

From Gaps to Active Retrieval: Open-Domain Iterative Timeline Completion via Historical Trajectory Analogy and Dynamic Ontology

Yunhui Zheng*, Chenlong Bao*, Shasha Li, Ting Wang, Jintao Tang†

College of Computer Science and Technology, National University of Defense Technology

{zhengyunhui, baochenlong, shashali, tingwang, tangjintao}@nudt.edu.cn

Abstract

Open-Domain Timeline Summarization (Open-TLS) seeks to construct coherent evolutionary narratives for a topic by distilling information from large, unstructured web streams. Existing retrieve-and-generate pipelines often fail to capture logically necessary but semantically distant evidence, resulting in **macro-level causal gaps** (missing intermediate stages) and **micro-level element gaps** (underspecified milestones). We propose Active Timeline Completion (ATC), an iterative framework that converts implicit narrative gaps into targeted retrieval questions, enabling **reflection-driven evidence acquisition**. ATC leverages three complementary mechanisms: parametric causal gap detection, historical trajectory analogy, and dynamic ontology-guided completion—while a **Logical Utility Verification** module ensures only evidence directly addressing these gaps updates the timeline. Experiments on Open-TLS demonstrate that ATC consistently outperforms strong baselines in both content coverage and temporal alignment, showing that active, question-driven retrieval improves the completeness and coherence of open-domain timelines.

1 Introduction

Constructing coherent evolutionary narratives from massive streams of unstructured web text is a central problem in open-domain Timeline Summarization (TLS). Given a topic, TLS aims to distill large-scale documents into a compact, chronologically organized summary that helps users understand how complex events evolve over time. In open-domain settings, however, timeline construction faces a persistent tension between **information overload** and **data sparsity**: high-frequency news streams introduce substantial redundancy

*Equal contribution.

†Corresponding author.

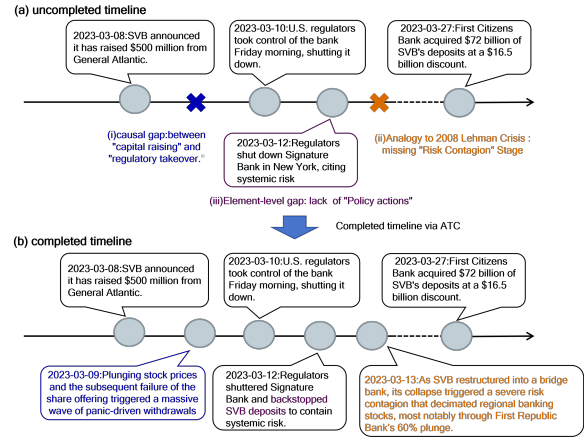


Figure 1: Illustrative example of completion gaps in Open-Domain TLS. The baseline timeline exhibits (i) a macro-level causal gap between capital raising and regulatory takeover, (ii) a missing risk-contagion stage revealed by trajectory analogy, and (iii) micro-level element gaps (e.g., missing policy actions), which are completed by ATC.

and noise, whereas critical long-tail milestones are often sparsely reported. As a result, generated timelines are frequently incomplete, exhibiting **macro-level causal gaps** such as missing intermediate stages and **micro-level element gaps** such as under-specified milestone descriptions.

A key bottleneck lies in the mismatch between passive retrieval and narrative logic. Most existing pipelines follow a *retrieve-then-generate* paradigm, where documents are retrieved according to semantic relevance and then summarized into a timeline. However, many missing timeline events are not simply semantically similar to observed events; instead, they are *logically necessary* for maintaining causal continuity. For example, given a timeline transition from “indictment” to “verdict”, the missing “trial” stage is causally required, but reports about the trial may not be retrieved if the search space is dominated by repetitive documents about the indictment or

verdict. Consequently, relevance-based retrieval may return documents that are topically related but evolutionarily uninformative, leaving downstream LLMs to either preserve unresolved gaps or hallucinate unsupported transitions.

Recent LLM-based TLS methods have improved timeline construction, but they still provide limited mechanisms for explicitly identifying and retrieving missing evidence. CHRONOS (Wu et al., 2025) adopts an iterative self-questioning strategy to expand timelines, yet the generated questions are still largely driven by the current context and may overlook logically necessary but weakly signaled stages. Reflection-based methods such as R2A-TLS (Bao et al., 2025) further incorporate feedback and verification to reduce omissions, but their verification relies on relatively static semantic frames, which can be brittle when news streams contain schema drift or emerging event types. More broadly, these approaches tend to retrieve what appears relevant to the topic, rather than what is required to complete the event evolution. This limits their ability to recover missing causal links and under-specified event elements in open-domain scenarios, motivating a shift from passive relevance matching to active gap discovery and completion.

To address these challenges, we propose **Active Timeline Completion (ATC)**,¹ an iterative framework that reformulates open-domain TLS as a process of active, question-driven evidence acquisition. Instead of passively summarizing retrieved documents, ATC first reflects on the current timeline to identify potential narrative gaps, then converts these gaps into explicit “what to retrieve” questions. These questions are rewritten into search queries, used to retrieve additional evidence, and verified before the working timeline is updated. This design makes the retrieval process gap-aware: evidence is not incorporated merely because it is topically relevant, but because it helps resolve a specific causal or element-level deficiency.

ATC operationalizes this idea through three complementary mechanisms, each targeting a different source of timeline missingness. First, **Parametric Causal Gap Detection** leverages the parametric knowledge of LLMs to inspect adjacent milestones and identify suspicious causal discon-

tinuities, such as a missing trial between indictment and verdict. Second, **Historical Trajectory Analogy** uses structurally similar historical timelines as priors to infer missing intermediate stages that may not be obvious from the current topic alone. Rather than treating these trajectories as evidence, ATC uses them to generate targeted retrieval questions whose answers must still be supported by topic-specific web documents. Third, **Dynamic Ontology-Guided Completion** maintains an evolving set of event categories and elements, allowing the system to detect under-specified milestones and generate element-oriented questions under schema drift. Finally, a **Logical Utility Verification (LUV)** module judges whether each candidate event directly answers the current gap question, filtering out topically related but logically unhelpful updates.

The main contributions of this work are summarized as follows:

- We formulate Open-Domain TLS as an *active timeline completion* problem, shifting the retrieval objective from passive semantic relevance matching to question-driven evidence acquisition. ATC explicitly identifies missing causal stages and underspecified event elements, converts them into retrieval-oriented questions, and verifies whether retrieved evidence repairs the working timeline.
- We propose a three-stage completion framework that combines parametric causal gap detection, historical trajectory analogy, and dynamic ontology-guided event completion. Historical trajectories provide structural priors for hypothesizing missing intermediate stages, while the dynamic ontology adapts to open-domain schema drift and supports element-level milestone enrichment.
- We introduce Logical Utility Verification (LUV) to filter topically related but gap-irrelevant evidence, and conduct extensive experiments on Open-TLS. Results show that ATC consistently improves content coverage, temporal alignment, and date accuracy over strong baselines, with additional analyses on ablation, efficiency, LUV sensitivity, and closed-domain transfer.

¹Code and resources are available at https://github.com/zzyunh/ATC_Timeline.

2 Related Work

2.1 Timeline Summarization

Timeline Summarization (TLS) aims to organize salient events from large document collections into chronologically structured summaries. Early work mainly relied on extractive methods, using textual salience, temporal signals, and chronological constraints to select representative events (Allan et al., 2001; Yu et al., 2021). Later studies revisited the design and evaluation of TLS systems, showing that integrated retrieval and summarization pipelines can be competitive with more complex multi-stage systems (Gholipour Ghalandari and Ifrim, 2020). With the development of large language models (LLMs), recent TLS research has shifted toward more flexible generative and interactive paradigms. CHRONOS decomposes open-domain timeline construction into iterative self-questioning, retrieval, and summarization (Wu et al., 2025), while incremental TLS methods use LLMs to bridge fine-grained moments and high-level milestones in evolving streams (Hu et al., 2024). Other studies extend TLS to constrained or multi-dimensional settings, such as user-constrained timeline generation (Qorib et al., 2025), stakeholder-aware modeling (Sim et al., 2025), and dynamic-granularity timelines (Zhang et al., 2025).

Despite these advances, existing TLS systems still struggle in open-domain scenarios where relevant evidence is both noisy and sparse. In particular, current methods often retrieve and summarize topically relevant documents, but they do not explicitly model which missing events are logically required to complete an evolving narrative. Our work addresses this limitation by formulating timeline construction as an active completion process that converts narrative gaps into targeted retrieval questions.

2.2 Retrieval-Augmented and Reflective Generation

Retrieval-Augmented Generation (RAG) grounds LLM outputs in external evidence and has become a standard approach for knowledge-intensive generation (Lewis et al., 2020). Recent studies further introduce reflection and verification mechanisms to improve retrieval quality and generation faithfulness. Self-RAG trains models to decide when to retrieve and to critique retrieved passages and generated outputs (Asai et al.,

2023). In TLS, R2A-TLS introduces a retrieval–reflection–deep retrieval pipeline to reduce omissions through causal-chain analysis and frame-based validation (Bao et al., 2025). These methods show that reflection can improve evidence acquisition, but their retrieval objectives are still largely tied to semantic relevance or predefined validation structures.

This limitation is important for timeline construction because semantically relevant evidence is not always logically useful. NLP studies on causal inference have shown that correlation-based models may rely on spurious associations rather than causal necessity (Feder et al., 2022). Similarly, embedding-based retrieval can be limited when relevance depends on complex relational or logical patterns rather than single-vector similarity (Weller et al., 2026). Causal-aware RAG begins to incorporate explicit causal structures into retrieval (Wang et al., 2025), but it rarely considers cross-event structural priors for timeline completion. ATC differs from these approaches by using reflection to generate gap-specific questions and by verifying whether retrieved evidence logically resolves the current timeline deficiency.

2.3 Event Schemas and Ontology Induction

Event-centric NLP often relies on predefined event types and argument roles, which can be effective in constrained domains but brittle in open-domain news streams where event structures evolve over time. To address this issue, recent work explores bottom-up schema or ontology induction. IncSchema incrementally constructs hierarchical event schemas with prompting and relation verification (Li et al., 2023), CEO induces open-domain event ontologies from corpora (Xu et al., 2024), and Event Ontology Completion studies continual ontology growth through clustering, hierarchy expansion, and type naming (Cao et al., 2023). In parallel, event extraction methods such as ULTRA and MEET improve document-level or multi-document event understanding with hierarchical modeling, self-refinement, and long-context reasoning (Zhang et al., 2024; Min et al., 2025). Continual event detection further highlights the challenge of distribution shift and emerging event types in evolving streams (Dao et al., 2024; Qiu et al., 2025). Taken together, these advances motivate further investigation into how dynamically induced ontologies can be connected to open-domain TLS, especially for handling miss-

ing milestones and under-specified event elements in evolving news streams.

3 Methodology

3.1 Task Formulation and Overview

Given a topic query q and access to the open Web $\mathcal{W}$, Open-Domain Timeline Summarization (Open-TLS) aims to construct a chronological timeline

$$\mathcal{T} = \{(t_i, y_i)\}_{i=1}^n, \quad (1)$$

where t_i denotes a timestamp and y_i is the corresponding milestone summary. Unlike closed-domain TLS, where the document collection is fixed, Open-TLS requires the system to actively acquire evidence from noisy and evolving web sources. We therefore treat timeline construction as an iterative retrieval-driven process, where the timeline length n is not predefined but adapts to the complexity of the target event and the amount of verified evidence.

3.2 Active Timeline Completion Pipeline

As shown in Fig. 2, ATC operates in two phases. It first builds a cold-start timeline using a standard search-and-summarize procedure over an initial open-domain document pool. It then applies a shared iterative completion loop, *Reflect–Rewrite–Retrieve–Verify–Update*, to progressively fill missing timeline information. At each iteration, a stage-specific reflector identifies a potential narrative gap and formulates it as a natural-language question; a Q2Q rewriter converts the question into search queries; retrieved documents are transformed into candidate events; and a verifier determines whether each candidate should update the working timeline. The completion loop is shared across all stages, while the reflectors are stage-specific (Sec. 3.3).

Formally, for a gap question q^* , the Q2Q module produces a small set of executable search queries:

$$\text{Q2Q}(q^*) = \{k_1, k_2, k_3\}. \quad (2)$$

Fixing the number of rewrites provides a stable retrieval budget across stages and prevents unbounded query expansion. The returned documents are then used to extract candidate events, each represented as $x = (t(x), \hat{y}(x))$, where $t(x)$ is the extracted timestamp and $\hat{y}(x)$ is a concise

event description. This event-centric representation allows ATC to insert newly verified information into the timeline without assuming a predefined event schema.

Open-domain retrieval may return documents from which the extracted candidate events are topically related but do not resolve the current gap. ATC therefore applies **Logical Utility Verification (LUV)** as a question-conditioned filter over candidate events. Given a candidate event $x = (t(x), \hat{y}(x))$ extracted from retrieved documents and the corresponding gap question q^* , an LLM-based judge assigns a utility score:

$$u(x, q^*) = \text{Judge}(q^*, x) \in [0, 1]. \quad (3)$$

LUV focuses on the logical utility of the extracted candidate event itself, while evidence grounding is established by the preceding retrieval and event extraction steps. The score reflects three criteria: whether x directly answers the gap question, whether it repairs a missing causal stage, temporal transition, or event element, and whether its timestamp and description are compatible with the chronological implication of the question. The event is accepted only if $u(x, q^*) \geq \gamma$; otherwise, it is discarded even if it is semantically related to the topic. For each question, ATC keeps up to top- m accepted candidate events to preserve recall under sparse reporting while bounding noisy insertions. Accepted events are inserted into the working timeline according to their timestamps. After all stages finish, overlapping or same-date events are handled by the final consolidation step (Sec. 3.4). Stage budgets, stopping criteria, and acceptance thresholds are specified in Sec. 4.1, with additional implementation details provided in Appendix A.1.

3.3 Stage-aware Reflective Gap Discovery

ATC uses stage-aware reflectors to instantiate different gap types as actionable questions, thereby steering retrieval toward evidence that is maximally informative for timeline completion. We next detail the three reflectors in Fig. 2(c), while all generated questions are executed by the shared pipeline in Sec. 3.2.

Parametric Causal Gap Detection Stage 1 in Fig. 2(c) targets **macro-level causal discontinuities** by leveraging the LLM’s parametric knowledge as a plausibility prior. Given the current time-sorted working timeline $\mathcal{T} = \{(t_i, y_i)\}_{i=1}^n$,

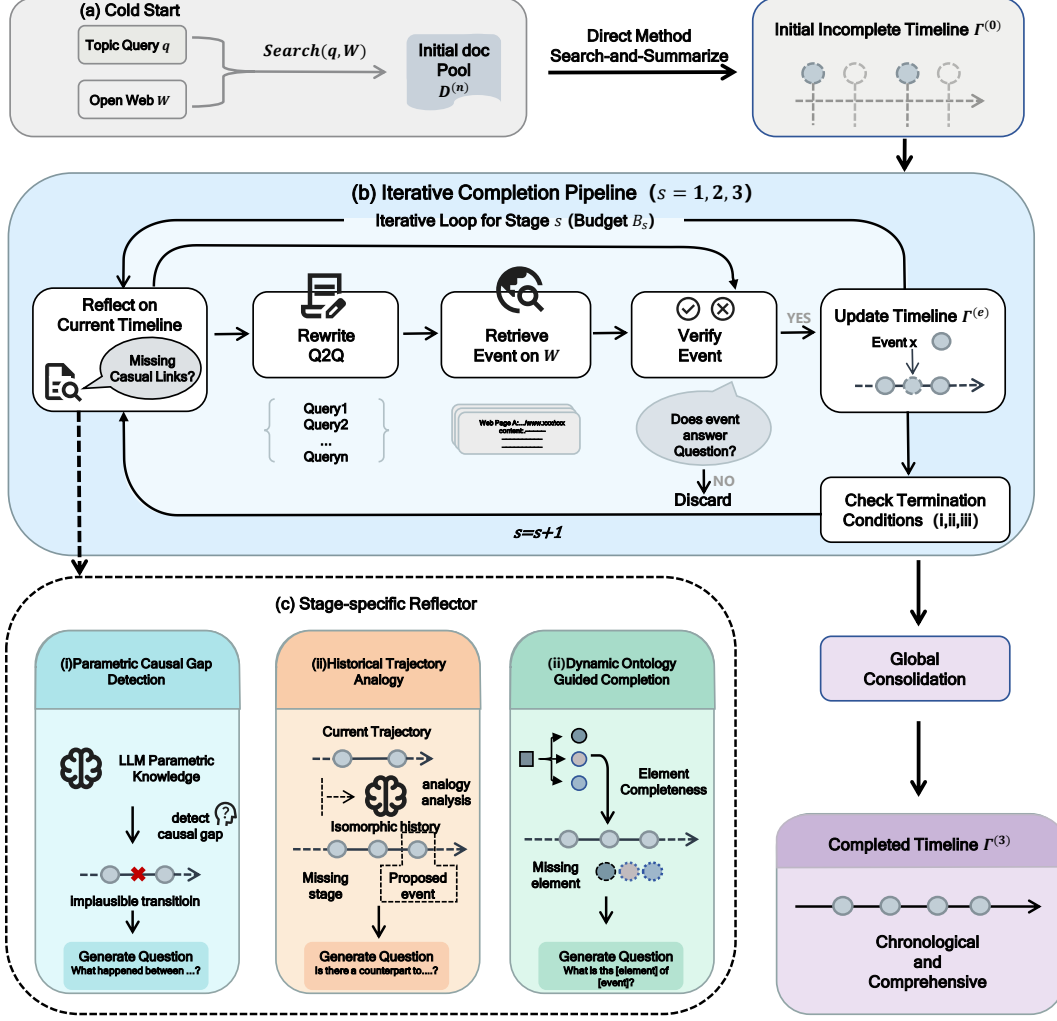


Figure 2: Overview of Active Timeline Completion (ATC). (a) Cold start: retrieve an initial open-domain document pool and produce an initial incomplete timeline via search-and-summarize. (b) Iterative completion pipeline (shared across stages): iteratively *Reflect–Rewrite–Retrieve–Verify–Update* under stage budget B_s and termination conditions. (c) Stage-specific reflectors: parametric causal gap detection, historical trajectory analogy, and dynamic ontology-guided completion.

we inspect adjacent transitions and ask whether the evolution between two consecutive milestones appears causally incomplete. For each pair $(t_i, y_i) \rightarrow (t_{i+1}, y_{i+1})$, we form a compact transition context:

$$c_i = \langle y_i, t_i; y_{i+1}, t_{i+1} \rangle. \quad (4)$$

A stage-1 reflector Reflect_1 then generates a small set of natural-language gap questions:

$$\mathcal{Q}^{(1)} = \bigcup_{i=1}^{n-1} \text{Reflect}_1(c_i). \quad (5)$$

Intuitively, Reflect_1 checks whether the transition violates common event scripts (e.g., “indictment $\rightarrow$ verdict” without an intervening “trial”) and expresses the suspected missing stage as a question.

The resulting questions are executed by the shared iterative pipeline in Sec. 3.2 (Fig. 2(b)) to retrieve and insert bridging events. By decoupling gap detection (parametric) from evidence acquisition (retrieval), Stage 1 exposes missingness even when the current evidence pool is sparse.

Historical Trajectory Analogy Stage 2 in Fig. 2(c) introduces **Historical Trajectory Analogy** as a trajectory-level structural prior for proposing missing stages that are difficult to infer from local transitions alone. Let $\mathcal{H}$ denote a repository of historical timelines, where each trajectory is an ordered sequence of date-grounded milestones. Given the current partial timeline $\mathcal{T}$, ATC retrieves

a small set of analogous reference trajectories:

$$\mathcal{R} = \text{TrajRetrieve}(\mathcal{T}, \mathcal{H}). \quad (6)$$

The retrieval considers both topic-level similarity and sequence-level similarity: the former captures whether a reference trajectory describes a similar type of event, while the latter captures whether its event evolution follows a similar temporal pattern. Details of the repository construction and trajectory retrieval are provided in Appendix A.2.

For each retrieved reference trajectory, ATC performs a lightweight monotonic alignment with the current timeline. The alignment preserves temporal order and maps observed milestones in $\mathcal{T}$ to semantically compatible stages in the reference trajectory. Reference stages that are weakly matched or unmatched in $\mathcal{T}$ are treated as analogical gap hypotheses. The stage-2 reflector Reflect_2 then converts these hypotheses into retrieval-oriented questions:

$$\mathcal{Q}^{(2)} = \text{Reflect}_2(\mathcal{T}, \mathcal{R}). \quad (7)$$

Different from Stage 1, which inspects local adjacent transitions in the current timeline, Stage 2 uses cross-event structural priors from analogous historical trajectories to guide retrieval toward potentially missing intermediate stages. Importantly, historical trajectories are not used as direct evidence for timeline insertion. They only suggest what may be missing in the target timeline; every inserted event must still be retrieved from target-event-specific web evidence and pass Logical Utility Verification before being added to the working timeline.

Dynamic Ontology-guided Event Completion

Stage 3 in Fig. 2(c) targets micro-level element missingness. Even when a milestone has been discovered, its description may remain under-specified, such as missing the actor, location, trigger, policy response, or outcome. To support systematic completion under open-domain schema drift, ATC maintains a dynamic ontology:

$$\mathcal{O} = \{(c, \mathcal{E}(c))\}, \quad (8)$$

where c denotes an induced event category and $\mathcal{E}(c)$ denotes the typical elements associated with that category. The ontology is initialized from historical trajectories and updated with newly verified target-event evidence. Implementation details are provided in Appendix A.3.

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

Open-TLS	Value
# topics	50
# timelines	50
Avg. duration (days)	4139
Avg. l (dates per timeline)	23
Avg. k (sentences per date)	1.8

For each milestone in the current timeline, ATC maps it to the most compatible category in $\mathcal{O}$ and uses the associated element set as an adaptive completeness checklist. Missing or weakly specified elements are converted into element-oriented retrieval questions, which are then executed by the shared Reflect–Rewrite–Retrieve–Verify–Update loop in Sec. 3.2. Retrieved evidence is accepted only when LUV confirms that it directly answers the element-level question. Verified evidence is used to refine an existing milestone or, when it supports a temporally distinct update, to insert a new date-grounded event.

3.4 Global Timeline Consolidation

After all three stages are completed, we apply a global consolidation step to transform the accumulated verified events into the final milestone-level timeline. Since open-domain retrieval may introduce redundant or overlapping event descriptions, we first group all accepted events by calendar date according to their extracted timestamps $t(x)$. For each date group, we then invoke an LLM to merge multiple event descriptions $\{\hat{y}(x)\}$ into a single milestone summary y_i that preserves complementary details while removing duplicates and resolving minor phrasing variations. This LLM-based consolidation aligns event-centric insertions with milestone-centric TLS outputs, improving readability and reducing redundancy under high-frequency streams. The consolidated milestones are finally sorted by date to form the output timeline.

4 Experiments

4.1 Experimental Setup

Dataset. We conduct experiments on **Open-TLS benchmark** (Wu et al., 2025), a benchmark specifically crafted for open-domain timeline summarization. Open-TLS contains 50 topics, each associated with an open-domain timeline. On av-

Table 2: Main results on the Open-TLS task. Metrics include ROUGE-1 (R1) and ROUGE-2 (R2) variants for different evaluation settings. The best results for each backbone are highlighted in **bold**.

Model	Method	ConcatF1		AgreeF1		AlignF1		DateF1
		R1	R2	R1	R2	R1	R2	
Qwen3-8B	DIRECT	0.2653	0.0776	0.0584	0.0227	0.0692	0.0245	0.1908
	CHRONOS	0.2967	0.0864	0.0725	0.0308	0.0896	0.0334	0.2326
	R2A-TLS	0.3124	0.1117	0.0956	0.0453	0.1102	0.0498	0.2685
	ATC (Ours)	0.3558	0.1294	0.1147	0.0566	0.1325	0.0593	0.3157
Qwen3-32B	DIRECT	0.3056	0.0883	0.0827	0.0345	0.1224	0.0386	0.2752
	CHRONOS	0.3725	0.1148	0.1094	0.0467	0.1296	0.0483	0.3274
	R2A-TLS	0.4097	0.1473	0.1416	0.0624	0.1528	0.0716	0.3955
	ATC (Ours)	0.4271	0.1587	0.1504	0.0705	0.1644	0.0815	0.4437

erage, a topic spans 4,139 days, and a reference timeline contains $l = 23$ dates, with each date accompanied by $k = 1.8$ sentences describing the events at that time. Table 1 summarizes the dataset statistics.

Models. We instantiate all LLM-based components with two representative backbones: **Qwen3-8B** (small) and **Qwen3-32B** (large). The same backbone is used for (i) reflection question generation, (ii) Q2Q rewriting, (iii) Logical Utility Verification, (iv) event extraction, and (v) global consolidation. Unless stated otherwise, we use a temperature of 0.1 for all LLM-based components to reduce generation variance.

Baselines. We compare ATC against three baselines: **DIRECT**, a minimal pipeline that directly searches the web with the topic query and summarizes the retrieved news into a timeline without any reflection or iterative completion; **CHRONOS**, an iterative search-then-summarize framework; and **R2A-TLS**, a reflective TLS approach with feedback-driven retrieval and verification. For fairness, we use the same search backend across methods and align the retrieval footprint whenever applicable (e.g., comparable question/iteration budgets and the same number of Q2Q rewrites per question).

Event-centric handling and verification. Different from passage-based RAG pipelines, we extract events directly from retrieved documents. Each candidate event is represented as $x = (t(x), \hat{y}(x))$, where $t(x)$ is the extracted timestamp (or interval) and $\hat{y}(x)$ is a concise event description. A candidate event is accepted only if it passes question-conditioned Logical Utility Verification, otherwise it is discarded. Accepted events are inserted into the working timeline in chronological

order.

Reproducibility under open-domain retrieval. Since the open web is mutable, we cache retrieved URLs/documents and run all methods on the same cached snapshot.

Evaluation metrics. We adopt the official Open-TLS evaluation metrics, including **ConcatF1**, **AgreeF1**, **AlignF1**, and **DateF1**. The first three metrics are ROUGE-style overlap scores computed under different date-alignment assumptions: (i) **ConcatF1** concatenates all date summaries in a timeline into a single document and computes ROUGE between the concatenated prediction and reference, measuring overall content coverage; (ii) **AgreeF1** computes ROUGE only on summaries whose dates exactly match between prediction and reference, emphasizing factual agreement on the same time points; (iii) **AlignF1** first aligns predicted date summaries to reference summaries based on both textual similarity and temporal proximity, then computes ROUGE on the aligned pairs, penalizing alignments that require large temporal shifts. Finally, **DateF1** evaluates temporal accuracy by computing the F1 score between the set of predicted dates and the set of reference dates. Following the Open-TLS evaluation protocol, we report both ROUGE-1 (R1) and ROUGE-2 (R2) F1 scores for Concat, Agree, and Align.

4.2 Main Results

Table 2 reports the main results on Open-TLS under both Qwen3-8B and Qwen3-32B backbones. Across all evaluation settings, ATC consistently outperforms DIRECT, CHRONOS, and the strongest baseline R2A-TLS, demonstrating the effectiveness of reflection-driven completion in the open-web setting. With Qwen3-8B, ATC improves over R2A-TLS by 13.89%/15.85% on

ConcatF1 (R1/R2), 19.98%/24.94% on AgreeF1, and 20.24%/19.08% on AlignF1, indicating better content coverage and more reliable alignment to reference timelines. ATC also yields a substantial gain on DateF1 (0.2685→0.3157, +17.58%), suggesting that question-conditioned verification helps suppress temporally inconsistent updates under retrieval uncertainty. With the stronger Qwen3-32B backbone, ATC remains best on all metrics and further improves over R2A-TLS, achieving 4.25%/7.74% on ConcatF1, 6.21%/12.98% on AgreeF1, 7.59%/13.83% on AlignF1, and 12.19% on DateF1. While scaling the backbone from Qwen3-8B to Qwen3-32B boosts overall performance, ATC distinguishes itself through superior results in AgreeF1, AlignF1, and DateF1. This excellence stems from its ability to better recover missing intermediate stages and minimize spurious timeline jumps, outperforming traditional iterative search-and-summarize pipelines.

To assess the robustness of the improvements, we further conduct a paired topic-level statistical analysis against R2A-TLS on four pre-specified primary metrics. ATC shows significant positive shifts on all four metrics under the paired Wilcoxon signed-rank test after Holm correction, while DateF1 additionally exhibits a paired-bootstrap 95% confidence interval entirely above zero. Full results are reported in Appendix B.4.

4.3 Ablation Study

We conduct ablations on Qwen3-8B to isolate the contribution of each completion stage and Logical Utility Verification (LUV). The evaluated variants include Stage-1 only, adding Stage-2 historical trajectory analogy, the full system with Stage-3 dynamic ontology-guided completion, and full ATC without LUV. All variants use the same budgets ($B_s = 5$, $Q_{\max} = 5$), three Q2Q rewrites per question, and the same acceptance threshold ($\gamma = 0.8$).

As shown in Table 3, adding Stage-2 improves alignment and date accuracy over Stage-1, suggesting that historical trajectory analogy helps recover structurally missing intermediate stages. Adding Stage-3 further improves agreement- and date-oriented metrics, showing that ontology-guided element completion produces more verifiable milestone updates. Removing LUV substantially degrades AgreeF1, AlignF1, and DateF1 despite a slightly higher ConcatF1, confirming that question-conditioned filtering is necessary to prevent topically related but gap-irrelevant candidate

events from entering the timeline.

4.4 Event-level Human Evaluation

The automatic Open-TLS metrics primarily measure content overlap and temporal alignment and do not directly evaluate factual support, gap relevance, or local timeline coherence. We therefore conduct a blinded event-level human evaluation on 52 candidate events sampled from a 10-topic diagnostic rerun, including 17 LUV-accepted and 35 LUV-rejected candidates across all three completion stages.

Two independent annotators are shown the topic, gap question, available preceding timeline context, candidate event, source metadata, and supporting evidence passage. The stage identity, LUV score, accept/reject decision, system identity, and method configuration are hidden. Each candidate is rated on three dimensions using a 0–2 ordinal scale: factual support, gap relevance, and local timeline coherence. We report the proportion of judgments receiving the maximum score for each dimension, as well as a strict Valid Completion Rate for which all three dimensions receive the maximum score from the same annotator. As shown in Table 4, LUV-accepted candidates score higher on all four measures. The clearest separation occurs in Direct Gap Filling, which increases from 24.3% for rejected candidates to 58.8% for accepted candidates (+34.5 percentage points). Accepted candidates also achieve higher Full Support, Coherence Improvement, and Valid Completion rates, although their absolute values remain modest.

Inter-annotator agreement is high, with quadratic weighted Cohen’s κ values of 0.917 for factual support, 0.866 for gap relevance, and 0.863 for local timeline coherence. These results indicate that LUV preferentially retains candidate events that directly address the identified information need. At the same time, the relatively modest absolute factual-support rate reinforces the intended scope of LUV: it is a logical-utility filter rather than a complete source-entailment verifier.

5 Conclusion

We presented Active Timeline Completion (ATC), an iterative active-retrieval framework for Open-Domain Timeline Summarization (Open-TLS). ATC mitigates retrieval uncertainty by convert-

Table 3: Ablation results on the Open-TLS task. All variants are evaluated using the same backbone and hyperparameters ($B_s=5$, $Q_{\max}=5$, $\gamma=0.8$). **Full ATC** denotes our complete framework. Best results are highlighted in **bold**.

Variant	ConcatF1		AgreeF1		AlignF1		DateF1
	R1	R2	R1	R2	R1	R2	
Model: Qwen3-8B							
Stage-1 (Parametric Causal Gap Detection)	0.3079	0.1048	0.0846	0.0387	0.1006	0.0409	0.2518
+ Stage-2 (Historical Trajectory Analogy)	0.3364	0.1197	0.0976	0.0422	0.1238	0.0546	0.2927
+ Stage-2 + Stage-3 (Dynamic Ontology)	0.3558	0.1294	0.1147	0.0566	0.1325	0.0593	0.3157
Full ATC w/o LUV	0.3623	0.1328	0.0865	0.0317	0.0946	0.0335	0.2358

Table 4: Blinded event-level human evaluation of LUV-accepted and rejected candidates.

Candidate Set	N	Support	Gap Fill	Coherence	Valid
Accepted	17	29.4	58.8	14.7	5.9
Rejected	35	11.4	24.3	2.9	0.0
Δ	–	+18.0	+34.5	+11.8	+5.9

ing missingness into explicit information needs and resolving them through a shared completion loop. To progressively strengthen completion priors, ATC integrates three stage-specific reflectors: Stage-1 *Parametric Causal Gap Detection*, Stage-2 *Historical Trajectory Analogy* for recovering structurally missing stages, and Stage-3 *Dynamic Ontology*-guided element completion to fill under-specified milestone facts under schema drift. A question-conditioned *Logical Utility Verification* (LUV) module filters topically related but logically unhelpful retrieval results, while an LLM-based global consolidation step groups same-date events into readable milestones. On Open-TLS, ATC improves over the strongest baseline R2A-TLS by 13.89%/15.85% on ConcatF1 (R1/R2), 19.98%/24.94% on AgreeF1, 20.24%/19.08% on AlignF1, and 17.58% on DateF1 with Qwen3-8B; with Qwen3-32B, the gains remain consistent at 4.25%/7.74% (ConcatF1), 6.21%/12.98% (AgreeF1), 7.59%/13.83% (AlignF1), and 12.19% (DateF1).

Several directions remain open. Future work can reduce the computational cost of iterative retrieval via adaptive budgeting and calibrated early stopping, improve robustness to misinformation and conflicting reports through multi-source consistency checks, and further strengthen Stage-2 by extending trajectory analogy retrieval with richer structural representations. We hope ATC provides a practical foundation for reliable and logically grounded timeline construction from open-domain

evidence.

Limitations

ATC has several practical limitations. Its iterative completion loop introduces additional computation compared with one-pass generation, since each cycle involves gap reflection, Q2Q rewriting, retrieval, event extraction, and LUV-based verification. Although this cost is bounded by stage budgets and early stopping, improving efficiency through adaptive budgeting and intermediate-result reuse remains important. ATC also depends on open-web evidence converted from URLs to text. Dynamically rendered pages, complex layouts, source availability, and regional language coverage may affect evidence accessibility and thus timeline completeness. Another limitation lies in the use of historical trajectories as analogical priors. When the target topic is emerging, atypical, or poorly covered by the repository, trajectory-guided questions may become less effective. Future work can address these issues by improving evidence parsing, expanding trajectory coverage, and developing fallback strategies with weaker dependence on historical analogies.

Acknowledgments

This work was supported by the Key Research and Development Project of Hunan Province (No.2025JK2119), Foundation of NUDT (HQKYZH2025KD004), the Leading Science and Technology Innovation Talents Program of Furong Project (2025RC1048).

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 1018, New

- York, NY, USA. Association for Computing Machinery.
- Asai, Akari, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. [Self-rag: Learning to retrieve, generate, and critique through self-reflection](#). *Preprint*, arXiv:2310.11511.
- Bao, Chenlong, Shijie Li, Minghao Hu, Ming Qiao, Bin Zhang, Jin-Tao Tang, Shasha Li, and Ting Wang. 2025. [R2A-TLS: Reflective retrieval-augmented timeline summarization with causal-semantic integration](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 752–766, Suzhou, China. Association for Computational Linguistics.
- Cao, Pengfei, Yupu Hao, Yubo Chen, Kang Liu, Jiexin Xu, Huaijun Li, Xiaojian Jiang, and Jun Zhao. 2023. [Event ontology completion with hierarchical structure evolution networks](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 306–320, Singapore. Association for Computational Linguistics.
- Dao, Viet, Van-Cuong Pham, Quyen Tran, Thanh-Thien Le, Linh Van Ngo, and Thien Huu Nguyen. 2024. [Lifelong event detection via optimal transport](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12610–12621, Miami, Florida, USA. Association for Computational Linguistics.
- Feder, Amir, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. 2022. [Causal inference in natural language processing: Estimation, prediction, interpretation and beyond](#). *Transactions of the Association for Computational Linguistics*, 10:1138–1158.
- Gholipour, Demian, 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, Online. Association for Computational Linguistics.
- Hu, Qisheng, 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.
- Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20*, Red Hook, NY, USA. Curran Associates Inc.
- Li, Sha, Ruining Zhao, Manling Li, Heng Ji, Chris Callison-Burch, and Jiawei Han. 2023. [Open-domain hierarchical event schema induction by incremental prompting and verification](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5677–5697, Toronto, Canada. Association for Computational Linguistics.
- Min, Qingkai, Zitian Qu, Qipeng Guo, Xiangkun Hu, Zheng Zhang, and Yue Zhang. 2025. [Multi-document event extraction using large and small language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19254–19285, Suzhou, China. Association for Computational Linguistics.
- Qiu, Zitai, Congbo Ma, Jia Wu, and Jian Yang. 2025. [Text is all you need: LLM-enhanced incremental social event detection](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4666–4680, Vienna, Austria. Association for Computational Linguistics.
- Muhammad Reza Qorib, Qisheng Hu, and Hwee Tou Ng. 2025. [Just what you desire: Constrained timeline summarization with self-reflection for enhanced relevance](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):25065–25073.
- Sim, Tiviatis, Kaiwen Yang, Shen Xin, and Kenji Kawaguchi. 2025. [Who’s important? – sunset: Synergistic understanding of stakeholder, events and time for timeline generation](#). *Preprint*, arXiv:2507.21903.
- Wang, Nengbo, Xiaotian Han, Jagdip Singh, Jing Ma, and Vipin Chaudhary. 2025. [CausalRAG: Integrating causal graphs into retrieval-augmented generation](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22680–22693, Vienna, Austria. Association for Computational Linguistics.
- Weller, Orion, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. 2026. On the theoretical limitations of embedding-based retrieval. In *International Conference on Learning Representations*, volume 2026, pages 11202–11216.
- Wu, Weiqi, Shen Huang, Yong Jiang, Pengjun Xie, Fei Huang, and Hai Zhao. 2025. [Unfolding the headline: Iterative self-questioning for news retrieval and timeline summarization](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4385–4398, Albuquerque, New Mexico. Association for Computational Linguistics.
- Xu, Nan, Hongming Zhang, and Jianshu Chen. 2024. [CEO: Corpus-based open-domain event ontology induction](#). In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 946–964, St. Julian’s, Malta. 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](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 377–387, Online. Association for Computational Linguistics.

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.

Xinliang Frederick Zhang, Carter Blum, Temma Choji, Shalin Shah, and Alakananda Vempala. 2024. [UL-TRA: Unleash LLMs’ potential for event argument extraction through hierarchical modeling and pairwise self-refinement](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8172–8185, Bangkok, Thailand. Association for Computational Linguistics.

A Reproducibility and Implementation Details

A.1 Shared Completion Loop and Configuration

This section provides the implementation details of the shared completion loop used by all ATC stages. Table 5 lists the default configuration used in our experiments, including the stage budget, question cap, query rewriting budget, LUV threshold, and update constraints. Algorithm 1 summarizes how each stage executes gap-oriented retrieval, verification, and timeline update under this configuration.

Table 5: Default configuration of ATC.

Configuration	Value
Stage iteration budget B_s	5
Question cap per iteration $Q_{\max}$	5
Q2Q rewrites per question	3
LUV threshold γ	0.8
Maximum accepted events per question m	3
URL-level deduplication	Yes
Final same-date consolidation	Yes

A.2 Historical Trajectory Repository and Retrieval

This subsection provides implementation details for the historical trajectory repository and the re-

Algorithm 1 Shared ATC Completion Loop for Stage s

Require: Working timeline $\mathcal{T}$, stage-specific reflector Reflect_s , stage-specific state $\mathcal{S}^{(s)}$, open web $\mathcal{W}$, stage budget B_s , question cap $Q_{\max}$, LUV threshold γ , maximum accepted events m

Ensure: Updated timeline $\mathcal{T}$

```

1: for  $r = 1$  to  $B_s$  do
2:   Generate gap questions  $Q_r^{(s)} \leftarrow \text{Reflect}_s(\mathcal{T}, \mathcal{S}^{(s)})$ .
3:   Keep at most  $Q_{\max}$  questions from  $Q_r^{(s)}$ .
4:   if  $Q_r^{(s)} = \emptyset$  then
5:     break
6:   end if
7:    $\Delta\mathcal{T} \leftarrow \emptyset$ 
8:   for each question  $q^* \in Q_r^{(s)}$  do
9:     Generate search queries  $\{k_1, k_2, k_3\} \leftarrow \text{Q2Q}(q^*)$ .
10:    Retrieve documents  $\mathcal{D}_{q^*}$  from  $\mathcal{W}$  with all queries.
11:    Remove duplicate URLs from  $\mathcal{D}_{q^*}$ .
12:    Extract candidate events  $\mathcal{X}_{q^*} = \{x = (t(x), \hat{y}(x))\}$  from  $\mathcal{D}_{q^*}$ .
13:    Score each candidate by LUV:  $u(x, q^*) = \text{Judge}(q^*, x)$ .
14:    Select  $\mathcal{A}_{q^*} \leftarrow \text{TopM}_m(\{x \in \mathcal{X}_{q^*} \mid u(x, q^*) \geq \gamma\})$ .
15:     $\Delta\mathcal{T} \leftarrow \Delta\mathcal{T} \cup \mathcal{A}_{q^*}$ .
16:   end for
17:   if  $\Delta\mathcal{T} = \emptyset$  then
18:     break
19:   end if
20:   Insert events in  $\Delta\mathcal{T}$  into  $\mathcal{T}$  by timestamp.
21:   Update  $\mathcal{S}^{(s)}$  with accepted events when applicable.
22: end for
return  $\mathcal{T}$ 

```

trieval procedure used in Stage 2. The goal of Stage 2 is not to transfer events from previous timelines into the target timeline, but to use analogous event evolutions as structural priors for generating better gap questions.

Repository construction. We construct the historical trajectory repository $\mathcal{H}$ from the open-source DTELS-Bench dataset. Each trajectory is represented as an ordered sequence of date-grounded milestones:

$$\tau = \{(d_1, s_1), \dots, (d_n, s_n)\},$$

where d_i is a normalized date and s_i is the corresponding milestone summary. To make the repository compatible with Open-TLS, we apply a distribution-aware filtering procedure. Specifically, we retain timelines whose event types are broadly aligned with Open-TLS topics, including political events, international conflicts, public crises, entertainment-related events, corporate events, and social events. We further filter candidate trajectories by temporal span, milestone

count, temporal density, and stage clarity. Trajectories with missing dates, unstable ordering, fragmented descriptions, or ambiguous topic boundaries are removed. This process yields 200 complete historical trajectories, with an average of 28 milestone nodes per trajectory; the longest trajectory contains 37 nodes and the shortest contains 15 nodes.

Overlap and leakage audit. Because Stage 2 uses historical trajectories from DTELS-Bench as structural priors, we further examine whether the historical repository overlaps with the Open-TLS evaluation set at the event-instance level. We cross-check the 50 Open-TLS evaluation topics against the 200 historical trajectories retained in $\mathcal{H}$ and find no identical real-world event topics. Here, an instance-level overlap refers to a case in which an Open-TLS evaluation topic and a historical trajectory describe the same underlying real-world event and its associated event evolution, rather than merely sharing similar event structures or domains.

Table 6 summarizes the audit. Open-TLS reference timelines, annotations, system outputs, and evaluation scores are not used to construct or select the historical trajectory repository. Moreover, events appearing in historical trajectories are never directly inserted into the target timeline or treated as factual evidence for an Open-TLS topic. Structural similarity between trajectories is expected by design and is not considered instance-level overlap; Stage 2 uses such similarity only to hypothesize potentially missing stages and formulate retrieval questions.

This audit establishes a clear boundary between *structural transfer* and *factual transfer*. Historical trajectories may affect what ATC searches for by suggesting abstract stages, temporal orderings, or causal relations that could be missing from the current timeline. They do not determine what occurred in the target event. Every event added to an Open-TLS timeline must still be independently extracted from target-specific retrieved web documents and pass Logical Utility Verification before updating the working timeline.

Trajectory representation and retrieval. Given a current partial timeline $\mathcal{T}$, ATC retrieves reference trajectories from $\mathcal{H}$ using both topic-level and sequence-level signals. For the topic-level signal, we encode the Open-TLS topic description and compare it with the topic

Table 6: Overlap and leakage audit between the Open-TLS evaluation set and the historical trajectory repository constructed from DTELS-Bench.

Audit Dimension	Criterion	Result
Topic-level overlap	Same underlying real-world event and event evolution	0
Reference leakage	Open-TLS reference timelines used to construct or select $\mathcal{H}$	No
Annotation leakage	Open-TLS labels or annotations used during repository construction	No
Output leakage	Open-TLS system outputs used during repository construction or selection	No
Metric leakage	Open-TLS evaluation scores used to select historical trajectories	No
Direct event transfer	Historical events directly inserted into target timelines	No
Evidence usage	Historical trajectories treated as factual evidence for target events	No

description of each historical trajectory. For the sequence-level signal, we aggregate the milestone summaries in $\mathcal{T}$ into a trajectory-level text representation and compare it with the aggregated milestone representation of each historical trajectory. The final retrieval score combines these two signals, so that the selected references are not only topically relevant but also structurally similar in terms of event evolution. We then retrieve the top- K trajectories as the analogical reference set $\mathcal{R}$.

Monotonic trajectory alignment. After retrieving $\mathcal{R}$, ATC aligns each reference trajectory with the current partial timeline. The alignment is monotonic: it preserves temporal order and prevents later reference stages from being matched to earlier target milestones. This constraint reflects the fact that both the target timeline and the historical references describe event evolution over time. During alignment, observed milestones in $\mathcal{T}$ are mapped to semantically compatible stages in the reference trajectory. Reference stages that are weakly matched or unmatched are treated as potential structural gaps. These gaps are then converted by the Stage 2 reflector into retrieval-oriented questions, which are executed by the shared Reflect–Rewrite–Retrieve–Verify–Update loop.

Evidence boundary. Retrieved historical trajectories are used only as analogical priors for question generation. They are never used as direct evidence for inserting events into the target timeline. A newly proposed event can be added to the working timeline only if it is retrieved from target-event-specific web evidence and passes Logical Utility Verification. This design reduces the risk of false analogy: historical trajectories help ATC ask better questions, but they do not determine the factual content of the target timeline.

A.3 Dynamic Ontology Induction and Stage-3 Completion

This subsection provides implementation details for the dynamic ontology used in Stage 3. While the main text uses a lightweight event representation $x = (t(x), \hat{y}(x))$, our implementation additionally stores optional metadata for bookkeeping:

$$x^{\text{meta}} = (t(x), \hat{y}(x), \phi(x), e(x)),$$

where $t(x)$ is the timestamp, $\hat{y}(x)$ is the event description, $\phi(x)$ is an optional element dictionary, and $e(x)$ is the source evidence record from which the candidate event is extracted. These metadata fields are used for provenance tracking, ontology updates, and later analysis, but they are not passed to Logical Utility Verification (LUV). Consistent with the main formulation, LUV scores only the extracted candidate event against the current gap question:

$$u(x, q^*) = \text{Judge}(q^*, x).$$

Thus, evidence grounding is handled by the preceding retrieval and event extraction steps, whereas LUV focuses on whether the extracted event is logically useful for repairing the current timeline gap.

Since Open-TLS covers open-domain topics, ATC does not assume a predefined event schema. Instead, it maintains a dynamic ontology $\mathcal{O} = \{(c, \mathcal{E}(c))\}$, where c denotes an induced event category and $\mathcal{E}(c)$ denotes the set of typical elements associated with that category. The initial ontology is induced from the curated historical trajectories in $\mathcal{H}$ by grouping recurring milestone types and extracting commonly observed elements. These historical trajectories provide reusable category and element priors, but they are not used as factual evidence for the target timeline. During iterative completion, newly accepted target-event candidate

events are used to update the ontology: compatible events are assigned to existing categories, while unmatched events may introduce provisional new categories.

To keep the ontology compact and stable, ATC applies LLM-assisted canonicalization to merge semantically equivalent categories and synonymous elements. Low-support categories and infrequent elements are pruned to reduce schema noise and prevent uncontrolled schema expansion. In practice, the ontology update is budgeted by retaining only high-support categories and representative elements for Stage 3 question generation.

During Stage 3 completion, each milestone in the current timeline is mapped to the most compatible category in $\mathcal{O}$. The associated element set $\mathcal{E}(c)$ is used as an adaptive completeness checklist. Missing or weakly specified elements are converted into targeted questions and processed by the shared Reflect–Rewrite–Retrieve–Verify–Update loop. Candidate events extracted from retrieved documents are filtered by LUV according to their question-conditioned logical utility. An accepted candidate event refines an existing milestone when it supplies missing details for the same date-grounded event, or inserts a new milestone when it corresponds to a temporally distinct update.

B Additional Analysis

To better understand the behavior, efficiency, and generality of ATC, we conduct three additional analyses. First, we study the effect of the query rewriting budget to justify the default number of rewritten queries per question. Second, we analyze the sensitivity of Logical Utility Verification (LUV) to different acceptance thresholds. Finally, we evaluate ATC on a closed-domain timeline summarization setting to examine whether the proposed active completion framework generalizes beyond open-domain retrieval.

B.1 Query Rewriting Budget Sensitivity

Query rewriting controls how each reflective question is converted into executable search queries. Although a larger rewriting budget can potentially improve retrieval coverage, it also increases search operations, LLM calls, and runtime. We therefore analyze the effectiveness–cost trade-off under different maximum numbers of rewritten queries per question.

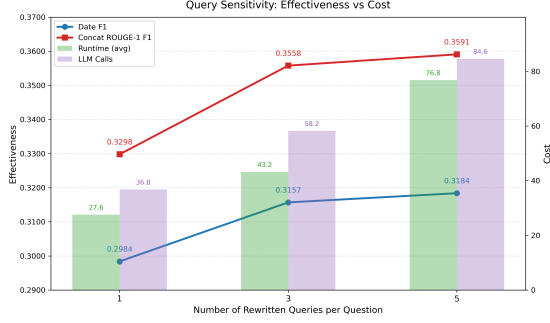


Figure 3: Effectiveness–cost trade-off under different query rewriting budgets. Line plots report DateF1 and Concat ROUGE-1 F1, while bars report average runtime and LLM calls.

As shown in Figure 3, increasing the rewriting budget from one to three queries brings a clear improvement in both effectiveness metrics, suggesting that multiple query formulations are useful for covering diverse expressions of missing evidence in open-web sources. However, further increasing the budget to five queries yields only limited additional gains, while the runtime and LLM-call overhead increase substantially. This diminishing-return pattern indicates that three rewritten queries provide a practical balance between retrieval coverage and computational efficiency. Therefore, we use 3 as the default maximum number of rewritten queries per question in ATC.

B.2 Sensitivity to Logical Utility Verification Threshold

As shown in Table 7, ATC is sensitive to overly permissive or overly strict LUV thresholds. Without LUV, the model obtains relatively high Concat scores but suffers clear drops in agreement, alignment, and date accuracy, indicating that unverified retrieved content may improve surface overlap while harming logical and temporal quality. Among the tested thresholds, $\gamma = 0.8$ achieves the best overall performance. A lower threshold accepts more noisy or weakly useful updates, whereas a higher threshold filters out useful long-tail evidence and reduces recall. This confirms the role of LUV in controlling update quality and supports our default choice of $\gamma = 0.8$.

B.3 Generalization to Closed-domain Timeline Summarization

The closed-domain results show that ATC obtains much smaller gains than in the open-domain setting and does not consistently outperform all

Table 7: Sensitivity analysis of the LUV threshold γ .

Setting	ConcatF1	AgreeF1	AlignF1	DateF1
w/o LUV	0.3623	0.0865	0.0946	0.2358
$\gamma = 0.7$	0.3589	0.1056	0.1213	0.2926
$\gamma = 0.8$	0.3558	0.1147	0.1325	0.3157
$\gamma = 0.9$	0.3496	0.1092	0.1268	0.3019

Table 8: Generalization results on closed-domain timeline summarization datasets.

Dataset	Method	Align R-1	Align R-2	DateF1
Crisis	CHRONOS	0.108	0.045	0.323
	R2A-TLS	0.112	0.046	0.331
	ATC	0.111	0.047	0.328
T17	CHRONOS	0.118	0.036	0.528
	R2A-TLS	0.119	0.042	0.522
	ATC	0.121	0.042	0.519

closed-domain baselines on DateF1. This trend is expected because closed-domain TLS provides a fixed topic-specific document collection, which limits the opportunity for active evidence discovery. In this setting, ATC can still use reflection and LUV to better organize and filter candidate evidence, but it cannot recover missing long-tail milestones if the provided corpus does not contain them.

Therefore, the closed-domain experiment should be interpreted as a stress test of ATC under constrained evidence coverage rather than as evidence that ATC always generalizes across retrieval settings. The results suggest that ATC is most beneficial when the evidence space contains recoverable but initially missed information, especially in open-domain scenarios with noisy, redundant, and incomplete web sources. When the evidence pool is already fixed or insufficient, the advantage of active retrieval naturally becomes limited.

B.4 Topic-level Statistical Significance Analysis

To further assess whether the improvements of ATC over the strongest baseline are systematic across topics, we conduct a paired topic-level significance analysis on four pre-specified primary metrics: **Concat R2**, **Agree R2**, **Align R2**, and **DateF1**. For each metric, the ATC and R2A-TLS scores are paired by the same Open-TLS topic.

Under the directional hypothesis that ATC improves over R2A-TLS, we apply a one-sided paired Wilcoxon signed-rank test to examine

whether the topic-level differences exhibit a systematic positive shift. In addition, we compute paired bootstrap 95% confidence intervals (CIs) for the mean differences across topics to quantify the uncertainty of the average improvements. The Wilcoxon test and bootstrap interval characterize different aspects of the paired results: the former evaluates the rank-based direction of topic-level differences, whereas the latter measures uncertainty in the mean score difference.

As shown in Table 9, ATC achieves higher mean scores than R2A-TLS on all four primary metrics. The one-sided paired Wilcoxon tests show significant positive topic-level shifts for all four metrics, with raw p -values ranging from 0.0025 to 0.0124. All four comparisons remain significant after Holm correction for multiple testing.

The paired bootstrap analysis provides a complementary view of the uncertainty in the mean improvements. For Concat R2, Agree R2, and Align R2, the estimated mean differences are positive, but their 95% bootstrap confidence intervals include zero. This indicates that although the topic-level differences exhibit significant positive rank shifts under the Wilcoxon test, the corresponding average improvements are less stable under bootstrap resampling. In contrast, DateF1 achieves a mean improvement of 0.0482 with a 95% confidence interval of [0.0024, 0.0947], which lies entirely above zero.

These results therefore distinguish systematic topic-level improvement from uncertainty in the average effect size. Overall, ATC exhibits significant positive topic-level shifts over R2A-TLS on all four primary metrics, while the strongest bootstrap-supported mean improvement is observed on DateF1.

C Case Study

Table 10 presents a case study on the topic query “*Astronauts Stuck*” (Boeing Starliner). We excerpt key milestones spanning early program development and the 2024 crewed test flight, and show stage-specific questions, accepted updates, and LUV-rejected candidates.

ATC refines the timeline through three reflection stages. In **Stage-1** (Parametric Causal Gap Detection), the model detects a causal break between “docking with anomalies” and “the delayed return plan”, and thus asks decision-oriented questions about schedule changes driven

by thruster/helium issues. This triggers targeted retrieval and inserts the missing operational bridge at **2024-06-18** (return postponed to June 26). Meanwhile, LUV rejects topically related but gap-irrelevant candidates (e.g., **2024-06-15**, **2024-08-25**), preventing noisy updates from polluting the causal narrative.

In **Stage-2** (Historical Trajectory Analogy), ATC addresses a structural omission: semantic retrieval can jump from early certification concerns to the OFT-1 failure without intermediate test-stage anomalies. Guided by trajectory-level priors, Stage-2 queries for missing test-stage events between the **2018-05-01** concern and the **2019-12-20** failure, while LUV filters tangential ISS-related records (e.g., **2019-10-01**) that do not resolve the completion gap.

Finally, **Stage-3** (Ontology-Guided Element Completion) enriches milestones with verifiable elements. For the docking incident, element-oriented questions yield a strengthened update at **2024-06-06** with checkable elements (docking after ~ 26 hours; helium leaks; thrusters offline), closely matching the reference. LUV further rejects generic paraphrases of the same incident that lack the required element-level evidence. Overall, the case study illustrates how staged reflection plus question-conditioned utility filtering improves structural coherence and temporal fidelity under open-domain noise.

D Prompt Templates

The prompt templates for the LLM-based modules in ATC are shown in Listings 1–4. Listings 1 and 2 present the prompts for stage-wise reflective gap discovery, where the model identifies missing temporal nodes from historical trajectory analogies and timeline ontology elements. Listings 3 and 4 show the prompts for question-to-query rewriting and logical utility verification, respectively.

D.1 Stage-2 Historical Trajectory Analogy Prompt

You are a senior analyst specializing in temporal causality verification.

Reference Knowledge
Use the following timelines. Observe the causal chains, such as:
[Hidden Risks -> Trigger Event -> Panic/
Contagion -> Regulatory Intervention -> Market
Reaction -> Resolution]

Table 9: Paired topic-level significance analysis of ATC versus R2A-TLS on the four pre-specified primary metrics. Raw p -values are obtained using a one-sided paired Wilcoxon signed-rank test under the directional hypothesis that ATC improves over R2A-TLS. The last column reports paired bootstrap 95% confidence intervals for the mean differences.

Primary Metric	R2A-TLS Mean	ATC Mean	Difference	Raw p	95% Bootstrap CI
Concat R2	0.1473	0.1587	+0.0114	0.0124	[−0.0233, 0.0418]
Agree R2	0.0624	0.0705	+0.0081	0.0025	[−0.0175, 0.0308]
Align R2	0.0716	0.0815	+0.0099	0.0070	[−0.0196, 0.0365]
DateF1	0.3955	0.4437	+0.0482	0.0087	[0.0024, 0.0947]

```

Reference Timelines:
{reference_timelines}

# Task
You need to examine the timeline of a target
news event or entity.

Target News/Event Entity:
{query}

Current Draft Timeline:
{timeline}

1. Comparative Analysis: Compare the logic
of the `Current Draft Timeline` with that of the
`Reference Timelines`. Identify structural
similarities and potentially missing causal
links.
    For example, if a bank failure appears in the
    target timeline, check whether subsequent
    regulatory responses or market contagion steps
    are missing, as these may appear in the
    reference timelines.

2. Gap Detection: Based on this analogy,
infer potential missing temporal nodes in the
target timeline.

3. Question Generation: Construct retrieval
questions based on the identified gaps to
retrieve these missing nodes. Focus on event
origins, development processes, key figures, and
policy responses.

# Output
## Output Format:
["Question 1", "Question 2", "Question 3", ...]

Directly output your questions in the specified
format.

```

Listing 1: Prompt template for Stage-2 historical trajectory analogy.

D.2 Stage-3 Ontology-Guided Completion Prompt

```

You are a senior analyst specializing in
timeline completeness verification.

# Reference Knowledge
Use the following timeline ontology elements to
examine whether the current timeline contains

```

```

the key information required for a complete
event timeline.

Timeline Ontology Elements:
{frame_elements}

# Task
You need to examine the timeline of a target
news event or entity.

Target News/Event Entity:
{query}

Current Draft Timeline:
{timeline}

1. Ontology-based Completeness Analysis:
Compare the `Current Draft Timeline` against the
`Timeline Ontology Elements`. Check whether the
key elements required for a complete timeline
are sufficiently covered.

2. Gap Detection: Identify temporal nodes,
event stages, or ontology elements that are
missing, weakly described, or under-specified in
the current timeline.

3. Question Generation: Construct retrieval
questions based on the identified gaps to
retrieve the missing or under-specified
information. Each question should correspond to
a specific missing ontology element, missing
event stage, or incomplete temporal node. Do not
force a fixed number of questions; generate
only as many questions as needed according to
the detected gaps. Use concrete entities, dates,
actions, locations, organizations, policies, or
event names when available, and avoid redundant,
vague, or unsupported questions.

# Output
## Output Format:
["Question 1", "Question 2", "Question 3", ...]

Directly output your questions in the specified
format.

```

Listing 2: Prompt template for Stage-3 ontology-guided completion.

D.3 Q2Q Rewriting Prompt

You are a senior analyst specializing in search query rewriting for timeline completion.

Task

You need to rewrite the user question into 3 concise English search queries that can retrieve relevant evidence for timeline completion.

User Question:

{question}

1. ****Information Need Identification:**** Identify the key entity, event, time, location, actor, action, or policy response requested by the user question.

2. ****Query Rewriting:**** Convert the question into 3 short English search queries suitable for search engine retrieval.

3. ****Query Control:**** Keep each query focused on one clear retrieval target. If the original question contains multiple information needs, split them into separate queries. Avoid redundant, vague, or unsupported rewrites.

#example

1.user Question: When did the initial protests that led to the Egyptian Crisis begin?

output: ["Egyptian Crisis initial protests", "Time of protests lead to Egyptian Crisis", "Egypt protests began January 2011"]

2.user Question: When and where did Robert Jasmiden die?

output: ["Robert Jasmiden's death time", "Robert Jasmiden's death place", "Robert Jasmiden death"]

Output

Output Format:

["Query 1", "Query 2", "Query 3"]

Directly output the list only.

Listing 3: Prompt template for question-to-query rewriting.

D.4 Logical Utility Verification Prompt

You are a senior analyst specializing in logical utility verification for timeline completion.

Task

Score whether the candidate event, extracted from retrieved documents, is logically useful for answering the retrieval question and completing the timeline.

Retrieval Question:

{question}

Candidate Event:

{candidate_event}

Evaluation Criteria

Evaluate the candidate event according to three criteria:

1. Question Answering:

Does the candidate event directly answer the retrieval question?

2. Logical Utility:

Does the candidate event fill a missing causal stage, temporal transition, or event element?

Do not give a high score to an event that is merely topically related but does not repair the current timeline gap.

3. Temporal Consistency:

Is the event time and description compatible with the chronological evolution implied by the question?

Penalize events with unclear, conflicting, or obviously mismatched temporal information.

Scoring Rule

Assign a score from 0 to 1:

- 1.0: Fully answers the question and clearly fills a missing timeline node, causal link, temporal transition, or event element.
- 0.9: Almost completely answers the question and provides strong logical utility, with only minor details missing.
- 0.8: Answers the main information need and provides useful missing information; accept as logically useful.
- 0.7: Related and partially useful, but incomplete, weakly connected to the gap, or missing key details.
- Below 0.7: Merely topically related, redundant, temporally inconsistent, contradictory, vague, or irrelevant.

Use the above anchors as calibration points.

Scores greater than or equal to 0.8 are accepted as logically useful candidate events.

Output

Output Format:

0.0

Directly output only one numeric score between 0 and 1.

Listing 4: Prompt template for logical utility verification.

Topic Query: Astronauts Stuck		
Questions & Generated Timeline	LUV Filtering	Reference Timeline
Stage-1: Parametric Causal Gap Detection Gaps Questions: 1. What operational decision/event explains the transition from “docking with anomalies” to a delayed return plan? 2. Was the return schedule updated due to thruster/helium issues, and to what date? Generated Timeline : 2014-09-16: Boeing wins a contract with NASA for \$4.2 billion to send astronauts to and from the International Space Station, as part of a move to privatize space flight in the U.S. 2018-06-01: Starliner finds a problem during a launch test, leading to a propellant leak during the test of its launch abort system. ... ✓2024-06-18: NASA postpones Starliners return again, giving more time to address thruster malfunctions and helium leaks, rescheduling for June 26. 2024-07-02: Boeing reports major losses amounting to \$125 million on the Starliner program, bringing total losses to \$1.6 billion since 2016. ✓2024-09-07: Starliner returns to earth unmanned and lands in New Mexico as the thruster issues posed safety concerns for Williams and Wilmore.	LUV in Stage-1 Rejected ✗2024-06-15: Engineers discovered that most of the Starliner’s issues appeared partially resolved after testing the thrusters, though the exact causes remained unclear. ✗2024-08-25: NASA announced that the Boeing Starliner could not safely return the astronauts, and SpaceX would take over their rescue mission, set for February of the following year. ...	2014-09-16: Boeing wins a \$4.2 billion contract from NASA to send astronauts to and from the International Space Station. 2018-05-01: NASA expresses concerns about Starliner’s parachute system, as reported by the U.S. Government Accountability Office, predicting certification delays to 2020. 2018-07-21: Boeing identifies an ‘anomaly’ during a test of Starliner’s launch abort system, leading to a propellant leak. 2019-12-20: Starliner fails its flight test to reach the ISS as it does not enter the correct orbit, failing to dock at the ISS. 2021-08-04: Boeing and NASA cancel Starliner’s mission due to valve issues in the propulsion system. 2022-05-25: Starliner safely returns from its second orbital flight test, but the docking at the ISS was described as ‘nail-biting’. 2022-09-21: American astronaut Frank Rubio travels to the ISS with two Russian cosmonauts for a six-month mission aboard the Russian spacecraft Soyuz MS-22. ... 2024-06-05: First crewed test flight launches; helium leaks noted. 2024-06-06: Docks after ~26 hours; leaks persist and multiple thrusters go offline. 2024-06-18: Return postponed again (to June 26) due to thrusters/helium leaks. 2024-09-07: Starliner returns to earth unmanned, landing at White Sands Space Harbor in New Mexico.
Stage-2: Historical Trajectory Analogy Gaps Questions: 1. What intermediate <i>test-stage</i> events occurred between the early certification/safety concerns and the first orbital flight-test failure? 2. Did Starliner experience a launch-abort system test anomaly (e.g., propellant leak) during this interval, and when was it reported? Generated Timeline: ... ✓2018-05-01: NASA has concerns about Starliner, according to a report by the U.S. Government Accountability Office highlighting a potential delay to 2020 for Boeing’s spacecraft certification due to concerns over its parachute system. ... ✓2019-12-20: Starliner fails its flight test as it does not enter the correct orbit, preventing it from docking with the International Space Station; no crew was onboard. ...	LUV in Stage-2 Rejected ✗2019-10-01: Christina Kochs six-month ISS mission is extended to 11 months after relinquishing her Soyuz return seat to a UAE astronaut, breaking the record for the longest spaceflight by a woman. ...	
Stage-3: Ontology-Guided Element Completion Element Completion Questions: 1. How long did it take Starliner to dock with the ISS? Generated Timeline: ... ✓2024-06-06: Starliner docks at the ISS after ~26 hours; helium leaks persist and multiple reaction-control thrusters go offline during the rendezvous and docking phase. ...	LUV in Stage-3 Rejected ✗2024-06-06: NASA and Boeing discover more problems as Starliner docks at the ISS with persisting helium leaks and thruster issues. ...	

Table 10: Case study table with row-wise correspondence between module outputs (left), LUV decisions (middle), and a single continuous reference timeline column (right).