R2A-TLS: Reflective Retrieval-Augmented Timeline Summarization with Causal-Semantic Integration

Chenlong Bao[1*], **Shijie Li**[1*], **Minghao Hu**[2†],
Ming Qiao [3], **Bin Zhang**[1], **Jintao Tang**[1†], **Shasha Li**[1†], **Ting Wang**[1†]
[1]College of Computer Science and Technology, National University of Defense Technology
[2]Center of Information Research, Academy of Military Science
[3]North China University of Technology
{baochenlong, lishijie, zhangbin2021, tangjintao, shashali, tingwang}@nudt.edu.cn
{shawyh, ncut.qiaoming}@gmail.com

Abstract

Open-domain timeline summarization (TLS) faces challenges from information overload and data sparsity when processing large-scale textual streams. Existing methods struggle to capture coherent event narratives due to fragmented descriptions and often accumulate noise through iterative retrieval strategies that lack effective relevance evaluation. This paper proposes **R2A-TLS**: Reflective Retrieval-Augmented Timeline Summarization with Causal-Semantic Integration, which offers a novel perspective for open-domain TLS by time point completion and event element completion. R2A-TLS establishes an initial retrieval, reflection, and deep retrieval system that reduces noise through a double filtering mechanism that iteratively generates a timeline for each text which passes the filtering. Then, the system reflects on the initial timeline with the aim of identifying information gaps through causal chain analysis and FrameNet based element validation. These gaps are reformulated into targeted queries to trigger deep retrieval for refining timeline coherence and density. Empirical evaluation on Open-TLS dataset reveals that our approach outperforms the best prior published approaches.[1]

1 Introduction

Condensing a vast collection of texts into comprehensible summaries is a crucial and challenging task. Timeline Summarization (TLS) aims to distill large-scale text collections into structured summaries by extracting and chronologically organizing key events(Yan et al., 2011). Current TLS research is generally divided into open-domain and closed-domain paradigms(Wu et al., 2025). Open-domain TLS refers to the process of generating timelines from news directly searched and retrieved from the Internet. Recent advancements in Large Language Models (LLMs) have significantly improved closed-domain TLS by enabling precise event clustering(Hu et al., 2024). However, the information overload and data sparsity make it difficult to retrieve relevant and comprehensive information from the Internet. As illustrated in Figure 1, these challenges are amplified by the noisy nature of news streams. And the data sparsity constrains the quality of the timeline, manifesting as missing time points or missing details.

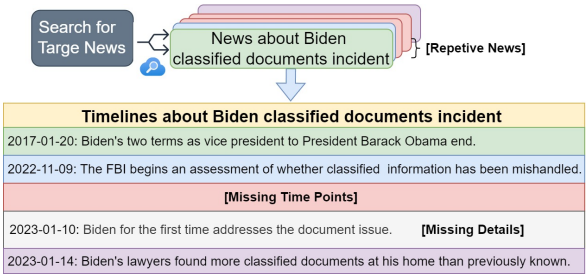

Figure 1: Triple Threat in Open-Domain TLS: Redundant Retrieval, Time Points Absence, and Incomplete Event Elements.

The inherent information overload and data sparsity in open-domain TLS require modeling temporal and causal dependencies across events to counteract narrative fragmentation. Such fundamental trade-offs expose critical limitations in conventional pipeline approaches that separately handle temporal modeling and event extraction. LLM-driven approaches establish temporal and causal relationships between events through the self-questioning iterations and refresh chronological summaries based on documents retrieved in each round(Wu et al., 2025). However, the effectiveness of this approach is inherently constrained by the number of self-questioning iterations and noise characteristics of open-sourced data. While additional rounds can theoretically enhance reason-

[*]These authors contribute equally to this work.
[†]Corresponding author.
[1]The codes are released at https://github.com/DocBao/R2A-TLS.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 752–766
November 4-9, 2025

ing depth, they simultaneously introduce scalability challenges through exponential growth in retrieved news articles, significantly complicating temporal integration during timeline construction.

To address these challenges, we consider reducing noise to tackle information overload, and optimizing the iterative questioning method to deal with data sparsity. Regarding noise reduction and reducing inference demands, we can filter articles, generate timeline fragments for each useful article, and finally merge and deduplicate. Humans can infer missing time points or identify incomplete event descriptions in timelines through commonsense reasoning, causal logic, and domain knowledge. In recent researches, the generic LLM has been trained on extensive corpora and has mastered common sense and causal logic(Liu et al., 2025; Gao et al., 2023), as for domain knowledge, it can be enhanced by fine-tuning the model or introducing an external knowledge base(Jin et al., 2024). Inspired by this sight, we can use the generic LLM and fine-tune a domain-adapted model to jointly identify the missing time point, which involves causal chains analysis and domain-informed pattern matching, and use FrameNet(Boas et al., 2024) to guide the LLM to discover incomplete descriptions.

Motivated by the above, we propose **R2A-TLS**: Reflective Retrieval-Augmented Timeline Summarization with Causal-Semantic Integration, which introduces a novel perspective for modeling event relationships in open-domain TLS by performing time point completion and event element completion through reflection and deep retrieval. R2A-TLS establishes an initial retrieval, reflection, and deep retrieval system, adding a dual-filtering mechanism that applies topical relevance judgment and information gain assessment to process retrieved texts. We use LLMs to generate timeline fragments for each texts that passed the filter and merge them to get an initial timeline. Then, by synergizing LLM-powered causal reasoning with FrameNet-based element validation, the system performs reflection to detect information gaps from the initial timeline. These gaps are then reformulated into targeted queries to trigger deep retrieval to complete the timeline.

Our contributions can be summarized as follows:

- We propose **R2A-TLS**, which is the first closed-loop system of "retrieval reflection deep retrieval", to solve the redundant interference caused by information overload.
- We introduce a novel perspective for modeling event relations in open-domain TLS. Through causal chain analysis and semantic framework driven core element completion, the problems of missing time points or details caused by data sparsity have been mitigated.
- Experiments demonstrate that our method achieves state-of-the-art (SOTA) results, with a 5.1% absolute improvement in Date F1 and a 2.8% gain in Align F1 over the previous best-performing approach.

2 Related Work

2.1 Timeline Summarization

Timeline summarization (TLS) aims to extract and organize key events along a temporal axis, traditionally relying on extractive methods that prioritize temporally coherent events through textual features and chronological reasoning(Allan et al., 2001; Yu et al., 2021). Recent advances in LLMs have shifted researchers focus on leveraging LLMs for TLS, enabling flexible narrative construction(Hu et al., 2024; Qorib et al., 2024). Notable examples include the CHRONOS framework(Wu et al., 2025), which decomposes large-scale timelines into subtasks via iterative questioning, achieving efficiency gains in both open-domain and closed-domain settings. Extensions such as Dynamic Granularity TLS (DTELS) and Hierachical VAEs TLS (TH-VAE) further address granular control and cross-entity interaction modeling(Zhang et al., 2025; Song et al., 2024). However, existing methods often struggle with causal chain breakage and event element incompleteness in open-domain scenarios due to information overload or sparse data.

2.2 Retrieval-Augmented Generation and Reflective Mechanisms

To mitigate information overload in open-domain TLS(Sheng et al., 2020), retrieval-augmented frameworks have emerged as critical solutions. The CHRONOS employs an "iterative self-questioning" mechanism to progressively expand timelines but relies heavily on initial retrieval quality without systematic reflection modules. Recent works integrate causal inference and symbolic logic to refine information filtering, such as identifying event gaps via causal chain analysis or validating semantic completeness through knowledge graphs(Kneale et al.,

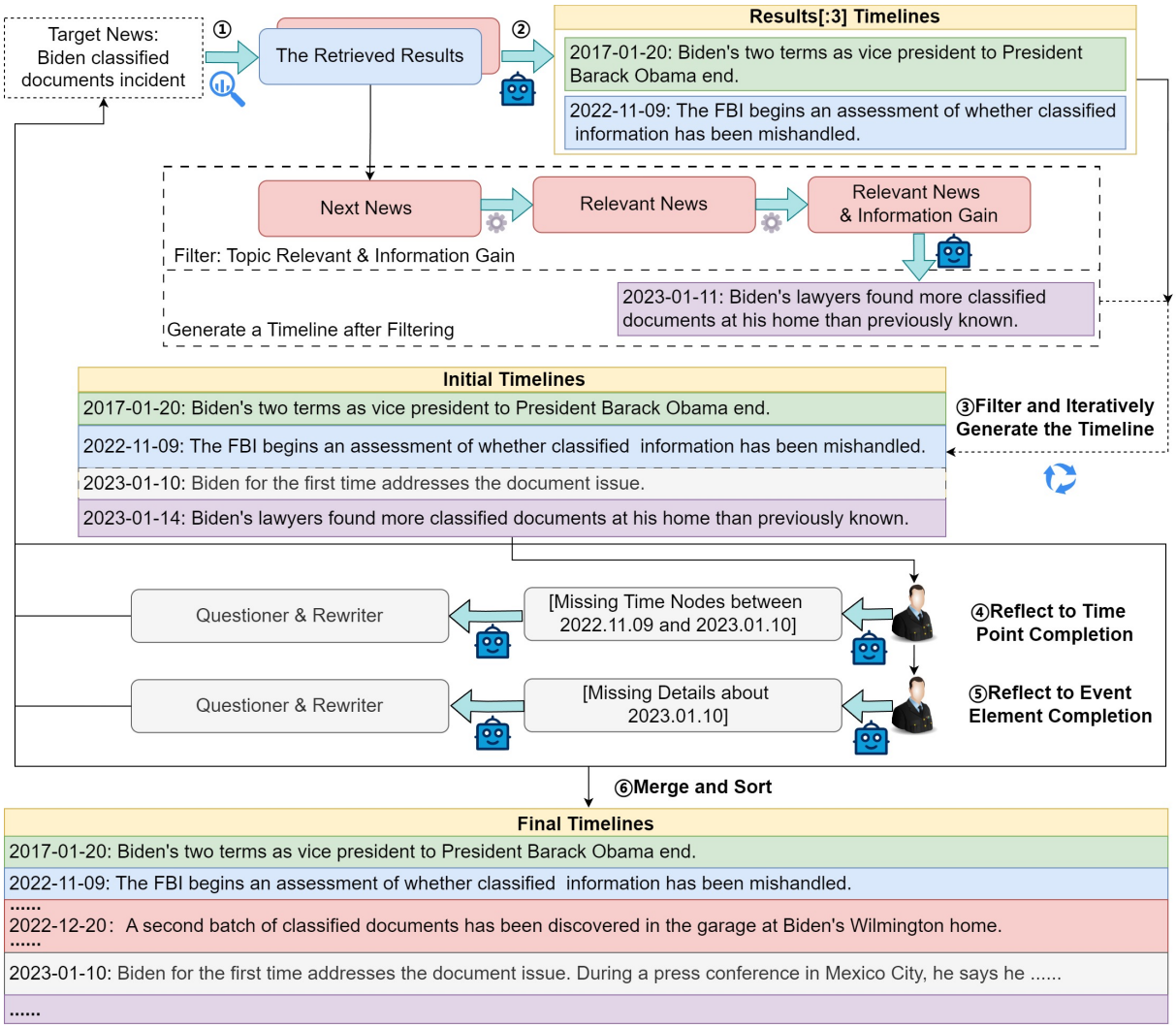

Figure 2: An overview of the R2A-TLS pipeline. Giving a target news, it first searches for it and iteratively generates timelines to compose an initial timeline. Then, reflect to time point completion and event element completion, including find missing time nodes or details, generate and rewrite questions, deep retrieval, filter and iteratively generate the timelines. Final, merge and sort timelines.

2018; Xue et al., 2024; Cui and Chen, 2022). However, these approaches are often confined to closed-domains or require manual rule definitions. LLMs have also been applied to multi-document summarization via "chain-of-thought" reasoning(Wei et al., 2022), yet their stability in cross-lingual or open-domain settings remains unproven.

2.3 Frame Semantics for Event Representation

FrameNet-based semantic analysis has proven effective in structuring event representations by mapping textual content to predefined semantic frames(Li et al., 2019a). While prior studies use FrameNet for argument filling or relation classification(Li et al., 2019b), its integration with TLS remains underexplored(Ponkiya et al., 2021). Neural models like BERT implicitly encode event structures but often fail to address core element gaps caused by data sparsity in open-domain scenarios(Wang et al., 2019). Furthermore, causal disconnections and event element incompleteness frequently coexist in timelines, yet existing methods lack unified mechanisms to address both challenges synergistically.

3 Methodology

We propose **R2A-TLS**, a novel approach that leverages LLMs and performs reflective completion based on causal chains and frame semantics to address the challenges in open-domain TLS. As shown in Figure 2, our method involves a six-step process: Retrieval, Generate Timelines for Top3 Results, Filter and Iteratively Generate the Time-

line, Reflect to Time Point Completion, Reflect to Event Element Completion, Merge and Sort. Steps 1, 2, 3 and 6 will be elaborated in detail in Section 3.1. Steps 4 and 5 will be introduced in Section 3.2 and Section 3.3 respectively.

3.1 Iterative Timeline Generation via Dual-Filtering Mechanisms

Open-domain TLS faces the problem of information overload, and directly generating timelines for all retrieved content increases the inference burden of LLM, and the retrieved content is not filtered, which introduces noise. Therefore, we search for the target news first, use the top3 results as a seed set to generate timelines, then use a dual-filtering mechanism to filter the remaining articles, generate timeline fragments for the articles that pass the filtering, and finally merge and sort them.

Retrieval We use a search engine to search using news headlines as keywords to obtain general information that is most directly relevant to the target news. During timeline completion in Steps 4 or 5, the retrieval target shifts from news headlines to rewritten queries.

Generate Timelines for Top3 Results According to the characteristics of search engines, the news with higher search rankings is more relevant to the target news. We use the LLM to generate a timeline summarization for top3 results, as shown in Figure 2, which is an initial timeline for the target news.

Filter Standards

Rules about Topic Relevant:
1. The article is related to the topic;
2.If the article only contains discussion of ideas or statistics without time connection, it will be considered unrelated.

Rules about Information Gain:
1. Discover new time nodes and provide key event progress;
2.Add new details to existing time points;
3.Satisfy at least one.

Figure 3: Filter standards about topic relevant and information gain.

Filter and Iteratively Generate the Timeline Iterate the remaining news in sequence, first determine whether the topic is relevant, and if so, determine whether new information can be introduced. The filtering criteria are shown in Figure 3. Generate a timeline fragment for the news that passes the dual-filtering mechanism, and add it to the initial timeline. Repeat this loop until the iterate is complete. Generation is completed by the LLM based on the prompt template and a timeline fragment contains one or more time points, each with a corresponding summary. Upon loop termination, perform reflection to complete the timeline.

Merge and Sort The final step is to merge the generated timeline fragments to ensure that only the most significant events are retained. The merging process is done by the LLM in response to prompts, including aligning events and resolving conflicts in dates and descriptions. All steps are shown in Algorithm 1.

Algorithm 1 R2A-TLS

Input: Target event E, News corpus $\mathcal{N}$
Output: Event timeline $\mathcal{T}$

$\mathcal{T}_{\text{init}} \leftarrow \emptyset$
$\mathcal{N} \leftarrow \text{Retrieval}(E)$
$\mathcal{T}_{\text{init}} \leftarrow \mathcal{T}_{\text{init}} \cup \text{LLM\_GenTimeline}(\mathcal{N}[:3])$
for $n_j \in \mathcal{N}[4:]$ **do**
  if $\text{RelevanceFilter}(n_j, E) \wedge \text{InfoGainFilter}(n_j, \mathcal{T}_{\text{init}})$ **then**
    $\mathcal{T}_j \leftarrow \text{LLM\_GenTimeline}(n_j)$
    $\mathcal{T}_{\text{init}} \leftarrow \mathcal{T}_{\text{init}} \cup \mathcal{T}_j$
  end if
end for
$\mathcal{G} \leftarrow \text{ReflectAndDeepRetrieval}(\mathcal{T}_{\text{init}})$
while $\mathcal{G} \neq \emptyset$ **do**
  $q \leftarrow \mathcal{G}.\text{pop}()$
  $q' \leftarrow \text{QueryRewriter}(q)$
  for $n_k \in \text{SearchEngine}(q')$ **do**
    if $\text{RelevanceFilter}(n_k, E) \wedge \text{InfoGainFilter}(n_j, \mathcal{T}_{\text{init}})$ **then**
      $\mathcal{T}_k \leftarrow \text{LLM\_GenTimeline}(n_k)$
      $\mathcal{T}_{\text{init}} \leftarrow \mathcal{T}_{\text{init}} \cup \mathcal{T}_k$
    end if
  end for
end while
$\mathcal{T} \leftarrow \text{MergeAndSort}(\mathcal{T}_{\text{init}})$
return $\text{EventTimeline}(\mathcal{T})$

3.2 Time Point Completion Based on Causal Chains

After step 3, the initial timeline is incomplete due to data sparsity. Missing milestone events can lead to a broken causal chain of events, which is reflected in missing time points. Therefore, we use both generic LLMs and a domain-adapted LLM to identify missing time points by reflecting on initial timelines and analyzing causal chains. By identifying the missing time points, relevant queries are generated and rewritten to obtain better search results to complement timelines.

Based on the causal reasoning and common sense, we designed prompt templates to use the generic LLM to locate missing time points by analyzing whether there are breaks in the causal chain in the timeline. Specifically, as shown in Figure 2, **the FBI begins an assessment of whether classified information has been mishandled on 2022-11-09**. However, next time point is **2023-01-10**, **Biden for the first time addresses the document issue**. This causal chain is not coherent, lacking a direct link between the FBI's assessment and Biden's response without an intervening event, so there are missing time points between 2022-11-09 and 2023-01-10.

Furthermore, the domain-adapted LLM detects anomalous temporal jumps that violate domain norms by leveraging instruction tuning to learn domain-specific causal constraints. The model is fine-tuned on a custom training set constructed from three annotated closed-domain timeline summarization datasets T17(Binh Tran et al., 2013), CRISIS (Tran et al., 2015) and ENTITIES (Gholipour Ghalandari and Ifrim, 2020), where each sample comprises: (1) an incomplete timeline of an event; (2) deleted time points; (3) corresponding answers derived from these missing points.

After obtaining two sets of candidate missing time points (from the generic LLM and the domain-adapted LLM), they were entered into the generic LLM for evaluation. The model is asked to rigorously assess whether each missing time point constituted a true causal discontinuity by means of a designed prompt template. Then, the model is used to generate questions that complement the timeline for the validated missing time points, which are then rewritten as targeted query statements. This strategy decomposes complex requirements into semantically precise subqueries to optimize search engine retrieval. For each query, perform step 1 and step 3 to generate timeline fragments and merge them after generation.

3.3 Event Element Completion Based on Frame Semantics

The completion of time points based on causal chains lays the foundation for the chronological continuity of the timeline. However, precisely localized time points alone cannot guarantee comprehensive event representation. This inherent limitation motivates the requirement for the event element completion based on frame semantics. This step is driven by the prompt template, using LLM to examine the timeline under the guidance of FrameNet, which addresses the issue of incomplete element coverage caused by open-domain data sparsity.

The step first employs a LLM to recommend appropriate FrameNet frames for each timeline point's summary. These frames define core event components (e.g., Participants, Motivations, Impacts) through standardized frame elements (FEs). Two operational protocols are enforced: When the LLM generates a frame name not present in the FrameNet corpus, it is classified as an LLM-generated frame, with its contents defined by the model; When the frame name matches an entry in the FrameNet corpus, automatically updates the frame's core elements using the corpus' validated ontological specifications.

Then using the LLM to check if there are core elements missing from the timeline. For missing core elements, we generate questions and rewrite them. For the question after rewriting, first retrieve whether the existing article can be answered, if it cannot be answered, call the search engine for deep retrieval. For newly retrieved articles, perform step 3 to complete the time element completion of the timeline.

4 Experiments

4.1 Experimental Setup

Datasets We construct experiments on Open-TLS(Wu et al., 2025), a typical dataset specifically crafted for open-domain TLS. The statistics of the datasets are summarized in Tables 1.

	Open-TLS
# of topics	50
# of timelines	50
Avg. duration (days)	4139
Avg.l	23
Avg.k	1.8

Table 1: Statistics of Open-TLS. A timeline contains l dates associated with k sentences describing the events that happened at each date.

Evaluation metrics We adopt the following metrics to evaluate the generated timeline:

- **ROUGE-N** Derived from the original ROUGE-N metrics(Martschat and Markert, 2017), these metrics measure the overlap of N-grams in generated and reference timelines: (1)*Concat F1* computes ROUGE by

	Method	Concat F1		Agree F1		Align F1		Date F1
		R-1	R-2	R-1	R-2	R-1	R-2	
GPT-4o	DIRECT	0.243	0.063	0.056	0.021	0.071	0.025	0.208
	REWRITE	0.233	0.067	0.054	0.022	0.070	0.026	0.205
	CHRONOS	0.351	0.103	0.105	0.047	0.121	0.051	0.343
	FILTER	0.365	0.113	0.115	0.051	0.125	0.053	0.345
	R2A-TLS	**0.379**	**0.144**	**0.145**	**0.088**	**0.148**	**0.090**	**0.364**
Qwen2.5-72B	DIRECT	0.328	0.101	0.087	0.044	0.104	0.049	0.265
	REWRITE	0.337	0.106	0.091	0.046	0.107	0.050	0.291
	CHRONOS	0.368	0.110	0.106	0.049	0.125	0.050	0.324
	FILTER	0.375	0.118	0.117	0.053	0.128	0.054	0.328
	R2A-TLS	**0.385**	**0.151**	**0.148**	**0.093**	**0.153**	**0.095**	**0.375**

Table 2: Experimental results on Open-TLS.

concatenating all date summaries; (2)*Agree F1* computes ROUGE using only summaries of matching dates. (3)*Align F1* initially aligns predicted summaries with reference summaries based on similarity and date proximity, then calculates ROUGE between the aligned summaries, penalizing distant alignments.

- **Date F1** This metric is the F1 score of dates in the generated timeline compared to the reference timeline.

Baselines Our baselines are consistent with CHRONOS(Wu et al., 2025), which is the SOTA approach, and an additional method, FILTER, is proposed as the baseline.

- **DIRECT** Directly search for the target news and output a timeline with the retrieved news.
- **REWRITE** Rewrite the target news to create 2-3 queries, search with these rewritten queries, and output a timeline with the retrieved news.
- **CHRONOS** Use LLMs to generate timelines through iterative self-questioning and retrieval based processes.
- **FILTER** We have added an information filtering mechanism in the CHRONOS framework to generate a timeline only for articles that have been filtered.

4.2 Main Results

We construct experiments on based on GPT-4o(OpenAI et al., 2023) and Qwen2.5-72B(Bai et al., 2023), which are popular used in TLS. And we also fine-tune Qwen2.5-7B as the domain-adapted LLM for time point completion. Details about fine-tuning the domain-adapted LLM are show in Appendix A. Table 2 shows the average results of 3 runs during evaluation.

The results demonstrate a consistent improvement across all metrics when using the R2A-TLS approach compared to the baselines for each evaluated model. This indicates that our method enhances both the quality of event summarization and the alignment of dates with the reference timelines. In addition, comparative analysis demonstrates that the FILTER achieves noise reduction capabilities relative to CHRONOS, resulting in measurable improvements across multiple performance metrics. Due to page limitations, some ablation experiments are presented in the Appendix B.

4.2.1 R2A-TLS VS Iterative self-questioning

We use *Chrono-Informativeness* (CI) to systematically evaluate the quality of the generated questions between our method and Iterative self-questioning. It is designed to assess the ability of the questions to retrieve relevant documents that align chronologically with a reference timeline produced by a professional journalist(Wu et al., 2025). The *Chrono-Informativeness* of a set of questions $Q = (q_1, \ldots, q_m)$ for a given news topic is calculated as:

$$\text{CI}(Q, N) = Date\_F_1(T_{Q,N}, T_{ref})$$

where $T_{Q,N}$ is the timeline generated from the N documents retrieved through the rewritten version of Q, and T_{ref} is the reference timeline.

As shown in Figure 4, it demonstrates that our

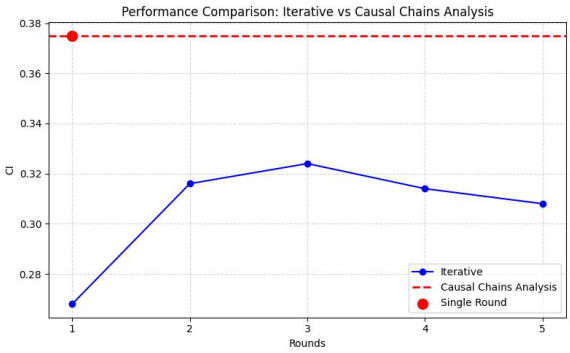

Figure 4: The comparison of the impact of multiple rounds of self-questioning on model performance and the results of single-round causal chain analysis.

question generation based on causal chains analysis achieves higher CI in a single round than the iterative strategy after multiple rounds of self-questioning. This indicates that our questions are more effective at retrieving documents and reconstructing accurate event timelines.

4.2.2 Time Intervals Analysis

We select six timelines from five different topics, as shown in Figure 5. To ensure the completeness of topic coverage, each topic has at least one timeline. To further explore whether there are patterns in the time intervals of different events within the same topic, two timelines with relatively similar names, **2024 Iran-Israel Conflict** and **Iran-Iraq War**, are specially selected in the political topic.

The interval distributions exhibit significant irregularity across datasets, with CV(coefficients of variation) ranging from 0.8 to 2.6 (Table 3). Extreme interval disparities, such as the 6,877-day gap in NBA records versus single-day intervals in Banking crises, further confirm the absence of periodic or clustered temporal structures. That is why we identify missing time points through causal chains analysis.

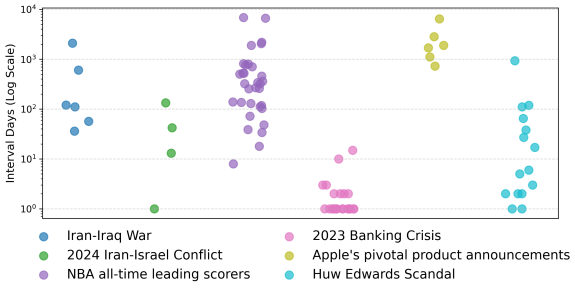

Figure 5: Time Intervals for Different Timelines.

Timeline	Mean	Std	Min	Max	CV
Iraq	507.5	743.9	36	2113	1.5
Israel	47.5	52.1	1	134	1.1
NBA	870.5	1618.3	8	6877	1.9
Crisis	2.3	3.3	1	15	1.4
Apple	2457.2	1916.1	729	64482	0.8
Huw	88.8	228.8	1	933	2.6

Table 3: Statistics of Time Intervals.

4.3 Case Study

Table 4 demonstrates how **R2A-TLS** summarizes a timeline of *Islamic State*. For the convenience of presentation, we have excerpted the parts of 2013 and 2014.

R2A-TLS performs two rounds of reflection, progressively refining the timeline from completing time points to completing event elements. In the time point completion, the causal chain of events is analyzed for completeness by analyzing the event, 2014-07-04, Baghdadi mentions allegiance to the caliphate, but the event of when and where Baghdadi became the caliphate is never mentioned before, so the questioning triggers a deep retrieval to complete the point in time. In the event element completion, the questioning shifts toward more specific event elements to enrich the timeline with finer details.

Comparing the generated timelines after the two rounds of completion with the reference timeline, R2A-TLS generates high quality timelines. However, some incorrect dates and incorrect events indicate that there is still room for improvement in extracting milestone events with correct dates.

5 Conclusion

In conclusion, this paper presents R2A-TLS, a novel framework for TLS that leverages LLMs through reflections for timeline completion and retrieval-based process. This framework establishes a closed-loop system of "retrieval reflection deep retrieval", to solve the redundant interference caused by information overload in traditional methods. And we address the problem of missing time points or missing event elements due to discrete open-domain information through collaborative reflection, leveraging causal chain analysis and FrameNet-based validation. Experiments demonstrates that R2A-TLS achieves SOTA results on Open-TLS dataset. Next, an interesting and useful topic would be to explore the future trend of events

<table>
<tr><th colspan="2">Target News: Islamic State</th></tr>
<tr><td>Questions for Time Point Completion</td><td rowspan="5">Reference Timeline:
2013-04-08: Abu Bakr al-Baghdadi, the leader of al-Qaida's branch in Iraq, announces the merger of his group with al-Qaida's franchise in Syria, forming the Islamic State in Iraq and the Levant.
2014-01-04: Al-Baghdadi's forces seize control of the city of Fallujah in western Iraq and parts of the nearby provincial capital of Ramadi.
2014-02-03: Al-Qaida leader Ayman al-Zawahri disavows al-Baghdadi after the Iraqi militant ignores his demands that IS leave Syria.
2014-06-10: IS captures Mosul, Iraqi's second-largest city, and pushes south as Iraqi forces crumble, eventually capturing Saddam Hussein's hometown of Tikrit and reaching the outskirts of Baghdad.", "When they threaten Shiite holy sites, Iraq's top Shiite cleric issues a call to arms, and masses of volunteers, largely backed and armed by Iran, join militias.
2014-06-29: The group renames itself the Islamic State and declares the establishment of a self-styled caliphate, a traditional model of Islamic rule, in its territories in Iraq and Syria.", "Al-Baghdadi is declared the caliph.
2014-07-04: Al-Baghdadi makes his first public appearance, delivering a Friday sermon in Mosul's historic al-Nuri Mosque. He urges Muslims around the world to swear allegiance to the caliphate and obey him as its leader.
2014-08-03: IS captures the town of Sinjar west of Mosul and begins a systematic slaughter of the tiny Yazidi religious community. Women and girls are kidnapped as sex slaves; hundreds remain missing to this day.
2014-08-08: The U.S. launches its campaign of airstrikes against IS in Iraq.
2014-09-22: The U.S.-led coalition begins an air campaign against IS in Syria.</td></tr>
<tr><td>Rewritten Questions:
1. When did Baghdadi become caliph?
2. Where did Baghdadi become caliph?</td></tr>
<tr><td>Generated Timeline:
2013-04-01: Abu Bakr al-Baghdadi, the leader of al-Qaida's branch in Iraq, announces the merger of his group with al-Qaida's franchise in Syria, forming the Islamic State in Iraq and the Levant
2014-01-04: Baghdadi's forces occupied parts of the area.
2014-02-01: Al-Qaida leader Ayman al-Zawahri disavows al-Baghdadi after the Iraqi militant ignores his demands that IS leave Syria.
2014-06-01: IS captures Mosul, Iraqi's second-largest city, and pushes south as Iraqi forces crumble, eventually capturing Saddam Hussein's hometown of Tikrit and reaching the outskirts of Baghdad. When they threaten Shiite holy sites, Iraq's top Shiite cleric issues a call to arms, and masses of volunteers, largely backed and armed by Iran, join militias.
2014-06-29: The group renames itself the Islamic State and declares the establishment of a self-styled 'caliphate', a traditional model of Islamic rule, in its territories in Iraq and Syria. Al-Baghdadi is declared the caliph.
2014-07-04: Al-Baghdadi makes his first public appearance, delivering a Friday sermon in Mosul's historic al-Nuri Mosque. He urges Muslims around the world to swear allegiance to the caliphate and obey him as its leader.
...</td></tr>
<tr><td>Questions for Event Element Completion</td></tr>
<tr><td>Rewritten Questions:
1. What areas Baghdadi's forces have captured in 2024-01-04?
2.What happened in 2024-01-04 regarding ISIS territorial gains?</td></tr>
<tr><td>Generated Timeline:
...
2014-01-04: Al-Baghdadi's forces overrun the city of Fallujah in Iraq's western Anbar province and parts of the nearby provincial capital of Ramadi. In Syria, they seize sole control of the city of Raqqa after driving out rival Syrian rebel factions, and it becomes their de facto capital.
...
2014-08-08: The U.S. launches its campaign of airstrikes against IS in Iraq.
2014-08-19: The Obama administration was at first reluctant to engage U.S. military power in the struggle against the Islamic State (ISIS) but became more deliberate in its approach. U.S. airstrikes into Syria during late 2014 and early 2015 were at least as heavily focused on the al-Qaaida-affiliated Khorasan Group known for its sophisticated bomb-making.
2014-09-22: The U.S.-led coalition begins an aerial campaign against IS in Syria.</td></tr>
</table>

Table 4: Case study of the part timeline generated by R2A-TLS. For the convenience of presentation, we have excerpted the parts of 2013 and 2014. The correctly predicted dates are in green, the incorrect dates with incorrect events are in are in red, and incorrect dates with correct events are in yellow. The missing time point or event element completed through our method are highlighted.

from the existing timeline. This research needs to improve temporal reasoning ability for LLMs, which has not yet received widespread attention.

Limitations

This study faces several limitations, including the computational resources required. The experimental results will be affected by the search engine, and we use JINA to read the content of the web pages. JINA can convert a URL that search engine returned to LLM-friendly input. However, JINA's URL-to-text conversion pipeline exhibits systematic failures: dynamic JavaScript-rendered content (charts, interactive elements) is frequently omitted during static HTML parsing, complex layouts (tables, multi-column formats) become unstructured text blocks, and non-English materials are automatically filtered, creating cultural perspective biases. Additionally, the design of prompts could impact the model's performance, underscoring the importance of meticulous prompt crafting to ensure high-quality outputs.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62476283).

References

James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal summaries of new topics. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 10–18, New York, NY, USA. Association for Computing Machinery.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. abs/2309.16609.

Giang Binh Tran, Mohammad Alrifai, and Dat Quoc Nguyen. 2013. Predicting relevant news events for timeline summaries. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13 Companion, page 91–92. Association for Computing Machinery.

Hans C Boas, Josef Ruppenhofer, and Collin Baker. 2024. Framenet at 25. *International Journal of Lexicography*, 37(3):263–284.

Tommaso Caselli and Piek Vossen. 2017. The event storyline corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the Events and Stories in the News Workshop*, pages 77–86.

Wanyun Cui and Xingran Chen. 2022. Instance-based learning for knowledge base completion. *Advances in Neural Information Processing Systems*, 35:30744–30755.

Jinglong Gao, Xiao Ding, Bing Qin, and Ting Liu. 2023. Is ChatGPT a good causal reasoner? a comprehensive evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11111–11126, Singapore. Association for Computational Linguistics.

Demian Gholipour Ghalandari and Georgiana Ifrim. 2020. Examining the state-of-the-art in news timeline summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1322–1334. Association for Computational Linguistics.

Goran Glavaš, Jan Šnajder, Parisa Kordjamshidi, and Marie-Francine Moens. 2014. Hieve: A corpus for extracting event hierarchies from news stories. In *Proceedings of 9th language resources and evaluation conference*, pages 3678–3683. ELRA.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Qisheng Hu, Geonsik Moon, and Hwee Tou Ng. 2024. From moments to milestones: Incremental timeline summarization leveraging large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7232–7246, Bangkok, Thailand. Association for Computational Linguistics.

Qiao Jin, Yifan Yang, Qingyu Chen, and Zhiyong Lu. 2024. Genegpt: Augmenting large language models with domain tools for improved access to biomedical information. *Bioinformatics*, 40(2):btae075.

Dylan Kneale, James Thomas, Mukdarut Bangpan, Hugh Waddington, and David Gough. 2018. Conceptualising causal pathways in systematic reviews of international development interventions through adopting a causal chain analysis approach. 10:422–437.

Wei Li, Dezhi Cheng, Lei He, Yuanzhuo Wang, and Xiaolong Jin. 2019a. Joint event extraction based on hierarchical event schemas from framenet. *IEEE Access*, 7:25001–25015.

Wei Li, Dezhi Cheng, Lei He, Yuanzhuo Wang, and Xiaolong Jin. 2019b. Joint event extraction based on hierarchical event schemas from framenet. 7:25001–25015.

Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, Julian McAuley, Wei Ai, and Furong Huang. 2025. Large language models and causal inference in collaboration: A comprehensive survey. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7668–7684, Albuquerque, New Mexico. Association for Computational Linguistics.

Sebastian Martschat and Katja Markert. 2017. Improving ROUGE for timeline summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 285–290, Valencia, Spain. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, and et al. 2023. Gpt-4 technical report. abs/2303.08774.

Girishkumar Ponkiya, Diptesh Kanojia, and Pushpak Bhattacharyya. 2021. Framenet-assisted noun compound interpretation. pages 2901–2911.

Muhammad Reza Qorib, Qisheng Hu, and Hwee Tou Ng. 2024. Just what you desire: Constrained timeline summarization with self-reflection for enhanced relevance. abs/2412.17408.

Yongpan Sheng, Xuefeng Fu, and Tianxing Wu. 2020. Multi-document conceptual graph construction research based on open domain extraction. 37:19–25.

Jiayu Song, Jenny Chim, Adam Tsakalidis, Julia Ive, Dana Atzil-Slonim, and Maria Liakata. 2024. Combining hierachical VAEs with LLMs for clinically meaningful timeline summarisation in social media. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14651–14672, Bangkok, Thailand. Association for Computational Linguistics.

Giang Tran, Mohammad Alrifai, and Eelco Herder. 2015. Timeline summarization from relevant headlines. In *Advances in Information Retrieval*, pages 245–256, Cham. Springer International Publishing.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage bert: A globally normalized bert model for open-domain question answering. volume abs/1908.08167.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models.

Weiqi Wu, Shen Huang, Yong Jiang, Pengjun Xie, Fei Huang, and Hai Zhao. 2025. Unfolding the headline: Iterative self-questioning for news retrieval and timeline summarization.

Dizhan Xue, Shengsheng Qian, and Changsheng Xu. 2024. Integrating neural-symbolic reasoning with variational causal inference network for explanatory visual question answering. 46:7893–7908.

Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. 2011. Timeline generation through evolutionary trans-temporal summarization. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 433–443, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Yi Yu, Adam Jatowt, Antoine Doucet, Kazunari Sugiyama, and Masatoshi Yoshikawa. 2021. Multi-timeline summarization (mtls): Improving timeline summarization by generating multiple summaries. pages 377–387.

Chenlong Zhang, Tong Zhou, Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. 2025. DTELS: Towards dynamic granularity of timeline summarization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2682–2703, Albuquerque, New Mexico. Association for Computational Linguistics.

A Experimental Details

In our experimental setup, we use the Google search engine. We set parameter N to 20, defining the maximum number of retrieved documents in the first round. The parameter m is set to 5, which represents the number of retrieved documents for each question used to complete the time point. In addition, we have designated s as 3 and retrieve the number of documents for each question used to complete event elements. All used prompts in this work will be made available in a public GitHub repository upon acceptance.

Parameter	Value
batchsize	8
nun_epochs	3
learning_rate	5e-5
lora_r	8
lora_alpha	16
lora_dropout	0.05
lora_target_modules	q_proj,v_proj

Table 5: Parameters for fine-tuning of domain-adapted LLM.

In fine-tuning for domain-adapted LLM, we adopted LoRA(Hu et al.) and fine-tuned Qwen2.5-7B on an A100 40GB GPU. We used 60 topics and 88 timelines from three datasets T17(Binh Tran et al., 2013), CRISIS (Tran et al., 2015) and ENTITIES (Gholipour Ghalandari and Ifrim, 2020) to form 352 training samples. The hyperparameter settings for LoRA fine-tuning are listed in Tables 5.

Method	Concat F1		Agree F1		Align F1		Date F1
	R-1	R-2	R-1	R-2	R-1	R-2	
-F-E	0.329	0.108	0.089	0.046	0.107	0.051	0.278
-TPC	0.344	0.137	0.129	0.083	0.134	0.086	0.371
-EEC	0.375	0.147	0.142	0.091	0.146	0.093	0.309
R2A-TLS	**0.385**	**0.151**	**0.148**	**0.093**	**0.153**	**0.095**	**0.375**

Table 6: Impact of Timeline Completion on model performance within the Open-TLS dataset. **-F-E** indicates that both event element completion and time point completion are not adopted, **-TPC** indicates that event element completion is not adopted, and **-EEC** indicates that time point completion is not adopted.

N	Concat-R1	Concat-R2	Agree-R1	Agree-R2	Align-R1	Align-R2	Date F1
10	0.378	0.146	0.145	0.090	0.150	0.091	0.345
20	**0.385**	**0.151**	**0.148**	0.093	**0.153**	0.095	0.375
30	0.384	0.149	0.147	**0.095**	0.152	**0.096**	**0.377**

Table 7: Performance on Open-TLS with different numbers of news retrieved in first round.

B Ablation Study

B.1 Effects of Timeline Completion

The ablation experiments systematically evaluate the contributions of the Time Point Completion (TPC) and Event Element Completion (EEC) modules through controlled component removal. As shown in Table 6, we can observe that:

1. The removal of TPC causes a substantial 6.6% absolute decline in Date F1 (0.375 $\rightarrow$ 0.309), contrasting with only 1% reduction in Concat R1. This disparity confirms that TLS requires explicit causal constraints.

2. EEC removal predominantly impacts summary quality, with ROUGE-N decreasing 1.2% (vs. 0.4% Date F1 decline). This aligns with Frame Semantics theory, where core elements omission directly degrades summary informativeness.

3. It is worth noting that the index decay of removing two modules at the same time exceeds the sum of individual removal, which indicates that time point completion ensures the integrity of time points through the constraints of event causal logic, which can effectively improve Date F1. Event element completion bears the semantic enrichment function, which effectively improves the quality of the summary. Through the collaboration of the two, the quality of the timeline is improved, which proves the effectiveness of Causal-Semantic Synergistic Reflection for Timeline Completion.

B.2 Number of Retrieved News

To determine the impact of retrieved news in first round, we experiment with retrieving 10, 20, 30 documents using Qwen2.5-72B on the Open-TLS dataset. Table 7 indicates that increasing the number from 10 to 20 documents significantly improves the results, with marginal improvements when increasing to 30 documents.

This verifies the core limitation of the first round of retrieval, whose time points obtained through keyword matching constitute only the base temporal scaffolding, and are unable to capture implicit causal or temporal relationships (e.g., causal chain breakages between events). When the document volume reaches 30, the performance tends to stabilize, revealing the objective existence of the retrieval threshold, and it is meaningless to simply increase the retrieval volume. This phenomenon stems from the cognitive reconstruction nature of timeline synthesis, and the fragmented time points provided by the first round of retrieval lack causal coherence, which needs to be improved through the reflection mechanism.

Strategy	Concat-R1	Concat-R2	Date-F1
Single	0.378	0.147	0.372
Dual	0.385	0.151	0.375

Table 8: Comparison of results between single-filtering and dual-filtering.

C Additional Analysis

C.1 Why decompose filtering into two steps?

We conduct a comparison between single-filtering (determine topic relevant and information gain simultaneously) and dual-filtering. As shown in

	Method	Concat F1		Agree F1		Align F1		Date F1
		R-1	R-2	R-1	R-2	R-1	R-2	
Gemini 2.5 Pro	CHRONOS	0.318	0.081	0.083	0.045	0.086	0.042	0.312
	R2A-TLS	0.336	0.095	0.098	0.079	0.102	0.052	0.325
GPT-4o	CHRONOS	0.351	0.103	0.105	0.047	0.121	0.051	0.343
	R2A-TLS	**0.379**	**0.144**	**0.145**	**0.088**	**0.148**	**0.090**	**0.364**
Qwen2.5-72B	CHRONOS	0.368	0.110	0.106	0.049	0.125	0.050	0.324
	R2A-TLS	**0.385**	**0.151**	**0.148**	**0.093**	**0.153**	**0.095**	**0.375**

Table 9: Comparison of results between reasoning models and generic LLMs.

Table 8, the dual-filtering achieves consistent improvements across all metrics compared to the single-filtering, which suggests that filtering in two steps is effective. Specifically, it yields a 1.8% relative increase in Concat-R1, 2.7% enhancement in Concat-R2, and Date-F1 improvement from 0.372 to 0.375.

C.2 Why not select reasoning models?

We try to use reasoning models such as Gemini in our evaluation. However, after careful consideration and experimentation, we believe that such models are not well-suited for this task for the following reasons. Cost: Reasoning models (e.g., Gemini) are significantly more expensive compared to general-purpose LLMs. For instance, a timeline can cost around 2 dollars, whereas using Qwen-2.5-72B only costs approximately 0.2 dollars per timeline.

Inference Speed: Reasoning models tend to be slower in generating responses, which hampers the efficiency of iterative processes like timeline summarization. Our framework relies on multiple rounds of retrieval and reflection, where response latency directly affects overall runtime.

Empirical Performance: We conduct preliminary experiments with the latest Gemini2.5-Pro using both CHRONOS and R2A-TLS frameworks. However, the results do not outperform generic LLMs like Qwen or GPT-4o on Date F1 or ROUGE-based metrics. The average decline of each indicator exceeds 2%, and the specific results are shown in Table 9.

Therefore, while reasoning models have shown impressive capabilities in certain domains, we find them currently less effective and efficient for the open-domain timeline summarization task.

C.3 Demonstration of Method Generalizability

We select the causal dataset Event Storylines Corpus (ESC)(Caselli and Vossen, 2017) and the sub-event relationship dataset HiEve(Glavaš et al., 2014) for our experiments to test the model's ability to recognize event relations.

Each training sample includes an incomplete timeline, deleted time points, and answers derived from those missing points. This formulation focuses the model on learning temporal coherence and causal consistency rather than memorizing specific event patterns. TLS entails extracting event relations, specifically sub-event and causal relations.

The baseline method is fine-tuned directly on the dataset, while our method fine-tunes on both the dataset and the closed-domain dataset simultaneously. As shown in Table 10, accuracy is significantly higher with the domain norms which proves that the method is not optimized for the Open-TLS dataset, but is universally applicable.

Dataset	Baseline/ACC	Ours/ACCS
HiEve	71.3	73.5
ESC	65.7	70.2

Table 10: Generalizability Validation of TLS across Domain-Specific Datasets.

Moreover, we emphasize that the prompts we designed and employed are generic, rather than being tailored to a specific dataset or domain. For instance, causal reasoning and FrameNet-based validation components can be applied to any event timeline without domain-specific customization. As shown in Table 11, we select the well-established benchmarks Crisis and T17 for evaluating closed-domain TLS and focus on representative performance metrics including Align

F1 (short for AR-1 and AR-2) and Date F1. Our method is comparable to the previous SOTA.

	Method	AR-1	AR-2	Date F1
Crisis	LLM-TLS	0.112	0.032	0.329
	CHRONOS	0.108	0.045	0.323
	R2A-TLS	0.112	0.046	0.331
T17	LLM-TLS	0.118	0.036	0.528
	CHRONOS	0.116	0.042	0.522
	R2A-TLS	0.119	0.048	0.535

Table 11: Comparison of our method with previous works on closed-domain TLS benchmarks.

D Prompt Template

D.1 Prompt Template for Iterative Timeline Generation

The prompt templates for LLMs to generate timelines of Top3 results are shown in List 1 and List 2.

Listing 1: Instruction templates for filter Top3 results.

```
/* Task prompt */
1.Analyze whether each article contains
a topic related to {news}.
2.Determine if the article provides key
event progress (e.g. policy releases,
major meetings, milestone results, etc.)
.
3.If the article only contains
discussion of ideas or statistics
without time correlation, it is
considered useless.
/* Input */
topic:{news}
Article 1:
    title:{top_articles[0]['title']}
    content:{content}
Article 2:
    title:{top_articles[1]['title']}
    content:{content}
Article 3:
    title:{top_articles[2]['title']}
    content:{content}
/* Output rules */
Only output "Article X  useful" or "
Article X useless" (X is 1/2/3).
One result per line, output strictly in
article order.
Explanations, punctuation, or formatting
 are prohibited.
/* Examples*/
Correct output:
    Article 1  useful
    Article 2  useless
    Article 3  useful
Error output:
   1.Article 1 (useful)
   2.Article 2 may be useless
   3.The third article is useful because
    the 2023 data is mentioned
```

Listing 2: Instruction templates for generate timelins.

```
/* Task prompt */
You are an experienced journalist
building a timeline for the target news.
1.Read each background news item and
extract all significant milestone events
 related to the target news from your
news database, along with their dates.
2.Write a description for each event,
including key detail information about
the event, using the phrasing from the
news database as much as possible. Save
all events as a list. The format should
be: [{{"start": <date|format as
"2023-02-02">, "summary": "<event
description|no quotes allowed>"}}, ...]
3.Systematically sort the event list
from Step 2 chronologically by first
converting all date strings to standard
Date objects to ensure accurate cross-
year/month sorting, finally outputting a
 timeline list that strictly follows
temporal progression and maintains the
original JSON format.
/* Input */
topic:{news}
Articles:{docs}
/* Output rules */
Directly output your answer in the
following format, as a list:
[{{"start": <date|format as
"2023-02-02", cannot be empty, must
include specific year, month, and day>,
"summary": "<event description|no quotes
 allowed>"}}, ...]
/* Examples*/
[{{"start": "2024-01-01", "summary": "An
 event happens."}}]]
```

The prompt templates for LLMs to filter the rest of results via dual-filtering mechanisms are shown in List 3 and List 4.

Listing 3: Instruction templates for topic relevance assessment.

```
/* Task prompt */
1.Analyze whether the content of the
article is related to the topic.
2.If the article only contains
discussion of ideas or statistics
without time connection, it will be
considered unrelated.
/* Input */
topic:{news}
title:{article['title']}
content:{content}
/* Output rules */
Directly output "relevant" or "unrelated
".
Explanations, punctuation or formatting
are prohibited.
```

Listing 4: Instruction templates for information gain judgement.

```
/* Task prompt */
Analyze whether the new article is
related to the exist timeline that has
already been obtained.
```

```
1.Discover new time nodes and provide
key event progress (such as policy
releases, major meetings, milestones,
etc.).
2.Add new details to existing time nodes
 (roles, data, or causal descriptions).
3.If any of the above points are met, it
 is considered relevant. If none are
satisfied, it is considered unrelated.
/* Input */
topic:{news}
exist timeline:{useful_timeline}
title:{article['title']}
content:{content}
/* Output rules */
Directly output "relevant" or "unrelated
".
Explanations, punctuation or formatting
are prohibited.
```

The prompt templates for LLMs to merge and sort timelines are shown in List 5.

Listing 5: Instruction templates for information gain judgement.

```
/* Task prompt */
You are an experienced journalist
building a timeline for the target news.
 Merge the existing news summaries and
timelines in chronological order.
1.When merging the news summaries,
select the top-{k} significant news from
 the original timeline, and strictly
follow the chronological order from past
 to present without changing the
original date.
2.Using "\n" to separate events that
occurred on different dates.
/* Input */
topic:{news}
Original Timeline:{timelines}
/* Output rules */
Directly output your answer in the
following format: [{{"start": <date|
format as "2023-02-02", cannot be empty,
 must include specific year, month, and
day>, "summary": "<event description|no
quotes allowed>"}}, ...]
```

D.2 Prompt Template for Time Point Completion

The prompt templates for LLMs to time point completion are shown in List 6, and the prompts for question rewriting is similar to CHRONOS(Wu et al., 2025).

Listing 6: Instruction templates for time point completion.

```
/* Task prompt */
You are a senior analyst specializing in
 temporal causality verification, tasked
 with examining timelines for targeted
news or entities.
1.Potential missing time nodes are
deduced based on the logical
relationships of the events.
2.At least three questions are
formulated, which should assist in
retrieving the potential missing time
nodes associated with the target news.
This process aims to help continue
organizing the timeline of news
developments or the life history of
individuals, focusing on the origins,
development processes, and key figures
of related events, while emphasizing
factual news knowledge rather than
subjective evaluative content.
/* Input */
topic:{news}
Initial timelines:{timelines}
/* Output rules */
Directly output your questions in the
specified format:["Question 1", "
Question 2", "Question 3", ...]
```

D.3 Prompt Template for Event Element Completion

The prompt templates for LLMs to event element completion are shown in List 7 and List 8.

Listing 7: Instruction templates for FrameNet recommendation.

```
/* Task prompt */
You are a linguist proficient in frame
semantics theory, very familiar with
FrameNet, responsible for providing
appropriate FrameNet framework names for
 specific news or entity timelines.
1.The purpose is to detect whether there
 is a missing core frame emelens at the
time point based on the content of the
list in the next step.
2.Based on the given event name and
timeline, provide a list for the
appropriate FrameNet framework name and
corresponding core frame elements for
that timeline.
/* Input */
topic:{news}
Current timeline:{timeline}
/* Output rules */
Return a list, where the elements in the
 list are a dictionary, the key is the
frame name, and the value is the core
frame elements,without the need to
interpret the frame.
/* Examples */
[{{"Military_operation":"Area , Force ,
Goal, Opponent"}},
{{"Communication":"Communicator, Medium,
 Message , Topic"}}]
```

Listing 8: Instruction templates for core element check.

```
/* Task prompt */
You are a seasoned journalist tasked
with verifying the completeness of a
timeline for a specific news story or
entity.
1. Check if the time nodes are complete
based on framenet_core_elements.
```

```
2. Identify nodes with information gaps
and develop targeted problems to address
 these gaps.
- Avoid redundant questions across nodes
- Prioritize clarity, verifiability, and
 search engine optimization (SEO) in
question formulation.
/* Input */
topic:{news}
Current timeline:{timeline}
framenet_core_elements:{frame_elements}
/* Output rules */
Each question must:
- Be a standalone item addressing a
specific information deficit
- Use concrete, simple language to
facilitate search engine retrieval
- Avoid jargon and complex syntax
Output format: ["Question 1", "Question
2", "Question 3", ...]
```