MILES: A MILESTONE-CENTRIC FRAMEWORK FOR LLM-DRIVEN NEWS TIMELINE SUMMARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A news timeline must preserve the events that explain how a topic develops, even when those events are buried in articles about something else. This makes timeline summarization a problem of retaining relevant events and selecting the final milestones. We introduce **MILES** (**MI**lestone selection with **L**LM-guided **E**vidence **S**tructuring), a training-free framework for topic-focused timeline summarization (Topic-TLS) from a fixed collection of news articles. Topic-aware multi-event extraction retains both article-main events and embedded topic milestones; date-cluster shortlisting exposes a bounded candidate set; and an LLM selects the final milestones by their semantic importance. On Entities, Crisis, and T17, MILES achieves the highest semantic-alignment F1 among the evaluated systems, improving over LLM-TLS by 35.6%, 33.3%, and 64.1%, respectively. Controlled experiments locate the benefit: on identical candidate pools, every tested LLM selector outperforms the mechanical alternatives, while extraction and shortlisting determine which milestones remain available to select. Human preference studies and a diagnostic with four LLM judges support the semantic evaluation. These results support retaining topic events and using an LLM to select milestones from a bounded candidate pool.

1 INTRODUCTION

Timeline summarization (TLS) explains how a news topic changes: what triggered a crisis, which decisions altered its course, and when a new outcome emerged. A useful short timeline preserves these transitions. Frequently reported updates may repeat known information, while a brief passage may contain the decision that changes the story. In closed-corpus topic-focused timeline summarization (Topic-TLS), the system must recover such milestones from a fixed collection of dated news articles and select a timeline of a given length.

This requires retaining topic-relevant events even when they are not an article's main news, then choosing which candidates deserve the limited timeline slots. In the example for the "BP oil spill" topic (Section 3.3), a May article revisits the April rig explosion: preserving that earlier event makes the initiating milestone available for selection. Semantic identity also matters in evaluation: a different event cannot explain a reference milestone merely because their dates coincide. Our metric-blind human study supports this distinction (Appendix A).

Earlier TLS methods organize evidence through salient dates, event extraction, sentence ranking, clustering, and graph compression Allan et al. (2001); Chieu & Lee (2004); Kessler et al. (2012); Nguyen et al. (2014); Gholipour Ghalandari & Ifrim (2020); Li et al. (2021). LLM-TLS generates article-level event summaries and clusters them through pairwise LLM comparison Hu et al. (2024). This motivates using an LLM to compare candidate meanings and select a timeline of fixed length.

We propose **MILES**, a training-free pipeline that makes this decision explicit. Topic-aware extraction preserves article-main events and embedded topic milestones; date-cluster shortlisting bounds the candidate pool; and an LLM selects source-linked event identifiers using milestone criteria. Cluster size determines which candidates reach the LLM; semantic judgment determines the final timeline. This separation makes each stage testable: extraction is evaluated by the evidence it supplies, shortlisting by the evidence it preserves, and final selection by the milestones it recovers. Candidate identifiers link each selected milestone to its source article, allowing the resulting timeline to be inspected.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

On Entities (Gholipour Ghalandari & Ifrim, 2020), Crisis (Tran et al., 2015a), and T17 (Tran et al., 2013), MILES improves semantic-alignment F1 over LLM-TLS. On fixed candidate pools, every tested LLM selector outperforms all tested mechanical selectors, with MILES improving over the strongest mechanical rule by 54.2%. Upstream analyses establish how extraction exposes reference-relevant events and shortlisting preserves them.

Our contributions are:

- A formulation of closed-corpus news Topic-TLS around bounded semantic milestone selection, separating evidence retention from final timeline decisions.
- MILES, which implements this formulation through multi-event extraction, date-cluster shortlisting, and traceable LLM selection.
- Three-benchmark results and controlled component comparisons, supported by human preferences, judge stability, selector model comparisons, and LLM call counts.

2 Related Work

2.1 News Timeline Summarization

Early TLS systems studied temporal summaries, salient dates, event extraction and ranking, optimization, and graph compression Allan et al. (2001); Chieu & Lee (2004); Kessler et al. (2012); Li & Li (2013); Yan et al. (2011); Nguyen et al. (2014); La Quatra et al. (2021); Li et al. (2021); You et al. (2022). Later variants model graphical date selection, salient updates, multimodal ranking, and comparative timelines Tran et al. (2015b); Kedzie et al. (2015); Wang et al. (2016); Duan et al. (2020). DATEWISE synthesizes this line through date selection, salience, and redundancy control Gholipour Ghalandari & Ifrim (2020). LLM-TLS generates article-level event summaries and clusters them through pairwise LLM comparison, reporting on the same three closed-corpus benchmarks: T17, Crisis, and Entities Hu et al. (2024). Within this setting, MILES retains multiple topic events from each article, then uses an LLM to select the final timeline. This separation allows evidence retention and final selection to be evaluated independently.

2.2 LLMs for Evidence Construction and Selection

CHRONOS and R2A-TLS use search, retrieval, reflection, and query rewriting to construct timeline evidence Wu et al. (2025); Bao et al. (2025). TimelineReasoner studies reasoning-based construction in open- and closed-domain settings Zhang et al. (2026). DTELS addresses granularity Zhang et al. (2025), while CTLS studies constraints Qorib et al. (2025). SUnSET organizes stakeholders, events, and time Sim et al. (2025); NTS-CoT addresses hallucination Lyu et al. (2026).

News summarization and generative event extraction determine which evidence survives into later processing Zhang et al. (2024); Ravaut et al. (2024); Chen et al. (2024); Simon et al. (2024). Final candidate selection has traditionally used graph or salience ranking Mihalcea & Tarau (2004); Kessler et al. (2012); Nguyen et al. (2014); LLM rerankers and pairwise rankers show that models can instead compare candidates under task criteria Sun et al. (2023); Qin et al. (2024). MILES brings these roles together: extraction preserves topic milestones, and the LLM makes the final selection within a bounded candidate pool.

For timeline construction, an article summary is an intermediate representation: a topic-relevant event omitted there cannot be recovered by ranking its summary alone. Multi-event extraction addresses this evidence boundary. Candidate selection poses a different question, comparing retained events by their role in the topic's development. Our controlled experiments distinguish these stages by fixing the candidate pools when comparing selectors and excluding final selection when measuring candidate visibility.

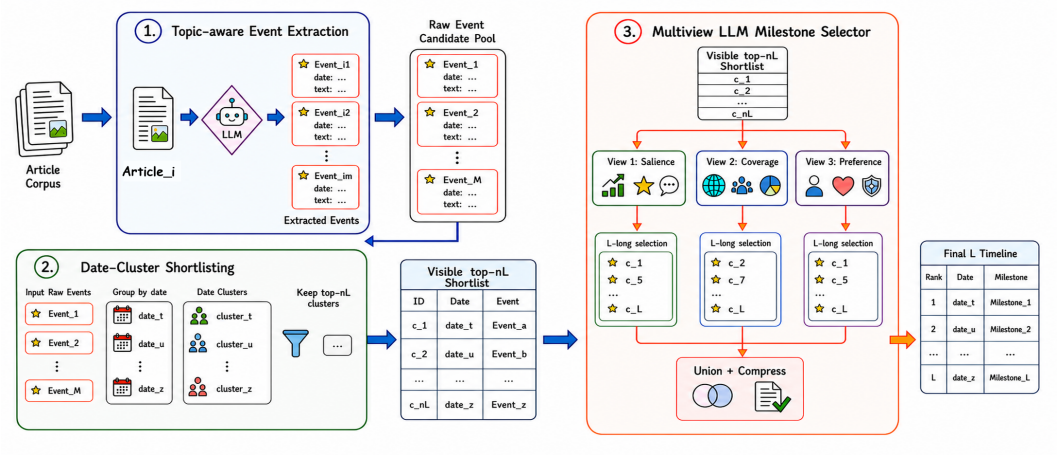

Figure 1: Overview of MILES. Topic-aware extraction and date-cluster shortlisting build a bounded shortlist; the multiview LLM selector makes the final semantic milestone decision.

3 Methodology

3.1 Design: Preserve, Bound, and Select

An article's main news and a topic's key milestones need not coincide: a later report may revisit an earlier event that explains the topic's development. Multi-event extraction keeps such evidence available for comparison across articles. The subsequent stages separate a manageable candidate pool from the final importance judgment. Date clustering consolidates repeated evidence within a bounded pool; semantic selection then compares the visible events by their roles in the topic, beyond how frequently they were reported.

MILES separates three responsibilities (Figure 1): **extraction preserves** article-main and embedded topic events; **shortlisting limits** the candidate count by reducing repeated date evidence; and **LLM selection chooses** milestones by importance, process change, coverage, and non-redundancy. Cluster size determines candidate visibility; the LLM judges final importance. Returning candidate identifiers keeps selected milestones traceable to source articles.

3.2 Pipeline Implementation

For topic q, let $\mathcal{A}_q = \{(d_i, x_i)\}_{i=1}^{N_q}$ be its dated article set and L_q its target timeline length. A multiplier n bounds the visible candidate count by nL_q. The output is the chronologically ordered sequence of selected milestone events

$$Y_q = (y_1, \ldots, y_{L_q}), \qquad y_j = (t_j, s_j),$$

where $t_j \le t_{j+1}$ and s_j is the selected event statement. The pipeline is

$$\mathcal{A}_q \to \mathcal{E}_q \to \mathcal{C}_q^{(nL)} \to Y_q.$$

Here $\mathcal{E}_q$ is the set of extracted events and $\mathcal{C}_q^{(nL)}$ is the selector-visible shortlist. Appendix B supplies the representative and parsing rules; full prompt templates are supplied in `MILES_prompts_v1.md`.

Topic-Aware Multi-Event Extraction

The extractor processes each article independently with its topic and article content. The content carries the dated evidence; the article's publication date is retained in the source metadata. The model returns one article-main event and up to six topic-meaningful non-main events, each with an event date and statement. If g is the extraction function,

$$\mathcal{E}_q = \bigcup_{i=1}^{N_q} g(q, x_i).$$

The prompt distinguishes successive milestones, such as an investigation, charge, verdict, and sentence, and preserves dated events that alter the topic's trajectory. Each candidate e stores its event date $\tau(e)$, statement $s(e)$, source article, and whether it was extracted as a main or non-main event (its extraction slot). This stage supplies evidence; it does not decide the final timeline.

DATE-CLUSTER CANDIDATE SHORTLISTING

Extracted events are grouped by normalized event date:

$$D_{q,t} = \{e \in \mathcal{E}_q : \tau(e) = t\}.$$

Each nonempty cluster contributes one representative $r_{q,t} = \rho(D_{q,t})$. The deterministic representative rule prioritizes repeated event wording, extraction slot, and confidence, with stable tie-breaking. Clusters are ranked by the number of extracted events they contain, $\kappa(D_{q,t}) = |D_{q,t}|$, yielding

$$\mathcal{C}_q^{(nL)} = \mathrm{Top}_{nL_q}\big(\{r_{q,t}\}; \kappa\big).$$

The main Entities and T17 configurations use $n = 12$. Crisis candidates are filtered to the task time interval, grouped by date, and capped at $8L$; all are retained when fewer are available. A topic focus is inferred from the candidate evidence (Appendix B.1). Cluster size determines which candidates reach the LLM, which selects the final L_q milestones.

MULTIVIEW LLM MILESTONE SELECTION

Selection favors events that change the topic's trajectory: triggers, formal decisions, changes in status, outcomes, and durable consequences. It distinguishes successive milestones within a process from repeated descriptions of the same event, and discourages routine commentary or filling the timeline with one sub-process. The three views emphasize the central story arc, institutional or status changes, and consequences or lasting impact. Each applies its emphasis to the same evidence, exposing complementary choices for the final comparison. The final compression call selects milestones from the union of the three views. It favors distinct milestones and coverage of different developments within the topic.

Each candidate has a stable row identifier and visible date and statement. Crisis also exposes cluster support alongside its induced topic focus. Each selection view $r \in \mathcal{R}$ receives the topic, target length, and the same shortlist, and is instructed to return L_q identifiers:

$$I_q^r = f_r(q, \mathcal{C}_q^{(nL)}, L_q).$$

Parsing removes invalid identifiers and duplicates. The valid selections form the union $U_q = \bigcup_{r \in \mathcal{R}} I_q^r$. The final compression call receives the candidate rows identified by U_q and selects L_q distinct milestones:

$$\widehat{I}_q = f_{\mathrm{comp}}(q, \mathcal{C}_q^{(nL)}|_{U_q}, L_q).$$

Returned identifiers are validated against the union, with excess valid items truncated. Length completion (repair) supplements short outputs from the original shortlist while preserving existing selections; empty outputs are reselected from that pool. Repair excludes references and evaluation labels. Selected statements are ordered chronologically, fixed, and saved (sealed) before evaluation; Appendix B gives the parsing rules.

3.3 A TRACEABLE EXAMPLE

A May 8, 2010 BP oil-spill article yields a non-main event about the April 20 rig explosion. The publication date locates the report, while the extracted event date places the initiating milestone earlier in the topic's history. Preserving this non-main event makes that distinction explicit before final selection. Its date-cluster representative `E00001` enters a 370-candidate pool; all three views and final compression retain it in the 48-item timeline. The example connects the three responsibilities: extraction preserves the embedded event, shortlisting exposes its representative, and selection retains it as a milestone. Candidate identifiers link these decisions back to the source article (Appendix C).

4 EXPERIMENTS

4.1 SETUP

Dataset	Method	C/W	SA Score	SA Date-F1	SA-R1	SA-R2	Date-F1	ARouge1
Entities	DATEWISE	87/974	0.0816	0.0694	0.0279	0.0122	0.2049	0.0430
Entities	LLM-TLS	226/835	0.2120	0.1546	0.0794	0.0372	0.2431	0.0678
Entities	CHRONOS-CR	201/824	0.1918	0.1374	0.0787	0.0408	0.2363	0.0738
Entities	MILES (Ours)	**308/763**	**0.2876**	**0.2148**	**0.1042**	**0.0506**	**0.2564**	**0.0750**
Crisis	DATEWISE	69/574	0.1073	0.0840	0.0392	0.0199	0.2945	0.0788
Crisis	LLM-TLS	87/556	0.1353	0.1042	0.0422	0.0173	0.3114	0.0848
Crisis	CHRONOS-CR	55/1957	0.0414	0.0264	0.0344	0.0191	0.1560	0.0448
Crisis	MILES (Ours)	**116/527**	**0.1804**	**0.1275**	**0.0556**	**0.0238**	**0.3324**	**0.0902**
T17	DATEWISE	119/557	0.1740	**0.1374**	0.0632	**0.0311**	**0.5429**	**0.1129**
T17	LLM-TLS	80/481	0.1277	0.0287	0.0358	0.0149	0.3523	0.0458
T17	CHRONOS-CR	117/1419	0.1050	0.0628	0.0583	0.0302	0.3383	0.0624
T17	MILES (Ours)	**145/547**	**0.2095**	0.1199	**0.0636**	0.0287	0.3686	0.0768

Table 1: Main comparison across three datasets under semantic alignment, with auxiliary Date-F1 and ARouge1 recomputed from the same sealed predictions. C/W denotes Correct/Wrong under the hard SA policy. DATEWISE is the date-wise baseline; CHRONOS-CR is the closed-retrieval CHRONOS reproduction evaluated with its actual output length.

DATA AND CONFIGURATION

We evaluate on Entities Gholipour Ghalandari & Ifrim (2020), Crisis Tran et al. (2015a), and T17 Tran et al. (2013). Each reference timeline defines an evaluation task: Entities has 47 tasks across 47 entity-centered topics, Crisis has 22 tasks, and T17 has 19 tasks across 9 topics. All preprocessing, candidate construction, baseline execution, and evaluation aggregation run on a server with eight NVIDIA A800 GPUs.

Unless otherwise specified, MILES uses Qwen3.6-27B Qwen Team (2026b) for topic-aware event extraction, selector calls, and LLM-assisted repair. The main Entities and T17 configurations use a top-$12L$ shortlist; the Crisis configuration filters candidates to the task time interval and retains at most $8L$ date-cluster representatives. For the reported Crisis candidate pools, at most $8L$ candidates are available: raising the cap to $12L$ exposes no additional candidates (Appendix D). Complete templates accompany the paper in `MILES_prompts_v1.md`.

BASELINES

We primarily compare with LLM-TLS Hu et al. (2024), the closest peer-reviewed LLM-based Topic-TLS baseline covering all three closed-corpus datasets. We reproduce its official implementation, replacing only the LLM backend (Appendix E), and also report DATEWISE Gholipour Ghalandari & Ifrim (2020). CHRONOS Wu et al. (2025) primarily addresses open-domain evidence acquisition through iterative self-questioning and retrieval. Its closed-domain adaptation searches the supplied corpus and retains the generation-and-merging procedure; its main methodological emphasis is retrieval, rather than optimizing corpus-wide extraction and bounded candidate selection. We include its closed-retrieval reproduction, CHRONOS-CR, as an auxiliary comparison supporting the choice of LLM-TLS as the primary closed-corpus baseline. CHRONOS-CR is evaluated at its actual output length. All systems are evaluated from prediction files fixed before evaluation.

EVALUATION PROTOCOL AND VALIDATION

Our primary metric is semantic-alignment (SA) F1, reported as SA Score in the tables. A fixed GPT-5-family OpenAI (2025) evaluator aligns predicted and reference events one-to-one and labels each pair as correct, partial, or false. Correct pairs must describe the same concrete milestone, beyond sharing a date or broad topic. For task q with m_q predictions and L_q references, let $\mathrm{Correct}_q$ count correct pairs. Under the hard policy, partial, false, and unmatched predictions contribute to

$\text{Wrong}_q = m_q - \text{Correct}_q$, and

$$\text{SA}_q = \frac{2\,\text{Correct}_q}{m_q + L_q}.$$

Dataset scores pool the counts across tasks. An unmatched prediction is a semantic mismatch to the reference, not necessarily an unsupported factual statement.

SA Date-F1 credits semantically correct pairs with matching dates, normalizing precision by prediction length and recall by reference length. SA-R1/R2 sum ROUGE-1/2 F1 over semantically correct pairs and normalize by reference length; semantic misses receive zero credit. Conventional Date-F1 compares date sets Martschat & Markert (2018), while ARouge1 measures ROUGE-1 F1 after date/content alignment Martschat & Markert (2017); Lin (2004). We retain these as auxiliary measures of date and wording overlap. Appendix F gives the complete formulas, matching protocol, and aggregation rules.

Validation. Seven annotators completed metric-blind event-pair and full-timeline studies without seeing system names, metric names, scores, or the expected outcome of the study. They preferred the semantically closer, date-shifted prediction in 53/70 conflict votes (9/10 case majorities), and the higher-SA timeline in 50/63 votes (8/9 topic majorities) when its competitor had higher Date-F1. These tests support the semantic milestone criterion at both event and timeline level. A separate audit of fixed diagnostic outputs with four LLM judges preserves the ordering DATEWISE $<$ LLM-TLS $<$ the diagnostic variant FS-LLM-TLS within each dataset. Appendix A provides the designs, topic-level votes, and judge scores.

4.2 Main Results

Table 1 evaluates full timelines, recomputing conventional metrics from the same fixed outputs. MILES achieves the highest SA Score among the evaluated systems on all three datasets. Against LLM-TLS, SA improves from 0.2120 to 0.2876 on Entities, 0.1353 to 0.1804 on Crisis, and 0.1277 to 0.2095 on T17: **relative gains of 35.6%, 33.3%, and 64.1%**. Two-sided paired Wilcoxon tests against LLM-TLS yield $p = 8.99 \times 10^{-4}$, 0.0340, and 0.0075 on Entities, Crisis, and T17, respectively (Appendix G).

MILES also leads Date-F1 and ARouge1 on Entities and Crisis. On T17, DATEWISE leads those conventional metrics while MILES leads SA, distinguishing date/wording overlap from semantic milestone identity. LLM-TLS also has higher SA than CHRONOS-CR on all three datasets, supporting its role as the primary closed-corpus comparator.

The next two tests isolate final selection under controlled and real candidate visibility; we then examine whether extraction supplies milestones and shortlisting preserves them.

4.3 Oracle-Visible Selector Ablation

Dataset	Correct	Wrong	SA Score	SA Date-F1	SA-R1	SA-R2
Entities	478	593	0.4463	0.4304	0.3683	0.3464
Crisis	308	335	0.4790	0.4728	0.4529	0.4454
T17	479	213	0.6922	0.6850	0.6742	0.6698
Overall	1265	1141	0.5258	0.5150	0.4789	0.4658

Table 2: Oracle-visible selector ablation. Each top-$12L$ pool contains the complete reference timeline as unlabeled candidate texts, and the multiview union-compress selector must choose exactly L_q final milestones. Under exact-length output, SA Score can be read as reference-milestone recall counting only correct matches.

We first ask whether the selector can recover milestones when they are already visible, separating selection capability from upstream recall. We randomly replace L_q candidates in each top-$12L$ pool with reference milestone texts, holding pool size fixed. The selector sees candidate rows without labels and must return L_q items (Appendix H).

In Table 2, overall SA is 0.5258, with 0.4463 on Entities, 0.4790 on Crisis, and 0.6922 on T17. With exact-length output, SA also equals reference-milestone recall when only correct matches receive credit. The result establishes that an LLM can recover reference-aligned milestones from a bounded candidate pool; the next test uses real shortlists without reference injection.

4.4 Selector Ablation on Fixed Shortlists

Selector	Type	Correct	Wrong	SA Score	SA Date-F1	SA-R1	SA-R2
Input-order top-L	Mechanical	330	2076	0.1372	0.1097	0.0447	0.0211
Cluster-size top-L	Mechanical	328	2078	0.1363	0.1101	0.0445	0.0210
Diverse-cluster top-L	Mechanical	326	2080	0.1355	0.1106	0.0440	0.0209
Representative-first top-L	Mechanical	267	2139	0.1110	0.0894	0.0362	0.0164
Chronological merge	LLM	483	1923	0.2007	0.1421	0.0694	0.0326
Reflection + information gain	LLM	503	1903	0.2091	0.1546	0.0714	0.0339
Single-pass plain	LLM	497	1909	0.2066	**0.1617**	0.0697	0.0331
Multiview union-compress (Ours)	LLM	509	1897	0.2116	0.1596	0.0734	0.0353
*Single-pass preference prompt**	*LLM*	***517***	***1889***	***0.2149***	*0.1588*	***0.0739***	***0.0355***

Table 3: Selector ablation on fixed top-$12L$ shortlists. All rows use the same candidate pools and output exactly L_q items; only the selector changes. Results are aggregated over Entities, Crisis, and T17. *Dataset-specific preferences refined through multiple rounds of tuning informed by multiview experiments; reported separately as a tuned diagnostic.

After verifying oracle-visible selection, we test whether LLM selection still helps on real pipeline shortlists drawn from the original news corpus. We conduct a selector ablation on shared top-$12L$ pools, holding extraction, clustering, and shortlisting constant for 88 tasks across Entities, Crisis, and T17. We replace only the selector that maps shortlisted candidates to the final L_q timeline; results are shown in Table 3, with definitions in Appendix I.

The comparison separates mechanical ordering rules from LLM-based semantic selection. Mechanical selectors use input order, cluster frequency, representative-first ranking, or diversity-constrained cluster ordering; LLM selectors include direct, chronological, reflection-based, and multiview prompts. Because all rows share the same upstream candidates, differences isolate the final selection step.

The strongest mechanical selector reaches 0.1372 SA Score, while every LLM selector reaches at least 0.2007. Multiview union-compress reaches 0.2116, a **54.2% relative improvement over the strongest mechanical selector**. It also improves on the general-purpose single-pass prompt (0.2066 SA), with higher SA-R1/R2 (0.0734/0.0353 versus 0.0697/0.0331).

The starred single-pass preference prompt reaches 0.2149 after repeated tuning of dataset-specific preferences using insights from the multiview experiments. This result reflects additional adaptation, so we report it separately as a tuned diagnostic and retain multiview as the general-purpose method. The controlled comparison supports **LLM-based semantic judgment as the main source of improvement over mechanical ranking**.

Selection quality also depends on which events reach the selector. We therefore examine evidence retention during extraction and shortlisting.

4.5 Candidate Visibility from Multi-Event Extraction

We next examine whether multi-event extraction supplies reference-aligned evidence for subsequent shortlisting. We compare LLM-TLS-style one-news-one-summary extraction with topic-aware multi-event extraction using uncapped date-cluster candidates and the cluster-size top-$12L$ shortlist; Table 4 reports candidate count, count relative to the cap, exact-date recall, and $\pm$3d recall. Appendix J provides audit cases (J.1) and the visibility protocol (J.2). These visibility metrics test whether a later semantic selector can see plausible evidence, not whether the final selected timeline is correct.

Dataset	Extraction	Candidates	Fill	Exact-Date Recall	±3d Recall
Entities	One-summary	27655/11825	2.152/0.920	0.5668/0.4370	0.7348/0.5537
Entities	Multi-event (Ours)	47418/12620	3.690/0.982	**0.7143/0.5612**	**0.8422/0.6713**
T17	One-summary	1246/1232	0.150/0.148	0.4075/0.4061	0.9682/0.9668
T17	Multi-event (Ours)	5646/4442	0.680/0.535	**0.8801/0.8526**	**0.9899/0.9827**
Crisis	One-summary	6543/5373	0.848/0.696	0.8911/**0.8383**	0.9285/0.9191
Crisis	Multi-event (Ours)	22085/7716	2.862/1.000	**0.9425**/0.8040	**0.9844/0.9331**

Table 4: Stage-1 candidate visibility ablation. Each metric cell reports Uncapped/Top-$12L$ cap. One-summary follows the LLM-TLS-style one-news-one-summary extraction path, while Multi-event (Ours) uses topic-aware event extraction. Both stages contain one representative per date cluster. Fill is candidate count divided by $12L$; it can exceed 1 before capping. Bold indicates the higher recall within each dataset and stage.

On Entities and T17, multi-event extraction improves reference-date visibility at both the uncapped-pool and top-$12L$ stages. The top-$12L$ exact-date recall rises from 0.4370 to 0.5612 on Entities and from 0.4061 to 0.8526 on T17; the T17 gain is especially large, **more than doubling exact-date visibility after the cap**. The uncapped-pool metrics show the same direction, which means the improvement begins at extraction rather than only at later ranking.

A complementary audit checks semantic recovery in 48 cases already confirmed to lose the reference milestone under one-summary extraction. Multi-event extraction fully recovers 35 reference milestones; including partial recoveries raises the count to 43 (Appendix J.1). This targeted audit shows how preserving non-main events can recover specific milestones beyond improving date coverage.

For example, in the Snowden audit case, one-summary extraction retains the broad event that he leaked classified information about US surveillance. The reference milestone instead concerns the US government's revocation of his passport. Topic-aware multi-event extraction recovers this specific legal-status change from the article. The contrast illustrates why preserving the article's salient event can still omit a distinct milestone needed to explain the topic's development.

On Crisis, multi-event extraction improves uncapped-pool visibility and top-$12L$ ±3d recall, while exact-date visibility after the cap is shaped by the downstream ordering rule. Richer extraction expands the evidence; shortlisting determines which candidates reach the selector.

4.6 Candidate Shortlisting Ablation

Method	Task runs	Candidates	Date Exact	±3d Date	Semantic recall
Random 12L (3 seeds)	264	24778	0.4849	0.7926	0.2790
Date-cluster top-$12L$ (Ours)	88	24778	**0.7099**	**0.8308**	**0.4027**
Date-density top-$12L$	88	28274	0.1575	0.2411	0.1421
Window-density top-$12L$	88	24815	0.6758	0.7648	0.3753
Text+cluster hybrid	88	23983	0.6550	0.8130	0.3745
Text+all-frequency hybrid	88	23983	0.6617	0.8013	0.3766
Diversity-aware MMR	88	23983	0.6234	0.8155	0.3570
Chronological coverage	88	23983	0.6172	0.8026	0.3591

Table 5: Candidate ranking ablation before final LLM selection. Date Exact and ±3d Date measure reference-date visibility. Semantic recall is the number of correctly matched reference milestones divided by all reference milestones, using the one-to-one matching protocol discussed in Section 4.6. Every method includes all 88 reference tasks, including tasks with zero eligible candidate–reference pairs. Random uses three seeds (264 task runs); its candidate count is the mean total per seed, and its semantic recall pools matches and reference counts across all three seeds.

Having examined what extraction makes available, we test whether Stage-2 shortlisting retains that evidence when reducing the uncapped date-cluster pool to a bounded shortlist. Holding the uncapped pool fixed and excluding final selection, Table 5 compares eight strategies for retaining up to $12L$ candidates on exact-date, ±3d date, and semantic visibility; definitions are in Appendix K. Semantic

recall asks whether the shortlist contains an event matching each reference milestone; it does not evaluate a final timeline. The denominator includes every reference milestone from all 88 tasks, including tasks with no matches; the random baseline repeats each task under three seeds.

Date-cluster top-$12L$ (Ours) obtains the highest exact-date visibility, 0.7099, the highest $\pm$3d visibility, 0.8308, and the **strongest semantic visibility, with 0.4027 semantic recall (969 of 2406 reference milestones)**. The strongest hybrid, text+all-frequency, reaches 0.3766 semantic recall.

4.7 Selector Models and LLM Calls

Dataset	Tasks	Views+comp.	Repair	Focus	Total
Entities	47	188	3	0	191
T17	19	76	1	0	77
Crisis	22	88	0	22	110
All	88	352	4	22	378

Table 6: MILES calls after article extraction. Views+comp. counts the three selection calls and final compression call. Focus denotes Crisis topic induction. Counts cover the recorded prediction-construction stages and exclude evaluation.

Replacing the selector. We hold the $12L$ candidate pools and multiview procedure fixed and replace only the selector model. Under the common internal semantic judge, overall SA is 0.2626 for Qwen3.5-9B, 0.2876 for Qwen3.6-27B, 0.3048 for DeepSeek-V4-Flash, and 0.3106 for GPT-5.5-High (Table 20). These results support selector replacement without changing upstream evidence construction. The internal diagnostic scores are reported separately from the unified main-table evaluation; the Qwen3.5-9B row produces 2400 items for 2406 references, while the other rows have exact total length. Appendix L also tests pool size, prompt rules, and visible input fields.

Calls after extraction. MILES uses one extraction call per article and deterministic shortlisting. Each timeline needs three selection-view calls and one compression call; repairs add three calls on Entities and one on T17. Crisis also uses one topic-focus call per task. These stages total 378 calls after extraction across 88 tasks (Table 6). LLM-TLS invokes its LLM during same-event clustering; Appendix M gives the comparison-pair estimate and per-topic counts. Call counts exclude token use and elapsed time.

Run counts and variation diagnostics are in Appendix N; software dependencies in Appendix O; and the supplementary artifact map in Appendix P.

5 Conclusion

Within a fixed news collection, MILES constructs news timelines through topic-aware multi-event extraction, date-cluster shortlisting, and multiview LLM selection. Across three benchmarks, it improves semantic-alignment F1 over LLM-TLS by 33.3–64.1% relative. Controlled comparisons show that LLM selection improves milestone recovery from shared candidate pools, while extraction and shortlisting determine which evidence remains available. Candidate identifiers link selected milestones to source articles, making the timeline traceable. Together, these findings support a practical design for news timeline summarization: preserve topic-relevant events, bound the candidate pool, and use semantic judgment to select the milestones that explain the topic's development.

6 Limitations

Date-cluster support bounds the selector's context efficiently but favors frequently represented dates; combining it with semantic salience could preserve more low-frequency milestones. Multi-event extraction increases generation volume and downstream processing, motivating adaptive extraction budgets for large corpora. Semantic evaluation uses an LLM judge: fixed predictions, fixed matching instructions, deterministic aggregation, and the reported human/judge checks support its interpretation. Further applications should retain these checks when changing the evaluator.

AI USE STATEMENT

Generative AI tools assisted with language polishing and section organization and formatting, primarily in Sections 2–4. They also assisted with checking technical explanations and interpreting reported results during manuscript revision. The authors are responsible for the final text, claims, and supporting evidence.

ETHICS STATEMENT

MILES studies timeline summarization using existing news benchmarks. The human evaluation described in Appendix A.1 assesses event correspondence and timeline preferences under metric-blind conditions. News selection and summarization can amplify source biases or misrepresent events through omissions and factual errors. Applications should preserve links to source evidence and support human verification of generated timelines, with particular care for sensitive news topics.

REPRODUCIBILITY STATEMENT

The appendices document pipeline algorithms and a worked example, experimental configurations, baseline reproduction, evaluation protocols, and additional analyses. They also report LLM call counts, run counts and seeds, and software dependencies. The separate code package provides the compact reproduction scripts and dependency files, with its contents and scope described in Appendices O and P. The separate prompt package, `MILES_prompts_v1.md`, supplies full prompt templates and request constructors, including the length-completion prompts used in the experiments. These materials complement the implementation details in Appendix B and the evaluation protocol in Appendix F.

REFERENCES

James Allan, Rahul Gupta, and Vikas Khandelwal. Temporal summaries of news topics. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 10–18. ACM, 2001. doi: 10.1145/383952.383954.

Chenlong Bao, Shijie Li, Minghao Hu, Ming Qiao, Bin Zhang, Jin-Tao Tang, Shasha Li, and Ting Wang. R2A-TLS: Reflective retrieval-augmented timeline summarization with causal-semantic integration. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 752–766, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.40. URL https://aclanthology.org/2025.findings-emnlp.40/.

Ruirui Chen, Chengwei Qin, Weifeng Jiang, and Dongkyu Choi. Is a large language model a good annotator for event extraction? *Proceedings of the AAAI Conference on Artificial Intelligence*, 38 (16):17772–17780, 2024. doi: 10.1609/aaai.v38i16.29730. URL https://ojs.aaai.org/index.php/AAAI/article/view/29730.

Hai Leong Chieu and Yoong Keok Lee. Query based event extraction along a timeline. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 425–432. ACM, 2004. doi: 10.1145/1008992.1009065.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Yijun Duan, Adam Jatowt, and Masatoshi Yoshikawa. Comparative timeline summarization via dynamic affinity-preserving random walk. In *ECAI 2020: 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pp. 1778–1785. IOS Press, 2020. doi: 10.3233/FAIA200292.

Demian Gholipour Ghalandari and Georgiana Ifrim. Examining the state-of-the-art in news timeline summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1322–1334, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.122. URL https://aclanthology.org/2020.acl-main.122/.

Qisheng Hu, Geonsik Moon, and Hwee Tou Ng. From moments to milestones: Incremental timeline summarization leveraging large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7232–7246, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.390. URL https://aclanthology.org/2024.acl-long.390/.

Chris Kedzie, Kathleen McKeown, and Fernando Diaz. Predicting salient updates for disaster summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1608–1617, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1155. URL https://aclanthology.org/P15-1155/.

Rémy Kessler, Xavier Tannier, Caroline Hagège, Véronique Moriceau, and André Bittar. Finding salient dates for building thematic timelines. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 730–739, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/P12-1077/.

Moreno La Quatra, Luca Cagliero, Elena Baralis, Alberto Messina, and Maurizio Montagnuolo. Summarize dates first: A paradigm shift in timeline summarization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 418–427. ACM, 2021. doi: 10.1145/3404835.3462954.

Jiwei Li and Sujian Li. Evolutionary hierarchical Dirichlet process for timeline summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 556–560, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-2099/.

Manling Li, Tengfei Ma, Mo Yu, Lingfei Wu, Tian Gao, Heng Ji, and Kathleen McKeown. Timeline summarization based on event graph compression via time-aware optimal transport. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6443–6456, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.519. URL https://aclanthology.org/2021.emnlp-main.519/.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Feng Lyu, Huiqin Yan, Sijing Duan, Hao Wu, Shuang Gu, Xue Qiao, Weixu Zhang, and Haolun Wu. NTS-CoT: Mitigating hallucinations in LLM-based news timeline summarization with chain-of-thought reasoning, 2026. URL https://arxiv.org/abs/2606.13171.

Sebastian Martschat and Katja Markert. Improving ROUGE for timeline summarization. In Mirella Lapata, Phil Blunsom, and Alexander Koller (eds.), *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 285–290, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2046/.

Sebastian Martschat and Katja Markert. A temporally sensitive submodularity framework for timeline summarization. In Anna Korhonen and Ivan Titov (eds.), *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pp. 230–240, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-1023. URL https://aclanthology.org/K18-1023/.

Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-3252/.

Kiem-Hieu Nguyen, Xavier Tannier, and Veronique Moriceau. Ranking multidocument event descriptions for building thematic timelines. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 1208–1217, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL https://aclanthology.org/C14-1114/.

OpenAI. GPT-5 system card. https://cdn.openai.com/gpt-5-system-card.pdf, 2025.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. Large language models are effective text rankers with pairwise ranking prompting. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 1504–1518, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.97. URL https://aclanthology.org/2024.findings-naacl.97/.

Muhammad Reza Qorib, Qisheng Hu, and Hwee Tou Ng. 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, 2025. doi: 10.1609/aaai.v39i23.34691. URL https://ojs.aaai.org/index.php/AAAI/article/view/34691.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026b. URL https://qwen.ai/blog?id=qwen3.6-27b.

Mathieu Ravaut, Aixin Sun, Nancy Chen, and Shafiq Joty. On context utilization in summarization with large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2764–2781, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.153. URL https://aclanthology.org/2024.acl-long.153/.

Tiviatis Sim, Kaiwen Yang, Xin Shen, and Kenji Kawaguchi. Who's important? – SUnSET: Synergistic understanding of stakeholder, events and time for timeline generation, 2025. URL https://arxiv.org/abs/2507.21903. arXiv:2507.21903v3.

Étienne Simon, Helene Olsen, Huiling You, Samia Touileb, Lilja Øvrelid, and Erik Velldal. Generative approaches to event extraction: Survey and outlook. In Joel Tetreault, Thien Huu Nguyen, Hemank Lamba, and Amanda Hughes (eds.), *Proceedings of the Workshop on the Future of Event Detection (FuturED)*, pp. 73–86, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.futured-1.7. URL https://aclanthology.org/2024.futured-1.7/.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large language models as re-ranking agents. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14918–14937, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.923. URL https://aclanthology.org/2023.emnlp-main.923/.

Giang Tran, Mohammad Alrifai, and Eelco Herder. Timeline summarization from relevant headlines. In Allan Hanbury, Gabriella Kazai, Andreas Rauber, and Norbert Fuhr (eds.), *Advances in Information Retrieval*, volume 9022 of *Lecture Notes in Computer Science*, pp. 245–256, Cham, 2015a. Springer International Publishing. ISBN 978-3-319-16354-3. doi: 10.1007/978-3-319-16354-3_26. URL https://doi.org/10.1007/978-3-319-16354-3_26.

Giang Tran, Eelco Herder, and Katja Markert. Joint graphical models for date selection in timeline summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1598–1607, Beijing, China, July 2015b. Association for Computational Linguistics. doi: 10.3115/v1/P15-1154. URL https://aclanthology.org/P15-1154/.

Giang Binh Tran, Mohammad Alrifai, and Dat Quoc Nguyen. Predicting relevant news events for timeline summaries. In *Companion Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pp. 91–92, Rio de Janeiro, Brazil, May 2013. ACM. doi: 10.1145/2487788.2487829. URL https://doi.org/10.1145/2487788.2487829.

William Yang Wang, Yashar Mehdad, Dragomir R. Radev, and Amanda Stent. A low-rank approximation approach to learning joint embeddings of news stories and images for timeline summarization. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 58–68, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1008. URL https://aclanthology.org/N16-1008/.

Weiqi Wu, Shen Huang, Yong Jiang, Pengjun Xie, Fei Huang, and Hai Zhao. Unfolding the headline: Iterative self-questioning for news retrieval and timeline summarization. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4385–4398, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.248. URL https://aclanthology.org/2025.findings-naacl.248/.

Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. Evolutionary timeline summarization: A balanced optimization framework via iterative substitution. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 745–754. ACM, 2011. doi: 10.1145/2009916.2010016.

Jingyi You, Dongyuan Li, Hidetaka Kamigaito, Kotaro Funakoshi, and Manabu Okumura. Joint learning-based heterogeneous graph attention network for timeline summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4091–4104, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.301. URL https://aclanthology.org/2022.naacl-main.301/.

Chenlong Zhang, Tong Zhou, Pengfei Cao, Zhuoran Jin, Yubo Chen, Kang Liu, and Jun Zhao. DTELS: Towards dynamic granularity of timeline summarization. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *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)*, pp. 2682–2703, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.136. URL https://aclanthology.org/2025.naacl-long.136/.

Liancheng Zhang, Xiaoxi Li, and Zhicheng Dou. Timelinereasoner: Advancing timeline summarization with large reasoning models, 2026. URL https://arxiv.org/abs/2605.12518.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57, 2024. doi: 10.1162/tacl_a_00632. URL https://aclanthology.org/2024.tacl-1.3/.

A Validation Evidence for Semantic-Alignment Evaluation

This appendix validates the semantic-alignment (SA) protocol defined in Appendix F. We test whether semantic matching agrees with human preferences, how it affects system rankings, and how results vary across judges and implementations.

A.1 Metric-Blind Human Validation

Table 7: Metric-blind human validation of semantic event identity. Seven annotators were not shown metric names, system names, scores, or the expected study outcome. C/W denote correct/wrong event-identity judgments.

Study	Condition	Unit	Semantic-side votes	Date/opposite votes	Majority outcome
Pair identity	Correct pairs with one-day shifts	8 items	52/56 C	4/56 W	8/8 C-majority
Pair identity	Wrong or insufficient pairs	16 items	104/112 W	8/112 C	16/16 W-majority
Shared-reference	Semantic match vs. closer date	10 cases	53/70 semantic	17/70 date-closer	9/10 semantic-majority
Full timeline	Higher-SA vs. higher-Date-F1	9 topics	50/63 higher-SA	13/63 higher-Date-F1	8/9 higher-SA majority

Table 8: Metric-blind full-timeline preference test. Each row compares two predicted timelines for the same reference: one has higher SA but lower Date-F1, and the other has higher Date-F1 but lower SA.

Topic	Higher-SA votes	Higher-Date-F1 votes
Syria	7	0
Libya	4	3
Egypt	5	2
Charles Taylor	7	0
Dmitry Medvedev	6	1
Dilma Rousseff	5	2
Haiti	7	0
H1N1	6	1
BP oil	3	4
Total	**50/63**	**13/63**

We first test whether humans prefer semantic event identity when it conflicts with timestamp proximity. Seven annotators participated in three metric-blind studies. They were not shown metric names, system names, Date-F1, A-ROUGE, SA, or the expected outcome of the study.

Table 7 summarizes the three studies. At the predicted–reference event-pair level, annotators reliably recognized correct event identity despite one-day timestamp shifts, while rejecting exact-date or near-date pairs whose event identity was wrong or insufficient. In the same-reference conflict test, annotators chose the semantically closer but date-shifted prediction in 53/70 votes and gave it a majority in 9/10 cases. At the full-timeline level, annotators preferred the higher-SA timeline in 50/63 votes and in 8/9 topic-level comparisons, even though the competing timeline had higher Date-F1.

Table 8 gives the topic-level votes, extending the event-pair preference test to complete predicted timelines.

A.2 Ten-Source Metric Meta-Evaluation

Table 9: Dataset-level Spearman correlation with SA Score over ten fixed TLS output sources. Values are computed within each dataset, so the table tests whether other metrics induce similar or different system orderings from the semantic matching criterion.

Dataset	Date-F1	A-R1	A-R2	SA-D	SA-R1	SA-R2	STA-R1	STA-R2
Crisis	-0.222	0.926	0.926	0.936	0.948	0.936	0.893	0.924
Entities	0.654	0.901	0.901	0.960	0.960	0.960	0.966	0.948
T17	-0.534	0.718	0.951	0.875	0.952	0.915	0.976	0.952

Table 10: Pairwise inversion rates against SA Score over ten fixed TLS output sources. For each dataset, all system pairs are compared; an inversion means that a metric ranks a comparable pair in the opposite order from SA Score, excluding ties on either side. Rates are percentages.

Dataset	Date-F1	A-R1	A-R2	SA-D	SA-R1	SA-R2	STA-R1	STA-R2
Crisis	56.1	7.3	7.3	7.0	4.7	7.0	11.6	9.3
Entities	24.4	9.8	9.8	4.8	4.7	4.7	4.7	7.0
T17	64.3	21.4	4.8	13.6	6.7	8.9	4.4	6.7

Table 11: Conditional reordering among LLM-based output sources with close event coverage. Pairs are retained only when $|\Delta \mathrm{SA}| \leq 0.01$, so these rows test whether conditional SA metrics still expose date and wording differences when event identity is nearly tied.

Metric	Close pairs	Reordered	Rate (%)
SA-D	21	5	23.8
SA-R1	21	6	28.6
SA-R2	21	7	33.3
STA-R1	21	5	23.8
STA-R2	21	7	33.3

We next test how semantic matching affects system rankings across ten fixed TLS output sources: C1, M1–M4, L1–L2, O1–O2, and R1. These correspond to the selector variants defined in Appendix I. The analysis compares metrics on the same fixed predictions, using hard semantic labels and a three-day tolerance for STA-R1/R2. A-R1/R2 denote alignment-based ROUGE-1/2, and SA-D denotes SA Date-F1.

Table 9 reports dataset-level Spearman correlation with SA Score. Date-F1 is negatively correlated with SA on Crisis and T17, while A-ROUGE is closer to SA but still not identical to the semantic-first family. The relationship between date overlap and semantic matching therefore differs across datasets.

Table 10 measures the practical ordering effect. For each dataset, all system pairs are compared, and an inversion means that a metric ranks two systems in the opposite order from SA Score. Date-F1 reverses many comparable pairs relative to semantic event identity, especially on Crisis and T17. A-ROUGE reduces this mismatch, but still produces non-trivial inversions. The conditional SA metrics have lower inversion rates because they share the same semantic gate.

SA Score measures event identity. SA-D, SA-R1/R2, and STA-R1/R2 describe date and wording properties conditional on semantic matching; their formulas are given in Appendix F. Table 11 shows that these metrics distinguish systems with nearly tied SA Score.

A.3 Judge-Side Stability

Table 12: Cross-judge stability of SA Score on fixed TLS outputs. Four heterogeneous LLM judges use the same semantic-alignment instruction. Absolute scores vary slightly, but the source ordering is preserved within every dataset. Variance is row-wise population variance multiplied by 10^6.

Dataset	Output source	GPT-5.6 Sol	GPT-5.6 Luna	Claude Opus 4.8	DeepSeek-V4	Var. ($\times 10^{-6}$)
Entities	DATEWISE	0.194	0.195	0.192	0.190	3.69
Entities	LLM-TLS	0.281	0.280	0.280	0.275	5.50
Entities	FS-LLM-TLS	0.330	0.330	0.330	0.330	0.00
Crisis	DATEWISE	0.350	0.350	0.348	0.346	2.75
Crisis	LLM-TLS	0.380	0.380	0.383	0.378	3.19
Crisis	FS-LLM-TLS	0.395	0.395	0.395	0.390	4.69
T17	DATEWISE	0.247	0.245	0.247	0.245	1.00
T17	LLM-TLS	0.482	0.482	0.482	0.479	1.69
T17	FS-LLM-TLS	0.523	0.520	0.520	0.521	1.50

Because scalable semantic matching uses a judge model, we directly test whether evaluator choice preserves system-level comparisons over fixed outputs.

Table 12 evaluates fixed diagnostic outputs with GPT-5.6 Sol, GPT-5.6 Luna, Claude Opus 4.8, and DeepSeek-V4 under the same semantic-alignment instruction. The output sources are DATEWISE, LLM-TLS, and FS-LLM-TLS, a diagnostic LLM-TLS variant with structured event extraction, LLM cluster summaries, and LLM-assisted final selection. Within every dataset, all four judges preserve the ordering DATEWISE $<$ LLM-TLS $<$ FS-LLM-TLS. Row-wise variance is on the order of 10^{-6}. These fixed-output comparisons test evaluator stability separately from the MILES main results.

A.4 Evaluation Implementation and Sensitivity

Table 13: Retrieval and temporal sensitivity on fixed LLM-TLS outputs, reusing existing semantic labels. Recall values are percentages; temporal rows report soft STA-R1 with $K = 5$.

Check	Setting	Quantity	Value
Top-K retrieval	$K = 5$	Candidate pairs	11959
Top-K retrieval	$K = 5$	Correct recall	96.69
Top-K retrieval	$K = 5$	Soft recall	95.59
Top-K retrieval	$K = 10$	Candidate pairs	23724
Top-K retrieval	$K = 10$	Correct recall	99.49
Top-K retrieval	$K = 10$	Soft recall	98.67
STA window	$\epsilon = 0$	Soft STA-R1	0.0405
STA window	$\epsilon = 1$	Soft STA-R1	0.0516
STA window	$\epsilon = 3$	Soft STA-R1	0.0590
STA window	$\epsilon = 7$	Soft STA-R1	0.0616

This diagnostic uses fixed LLM-TLS predictions on Entities, Crisis, and T17: 88 tasks, 2265 predicted events, and 2396 reference milestones. A BGE-large-en-v1.5 encoder retrieves the top-K predictions for each reference by cosine similarity. The diagnostic reuses existing semantic pair labels, treating pairs absent from the positive-label inventory as false. One-to-one alignment is then resolved deterministically by semantic label priority, retrieval similarity, date distance, and input order. This evaluates retrieval and matching on fixed outputs separately from the main evaluator.

Table 13 reports two checks on these fixed outputs. With $K = 5$, retrieval covers 96.69% of known correct semantic matches; with $K = 10$, coverage reaches 99.49%. Soft recall also credits known partial pairs with weight 0.5. The temporal sensitivity rows use $K = 5$ and report soft STA-R1, defined in Appendix F. Increasing the tolerance admits more aligned pairs to the wording score; it does not change their semantic labels.

B Pipeline Implementation and Algorithms

This appendix expands Section 3 with the implementation used for the fixed main-result predictions. It specifies dataset-specific inputs, the shared representative rule, the complete algorithms, and output recording. Full instructions and request constructors remain in `MILES_prompts_v1.md`: P01–P08 document MILES construction, and D01–D04 give compact extraction and selector interfaces.

B.1 Extraction and Dataset-Specific Inputs

LLM-TLS extracts one topic-related event summary per article. MILES instead preserves one article-main event and up to six topic-meaningful non-main events, grounded in article text. These fields distinguish evidence retention from article summarization.

Entities and T17 expose candidate row identifiers, dates, and event statements. Their views emphasize the main topic trajectory, institutional or status changes, and lasting consequences. For Crisis, a country-level corpus is focused on the task's crisis process by inducing a second topic from its candidate evidence and temporal scope. The selector additionally receives cluster support and the induced scope. Topic induction does not read reference milestone text or external sources.

B.2 Date-Cluster Representative Rule

The shortlist stage consolidates repeated evidence and limits the number of candidates supplied to the selector. Let $z(e)$ be the normalized event sentence. The representative for date cluster $D_{q,t}$ is

$$\arg\max_{e \in D_{q,t}} \left(\mathrm{freq}(z(e)), \mathrm{slot}(e), \mathrm{conf}(e), |s(e)|, z(e)\right),$$

where $\mathrm{freq}(z(e))$ is normalized sentence frequency inside the date cluster, $\mathrm{slot}(e)$ prefers main-event extractions, $\mathrm{conf}(e)$ orders high before medium before low, and $|s(e)|$ is event-statement length. Clusters are ordered by $|D_{q,t}|$ in descending order and by date for deterministic tie-breaking.

B.3 Main-Result Algorithms

Algorithm B.1: End-to-End Main Pipeline

Input: topic or task q, articles $\mathcal{A}_q$, target length L_q, shortlist multiplier n, and dataset type.

1. Initialize the extracted-event set $\mathcal{E}_q \leftarrow \emptyset$.
2. For each article, call the topic-aware multi-event extractor; parse `news_main_event` and `topic_meaningful_non_main_events`; append every valid dated event to $\mathcal{E}_q$.
3. For Entities and T17, build an nL_q date-cluster shortlist using Algorithm B.2. In the main experiment, $n = 12$.
4. For Crisis, build a task-local windowed shortlist using Algorithm B.3 and attach the induced second-topic scope.
5. Run the dataset-specific multiview selector in Algorithm B.4.
6. Parse selected row identifiers, remove invalid or duplicate identifiers, repair exact length only from the same visible candidate pool, and seal the prediction before evaluation.

Output: sealed prediction for q.

Algorithm B.2: Date-Cluster Candidate Shortlisting

Input: extracted events $\mathcal{E}_q$, target length L_q, and multiplier n.

1. Remove events without a valid date or event sentence.
2. Group events by normalized event date, producing date clusters $D_t = \{e : \mathrm{date}(e) = t\}$.
3. For each D_t, choose one representative by sorting events by normalized text frequency, main-event slot preference, extraction confidence, event length, and deterministic text order.
4. Create one candidate row for each retained date cluster, storing row identifier, date, representative text, and cluster size.
5. Sort date clusters by cluster size descending and date ascending.
6. Keep the first $\min(|\{t : D_t \neq \emptyset\}|, nL_q)$ rows as the visible pool.

Output: visible nL_q date-cluster shortlist.

Algorithm B.3: Crisis Task-Scoped Shortlisting

Input: Crisis reference task with target length L_q.

1. Build the task-local time window around the reference temporal scope.
2. Retain country-level extracted events whose dates fall inside the window.
3. Group filtered events by date and choose date-cluster representatives as in Algorithm B.2.
4. If at least $8L_q$ date clusters are available, keep the top $8L_q$ by cluster size and date tie-break; otherwise keep all date clusters.
5. Attach the induced second topic and scope rules to the visible pool.

Output: crisis-scoped visible pool.

Algorithm B.4: Multiview Union-Compress Selection

Input: visible candidate pool $\mathcal{C}_q$, target length L_q, and the dataset-specific selector prompt.

1. Run each of the three view prompts on the same visible pool and require exactly L_q selected row identifiers.
2. Parse and validate identifiers from each view.
3. Form the union of valid unique identifiers selected by any view.

4. Run a final compression prompt over the union only, selecting distinct milestones with coverage of different developments within the topic; request JSON keys `selected_row_ids` and `brief_rationale`.
5. Validate identifiers against the union and deduplicate; truncate valid excess identifiers in returned order. If the result is short, supplement it from the original visible pool; if empty, reselect from that pool. Check the final length and seal the selected identifiers.

Output: exactly L_q final row identifiers.

B.4 Saving Predictions and Completing the Target Length

The prediction record stores the visible pool identifier, topic, target length, selected row identifiers, parse status, exact-length status, and raw selector metadata. Parsed identifiers are validated and deduplicated; valid identifiers beyond L_q are truncated in returned order. If fewer than L_q valid identifiers remain, an additional LLM call fills missing slots only from the same visible shortlist, keeping existing valid selections; an empty selection is reselected from that shortlist. Repair uses only this shortlist and the existing valid selections; reference timelines, evaluation results, and external documents are excluded.

Selected candidates are mapped back to their dates and statements and represented in chronological order. The full repair templates are in P07–P08 of `MILES_prompts_v1.md`.

C A Traceable Pipeline Example

The BP oil-spill example in Section 3.3 follows one candidate through the Qwen pipeline. The source article is dated May 8, 2010 (article index 126). The extractor emits three events; its non-main slot retains the April 20 rig explosion. The normalized candidate statement is: "Deepwater Horizon rig explodes, killing 11 workers and initiating the BP oil spill."

Stage	Traceable record
Extraction	Topic `bpoil`; article 126; event `bpoil::126::1`; non-main slot; event date April 20, 2010.
Date cluster	The candidate is the representative of the April 20 cluster, whose recorded support is 674 extracted events.
Visible shortlist	Row `E00001`; 370 visible candidates; target timeline length $L = 48$.
View selection	All three views include `E00001`, so it belongs to the union supplied to compression.
Compression	The final call retains `E00001` and returns 48 valid identifiers.
Sealing	The 48-item output includes the source-linked statement. This task does not require repair.

Table 14: A single event traced from article evidence to the selected timeline. Identifiers connect extraction, cluster representative, view selection, and sealed output.

The event illustrates the role of non-main extraction: an earlier initiating milestone remains explicitly available even when the source article is later in the topic's development. The view and compression records then show how that candidate becomes part of the final timeline.

D Final Configuration Settings

This appendix records the final settings used in Section 4.1. Controlled changes to pool size, prompt rules, selector model, and input fields are documented separately in Appendix L.

The submitted main configuration uses Qwen3.6-27B for extraction, selection, and repair unless otherwise specified. Entities and T17 use date-cluster top-$12L$ shortlists; Crisis uses task-time filtering followed by an $8L$ candidate cap. The final selector is multiview union-compress, and the Entities/T17 visible row format is date plus event text; the task-specific Crisis prompt additionally includes cluster support and induced scope. Selector outputs are row identifiers, not free-form summaries; exact-length repair is allowed only against the same visible shortlist.

Configuration selection considered (i) exact-length output, (ii) SA Score, (iii) reference-like candidate visibility before final selection, (iv) cross-dataset robustness, and (v) context/cost control. The $12L$ default combines exact-length output with strong internal SA, while date plus text is the default row format in the controlled selector diagnostics. The Crisis-specific main pipeline uses the additional fields described above.

Crisis candidate-cap saturation. For the reported Crisis configuration, the available task-scoped pool reaches at most $8L$. With the eligible pool and ordering held fixed, retaining $\min(N, 8L)$ or $\min(N, 12L)$ candidates therefore gives the same shortlist because $N \leq 8L$.

E LLM-TLS Reproduction

We reproduce LLM-TLS using its official Topic-TLS implementation, changing only the LLM backend. The original method extracts one topic-related dated event summary per article, retrieves nearby candidate events by embedding similarity, uses an LLM to determine whether event pairs refer to the same event, and incrementally builds event clusters. Clusters are ranked by support, and the implementation provides TextRank-based representative selection to construct the dated timeline.

The official source is https://github.com/nusnlp/LLM-TLS; the Topic-TLS extraction templates, event-clustering code, and timeline-construction code specify these steps.

F Semantic-Alignment Metrics: Protocol and Rationale

This appendix defines the metrics, matching protocol, and aggregation rules used in Section 4.1.

F.1 Why Semantic-Alignment Metrics

Topic-TLS evaluation must verify that a predicted item refers to the same milestone as a human reference item. Date overlap alone can reward an event that happens on the correct day but describes the wrong development. Lexical overlap can reward a topically similar sentence that misses the milestone boundary. Conventional Date-F1 Martschat & Markert (2018) and ROUGE Lin (2004) measure date and wording overlap. Their semantic-alignment counterparts credit these properties only for semantically matched milestones.

SA metrics implement this ordering. The evaluator first decides whether a predicted event and a reference milestone describe the same concrete milestone. Only semantically correct one-to-one matches can contribute to date and text-overlap scores. The metric family therefore answers three questions in order: whether the system selected the right milestone, whether it assigned the right date to that milestone, and how closely the selected event text matches the reference wording.

F.2 Notation and Alignment

For task q, let the predicted timeline be

$$P_q = \{p_i = (d_i^P, s_i^P)\}_{i=1}^{m_q},$$

and let the reference timeline be

$$G_q = \{g_j = (d_j^G, s_j^G)\}_{j=1}^{L_q}.$$

Here d denotes the event date, s denotes the event text, m_q is the predicted length, and L_q is the reference timeline length.

The evaluator produces a one-to-one alignment set

$$A_q \subseteq \{1, \ldots, m_q\} \times \{1, \ldots, L_q\}.$$

Each aligned pair $(i, j) \in A_q$ receives a label $\ell_{ij} \in \{\texttt{correct}, \texttt{partial}, \texttt{false}\}$. Under the hard policy used in the main result and selector-ablation tables, only `correct` receives credit:

$$h_{ij} = \mathbf{1}[\ell_{ij} = \texttt{correct}].$$

Unmatched predictions and unmatched references receive zero credit. A `partial` label is retained in the audit file, but is counted as wrong in the reported hard metrics.

F.3 Conventional Auxiliary Metrics

The main text reports conventional Date-F1 and ARouge1 as auxiliary diagnostics. For Date-F1, let $T_q^P = \{d_i^P : p_i \in P_q\}$ and $T_q^G = \{d_j^G : g_j \in G_q\}$ be the predicted and reference date sets. Date precision, recall, and F1 are

$$P_q^{date} = \frac{|T_q^P \cap T_q^G|}{|T_q^P|}, \qquad R_q^{date} = \frac{|T_q^P \cap T_q^G|}{|T_q^G|},$$

$$\text{Date-F1}_q = \frac{2P_q^{date} R_q^{date}}{P_q^{date} + R_q^{date}},$$

with zero assigned when a denominator is zero.

For ARouge1, the standard TLS evaluator first constructs an alignment B_q between predicted and reference timeline items under date/content costs Martschat & Markert (2017). Let $O_1(B_q)$ be the total unigram overlap over the aligned text pairs, and let $C_1^P(B_q)$ and $C_1^G(B_q)$ be the corresponding predicted-side and reference-side unigram counts. Alignment-based ROUGE-1 precision, recall, and F1 are

$$P_q^{AR1} = \frac{O_1(B_q)}{C_1^P(B_q)}, \qquad R_q^{AR1} = \frac{O_1(B_q)}{C_1^G(B_q)},$$

$$\text{ARouge1}_q = \frac{2P_q^{AR1} R_q^{AR1}}{P_q^{AR1} + R_q^{AR1}}.$$

These two metrics do not require semantic event identity, which is why they are treated as auxiliary rather than primary evidence.

F.4 Semantic-Alignment Metric Definitions

Correct and Wrong. The two count metrics directly summarize hard semantic decisions:

$$\text{Correct}_q = \sum_{(i,j)\in A_q} h_{ij}, \qquad \text{Wrong}_q = m_q - \text{Correct}_q.$$

Thus Wrong includes unmatched predictions, `false` alignments, and `partial` alignments.

SA Score. SA Score measures milestone selection quality independent of exact dates or surface wording. We first compute semantic precision and recall:

$$P_q^{SA} = \frac{\text{Correct}_q}{m_q}, \qquad R_q^{SA} = \frac{\text{Correct}_q}{L_q}.$$

The task-level SA Score is their F1 score:

$$\text{SA Score}_q = \frac{2P_q^{SA} R_q^{SA}}{P_q^{SA} + R_q^{SA}},$$

with zero assigned when the denominator is zero.

SA Date-F1. SA Date-F1 measures temporal grounding only after milestone identity has been established. Let

$$D_q = \sum_{(i,j)\in A_q} h_{ij}\, \mathbf{1}[d_i^P = d_j^G].$$

Date precision and recall under semantic gating are

$$P_q^D = \frac{D_q}{m_q}, \qquad R_q^D = \frac{D_q}{L_q}.$$

SA Date-F1 is their harmonic mean:

$$F_q^{SA-D} = \frac{2P_q^D R_q^D}{P_q^D + R_q^D}.$$

This is a Date-F1-style score Martschat & Markert (2018), but a predicted date can receive credit only when the corresponding event is also semantically correct.

SA-R1 and SA-R2. SA-R1 and SA-R2 measure similarity to the reference wording for semantically correct milestone pairs. For $n \in \{1, 2\}$, let $\mathrm{ROUGE}_n(s_i^P, s_j^G)$ be the ROUGE-n F1 overlap between the predicted and reference texts Lin (2004). We compute

$$\text{SA-R}n_q = \frac{1}{L_q} \sum_{(i,j)\in A_q} h_{ij}\, \mathrm{ROUGE}_n(s_i^P, s_j^G).$$

This follows the alignment-based use of ROUGE in TLS evaluation Martschat & Markert (2017), but gives zero text-overlap credit to semantic misses.

Temporal and soft diagnostic metrics. The validation diagnostics in Appendix A additionally use STA-R1/R2, which apply a date-distance condition after semantic matching. For label weight w_{ij} and a tolerance ϵ in days,

$$\text{STA-R}n_q(\epsilon; w) = \frac{1}{L_q} \sum_{(i,j)\in A_q} w_{ij}\, \mathbf{1}[|d_i^P - d_j^G| \le \epsilon]\, \mathrm{ROUGE}_n(s_i^P, s_j^G).$$

The hard policy uses $w_{ij} = h_{ij}$. The soft policy assigns weights 1, 0.5, and 0 to `correct`, `partial`, and `false` labels, respectively; unmatched items receive zero. The ten-source metric comparison uses hard scores with $\epsilon = 3$. The retrieval sensitivity diagnostic reports soft STA-R1 at the stated values of ϵ. Both variants normalize by all reference milestones, giving zero when no aligned pair qualifies.

Retrieval recall measures coverage of a fixed inventory of known semantic matches before one-to-one alignment. Correct recall is the fraction of known `correct` pairs retained by top-K retrieval. Soft recall is the retained label weight divided by the total weight of all known `correct` and `partial` pairs, using the weights above. These retrieval percentages have a different denominator from final-timeline semantic recall.

Table-Level Aggregation. For a dataset or table split, the same formulas are applied after summing the corresponding numerators and denominators over all tasks. One-to-one matching is still enforced within each task before aggregation.

F.5 Evaluation Flow

SA Evaluation Protocol

Input: sealed prediction timelines, reference timelines, dataset metadata, and task identifiers.

1. For each task, load the predicted timeline P_q and reference timeline G_q from sealed files.
2. Send only P_q, G_q, and task metadata to the fixed SA evaluator. The evaluator cannot browse or use external documents.
3. Ask the evaluator to produce one-to-one predicted-reference alignments with labels `correct`, `partial`, or `false`.
4. Apply the hard policy: `correct` receives credit, while `partial` and `false` are both counted as wrong.
5. Compute Correct, Wrong, SA Score, SA Date-F1, SA-R1, and SA-R2 from the alignment file with the formulas above.

This protocol separates semantic matching from metric aggregation. The LLM evaluator is used only to decide milestone identity under a fixed prompt; all reported numeric metrics are then computed deterministically from the evaluator output and the sealed prediction/reference files.

F.6 Evaluator Interface and Labels

The main experiments use a fixed Codex semantic-alignment evaluator implemented with a GPT-5-family model OpenAI (2025). The evaluator receives the selected timeline events and the reference milestones for one task, and outputs one-to-one alignments. It evaluates sealed prediction/reference files only, and does not browse or use external knowledge.

The corresponding instruction is provided in Section D05 of the companion `MILES_prompts_v1.md`.

Dataset-Specific Reminders. For Entities, the evaluator treats biography, career, public role, legal status, family/identity, health, canonical work, death, and legacy as possible milestone dimensions. Stage distinctions such as nomination versus election, accusation versus conviction, and diagnosis versus death are preserved. For T17, the evaluator matches concrete topic-process events or state-change nodes rather than broad topical relevance. For Crisis, it matches concrete political, military, humanitarian, institutional, casualty/impact, or process-changing international nodes within the crisis task.

F.7 Reproducibility and Validation Boundary

Given sealed inputs and an emitted alignment file, every reported score is a deterministic function of explicit pair labels. A reader can inspect which predicted item was matched to which reference item, which label was assigned, and how that label contributes to each metric. Reproducing aggregation from these labels is distinct from validating the semantic judgments that produced them.

Appendix A reports the empirical evidence: metric-blind human studies, metric meta-evaluation, cross-judge stability, and implementation sensitivity. These checks address the semantic criterion and its implementation; they are separate from the deterministic aggregation protocol specified here.

G Paired Significance Checks

Baseline	Split	Units	Base SA	MILES SA	Δ	W/T/L	p
LLM-TLS	Entities	47	0.2147	0.2859	+0.0712	34/3/10	8.99e-4
LLM-TLS	Crisis	22	0.1350	0.1657	+0.0306	11/6/5	0.0340
LLM-TLS	T17	19	0.1306	0.2162	+0.0856	13/2/4	0.0075
LLM-TLS	All	88	0.1766	0.2408	+0.0642	58/11/19	3.14e-6
DATEWISE	Entities	47	0.0850	0.2859	+0.2010	44/2/1	6.99e-9
DATEWISE	Crisis	22	0.1102	0.1657	+0.0555	14/2/6	0.0276
DATEWISE	T17	19	0.1807	0.2162	+0.0354	12/1/6	0.3270
DATEWISE	All	88	0.1120	0.2408	+0.1289	70/5/13	6.62e-10
CHRONOS-CR	Entities	47	0.2053	0.2859	+0.0806	31/7/9	1.08e-4
CHRONOS-CR	Crisis	22	0.0458	0.1657	+0.1199	20/0/2	3.34e-5
CHRONOS-CR	T17	19	0.1010	0.2162	+0.1152	14/1/4	0.0016
CHRONOS-CR	All	88	0.1429	0.2408	+0.0979	65/8/15	5.76e-10

Table 15: Paired task-level significance checks on strict SA Score. Base SA and MILES SA are paired-unit means; Δ is MILES minus the baseline; p is from a two-sided Wilcoxon signed-rank test.

This appendix provides the paired significance checks referenced in Section 4.2. Each paired unit is one entity timeline in Entities or one reference timeline in Crisis and T17. We compare systems by the strict SA Score of each unit and apply a two-sided Wilcoxon signed-rank test to the paired differences. The win/tie/loss column counts how often MILES obtains a higher, equal, or lower per-unit SA Score than the baseline. For Crisis, one metric export uses different reference identifiers across systems, so pairs are matched by topic and reference order; the aligned gold lengths are identical within each topic.

The table reports arithmetic means over paired units, while Table 1 reports corpus-level micro scores from pooled Correct/Wrong counts. The two summaries therefore differ slightly in value, but they are computed from the same sealed predictions and semantic-alignment judgments.

H Oracle-Visible Selector Ablation Details

This appendix expands Section 4.3. The experiment isolates the final selector by ensuring that the bounded candidate pool contains one reference milestone text for each target slot while preserving the original pool size.

H.1 Oracle-Visible Pool Construction

Algorithm H.1: Random-L Reference Injection

Input: task q, original visible pool $\mathcal{C}_q$, target length L_q, reference timeline $\mathcal{G}_q$, and fixed random seed.

1. Copy the original visible candidate list $\mathcal{C}_q$.
2. Sample L_q replacement positions uniformly from the candidate positions in $\mathcal{C}_q$.
3. Convert each reference milestone $g_j \in \mathcal{G}_q$ into a candidate row with its reference date and reference text.
4. Replace the sampled original candidates with the converted reference rows.
5. Preserve the total candidate count and the original task metadata.
6. Write a manifest containing the source pool identifier, oracle pool identifier, replacement positions, injected rows, replaced original rows, and reference rows.

Output: oracle-visible candidate pool and injection manifest.

The injected rows are candidate rows, not labels. They create an oracle-visible candidate pool in which the selector could recover every reference milestone if it chose the human-preferred rows.

H.2 Selector Boundary

The ablation uses the same multiview union-compress selector protocol as the main pipeline. No new selector prompt is introduced. The selector receives the same visible-row interface as the main-result pipeline: row identifiers, dates, event texts, and, when present in the route, cluster-size evidence. It does not receive reference labels, alignment judgments, SA scores, ROUGE scores, or the injection manifest.

A selected row can receive credit either by selecting an injected reference-text row or by selecting an original candidate that the unified SA evaluator judges semantically aligned with the same reference milestone.

H.3 Evaluation

Predictions are sealed and evaluated with the unified SA evaluator described in Appendix F. Direct injected-row hit rate is retained as an internal diagnostic, but the paper reports the same SA metrics as the main experiments. This makes the ablation comparable with Table 1: it answers how much reference-aligned recall is achievable when the final LLM selector has a compact candidate pool that already contains reference-like milestones.

I Fixed-Shortlist Selector Ablation Details

This appendix expands Section 4.4. All variants receive the same top-$12L$ candidate pools and the same target length L_q; the only changed component is the final selector that maps visible candidates to L_q output rows.

I.1 Mechanical Selectors

Algorithm I.1: Rule-Based Selection Variants

Input: task q, fixed candidate pool $\mathcal{C}_q$, and target length L_q.

1. **M1 input-order top-L.** Sort by row identifier order and select the first L_q candidates.
2. **M2 cluster-size top-L.** Sort by cluster size descending, date ascending, and row identifier ascending; select the first L_q candidates.
3. **M3 representative-first cluster top-L.** Prefer main-slot representatives first, then sort by cluster size descending, date ascending, and row identifier ascending; select the first L_q candidates.
4. **M4 diverse-cluster top-L.** Rank candidates by cluster size descending, date ascending, and row identifier ascending. First select at most one candidate from each date/cluster key, then fill remaining slots by the same ranking until L_q rows are selected.

Output: L_q mechanically selected candidate identifiers.

These variants represent mechanical ordering rules: none reads the candidate text as a semantic preference judgment beyond stored metadata.

I.2 LLM Selector Variants

All LLM variants use visible row identifiers only, forbid browsing and external sources, and require JSON output with selected row identifiers. They differ in how the final selection decision is decomposed.

L1 single-pass plain. The minimal prompt asks the model to select exactly L_q important, non-duplicate events from the candidate list.

The corresponding instruction is provided in Section D06 of the companion `MILES_prompts_v1.md`.

C1 chronological merge. The chronological merge variant splits candidates into temporal chunks, selects high-recall local candidates from each chunk, and then merges local selections into a global top-L_q timeline using the same preference rules.

R1 reflection-gap-information-gain. The reflection variant instructs the model to internally inspect coverage, causal gaps, information gain, and duplicate groups before outputting the final JSON selection. The reflection itself is not emitted.

O1 multiview union-compress. The multiview variant uses three complementary selection views: central public/career/crisis trajectory, formal institutional or authority events, and durable impact or legacy events. It unions valid identifiers from the views and compresses the union to exactly L_q.

O2 staged selection with length completion. This staged selector first selects up to L_q high-confidence final-quality milestones, then supplements missing slots from remaining candidates, trims additions if necessary, and repairs to exact length from the same pool.

L2 single-pass preference prompt*. This variant encodes dataset-specific selection preferences developed through multiple rounds of refinement informed by the multiview experiments. It frames selection as human timeline editing, but its score reflects additional preference tuning. We therefore report it separately as a tuned diagnostic rather than use it as the general-purpose method. The compact interface is given in Section D07 of `MILES_prompts_v1.md`.

I.3 Exact-Length Handling

Every LLM variant is parsed against the candidate row-id set. Invalid identifiers and duplicates are removed. If a parsed output is short, the repair call can only use the same fixed shortlist; if it is long, it is truncated or compressed according to the variant's rule. No variant has access to reference timelines, semantic labels, or evaluation results.

J Multi-Event Extraction: Evidence and Visibility Protocol

This appendix supports Section 4.5 with complementary evidence: audited summary-loss cases test whether specific milestones survive extraction, while candidate-visibility diagnostics measure the evidence available before final selection.

J.1 Summary-Loss Audits and Cases

Setting	Events	Events/case	Full	Partial/No
One-summary	–	–	6	4 / 3
Multi-event	30	2.31	10	1 / 2
Forced main+non-main	26	2.00	8	2 / 3
Optional non-main	34	2.62	10	1 / 2
Strong optional non-main	33	2.54	11	1 / 1

Table 16: Stage-1 audit on 13 representative blind summary-loss cases. Full, partial, and no denote recovery of the reference milestone; Events/case is the mean extraction count.

Metric	Value
Cases	48
Total extracted events	154
Mean events per case	3.21
Article-main events	48
Topic-meaningful non-main events	106
Full recoveries	35 / 48
Partial recoveries	8 / 48
No recovery	5 / 48
Full + partial recoveries	43 / 48 = 89.6%

Table 17: Expanded Stage-1 audit on 48 previously identified cases where one-summary extraction omitted a reference milestone, using topic-aware multi-event extraction.

This audit tests whether extraction retains reference milestones that are present in the source article. In several audited cases, one-summary extraction keeps the headline event or a nearby salient event and omits the reference milestone.

We first tested this failure mode on 13 representative blind summary-loss cases. The worker saw the article and extraction prompt, but not the reference timeline or earlier system outputs. Table 16 shows that the original one-summary setting fully recovers 6 reference milestones, while the strongest topic-aware multi-event setting fully recovers 11 milestones; including partial recoveries raises this count to 12. The latter emits 33 events in total: 13 article-main events and 20 topic-meaningful non-main events. Among the non-main events, 13 are high confidence, 7 are medium confidence, and none are low confidence.

We then applied the same topic-aware setting to 48 previously identified cases in which one-summary extraction omitted a reference milestone. Table 17 shows that it fully recovers 35 reference milestones, with 43 recovered when partial matches are included. These cases show how extracting non-main events can recover milestones omitted by one-summary extraction.

The following three cases illustrate the same pattern as the aggregate audit: retaining topic milestones requires preserving events beyond the main news of each article.

John Kerry. The reference milestone is that John Kerry announced he would not run for president in 2008 and would seek a fifth Senate term instead. The one-summary extraction follows the surrounding election-news salience, keeping an Iowa caucus event about John Edwards and Rudolph Giuliani. Topic-aware extraction recovers Kerry's own political decision from the Senate floor.

Edward Snowden. The reference milestone is that the US government revoked Snowden's passport. The one-summary extraction keeps the broader and more familiar event that Snowden leaked classified information about US surveillance. Topic-aware extraction recovers the narrower legal-status milestone: US authorities revoked Snowden's passport before he boarded the flight from Hong Kong.

Michelle Obama. The reference milestone is that Michelle Obama presented the Academy Award for best picture at the Oscars. The one-summary extraction keeps a nearby public-appearance event, Michelle Obama meeting Big Bird for a Let's Move campaign video. Topic-aware extraction recovers the Oscars best-picture announcement.

J.2 Candidate-Visibility Protocol

J.2.1 Compared Extraction Paths

One-summary path. The baseline path uses the LLM-TLS-style one-news-one-summary extraction schema described in Appendix B.1. Each article contributes one article-level summary event. After clustering, candidate rows are represented by date-cluster representatives.

Multi-event path. The proposed path uses topic-aware multi-event extraction. Each article may contribute one main event and additional topic-meaningful non-main events. These extracted events are then normalized into the same date-cluster representation used by the downstream shortlist stage.

J.2.2 Visibility Computation

Algorithm J.1: Stage-1 Candidate Visibility

Input: reference task q with reference milestones $\mathcal{G}_q$ and target length L_q.

1. Load the one-summary candidate artifacts and the multi-event candidate artifacts.
2. For each extraction path, group events by date and choose one representative per cluster to form the uncapped date-cluster candidate pool.
3. Build a capped pool for each extraction path by sorting candidates by date-cluster size descending, date ascending, and deterministic text tie-break; keep at most $12L_q$ candidates.
4. For each method-stage pair, compute the candidate count and fill ratio, where fill is candidate count divided by $12L_q$.
5. For each reference milestone, check whether at least one candidate has the exact same date. Exact-date recall is the fraction of reference milestones covered by this condition.
6. For each reference milestone, check whether at least one candidate has a date within three days. $\pm$3d recall is the fraction of reference milestones covered by this relaxed condition.

Output: uncapped-pool and capped-pool visibility statistics.

These metrics are candidate-visibility diagnostics. They test whether downstream stages have plausible evidence to inspect; they do not assign final timeline credit by themselves.

J.2.3 Candidate–Reference Semantic Pair Judge

The run also exports exact-date and $\pm$3d candidate-reference pairs for semantic candidate-visibility judging. The corresponding instruction is provided in Section D08 of the companion `MILES_prompts_v1.md`.

This judge is separate from final timeline evaluation. It measures how many reference-like candidates exist in the pool before final selection.

K Candidate Shortlisting Ablation Details

This appendix expands Section 4.6. All methods start from the same uncapped date-cluster candidate pool and construct a shortlist of at most $12L_q$ candidates before final selection.

K.1 Candidate Ranking Methods

Algorithm K.1: Shortlist Construction Variants

Input: task q, uncapped candidate pool $\mathcal{R}_q$, and target size $K = \min(|\mathcal{R}_q|, 12L_q)$.

1. **Random 12L.** Sample K candidates uniformly at random under three fixed seeds and aggregate the resulting visibility metrics.
2. **Date-cluster top-$12L$ (Ours).** Sort by date-cluster size descending and date ascending; keep the first K rows.
3. **Date-density top-$12L$.** Rank by same-day density, computed from raw-event support on the candidate date, then apply deterministic tie-breakers.
4. **Window-density top-$12L$.** Rank by local temporal density in a $\pm$3 day window, then by window count, own cluster size, date, and deterministic text tie-breakers.
5. **Text+cluster hybrid.** Compute a hybrid score from normalized text score and normalized log cluster size, sweep the weight, and report the best selected run.
6. **Text+all-frequency hybrid.** Combine normalized text score with normalized cluster size, same-day density, and window density; sweep the weight and report the best selected run.
7. **Diversity-aware MMR.** Start from a hybrid base score and iteratively select candidates with an MMR-style penalty for high similarity to already selected items, duplicate pressure, and same-date over-concentration.
8. **Chronological coverage.** Divide the task time span into chronological buckets, allocate a quota to each bucket, fill bucket quotas by text/frequency score, and fill remaining slots globally.

Output: one $12L_q$ shortlist per ranking variant.

Only the candidate ranking rule changes across these methods. The upstream uncapped pool, reference tasks, target lengths, and visibility evaluator are held fixed.

K.2 Semantic Visibility Pair Judge

The unified semantic visibility pass uses candidate-reference pairs prefiltered to candidate dates within three days of the reference date.

The corresponding instruction is provided in Section D09 of the companion `MILES_prompts_v1.md`.

K.3 Semantic Visibility Aggregation

Algorithm K.2: Semantic Recall from Pair Labels

Input: method m, task q, candidate-reference pairs, and pair-judge labels.

1. Initialize one evaluation unit for every method, reference task, and seed, with its full reference count and zero matches.
2. Build all candidate-reference pairs whose dates differ by at most three days.
3. Apply the pair judge to obtain `correct`, `partial`, or `false` labels.
4. Sort non-false pairs by label strength and confidence.
5. Greedily match each candidate and each reference milestone at most once.
6. Compute semantic recall as matched correct references divided by $|\mathcal{G}_q|$.
7. Sum matched correct references and full reference counts over all initialized units, including units with no eligible pairs or no correct matches; report their ratio. Random contributes three units per reference task, one per seed.

Output: semantic recall.

This aggregation evaluates whether a shortlist contains reference-like candidates. It does not run the final selector and does not measure final timeline quality.

The deterministic methods each contribute 88 tasks (47 Entities, 22 Crisis, and 19 T17); random contributes 264 task–seed units. The total reference count is 2406 per complete pass, or 7218 across the three random seeds.

L Additional Selector-Side Ablations

This appendix expands the selector-side diagnostics introduced in Section 4.7. The experiments below are internal diagnostics evaluated with a DeepSeek-V4-family semantic judge DeepSeek-AI (2026). They analyze selector behavior under controlled changes and complement the unified Codex SA results used for the main claims.

L.1 Diagnostic Scope and Internal Judge

All four diagnostics use sealed candidate pools, candidate-only prompts, JSON row-id outputs, and no access to reference timelines during selection. The internal judge matches selected events to reference milestones when both refer to the same concrete milestone, event, result, official action, status update, impact update, or resolution node.

The corresponding instruction is provided in Section D10 of the companion `MILES_prompts_v1.md`.

This internal judge is used only for the selector-side diagnostics in this appendix. Main results and main ablations rely on the unified Codex SA evaluator described in Appendix F.

L.2 Pool-Size Ablation

Pool Size	Selected / GT	Overall SA	Entities SA	T17 SA	Crisis SA
$8L$	2406 / 2406	0.2785	0.3389	0.1951	**0.2675**
$10L$	2292 / 2406	0.2695	0.3281	0.2172	0.2283
$12L$ (Ours)	2406 / 2406	0.2876	**0.3417**	0.2225	**0.2675**
$15L$	2268 / 2406	0.2867	0.3240	**0.2570**	0.2553
$20L$	2303 / 2406	**0.2897**	0.3319	0.2542	0.2551

Table 18: Pool-size ablation with the same selector procedure. The $12L$ setting combines exact output length with strong internal SA; $20L$ has the highest overall SA and returns fewer items than requested.

The pool-size experiment fixes upstream extraction, date-cluster representative rules, and the multiview union-compress selector, and varies only the selector-visible shortlist size. It compares $8L$, $10L$, $12L$, $15L$, and $20L$ candidate pools. The results are reported in Table 18.

Both $8L$ and $12L$ return 2406 items for 2406 references. The $12L$ setting has higher overall internal SA than $8L$, is strongest on Entities, and ties the best Crisis score. The $20L$ setting has the highest overall SA and returns 2303 items. The $12L$ default thus combines complete output length with strong scores across the datasets.

L.3 Prompt-Rule Ablation

Variant	Prompt Change	Selected / GT	Overall SA	Entities SA	T17 SA	Crisis SA
P0	Full prompt (Ours)	2406 / 2406	0.2751	0.3287	0.2269	0.2379
P1	Minimal importance rule	2262 / 2406	0.2721	0.3301	0.2097	0.2426
P2	No editor identity	2205 / 2406	**0.2962**	**0.3493**	0.2550	**0.2510**
P3	No prefer/avoid rules	2246 / 2406	0.2885	0.3303	**0.2711**	0.2371

Table 19: Prompt-rule ablation with fixed $12L$ pools and the same selector procedure. P2/P3 improve internal SA over P0 and return fewer than the requested 2406 items.

The prompt-rule experiment fixes the $12L$ pool, selector model, multiview structure, and visible row format. It changes only the rule module in the selector prompt, comparing the full prompt with shorter variants that remove editor identity, detailed preference rules, or both. The results are reported in Table 19.

Removing editor identity (P2) yields the highest internal SA, and removing detailed preference rules (P3) yields the second highest. These variants return 2205 and 2246 items, respectively, for 2406 references, whereas the full prompt returns all 2406 items.

L.4 Model-Swap Ablation

Selector Model	Selected / GT	Overall SA	Entities SA	T17 SA	Crisis SA
Qwen3.6-27B selector baseline (Ours)	2406 / 2406	0.2876	0.3417	0.2225	0.2675
Qwen3.5-9B selector	2400 / 2406	0.2626	0.2932	0.2113	0.2668
deepseek-v4-flash selector	2406 / 2406	0.3048	0.3667	0.2405	0.2708
GPT-5.5-High selector	2406 / 2406	**0.3106**	**0.3704**	**0.2480**	**0.2785**

Table 20: Model-swap ablation with fixed $12L$ pools and the same selector procedure. Model families follow Qwen3.6-27B Qwen Team (2026b) and Qwen3.5-9B Qwen Team (2026a), DeepSeek-V4 DeepSeek-AI (2026), and GPT-5 OpenAI (2025). GPT-5.5-High obtains the strongest scores, while Qwen3.6-27B remains a strong default selector.

The model-swap experiment fixes the $12L$ visible pools and multiview union-compress selector procedure, then replaces the selector model. The purpose is to check whether selection quality changes with model capability under the same candidate pool and prompt structure. The results are reported in Table 20.

Under the same selector inputs and procedure, Qwen3.6-27B improves over Qwen3.5-9B overall, deepseek-v4-flash DeepSeek-AI (2026) is stronger than both Qwen variants, and GPT-5.5-High obtains the strongest scores as a GPT-5-family selector OpenAI (2025). This supports the interpretation that stronger selectors can further improve milestone selection when the candidate pool and prompt constraints are held fixed.

L.5 Selector Input-Feature Ablation

Variant	Visible Fields	Selected / GT	Overall SA	Entities SA	T17 SA	Crisis SA
F0	date + text (Ours)	2406 / 2406	**0.2718**	**0.3287**	**0.2139**	0.2395
F1	text only	2351 / 2406	0.2708	0.3185	0.2056	0.2614
F2	cluster size + text	2273 / 2406	0.2659	0.3214	0.1796	**0.2659**
F3	date + cluster size + text	2349 / 2406	0.2583	0.3241	0.1750	0.2360

Table 21: Selector input-feature ablation with fixed $12L$ pools, model, prompt rules, and selector procedure. In this diagnostic, date plus text achieves the highest overall SA and returns all 2406 requested items.

The input-feature experiment fixes the $12L$ pool, model, prompt rules, and selector procedure. It changes only the candidate row format: `row_id | date | text`, `row_id | text`, `row_id | cluster_size | text`, or `row_id | date | cluster_size | text`. The results are reported in Table 21.

In this diagnostic, date plus event text achieves the highest overall SA and returns all 2406 requested items. Text-only input has a similar overall SA, indicating that event text supplies much of the selection signal. The Crisis main configuration additionally includes cluster support and induced scope (Appendix D).

M Per-Topic LLM Call Counts

Dataset	Topics	Tasks	Ext.	LLM-TLS $\pm$1d Cmp. Est.	MILES Cmp.	MILES Extra	Max Extra
Entities	47	47	51,183	158,272	0	191	235
T17	9	19	4,203	43,315	0	77	95
Crisis	4	22	17,573	436,206	0	110	132

Table 22: Dataset-level LLM call accounting. Ext. is article-level extraction and is shared by both systems. LLM-TLS $\pm$1d Cmp. Est. estimates same-event comparison pairs among article-level candidate sentences within a one-day window. MILES Cmp. is zero because shortlisting uses deterministic date grouping. MILES Extra includes selection/compression, LLM repair, and Crisis topic induction. Max Extra is five calls per task for Entities/T17 and six for Crisis, including its topic-focus call.

Dataset	Topic	Ext.	LLM-TLS ±1d Cmp. Est.	Tasks	MILES Extra
Entities	Al Gore	1,035	2,310	1	4/5
Entities	Angela Merkel	1,728	2,569	1	4/5
Entities	Ariel Sharon	1,310	2,469	1	4/5
Entities	Arnold Schwarzenegger	587	402	1	4/5
Entities	Bashar al-Assad	1,206	1,864	1	4/5
Entities	Bill Clinton	2,041	2,822	1	4/5
Entities	Charles Taylor	203	250	1	4/5
Entities	Chris Brown	166	41	1	4/5
Entities	David Beckham	2,945	5,576	1	4/5
Entities	David Bowie	889	1,838	1	4/5
Entities	Dilma Rousseff	198	144	1	4/5
Entities	Dmitry Medvedev	330	342	1	4/5
Entities	Dominique Strauss-Kahn	246	983	1	4/5
Entities	Edward Snowden	976	3,105	1	4/5
Entities	Ehud Olmert	434	599	1	5/5
Entities	Enron	820	2,175	1	4/5
Entities	Hamid Karzai	695	600	1	4/5
Entities	Hassan Rouhani	245	267	1	4/5
Entities	Hu Jintao	251	234	1	4/5
Entities	Jacob Zuma	353	383	1	4/5
Entities	John Boehner	176	104	1	5/5
Entities	John Kerry	1,230	3,233	1	4/5
Entities	Julian Assange	662	2,186	1	4/5
Entities	Lance Armstrong	709	1,347	1	4/5
Entities	Mahmoud Ahmadinejad	845	1,091	1	4/5
Entities	Marco Rubio	288	489	1	4/5
Entities	Margaret Thatcher	2,649	12,071	1	4/5
Entities	Michael Jackson	1,411	5,369	1	4/5
Entities	Michelle Obama	492	620	1	5/5
Entities	Mitt Romney	1,348	6,880	1	4/5
Entities	Morgan Tsvangirai	392	542	1	4/5
Entities	Nawaz Sharif	241	248	1	4/5
Entities	Nelson Mandela	1,287	5,790	1	4/5
Entities	Osama bin Laden	2,075	15,364	1	4/5
Entities	Oscar Pistorius	382	1,208	1	4/5
Entities	Phil Spector	158	80	1	4/5
Entities	Prince William	648	918	1	4/5
Entities	Robert Mugabe	1,907	3,447	1	4/5
Entities	Rupert Murdoch	4,485	14,030	1	4/5
Entities	Saddam Hussein	3,579	21,714	1	4/5
Entities	Sarah Palin	992	8,323	1	4/5
Entities	Silvio Berlusconi	1,601	2,029	1	4/5
Entities	Steve Jobs	946	1,348	1	4/5
Entities	Taliban	2,732	5,264	1	4/5
Entities	Ted Cruz	426	1,079	1	4/5
Entities	Tiger Woods	1,613	2,196	1	4/5
Entities	WikiLeaks	1,251	12,329	1	4/5

Table 23: Per-topic LLM call accounting for Entities. MILES Extra is reported as sealed calls / maximum calls under the five-call-per-task bound.

Dataset	Topic	Ext.	LLM-TLS ±1d Cmp. Est.	Tasks	MILES Extra
T17	bpoil	1,307	19,098	5	21/25
T17	egypt	216	3,817	1	4/5
T17	finan	256	473	1	4/5
T17	h1n1	370	5,368	3	12/15
T17	haiti	278	7,048	1	4/5
T17	iraq	315	486	1	4/5
T17	libya	716	5,350	2	8/10
T17	mj	121	216	1	4/5
T17	syria	624	1,459	4	16/20
Crisis	egypt	4,083	172,602	4	20/24
Crisis	libya	4,274	101,512	7	35/42
Crisis	syria	5,170	118,283	5	25/30
Crisis	yemen	4,046	43,809	6	30/36

Table 24: Per-topic LLM call accounting for T17 and Crisis. The Tasks column counts reference timelines; Extra is observed/maximum calls after extraction. Crisis includes one topic-focus call per reference task, giving a six-call bound; T17 uses a five-call bound.

This appendix provides the call accounting referenced in Section 4.7. We count LLM calls used to construct predicted timelines and exclude local embedding search, parsing, file I/O, and metric computation.

For a topic q with N_q articles, both LLM-TLS and MILES make N_q article-level generation calls. LLM-TLS produces one article-level summary/event representation, while MILES uses one topic-aware extraction call that may return multiple candidate events. Thus the extraction-call count is identical at the article level.

The key difference is the clustering/selection boundary. LLM-TLS/Topic-TLS performs LLM-based same-event comparison during incremental event clustering, but the comparison is not global: it is constrained to candidate sentences within a short date window. The released artifacts do not retain the exact comparison trace, so we report a reproducible scale estimate based on article-level candidate sentences within ± 1 day:

$$B_q = \sum_{a<b} \mathbf{1}[|d_a - d_b| \leq 1],$$

where d_a and d_b are article dates. This is an estimate rather than an exact log-derived count because the actual number also depends on parsing and online clustering state. MILES instead uses deterministic date grouping for candidate shortlisting, so its same-event comparison call count is 0.

MILES adds three view calls and one union-compression call per reference-level timeline, with at most one additional LLM repair call in the reported repair procedure. Entities uses 188 selection/compression calls plus 3 repairs; T17 uses 76 plus 1. Crisis uses 88 selection/compression calls and 22 task-focus induction calls, with no repair in the sealed Crisis selections. Extraction is shared across reference tasks for the same topic. The per-task budget is therefore up to five calls after extraction for Entities/T17 and up to six when Crisis topic induction is included. The recorded total is 378 calls after extraction.

These are logical prediction-construction calls recorded for the listed stages; transport retries and evaluator calls are outside this accounting. The LLM-TLS comparison count remains the pair-based estimate defined above. Tables 22–24 provide the dataset and topic breakdowns; Table 6 exposes the MILES stage totals in the main text.

N Run Counts, Seeds, and Variation Diagnostics

Dataset	Extraction	Uncapped Candidates	Uncapped Exact-Date Recall	Range
Entities	One-summary	–	0.5668/0.5656/0.5660/0.5654	0.0014
T17	One-summary	–	0.4075/0.4072/0.4062/0.4060	0.0015
Crisis	One-summary	–	0.8911/0.8904/0.8909/0.8906	0.0007
Entities	Multi-event	47418/47401/47425/47410	0.7143/0.7139/0.7140/0.7137	0.0006
T17	Multi-event	5646/5659/5643/5652	0.8801/0.8795/0.8796/0.8790	0.0011
Crisis	Multi-event	22085/22069/22078/22073	0.9425/0.9317/0.9419/0.9421	0.0108

Table 25: Stage-1 candidate-pool replay diagnostics. Slash-separated values list the reported run followed by three additional replays. The range column reports the max-min spread of uncapped exact-date recall.

Dataset	Final SA Score	Range
Entities	0.2876/0.2872/0.2872/0.2875	0.0004
Crisis	0.1804/0.1799/0.1804/0.1801	0.0005
T17	0.2095/0.2088/0.2092/0.2094	0.0007

Table 26: Final selector replay stability under the MILES pipeline configuration. Slash-separated values list the reported run followed by three additional runs with the same model.

The main results use one fixed prediction file per system and dataset. The extraction and final-selection diagnostics below each include three additional runs. MILES is training-free. Given extracted events, date grouping, top-nL shortlisting, row-id mapping, and metric aggregation are deterministic; variation arises from LLM generation.

Appendix M gives the per-topic LLM call accounting for extraction, same-event comparison, and added selector calls.

We report three kinds of stability evidence. First, Table 25 checks the Stage-1 candidate-pool construction used in Table 4. Each cell lists the reported run followed by three additional extraction

replays. One-summary extraction is included only for uncapped exact-date recall to track date coverage across runs. For multi-event extraction, we also report uncapped candidate counts because pool size is part of the candidate-pool claim.

For one-summary extraction, uncapped exact-date recall varies by at most 0.0015 across the three datasets. For multi-event extraction, uncapped candidate counts change only slightly, and uncapped exact-date recall is nearly unchanged on Entities and T17; Crisis has the largest uncapped exact-date range (0.0108) but preserves the same qualitative conclusion. The $\pm$3d recall diagnostics are even less sensitive, so they are not tabulated separately; for a representative uncapped $\pm$3d replay, the four values are 0.7348, 0.7348, 0.7348, and 0.7347. At the top-$12L$ cap, both exact-date and $\pm$3d recall change negligibly, because the few uncapped-pool differences rarely alter the retained date clusters.

Second, final selector stability is the central stability concern because it is the stage that performs milestone judgment. Table 26 therefore reports four runs of the final selector using the same model and MILES pipeline configuration. The entries are final timeline SA Scores for the final compressed timelines.

The selector replay is highly stable: the maximum SA Score spread is 0.0004 on Entities, 0.0005 on Crisis, and 0.0007 on T17. Appendix L then complements this same-setting replay with controlled selector-side changes in pool size, prompt rules, model choice, and input fields. Third, evaluation stability is addressed by the semantic-alignment validation in Appendix A and the paired task-level tests in Appendix G.

The random shortlist baseline in Table 5 uses the fixed seeds 13701, 13702, and 13703 and is reported as a three-seed aggregate.

O Software Environment and Execution Dependencies

This appendix expands the software and execution environment referenced in Section 4.7. Experiments were executed inside a Docker container on a server with eight NVIDIA A800 GPUs. The host machine reported NVIDIA-SMI 535.104.05, driver version 535.104.05, and CUDA 12.2. This host CUDA value is reported only as the driver-visible environment; the experiment runtime is defined by the container-level Python dependency snapshot.

The submitted code package contains two dependency files. The minimal `requirements.txt` is sufficient for the compact API-based reproduction scripts, which only require standard Python modules plus HTTP requests. The full container snapshot is provided as `requirements_full.txt`. It contains 301 pinned Python packages. The main runtime stack includes PyTorch 2.8.0+`cu128`, CUDA 12.8 Python/runtime packages, vLLM 0.11.0, Transformers 4.57.1, Accelerate 1.10.1, Datasets 4.3.0, NumPy 1.26.4, Pandas 2.3.3, SciPy 1.16.2, Requests 2.32.5, and an OpenAI-compatible client stack.

The supplementary code is organized around the three MILES stages. `run_pipeline.sh` is the wrapper entry point. `src/miles_event_extraction.py` builds topic-aware event candidates from article JSONL files; `src/date_cluster_shortlisting.py` groups events by date and builds the visible top-nL shortlist; `src/multiview_selector.py` calls an API-compatible endpoint for multiview selection, union, and compression; and `src/evaluate_date_metrics.py` recomputes conventional date-level metrics from selections and references. API-based stages use neutral environment variables `MILES_API_KEY`, `MILES_API_URL`, and `MILES_MODEL`; no local filesystem paths or private endpoint identifiers are required in the scripts.

The compact selector illustrates the core selection procedure. The exact-length repair used by the sealed experimental runs is specified in Appendix B, with templates P07–P08 in `MILES_prompts_v1.md`; it is not implemented in this compact script.

P Supplementary Package Manifest

This appendix summarizes the supplementary archive referenced in Section 4.7. The archive is organized into a code package and a data/result package.

Code package. The top-level `MANIFEST.json` and `code/README.md` describe the submitted files. The wrapper `code/run_pipeline.sh` runs the compact API-based reproduction pipeline. Stage 1 is implemented by `code/src/miles_event_extraction.py`, which applies the topic-aware event extraction prompt stored in `code/prompts/`. Stage 2 is implemented by `code/src/date_cluster_shortlisting.py`, which groups raw events by date and writes the visible top-nL shortlist. Stage 3 is implemented by `code/src/multiview_selector.py`, which performs multiview selection, union, and compression. The formal runs additionally use exact-length repair when triggered; the compact script omits this step, whose protocol is documented in Appendix B. Conventional date-based evaluation is implemented by `code/src/evaluate_date_metrics.py`. The code package also includes `code/requirements.txt` for the compact scripts and `code/requirements_full.txt` for the container-level dependency snapshot.

Data and result package. The file `data_results_gt/README.md` gives the package-level layout. Gold timelines are stored in `data_results_gt/gt/` as one JSONL file per dataset: `entities_gt_reference.jsonl`, `t17_gt_reference.jsonl`, and `crisis_gt_reference.jsonl`. The raw article directory contains only `datasets/articles/README.md`; full article text is not included in the anonymous archive. MILES intermediate and final outputs are stored under `data_results_gt/miles/`: Stage-1 event pools in `stage1_events/`, visible shortlists and selector packs in `shortlists/`, and final predicted timelines in `final_selections/`. Baseline outputs and semantic-alignment metrics are stored under `data_results_gt/results/baselines/`, with separate subdirectories for `llm_tls/`, `datewise/`, and `chronos_closed_retrieval/`. The main aggregate result table is reproducible from `main_result_metrics.json` under `data_results_gt/results/metrics/`. Task-level MILES-versus-LLM-TLS semantic-alignment metrics for the three datasets are additionally stored in `data_results_gt/results/miles_sa/`. Per-topic LLM call accounting is stored in `data_results_gt/results/call_counts/`, including both the dataset-level summary and the complete per-topic table.

Companion prompt document. The separately supplied `MILES_prompts_v1.md` contains P01–P11 (full templates and request constructors) and D01–D10 (compact interfaces). It distinguishes dataset-specific experimental templates from compact reproduction interfaces. P07–P08 document formal-run repair; P10–P11 document the compact selector and extractor. This document accompanies the manuscript separately from the code and data/result archive described above.