. 2023. Santacoder: don't reach for the stars! *arXiv preprint arXiv:2301.03988*.

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

Akari Asai, Sneha Kudugunta, Xinyan Velocity Yu, Terra Blevins, Hila Gonen, Machel Reid, Yulia Tsvetkov, Sebastian Ruder, and Hannaneh Hajishirzi. 2023. Buffet: Benchmarking large language models for few-shot cross-lingual transfer.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Jeremy R. Cole, Aditi Chaudhary, Bhuwan Dhingra, and Partha Talukdar. 2023. Salient span masking for temporal understanding.

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.

Taishi Hosokawa, Adam Jatowt, and Kazunari Sugiyama. 2023. *Temporal Natural Language Inference: Evidence-Based Evaluation of Temporal Text Validity*, pages 441–458.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2023. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models.

Hideo Joho, Adam Jatowt, and Roi Blanco. 2015. Temporal information searching behaviour and strategies. *Information Processing Management*, 51(6):834–850.

Mayuko Kimura, Lis Kanashiro Pereira, and Ichiro Kobayashi. 2021. Towards a language model for temporal commonsense reasoning. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, pages 78–84, Online. INCOMA Ltd.

Mayuko Kimura, Lis Kanashiro Pereira, and Ichiro Kobayashi. 2022. Toward building a language model for understanding temporal commonsense. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 17–24.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2022. A systematic investigation of commonsense knowledge in large language models.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. Language models of code are few-shot commonsense learners. *arXiv preprint arXiv:2210.07128*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

John J. Nay, David Karamardian, Sarah B. Lawsky, Wenting Tao, Meghana Bhat, Raghav Jain, Aaron Travis Lee, Jonathan H. Choi, and Jungo Kasai. 2023. Large language models as tax attorneys: A case study in legal capabilities emergence.

Erik Nijkamp, Hiroaki Hayashi, Caiming Xiong, Silvio Savarese, and Yingbo Zhou. 2023. Codegen2: Lessons for training llms on programming and natural languages. *arXiv preprint*.

Lianhui Qin, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi, and Manaal Faruqui. 2021. TimeDial: Temporal Commonsense Reasoning in Dialog. In *Proc. of ACL*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question answering over temporal knowledge graphs.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, and multiple authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Julius Steen and Katja Markert. 2019. Abstractive timeline summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 21–31, Hong Kong, China. Association for Computational Linguistics.

Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324, Uppsala, Sweden. Association for Computational Linguistics.

Eshaan Tanwar, Manish Borthakur, Subhabrata Dutta, and Tanmoy Chakraborty. 2023. Multilingual llms are better cross-lingual in-context learners with alignment.

Shivin Thukral, Kunal Kukreja, and Christian Kavouras. 2021. Probing language models for understanding of temporal expressions. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 396–406, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Somin Wadhwa, Silvio Amir, and Byron C. Wallace. 2023. Revisiting relation extraction in the era of large language models.

Jiexin Wang, Adam Jatowt, Masatoshi Yoshikawa, and Yi Cai. 2023. Bitimebert: Extending pre-trained language representations with bi-temporal information.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Xiaona Xia and Wanxue Qi. 2022. Temporal tracking and early warning of multi semantic features of learning behavior. *Computers and Education: Artificial Intelligence*, 3:100045.

Kailai Yang, Shaoxiong Ji, Tianlin Zhang, Qianqian Xie, Ziyan Kuang, and Sophia Ananiadou. 2023. Towards interpretable mental health analysis with chatgpt.

Li Zhang, Liam Dugan, Hainiu Xu, and Chris Callison-Burch. 2023. Exploring the curious case of code prompts.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020. Reasoning about goals, steps, and temporal ordering with WikiHow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639, Online. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. Opt: Open pre-trained transformer language models.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369, Hong Kong, China. Association for Computational Linguistics.

Ben Zhou, Kyle Richardson, Qiang Ning, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2020. Temporal reasoning on implicit events from distant supervision. *arXiv preprint arXiv:2010.12753*.

Appendix

A Frequently Asked Questions (FAQs)

✻ **What was the rationale behind selecting these specific LLMs?**

➠ Our primary objective for utilizing a diverse set of Large Language Models (LLMs) in the analysis of temporal commonsense was to ensure a wide representation of different models. This includes those ranging from autoregressive decoder-only models to encoder-decoder models, as well as instruction-tuned models. We selected GPT-3.5 as it is among the top-performing LLMs at present. However, given its proprietary nature, we complemented it with the newly launched instruction-tuned LLMs such as BLOOMZ and Dolly. A unique aspect of these models is that ChatGPT was trained with a Reinforcement Learning from Human Feedback (RLHF) approach, while BLOOMZ and Dolly used supervised instruction fine-tuning. We added FLAN-T5 to our roster due to its distinct encoder-decoder architecture, which offers a contrast to other instruction-tuned LLMs. The inclusion of models like LLama, OPT, GPT-J, GPT-Neo served to exhibit the capabilities of an autoregressive decoder-only model. Furthermore, we integrated two code generation Language Models to contrast their performance in temporal commonsense reasoning with the general purpose LLMs.

✻ **Who was responsible for manually validating the datasets used in the Temporal-Aspect based analysis section?**

➠ We implemented heuristic and tagger-based techniques to categorize our dataset into different classes, and as authors, we also personally conducted manual verification to eliminate any potential misclassification. However, it is important to highlight that these samples were sourced from well-established and trusted datasets in the community: *MC-TACO* and *TimeDial*. Given their standardization and wide acceptance in the field, we did not find it necessary to compute Inter-Annotator Agreement (IAA) scores. We are confident in the integrity and relevance of these datasets for our study. Still, in the interest of transparency and further research, we commit to publicly sharing the manually curated data utilized in our experiments upon acceptance of our work. We believe that this does not only contribute to the reproducibility of our findings but also fosters research advancements in this field.

✻ **What is the rationale behind selecting *MC-TACO* and *TimeDial* as the foundational datasets for the Temporal-Aspect based analysis section?**

➠ Our choice to utilize *MC-TACO* and *TimeDial* as the base datasets for the Temporal-Aspect based analysis section was primarily driven by their significant size and well-established reputation in the research community. *MC-TACO* provides a vast array of approximately 9,441 data samples, while *TimeDial* furnishes an additional 5,784 examples. The wealth of data they offer surpasses most other datasets in the field. In terms of research queries like analyzing performance with varying context lengths, past versus future reasoning capabilities, and performance across different timeframes, *TimeDial* was a particularly suitable choice. Its emphasis on timing-based contexts rendered it a valuable resource. Moreover, for specialized inquiries into ambiguous temporal expression reasoning and single versus multiple event reasoning, *MC-TACO* proved indispensable. It presents a broad spectrum of diverse examples catering to these specific areas of interest. Overall, these datasets, by virtue of their depth, diversity, and relevance, enable robust, comprehensive, and nuanced analyses. They represent reliable sources for generating insights into temporal-aspect based reasoning with large language models.

✻ **What was the reason behind choosing only three prompting strategies for analysis?**

➠ In this study, we focused on utilizing three widely recognized prompting strategies: Few-shot, Zero-shot, and CoT prompting. These strategies were chosen due to their established prominence in the field and their frequent inclusion in related analysis and benchmarking works (Wadhwa et al., 2023; Nay et al., 2023). By employing these standard techniques, we ensured consistency with existing literature and provided a solid foundation for comparison and benchmarking. While we acknowledge that there have been recent advancements and emerging techniques beyond the scope of this study, we aim to emphasize that our research serves as a starting point in exploring these methods for temporal commonsense tasks. Our intention is to pave the way for future investigations that delve into the latest developments in prompting strategies, thus enabling a more comprehensive understanding of their effectiveness in temporal commonsense analysis. We remain committed to staying abreast of the evolving landscape of prompting techniques and incorporating them in future endeavors to expand the breadth and depth of our findings.

B Supplementary Material

This section provides supplementary material in the form of additional results, implementation details, etc. to bolster the reader's understanding of the concepts presented in this work.

B.1 Dataset Example for Temporal Task based Analysis

- **Event Duration (ED):** This task necessitates reasoning about event durations.
- **Event Ordering (EO):** This task calls for reasoning about the typical sequence of events.
- **Frequency (F):** This task requires reasoning about the frequency of event occurrences.
- **Stationarity (S):** This task demands reasoning about the length of state persistance.
- **Typical Time (TT):** This task needs reasoning about the specific timing of events.

Table 7 contains examples for each of these categories.

B.2 Experimental Setup

Operating System: Ubuntu 18.04.5 LTS
RAM: 220GB
GPU: NVIDIA GeForce RTX 3090 (24 GB)
Python: 3.10.11
Hyperparameters: We used temperature=0.5, top_p=1, and top_k=50, fixed across experiments, as commonly adopted values based on prior work.

B.3 Prompts

Prompt samples for all the datasets.

B.3.1 Prompts for *MC-TACO*

Few-shot Prompt for *MC-TACO*

Given the passage, the question, and the candidate answer, the task is to determine whether the candidate answer is plausible ("yes") or not ("no").
Passage: Durer's father died in 1502, and his mother died in 1513.
Question: How long was his mother ill? answer: six centuries
Response:No
###
Given the passage, the question, and the candidate answer, the task is to determine whether the candidate answer is plausible ("yes") or not ("no").
Passage: Durer's father died in 1502, and his mother died in 1513.
Question: How long was his mother ill?
answer: 6 months
Response:Yes
###
Given the passage, the question, and the candidate answer, the task is to determine whether the candidate answer is plausible ("yes") or not ("no").
Passage: Durer's father died in 1502, and his mother died in 1513.
Question: How long was his mother ill?
answer: 3 minutes
Response:No
###
Given the passage, the question, and the candidate answer, the task is to determine whether the candidate answer is plausible ("yes") or not ("no")
Passage: {Passage}
Question: {Question}
Answer: {Answer}

Response:

Zero-shot Prompt for *MC-TACO*

Given the passage, the question, and the candidate answer, the task is to determine whether the candidate answer is plausible ("yes") or not ("no").
Passage: {Passage}
Question: {Question}
Answer: {Answer}
Return your Response in Yes or No only.

Response:

Context	**Question**	**Answer**	**Label**	**Temporal Task**
Safti admits his love for Edwina to Lord Esketh , who is now sympathetic toward this good man's plight .	How long has Safti been in love with Edwina?	a year	yes	Event Duration
Durer's father died in 1502, and his mother died in 1513.	How long was his mother ill?	she was ill for 30 seconds	no	Event Duration
What we call the lab-to-fab time should be as close to zero as possible, Kelly said.	What happened after Kelly spoke?	many people agreed	yes	Event Ordering
Tim knew if the bike was going to be in any of the presents it was going to be in this box.	After Tim found the box, what happened?	he tossed it away	no	Event Ordering
Most of us have seen steam rising off a wet road after a summer rainstorm.	How often does it rain in the summer?	a couple times every month	yes	Frequency
The organization has already lost some staff through attrition and has turned away some cases, she said.	How often does the organization turn away cases?	always	no	Frequency
A thwarted Mongol invasion in 1274 weakened the Kamakura regime.	Is the invasion still taking place today?	no	yes	Stationarity
Tony and Ally like to play other games like hopscotch or jump rope but that day they joined the game of tag.	Do Tony and Ally still enjoy jump rope?	no	no	Stationarity
Johnson is a justice on California's Second District Court of Appeal.	When did Johnson arrive to court?	9:00 AM	yes	Typical Time
She had it for a long time so it is now a dark brown color.	What time did she buy it?	1:00 AM	no	Typical Time

Table 7: Examples from datasets for Temporal Task based Analysis

CoT Prompt for *MC-TACO*

Given the passage, the question, and the candidate answer, the task is to determine whether the candidate answer is plausible ("yes") or not ("no"). Final Label will be yes or no only.
Passage: Durer’s father died in 1502, and his mother died in 1513.
Question: How long was his mother ill?
Answer: six centuries
Response: Let’s think step by step. Question is asking about the duration for which Durer’s mother was ill and answer mentioned it is six centuries. It is not possible as any human can’t live for such a long period of time. So the answer is not plausible.
Final Label: No
###
Given the passage, the question, and the candidate answer, the task is to determine whether the candidate answer is plausible ("yes") or not ("no").
Passage: Durer’s father died in 1502, and his mother died in 1513.
Question: How long was his mother ill?
Answer: 6 months
Response: Let’s think step by step. Question is asking about the duration for which Durer’s mother was ill and answer mentioned it was 6 months. This can be possible as duration for illness as many illness lasts for such periods only.
Final Label: Yes
###
Given the passage, the question, and the candidate answer, the task is to determine whether the candidate answer is plausible ("yes") or not ("no").
Passage: Durer’s father died in 1502, and his mother died in 1513.
Question: How long was his mother ill?
Answer: 3 minutes
Response: Let’s think step by step. Question is asking about the duration for which Durer’s mother was ill and answer mentioned it was 3 minutes which can’t be possible as no illness last for such small time period.
Final Label: No
###
Given the passage, the question, and the candidate answer, the task is to determine whether the candidate answer is plausible ("yes") or not ("no").
Passage: {Passage}
Question: {Question}
Answer: {Answer}
Response: Let’s think step by step.

Code Prompt for *MC-TACO*

```
import Question_answering

class Context:
    '''Strictly return only "0" or "1" for the
   given answer to the question, based on context
   and type of answer, task is to determine whether
    the given candidate answer is
    plausible ("1") or not ("0").'''


   def __init__(self, question,context,answer,type):
        self.context = context # The context
        self.question = question # The question
        self.answer = answer # answer
        self.type = type # type of the answer

    def get_answer(self):
        ans = Question_answering(self.question,
                                self.context,
                                self.answer,
                                self.type)
        return ans  #Strictly return 0/1 only


context = Context(
    context = "{}",
    question = "{}"
    answer = "{}"
    type = "{}"
)
assert(context.get_answer
       == .format(question,context,answer,type)
```

B.3.2 Prompts for *TimeDial*

Few-shot Prompt for *TimeDial*

You are given a conversation between two persons. Each conversation has a fill-in-the-blank in them which is represented by a <MASK> token. You are also given a candidate answer for that <MASK> token. Your task is to determine whether the candidate answer is plausible (Yes) or not (No). Return your answer in Yes and No only.
Conversation:
A:We need to take the accounts system offline to carry out the upgrade . But don't worry , it won't cause too much inconvenience . We're going to do it over the weekend.
B: How long will the system be down for?
A: We'll be taking everything offline in about two hours ' time . It'll be down for a minimum of twelve hours . If everything goes according to plan , it should be up again by 6 pm on Saturday.
B: That's fine . We've allowed <MASK> to be on the safe side.
Answer: forty-eight hours
Label:Yes
###
Conversation:
A:Mr . Emory , I ' d like to take this afternoon off if it's all right with you.
B: But Steven , you've called in sick 3 times during <MASK>.
A: I know , Mr . Emory . I'm sorry . But I really need to see the doctor this afternoon. I feel dizzy and I can't concentrate on my work.
B: All right , then . But don't forget to bring a doctors note tomorrow.
A: OK , thank you !
Answer: last 15 seconds
Label:No
###
Conversation: {Conversation}
Answer: {Answer}

Label:

Zero-shot Prompt for *TimeDial*

You are given a conversation between two persons. Each conversation has a fill-in-the-blank in them which is represented by a <MASK> token. You are also given a candidate answer for that <MASK> token. Your task is to determine whether the candidate answer is plausible (Yes) or not (No). Strictly return your answer in Yes and No only.
Conversation: {Conversation}
Answer: {Answer}

Label:

CoT Prompt for *TimeDial*

You are given a conversation between two persons. Each conversation has a fill-in-the-blank in them which is represented by a <MASK> token. You are also given a candidate answer for that <MASK> token. Your task is to determine whether the candidate answer is plausible (Yes) or not (No) by first generating first reasoning and then final label as Yes or No.
Conversation:
A:We need to take the accounts system offline to carry out the upgrade . But don't worry , it won't cause too much inconvenience . We're going to do it over the weekend.
B: How long will the system be down for?
A: We'll be taking everything offline in about two hours ' time . It'll be down for a minimum of twelve hours . If everything goes according to plan , it should be up again by 6 pm on Saturday.
B: That's fine . We've allowed <MASK> to be on the safe side .
Answer: forty-eight hours
Label: Let's think step by step. The conversation seems to be taking place in an office environment where speaker A says that they will take systems down for 12 hrs over the weekend for maintenance. and speaker B says they are allowed <mask> hrs without systems and answer here says mask should be 48 hrs. This answers seems to be correct as it is weekend which means for 48 hrs they don't need to work on these systems.
Final Label:Yes
###
Conversation:
A:Mr . Emory , I ' d like to take this afternoon off if it's all right with you.
B: But Steven , you've called in sick 3 times during <MASK>.
A: I know , Mr . Emory . I'm sorry . But I really need to see the doctor this afternoon. I feel dizzy and I can't concentrate on my work.
B: All right , then . But don't forget to bring a doctors note tomorrow.
A: OK , thank you !
Answer: last 15 seconds
Label: Let's think step by step. The conversation seems to be taking place in an office environment where speaker A is asking for a leave to speaker B and Speaker B said that he already took 3 leaves during <MASK> time period. The answer here says its 15 seconds which is not possible as leaves are taken during a week or a month. That's why this answer is incorrect.
Final Label:No
###
Conversation: {Conversation}
Answer: {Answer}
Label: Let's think step by step.

Code Prompt for *TimeDial*

```
import Fill_blank_from_conversation

class Conversation:
    '''Conversation between two persons is given
   and each conversation has a fill-in-the-blank
   in them which is represented by a <MASK> token.
    You are also given a candidate answer
    for that <MASK> token. Determine whether the
   candidate answer is plausible ("1") or not ("0").
    Strictly return "0" or "1" only.'''

    def __init__(self,conversation,answer):
        self.conversation = conversation
        # Converstion between two persons
        self.answer = answer
        # answer to the fill in the blank

    def get_answer(self):
        ans = Fill_blank_from_conversation(
                             self.conversation,
                                self.answer)
        return ans  #Strictly return 0/1 only


conversation = Conversation(
    conversation = "{}",
    answer = "{}"
)
assert(conversation.get_answer
        == .format(conversation,answer)
```

B.3.3 Prompt for *TNLI*

Few-shot Prompt for *TNLI*

You are given two sentences, Sentence 1, and Sentence 2 where Sentence 1 is a hypothesis, and Sentence 2 is a premise sentence. The task is to assign one of the following three classes to Sentence 1 based on the inference using the content of Sentence 2. The labels are Support, Invalidate, and Neutral. The Support class means that Sentence 1 is still valid given the information in Sentence 2. The Invalidate class, on the other hand, means that Sentence 1 ceased to be valid in view of Sentence 2. The third one, Neutral class, indicates that the situation evidence is not conclusive or clear, and we cannot verify the validity of the hypothesis.
Sentence 1: A female is scrambling eggs in a bowl.
Sentence 2: Eggs are scrambled in a bowl.
Label: Support
###
Sentence 1: A group of people sing and dance at a concert.
Sentence 2: A group of people going to take rest.
Label: Invalidate
###
Sentence 1: The horses race on the dirt track while their riders urge them on.
Sentence 2: Most people enjoy watching horse racing.
Label: Neutral
###
Sentence 1: {Sentence 1}
Sentence 2: {Sentence 2}

Label:

Zero-shot Prompt for *TNLI*

You are given two sentences, Sentence 1, and Sentence 2 where Sentence 1 is a hypothesis, and Sentence 2 is a premise sentence. The task is to assign one of the following three classes to Sentence 1 based on the inference using the content of Sentence 2. The labels are Support, Invalidate, and Neutral. The SUPPORT class means that Sentence 1 is still valid given the information in Sentence 2. The INVALIDATE class, on the other hand, means that Sentence 1 ceased to be valid in view of Sentence 2. The third one, Neutral class, indicates that the situation evidence is not conclusive or clear, and we cannot verify the validity of the hypothesis.
Sentence 1: {Sentence 1}
Sentence 2: {Sentence 2}

Label:

CoT Prompt for *TNLI*

You are given two sentences, Sentence 1, and Sentence 2 where Sentence 1 is a hypothesis, and Sentence 2 is a premise sentence. The task is to assign one of the following three classes to Sentence 1 based on the inference using the content of Sentence 2. The labels are Support, Invalidate, and Neutral. The SUPPORT class means that Sentence 1 is still valid given the information in Sentence 2. The INVALIDATE class, on the other hand, means that Sentence 1 ceased to be valid in view of Sentence 2. The third one, Neutral class, indicates that the situation evidence is not conclusive or clear, and we cannot verify the validity of the hypothesis.
Sentence 1: A group of people sing and dance at a concert
Sentence 2: A group of people going to take rest.
Label: Let's think step by step. In Sentence 2, it is mentioned that group of people are taking rest. This implies that they won't perform any activity. But in Sentence 1 it is mentioned that group of people are singing and dancing at concert. But based on information from Sentence 1, it is not possible as they are taking rest. So the final label should be Invalidate.
Label:Invalidate
###
Sentence 1: A female is scrambling eggs in a bowl.
Sentence 2: Eggs are scrambled in a bowl.
Label: Let's think step by step. In Sentence 2, it is mentioned that eggs are scrambled in a bowl. However, in Sentence 1, a female is scrambling eggs in a bowl which supports the statement of Sentence 2 that eggs are scrambled. So the final label should be Support.
Label:Support
###
Sentence 1: The horses race on the dirt track while their riders urge them on.
Sentence 2: Most people enjoy watching horse racing.
Label: Let's think step by step. In Sentence 2,it is mentioned that Most people enjoy watching horse racing. However, in Sentence 1, The horses race on the dirt track while their riders urge them on. Both these statements are neither supporting each other nor invalidating each other. So the final label should be Neutral.
Label:Neutral
###
Sentence 1: {Sentence 1}
Sentence 2: {Sentence 2}
Label: Let's think step by step.

Code Prompt for *TNLI*

```
import neuralnli

class NaturalLanguageInference():
    '''function to answer the natural language
   inference task given premise and hypothesis.''''

    def __init__(self):
        self.model = neuralnli()

    def forward(self, premise, hypothesis):
        answer = self.model(premise,
                          hypothesis)['answer']
        return answer

nli_model = NaturalLanguageInference()

premise = "{}"
hypothesis = "{}.

        #Invalidate, Support, or Neutral?"

answer = nli_model.forward(premise, hypothesis)

assert answer ==.format(statement2,statement1)
```

B.3.4 Prompts for *WikiHow* dataset

Few-shot Prompt for *WikiHow*

You are given a goal and steps to accomplish that goal. Your task is to determine whether the steps are in right order (Yes) or not (No). Return your answer as Yes and No only.
Goal: How to Select a Dog Bed - Understanding Different Types of Beds
Steps: Buy a mat for the easiest solution. Pick a pillow bed for a large dog. Select a donut bed if your dog likes to feel secure. Purchase a nest bed for cuddling comfort. Buy a bolster-type bed if your dog is a leaner. Look for a cave-style bed if your dog likes to burrow. Consider a hammock bed for ease of cleaning.
Answer: No
###
Goal: How to Get Married in Oregon - Planning a Wedding Ceremony
Steps: Decide on the type of ceremony. Choose a season. Hire wedding vendors. Confirm the date with vendors and officiants. Make final payments.
Answer: Yes
###
Goal: {Goal}
Steps: {Steps}

Answer:

Zero-shot Prompt for *WikiHow*

You are given a goal and steps to accomplish that goal. Your task is to determine whether the steps are in right order (Yes) or not (No). Return your answer in Yes and No only.
Goal: {Goal}
Steps: {Steps}

Answer:

CoT Prompt for *WikiHow*

You are given a goal and steps to accomplish that goal. Your task is to determine whether the steps are in right order (Yes) or not (No). Return your answer in Yes and No only.
Goal:How to Select a Dog Bed - Understanding Different Types of Beds
Steps:Buy a mat for the easiest solution. Pick a pillow bed for a large dog. Select a donut bed if your dog likes to feel secure. Purchase a nest bed for cuddling comfort. Buy a bolster-type bed if your dog is a leaner. Look for a cave-style bed if your dog likes to burrow. Consider a hammock bed for ease of cleaning.
Answer: Let's think step by step. The correct order should be: Look for a cave-style bed if your dog likes to burrow. Select a donut bed if your dog likes to feel secure. Purchase a nest bed for cuddling comfort. Buy a bolster-type bed if your dog is a leaner. Consider a hammock bed for ease of cleaning. Buy a mat for the easiest solution. Pick a pillow bed for a large dog. As this sequence is not in match with given steps, so the final answer is No.
Answer: No
###
Goal:How to Get Married in Oregon - Planning a Wedding Ceremony
Steps:Decide on the type of ceremony. Choose a season. Hire wedding vendors. Confirm the date with vendors and officiants. Make final payments.
Answer: Let's think step by step. The correct order should be: To get married, one first need to decide a ceremony. Then choose a season. Then hire a wedding vendor for organization. Confirm and finalize a date. Then make the final payments. As this sequence is in the match with given steps, so the final answer is Yes.
Answer: Yes
###
Goal: {Goal}
Steps: {Steps}
Answer: Let's think step by step.

Code Prompt for *WikiHow*

```
import order_steps

class Event:
   '''Given a goal and steps to achieve, determine
   whether the steps are in right order or not. Return
   Yes if right order and No if order is wrong.'''

    def __init__(self, goal, steps):
        self.goal = goal

                '''The goal that someone
                is trying to accomplish'''

        self.steps = steps # All the steps

    def get_order_of_steps(self):
        # Output a Binary response Yes or no
      return order_steps(self.goal, self.steps)

event = Event(
    goal = "{goal}"
    steps = "{steps}"
)
assert(event.get_order_of_steps
             == <fim-suffix>.format(goal,steps)
```

B.3.5 Prompts for *BIG-bench*

Few-shot Prompt for *BIG-bench*

Q: Today, Emily went to the museum. Between what times could they have gone?
We know that: Emily woke up at 1pm. Elizabeth saw Emily reading at the library from 2pm to 4pm. Jessica saw Emily watching a movie at the theater from 4pm to 5pm. Leslie saw Emily waiting at the airport from 5pm to 6pm. William saw Emily buying clothes at the mall from 6pm to 7pm. The museum was closed after 7pm.
Between what times could Emily have gone to the museum?
Options:
(A) 1pm to 2pm
(B) 6pm to 7pm
(C) 5pm to 6pm
(D) 2pm to 4pm
Strictly return the correct option which means return the letter of choice only
Ans:A
###
Q: Today, Tiffany went to the beach. Between what times could they have gone?
We know that: Tiffany woke up at 5am. Betty saw Tiffany getting a coffee at the cafe from 5am to 6am. Jessica saw Tiffany working at the office from 6am to 9am. John saw Tiffany stretching at a yoga studio from 9am to 12pm. Sean saw Tiffany sitting on a rooftop from 12pm to 2pm. Sarah saw Tiffany playing tennis at the tennis court from 2pm to 3pm. The beach was closed after 4pm.
Between what times could Tiffany have gone to the beach?
Options:
(A) 9am to 12pm
(B) 12pm to 2pm
(C) 5am to 6am
(D) 3pm to 4pm
Strictly return the correct option which means return the letter of choice only
Ans:D
###
input: {input}
Strictly return the correct option which means return the letter of choice only

Ans:

Zero-shot Prompt for *BIG-bench*

Task description: Answer questions about which times certain events could have occurred Always return option letter at the end. There won't be any case when answer will be none of the options. Return the correct option only A,B,C or D.
Input: {Input}

Ans:

CoT Prompt for *BIG-bench*

Task description: Answer questions about which times certain events could have occurred.
Q: Today, Emily went to the museum. Between what times could they have gone?
We know that: Emily woke up at 1pm. Elizabeth saw Emily reading at the library from 2pm to 4pm. Jessica saw Emily watching a movie at the theater from 4pm to 5pm. Leslie saw Emily waiting at the airport from 5pm to 6pm. William saw Emily buying clothes at the mall from 6pm to 7pm. The museum was closed after 7pm.
Between what times could Emily have gone to the museum?
Options:
(A) 1pm to 2pm
(B) 6pm to 7pm
(C) 5pm to 6pm
(D) 2pm to 4pm
Answer: Let's think step by step. Wake-up time: 1pm. 1pm-2pm: free. 2pm-4pm: reading at the library. 4pm-5pm: watching a movie at the theater. 5pm-6pm: waiting at the airport. 6pm-7pm: buying clothes at the mall. The museum closure time: 7pm. The only time when Emily could have gone to the museum was 1pm to 2pm. So the answer is (A).
Answer: (A)
###
Input: {Input}
Answer: Let's think step by step.

Code Prompt for *BIG-bench*

```
import Scheduling_question_answering

class Context:
    '''choose the right option number for
       the question depending on the context'''

   def __init__(self, question,context,options):
        self.context = context # The context
        self.question = question # The question
        self.options = options # options

    def get_answer(self):
        answer = Scheduling_question_answering(

      self.question, self.context,self.options)
        return answer


context = Context(
    context = "{}",
    question = "{}"
    Options = "{}"
)
assert(context.get_answer
==.format(listToString(Context),Question,options)
```

B.3.6 Prompts for TimeQA

Zero-shot Prompt for *TimeQA*

Answer the question based on the context.
Context: {Context}
Question: {Question}
Answer:.

CoT Prompt for *TimeQA*

Answer the question based on the context. Answer will be in the context. First generate a reasoning and then provide final answer in a new line as Final Answer:.
Context: {Context}
Question: {Question}
Answer: Let's think step by step.

Code Prompt for *TimeQA*

```
import Question_answering

class Context:
'''Return answer for this question
                based on context'''

     def __init__(self,context,question):
     self.context = context # The context
     self.question = question # The question

     def get_answer(self):
     ans = Question_answering(self.context,
                                 self.question)
     return ans


context = Context(
context = "{}",
question = "{}"
)
assert(context.get_answer ==.format(context,
                                      question)
```

B.4 Dataset Samples

Example instances for all the dataset present in this study.

B.4.1 *MC-TACO*

***MC-TACO* dataset instance examples**

Context: Durer's father died in 1502, and his mother died in 1513.
Question: How long was his mother ill?
Answer: she was ill for 30 seconds
Label: no
Temporal Reasoning: Event Duration

Context: Safti admits his love for Edwina to Lord Esketh , who is now sympathetic toward this good man's plight.
Question: Has Safti always been in love with Edwina?
Answer: no this ' s a new thing
Label: yes
Temporal Reasoning: Stationarity

Context: The next evening, she arrived with a stack of glistening stopboxes containing sushi, sashimi, oysters in their shells, and Terran vegetables fresh plucked from their hydroponic beds.
Question: At what time did she arrive?
Answer: 6:00 PM
Label: yes
Temporal Reasoning: Typical Time

Context: The CIA now estimates that it cost al Qaeda about $30 million per year to sustain its activities before 9/11 and that this money was raised almost entirely through donations.
Question: What happened to al Qaeda's finances after 9/11?
Answer: they were dealt a big blow
Label: yes
Temporal Reasoning: Event Ordering

Context: This is an astonishing new record for a coin, he said.
Question: How often are new records established?
Answer: three times an second
Label: no
Temporal Reasoning: Frequency

B.4.2 *TimeDial*

***TimeDial* dataset instance examples**

Conversation: A:We need to take the accounts system offline to carry out the upgrade . But don’t worry , it won’t cause too much inconvenience . We’re going to do it over the weekend .
B: How long will the system be down for ?
A: We’ll be taking everything offline in about two hours ’ time . It’ll be down for a minimum of twelve hours . If everything goes according to plan , it should be up again by 6 pm on Saturday .
B: That’s fine . We’ve allowed <MASK> to be on the safe side .
Answer: forty-eight hours
Label: 1

Conversation: A:Excuse me , Miss .
B: Yes . May I help you ?
A: I’m a graduate student here in mathematics . I’ve just come from China and I’ve never used a western library before . I’ll be here for <MASK> , so I’d like to learn to use the library as efficiently as possible . I wonder if someone might have time to show me around .
B: I’d be very glad to show you around , but I’m very busy right now . Could you come back about 3 thirty ?
A: Sure . 3 thirty this afternoon .
B: Good . See you later .
A: Thank you . Good-bye .
Answer: 3 decades
Label: 0

B.4.3 *TNLI*

***TNLI* dataset instance examples**

Sentence 1: The woman wearing the pink jacket has thrown a Frisbee for the dog to catch.
Sentence 2: The dog falling into a lake trying to catch the frisbee.
Label: Support

Sentence 1: A young boy skipping down a tennis court in absolute glee.
Sentence 2: He is now at a volleyball court.
Label: Invalidate

Sentence 1: A man sitting on sidewalk with shirt over his head.
Sentence 2: Most people prefer sitting to standing.
Label: Neutral

B.4.4 *WikiHow*

***WikiHow* dataset instance examples**

Goal: How to Buy a Used Sailboat - Engine
Steps: Steer clear of rare or very old engines unless you’re certain there’s an adequate supply of parts. Do the Smoke Test: healthy diesels make small amounts of black smoke with some white on cold starts. Check for fuel leaks and a working bilge blower in gasoline engines. Before the seller cranks the engine, check to see if it is already warm.
Ordered?: 0

Goal: How to Breed Alpacas - Encouraging Reproduction
Steps: Expose the breeding male to the female. Induce ovulation in the female alpaca. Place the male and female alpaca in the breeding pen. Separate the alpacas if the female is not receptive. Wait a week or two after copulation to re-mate alpacas.
Ordered?: 1

B.4.5 *BIG-bench*

***BIG-bench* dataset instance examples**

Input: Today, James went to the beach. Between what times could they have gone?
We know that: James woke up at 5am. Sean saw James walking towards the Statue of Liberty from 5am to 6am. Michael saw James driving to the water park from 6am to 7am. Anthony saw James reading at the library from 7am to 3pm. William saw James getting a coffee at the cafe from 4pm to 9pm. The beach was closed after 9pm.
Between what times could James have gone to the beach?
Options:
(A) 7am to 3pm
(B) 5am to 6am
(C) 4pm to 9pm
(D) 3pm to 4pm
Answer: (D)

Input: Today, David went to the art studio. Between what times could they have gone?
We know that: David woke up at 5am. Linda saw David watching a movie at the theater from 5am to 7am. James saw David buying lunch at the deli from 9am to 10am. Mary saw David buying a phone at the electronics store from 10am to 11am. Leslie saw David driving to the water park from 11am to 2pm. Jessica saw David buying a bike at the bike shop from 2pm to 7pm. The art studio was closed after 7pm.
Between what times could David have gone to the art studio?
Options:
(A) 7am to 9am
(B) 2pm to 7pm
(C) 5am to 7am
(D) 11am to 2pm
Answer: (A)

B.4.6 *TimeQA*

***TimeQA* dataset instance examples**

Context: HMAS Wollongong (J172) HMAS Wollongong (J172) , named for the city of Wollongong , New South Wales , was one of 60 s constructed during World War II and one of 20 built for the Admiralty but manned by personnel of and commissioned into the Royal Australian Navy (RAN) . Design and construction . In 1938 , the Australian Commonwealth Naval Board (ACNB) identified the need for a general purpose local defence vessel capable of both anti-submarine and mine-warfare duties , while easy to construct and operate . The vessel was initially envisaged as having a displacement of approximately 500 tons , a speed of at least , and a range of The opportunity to build a prototype in the place of a cancelled Bar-class boom defence vessel saw the proposed design increased to a 680-ton vessel , with a top speed , and a range of , armed with a 4-inch gun , equipped with asdic , and able to fitted with either depth charges or minesweeping equipment depending on the planned operations : although closer in size to a sloop than a local defence vessel , the resulting increased capabilities were accepted due to advantages over British-designed mine warfare and anti-submarine vessels. ...
Question: Which Navy operated the warship HMAS Wollongong from 1950 to 1951?
Answer: Indonesian Navy

B.5 Detailed Tables for Model and Prompt based Analysis

Table 8 compares the performance of eight Large Language Models on six datasets, analyzed under two different prompting strategies across F1 score and accuracy. Table 9 compares the performance of Instruction Tuned LLMs with CoT prompting strategy across both F1 and Accuracy. Table 10 compares the performance of Code Generation LMs with Code prompts across all datasets on both F1 and accuracy.

	MC-TACO		*TNLI*		*TimeDial*		*WikiHow*		*BIG-bench*		*TimeQA*
	Few-shot	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Few-shot**	**Zero-shot**	**Zero-shot**
Model	**Acc/F1**	**Acc/F1**	**Acc/F1**	**Acc/F1**	**Acc/F1**	**Acc/F1**	**Acc/F1**	**Acc/F1**	**Acc/F1**	**Acc/F1**	**EM/F1**
GPT-3.5	**0.8/0.8**	**0.75/0.72** (⇓5%)	**0.62/0.62**	**0.5/0.49** (⇓12%)	**0.65/0.64**	**0.65/0.65** (=0%)	**0.55**/0.51	0.49/0.45 (⇓6%)	0.25/**0.24**	0.26/0.23 (⇑1%)	0.1/0.21
FLAN-T5	0.7/0.74	0.74/0.71 (⇑4%)	0.39/0.3	0.37/0.25 (⇓2%)	0.54/0.48	0.54/0.49 (=0%)	0.45/0.3	0.45/0.28 (=0%)	0.16/0.15	0.14/0.13 (⇓2%)	**0.4/0.5**
BLOOMZ	0.59/0.6	0.63/0.59 (⇑4%)	0.34/0.2	0.31/0.21 (⇓3%)	0.49/0.44	0.46/0.44 (⇓3%)	0.53/0.41	0.46/0.29 (⇓7%)	**0.28**/0.13	**0.28**/0.12 (=0%)	-
Dolly	0.45/0.47	0.52/0.53 (⇑7%)	0.42/0.37	0.32/0.16 (⇓10%)	0.5/0.39	0.49/0.45 (⇓1%)	0.45/0.37	0.44/0.29 (⇓1%)	0.23/0.17	0.27/0.11 (⇑4%)	0/0.18
GPT-J	0.59/0.6	0.37/0.27 (⇓22%)	0.34/0.2	0.33/0.16 (⇓1%)	0.45/0.33	0.33/0.33 (⇓12%)	0.46/0.3	0.45/0.3 (⇓1%)	0.26/0.22	0.26/**0.24** (=0%)	-
GPT Neo	0.59/0.58	0.37/0.32 (⇓22%)	0.32/0.23	0.33/0.22 (⇑1%)	0.49/0.43	0.48/0.38 (⇓1%)	0.47/0.33	0.44/0.39 (⇓3%)	0.26/0.13	0.27/0.12 (⇑1%)	-
LLaMA	0.65//0.55	0.65/0.53 (=0%)	0.32/0.21	0.32/0.26 (=0%)	0.5/0.45	0.35/0.26 (⇓15%)	**0.55/0.52**	**0.54/0.52** (⇓1%)	**0.28**/0.13	0.18/0.16 (⇓10%)	0.1/0.12
OPT	0.65/0.54	0.45/0.46 (⇓20%)	0.33/0.16	0.35/0.26 (⇑2%)	0.5/0.38	0.49/0.33 (⇓1%)	0.46/0.3	0.49/0.43 (⇑3%)	0.24/0.1	0.23/0.10 (⇓1%)	-

Table 8: The performance of eight Large Language Models on six datasets, analyzed under two different prompting strategies. "Acc" stands for Accuracy, "F1" signifies Weighted-F1 score, and "EM" corresponds to Exact Match. The percentage changes in accuracy performance between zero-shot and Few-shot prompting are indicated in parentheses.

	MC-Taco	*TNLI*	*TimeDial*	*WikiHow*	*BIG-bench*	*TimeQA*
Model	**Acc/F1**	**Acc/F1**	**Acc/F1**	**Acc/F1**	**Acc/F1**	**EM/F1**
GPT-3.5	**0.82/0.82** (⇑2%)	**0.46/0.45** (⇓16%)	**0.7/0.69** (⇑5%)	**0.54/0.53** (⇓1%)	**0.67/0.67** (⇑42%)	**0.15/0.22**
FLAN-T5	0.73/0.74 (⇑3%)	0.38.0.3 (⇓1%)	0.54/0.48 (0%)	0.48/0.46 (⇑3%)	0.21/0.21 (⇑5%)	0.05/0.15
BLOOMZ	0.45/0.45 (⇓14%)	0.36/0.29 (⇑2%)	0.53/0.52 (⇑4%)	0.5/0.48 (⇓3%)	0.3/0.22 (⇑2%)	-
Dolly	0.48/0.48 (⇑3%)	0.34/0.2 (⇓8%)	0.5/0.47 (0%)	0.47/0.37 (⇑2%)	0.29/0.21 (⇑6%)	0/0.15

Table 9: Performance of Instruction Tuned LLMs with CoT prompting strategy. The percentage changes in accuracy performance between CoT and Few-shot prompting (from Table 8) are indicated in parentheses.

B.6 Detailed Table for Temporal Aspect Based Analysis

Table 11 compares the performance of LLMs across different temporal expressions. Table 12 compares the performance of LLMs across past and future reasoning events. Table 13 compares the performance of LLMs across different timeframes. Table 14 compares the performance of LLMs across multiple events and a single event. Table 15 compares the performance of LLMs across different context lengths.

B.7 Detailed Table for Temporal Task-based Analysis

Table 16 compares the performance of eight Large Language Models on MC-TACO dataset and its fine-grained temporal task categories across both F1 and accuracy metrics.

	MC-TACO	*TNLI*	*TimeDial*	*WikiHow*	*BIG-bench*	*TimeQA*
Model	**Acc/F1**	**Acc/F1**	**Acc/F1**	**Acc/F1**	**Acc/F1**	**EM/F1**
`GPT-3.5`	0.5/0.5	**0.45/0.41**	0.49/0.43	**0.45/0.29**	**0.29/0.29**	**0.1/0.13**
`SantaCoder`	0.5/0.51	0.36/0.29	**0.5/0.5**	**0.45/0.29**	0.27/0.19	-
`CodeGen2`	**0.61/0.6**	0.35/0.25	**0.5**/0.41	**0.45/0.29**	0.25/0.19	**0.1/0.13**

Table 10: Performance of Code Generation LMs with Code prompts in zero-shot setting

	ED		EO		F		S		TT	
Model	**Exact**	**Ambiguous**	**Exact**	**Ambiguous**	**Exact**	**Ambiguous**	**Exact**	**Ambiguous**	**Exact**	**Ambiguous**
`GPT-3.5`	**0.87**	**0.89**	**0.84**	**0.85**	**0.83**	**0.85**	**0.76**	**0.79**	0.62	**0.81**
`FLAN T5`	0.81	0.85	0.7	0.73	0.77	0.79	0.58	**0.79**	0.56	0.77
`LLaMA`	0.75	0.74	0.5	0.54	0.41	0.61	0.37	0.57	**0.63**	0.63

Table 11: Performance of LLMs in accuracy across different temporal expressions in few-shot setting (ED: Event Duration, EO: Event Ordering, F: Frequency, S: Stationarity, TT: Typical Time)

Model	**Past**	**Future**
`GPT-3.5`	**0.62**	**0.60**
`FLAN-T5`	0.55	0.50
`LLaMA`	0.52	0.50

Table 12: Performance of LLMs in accuracy on past vs. future based reasoning tasks in few-shot setting

Model	**Seconds**	**Minutes**	**Hours**	**1 Day**	**More than a Day**
`GPT-3.5`	**0.5**	**0.65**	**0.6**	**0.69**	**0.7**
`FLAN-T5`	0.43	0.48	0.55	0.5	0.56
`LLaMA`	0.43	0.48	0.52	0.55	0.5

Table 13: Performance of LLMs in accuracy across different Timeframes in few-shot setting

	ED		**EO**		**F**		**S**		**TT**	
Model	**ME**	**SE**	**ME**	**SE**	**ME**	**SE**	**ME**	**SE**	**ME**	**SE**
`GPT-3.5`	**0.91**	**0.82**	**0.75**	**0.88**	**0.9**	**0.88**	**0.7**	**0.8**	**0.68**	**0.68**
`FLAN-T5`	0.85	0.8	0.57	0.75	0.89	0.79	0.5	0.72	0.58	0.6
`LLaMA`	0.72	0.41	0.5	0.5	0.78	0.6	0.4	0.62	0.44	0.54

Table 14: Performance of LLMs in accuracy across single and multiple events in few-shot setting. (ME: Multiple events, SE: Single Events, ED: Event Duration, EO: Event Ordering, F: Frequency, S: Stationarity, TT: Typical Time)

Model	**0-200**	**200-400**	**400-600**
`GPT-3.5`	**0.66**	**0.63**	**0.58**
`FLAN-T5`	0.53	0.55	0.55
`LLaMA`	0.51	0.48	0.46

Table 15: Performance of LLMs in accuracy across different context length in few-shot setting

	MC						***MC-ED***						***MC-EO***						***MC-F***						***MC-S***						***MC-TT***					
	FS		**ZS**		**CoT**		**FS**		**ZS**		**CoT**		**FS**		**ZS**		**CoT**		**FS**		**ZS**		**CoT**		**FS**		**ZS**		**CoT**		**FS**		**ZS**		**CoT**	
Model	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**
GPT-3.5	**0.8**	**0.8**	**0.75**	**0.72**	**0.82**	**0.82**	**0.85**	**0.85**	**0.8**	0.77	**0.87**	**0.86**	**0.83**	**0.82**	**0.77**	**0.76**	**0.84**	**0.84**	**0.84**	**0.82**	**0.82**	**0.8**	**0.85**	**0.85**	**0.77**	**0.77**	**0.69**	**0.67**	**0.76**	**0.75**	**0.69**	**0.67**	**0.62**	0.53	**0.71**	**0.69**
FLAN-T5	0.7	0.74	0.74	0.71	0.73	0.74	0.8	0.77	**0.8**	**0.78**	0.78	0.77	0.73	0.71	0.76	0.74	0.76	0.76	0.77	0.73	0.79	0.75	0.76	0.73	0.7	0.69	0.68	0.66	0.62	0.6	0.62	0.53	**0.62**	**0.56**	0.6	0.54
BLOOMZ	0.59	0.6	0.63	0.59	0.45	0.45	0.61	0.62	**0.8**	0.76	0.35	0.32	0.59	0.59	**0.77**	**0.76**	0.6	0.6	0.63	0.63	0.83	0.8	0.45	0.47	0.51	0.5	0.68	0.66	0.43	0.33	0.53	0.53	0.61	0.5	0.5	0.5
Dolly	0.45	0.47	0.52	0.53	0.48	0.48	0.45	0.46	0.5	0.52	0.46	0.48	0.50	0.50	0.55	0.55	0.52	0.48	0.49	0.50	0.52	0.55	0.48	0.5	0.49	0.48	0.61	0.61	0.53	0.51	0.52	0.51	0.52	0.5	0.48	0.45
GPT-J	0.59	0.6	0.37	0.27	-	-	0.58	0.6	0.32	0.25	-	-	0.61	0.61	0.5	0.38	-	-	0.65	0.64	0.3	0.21	-	-	0.56	0.55	0.42	0.37	-	-	0.55	0.54	0.45	0.31	-	-
GPT Neo	0.59	0.58	0.37	0.32	-	-	0.66	0.64	0.29	0.19	-	-	0.52	0.51	0.5	0.5	-	-	0.61	0.62	0.31	0.25	-	-	0.5	0.49	0.44	0.36	-	-	0.54	0.49	0.5	0.48	-	-
LLaMA	0.65	0.55	0.65	0.53	-	-	0.73	0.65	0.65	0.73	-	-	0.54	0.48	0.53	0.4	-	-	0.71	0.64	0.73	0.62	-	-	0.57	0.49	0.54	0.43	-	-	0.56	0.42	0.56	0.41	-	-
OPT	0.65	0.54	0.45	0.46	-	-	0.73	0.63	0.35	0.32	-	-	0.52	0.36	0.47	0.42	-	-	0.75	0.63	0.55	0.56	-	-	0.46	0.45	0.45	0.31	-	-	0.55	0.42	0.48	0.47	-	-

Table 16: Performance of eight Large Language Models on MC-TACO dataset and it's fine-grained temporal task categories (MC: MC-TACO, FS: Few-Shot, ZS: Zero-Shot)