Benchmarking the Ability of Large Language Models to Reason about Event Sequences

Anonymous ACL submission

Abstract

The ability to reason about events and their temporal relations is a key aspect in Natural Language Understanding. In this paper, we investigate the ability of Large Language Models to resolve temporal references with respect to longer event sequences. Given that events rarely occur in isolation, it is crucial to determine the extent to which Large Language Models can reason about longer sequences of events. Towards this goal, we introduce a novel synthetic benchmark dataset comprising of 2200 questions to test the abilities of LLMs to reason about events using a Question Answering task as proxy. We compare the performance of 4 state of the art LLMs on the benchmark, analyzing their performance in dependence of the length of the event sequence considered as well as of the explicitness of the temporal reference. Our results show that, while the benchmarked LLMs can answer questions over event sequences with a handful of events and explicit temporal references successfully, performance clearly deteriorates with larger event sequence length and when temporal references get less explicit.

1 Introduction

Events are pervasive in our lives and as such we frequently refer to events when we speak. In fact, the ability to reason about events is an important aspect in understanding natural language (van Lambalgen and Hamm, 2006).

Take as example the following questions:

(i) Did Mary watch TV on the 13th of January 2023?

(ii) Who prepared Risotto on Christmas?

(iii) When was the last time that Peter prepared a Risotto?

Such and other questions require to reason with respect to a chain or sequence of events that have happened in the past. The last question, for instance, requires retrieving all the times that Peter prepared Risotto vs. all the other times he cooked something different and finding the instance that is closest to the speaking time.

Motivated by the recent success of Large Language Models (LLMs) on reasoning tasks in general (Wei et al., 2022), we ask the question whether Large Language Models are capable of reasoning on the basis of a sequence of events to answer temporal questions. Towards this goal we compile a new English synthetic benchmark dataset comprising of temporal questions over sequences of events, and experimentally validate the ability of different LLMs to answer such questions. Our focus lies on two crucial dimensions. On the one hand, we quantify the impact of varying the degree of explicitness of a temporal reference. As an example, the temporal reference in question (i) is maximally specific, referring to a concrete day. The reference to Christmas in (ii) is less explicit, as knowledge about Christmas is needed to infer a specific day. The expression *'last time that Peter prepared risotto'* in (iii) requires temporal reasoning to infer a date, being thus a very implicit reference. On the other hand, our goal is to analyze the ability of large models to cope with longer event sequences, so that we analyze the performance on the task by systematically varying the length of the sequence to be considered. Considering that LLMs currently lack explicit memory and explicit temporal reasoning abilities, we formulate two hypotheses:

- H1: The performance of LLMs will degrade with increasing level of implicitness of temporal references.
- H2: The performance of LLMs will degrade the longer the event sequences to be considered are.

Starting from these two hypotheses, we construct our synthetic benchmark dataset and define our experiments such that one can measure the performance of LLMs along these two dimensions: event sequence length and degree of explicitness of the temporal reference. Our benchmark consists of 2200 questions in the domain of activities carried out at home. We will make the dataset publicly available via a GitHub Link so that it can be used by the community upon publication of the paper.

Our contributions are the following:

- We propose a new task, that is, temporal reasoning over event sequences. We propose to investigate the ability of systems to reason about such sequences in a QA setting in which the sequence of events is encoded by a LLM which is then asked to answer a specific temporal question.
- We present a synthetically generated benchmark comprising 2200 questions over common household events as a domain.
- We systematically test different prompt engineering methods to find an effective prompt for the task.
- We compare four LLMs (`Gemma-7b-it`, `Llama3-8B-Instruct`, `Llama3-70B-Instruct`, `GPT-4-0125`) on the task, reporting results for different event sequence lengths and levels of explicitness.

Overall, our findings corroborate our two hypotheses, e.g. that LLMs have more difficulties with a higher volume of events in the event sequence and that they struggle with questions involving more implicit temporal references. Our results show that performance indeed deteriorates with increasing size of event sequences for all benchmarked LLMs. Further, the performance on questions involving implicit temporal references is roughly a third worse compared to the performance on questions with explicit references. In addition, we observe that LLM size clearly correlates with performance on the task.

2 Related Work

Events can ontologically be regarded as *things that happen in time* in which participants play different roles, e.g. agent, patient, beneficiary, etc. In his early foundational work, Davidson (2001) has argued that action sentences can be formalized as referring to an event as an ontologically reified object to which further roles can be attached. Further work has attempted to distinguish different types of events and unveiling their internal structure. Vendler (1957) introduced the important distinctions between subtypes of events, including activities, achievements and accomplishments. Moens and Steedman (1988) have proposed that an event consists of a nucleus with an associated preparatory phase, a culmination and a consequent phase. The ability to reason about events when interpreting natural language is key, and there has been work defining how events can be formalized and treated 'properly' (van Lambalgen and Hamm, 2006). Further, specific markup languages have been proposed to allow for annotating temporal expressions in corpora and documents, with TimeML (Pustejovsky, 2005) being the most prominent representative. Other markup Languages are TIE-ML (Cavar et al., 2021) and ISO-TimeML (Pustejovsky et al., 2010). ISO-TimeML is a revised and interoperable version of TimeML and the ISO/TC37 standard for time and event markup and annotation.

2.1 Categories of Temporal Questions

Temporal questions are often categorized depending on the explicitness by which temporal expressions contained therein refer to a particular date. In our discussion we follow previous categorisations as proposed by (Huang, 2018; Alonso et al., 2007; Strötgen, 2015).

We distinguish on the one hand *temporally explicit* questions, in which the temporal expression unambiguously and explicitly refers to a certain point in time in a way that is context-independent, e.g. *'25th of December 2023'*. Other questions refer to a time point in a more *implicit* way, thus requiring additional knowledge to resolve the temporal expression, such as for *'Christmas 2023'*, *'yesterday'* and *'Tom's Birthday'*. The category of temporally implicit questions can be further subdivided into four subcategories: i) questions requiring common sense knowledge, ii) referential relative to speech time, iii) referential relative to an arbitrary time point, and iv) referring to personal knowledge. Questions requiring common sense knowledge involve expressions such as *'Christmas 2023'* that can be resolved to a particular date using common sense knowledge, e.g. that Christmas is on the 25th of December of each year. Temporal questions that are *referential relative to speech time* require interpreting a certain temporal expression relative to the point in time in which the question is spoken or written. Such questions contain temporal expressions such as *'today'*, *'yesterday'*, *'two days ago'*, etc. Temporal questions that are *referential relative to an arbitrary time point* involve expressions such as *'two days before Christmas 2022'* that need to be resolved in relation to some other event. Finally, there are temporal questions requiring personal or private knowledge such as in the question: *'Who watched TV on Tom's birthday?'*. In our benchmark, we consider two types of questions, *explicit* and *implicit* questions of subtype *referential relative to speech time*.

2.2 Benchmarks for Temporal Questions

Several benchmarks for temporal question answering (QA) have been proposed so far. *TempQuestions* (Jia et al., 2018) and *TimeQuestions* (Jia et al., 2021) are two related datasets comprising 12k and 16k questions, respectively. The questions pertain to historical events such as Obama's presidency and Brad Pitt's 2001 award. Event knowledge is stored in a Knowledge Graph (KG), so that answers are retrieved by mapping questions to a KG query.

The *Test of Time (ToT)* Benchmark (Fatemi et al., 2024) is designed to evaluate two fundamental aspects of temporal cognition independently: ToT Semantic assesses comprehension of temporal semantics and logic without dependence on prior knowledge, while ToT Arithmetic evaluates the ability to perform calculations involving time points and durations. Two QA sets (*Date Understanding* and *Temporal Sequences*) in the *'Beyond the Imitation Game Benchmark'* (Srivastava and et al., 2023) rely on textually encoded contexts on the basis of which to answer questions. However, these benchmarks are not suited for our research questions. *Date Understanding*, *Temporal Sequences* and *ToT* do not allow to benchmark models with respect to their ability to consider longer sequences of events with different participants as we do.

2.3 Large Language Models for Reasoning

Large Language Models have been successfully applied to multiple reasoning tasks (see Huang and Chang, 2023 for a recent overview). Examples of these tasks include symbolic manipulation, such as concatenating the last letter of words (Last Letter Concatenation[1]), mathematical reasoning, and arithmetic tasks like algebraic problems (AQuA, Ling et al., 2017), Math World Problems (MWP), (SVAMP Patel et al., 2021), or Graduate School Math Word Problems (GSM8K, Cobbe et al., 2021). In general, the performance on reasoning tasks seems to increase with the size of the model (Wei et al., 2022, Saparov and He, 2023). It has further been shown that Chain-of-Thought prompting enhances LLMs performance (Suzgun et al., 2022). So far, however, LLMs have not been evaluated on the task of resolving temporal references in the context of longer event sequences, a gap we close in this paper.

On the other hand, LLMs struggle with reasoning tasks that more closely resemble real-world situations, such as commonsense planning domains (Valmeekam et al., 2023, Joublin et al., 2023). Parmar et al., 2024 also demonstrate that LLMs often overlook contextual information when engaged in logical reasoning over natural Language text. According to Saparov and He, 2023, while LLMs are capable of handling reasoning tasks that involve single deductive steps, they encounter difficulties when dealing with tasks that require multiple deductive steps. Thus, it is an interesting research question to examine the ability of LLMs to resolve explicit and implicit temporal expression in settings where multiple events take place and several steps might be involved in answering a temporal question involving such a reference.

3 Methods

In this section, we describe the methodology for constructing the dataset consisting of event sequences of varying length (Section 3.1) with corresponding questions (Section 3.2). In addition, we describe the prompting strategies we use for the LLMs (Section 3.3).

3.1 Generation of synthetic event sequences

We generate event sequences automatically by randomly sampling from a set of action predicates, agents which can carry out the action, objects on which the action is carried out and the location of the event. For this, we consider events that might typically take place in a home environment. Events are described in terms of five variables (with their potential fillers in brackets): i) Action (Watch, Eat, Read, Dance, Store, Drink, Chat) ii) Object (Film, Risotto, Book, Salsa, Wine Bootle, Juice, Friend), iii) Agent (Mary, Tom, Ria), iv) Location (Living Room, Kitchen), and v) Timestamp. Timestamps are provided as a Unix timestamp ranging from 2023-01-01 to 2023-09-29.

For instance, our procedure would generate events such as the following:

- Action:watch, Object:film, Location:living room, Subject:mary, Timestamp:1695948843

[1] https://huggingface.co/datasets/ChilleD/LastLetterConcat

Question Category	**Temporal Expression**
Temporally Explicit	on yyyy-mm-dd in yyyy-mm in the year yyyy
Referential relative to speech time	today yesterday this year this month last month

Table 1: Temporal Expressions for the 2 categories of temporal questions. yyyy is the year with four digits, mm the month of the year with two digits, and dd the day of the month with two digits.

- Action:eat, Object:risotto, Location:kitchen, Subject:tom, Timestamp:1695852168
- ...

We randomly generate event sequences, with a length of 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. Given the many possibilities and timestamps in particular, the probability of generating the same event twice is negligible.

3.2 Question Generation

For each event sequence, we automatically generate a set of questions together with a ground truth answer that is computed on the basis of a symbolic representation of the event sequences. In order to generate questions, we rely on the question templates shown in Table 2. As an example, we would generate questions such as: *Who washed a mug in the kitchen today?*

For each event instance in a generated event sequence, we instantiate each of the 4 question templates in Table 2 with each of the temporal expression in Table 1, whereby the fourth pattern (‘*When was the last time...?* ´) is instantiated only for the category *referential relative to speech time* without a temporal expression. This yields 25 questions for each event instance $(8 * 3 + 1 = 25)$.

Given the event instance: *Action:wash, Object:mug, Location:kitchen, Subject:tom, Timestamp:1695852168*, we would generate 25 questions for all possible choices of temporal expressions, generating questions such as:

- Who washed a mug in the kitchen on 2023-08-16?
- When was the last time Tom washed a mug in the kitchen?
- Did Tom wash a mug in the kitchen yesterday?

Overall, we generate 100 questions for each length of event sequence and question category. This makes $100 * 2 * 11 = 2200$ questions in total.

> Today is the 2023-09-29 22:18. I will give you a list indicating events and when they have taken place (event sequence): {Action: watch, Object: film, Location: living room, Subject: Mary, Date: 2023-09-29 08:01}, {Action: eat, Object: risotto, Location: kitchen, Subject: Tom, Date: 2023-09-28 14:27}, {Action: read, Object: book, Location: living room, Subject: Ria, Date: 2023-06-11 12:44}, {Action: dance, Object: lively salsa, Location: kitchen, Subject: Mary, Date: 2023-08-11 10:57}, {Action: store, Object: wine bottle, Location: living room, Subject: Tom, Date: 2023-09-01 20:44}. Who watched a film in the living room on 2023-09-29? Answer with the the name of the subject or say ’nobody’.

Figure 1: Exemplary zero-shot prompt for an event sequence length of 5 events.

3.3 Prompting Strategies

As baseline prompting strategy, we rely on a zero-shot prompt, where we only define the expected answer of the LLM corresponding to the question templates from Table 2. The basic prompt is given in Figure 1. Hereby, we experimentally vary the granularity in which the temporal information is presented. We distinguish two granularities: *Date-Only* and *Date-Extended*. In the first case, *Date-Only*, the date and its corresponding hour and minute is provided. In the second case, *Date-Extended*, the date, corresponding weekday and calendar week are included, as in the following example

Date: 2023-08-11 10:57, Weekday: Friday, Calendar Week: 32

Beyond varying the date granularity, we vary the way in which the events and their dates are presented. In the *Json* condition (see example in Figure 1), the event is encoded in JSON format. In the *Language* condition, the event and its corresponding date granularity is transformed into a natural Language sentence. For *Date-Only*, this would look as follows:

On September 29, 2023 at 08:01, Mary watched a film in the living room.

Beyond relying on a zero-shot prompting approach as proposed above, we also experiment with an advanced prompting strategy relying on Chain of Thought (CoT). We distinguish two different strategies: *CoT Review*, and *CoT Step-by-Step* reasoning. In the *CoT Review* case, the model receives

Template	Return Type
Who <action><object><location><ref_date>?	String - Persons Name(s)
Did <subject><action><object><location><ref_date>?	Bool
How often did <subject><action><object>location><ref_date>?	Integer
lightgrayheightWhen was the last time <subject><action><object><location>?	Date

Table 2: Templates for the Questions of the QA Set

instructions on how to approach the task. For a "Who...?" question this would be like this:

Review each event out of the event history sequentially. If the action, object, location and date of an event match the information in the question, record the subject of that event. At the end return the subjects of all matching events.

In the *CoT Step-by-Step* reasoning condition, we extend the *CoT Review* prompt by the sentence 'Let's think step by step.'

4 Experiments

4.1 Experimental Plan

We consider state-of-the-art LLMs, selecting the following models: `Gemma-7b-it` (Team et al., 2024), `Llama3-8B-Instruct`, `Llama3-70B-Instruct` (Lla, 2024) and `GPT-4-0125` (OpenAI, 2023). We proceed as follows: we first carry out experiments with all possible different prompting strategies and event sequence lengths of 5 and 50 for `GPT-4`. On the basis of this initial experiment, we identify the top four best performing prompting strategies and test these for all Language models and event sequence lengths of between 5 and 50 events to determine the best prompting strategy for all models. We then present results showing how performance differs depending on question type, question category and event sequence length for the top performing prompting strategy.

4.2 Experimental Settings

The individual experiments are conducted on GPU (`Llama3`, `Gemma`) and over API (`GPT-4`). We used the `Llama3`[2] in the 8B and 70B instruction variant and `Gemma`[3] in the 7B instruction variant without further fine-tuning from HuggingFace. We evaluate the performance of the models using accuracy. For all models we use a temperature of 0 or corresponding settings so that the responses are deterministic.

Review each event out of the event history sequentially. If the action, object, location and date of an event match the information in the question, record the subject of that event. At the end return the subjects of all matched events. Today's date is September 29, 2023, and the time is 22:18. I have a list of events (event sequence) that have occurred in the past, including who did what, where and when: On September 29, 2023 at 08:01, Mary watched a film in the living room. On September 28, 2023 at 14:27, Tom ate a risotto in the kitchen. On June 11, 2023 at 12:44, Ria read a book in the living room. On August 11, 2023 at 10:57, Mary danced a lively salsa in the kitchen. On September 01, 2023 at 20:44, Tom stored a wine bottle in the living room. Now, I want to know: Who watched a film in the living room on September 29, 2023?

Figure 2: Exemplary final prompt for an event sequence length of 5 events.

4.3 Results

We report our results by analysing first the impact of all possible prompting strategies for `GPT-4` in Section 4.3.1. In the following Section 4.3.2 we further present the results of all models for the four best performing prompting strategies identified in Section 4.3.1. Then we present the difference in performance of the benchmarked LLMs depending on question type in Section 4.3.3. Finally, we investigate the relation between length of the event sequence and performance in Section 4.3.4.

4.3.1 Prompting Strategies

Given the variability of our prompting strategies (3 Prompt types: *zero-shot*, *CoT Review*, *CoT Step-by-Step*; 2 date representations: *Date-Only*, *Date-extended*; 2 event presentations: *Json*, *Language*), we have 12 possible prompt types that we evaluate using *GPT-4* and event sequence lengths of 5 and 50 events. The accuracy scores for the different configurations are given in Table 3. We observe that for all prompting strategies, performance is higher for 5 compared to 50 events. Generally, the impact of CoT seems to be positive as results are generally

[2] https://huggingface.co/collections/meta-llama/meta-llama-3-66214712577ca38149ebb2b6

[3] https://huggingface.co/google/gemma-7b-it

better compared to the baseline Zero-Shot prompt. Extended date encoding (*Date-extended*) does not seem to have any positive impact beyond the simple date encoding (*Date-Only*). The top performing prompting strategies rely on CoT prompting and *Date-only* date in combination with either of the two event presentation approaches.

4.3.2 Model Impact

Table 4 shows the accuracy for the 4 best prompting strategies for all models with respect to event sequences of 5 and 50 events. We see that the models with the most parameters (`GPT-4`, `Llama3-70B`) have the top performance with accuracies between 83%-84% (`GPT-4`) and 84%-90% (`Llama3-70B`) across the different configurations. `Llama3-70B` seems thus to be slightly ahead of `GPT-4`. The other models (`Gemma`, `Llama3-8B`) have lower results of between 63%-86% (`Gemma`) and 68%-74% (`Llama3-8B`).

For our further experiments, we select the configuration with highest average performance across all models: *CoT Review*, *Date-Only*, *Language*.

4.3.3 Type of Questions

Table 5 shows the results for the two question categories *Temporally Explicit* and *Referential relative to speech time* as well as the different question templates from Table 2.

Impact of degree of explicitness: The performance across models for *temporally explicit* questions ranges between 75% (`Llama3-8B`) and 92% (`Llama3-70B`). We observe a significant performance drop when considering questions with expressions that need to be resolved with respect to speech time. Here results range between 34% (`Gemma`) and 74% (`Llama3-70B`). The performance is reduced by around 17%-50% when shifting from explicit to implicit temporal references.

Results by template type: Regarding the performance by template type, we see that the investigated models have the best performance on questions following the template *Did ...?* with accuracies ranging between 78% (`Gemma`) and 92% (`GPT-4`). The question template with the worst performance is the *When was the last time ...?* template, yielding results of 34% for `Gemma`, 53% for `Llama3-8B` and 66% for `Llama3-70B`. `GPT-4` has the lowest accuracy for *Who ...?* with 59%.

4.3.4 Sequence Length

The results for the two question categories *Referential relative to speech time* and *Temporally Explicit* for different event sequence lengths (5, 10, 20, 30, 40, 50 , 60, 70, 80, 90, 100) are shown in Figures 3 and 4, respectively. From these plots we see that performance of the models decreases substantially with increasing sequence length. For the **Temporally explicit** question category, the decrease from 5 to 100 events ranges between 18% (`Gemma`) and 10% (`Llama3-70B`). Overall. the performance decreases by between 1,0% (`Llama3-70B`) and 3,6% (`Llama3-8B`) at each step.

For the **Referential relative to speech time**, the performance decreases are even more pronounced, ranging between 39% (`GPT-4` and `Llama3-8B`) and 29% (`Llama3-70B`). The performance decreases stepwise by between 2,9% (`Llama3-70B`) and 4,9% (`Llama3-8B`) from 5 to 100 events considered.

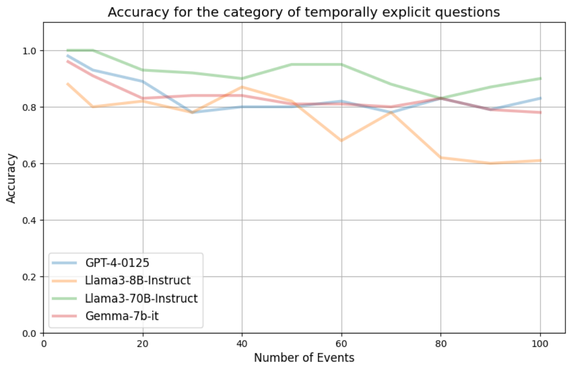

Figure 3: Accuracy for the *Temporally Explicit* question category depending on sequence length.

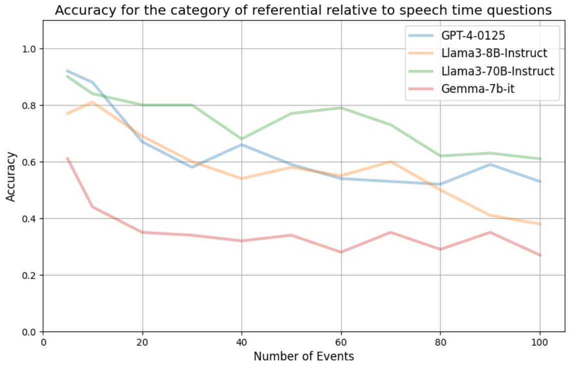

Figure 4: Accuracy for the *Referential relative to speech time* question category depending on sequence length

5 Discussion

Our results clearly corroborate our two hypotheses. Regarding H1, our results show that the average

Prompting Strategy	Date Information	Event Presentation	Events		Average
			5	50	
Zero-Shot	Date-Only	Json	.97	.67	.82
Zero-Shot	Date-Only	Language	.96	.67	.82
Zero-Shot	Date-Extended	Json	.97	.64	.81
Zero-Shot	Date-Extended	Language	.96	.68	.82
CoT Review	**Date-Only**	**Json**	.97	.71	**.84**
CoT Review	**Date-Only**	**Language**	.94	.71	**.83**
CoT Review	Date-Extended	Json	.95	.68	.82
CoT Review	Date-Extended	Language	.93	.71	.82
CoT Step-by-Step	**Date-Only**	**Json**	.94	.71	**.83**
CoT Step-by-Step	**Date-Only**	**Language**	.95	.71	**.83**
CoT Step-by-Step	Date-Extended	Json	.94	.66	.80
CoT Step-by-Step	Date-Extended	Language	.94	.70	.82

Table 3: Accuracy of all possible prompts for GPT-4-0.125 averaged for the two question categories *Temporally Explicit* and *Referential relative to speech time* over event sequence lengths of 5 and 50 events. The last column is the average of the accuracy for 5 and 50 Events. The 4 highest results are marked in bold.

Prompting Strategy	Date Information	Event Presentation	Gemma -7b-it	Llama3 -8B-Instr.	Llama3 -70B-Instr.	GPT-4 -0125	Average
CoT Review	Date-Only	Json	**.68**	.68	.86	**.84**	.76
CoT Review	**Date-Only**	**Language**	**.68**	**.74**	.88	.83	**.78**
CoT Step-by-Step	Date-Only	Json	.63	.69	.84	.83	.75
CoT Step-by-Step	Date-Only	Language	.65	.72	**.90**	.83	.77

Table 4: Accuracy of the 4 best performing prompt configurations for GPT-4-0.125 on all evaluated LLMs averaged over event sequence lengths of 5 and 50 events for both question categories. The highest result for each model and the highest average result is marked in bold.

performance of all models is 26% lower for questions involving implicit temporal references compared to questions with explicit dates. This shows that it is a challenge for LLMs to interpret temporal expressions beyond explicit dates. Given that in the case of *temporally explicit* expressions the dates in the questions match exactly a date in the event history, there might be sufficient cues for the LLMs to perform well on this.

Regarding H2, our results clearly convey a trend, i.e. that performance deteriorates with increasing length of event history. This is understandable, as LLMs do not have an explicit memory and can not 'store' events for later random access. The performance decrease varies from model to model, with the most pronounced drop of 39% being observed for GPT-4 and Llama3-8B between the sequences of 5 and 100 events and the question category *Referential relative to speech time*.

Considering the different prompting strategies, even if the performances only vary by up to 6%, we can clearly see that using *CoT* always leads to better performances. This has also been shown in other studies ((Wei et al., 2023), (Suzgun et al., 2022)). Representing the date information in the *Date-Only* format is always better than *Date-Extended*. This may be because we do not ask questions about information in the *Date-Extended* format, such as questions about the day of the week. Then the extended format would just make the final prompt longer. Presenting events in natural language outperforms the presentation by way of JSON. This is likely due to the fact that models have been mainly trained with language as input and might have seen JSON structures more rarely.

Considering the different question templates, it is interesting to observe that the best performance across models is reached for the question following the template *Did...?*. The reason for this high performance is likely due to the fact that a binary yes/no answer is required and chances of getting it right are a priori high.

The performance on the other question templates (*How often did...?* and *Who...?*) are around 20%

		`Gemma-7b-it`	`Llama3-8B-Instruct`	`Llama3-70B-Instruct`	`GPT-4-0125`	Average
Question Categories	*Temporally Explicit*	**.84**	**.75**	**.92**	**.84**	.84
	Referential relative to speech time	.34	.58	.74	.64	.58
Question Templates	Who ...?	.58	.58	.83	.59	.65
	Did ...?	**.78**	**.80**	**.90**	**.92**	.85
	How often did ...?	.44	.63	.78	.70	.64
	When was the last time ...?	.34	.53	.66	.75	.57

Table 5: Accuracy for the different question template types averaged over all evaluated event sequence lengths. The right column is the average of all models. The highest results of each model for each question category and question template are marked in bold.

worse than *Did...?*. Answering questions of type *Who...?* requires extracting a list of agents that participated in an event instance of the given type in the period selected. This seems to be a challenging task for all models. The questions of type *How often did...?* require deeper reasoning ability to identify all events that meet the criteria and counting them. The benchmarked models do not seem to be capable of such an advanced reasoning. Performance on questions *When was the last time...?* are the worst for all models except `GPT-4`.

Our results clearly show that size matters in that the two models with the largest parameters also perform best on the task. Interestingly `Llama3-70B` performs slightly better than `GPT-4` in spite of having less parameters than `GPT-4`, that is 70 Bn. vs. 1760 Bn. This could be an indication that model size is only important up to a certain extent. Further research is needed to find out which factors make `Llama3-70B` so successful.

6 Conclusion & Future Work

We have analysed the ability of Large Language Models to reason about event sequences, proposing a benchmark that relies on a question answering proxy task. Our focus has been on analyzing the performance of four state-of-the-art language models on the task depending on the size of the event sequences and the explicitness of temporal references included in the questions. The two hypotheses have been validated on the basis of our results. While LLMs can answer questions containing explicit temporal expression with high accuracy, they struggle when the temporal expressions become more implicit. Further, the performance deteriorates significantly with the size and length of event sequences to consider.

Future work could investigate how such models can be extended with some explicit memory to store events and access them explicitly. A further line of work might explore how such models can be endowed with explicit temporal reasoning abilities by extending them with logical temporal theories, e.g. by function calls such as supported by some recent LLMs. One relevant work in this context is by (Xiong et al., 2024), where they generated a temporal graph from a prompt containing historical events and a corresponding question to incorporate explicit memory. They then applied Chain-of-Thought reasoning on this temporal graph to improve the temporal reasoning capabilities of LLMs.

7 Optional Supplementary Materials

7.1 Limitations

Our benchmark consists of synthetically generated data, which raises the question of how representative this benchmark is for real events taking place in home environments and potential questions users might ask. As an example our generated events are independent from each other which is rather rare in real life, as a meal, for example, has to be prepared before it is eaten. Regarding the degree of explicitness we have only contrasted the two categories (*Temporally Explicit*, *Referential relative to speech time*). However, it would be interesting to consider further distinctions along the categories of questions described in Related Work. We acknowledge that our experimental settings might be perceived as "unfair" because LLMs do not have explicit memory. This means that one might expect them to have difficulty on tasks requiring storage. Nevertheless, there are increasingly investigations regarding the ability of LLMs to perform reasoning, and in this context it is relevant to test the abilities also on tasks that might require some memory to understand their limitations.

7.2 Ethics

We do not think that the specific methods or benchmark presented in this work raises ethical issues. However, there are some ethical issues worth considering when thinking of the productive application of models such as benchmarked in this paper, e.g. in smart homes or other environments in which humans perform tasks. Building up a memory of events might require rich sensors, cameras etc. The storage of such information in multimodal memories would clearly raise data protection issues and ethical questions related to profiling and privacy protection.

References

2024. Introducing meta llama 3: The most capable openly available llm to date.

Omar Alonso, Michael Gertz, and Ricardo Baeza-Yates. 2007. On the value of temporal information in information retrieval. *SIGIR Forum*, 41:35–41.

Damir Cavar, Billy Dickson, Ali Aljubailan, and Soyoung Kim. 2021. Temporal information and event markup language: Tie-ml markup process and schema version 1.0. *Preprint*, arXiv:2109.13892.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Donald Davidson. 2001. The Logical Form of Action Sentences. In *Essays on Actions and Events*. Oxford University Press.

Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. 2024. Test of time: A benchmark for evaluating llms on temporal reasoning. *Preprint*, arXiv:2406.09170.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. *Preprint*, arXiv:2212.10403.

Ruihong Huang. 2018. Domain-sensitive temporal tagging by jannik strötgen, Michael gertz. *Computational Linguistics*, 44(2):375–377.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. Tempquestions: A benchmark for temporal question answering. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1057–1062, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Complex temporal question answering on knowledge graphs. *CoRR*, abs/2109.08935.

Frank Joublin, Antonello Ceravola, Pavel Smirnov, Felix Ocker, Joerg Deigmoeller, Anna Belardinelli, Chao Wang, Stephan Hasler, Daniel Tanneberg, and Michael Gienger. 2023. Copal: Corrective planning of robot actions with large language models. *Preprint*, arXiv:2310.07263.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Marc Moens and Mark Steedman. 1988. Temporal ontology and temporal reference. In *International Conference on Computational Logic*.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. Logicbench: Towards systematic evaluation of logical reasoning ability of large language models. *Preprint*, arXiv:2404.15522.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

J. Pustejovsky. 2005. Time and the semantic Web. In *12th International Symposium on Temporal Representation and Reasoning (TIME'05)*, pages 5–8. ISSN: 2332-6468.

James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. ISO-TimeML: An international standard for semantic annotation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *Preprint*, arXiv:2210.01240.

Aarohi Srivastava and et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Jannik Strötgen. 2015. Domain-sensitive temporal tagging for event-centric information retrieval.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *Preprint*, arXiv:2210.09261.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Preprint*, arXiv:2206.10498.

Michiel van Lambalgen and Fritz Hamm. 2006. The proper treatment of events. *Bulletin of Symbolic Logic*, 12(1):139–141.

Zeno Vendler. 1957. Verbs and times. *The Philosophical Review*, 66(2):143–160.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Preprint*, arXiv:2206.07682.

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. *Preprint*, arXiv:2201.11903.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning. *Preprint*, arXiv:2401.06853.

A Example Appendix

This is an appendix.