Large Language Models Can Learn Temporal Reasoning

Siheng Xiong[1]*, **Ali Payani**[2]*, **Ramana Kompella**[2], **Faramarz Fekri**[1]
[1]Georgia Institute of Technology [2]Cisco Research
sxiong45@gatech.edu {apayani, rkompell}@cisco.com
faramarz.fekri@ece.gatech.edu

Abstract

While large language models (LLMs) have demonstrated remarkable reasoning capabilities, they are not without their flaws and inaccuracies. Recent studies have introduced various methods to mitigate these limitations. Temporal reasoning (TR), in particular, presents a significant challenge for LLMs due to its reliance on diverse temporal concepts and intricate temporal logic. In this paper, we propose TG-LLM, a novel framework towards language-based TR. Instead of reasoning over the original context, we adopt a latent representation, temporal graph (TG) that enhances the learning of TR. A synthetic dataset (TGQA), which is fully controllable and requires minimal supervision, is constructed for fine-tuning LLMs on this text-to-TG translation task. We confirmed in experiments that the capability of TG translation learned on our dataset can be transferred to other TR tasks and benchmarks. On top of that, we teach LLM to perform deliberate reasoning over the TGs via Chain-of-Thought (CoT) bootstrapping and graph data augmentation. We observed that those strategies, which maintain a balance between usefulness and diversity, bring more reliable CoTs and final results than the vanilla CoT distillation.[1]

1 Introduction

As one of the fundamental abilities, temporal reasoning (TR) plays an important role in human perception. It is not just about understanding basic concepts such as ordering or duration; it extends to more intricate aspects, e.g., task planning or causal relation discovery. Recently, large language models (LLMs) (Ouyang et al., 2022; Achiam et al., 2023; Touvron et al., 2023a) have emerged with some reasoning capabilities (Huang and Chang, 2022). However, there is observation that they still can not perform TR sufficiently well (Wang and Zhao, 2023; Chu et al., 2023; Qiu et al., 2023), preventing their applications from solving complex real-world problems. In particular, TR requires a combination of various skills including mathematical and logical reasoning as well as commonsense knowledge (Zhou et al., 2019; Vashishtha et al., 2020; Qin et al., 2021).

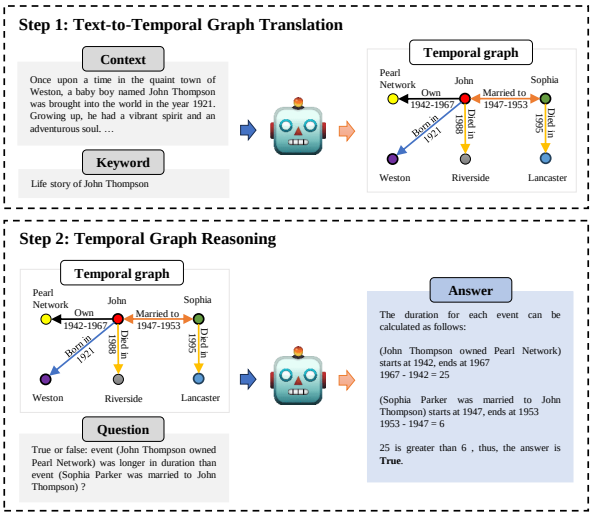


Figure 1: Our framework (TG-LLM) performs temporal reasoning in two steps: 1) Text-to-Temporal Graph translation: generate (relevant) temporal graph given the context and keyword (extracted from questions); 2) Temporal Graph Reasoning: perform Chain-of-Thought reasoning over the temporal graph.

Recent works mainly adopt general approaches to investigate and improve the TR capability of LLMs. For example, (Wang and Zhao, 2023; Chu et al., 2023; Qiu et al., 2023) benchmark the leading LLMs on different TR tasks with standard Input/Output prompt, few-shot in-context learning (ICL) and Chain-of-Thought (CoT) reasoning (Wei et al., 2022). Similarly, (Wei et al., 2023) designs several specific types of prompts as prompt tuning. (Li et al., 2023; Yuan et al., 2024) introduce predefined Python programs/rule-based templates to perform supervised fine-tuning (SFT). In addition,

*Equal contribution.
[1]Code and data are available at https://github.com/xiongsiheng/TG-LLM.

(Tan et al., 2023a,b) adopt some extra strategies, which include specific pre-training, instruction tuning and reinforcement learning.

Despite the effectiveness of such methods, they either ignore or not explicitly involve the intrinsic nature of TR. Humans perform complex TR on a timeline of events which are aligned with the entities and relations. These temporal concepts (e.g., ordering, duration, frequency, typical time) are then rigorously defined based on the timeline information. In other words, the aligned timeline (more generally, the temporal graph, TG) serves as a latent representation to help humans understand the patterns in TR. However, due to the lack of ground truth, the high-quality TG translation is a challenging task for most TR benchmarks. To solve this problem, we propose a synthetic dataset (TGQA), which is fully controllable and requires minimal supervision. We demonstrate the capability of TG translation learned on our dataset can be transferred to other TR tasks and benchmarks.

Given a reliable TG, the key challenges of teaching TR to LLMs include: (1) How can one introduce the necessary arithmetic and commonsense knowledge involved in TR? Prior work (Lewis et al., 2020) shows that explicitly introducing knowledge into context enhances the performance of LLMs. In this paper, we first identify all the valid time expressions, and then generate related knowledge (e.g., temporal relation and time gap between the timestamps, and the relative order of the gaps). (2) How can one teach LLM to perform deliberate reasoning? Generally, there exist two roadmaps: (i) translating natural language into logical statements, and using external symbolic engine for reasoning (Pan et al., 2023); (ii) using LLMs directly as the reasoning engine (Zhu et al., 2023). For (i), the difficulty lies in accurate translation (Yang et al., 2023c) and the limited expressive power of formal logic. For (ii), there is no guarantee for the correctness of generated intermediate steps especially with insufficient training data (Yang et al., 2023b). In this paper, we adopt (ii) with the proposed bootstrapping method to generate reliable intermediate steps for supervised fine-tuning. We further improve the model performance with graph data augmentation, which mitigates the data deficiency in TR tasks.

To be specific, our contributions are summarized as follows:

- We propose a new paradigm, TG-LLM, for language-based TR. In this framework, we first translate the context into a latent representation (temporal graph), and then perform reasoning on it. Extensive experiments prove that our novel approach results in superior performance compared to the baselines.
- We design two approaches including Chain-of-Thought bootstrapping and graph data augmentation to teach LLM to generate consistent and faithful CoTs, which brings better performance than the vanilla CoT distillation.
- We present a pipeline to create a synthetic dataset (TGQA) for question answering that requires TR. It is fully controllable and requires minimal supervision for text-temporal graph alignment. We show in experiments that fine-tuning on our dataset benefits LLM on other TR tasks and benchmarks.

Temporal Graph [sub; rel; obj; start/end; time]:
[1] (John Thompson was born in Weston) starts at 1921; [2] (John Thompson owned Pearl Network) starts at 1942; [3] (Sophia Parker was married to John Thompson) starts at 1947; [4] (John Thompson was married to Sophia Parker) starts at 1947; [5] (Sophia Parker was married to John Thompson) ends at 1953; [6] (John Thompson was married to Sophia Parker) ends at 1953; [7] (John Thompson owned Pearl Network) ends at 1967; [8] (John Thompson died in Riverside) starts at 1988; [9] (Sophia Parker died in Lancaster) starts at 1995.
Graph-based Story (from GPT-3.5):
Once upon a time in the quaint town of Weston, a baby boy named John Thompson was brought into the world in the year 1921. Growing up, he had a vibrant spirit and an adventurous soul. · · ·
Graph-based QAs (from rule-based Python script):
Q1: Which event started first, (John Thompson owned Pearl Network) or (John Thompson was married to Sophia Parker)? **A1:** (John Thompson owned Pearl Network). **Q2:** True or false: event (John Thompson owned Pearl Network) was longer in duration than event (Sophia Parker was married to John Thompson)? **A2:** True. · · ·

Table 1: Each sample from TGQA dataset is in the form of (temporal graph, story, questions, answers).

2 Dataset Construction

In this section, we present the construction pipeline for TGQA dataset that is fully controllable and re-

Reasoning Type	Question	Answer
Sequencing	Given the following <N> events: <Event_A>, <Event_B>, <Event_C>, <Event_D>, ···, which event is the first/second/third/fourth/··· one in the chronological order?	<Event_A>/<Event_B>/ <Event_C>/<Event_D>/ ···
Duration	How long did the event <Event_A> last?	<Duration_of_Event_A>
Temporal Relation	How much time passed between the start of <Event_A> and the start of <Event_B>?	<Gap_between_Event_A _and_Event_B_startTime>
	What happened right before/after <Event_A> started?	<Event_B> right before/ after <Event_A>
Facts Extraction	When did the <Event_A> occur?	<Event_A_startTime>
Simultaneity	True or false: <Event_A> and <Event_B> happened at the same year?	True / False
	True or false: <Event_A> was still happening when <Event_B> started?	True / False
Comparative Analysis	True or false: <Event_A> was longer in duration than <Event_B>?	True / False

Table 2: Reasoning types with the corresponding questions and answers in TGQA.

quires minimal supervision for text-temporal graph alignment. Compared with existing datasets (Chen et al., 2021; Tan et al., 2023a), we have the ground-truth timelines and more diverse categories and types of TR questions (Table 2). More importantly, the pipeline can be used for various scenarios and tasks. We first split the large temporal knowledge graph, YAGO11k (Dasgupta et al., 2018), into subgraphs with a restriction on the number of events, and anonymize the entities to avoid data leakage. Then each story is translated from the subgraph by GPT-3.5 (Ouyang et al., 2022). By using some rule-based templates, we obtain reliable question and answer (QA) pairs from the graph. Finally, to reduce the noise introduced from the misalignment between the subgraph and generated story, we propose a semi-automatic verification method.

Step 1: Graph Splitting & Anonymization. Existing temporal knowledge graphs (Leetaru and Schrodt, 2013; Dasgupta et al., 2018; García-Durán et al., 2018) usually have a large size. To facilitate the learning process, we split YAGO11k into subgraphs for story generation. Specifically, given a certain entity, we find its neighbors within three hops, and extract all the events happening between them. Since we hope LLMs to do reasoning instead of memorization, it is ensured that no overlapping exists between the events in training, validation and test sets. Additionally, we notice the data leakage problem of LLMs, i.e., prior knowledge of the test data has been implicitly obtained from the extensive pre-training. Thus, an anonymization strategy, i.e., changing entity names into random ones of the same type, is adopted. For each relation, we generate a global mapping of entity names with GPT-3.5. To avoid confusion, we adopt obscure names that do not exist in YAGO11k.

Step 2: Graph-based Open QA Creation. In TGQA, each sample is in the form of (temporal graph, story, questions, answers) (Table 1). Based on the given subgraph, we generate a story and multiple QAs. We first ask GPT-3.5 to write a story based on the subgraph with the requirement to include all the events. It is observed that GPT-3.5 tends to ignore the end time of some events in the created story. To solve this problem, we separate the start and end time of the same event in the prompt. On the other hand, to obtain a comprehensive benchmark, we consider all types of temporal reasoning, which include sequencing, duration, frequency, simultaneity, temporal relation, comparative analysis and facts extraction. Given these categories, we design multiple question types (Table 2) with a rule-based Python script to generate the corresponding Qs and As.

Step 3: Quality Control. In TGQA, noise might be introduced from the misalignment between the given subgraph and LLM-generated story. To address this problem, we propose a semi-automatic verification method. Fully manual inspection is expensive, but by first utilizing LLM we can narrow down potential errors, and only manually inspect the set of unanswered questions by LLM. Specifically, given a generated story, GPT-3.5 is queried on the time of each event in the graph. If it cannot give the correct answer, we consider the event as possibly missing in the story, which requires further manual verification. We proved the effectiveness of our semi-automatic pipeline with fully manual verification of the test stories. Note that su-

pervision is only required for story-TG alignment verification, since all the QAs are generated from rules. We show dataset statistics in Appendix A, and all the prompts involved in Appendix C.

3 TG-LLM

Motivated by human perception, we propose a new paradigm called TG-LLM. We first translate the text into a temporal graph (TG), and then guide the LLM to perform deliberate reasoning on it.

3.1 Text-to-TG Translation

Although LLM (with ICL) might have such capability, we observed a misalignment between the generated TG and pre-defined QAs, i.e, LLM making mistakes or focusing on irrelevant events. Since TG serves as the foundation of the following deliberate reasoning process, we provide a pipeline for high-quality TG data generation to fine-tune the LLM.

Ground-truth TG Generation. For some datasets such as TGQA, a verified TG (corresponding to the story) is provided. We can directly fine-tune the LLM on this task with the ground truth. However, for most real-world applications, the biggest challenge is the lack of ground truth TG. We provide a pipeline for high-quality TG data generation. We first extract all the entities and relations from the QAs to help LLM focus on specific events. We then identify all the valid time expressions in the story. We provide these information to LLM for better alignment in TG construction, and verify the generated TG in a semi-automatic way.

(1) Entity & Relation Extraction: To obtain the entity and relation in pre-defined QAs, we consider two strategies: parsing with rules or using GPT-3.5. Rule-based parsing, if applicable, is more efficient and reliable, which is prioritized in our experiments.

(2) Temporal Info Identification: To identify all the valid time expressions, we first use GPT-3.5 to extract from the story, and then adopt a rule-based Python script for invalid output filtering and normalization. Besides bringing better alignment, the pre-processing of time expressions facilitates the introduction of external knowledge (their temporal relations and time gaps) into TR. We proved in experiments that explicitly introducing this information further improves model performance.

(3) TG Construction & Verification: We generate the TG using GPT-3.5 with ICL. Specifically, we provide several in-context demonstrations which include a story and extracted entities, relations and time expressions as input and the corresponding TG as output (Table 15). For those stories without explicit time expressions, we skip Step (2), and ask GPT-3.5 to provide the temporal order of events. Similar to TGQA, we verify the generated TG by first asking GPT-3.5 the pre-defined QAs, and then manually checking the alignment if GPT-3.5 fails.

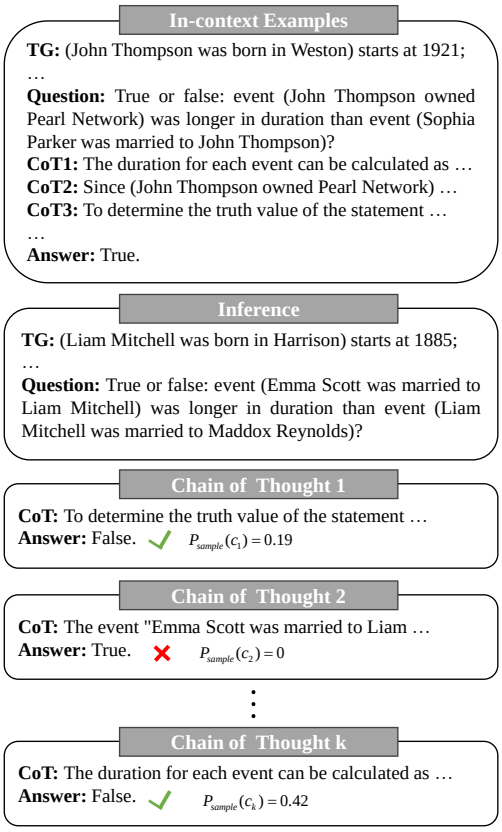


Figure 2: In Chain-of-Thought (CoT) bootstrapping, we only accept CoTs that lead to correct final answers and sample them according to their contrastive learning scores to balance usefulness and diversity.

Supervised Fine-tuning. We first conducted experiments to decide the best in-context format of TG. It is found out that providing a chronological list of events helps LLM perform better TR. In addition, it offers advantages to separate the start and end time of the same event. With these steps, the TG in context is transferred into a timeline with the alignment between entities, relations and times. We perform supervised fine-tuning (SFT) using Llama-2 model (Touvron et al., 2023b) with Low-Rank Adaptation (LoRA) (Hu et al., 2021). The input and output of the model are the story and aligned timeline, respectively. In experiments, we observe benefits from the SFT on our dataset to

other temporal reasoning tasks.

3.2 Temporal Graph Reasoning

Given the generated TGs, we teach LLM deliberate reasoning with SFT enhanced by CoT bootstrapping and graph data augmentation.

3.2.1 Bootstrapping Chain of Thoughts

It is observed that SFT on reliable CoTs brings better reasoning performance than that on standard Input/Output prompts (Wang et al., 2023; Ho et al., 2022). Since asking humans to create the CoT data is not scalable, we use LLM to replace humans. This task is non-trivial for reasoning over knowledge graphs (Saparov and He, 2022). In this section, we propose a bootstrapping pipeline, i.e., given a query, using LLM to generate several CoTs and selecting them as training data with a weighted sampling strategy. Compared with the conventional Best-of-N sampling, our proposal allows more training data diversity.

Specifically, we first prepare ICL examples for high-quality CoT generation. To facilitate the learning, the pre-defined QAs are classified into different categories $\{Q_1, Q_2, \cdots, Q_M\}$. For each category Q_i, we randomly choose N_i samples $\{(g_j, e_j, q_j, a_j)\}_{j=1}^{N_i}$, where g, e, q, a denote TG, external knowledge, question and answer, respectively, and ask both GPT-3.5 and GPT-4 to provide diverse CoTs $\{c_j\}_{j=1}^{N_i}$. These CoTs will then be manually verified as ICL examples. Given a new training sample $(g_{j'}, e_{j'}, q_{j'})$, we bootstrap K CoTs with final answers $\{(c_{j',k}, \hat{a}_{j',k})\}_{k=1}^{K}$ from GPT-3.5. We first refuse the CoTs leading to incorrect answers, i.e., $\hat{a}_{j',k} \neq a^*_{j'}$, where $a^*_{j'}$ denotes the correct answer. For the accepted CoTs, we consider a weighted sampling strategy to balance usefulness and diversity. The sampling probability $P_{sample}(\cdot)$ is based on a contrastive learning *score* (Eq. 1). The *score* design, inspired by (Wang et al., 2023), considers both normalized probability $P(\cdot)$ of the correct answer and plausibility growth $G(\cdot)$ (Eq. 2). The definition of $G(\cdot)$ and $P(\cdot)$ are given in Eq. 3 and Eq. 4, respectively.

$$P_{sample}(c_k) = \text{softmax}(score(c_k)) \quad (1)$$

$$score(c_k) = \log P\left(a^* \mid q^\dagger, c_k\right) + \gamma G\left(c_k\right) \quad (2)$$

$$G\left(c_k\right) = \log \frac{P\left(a^* \mid q^\dagger, c_k\right)}{\bar{P}_{\{a' \in A'\}}\left(a' \mid q^\dagger, c_k\right)} \quad (3)$$

$$\log P\left(a \mid q^\dagger, c_k\right) = \frac{1}{|a|} \sum_{l=0}^{|a|} \log P\left(t_l \mid q^\dagger, t_{<l}\right) \quad (4)$$

where c_k denotes the current CoT in consideration; $q^\dagger := \{g, e, q\}$; a' denotes a certain wrong answer from the candidate set A'; weight γ is a hyperparameter; t_l denotes the l-th token in a, and $t_{<l}$ denotes the sequence of tokens before t_l.

3.2.2 Graph Data Augmentation

Compared with other tasks, reasoning suffers more from data insufficiency since more information (such as evidence, arguments, and logics) are involved in the intermediate steps (Huang and Chang, 2022). To address this issue, we propose several graph data augmentation strategies (Figure 3).

Our framework involves two steps: text-to-TG translation and temporal graph reasoning. Notably, for the reasoning part, the model is trained over ground-truth/verified TGs but infers on the estimated graphs. This discrepancy actually hurts the robustness of reasoning. Thus, we introduce some disturbances on the TG during training. Note that the type of disturbances should be carefully designed in order to avoid confusing the LLM. We first investigate two types of disturbances: (i) remove irrelevant edges (Eq. 5) and (ii) replace edges by using relation synonyms (Eq. 6). An edge (event) is considered irrelevant if not involved in both QA and CoT.

$$F_{irr} \in \{F : P(c|g_{\setminus\{F\}}, e, q) \approx P_0\}_{F \in g} \quad (5)$$

$$F' \in \{f_R(F) : P(c|f_R(g), e, q) \approx P_0\}_{F \in g} \quad (6)$$

where $P_0 := P(c|g, e, q)$ denotes the original conditional probability, irrelevant event F_{irr} will be randomly removed from g, and disturbed event F'

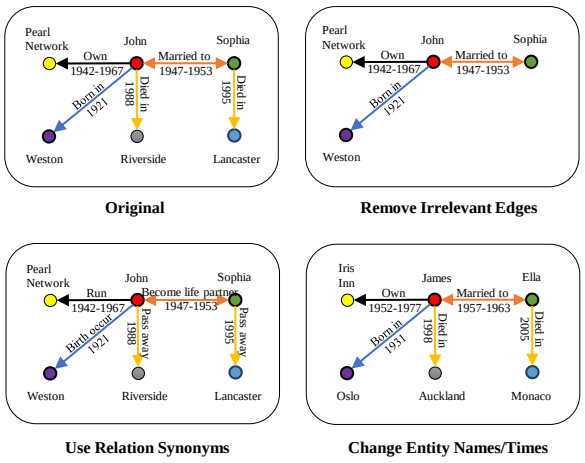


Figure 3: We further boost the model performance with several graph data augmentation strategies: remove irrelevant edges, use relation synonyms and change entities/times.

is obtained from the global mapping of relation names $f_R(\cdot)$.

Furthermore, to make sure LLM learns the underlying logic of TR instead of just memorizing semantic information, we introduce another two types of disturbances: (iii) globally map all the entity names in training data to some random names of the same type (Eq.7) , and (iv) change the times based on a global offset (Eq.7). For each relation, we generate these random entity names with GPT-3.5 by providing several examples of existing names.

$$\begin{aligned} g' &= f_T(f_E(g)),\ e' = f_T(e), \\ q' &= f_T(f_E(q)),\ c' = f_T(f_E(c)), \\ a' &= f_T(f_E(a)) \end{aligned} \tag{7}$$

where $f_E(\cdot), f_T(\cdot)$ denote the global mapping of entity names and time changing, respectively.

4 Experiments

We aim to answer the following research questions in our experiments: (1) Can our strategies (CoT bootstrapping and graph data augmentation) bring more reliable reasoning over TGs? (2) Can our two-step framework lead to better TR performance? (3) Do these learned capabilities of TR generalize to other tasks?

4.1 Experimental Setup

We demonstrate TG-LLM is a general framework by applying it to, besides TGQA, the two existing datasets, TimeQA (Chen et al., 2021) and TempReason (Tan et al., 2023a), which are constructed using Wikipedia articles, excerpts, and summaries. Examples from other datasets are listed in Appendix B. We thoroughly evaluate our framework on all the datasets with a combination of the metrics, i.e., token-level F1, exact match (EM) and perplexity-based accuracy (Acc). Besides choosing F1 and EM, which are two basic metrics for span-based QA tasks, we consider Acc for LLM evaluation, i.e., selecting from a candidate set the final answer with the lowest perplexity as the prediction. The rationale and detailed construction of the candidate set for all datasets are listed in Appendix B.

We primarily compare our framework with the leading LLMs, i.e., Llama2 (Touvron et al., 2023b), GPT-3.5 (Ouyang et al., 2022) and GPT-4 (Achiam et al., 2023). Specifically, we ran the Llama2 family on local machines (4-bit quantization for Llama2-70B due to limited computational resources), and used APIs provided by OpenAI for GPT models. We evaluated their few-shot ICL performance on the test set. To prove the effectiveness of our method, we also show the model performance with SFT on standard Input/Output prompts (SP) and CoTs from GPT-3.5. The model versions and prompt templates are provided in Appendix B and C, respectively. We also include the T5-based models (Tan et al., 2023a; Yang et al., 2023a) for a comprehensive comparison. Results on TimeQA and TempReason reported in the original papers are used, and these models are fine-tuned and evaluated on TGQA.

4.2 Implementation Details

We use Llama2-13B as the baseline due to limited computational resources. We inject two adapters with selectors into the base model for the text-to-TG translation and temporal graph reasoning. The adapters are trained in parallel. For inference, we first translate the original story into a temporal graph, and then perform reasoning on it, i.e., the adapters are used in sequence. For data generation, we use GPT-3.5 for story, TG and CoT generation, and the verification of stories and TGs. We use GPT-4 to create the ICL demonstrations of CoT generation, due to its high generation quality. All the prompt templates are given in Appendix C.

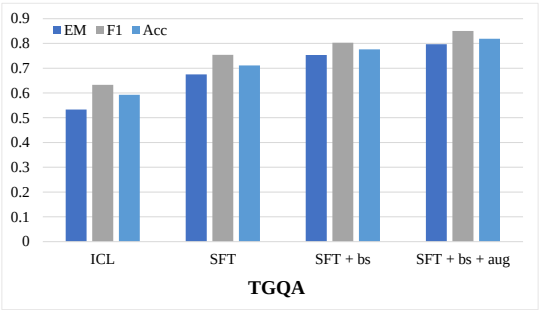


Figure 4: Performance comparison between different CoT generation strategies on TGQA.

Strategy	ER-T1	ER-T2	ER-T3	ER-T4
ICL	0.13	0.13	0.31	0.10
SFT	0.04	0.17	0.13	0.10
SFT + bs	0.06	0.13	0.03	0.08
SFT + bs + aug	0.03	0.05	0.04	0.05

Table 3: Human evaluation on the generated CoTs by different strategies. ER: error rate; T1: using wrong info; T2: logical inconsistency; T3: external knowledge error; T4: temporal graph error. (Error type explanations are listed in Appendix B.)

Model	TGQA			TimeQA						TempReason					
				Easy-mode			Hard-mode			OBQA-L2			OBQA-L3		
	EM	F1	Acc	EM	F1	Acc	EM	F1	Acc	EM	F1	Acc	EM	F1	Acc
T5-base†	0.410	0.608	-	0.600	0.682	-	0.556	0.641	-	0.260	0.450	-	0.238	0.418	-
T5-large†	0.548	0.713	-	0.631	0.716	-	0.595	0.681	-	0.327	0.509	-	0.288	0.468	-
Temp-T5†	0.640	0.778	-	-	-	-	-	-	-	0.318	0.496	-	0.261	0.430	-
REMEMO-base†	0.435	0.633	-	0.614	0.704	-	0.582	0.673	-	0.336	0.516	-	0.285	0.449	-
REMEMO-large†	0.461	0.660	-	0.637	0.723	-	0.605	**0.693**	-	0.374	**0.549**	-	0.334	**0.493**	-
GPT-3.5 (ICL-SP)	0.598	0.699	-	0.640	0.668	-	0.512	0.506	-	0.303	0.409	-	0.365	0.453	-
GPT-3.5 (ICL-CoT)	0.706	0.788	-	0.565	0.603	-	0.436	0.464	-	0.340	0.478	-	0.243	0.348	-
GPT-4* (ICL-SP)	0.772	0.829	-	**0.716**	**0.742**	-	0.571	0.546	-	**0.454**	0.525	-	**0.431**	0.485	-
GPT-4* (ICL-CoT)	**0.821**	**0.865**	-	0.662	0.693	-	0.618	0.636	-	0.388	0.480	-	0.352	0.447	-
Llama2-7B (ICL-SP)	0.415	0.596	0.447	0.352	0.408	0.367	0.341	0.404	0.354	0.205	0.344	0.226	0.044	0.084	0.153
Llama2-7B (ICL-CoT)	0.548	0.686	0.578	0.367	0.425	0.393	0.302	0.354	0.360	0.233	0.410	0.263	0.179	0.357	0.187
Llama2-13B (ICL-SP)	0.440	0.609	0.526	0.439	0.493	0.450	0.427	0.481	0.437	0.284	0.452	0.289	0.189	0.370	0.183
Llama2-13B (ICL-CoT)	0.628	0.762	0.668	0.518	0.572	0.535	0.434	0.490	0.480	0.330	0.498	0.368	0.242	0.391	0.272
Llama2-70B (ICL-SP)	0.618	0.736	0.665	0.583	0.627	0.631	0.493	0.537	0.551	0.358	0.491	0.387	0.128	0.181	0.148
Llama2-70B (ICL-CoT)	0.761	0.838	0.796	0.552	0.612	0.623	0.447	0.501	0.512	0.325	0.477	0.345	0.303	0.393	0.318
Llama2-13B (SFT-SP)	0.550	0.720	0.652	0.412	0.449	0.455	0.362	0.404	0.412	0.337	0.506	0.338	0.244	0.408	0.253
Llama2-13B (SFT-CoT)	0.646	0.722	0.705	0.550	0.586	0.564	0.332	0.391	0.379	0.256	0.433	0.281	0.285	0.409	0.305
Llama2-13B (SFT-TGR)	0.797	0.850	**0.819**	0.664	0.691	**0.673**	**0.631**	0.664	**0.649**	0.424	0.522	**0.432**	0.356	0.469	**0.399**

Table 4: Main results using different models and strategies. We report exact match (EM), token-level F1 scores, and perplexity-based accuracy (Acc). Note: (1) Results with † are reported in the original papers. We only fine-tune and evaluate the models on our dataset. (2) Results with * are evaluated on 1000 random test samples.

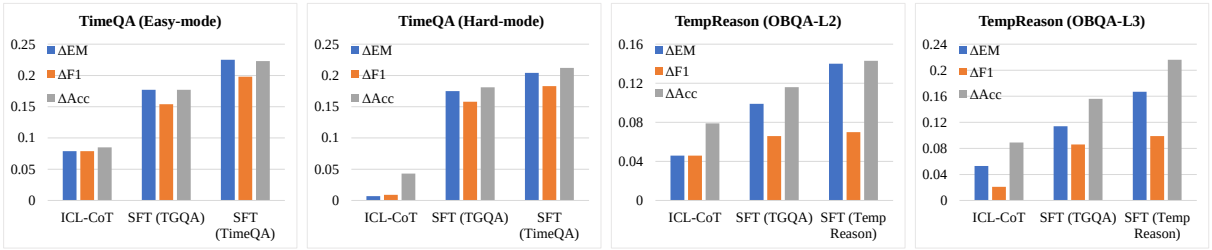


Figure 5: Performance comparison between different strategies on TimeQA and TempReason. To obtain a fair comparison, we use Llama2-13B as the base model for all strategies. The basic strategy used to calculate the performance changes is in-context learning with standard Input/Output prompt (ICL-SP).

4.3 Main Results

Can our strategies bring more reliable reasoning over TGs? We show the comparison between ICL with CoTs, SFT with CoTs, bootstrapping CoTs, and graph data augmentation on TGQA (Figure 4). It can be seen that LLM learns TR better with SFT than ICL. By providing CoTs with bootstrapping and graph data augmentation strategies, the model performance gets further enhanced. Furthermore, inspired by (Wang et al., 2023), we manually check 100 CoTs generated by different strategies (Table 3). Evaluators are asked to classify the errors into four types (using wrong info, logical inconsistency, external knowledge error, and temporal graph error). It can be seen that our strategies reduce all types of error rate.

Can our two-step framework lead to better temporal reasoning performance? We show the comparison between different models and strategies on all datasets (Table 4). First, we observed that among all the LLMs with ICL, GPT-4 has the strongest performance as expected. For the Llama2 family, larger models have better performance due to advanced context understanding and improved generalization. We also found that CoTs not always bring better TR due to unreliable intermediate results and hallucinations. From the results of some alternative strategies, where SFT-SP and SFT-CoT denote supervised fine-tuning on standard Input/Output prompts and vanilla CoT distillation, respectively, we prove the effectiveness of our framework (SFT-TGR). More importantly, our model, which is based on Llama2-13B, shows a comparable or even better performance than GPT-4 on all datasets. It can be seen that the two-step framework brings a substantial performance improvement on different datasets. We hypothesize

this performance improvement is because our two-step reasoning process provides an easier path toward answering the temporal questions for LLM.

Do these learned capabilities of temporal reasoning generalize to other tasks? We show the comparison between different strategies (ICL with SP/CoT, SFT with TGQA/original data) on the two existing datasets, TimeQA and TempReason (Figure 5). Our framework learns the capabilities of text-to-TG translation and temporal graph reasoning, which brings better TR. More importantly, we observed that SFT on TGQA improves the model performance compared with ICL. It can be concluded that these necessary capabilities in TR are generalizable to different data distributions. Since TGQA is fully controllable and requires minimal supervision, we actually provide a general and effective way of TR capability improvement.

4.4 Ablation Study

We ablate different modules to see their contributions to the performance. We show the performance comparison between different configurations (Table 5). To obtain a fair comparison, we use Llama2-13B as the base model for all configurations. From the ablation study, we obtain some insights: (1) LLM can benefit from explicitly presented (temporal) graph which is intuitive, concise and structured. (2) Given a reliable graph, CoT bootstrapping with contrastive learning brings better performance than vanilla CoT distillation. (3) Data augmentation is necessary for LLMs to perform complex tasks such as temporal reasoning. (4) The introduction of external knowledge such as mathematics and commonsense can further augment the generation.

Config.	EM	F1	Acc
Baseline (SFT-CoT)	0.646	0.722	0.705
TG	0.675	0.754	0.711
TG + CoT(bs)	0.723	0.797	0.736
TG + CoT(bs + aug)	0.782	0.838	0.795
TG + EK + CoT(bs)	0.753	0.803	0.776
TG + EK + CoT(bs + aug)	**0.797**	**0.850**	**0.819**

Table 5: Ablation study results on TGQA with Llama2-13B. TG: temporal graph, CoT: chain of thought, EK: external knwoledge, bs: bootstrapping, aug: graph data augmentation.

5 Related Work

Language-based Temporal Reasoning. Recently, language-based TR has gained substantial research focus (Liu et al., 2023, 2024b; Chen et al., 2024a,b; Wang et al., 2024; Jiayang et al., 2024; Xia et al., 2024). The vision here is to help LMs understand temporal concepts and logic such that they can perform more complicate tasks. Existing methods mainly solve this problem with time-aware language modeling. For example, (Rosin et al., 2022; Pereira, 2022; Tan et al., 2023a) propose specific pre-training/fine-tuning strategies for robust TR. On the other hand, (Ning et al., 2019; Zhou et al., 2020a,b; Yang et al., 2020, 2023a) design auxiliary objectives to introduce external temporal knowledge. Although these methods made some progress, representation learning for the underlying structure and logic of TR are either ignored or not explicitly involved.

Reasoning towards LMs. Although LLM has exhibited some emergent behaviors (Zhang et al., 2024a; Lai et al., 2024a,b; Lyu et al., 2024), it is still unknown whether they can actually perform reasoning and how strong their capability of reasoning is. Existing methods that try to elicit or enhance reasoning can be divided into two directions: reasoning-involved modeling or hybrid methods. For example, (Wei et al., 2022; Kojima et al., 2022; Yao et al., 2024) use in-context learning where some demonstrations including intermediate reasoning steps are provided in prompt. (Nye et al., 2021; Wang et al., 2023; Ho et al., 2022) fine-tune LMs with the intermediate thinking process before producing final answers. On the other hand, (Schick et al., 2024; Kynoch et al., 2023; Xu et al., 2023) combine LMs with domain-specific external tools, empowering the model to perform more complex tasks that require reasoning and interactions with environment.

Reasoning over Knowledge Graphs. Knowledge graphs (KGs) as the foundation representation of semantic and symbolic reasoning have been widely adopted in the past (Huang et al., 2023; Wan et al., 2024; Li et al., 2024). Related work includes symbolic reasoning over temporal KGs (Yang et al., 2022; Xiong et al., 2023, 2024), and language-based reasoning over static KGs (Cheng et al., 2024; Zhang et al., 2024b; Liu et al., 2024a; Zhao et al., 2024; Xu et al., 2024; Wei et al., 2024). In this paper, we build the connection between language-based and symbolic-based TR. This connection brings the potential for extending these KG-based methods to language-based tasks. Different from existing methods (Luo et al., 2023; Zhang et al., 2023; Yuan et al., 2024; Gao et al., 2024),

which limit their application to certain tasks, our framework offers enhanced generalization and usability, largely attributed to its innovative use of text-to-graph translation as a precursor to graph reasoning.

6 Conclusion

TG-LLM, a novel framework for LMs, has been proposed to improve their performance on temporal reasoning. To produce reliable final answers, our framework equips LLMs with the temporal graph and intermediate reasoning steps. Extensive experiments indicate that TG-LLM achieves better performance than existing pipelines. An interesting direction for future work is to extend it to more complex applications such as inductive and abductive reasoning. Due to the graph structure and capability of deliberate reasoning, it is promising to improve the model performance on these tasks as well.

Limitations

Graph-augmented approaches (Jin et al., 2024; Fan et al., 2024; Shang and Huang, 2024; He and Hooi, 2024) including TG-LLM help language models better learn related concepts from the perspective of graph. Although we demonstrate TG-LLM has good performance on understanding temporal relations, it still needs adaptations for temporal commonsense reasoning (Zhou et al., 2019; Qin et al., 2021). Explicit in-context integration of commonsense presents opportunities for this task. Further, we mainly improve the capability of LLMs by introducing a new paradigm and providing more plausible and informative training data. There can be opportunities such as simulating an environment to provide feedback to LLMs (Hao et al., 2023). For example, we can verify the generated TG based on prior knowledge such as the time gap between someone's birth date and death date, i.e., a person's lifespan, should fall into a certain range. In this way, we can further improve the performance.

Ethics Statement

In this paper, we adopt YAGO11k for fine-tuning the language models. The dataset is publicly available, and is for research purposes only. We also use GPT model to generate text based on YAGO11k, for which OpenAI has been committed to addressing ethical considerations. In addition, we adopt TimeQA and TempReason for evaluation. Both datasets are publicly available, and are for research purposes only. However, they may still contain improper or harmful content. None of such content reflects the opinions of the authors.

Acknowledgements

This work was supported by a sponsored research award by Cisco Research.

References

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

Jianhao Chen, Haoyuan Ouyang, Junyang Ren, Wentao Ding, Wei Hu, and Yuzhong Qu. 2024a. Timeline-based sentence decomposition with in-context learning for temporal fact extraction. *arXiv preprint arXiv:2405.10288*.

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

Zhuo Chen, Zhao Zhang, Zixuan Li, Fei Wang, Yutao Zeng, Xiaolong Jin, and Yongjun Xu. 2024b. Self-improvement programming for temporal knowledge graph question answering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14579–14594.

Sitao Cheng, Ziyuan Zhuang, Yong Xu, Fangkai Yang, Chaoyun Zhang, Xiaoting Qin, Xiang Huang, Ling Chen, Qingwei Lin, Dongmei Zhang, et al. 2024. Call me when necessary: Llms can efficiently and faithfully reason over structured environments. *arXiv preprint arXiv:2403.08593*.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2023. Timebench: A comprehensive evaluation of temporal reasoning abilities in large language models. *arXiv preprint arXiv:2311.17667*.

Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2001–2011.

Wenqi Fan, Shijie Wang, Jiani Huang, Zhikai Chen, Yu Song, Wenzhuo Tang, Haitao Mao, Hui Liu, Xiaorui Liu, Dawei Yin, et al. 2024. Graph machine learning in the era of large language models (llms). *arXiv preprint arXiv:2404.14928*.

Yifu Gao, Linbo Qiao, Zhigang Kan, Zhihua Wen, Yongquan He, and Dongsheng Li. 2024. Two-stage generative question answering on temporal knowledge graph using large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6719–6734, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4816–4821, Brussels, Belgium. Association for Computational Linguistics.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*.

Yufei He and Bryan Hooi. 2024. Unigraph: Learning a cross-domain graph foundation model from natural language. *arXiv preprint arXiv:2402.13630*.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Wenke Huang, Guancheng Wan, Mang Ye, and Bo Du. 2023. Federated graph semantic and structural learning. In *Proc. Int. Joint Conf. Artif. Intell*, pages 3830–3838.

Cheng Jiayang, Lin Qiu, Chunkit Chan, Xin Liu, Yangqiu Song, and Zheng Zhang. 2024. Eventground: Narrative reasoning by grounding to eventuality-centric knowledge graphs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6622–6642.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Suhang Wang, Yu Meng, and Jiawei Han. 2024. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. *arXiv preprint arXiv:2404.07103*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Brandon Kynoch, Hugo Latapie, and Dwane van der Sluis. 2023. Recallm: An adaptable memory mechanism with temporal understanding for large language models. *arXiv preprint arXiv:2307.02738*.

Zhixin Lai, Jing Wu, Suiyao Chen, Yucheng Zhou, Anna Hovakimyan, and Naira Hovakimyan. 2024a. Language models are free boosters for biomedical imaging tasks. *arXiv preprint arXiv:2403.17343*.

Zhixin Lai, Xuesheng Zhang, and Suiyao Chen. 2024b. Adaptive ensembles of fine-tuned transformers for llm-generated text detection. *arXiv preprint arXiv:2403.13335*.

Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Chen Li, Haotian Zheng, Yiping Sun, Cangqing Wang, Liqiang Yu, Che Chang, Xinyu Tian, and Bo Liu. 2024. Enhancing multi-hop knowledge graph reasoning through reward shaping techniques. *arXiv preprint arXiv:2403.05801*.

Xingxuan Li, Liying Cheng, Qingyu Tan, Hwee Tou Ng, Shafiq Joty, and Lidong Bing. 2023. Unlocking temporal question answering for large language models using code execution. *arXiv preprint arXiv:2305.15014*.

Jason Xinyu Liu, Ziyi Yang, Ifrah Idrees, Sam Liang, Benjamin Schornstein, Stefanie Tellex, and Ankit Shah. 2023. Grounding complex natural language commands for temporal tasks in unseen environments. In *Conference on Robot Learning*, pages 1084–1110. PMLR.

Xiaoze Liu, Feijie Wu, Tianyang Xu, Zhuo Chen, Yichi Zhang, Xiaoqian Wang, and Jing Gao. 2024a. Evaluating the factuality of large language models using large-scale knowledge graphs. *arXiv preprint arXiv:2404.00942*.

Yuchen Liu, Luigi Palmieri, Sebastian Koch, Ilche Georgievski, and Marco Aiello. 2024b. Delta: Decomposed efficient long-term robot task planning using large language models. *arXiv preprint arXiv:2404.03275*.

Linhao Luo, Jiaxin Ju, Bo Xiong, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Chatrule: Mining logical rules with large language models for knowledge graph reasoning. *arXiv preprint arXiv:2309.01538*.

Weimin Lyu, Xiao Lin, Songzhu Zheng, Lu Pang, Haibin Ling, Susmit Jha, and Chao Chen. 2024. Task-agnostic detector for insertion-based backdoor attacks. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2808–2822, Mexico City, Mexico. Association for Computational Linguistics.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2019. Joint reasoning for temporal and causal relations. *arXiv preprint arXiv:1906.04941*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295*.

Lis Kanashiro Pereira. 2022. Attention-focused adversarial training for robust temporal reasoning. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7352–7359.

Lianhui Qin, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi, and Manaal Faruqui. 2021. Timedial: Temporal commonsense reasoning in dialog. *arXiv preprint arXiv:2106.04571*.

Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo M Ponti, and Shay B Cohen. 2023. Are large language models temporally grounded? *arXiv preprint arXiv:2311.08398*.

Guy D Rosin, Ido Guy, and Kira Radinsky. 2022. Time masking for temporal language models. In *Proceedings of the fifteenth ACM international conference on Web search and data mining*, pages 833–841.

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Wenbo Shang and Xin Huang. 2024. A survey of large language models on generative graph analytics: Query, learning, and applications. *arXiv preprint arXiv:2404.14809*.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023a. Towards benchmarking and improving the temporal reasoning capability of large language models. *arXiv preprint arXiv:2306.08952*.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023b. Towards robust temporal reasoning of large language models via a multi-hop qa dataset and pseudo-instruction tuning. *arXiv preprint arXiv:2311.09821*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal reasoning in natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4070–4078, Online. Association for Computational Linguistics.

Guancheng Wan, Wenke Huang, and Mang Ye. 2024. Federated graph learning under domain shift with generalizable prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15429–15437.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023. Scott: Self-consistent chain-of-thought distillation. *arXiv preprint arXiv:2305.01879*.

Yucheng Wang, Ruibing Jin, Min Wu, Xiaoli Li, Lihua Xie, and Zhenghua Chen. 2024. K-link: Knowledge-link graph from llms for enhanced representation learning in multivariate time-series data. *arXiv preprint arXiv:2403.03645*.

Yuqing Wang and Yun Zhao. 2023. Tram: Benchmarking temporal reasoning for large language models. *arXiv preprint arXiv:2310.00835*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Lanning Wei, Jun Gao, and Huan Zhao. 2024. Towards versatile graph learning approach: from the perspective of large language models. *arXiv preprint arXiv:2402.11641*.

Yifan Wei, Yisong Su, Huanhuan Ma, Xiaoyan Yu, Fangyu Lei, Yuanzhe Zhang, Jun Zhao, and Kang Liu.

2023. Menatqa: A new dataset for testing the temporal comprehension and reasoning abilities of large language models. *arXiv preprint arXiv:2310.05157*.

Yuwei Xia, Ding Wang, Qiang Liu, Liang Wang, Shu Wu, and Xiaoyu Zhang. 2024. Enhancing temporal knowledge graph forecasting with large language models via chain-of-history reasoning. *arXiv preprint arXiv:2402.14382*.

Siheng Xiong, Yuan Yang, Faramarz Fekri, and James Clayton Kerce. 2023. TILP: Differentiable learning of temporal logical rules on knowledge graphs. In *The Eleventh International Conference on Learning Representations*.

Siheng Xiong, Yuan Yang, Ali Payani, James C Kerce, and Faramarz Fekri. 2024. Teilp: Time prediction over knowledge graphs via logical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16112–16119.

Binfeng Xu, Xukun Liu, Hua Shen, Zeyu Han, Yuhan Li, Murong Yue, Zhiyuan Peng, Yuchen Liu, Ziyu Yao, and Dongkuan Xu. 2023. Gentopia: A collaborative platform for tool-augmented llms. *arXiv preprint arXiv:2308.04030*.

Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Kang Liu, and Jun Zhao. 2024. Generate-on-graph: Treat llm as both agent and kg in incomplete knowledge graph question answering. *arXiv preprint arXiv:2404.14741*.

Sen Yang, Xin Li, Lidong Bing, and Wai Lam. 2023a. Once upon a time in graph: Relative-time pretraining for complex temporal reasoning. *arXiv preprint arXiv:2310.14709*.

Sen Yang, Xin Li, Leyang Cui, Lidong Bing, and Wai Lam. 2023b. Neuro-symbolic integration brings causal and reliable reasoning proofs. *arXiv preprint arXiv:2311.09802*.

Yuan Yang, Siheng Xiong, James C Kerce, and Faramarz Fekri. 2022. Temporal inductive logic reasoning. *arXiv preprint arXiv:2206.05051*.

Yuan Yang, Siheng Xiong, Ali Payani, Ehsan Shareghi, and Faramarz Fekri. 2023c. Harnessing the power of large language models for natural language to first-order logic translation. *arXiv preprint arXiv:2305.15541*.

Zonglin Yang, Xinya Du, Alexander Rush, and Claire Cardie. 2020. Improving event duration prediction via time-aware pre-training. *arXiv preprint arXiv:2011.02610*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Chenhan Yuan, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2024. Back to the future: Towards explainable temporal reasoning with large language models. In *Proceedings of the ACM on Web Conference 2024*, pages 1963–1974.

Ye Zhang, Kailin Gui, Mengran Zhu, Yong Hao, and Haozhan Sun. 2024a. Unlocking personalized anime recommendations: Langchain and llm at the forefront. *Journal of Industrial Engineering and Applied Science*, 2(2):46–53.

Yichi Zhang, Zhuo Chen, Wen Zhang, and Huajun Chen. 2023. Making large language models perform better in knowledge graph completion. *arXiv preprint arXiv:2310.06671*.

Zheyuan Zhang, Zehong Wang, Shifu Hou, Evan Hall, Landon Bachman, Jasmine White, Vincent Galassi, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. 2024b. Diet-odin: A novel framework for opioid misuse detection with interpretable dietary patterns. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6312–6323.

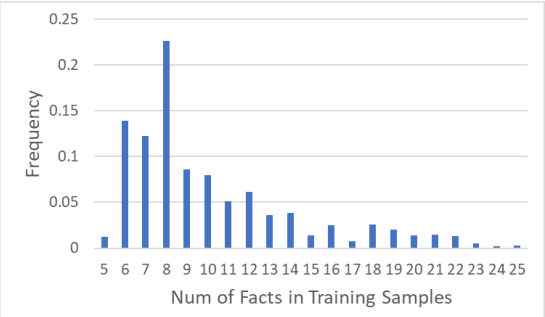
Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. 2024. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4443–4454.

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. *arXiv preprint arXiv:1909.03065*.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020a. Temporal common sense acquisition with minimal supervision. *arXiv preprint arXiv:2005.04304*.

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

Zhaocheng Zhu, Yuan Xue, Xinyun Chen, Denny Zhou, Jian Tang, Dale Schuurmans, and Hanjun Dai. 2023. Large language models can learn rules. *arXiv preprint arXiv:2310.07064*.



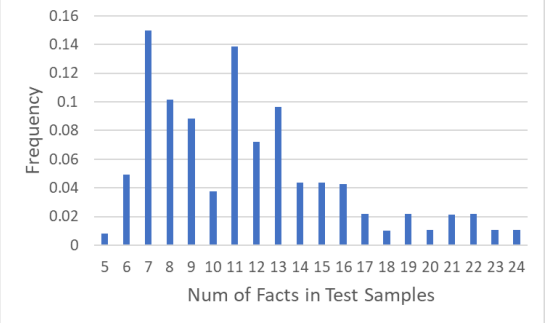
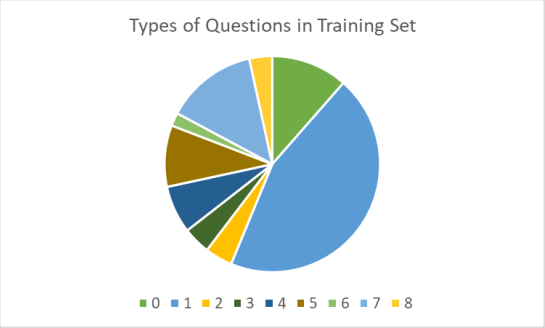


Figure 6: Number of facts distribution in TGQA.



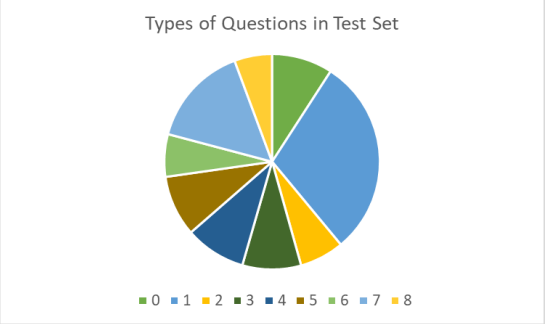


Figure 7: Types of questions distribution in TGQA.

A Dataset Statistics of TGQA

In this section, we provide statistics for our TGQA dataset. We obtain 400 samples for training, 100 for validation and another 100 for test, with about 30 QA pairs in a single sample. We show the distribution of the number of facts (in one sample) in training (with validation) and test set (Figure 6). It determines the complexity of the TR tasks to some extent. We also show the distribution of question types in training (with validation) and test set (Figure 7), where Q0: "Which event occurred first, <Event_A> or <Event_B>?", Q1: "Given the following <N> events: <Event_A>, <Event_B>, <Event_C>, <Event_D>, $\cdots$, which event is the first/second/third/fourth/$\cdots$ one in the chronological order?", Q2: "How long did the event <Event_A> last?", Q3: "", Q4: "How much time passed between the start of <Event_A> and the start of <Event_B>?", Q5: "What happened right before/after <Event_A> started?", Q6: "When did the <Event_A> occur?", Q7: "True or false: <Event_A> and <Event_B> happened at the same year?", Q8: "True or false: <Event_A> was still happening when <Event_B> started?". Note that we give eight question categories in Table 2, since in a strict sense Q0 and Q1 can be considered as the same type. Among all the question types, Q1 has the largest portion since it has multiple variants. Similarly, Q7 with two variants also has a larger portion. To mitigate question category imbalance, we first calculate the metrics for each category, and then use the average as the final scores. Additionally, we show examples for the global mapping for entity names (Table 6) and external knowledge (Table 7) used in our dataset.

Entity Name Mapping	
Ori	**New**
Beverly_Adams	Isabella_Thompson
Edmonton	Lancaster
Al_Gore	Chris_Evans
Carrara_Stadium	Maplewood_Arena
Bend_Sinister_(novel)	Lakefield_Chronicles_(novel)
...	...

Table 6: We use a global mapping for entity names from GPT-3.5 to avoid data leakage.

External Knowledge
1885 before 1893 before 1916 before 1918 before 1922 before 1928 before 1941
1918 - 1916 = 2
1928 - 1893 = 35
1928 - 1922 = 6
1941 - 1918 = 23
2 < 6 < 23 < 35

Table 7: We integrate the necessary mathematics and commonsense in context as external knowledge for TR.

B Experiment Details

In this section, we present more experiment details for further research. We include graph data aug-

Temporal Graph:
(Use relation synonyms/Remove irrelevant edges)
[1] (John Thompson was born in Weston) starts at 1921; [2] (John Thompson run Pearl Network) starts at 1942; [3] (Sophia Parker and John Thompson became life partner) starts at 1947; [4] (John Thompson and Sophia Parker became life partner) starts at 1947; [5] (Sophia Parker and John Thompson became life partner) ends at 1953; [6] (John Thompson and Sophia Parker) became life partner) ends at 1953; [7] (John Thompson run Pearl Network) ends at 1967.

Graph-based QAs:
(No need for change)
Q: True or false: event (John Thompson owned Pearl Network) was longer in duration than event (Sophia Parker was married to John Thompson)?
CoT: The duration for each event can be calculated as follows:
(John Thompson owned Pearl Network) starts at 1942, ends at 1967, 1967 - 1942 = 25
(Sophia Parker was married to John Thompson) starts at 1947, ends at 1953, 1953 - 1947 = 6
25 is greater than 6 , thus, the answer is True.
A: True.

Temporal Graph:
(Change entities/times)
[1] (James Brown was born in Oslo) starts at 1931; [2] (James Brown owned Iris Inn) starts at 1952; [3] (Ella Perry was married to James Brown) starts at 1957; [4] (James Brown was married to Ella Perry) starts at 1957; [5] (Ella Perry was married to James Brown) ends at 1963; [6] (James Brown was married to Ella Perry) ends at 1963; [7] (James Brown owned Iris Inn) ends at 1977; [8] (James Brown died in Auckland) starts at 1998; [9] (Ella Perry died in Monaco) starts at 2005.

Graph-based QAs:
(Change entities/times)
Q: True or false: event (James Brown owned Iris Inn) was longer in duration than event (Ella Perry was married to James Brown)?
CoT: The duration for each event can be calculated as follows:
(James Brown owned Iris Inn) starts at 1952, ends at 1977, 1977 - 1952 = 25
(Ella Perry was married to James Brown) starts at 1957 , ends at 1963, 1963 - 1957 = 6
25 is greater than 6 , thus, the answer is True.
A: True.

Table 8: We propose several graph data augmentation strategies for reasoning over temporal knowledge graph. The original TG is shown in Table 1. We highlight the changed information.

mentation examples, model versions and evaluation tasks & metrics.

Graph Data Augmentation. We show with an example our graph data augmentation strategies in Table 8. For TGs with relation synonym replacement or irrelevant edges removal, there is no need for change on graph-based QAs. To contrast, for TGs with entity names/times mapping, we need to change with the corresponding entity names/times in graph-based QAs to ensure the consistency.

Evaluation Tasks & Metrics. Besides TGQA, we consider in experiments the two existing datasets TimeQA (Chen et al., 2021) and TempReason (Tan et al., 2023a). Specifically, TimeQA contains two difficulty levels (Table 9). The easy-level split tends to be the information extraction task while the hard-level split involves understanding the relation between different temporal expressions. On the other hand, TempReason contains three levels (L1: Time-Time Relation, L2: Time-Event Relation, L3: Event-Event Relation) and three settings (OBQA: open-book QA, CBQA: close-book QA, ReasonQA: facts-based QA) (Table 10). We observed that some stories in TempReason are incomplete which partially leads to the low accuracy of LLMs.

We evaluate our framework on all the datasets with the metrics of token-level F1, exact match (EM) and perplexity-based accuracy (Acc). F1 and EM are two basic metrics for span-based QA tasks. However, the free-form prediction of LLMs might hurt their performance under these generation-based metrics. To solve this problem, we introduce perplexity-based accuracy, i.e., selecting from a candidate set the final answer with the lowest perplexity. For questions with multiple correct answers, we follow the strategy proposed in (Chen et al., 2021) to get the best result among them. Since there are multiple question categories in TGQA, we first calculate the metrics for each category, and then use the average as final scores to mitigate question category imbalance.

Human evaluation on CoT generation found out four types of error in total (Table 3). T1 means that LLM uses wrong information such as wrong start/end time during reasoning. T2 suggests that LLM makes logical errors, e.g., mentioning "Event A ends before Event B starts" in CoT but determining the statement "Event A was still happening when Event B starts" to be true. T3 denotes that LLM makes errors on external knowledge, e.g., claiming that "the date 1978 is after 1983". T4 indicates that there exists errors in the extracted TG

TimeQA
George Washington (February 22, 1732 – December 14, 1799) was an American Founding Father, military officer, politician and statesman who served as the first · · ·
Questions (Easy-mode): What position did George Washington hold in June 1775? What position was held by George Washington between 1778 and 1788? · · ·
Questions (Hard-mode): George Washington took which position before 1778? What was George Washington's position in early 1780s? · · ·

Table 9: Example questions of two difficulty levels in TimeQA. Easy-mode: the query time expression is explicitly mentioned in the story. Hard-mode: obtaining the answer needs inference based on the temporal relation between the query time expression and the one mentioned in the story.

TempReason
Lionel Andrés "Leo" Messi (born 24 June 1987) is an Argentine professional footballer who plays as a forward for and captains both Major League Soccer club · · ·
Questions (L1): What is the year after 2010? · · ·
Questions (L2): What team did Leo Messi play for in 2010? · · ·
Questions (L3): What team did Leo Messi play for after Barcelona? · · ·

Table 10: Example questions of three difficulty levels under the OBQA setting in TempReason. L1: Time-Time Relation, L2: Time-Event Relation, L3: Event-Event Relation.

that lead to the wrong conclusion.

Candidate Answer Generation. We involve candidate answers in the calculation of Acc. For TempReason, the authors provide negative answers since they perform time-sensitive reinforcement learning. That is, they use the score of the correct answers and wrong answers from the language model as reward to further finetune the model parameter. For TimeQA, we generate the candidates in the following way. Given a story, there are multiple related questions which share the same subject/object entity and relation. The correct answer changes with the query time. For example, "What position did George Washington hold in 1777/1790/1799?" Answer: "Commander in Chief/Presidency/Chancellor". We collect the answers of these related questions as candidates. More generally, if there are no such related questions, we will use all the entities in the corresponding TG as candidates.

Model Versions. The versions of the LLMs used in our experiments are listed below. For the Llama2 family, all the model weights can be downloaded from the platform of Hugging Face. For the GPT models, all the model weights can be accessed through the OpenAI APIs.

- Llama2-7B (meta-llama/Llama-2-7b-hf)
- Llama2-13B (meta-llama/Llama-2-13b-hf)
- Llama2-70B (meta-llama/Llama-2-70b-hf)
- GPT-3.5 (gpt-3.5-turbo)
- GPT-4 (gpt-4-1106-preview)

C Example Prompts

In this section, we show example prompts used in our framework. Specifically, Table 12 shows an example of the graph-based story generation in TGQA; Table 13 presents an example of the automatic story-temporal graph alignment verification in TGQA; Table 14 provides an example of the temporal info identification in TimeQA; Table 15 illustrates an example of the graph construction in TimeQA; Table 16 gives an example of the automatic temporal graph-QA alignment verification in TimeQA; Table 17 depicts an illustrative case of the CoTs bootstrapping in TGQA.

D Fine-grained Results of TGQA

We provide the fine-grained results of TGQA in Table 11. In TGQA, there are nine types of questions as explained in Appendix A. We show all the models with different strategies. Zero-shot performance is also considered to investigate the effect of in-context examples. Note that we do not include zero-shot performance of the Llama2 family since they are not fine-tuned on instructions, i.e., we can not obtain valid zero-shot learning results. For zero-shot learning results, the format of generation is not

Model	TGQA									
	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Total
GPT-3.5‡ (0-shot SP)	0.801	0.435	0.660	0.355	0.930	0.955	0.710	0.315	0.617	0.642
GPT-3.5 (1-shot SP)	0.842	0.411	0.656	0.555	0.924	0.650	0.500	0.353	0.495	0.598
GPT-3.5‡ (0-shot CoT)	0.930	0.565	0.680	0.665	0.920	0.945	0.695	0.395	0.596	0.710
GPT-3.5 (1-shot CoT)	0.884	0.457	0.783	0.791	0.921	0.832	0.755	0.357	0.574	0.706
GPT-4‡* (0-shot SP)	0.801	0.689	0.920	0.620	**0.930**	0.940	0.758	0.575	0.737	0.776
GPT-4* (1-shot SP)	0.820	0.660	0.870	0.630	0.910	**0.990**	0.760	0.610	0.700	0.772
GPT-4‡* (0-shot CoT)	0.950	0.692	0.901	**0.881**	0.919	0.980	0.778	0.647	**0.858**	**0.848**
GPT-4* (1-shot CoT)	0.930	**0.758**	**0.903**	0.869	0.892	0.920	0.758	**0.667**	0.680	0.821
Llama2-7B (1-shot SP)	0.554	0.216	0.163	0.147	0.353	0.924	0.683	0.115	0.579	0.415
Llama2-7B (1-shot CoT)	0.660	0.264	0.484	0.603	0.693	0.947	0.632	0.079	0.574	0.548
Llama2-13B (1-shot SP)	0.442	0.341	0.353	0.192	0.686	0.640	0.561	0.163	0.580	0.440
Llama2-13B (1-shot CoT)	0.597	0.263	0.670	0.678	0.881	0.947	0.736	0.258	0.622	0.628
Llama2-70B (1-shot SP)	0.752	0.447	0.520	0.349	0.941	0.891	0.665	0.413	0.585	0.618
Llama2-70B (1-shot CoT)	**0.980**	0.490	0.878	0.832	0.944	0.878	**0.807**	0.294	0.750	0.761
Llama2-13B (1-shot SFT - TG)	0.878	0.515	0.656	0.866	0.835	0.845	0.726	0.369	0.383	0.675
Llama2-13B (1-shot SFT - TG + CoT(bs))	**0.947**	0.732	**0.747**	0.825	**0.931**	**0.987**	0.495	0.466	0.649	0.753
Llama2-13B (1-shot SFT - TG + CoT(bs + aug))	0.931	0.710	0.729	**0.860**	0.927	0.977	0.627	**0.534**	**0.739**	0.782
Llama2-13B (1-shot SFT - TG + EK + CoT(bs + aug))	0.944	**0.775**	0.729	0.849	0.924	0.980	**0.783**	0.506	0.681	**0.797**

Table 11: Fine-grained results on TGQA using different models and strategies. We report exact match (EM) as the performance metric. Note: (1) Results with * are evaluated on 1000 random test samples. (2) Results with ‡ are parsed by GPT-3.5 during evaluation (might introduce errors).

guaranteed. The original accuracy is very low since rule-based parser cannot handle it. Instead, we use GPT-3.5 as parser to extract the answer from the generation (might introduce errors). Also, we only adopt one example in context due to context length limitation (long-context LLM performance could increase with more in-context demonstrations).

It can be seen that in-context demonstrations are not necessary for powerful pre-trained models such as GPT-3.5 and GPT-4. Undoubtedly, GPT-4 obtains the best in-context learning performance on most categories. However, for some categories, the best performance of GPT-4 is around 0.6 or 0.7, which is not sufficiently well. On the other hand, we show that our strategies improve the performance in a progressive manner. Note that our model achieves the best overall performance by improving on all the categories. Specifically, deliberate reasoning with CoT bootstrapping brings better performance for most categories. Furthermore, graph data augmentation and external knowledge further enhance the capability of the model while preserving the existing good performance on those categories.

(Molly Adams was born in Seattle) starts at 1896
(Liam Thomas Dawson was born in Seattle) starts at 1896
(Molly Adams was married to Liam Thomas Dawson) starts at 1920
(Liam Thomas Dawson was married to Molly Adams) starts at 1920
(Liam Thomas Dawson won prize James Parker Prize for Advanced Biomedical Studies) starts at 1946
(Liam Thomas Dawson won prize Champion Award in Biology or Science) starts at 1947
(Liam Thomas Dawson won prize Oakley Smith Prize) starts at 1948
(Liam Thomas Dawson won prize Member of the National Academy) starts at 1950
(Molly Adams was married to Liam Thomas Dawson) ends at 1957
(Liam Thomas Dawson was married to Molly Adams) ends at 1957
(Liam Thomas Dawson died in Kingsville, Texas) starts at 1984

Based on the above facts, generate a short story for me.

Table 12: Example of the graph-based story generation in TGQA.

Once upon a time, in the vibrant city of Seattle, two remarkable individuals, Molly Adams and Liam Thomas Dawson, were born in the year 1896. Little did they know that their destinies were entwined from the very beginning.

Growing up in the same neighborhood, Molly and Liam developed a deep friendship that blossomed into something more as they entered adulthood. In the year 1920, their love story officially began as they exchanged vows and embarked on a journey of companionship that would last for decades.

. . .

When did the event (Molly Adams was married to Liam Thomas Dawson) end?

Table 13: Example of the story-temporal graph alignment verification in TGQA.

Knox Cunningham

Sir Samuel Knox Cunningham, 1st Baronet, QC (3 April 1909 – 29 July 1976) was a Northern Irish barrister, businessman and politician. As an Ulster Unionist politician at a time when the Unionists were part of the Conservative Party, he was also a significant figure in United Kingdom politics as Parliamentary Private Secretary to Harold Macmillan. His nephew was Sir Josias Cunningham.

. . .

Extract all the time expressions such as 'June 1994', '1973', 'late 1980s'.

Table 14: Example of the temporal info identification in TimeQA.

Knox Cunningham

Sir Samuel Knox Cunningham, 1st Baronet, QC (3 April 1909 – 29 July 1976) was a Northern Irish barrister, businessman and politician. As an Ulster Unionist politician at a time when the Unionists were part of the Conservative Party, he was also a significant figure in United Kingdom politics as Parliamentary Private Secretary to Harold Macmillan. His nephew was Sir Josias Cunningham.

. . .

Construct a timeline for Knox Cunningham's position. You should only consider these time points (3 April 1909, 1930s, 1939, 1942, 1943, 1945, 1947, 1949, 1954, 29 July 1976).

Table 15: Example of the temporal graph construction in TimeQA.

3 April 1909: Knox Cunningham was born.
1930s: He studied law and began his legal career.
1935: On 2 July 1935, he married Dorothy Enid Riley JP.
1939: Cunningham was called to the Bar by the Middle Temple.
1942: He was called to the Bar in Northern Ireland.
1943: Cunningham fought the Belfast West by-election.
1945: He contested the same seat in the general election.
1947: Knox Cunningham became involved with the World Alliance of YMCAs.
1949: He became Chairman of the National Council of the YMCA.
1954: Cunningham was elected to Orpington Urban District Council.
29 July 1976: Knox Cunningham passed away.

Which position did Knox Cunningham hold from May 1955 to Apr 1956?

Table 16: Example of the temporal graph-QA alignment verification in TimeQA.

Example:
Timeline:
(Liam Mitchell was born in Harrison) starts at 1885
(Maddox Reynolds was born in Glenville) starts at 1893
(Liam Mitchell was married to Emma Scott) starts at 1916
(Emma Scott was married to Liam Mitchell) starts at 1916
(Liam Mitchell was married to Emma Scott) ends at 1918
(Emma Scott was married to Liam Mitchell) ends at 1918
(Maddox Reynolds was married to Liam Mitchell) starts at 1922
(Liam Mitchell was married to Maddox Reynolds) starts at 1922
(Maddox Reynolds was married to Liam Mitchell) ends at 1928
(Liam Mitchell was married to Maddox Reynolds) ends at 1928
(Liam Mitchell died in Boston) starts at 1941
(Emma Scott died in Oceanview) starts at 1984

Question: True or false: event (Liam Mitchell was married to Maddox Reynolds) was longer in duration than event (Emma Scott was married to Liam Mitchell)?

Useful information:
1885 before 1893 before 1916 before 1918 before 1922 before 1928 before 1941

1918 - 1916 = 2
1928 - 1893 = 35
1928 - 1922 = 6
1941 - 1918 = 23
2 < 6 < 23 < 35

Answer: Let's think step by step.

To determine whether the statement is true or false, we need to compare the durations of (Liam Mitchell was married to Maddox Reynolds) and (Emma Scott was married to Liam Mitchell) . The duration for each event can be calculated as follows:

(Liam Mitchell was married to Maddox Reynolds) :
(Liam Mitchell was married to Maddox Reynolds) starts at 1922
(Liam Mitchell was married to Maddox Reynolds) ends at 1928
1928 - 1922 = 6

(Emma Scott was married to Liam Mitchell) :
(Emma Scott was married to Liam Mitchell) starts at 1916
(Emma Scott was married to Liam Mitchell) ends at 1918
1918 - 1916 = 2

Now, we compare the duration of the two events: 6 is greater than 2 , thus, the answer is True .

Test:
Timeline:
(Chris Evans was born in Bristol, Connecticut) starts at 1948
(Oliver Jensen was born in Bristol, Connecticut) starts at 1948
(Chris Evans was married to Oliver Jensen) starts at 1970
(Oliver Jensen was married to Chris Evans) starts at 1970
(Chris Evans created Pearl Network) starts at 2005
(Chris Evans won prize Victory Achievement Award) starts at 2007
(Chris Evans was married to Oliver Jensen) ends at 2010
(Oliver Jensen was married to Chris Evans) ends at 2010
(Chris Evans created Pearl Network) ends at 2013

Question: True or false: event (Chris Evans owned Pearl Network) was longer in duration than event (Oliver Jensen was married to Chris Evans)?

Useful information:
1948 before 1970 before 2005 before 2007 before 2010
1970 - 1948 = 22
2007 - 1948 = 59
2007 - 2005 = 2
2010 - 1970 = 40
2 < 22 < 40 < 59

Answer: Let's think step by step.

Table 17: Example of the CoT bootstrapping in TGQA.