CHAIN-OF-TIMELINE: ENHANCING LLM ZERO-SHOT TEMPORAL REASONING WITH SQL-STYLE TIMELINE FORMALIZATION

Jiaying Wu, Bryan Hooi
National University of Singapore
jiayingwu@u.nus.edu, bhooi@comp.nus.edu.sg

ABSTRACT

Accurate reasoning about time-sensitive facts is essential in today's ever-evolving knowledge landscape. While Large Language Models (LLMs) possess impressive reasoning capabilities, they struggle with time-sensitive question answering (QA) in long documents due to the presence of (1) *irrelevant noisy context* and (2) *implicit expressions of temporal events*. To address these, we introduce Chain-of-Timeline (CoTime), a framework that constructs topic-relevant event timelines through structured code-style formalization. CoTime first extracts a high-level topic from the question (e.g., [subject]'s career history) to identify relevant temporal events in the document. These events are then organized into a temporal SQL-style schema, enabling CoTime to derive answers based on the question's specified time identifiers. Experiments on two datasets demonstrate CoTime's effectiveness.

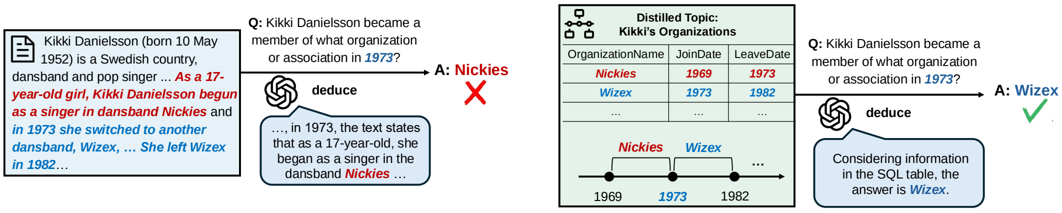
1 INTRODUCTION

The ability to process time is fundamental to how individuals perceive and understand the world (Robinson et al., 2019). In today's ever-changing world where knowledge is constantly being updated, it is essential for LLMs to develop similar capabilities in order to accurately reason about time-sensitive facts.

LLMs, despite their strengths in reasoning and planning (Kojima et al., 2022; Hao et al., 2023; Huang & Chang, 2023), face substantial challenges with temporal reasoning in long documents containing time-sensitive information. These documents often contain *irrelevant noisy context* and *implicit temporal expressions* that obscure temporal relationships, leading to inference errors. For instance, as shown in Figure 1(a), a Plan-and-Solve model (Wang et al., 2023a) using a GPT-4 backbone misinterprets the implicit temporal expression "as a 17-year-old girl" as referring to the year "1973" mentioned later in the same sentence. However, as the latter marks the transition to a subsequent event (i.e., "she switched to another dansband"), this misinterpretation results in an incorrect prediction.

Drawing inspiration from how humans reason about temporal events by selecting relevant occurrences and organizing them along a timeline (Helfrich, 2003), we propose the **C**hain-**o**f-**Time**line (CoTime) framework to enhance LLM temporal reasoning without additional training. CoTime addresses the aforementioned challenges by ***(1)*** constructing question-relevant timelines to filter out irrelevant information, and ***(2)*** converting implicit temporal expressions into explicit timestamps, thereby improving the accuracy of time-sensitive inferences.

Building on the capability of LLMs to extract and structure knowledge using code-style schemas (Guo et al., 2023; Li et al., 2024) and the proven utility of SQL for managing temporal data (Jensen & Snodgrass, 1999), CoTime introduces a structured *temporal SQL-style schema* to facilitate LLM temporal reasoning. The framework consists of three key phases: ***(1)*** topic distillation, ***(2)*** code-style timeline formalization, and ***(3)*** answer deduction. CoTime first distills a high-level topic from the question (e.g., "[subject]'s career history"). It then constructs a topic-focused timeline by formalizing the topic's subject and relation as attributes in a temporal SQL-style table schema, updating it with relevant temporal events within the document. This structured formalization eliminates irrelevant context, facilitating clearer representation of temporal facts. Finally, CoTime deduces the answer by identifying the table entry that aligns with the time specifiers specified in the question.



Kikki Danielsson (born 10 May 1952) is a Swedish country, dansband and pop singer ... *As a 17-year-old girl, Kikki Danielsson begun as a singer in dansband Nickies* and *in 1973 she switched to another dansband, Wizex, ... She left Wizex in 1982*...

…, in 1973, the text states that as a 17-year-old, she began as a singer in the dansband Nickies …

Distilled Topic: Kikki's Organizations

OrganizationName	JoinDate	LeaveDate
Nickies	1969	1973
Wizex	1973	1982
...	...	...

Considering information in the SQL table, the answer is *Wizex*.

Q: Kikki Danielsson became a member of what organization or association in 1973?
A: Nickies
✗
deduce
Q: Kikki Danielsson became a member of what organization or association in 1973?
A: Wizex
✓
deduce
Nickies
Wizex
...
1969
1973
1982

(a) LLM Reasoning

(b) Chain-of-Timeline (ours)

Figure 1: A motivating example of our Chain-of-Timeline (CoTime) framework for time-sensitive QA. In (a), the baseline GPT-4 reasoning model misinterprets the temporal expression "17-year-old", incorrectly associating it with 1973 instead of the correct year, 1969, based on the subject's birth date.

CoTime enables accurate temporal reasoning in a zero-shot, training-free setting and can be seamlessly integrated with any black-box LLM. Experiments on two datasets from the TIMEQA benchmark demonstrate the effectiveness of CoTime.

2 METHODOLOGY

Problem Formulation Let q be a time-sensitive question constrained by time specifier(s) (e.g., "[subject] joined which organization in 1973?") and D be a long document containing various temporal expressions (e.g., "as a 17-year-old girl, [subject] joined ..."). The goal of time-sensitive QA is to deduce the correct answer A to the question q by analyzing the document D.

As overviewed in Figure 1, by obtaining clear, relevant temporal expressions, CoTime facilitates LLM temporal reasoning through three key phases: ***(1)*** distilling a high-level topic from the question to focus on topic-relevant context; ***(2)*** constructing a topic-focused timeline through code-style formalization for clear, explicit temporal expressions; and ***(3)*** deducing the answer through temporally grounded inference.

Topic Distillation The document D contains a large amount of noisy information irrelevant to the question q, which can mislead the reasoning process. To focus on the question-relevant context, we utilize an LLM, denoted as $\mathcal{M}_{\text{distill}}$, to distill a high level topic x (e.g., "[subject]'s career history") underlying q:

$$x = \mathcal{M}_{\text{distill}}(q). \tag{1}$$

Topic x identifies the subject s_x and relation r_x (e.g., "joined organization") of interest to answering the question, enabling the construction of a topic-focused event timeline.

SQL-Style Timeline Formalization Leveraging the subject and relation attributes in x, we construct a temporal SQL-style table schema $\mathcal{T}$ with an LLM $\mathcal{M}_{\text{formalize}}$ to store topic-relevant information in a structured and unified format. By filtering through document D, we update $\mathcal{T}$ by inserting a set $\mathcal{E}_x$ of relevant temporal events, defined as:

$$\mathcal{E}_x = \{(s_x, r_x, o_i, t_i)\}_{i \in \{1,2,\cdots,n\}}, \tag{2}$$

where o_i, and t_i represnt the object and time specifier of event $e_i \in \mathcal{E}_x$, respectively, and n is the number of relevant temporal events in the document.

For each event in $\mathcal{E}_x$, we extract the relevant attributes (e.g., timestamp, object) and insert them as a new entry into the table schema $\mathcal{T}$. This updating process ensures that $\mathcal{T}$ is only populated with events directly related to the high-level topic defined by s_x and r_x, enabling effective inference for answering the time-sensitive question q.

$$\mathcal{T} = \mathcal{M}_{\text{formalize}}(D, \mathcal{E}_x). \tag{3}$$

Answer Deduction To answer the time-sensitive question q based on temporal facts, we employ an LLM denoted as $\mathcal{M}_{\text{deduce}}$ to analyze the formalized timeline in the SQL-style table schema $\mathcal{T}$ and produce the final predicted answer A':

$$A' = \mathcal{M}_{\text{deduce}}(\mathcal{T}, q). \tag{4}$$

Table 1: Exact match (EM) and token-level F1 scores (%) of CoTime and three baseline categories (C1-C3) on two benchmark datasets. Bold values indicate the best overall performance, and underlined values indicate the best baseline performance. (C1: without context; C2: with top-3 retrieved chunks from Wikipedia articles as context; C3: with full Wikipedia articles as context.)

	Dataset →	**TimeQA-Easy**				**TimeQA-Hard**			
	Base LLM →	GPT-4		Claude-3		GPT-4		Claude-3	
	Method ↓	EM	F1	EM	F1	EM	F1	EM	F1
C1	Closebook QA	24.4	38.7	12.3	21.2	19.1	32.4	10.5	18.9
C2	QAaP 3-shot (Zhu et al., 2023)	45.7	56.9	41.9	50.7	35.0	45.8	34.8	43.6
C3	Direct Prompting	51.9	63.2	48.4	60.2	44.5	53.6	41.6	50.5
	CoT (Wei et al., 2022)	51.0	63.3	48.9	60.7	44.2	54.1	40.3	50.1
	Plan-and-Solve (Wang et al., 2023a)	52.4	64.5	50.8	61.6	45.9	56.0	42.1	51.9
	Step-Back (Zheng et al., 2024)	53.1	65.0	50.4	62.1	46.4	56.9	42.8	52.7
Ours	CoTime w/ Top-3 Retrieved Chunks	54.6	65.5	52.2	62.1	48.4	56.8	44.7	53.6
	CoTime	**55.5**	**66.3**	**52.5**	**63.6**	**48.9**	**58.9**	**45.0**	**55.7**

3 Experiments

3.1 Experimental Setup

Datasets We conduct experiments using two datasets from TimeQA (Chen et al., 2021) benchmark for time-sensitive QA: ***(1)*** TimeQA-Easy, with 2,997 document-question pairs, and ***(2)*** TimeQA-Hard, with 3,078 pairs. The documents are Wikipedia articles related to the subjects of the questions. The Easy dataset contains more explicit mentions of the queried timestamps, while the Hard dataset includes more implicit references, requiring stronger temporal reasoning capabilities.

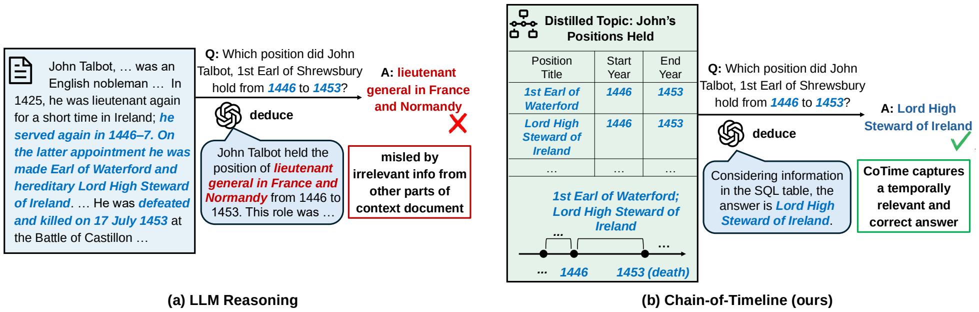
Baselines We implement six representative baselines: **Closebook QA**, **Direct Prompting**, **QAaP** (Zhu et al., 2023), **CoT** (Wei et al., 2022), **Plan-and-Solve** (Wang et al., 2023a), and **Step-Back** (Zheng et al., 2024). See details in Appendix C.1. QAaP relies on in-context demonstrations to guide the reasoning process; thus, we include 3-shot time-sensitive QA demonstrations, consistent with the original setup. As CoTime is designed as a training-free approach, we exclude fine-tuning methods like TG-LLM (Xiong et al., 2024) and MTGER (Chu et al., 2023) for a fair comparison.

Implementation and Evaluation We adopt GPT-4 and Claude-3 as the LLM backbones for CoTime and all baseline methods. For retrieval-based approaches, the chunk size is set to 512 tokens. Following prior work, we evaluate model performance using exact match (EM) and token-level F1 metrics. Detailed configurations and prompts are provided in Appendix C.2 and D.

3.2 Result Analysis

CoTime uses full articles as context. We also implement a retrieval-based variant of CoTime, using the top-3 retrieved chunks from Wikipedia articles (i.e., consistent with the retrieval approach in QAaP). From Table 1, we observe: ***(1)*** QAaP retrieves relevant information but underperforms compared to C3 baselines using full context, likely due to chunking-induced information loss. ***(2)*** Among C3 baselines, CoT lags behind Plan-and-Solve and Step-Back in temporal reasoning, suggesting the benefits of structured reasoning and planning. ***(3)*** CoTime achieves the best performance, surpassing standard CoT by 4.38% EM and 4.08% F1, averaged across datasets and base LLMs. Notably, its retrieval-based variant, using partial document chunks, also outperforms baselines in most cases. Additional investigations in Appendix A and Appendix B show that CoTime ***(4)*** achieves reasonable efficiency and ***(5)*** proves effective on cutting-edge reasoning models such as o1 (OpenAI, 2024b), further validating its adaptability. By leveraging SQL-style timeline formulation, CoTime explicitly structures topic-focused temporal events, enhancing LLM reasoning in time-sensitive QA.

Table 2 measures the contributions of CoTime's key components. Both topic distillation and SQL-style formalization significantly enhance temporal reasoning. Notably, even without topic distillation, CoTime outperforms standard CoT, confirming the effectiveness of topic-focused SQL-style timelines.



John Talbot, … was an English nobleman … In 1425, he was lieutenant again for a short time in Ireland; *he served again in 1446–7. On the latter appointment he was made Earl of Waterford and hereditary Lord High Steward of Ireland*. … He was *defeated and killed on 17 July 1453* at the Battle of Castillon …

Q: Which position did John Talbot, 1st Earl of Shrewsbury hold from *1446* to *1453*?

A: lieutenant general in France and Normandy ✗

John Talbot held the position of *lieutenant general in France and Normandy* from 1446 to 1453. This role was …

misled by irrelevant info from other parts of context document

(a) LLM Reasoning

Distilled Topic: John's Positions Held

Position Title	Start Year	End Year
1st Earl of Waterford	*1446*	*1453*
Lord High Steward of Ireland	*1446*	*1453*
…	…	…

Q: Which position did John Talbot, 1st Earl of Shrewsbury hold from *1446* to *1453*?

A: Lord High Steward of Ireland ✓

Considering information in the SQL table, the answer is *Lord High Steward of Ireland*.

CoTime captures a temporally relevant and correct answer

(b) Chain-of-Timeline (ours)

Figure 2: A success case of CoTime in filtering out irrelevant information and utilizing relevant information to deduce the correct answer. In contrast, the GPT-4 reasoning baseline mistakenly captures irrelevant details and arrives at an incorrect prediction.

Case Study Figure 1 illustrates CoTime's ability to accurately interpret implicit temporal expressions by constructing an SQL-style timeline of events. To further highlight the benefits of topic-focused reasoning, we present another successful case in Figure 2. The document implies the subject's position during the queried timeframe (1446–1453), noting their acquisition of the title "Lord High Steward of Ireland" in 1446–47 and its continuation until their death in 1453. However, the GPT-4-based Plan-and-Solve model erroneously prioritizes temporally irrelevant details about the subject's appointment as "lieutenant general in France and Normandy", neglecting the relevant information and leading to an incorrect prediction. In contrast, CoTime accurately extracts and utilizes the pertinent details within the timeframe, yielding the correct answer.

Table 2: Ablation of CoTime with GPT-4 demonstrates the benefits of topic distillation (T) and code-style formalization (C), reflected in F1 (%).

Method	TIMEQA-EASY	TIMEQA-HARD
CoTime	**66.3**	**58.9**
w/o T	64.4	55.5
w/o C	64.9	56.3

4 RELATED WORK

Time-Sensitive QA Existing efforts typically model temporal spans and relationships by fine-tuning language models with time-related objectives (Rosin et al., 2022; Tan et al., 2023b;a) or training on explicit temporal graphs constructed on the document (Chu et al., 2023; Xiong et al., 2024). Closely related to CoTime, information extraction methods QAaP (Zhu et al., 2023) and TempLogic (Li et al., 2023c) parse temporal events without training but lack explicit mechanisms to filter question-irrelevant information, which can lead to noisy reasoning under long contexts. TG-LLM (Xiong et al., 2024) employs a two-stage fine-tuning process to construct and reason over temporal graphs (TGs), yet it requires manual intervention between stages to ensure TG and reasoning quality. CoTime addresses these with a fully automated, training-free approach that formalizes a topic-focused timeline with flexible schema initialization, ensuring adaptability and scalability for time-sensitive QA.

LLMs for Information Extraction (IE) LLMs excel in extracting structured knowledge from natural language (Fei et al., 2022; Dyer, 2023; Xu et al., 2024). Specifically, recent approaches has validated the effectiveness in using code to represent and define knowledge under various schemas (Guo et al., 2023; Li et al., 2024; Sainz et al., 2024). However, existing work existing work typically address IE tasks under non-temporal scenarios, such as event argument extraction (Wang et al., 2023b) relation extraction (Li et al., 2023a; Zhang et al., 2023) named entity recognition (Li et al., 2023b). In this work, we investigate the effectiveness of applying LLMs to construct SQL-style formalized timelines for temporal reasoning.

5 CONCLUSION

In this paper, we present CoTime, a framework that enhances LLM temporal reasoning by constructing formalized, SQL-style timelines of events. Through topic distillation and timeline formalization, CoTime extracts relevant events and converts implicit temporal expressions into explicit ones, enabling effective time-sensitive QA in a zero-shot setting. Both quantitative and qualitative results on two datasets demonstrate CoTime's effectiveness.

REFERENCES

Anthropic. The claude 3 model family: Opus, sonnet, haiku. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf, 2024.

Wenhu Chen, Xinyi Wang, and William Yang Wang. A dataset for answering time-sensitive questions. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Zheng Chu, Zekun Wang, Jiafeng Liang, Ming Liu, and Bing Qin. MTGER: Multi-view temporal graph enhanced temporal reasoning over time-involved document. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15218–15233, 2023.

Andrew Thomas Dyer. Revisiting dependency length and intervener complexity minimisation on a parallel corpus in 35 languages. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pp. 110–119, 2023.

Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, Fei Li, Libo Qin, Meishan Zhang, Min Zhang, and Tat-Seng Chua. Lasuie: Unifying information extraction with latent adaptive structure-aware generative language model. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15460–15475, 2022.

Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, and Xueqi Cheng. Retrieval-augmented code generation for universal information extraction, 2023.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173, 2023.

Hede Ed Helfrich. *Time and mind II: Information processing perspectives.* Hogrefe & Huber Publishers, 2003.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1049–1065, 2023.

Christian S. Jensen and Richard Thomas Snodgrass. Temporal data management. *IEEE Trans. on Knowl. and Data Eng.*, 11(1):36–44, 1999.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, 2022.

Junpeng Li, Zixia Jia, and Zilong Zheng. Semi-automatic data enhancement for document-level relation extraction with distant supervision from large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5495–5505, December 2023a.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. CodeIE: Large code generation models are better few-shot information extractors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15339–15353, 2023b.

Xingxuan Li, Liying Cheng, Qingyu Tan, Hwee Tou Ng, Shafiq Joty, and Lidong Bing. Unlocking temporal question answering for large language models with tailor-made reasoning logic. *arXiv preprint arXiv:2305.15014*, 2023c.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Xiang Li, Zhilei Hu, Long Bai, Wei Li, Yidan Liu, Pan Yang, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. Knowcoder: Coding structured knowledge into llms for universal information extraction, 2024.

OpenAI. Gpt-4 technical report, 2024a.

OpenAI. Learning to reason with llms, 2024b.

Eva Robinson, Kelly Michaelis, James C. Thompson, and Martin Wiener. Temporal and spatial discounting are distinct in humans. *Cognition*, 190:212–220, 2019.

Guy D. Rosin, Ido Guy, and Kira Radinsky. Time masking for temporal language models. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 833–841, 2022.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. GoLLIE: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations*, 2024.

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

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

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2609–2634, 2023a.

Xingyao Wang, Sha Li, and Heng Ji. Code4Struct: Code generation for few-shot event structure prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3640–3663, Toronto, Canada, 2023b.

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

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. Large language models can learn temporal reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10452–10470, 2024.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. Large language models for generative information extraction: A survey, 2024.

Ruoyu Zhang, Yanzeng Li, Yongliang Ma, Ming Zhou, and Lei Zou. LLMaAA: Making large language models as active annotators. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 13088–13103, 2023.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Xinyu Zhu, Cheng Yang, Bei Chen, Siheng Li, Jian-Guang Lou, and Yujiu Yang. Question answering as programming for solving time-sensitive questions. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12775–12790, 2023.

A TOKEN EFFICIENCY OF COTIME

To assess CoTime's efficiency, we benchmark its effectiveness and token consumption against three representative training-free baselines that utilize the full context: Direct Prompting, CoT (Wei et al., 2022), and Plan-and-Solve (Wang et al., 2023a).

As shown in Table 3, while the retrieval-based CoTime variant is less effective than the full CoTime model due to information loss from arbitrary chunking, it still outperforms standard CoT by 2.2% while using 0.16k fewer tokens per sample. Although CoTime requires more tokens due to topic distillation and timeline formalization, Table 1 shows that it consistently achieves superior performance compared to its retrieval-based variant and all baselines. This suggests the advantage of leveraging complete documents to construct comprehensive timelines and maintain long-term dependencies.

Table 3: Token efficiency comparison between CoTime and baseline reasoning approaches on the TIMEQA-EASY dataset, with average token consumption calculated over the first 200 samples. The F1 score (%) represents overall performance on the full dataset. All methods use the same GPT-4 backbone.

Method	# Tokens	F1
Direct Prompting	2.61k	63.2
CoT	2.64k	63.3
Plan-and-Solve	2.82k	64.5
CoTime w/ Top-3 Retrieved Chunks	2.48k	65.5
CoTime	3.73k	66.3

B EFFECTS OF LARGE REASONING MODELS (LRMS) ON LONG-CONTEXT TIME-SENSITIVE QA

Recent advances in LRMs, such as OpenAI's o1 (OpenAI, 2024b) and o3-mini, represent a significant step forward in enhancing the reasoning capabilities of LLMs. This section investigates the following research question: **How effective are LRMs in time-sensitive QA, and how does CoTime perform when built upon LRMs?**

To answer this, we implement CoTime using two representative LRMs: o1 (`o1-2024-12-17`) and o3-mini (`o3-mini-2025-01-31`). The results in Table 4 yield three key insights: ***(1)*** Across Direct Prompting (DP) approaches, **LRMs significantly improve time-sensitive QA performance** compared to GPT-4. ***(2)*** While the performance gap between DP and CoTime narrows due to the enhanced reasoning capabilities of LRMs – partially addressing the limitations in base LLMs that CoTime was designed to mitigate – **CoTime consistently outperforms DP on LRMs**. ***(3)*** Despite their superior reasoning abilities, LRM performance still reflects limitations in time-sensitive QA, highlighting the need for further targeted research.

Table 4: While large reasoning models exhibit impressive capabilities that help bridge the gap between Direct Prompting (DP) and our proposed CoTime approach, CoTime still consistently outperforms DP. The o1 and o3-mini experiments are conducted on the first 1,000 instances of both datasets as a proof of concept, without loss of generality.

Method	TIMEQA-EASY	TIMEQA-HARD
GPT-4-DP	63.2	53.6
GPT-4-CoTime	**66.3**	**58.9**
o1-DP	66.1	58.2
o1-CoTime	**66.9**	**59.3**
o3-mini-DP	67.2	59.1
o3-mini-CoTime	**68.4**	**60.6**

C Implementation Details

In this section, we describe the baselines adopted in Section 3.1 and Table 1, and present the LLM configuration details for reproducibility.

C.1 Baselines

We consider six representative baselines in the following three categories. For a fair comparison, **all baselines except QAaP (Zhu et al., 2023) are implemented without any in-context demonstrations on time-sensitive QA**, in line with our proposed CoTime framework. The LLM configurations for CoTime and baselines are detailed in Appendix C.2.

C.1.1 C1: Without Context

To assess the capabilities of LLMs to answer time-sensitive questions based solely on their internal knowledge, we implement **Closebook QA** with the following prompt:

```
Only answer with the exact answer tokens.
Question: [question]
Answer: [predicted answer]
```

C.1.2 C2: With Retrieved Context Chunks

To measure the effects of incorporating external knowledge, we implement **QAaP** (Zhu et al., 2023), a representative retrieval-augmented approach for time-sensitive QA. Specifically, QAaP extracts the queried subject to retrieve the top relevant chunks from the subject's Wikipedia article, and identifies the temporal event in the context with the highest time overlap. We follow the official QAaP implementation available at `https://github.com/TianHongZXY/qaap`, and present three in-context time-sensitive QA demonstrations, as the method requires in-context learning.

C.1.3 C3: With Complete Context

To evaluate the effects of providing LLM-based QA models with gold context, we implement the following four representative baselines:

- **Direct Prompting**, which adopts the following instruction:

```
Answer the question based on a relevant context document. Only output the exact answer tokens.
Context: [context document]
Question: [question]
Answer: [predicted answer]
```

- **Chain-of-Thought (CoT)** (Wei et al., 2022), which prompts the LLM to generate a sequence of intermediate reasoning steps to derive the final answer.
- **Plan-and-Solve** (Wang et al., 2023a), which requires the LLM to first devise a comprehensive plan for solving the given question and then follow the plan step-by-step.
- **Step-Back** (Zheng et al., 2024), which abstracts the given detailed question into a high-level question that is easier to answer, and deduces the final answer based on the intermediate response. Step-Back is given 4-shot demonstrations on question abstraction.

C.2 LLM Configuration

We select two representative LLMs as backbones for time-sensitive QA models, namely GPT-4 (OpenAI, 2024a) (model version name: `gpt-4-turbo-2024-04-09`) and Claude-3 (Anthropic, 2024) (model version name: `claude-3-haiku-20240307`). Due to the extensive length of Wikipedia documents in TimeQA, typically comprising thousands of tokens, we exclude Claude-3-opus from consideration due to its strict daily token limits.

We utilize the GPT-4 API from OpenAI and the Claude-3 API from Anthropic, setting the temperature to 0 for stable answer predictions. During evaluation, since the generated responses can include extraneous tokens (e.g., “Based on the context, the answer is ...”), we extract the exact answer as a word or short phrase using a GPT-4 model. Based on this extraction, we compute the Exact Match (EM) and token-level F1 scores. The following prompt is used to extract the exact answer:

> Given a question and its corresponding response, your task is to extract the exact answer from the response. The extracted answer should be a substring of the response, represented as a short word or phrase.
> Question: [question]
> Response: [generated response]
> Extracted answer: [exact answer]

D CoTime Prompts

In Table 5, we provide a detailed illustration of CoTime’s temporal reasoning process to enhance clarity and comprehension.

Table 5: An illustration of CoTime temporal reasoning process.

Topic Distillation
You are an expert at world knowledge. Your task is to paraphrase a question to a high-level topic. Here are a few examples: Input: “What position did Gordon Brown take from Jul 1987 to Nov 1989?” Output: “Which positions have Gordon Brown held in his career?” Input: “Who was Rita Hayworth’s spouse from 1958 to 1961?” Output: “Who were the spouses of Rita Hayworth?” Input: “Jacob Timpano played for which team from 2005 to 2009?” Output: “Which teams did Jacob Timpano play for in his career?” Input: “What was the operator of GCR Class 8B from 1948 to 1950?” Output: “What were the operators of GCR Class 8B in history?” Input: [time-sensitive question] Output:
Generated Response: [Distilled question topic]
Code-Style Timeline Formalization
Only output the SQL expressions. Construct an SQL table to store the following time-related topic: [distilled question topic]
Generated Response: [Empty temporal SQL table]
Given the following context, extract all time-related events that align with the tableś keys. Then, update the table with the time-related events accordingly. Context: [context document]
Generated Response: [Updated temporal SQL table]
Answer Deduction
Translate the following question into an SQL query: [time-sensitive question]
Generated Response: [Question-converted SQL query]
Based on the information provided earlier in the table, deduce the answer to the query. Only output the deduced answer.
Generated Response: [Predicted answer]

E Limitations

We acknowledge the following limitations of our work. ***(1)*** **Training-free design.** CoTime is designed as a training-free approach, offering flexibility to work with any LLM, including black-box models, similar to CoT (Wei et al., 2022) and Plan-and-Solve (Wang et al., 2023a). While SQL-style timelines aid temporal reasoning, even Large Reasoning Models (LRMs) such as o1 (OpenAI, 2024b) struggle with zero-shot temporal reasoning, necessitating intensive manual efforts to refine automatically generated SQL-style timelines into gold references. Future work could focus on curating high-quality SQL-style timelines and exploring fine-tuning on these timelines to enhance performance. ***(2)* Dataset scope.** CoTime was evaluated on two distinct datasets within TimeQA, the only benchmark featuring long-sequence contexts (Wikipedia articles) with ground-truth answers. Other benchmarks, such as TempReason (Tan et al., 2023b), rely on structured knowledge bases such as Wikidata, which lack such textual contexts. Extending CoTime to Wikidata-style datasets remains an avenue for future research. ***(3)* Language diversity.** Due to the limited availability of document-level, time-sensitive QA datasets with gold contexts, our experiments were conducted on English datasets. We encourage future work to explore SQL-style timeline formalization in multilingual settings.