From Moments to Milestones: Incremental Timeline Summarization Leveraging Large Language Models

Qisheng Hu, Geonsik Moon, and **Hwee Tou Ng**
Department of Computer Science, National University of Singapore
qishenghu@u.nus.edu, gsmoon97@u.nus.edu, nght@comp.nus.edu.sg

Abstract

Timeline summarization (TLS) is essential for distilling coherent narratives from a vast collection of texts, tracing the progression of events and topics over time. Prior research typically focuses on either event or topic timeline summarization, neglecting the potential synergy of these two forms. In this study, we bridge this gap by introducing a novel approach that leverages large language models (LLMs) for generating both event and topic timelines. Our approach diverges from conventional TLS by prioritizing event detection, leveraging LLMs as pseudo-oracles for incremental event clustering and construction of timelines from a text stream. As a result, it produces a more interpretable pipeline. Empirical evaluation across four TLS benchmarks reveals that our approach outperforms the best prior published approaches, highlighting the potential of LLMs in timeline summarization for real-world applications.[1]

1 Introduction

Condensing a vast collection of texts into comprehensible summaries is a crucial and challenging task. Timeline summarization (TLS) represents a specialized topic within this area, focusing on distilling event narratives from many texts to summarize the development of specific events or topics over time. Research on TLS to date is split into two primary categories, each with its objective and granularity. For **event timeline summarization** (Faghihi et al., 2022), each timeline is a temporally sorted list of momentary updates about a certain event. The task for event TLS is to generate abstractive summaries of events over time from a text stream such as tweets. On the other hand, **topic timeline summarization** (Tran et al., 2015a; Ghalandari and Ifrim, 2020; Li et al., 2021) aims to systematically arrange timestamped descriptions of milestone events from a corpus of documents, such as compiling a timeline of a public figure's career from historical news articles. As illustrated in Figure 1, event TLS equips users with detailed summaries to follow the evolution of particular incidents, while topic TLS facilitates rapid understanding of milestone events about a topic.

Figure 1: Examples of event and topic TLS produced by LLM-TLS. For the topic TLS example about 'Steve Jobs', events matching the reference timeline are highlighted in blue and bolded.

While the task of timeline summarization is essential, it has not garnered as much attention in recent research. Prevailing studies have mainly focused on either event or topic-level TLS, overlooking the potential synergy between the two. Though distinct in their objectives, inputs, and outputs, both forms of TLS converge on the fundamental goal of distilling and structuring events from a vast collection of texts. Recognizing this shared purpose, our study seeks to bridge this gap by emphasizing event clustering as the common subtask for both

[1] Source code available at https://github.com/nusnlp/LLM-TLS

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7232–7246

event and topic TLS.

In contrast to the retrospective approach common in prior studies, we approach TLS in a streaming context, envisioning that both event and topic timelines should evolve as time progresses and as new texts are received. In this context, the text stream can be envisaged as an evolving event graph, where nodes represent entities (e.g., tweets, event descriptions) and edges indicate common event associations. To align with the dynamic nature of real-world events, we advocate for an incremental approach that is more suited to the continual evolution of timelines. We propose **LLM-TLS**, a novel approach that harnesses large language models (LLMs) for the incremental generation and summarization of timelines.

Large language models, pre-trained on vast text corpora, have proven effective in knowledge-intensive tasks and are comparable to crowd workers in data annotation tasks (Ding et al., 2023; Li et al., 2023a). Recent studies have also unveiled the potential of LLMs in roles traditionally fulfilled by large-scale manual efforts. (Huang et al., 2023; Kuzman et al., 2023). Inspired by these advances, we use LLMs as pseudo-oracles to emulate crowdsourced event clustering for TLS. Crowdsourced clustering (Chen et al., 2023b) typically entails grouping n items into K clusters, based on responses to pairwise queries from crowd workers, such as "Are items i and j part of the same cluster?" Similarly, we propose using LLMs as pseudo-oracles to resolve queries such as "Do events i and j refer to the same event?" or "Does the new tweet i pertain to the same event as the timeline j?" This approach is grounded in the hypothesis that the knowledge encapsulated within LLMs enables them to discern the relatedness between event descriptions, thus serving as a viable proxy for human judgment in this context.

Our contributions are summarized as follows:

- We propose a novel LLM-TLS approach, which pioneers the use of LLMs for incremental summarization, applicable to both event and topic timelines.
- LLM-TLS offers a shift towards incremental event clustering and abstractive summarization in the streaming context, better suited to the evolving nature of real-world text streams.
- Empirical evaluation demonstrates that LLM-TLS outperforms the best prior published approaches on four TLS benchmarks, validating its effectiveness for timeline summarization.

2 Related Work

2.1 Event Timeline Summarization

Event TLS aims to summarize evolving events from a text stream (Aslam et al., 2013; Kedzie et al., 2015; Faghihi et al., 2022). A notable contribution was made by Faghihi et al. (2022), who introduced the concept of generating timeline summaries for crisis events from noisy tweet streams and created the CrisisLTLSum dataset. CrisisLTLSum challenges conventional summarization by demanding the extraction and summarization of event timelines from noisy tweet streams, underscoring the necessity for event-level TLS methods adaptable to the evolving nature of real-world events.

2.2 Topic Timeline Summarization

Direct summarization Previous studies have explored TLS as an extension of multi-document summarization (MDS), aiming to directly generate summaries from multiple documents (Allan et al., 2001; Yan et al., 2011; Li and Li, 2013; Zhao et al., 2013; Nguyen et al., 2014; Yu et al., 2021). Chieu and Lee (2004) approached TLS as a sentence-level retrieval task, utilizing ranking functions. Martschat and Markert (2018) showed that optimization models used in MDS could be adapted for TLS using submodular functions.

Datewise summarization Datewise summarization conceptualizes TLS as a two-step procedure: identifying key dates and summarizing events for those dates. Ghalandari and Ifrim (2020) used supervised learning for date selection based on date features, followed by sentence selection through a ranking process. Tran et al. (2015b) developed a supervised graphical model utilizing graph-based ranking for identifying salient dates. Steen and Markert (2019) and Chen et al. (2023a) highlighted the TLS challenges of accurate date selection and abstractive summary generation, and introduced a memory-based model for TLS which integrates time-series data and event information.

Event detection Event detection and graph modeling for TLS have been introduced in recent years. Affinity propagation has been utilized for event detection by Duan et al. (2020) and Yu et al. (2021), while Ghalandari and Ifrim (2020) developed event graphs using temporal heuristics and

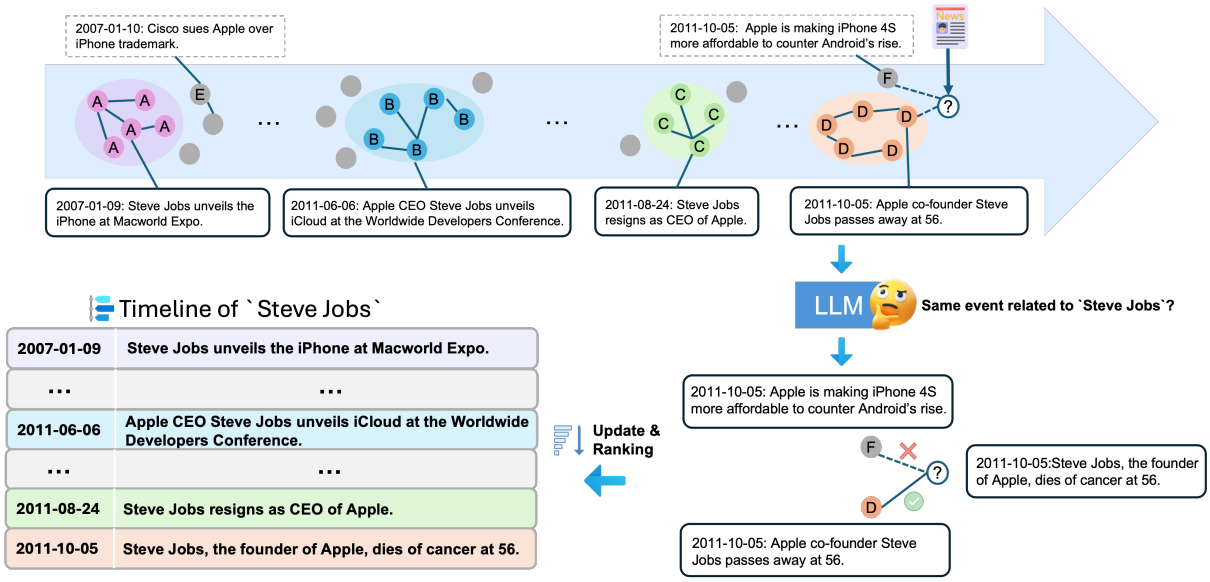

Figure 2: An overview of the LLM-TLS pipeline for topic TLS using the topic 'Steve Jobs' as an example. Events are grouped into clusters (A-F), with grey circles indicating less significant events. A newly arrived article is summarized into an event description labeled '?', and its relevance is assessed by comparing it with neighboring event nodes. Clusters are ranked and key events are chronologically sorted to showcase milestones in Steve Jobs' life and career.

Markov clustering for clustering. You et al. (2022) proposed modeling a heterogeneous graph for TLS. Li et al. (2021) introduced a technique for compressing event graphs for timeline generation.

3 Problem Definition

Our problem definition for event TLS (Faghihi et al., 2022) and topic TLS (Ghalandari and Ifrim, 2020) dovetails with the respective event and topic TLS datasets used in prior research.

3.1 Event Timeline Summarization

The input is a stream of tweets $S = \{t_1, t_2, \ldots, t_n\}$ in chronological order containing tweets belonging to different timelines that are intermixed. The objective is two-fold:

Timeline extraction Incrementally group tweets that pertain to the same event, and partition tweets into event timelines $T_{event} = \{T_1, T_2, \ldots, T_m\}$, where each timeline T_i contains tweets about an event.

Timeline summarization Produce a summary for each timeline that reflects the progression of an event. These summaries can then be evaluated with ground-truth summaries.

3.2 Topic Timeline Summarization

The input comprises a temporally ordered collection of news articles A, a set of topic query keyphrases Q, a value l denoting the number of dates, and a value k denoting the number of sentences per date. The objective is to construct a timeline T_{topic} comprising l dates, each populated with k sentences. A reference timeline r comprising l distinct dates, each associated with k sentences, is used for evaluation.

4 Method

We propose **LLM-TLS**, a novel approach that leverages a large language model (specifically Llama2-13B (Touvron et al., 2023)) to address the challenges in event and topic TLS. Our method involves a two-step iterative process: **cluster** and **summarize**, to construct coherent timelines.

Event TLS focuses on the extraction of event timelines from a noisy text stream, while topic TLS focuses on including key timestamps in timelines. However, for a broad topic where many events can occur within a single day, previous date selection methods struggle to select key dates for a timeline. Hence, we advocate for a strategy that mirrors human judgment, prioritizing the selection of milestone events. The adaptability of our event clustering component is crucial, as it caters to the distinct requirements of both event and topic TLS.

We model a text stream as an evolving event graph, where nodes are entities (e.g., tweets, event descriptions), and edges indicate common event as-

Algorithm 1 Event LLM-TLS

Input: A stream of tweets S, a new tweet t_i arrived at time i, the tweet database D, event clusters G, the number for retrieval N.
Output: Event timeline T that t_i belongs to, with summary e.

$t_i \leftarrow$ ARRIVE(S)
$Tweets \leftarrow$ RETRIEVE(t_i, N, D)
$Timelines \leftarrow$ MAPTOTIMELINES($Tweets$)
$D \leftarrow$ ADDTODATABASE(t_i, D)
$Edges \leftarrow \{\}$
for T_j in $Timelines$ **do**
  if ISSAMEEVENT(t_i, T_j) **then**
    $Edges \leftarrow$ ADD($[t_i, T_j]$, $Edges$)
  end if
end for
$T \leftarrow$ NEARESTTIMELINE($Edges$)
$G \leftarrow$ UPDATE($[t_i, T], G$)
if $OutputNeeded$ **then**
  $e \leftarrow$ SUMMARIZE(T)
  return T, e
end if

Algorithm 2 Topic LLM-TLS

Input: An article stream of one topic A, a set of topic keywords Q, a new article a_i arrived at time i, the event database D, event clusters G, the number for retrieval N, the number of dates l in the timeline, the number of sentences per date k in the timeline.
Output: Topic timeline T_{topic} with l timestamped event descriptions, each with k sentences.

$a_i \leftarrow$ ARRIVE(A)
$e_i \leftarrow$ KEYWORDEVENTSUM(a_i, Q)
$Events \leftarrow$ RETRIEVE(e_i, N, D)
$D \leftarrow$ ADDTODATABASE(e_i, D)
$Edges \leftarrow \{\}$
for e_j in $Events$ **do**
  if ISSAMEEVENT(e_i, e_j) **then**
    $Edges \leftarrow$ ADD($[e_i, e_j]$, $Edges$)
  end if
end for
$G \leftarrow$ UPDATE($Edges, G$)
if $OutputNeeded$ **then**
  $T_{topic} \leftarrow \{\}$
  $C \leftarrow$ RANKCLUSTERS(G, l)
  $C \leftarrow$ SORTBYTIME(C)
  for c in C **do**
    $t_c \leftarrow$ SUMMARIZE(c, k)
    $T_{topic} \leftarrow$ ADD(t_c, T_{topic})
  end for
  return T_{topic}
end if

sociations. Our proposal involves using LLMs like crowdsourced clustering (Chen et al., 2023b). We employ pairwise querying to determine either the relatedness of two texts or the membership of a text in a cluster, ensuring that each cluster corresponds to a single event.

4.1 Event LLM-TLS

4.1.1 Timeline Extraction

Upon receipt of a new tweet t_i, we employ the General Text Embeddings (GTE) model[2] (Li et al., 2023b), which excels on the MTEB benchmark (Muennighoff et al., 2023), to represent the tweet in a vector for retrieval. This vector is then queried against a vector database[3] to identify the top N similar tweets using cosine similarity, denoted as RETRIEVE in Algorithm 1. Each retrieved tweet is then mapped to the existing timeline to which it has already been assigned, a step denoted as MAPTOTIMELINES. A temporal heuristic confines the search to tweets within a 24-hour window of the query tweet for temporal relevance.

To determine the relevance of the new tweet t_i to an evolving timeline, denoted as ISSAMEEVENT, we fine-tuned an LLM for membership classification. Based on the annotated CrisisLTLSum dataset, we created a training set such that each sample includes an evolving timeline and a query tweet, annotated based on whether the tweet should be included in the timeline or excluded for reasons such as "not relevant", "repetitive", or "not informative". This training set was used for fine-tuning the LLM, allowing for clear criteria for membership classification. The fine-tuned LLM is then used for pairwise classification to determine if a new tweet belongs to an evolving timeline, adhering to the human-annotated standard. The prompt used is in Appendix A.1.1.

In post-processing, denoted as NEARESTTIMELINE and UPDATE, if a tweet is relevant to multiple timelines, it is assigned to the timeline with the nearest tweet in time. Conversely, a tweet that does not belong to any existing timeline indicates a new event and gives rise to a new timeline.

[2]Here, "GTE model" refers to the gte-large variant.
[3]https://www.trychroma.com/

4.1.2 Timeline Summarization

Timeline summarization is achieved by fine-tuning the LLM with the CrisisLTLSum training set, as detailed in Section 5.3.2. The LLM is provided with a context consisting of relevant tweets only for each timeline. The training process utilizes these relevant tweets as input, aiming to align the LLM's output with the provided ground-truth summaries. The prompt used can be found in Appendix A.1.2. This fine-tuning process equips the LLM to generate good summaries for timelines, ensuring that the summaries are accurately reflective of the event timeline's progression.

4.2 Topic LLM-TLS

4.2.1 Keyword Event Summarization

Prior research has highlighted the proficiency of LLMs in query-based summarization (Yang et al., 2023). Building on this capability, we leverage LLMs to perform event summarization related to the topic keywords upon the arrival of a new article, outlined as KEYWORDEVENTSUM in Algorithm 2. We designed a one-shot prompt to instruct the LLM to succinctly summarize the most pivotal event linked to the topic keywords. Event summaries are produced in a standardized format with a date and a brief event description for temporal anchoring, such as "2002-07-17: Apple CEO Steve Jobs announces the iPod at MacWorld in New York". An example prompt used can be found in Appendix A.2.1. This approach not only streamlines the summarization process but also organizes the summarized events into a clearly defined format for the subsequent clustering process.

4.2.2 Clustering

Different from the previous LLM few-shot clustering method (Viswanathan et al., 2024), which employs LLM to refine clustering outcomes, we consider the pre-trained LLM to be a pseudo-oracle for membership classification similar to crowdsourced clustering (Chen et al., 2023b). Similar to Section 4.1.1, the event summary is encoded using the GTE model and queried against a vector database[3] to find the top N similar event summaries, determined by cosine similarity. This step is denoted as RETRIEVE in Algorithm 2.

To determine if two summaries pertain to the same event, denoted as ISSAMEEVENT, we employed few-shot prompting with an LLM to serve as a pseudo-oracle for pairwise classification. This process is exemplified in Appendix A.2.2. An affirmative LLM response prompts the creation of an edge between the nodes. To ensure temporal relevance, we also apply a date heuristic that filters candidates to ensure they share the same date with the query event. Through iteratively executing this process, an event graph emerges, where a cluster represents an individual event.

Dataset	# Timelines	# Tweets
Train	550	4747
Dev	60	499
Test	162	1431

Table 1: Statistics of the event TLS dataset (CrisisLTLSum).

	T17	Crisis	Entities
# of topics	9	4	47
# of timelines	19	22	47
Avg. # of articles	508	2310	959
Avg. # of pub dates	124	307	600
Avg. duration (days)	212	343	4437
Avg. l	36	29	23
Avg. k	2.9	1.3	1.2

Table 2: Statistics of the topic TLS datasets.

4.2.3 Timeline Construction

To form a timeline, milestone event selection involves ranking each event cluster based on its connected node count, grounded in the principle that a higher node count signifies an event's increased prominence. This ranking process, outlined as RANKCLUSTERS, determines the top l clusters.

Within each selected cluster, the TextRank algorithm (Mihalcea and Tarau, 2004), a graph-based ranking algorithm, is applied in combination with the GTE embeddings. TextRank assesses the importance of nodes within its cluster. The top k nodes are identified to accurately represent the event's narrative. The concatenated event descriptions from these nodes form each cluster's summary. The concatenation of all clusters' summaries, sorted in chronological order denoted as SORTBYTIME, forms the complete timeline.

5 Experiments

5.1 Datasets

We conducted experiments on one dataset for event TLS and three datasets for topic TLS. The statistics of the datasets are summarized in Tables 1 and 2.

Event TLS Dataset The CrisisLTLSum (Faghihi et al., 2022) dataset is an event timeline dataset containing tweets about various crisis events. The dataset includes both clean and noisy tweets for each crisis event timeline. Noisy tweets typically consist of off-topic comments, and irrelevant or repetitive information that lacks relevance to the coherent narrative of the event at hand. Since some of the tweets had already been deleted when we downloaded the tweets ourselves to re-create the CrisisLTLSum dataset, we were only able to re-create 772 out of the original 1000 timelines. For each partition of the dataset (train, dev, test), we included all tweets that we downloaded for these timelines, both clean and noisy. We mixed and ordered in chronological order all tweets from all timelines to form a continuous stream of tweets.

Topic TLS Datasets The topic-based TLS datasets compile an extensive range of news articles organized by various topics. Each article is temporally tagged at the sentence level utilizing HeidelTime[4]. The T17 dataset (Tran et al., 2013) and the CRISIS dataset (Tran et al., 2015a) contain 9 and 4 topics, respectively, while the ENTITIES dataset (Ghalandari and Ifrim, 2020) contains 47 distinct topics.

These topic TLS datasets exhibit diverse characteristics in terms of thematic range, volume, and temporal extent. The number of articles associated with each topic ranges from a few hundred in T17 to several thousand in CRISIS. The period covered by these timelines is also variable, with T17 documenting events over 7 months, and ENTITIES extending up to 12 years.

5.2 Evaluation Metrics

5.2.1 Event TLS Evaluation

In evaluating event clustering for timeline extraction, we align each ground-truth timeline with the generated timeline based on the maximum tweet overlap, as detailed in Appendix B.4. Performance is measured by precision, recall, and F1. Detailed definitions are provided in Appendix B.3. For summarization, we assess with ROUGE F1 scores (ROUGE-1, ROUGE-2, ROUGE-L).

5.2.2 Topic TLS Evaluation

Alignment-based ROUGE F1-score (AR) This metric (Martschat and Markert, 2017) evaluates the textual overlap between the generated timeline and the reference timeline. Its alignment is based on temporal and semantic distance, using ROUGE-1 for unigram overlap and ROUGE-2 for bigram overlap in the summaries.

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

5.3 Event-TLS Experimental Settings

5.3.1 Timeline Extraction Task

We assess the clustering performance in timeline extraction in two distinct settings:

RETRIEVAL New tweets are embedded and the top N (N = 20) similar tweets with their timelines are retrieved. The fine-tuned LLM is then employed to perform membership classification between the query tweet and the retrieved timelines, determining the tweet's inclusion in a candidate timeline.

GLOBAL Each new tweet undergoes membership classification against all evolving timelines in the database. This setting contrasts with RETRIEVAL by expanding the candidate timelines to cover the entire database.

5.3.2 Timeline Summarization Task

To evaluate the quality of summaries generated from timelines automatically extracted, we compare our approach (LLM-TLS in global or retrieval mode) against the reproduced experiments of BART (Lewis et al., 2020) and Distill-BART (Shleifer and Rush, 2020) using CrisisLTLSum's oracle setting (Faghihi et al., 2022), where each input timeline is the gold-standard timeline that includes only all the relevant tweets for the timeline and excludes all noisy tweets that do not belong to the timeline. We also evaluated fine-tuned Llama2-13B in this oracle setting. As a comparison, we evaluated GPT-4 (Achiam et al., 2023) by giving it one or five in-context learning examples, consisting of timelines and their gold-standard summaries.

5.4 Topic-TLS Experimental Settings

We conducted experiments on the topic TLS datasets using the LLM-TLS approach with Llama2-13B (Touvron et al., 2023). We compare the performance of LLM-TLS with several prior works.

[4] https://github.com/HeidelTime/heideltime

Mode	Precision	Recall	F1
Global	87.27	**82.10**	**82.54**
Retrieval	**89.63**	79.74	81.94

Table 3: Clustering performance on the test set of event TLS (in%).

Model	Mode	R1	R2	RL
BART	Oracle	48.54	26.33	35.14
DistillBART	Oracle	49.36	26.81	35.99
GPT-$4_{1\text{-shot}}$	Oracle	50.02	26.23	35.88
GPT-$4_{5\text{-shot}}$	Oracle	50.79	27.10	36.74
Llama2-13B	Oracle	**51.01**	**29.66**	**38.95**
LLM-TLS	Global	47.83	27.34	36.81
LLM-TLS	Retrieval	**49.06**	**28.45**	**38.04**

Table 4: Summarization performance on the test set of event TLS (in%).

MARTSCHAT Martschat and Markert (2018) used submodular optimization for sentence selection, balancing content coverage with temporal and textual diversity.

DATEWISE Ghalandari and Ifrim (2020) used a supervised learning approach for date selection based on date features, combined with unsupervised summarization for each date.

SDF La Quatra et al. (2021) used "summarize date first", a strategy focusing on summarizing dates first, followed by summary-driven graph-ranking for date selection.

CLUST Ghalandari and Ifrim (2020) used an event clustering method based on Markov clustering with clusters ranked by the frequency of mentioned cluster date throughout the input collection.

EGC Li et al. (2021) used an event graph modeling method, employing time-aware optimal transport to compress the graph into a salient sub-graph for event selection.

5.4.1 Hybrid Input

The baselines in our study incorporate sentences with temporal tagging as input. Ghalandari and Ifrim (2020) have shown superior performance of DATEWISE when compared to only using titles as input. This suggests that temporally tagged sentences provide valuable information for the TLS task. In LLM-TLS_{hybrid}, the hybrid input mode of our LLM-TLS, all sentences in an input article that are tagged with date by HeidelTime (in addition to the event summary generated by the LLM) are candidate events to be clustered. Hence, LLM-TLS_{hybrid} enriches the set of candidate events, and results in higher-quality topic timelines.

6 Results & Analysis

6.1 Event LLM-TLS Performance

Clustering evaluation The evaluation presented in Table 3 compares the GLOBAL and RETRIEVAL modes and also serves as an ablation study, emphasizing the necessity of the retrieval component in our pipeline. The GLOBAL mode yields a high recall by comparing against all timelines in the database, but at the cost of increased processing time—approximately 35 minutes per experiment. In contrast, RETRIEVAL mode attains 89.63 precision and an 81.94 F1 score in just 7 minutes, an 80% time reduction. This efficiency makes a compelling case for the retrieval component in reducing complexity. The balanced performance confirms the efficacy of the LLM-TLS approach for real-time event summarization and tracking.

Summarization evaluation Table 4 outlines the ROUGE scores for models under different settings. In both GLOBAL and RETRIEVAL, which contend with the noise in the input text stream, LLM-TLS with Llama2-13B outperforms the ORACLE settings of BART and DistillBART. While GPT-4 shows a strong baseline, LLM-TLS consistently leads in Rouge-2 and Rouge-L metrics. LLM-TLS excels in RETRIEVAL, achieving notable ROUGE scores, which indicates its efficacy in distilling pertinent details from timelines. These results showcase our approach's adeptness at discerning and emphasizing event detail information from the tweet stream.

6.2 Topic LLM-TLS Performance

Table 5 compares the LLM-TLS pipeline with various baselines, demonstrating LLM-TLS's effectiveness. LLM-TLS outperforms others in the CRISIS and ENTITIES datasets, with superior AR-1, AR-2, and Date-F1 scores. Notably, the high Date-F1 scores indicate the approach's superior capability in selecting relevant dates, which is crucial for creating accurate timelines that align closely with human-annotated ground-truth timelines.

Event detection comparison CLUST and EGC, rooted in event-driven clustering, serve as the event-

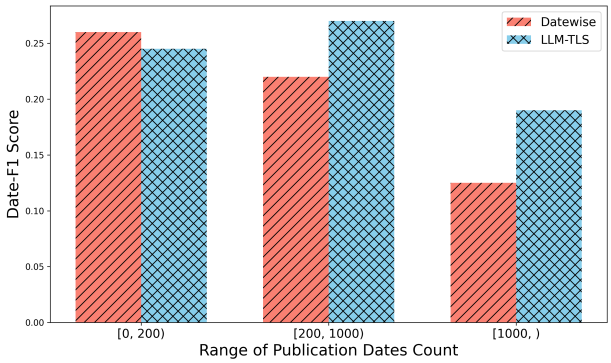

Figure 3: Comparison of Date-F1 scores between DATEWISE and LLM-TLS for ENTITIES timelines, with respect to different numbers of publication dates.

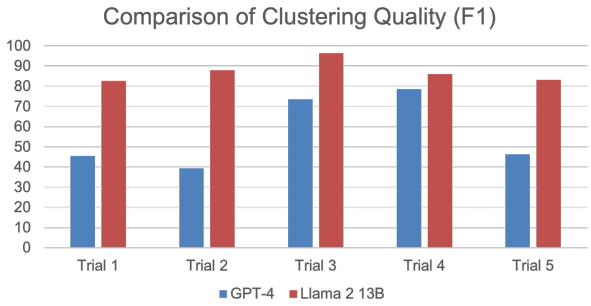

Figure 4: Comparison of clustering quality (F1) between GPT-4 and LLM-TLS (Llama2-13B) on sampled timelines over 5 trials.

wise baselines to compare with LLM-TLS. CLUST employs temporal proximity and TF-IDF for graph edges and Markov clustering for grouping events. EGC advances event graph creation with event coreference. Overall, LLM-TLS$_{\text{hybrid}}$ leverages an LLM for edge connection, enhancing event identification and graph construction, as evidenced by its improved Date-F1 scores on all three datasets.

Event-wise versus date-wise We analyze the performance variation w.r.t the number of publication dates. Ghalandari and Ifrim (2020) have suggested that an increased number of dates generally decreases performance. As shown in Figure 3, both methods demonstrate similar levels of performance when the number of publication dates is less than 200. Between 200 to 1000, LLM-TLS outperforms DATEWISE with a higher score. Beyond 1000 dates, the performance decreases more noticeably for DATEWISE than LLM-TLS. This indicates that LLM-TLS, which prioritizes milestone events, is better equipped to handle long-range topic TLS tasks. It suggests that LLM-TLS can indeed distill milestone events in a way that conforms better to human preferences for a large collection of articles covering a long period.

For further ablation studies, we conducted additional experiments using article headlines, the first sentence, and the first 3 sentences as summaries on the ENTITIES dataset. Our findings in Table 6 suggest that employing LLM for event summarization significantly enhances performance, bringing more identifiable event description and underscoring its necessity in our approach. Table 7 presents experiments with different values of N, the number of events retrieved. Overall, the change of N does not materially affect performance.

7 Discussion

Why not just use GPT-4? We explored GPT-4's ability to tackle timeline extraction by prompting it to cluster a set of tweets. Due to its output length limitation[7], we sampled 10 timelines with noisy tweets per trial. As shown in Figure 4, across five trials, GPT-4's average F1 score was 56.65 with a standard deviation of 16.08, suggesting significant fluctuation in handling noisy streaming data. Conversely, LLM-TLS with Llama2-13B performed much better and more consistently, achieving an average F1 score of 87.26 with a much lower standard deviation of 4.95.

8 Conclusion

In summary, we propose a novel approach that utilizes large language models (LLMs) for the purpose of incremental timeline summarization. Our approach addresses both event and topic timeline summarization (TLS) in a streaming context, which is more reflective of real-world scenarios. This represents a significant advance in the field, as demonstrated by extensive experiments conducted on multiple datasets. Our findings not only reveal the untapped potential of LLMs in enhancing timeline summarization, but also lay the foundation for further exploration of LLMs in understanding and organizing temporal information.

[5]We do not include the results of You et al., 2022 since that paper used a non-standard split of training, development, and test set, and so their results are not comparable.

[6]We employed an approximate randomization test (Koehn, 2004) with a p-value set at 0.05. Scores that were statistically significantly lower than those of LLM-TLS$_{\text{hybrid}}$ are marked with an asterisk (*).

[7]GPT-4's maximum completion token length is 4096. https://platform.openai.com/docs/models

	T17			Crisis			Entities		
	AR-1	**AR-2**	**Date-F1**	**AR-1**	**AR-2**	**Date-F1**	**AR-1**	**AR-2**	**Date-F1**
MARTSCHAT	0.105*	0.030	0.544	0.075*	0.016	0.281*	0.042*	0.009*	0.167*
DATEWISE	0.120	0.035	0.544	0.089*	0.026	0.295*	0.057*	0.017*	0.205*
SDF	0.120	0.035	0.553	0.086*	0.018	0.302*	0.051*	0.014*	0.197*
CLUST	0.082*	0.020*	0.407*	0.061*	0.013*	0.226*	0.051*	0.015*	0.174*
EGC	0.103	0.024	0.550	0.079	0.015	0.291	-	-	-
LLM-TLS	0.118	0.036	0.528	**0.112**	**0.032**	0.329	0.091	0.040	0.242
LLM-TLS$_{\text{hybrid}}$	**0.125**	**0.041**	**0.558**	0.111	0.031	**0.337**	**0.099**	**0.043**	**0.254**

Table 5: Experimental results on the T17, CRISIS, and ENTITIES datasets of Topic TLS.[56]

Method	**AR-1**	**AR-2**	**Date-F1**
Headline	0.038	0.009	0.142
First sentence	0.040	0.011	0.134
First 3 sentences	0.032	0.008	0.147
LLM-TLS	**0.091**	**0.040**	**0.242**

Table 6: Results of different summarization methods on ENTITIES.

N	**AR-1**	**AR-2**	**Date-F1**
10	0.092	0.036	0.235
20	0.091	0.040	0.242
50	0.093	0.036	0.240

Table 7: Results of different values of N on ENTITIES.

Acknowledgments

This research is supported by a grant from the Advanced Research and Technology Innovation Centre (ARTIC) project no. ELDT-RP1 (WBS no. A-8000975-00-00). We thank the anonymous reviewers for their helpful comments.

Limitations

This study faces several limitations, including the computational resources required. We explored the issue of hallucination in LLM outputs, which could compromise event detection accuracy. To evaluate the factual consistency of LLM-generated event summaries, we employed GPT-4 to analyze 1000 randomly selected article-summary pairs from the ENTITIES event summarization produced by Llama2-13B. Subsequent manual verification revealed that 4.8% of these pairs contained factual inconsistencies. Additionally, the design of prompts could impact the model's performance, underscoring the importance of meticulous prompt crafting to ensure high-quality outputs.

Ethics Statement

The research methodology adopted in this study adheres to the ethical guidelines set forth by the Association for Computational Linguistics (ACL). This study exclusively utilizes publicly accessible datasets, and all models employed are publicly accessible and are distributed under permissive licenses. Given that all experiments can be conducted locally, there is minimal risk of data leakage, thus ensuring the preservation of confidentiality and integrity.

References

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

James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal summaries of new topics. In *Proceedings of ACM SIGIR*, pages 10–18.

Javed Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Virgil Pavlu, and Tetsuya Sakai. 2013. TREC 2013 temporal summarization. In *Text Retrieval Conference*.

Xiuying Chen, Mingzhe Li, Shen Gao, Zhangming Chan, Dongyan Zhao, Xin Gao, Xiangliang Zhang, and Rui Yan. 2023a. Follow the timeline! Generating an abstractive and extractive timeline summary in chronological order. *ACM Transactions on Information Systems*, 41(1):1–30.

Yi Chen, Ramya Korlakai Vinayak, and Babak Hassibi. 2023b. Crowdsourced clustering via active querying: Practical algorithm with theoretical guarantees. In *Proceedings of AAAI*, pages 27–37.

Hai Leong Chieu and Yoong Keok Lee. 2004. Query based event extraction along a timeline. In *Proceedings of ACM SIGIR*, pages 425–432.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023.

Is GPT-3 a good data annotator? In *Proceedings of ACL*, pages 11173–11195.

Yijun Duan, Adam Jatowt, and Masatoshi Yoshikawa. 2020. Comparative timeline summarization via dynamic affinity-preserving random walk. In *Proceedings of ECAI 2020*, pages 1778–1785.

Hossein Rajaby Faghihi, Bashar Alhafni, Ke Zhang, Shihao Ran, Joel Tetreault, and Alejandro Jaimes. 2022. CrisisLTLSum: A benchmark for local crisis event timeline extraction and summarization. In *Findings of EMNLP*, pages 5455–5477.

Demian Gholipour Ghalandari and Georgiana Ifrim. 2020. Examining the state-of-the-art in news timeline summarization. In *Proceedings of ACL*, pages 1322–1334.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proceedings of ICLR 2022*.

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Is ChatGPT better than human annotators? potential and limitations of ChatGPT in explaining implicit hate speech. In *Proceedings of WWW Companion*, pages 294–297.

Chris Kedzie, Kathleen McKeown, and Fernando Diaz. 2015. Predicting salient updates for disaster summarization. In *Proceedings of ACL*, pages 1608–1617.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, pages 388–395.

Taja Kuzman, Igor Mozetič, and Nikola Ljubešić. 2023. ChatGPT: Beginning of an end of manual linguistic data annotation? Use case of automatic genre identification. *arXiv preprint arXiv:2303.03953v2*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of SOSP*, pages 611–626.

Moreno La Quatra, Luca Cagliero, Elena Baralis, Alberto Messina, and Maurizio Montagnuolo. 2021. Summarize dates first: a paradigm shift in timeline summarization. In *Proceedings of ACM SIGIR*, pages 418–427.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*, pages 7871–7880.

Jiwei Li and Sujian Li. 2013. Evolutionary hierarchical Dirichlet process for timeline summarization. In *Proceedings of ACL*, pages 556–560.

Manling Li, Tengfei Ma, Mo Yu, Lingfei Wu, Tian Gao, Heng Ji, and Kathleen McKeown. 2021. Timeline summarization based on event graph compression via time-aware optimal transport. In *Proceedings of EMNLP*, pages 6443–6456.

Minzhi Li, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy Chen, Zhengyuan Liu, and Diyi Yang. 2023a. CoAnnotating: Uncertainty-guided work allocation between human and large language models for data annotation. In *Proceedings of EMNLP*, pages 1487–1505.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Sebastian Martschat and Katja Markert. 2017. Improving ROUGE for timeline summarization. In *Proceedings of EACL*, pages 285–290.

Sebastian Martschat and Katja Markert. 2018. A temporally sensitive submodularity framework for timeline summarization. In *Proceedings of CoNLL*, pages 230–240.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of EMNLP*, pages 404–411.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of EACL*, pages 2014–2037.

Kiem-Hieu Nguyen, Xavier Tannier, and Véronique Moriceau. 2014. Ranking multidocument event descriptions for building thematic timelines. In *Proceedings of COLING*, pages 1208–1217.

Sam Shleifer and Alexander M Rush. 2020. Pre-trained summarization distillation. *arXiv preprint arXiv:2010.13002*.

Julius Steen and Katja Markert. 2019. Abstractive timeline summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 21–31.

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

Giang Tran, Mohammad Alrifai, and Eelco Herder. 2015a. Timeline summarization from relevant headlines. In *Proceedings of ECIR*, pages 245–256. Springer.

Giang Tran, Eelco Herder, and Katja Markert. 2015b. Joint graphical models for date selection in timeline summarization. In *Proceedings of ACL*, pages 1598–1607.

Giang Binh Tran, Mohammad Alrifai, and Dat Quoc Nguyen. 2013. Predicting relevant news events for timeline summaries. In *Proceedings of WWW Companion*, pages 91–92.

Vijay Viswanathan, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. 2024. Large language models enable few-shot clustering. *Transactions of ACL*, pages 321–333.

Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. 2011. Timeline generation through evolutionary trans-temporal summarization. In *Proceedings of EMNLP*, pages 433–443.

Xianjun Yang, Yan Li, Xinlu Zhang, Haifeng Chen, and Wei Cheng. 2023. Exploring the limits of ChatGPT for query or aspect-based text summarization. *arXiv preprint arXiv:2302.08081*.

Jingyi You, Dongyuan Li, Hidetaka Kamigaito, Kotaro Funakoshi, and Manabu Okumura. 2022. Joint learning-based heterogeneous graph attention network for timeline summarization. In *Proceedings of NAACL*, pages 4091–4104.

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

Xin Wayne Zhao, Yanwei Guo, Rui Yan, Yulan He, and Xiaoming Li. 2013. Timeline generation with social attention. In *Proceedings of ACM SIGIR*, pages 1061–1064.

A Prompts

A.1 Event TLS Prompts

A.1.1 Membership Classification Prompt

```
You are given a list of tweets, in chronological order, about some event in a timeline below:

{timeline_tweets}

Consider the following new tweet:

{new_tweet}

If this new tweet follows the given list of tweets in the same timeline (i.e., the new tweet is about the same event), then reply with "Yes".

If the new tweet is not relevant to the event in the given timeline, then reply with "No, it is not relevant".

If the new tweet is repetitive or redundant (i.e., it repeats information present in previous tweets in the given timeline), then reply with "No, it is repetitive".

If a new tweet is not informative (i.e., it is generic or expresses some opinion but does not add new information to the event in the given timeline), then reply with "No, it is not informative".

# Answer
```

A.1.2 Summarization Prompt

```
You are given a starting event which is defined under ##Seed. You may be given incoming information related to the starting event under ##Timeline. Write a summary combining the starting event and the incoming information. If there is no incoming information given, summarize the starting event.
##Seed
{First tweet in the timeline}

##Timeline
{Rest of the tweets in the timeline}

##Summary:
```

A.2 Topic TLS Prompts

A.2.1 Key Event Summarization Prompt

```
### Instruction
Review the news article associated with the provided keyword. Identify and summarize the most significant event related to the keyword.

### Format
YYYY-MM-DD: One-sentence Summary
######################
### Keyword 1
Bill Clinton
### Content 1
Title: Bush plea tries to rebuild US image
Publish Date: 2005-01-04
Content:
President George W. Bush yesterday appoints two former presidents...
(Rest of the news article)
```

```
### Event Related to 'Bill Clinton'
1999-02-05: Senate Republicans and Democrats vote against calling Monica Lewinsky for live testimony, setting a timeline for the conclusion of President Bill Clinton's impeachment trial.
######################
### Keyword 2
{keyword}

### Content 2
{content}

### Event Related to '{keyword}'
```

A.2.2 Membership Classification Prompt

```
Taking the timestamps into account, evaluate whether two prior news events are referring to the same event related to the keyword. If the two events occur on the same date or within a short time span, and they are about the same topic related to the keyword, then they should be considered as referring to the same event. If so, please respond directly with 'yes'. If not, respond with 'no'.
--
# Keyword
Bill Clinton
# Event 1
January 19, 2001 - The day before leaving office, Clinton agrees to give up his Arkansas law license for five years, and to pay a $25,000 fine to the state bar association, ending efforts by the Arkansas Supreme Court Committee on Professional Conduct to disbar him..
# Event 2
January 20, 2001 - Hours before leaving office, Clinton pardons 141 people, including Whitewater figure Susan McDougal and publishing heiress Patty Hearst. The most controversial pardon is that of financier Marc Rich, who had been a fugitive in Switzerland. The president also pardons his brother, Roger Clinton, who had been convicted on a cocaine charge in the 1980s.
# Answer
No.
--
# Keyword
Tiger Woods
Event 1
June 3, 2012 - With his win at the Memorial Tournament, ties Jack Nicklaus with 73 PGA Tour victories.
# Event 2
July 2, 2012 - Beats Nicklaus' PGA Tour record with the AT&T National win. Woods' 74th PGA Tour win ranks him in second place on the all-time list.
# Answer
No.
--
# Keyword
Mitt Romney
# Event 1
November 6, 2012 - Defeated in the general election by President Barack Obama. Romney wins 206 Electoral College votes to Obama's 332.
# Event 2
November 6, 2012 - President Barack Obama managed to secure his second term in office, triumphing over his Republican rival, Mitt Romney.
# Answer
Yes.
--
# Keyword
{keyword}
# Event 1
{event1}
# Event 2
{event2}
# Answer
```

B Experimental Details

B.1 Hardware & Libraries

In fine-tuning for event timeline summarization, we adopted LoRA (Hu et al., 2022), a parameter-efficient approach to fine-tuning LLM through low-rank adaptation. For both timeline extraction and summarization tasks, we fine-tuned Llama2-13B on an A100 80GB GPU.

For topic timeline summarization, we utilized an advanced library named vllm (Kwon et al., 2023), designed specifically for efficient Large Language Model (LLM) inference. The vllm library takes advantage of PagedAttention, a technique that optimizes LLM serving by managing memory more

Mode	Description	GPU Time
Event	Extraction fine-tuning	2 hr 30 min
Event	Summarize fine-tuning	45 min
Event	Clustering (GLOBAL)	35 min
Event	Clustering (RETRIEVAL)	7 min
Topic	T17	2 hr 12 min
Topic	T17 hybrid input	6 hr 42 min
Topic	CRISIS	7 hr 12 min
Topic	CRISIS hybrid input	32 hr 35 min
Topic	ENTITIES	18 hr 15 min
Topic	ENTITIES hybrid input	36 hr 20 min

Table 8: GPU hours for LLM-TLS experiments using Llama2-13B.

Parameter	Value
batchsize	16
num_epochs	3
learning_rate	1e-4
lora_r	8
lora_alpha	16
lora_dropout	0.05
lora_target_modules	q/k/v/o_proj

Table 9: Parameters for fine-tuning of timeline extraction.

effectively. Each trial in our topic TLS experiments was run on a single A100 40GB GPU.

B.2 Hyperparameter Settings

For the retrieval part for LLM-TLS, we used the gte-large model for text embedding and set the number of retrieved items N to 20.

Event LLM-TLS The hyperparameter settings for LoRA fine-tuning are listed in Tables 9 and 10.

Topic LLM-TLS We conducted experiments with Llama2-13B, setting the decoding temperature to 0. We ran each model through three few-shot trials to reduce randomness. The results reported in Table 5 are the averages across these trials.

Parameter	Value
batchsize	16
num_epochs	2
learning_rate	1e-4
lora_r	4
lora_alpha	16
lora_dropout	0.05
lora_target_modules	q_proj,v_proj

Table 10: Parameters for fine-tuning of timeline summarization.

B.3 Event TLS Clustering Metrics

- **Precision**: This metric assesses the precision of a predicted cluster in identifying relevant tweets. Precision refers to the proportion of tweets that both belong to the predicted cluster and the ground-truth timeline against all tweets in the predicted cluster. The calculation of precision is defined as:

$$Precision = \frac{Count(Tweet_{overlap})}{Count(Tweet_{prediction})}$$

- **Recall**: It measures the completeness of a predicted cluster in capturing the relevant tweets from the ground-truth timeline. It is the ratio of tweets that are common to both the predicted cluster and the ground-truth timeline to the total number of tweets in the ground-truth timeline, expressed as:

$$Recall = \frac{Count(Tweet_{overlap})}{Count(Tweet_{gold})}$$

- **F1 Score**: The harmonic mean of precision and recall, calculated as:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

The precision, recall, and F1 scores are macro-averaged across all timelines.

B.4 Alignment Details for Evaluation

The alignment process for event TLS evaluation matches each ground-truth timeline with the predicted timeline with the maximum tweet overlap. Ties are resolved using precision scores and date heuristics, ensuring unique pairing by excluding already aligned predicted timelines in subsequent matches.

1. Sort ground-truth timelines chronologically by the timestamp of their initial tweet.
2. If there is a tie in overlapping tweet counts, choose the predicted timeline with the higher precision (fewer tweets).
3. If there is a tie in precision scores, choose the predicted timeline with the temporal center nearest to the ground-truth timeline's center.
4. Ground-truth timelines without overlaps are not matched.

Mode	Precision	Recall	F1
Global	92.58	87.60	88.65
Retrieval	92.33	86.94	88.02

Table 11: Clustering performance on the dev set of event TLS (in%).

Model	Mode	R1	R2	RL
BART	Oracle	47.97	26.01	35.85
DistillBART	Oracle	48.44	26.00	35.32
GPT-$4_{1\text{-shot}}$	Oracle	48.60	25.43	35.84
GPT-$4_{5\text{-shot}}$	Oracle	49.85	25.74	36.24
Llama2-13B	Oracle	50.59	29.77	39.31
LLM-TLS	Global	49.11	28.30	38.46
LLM-TLS	Retrieval	48.97	28.54	38.53

Table 12: Summarization performance on the dev set of event TLS (in%).

C Event TLS Dev Set Results

The clustering performance and summarization performance on the development set of event TLS are given in Table 11 and Table 12, respectively.

D Direct Clustering with GPT-4

We also investigated the capability of GPT-4 to execute direct clustering for the purpose of timeline extraction. To this end, we prompted the model to organize a collection of tweets into clusters. Our experimental design included 10 sampled timelines with noisy tweets across 5 trials, utilizing the 'gpt-4-1106-preview' model.

D.1 Prompt to GPT-4

```
You are given the list of tweets below, in chronological order. You will need to break up the tweets into different timelines, where each timeline contains the tweets of one separate event sorted in chronological order. The tweets in one event should only contain tweets relevant to that event, and repetitive or redundant tweets that come later in time should be removed, and tweets that are generic or express some opinion but do not add new information to that event should be removed.
The list of tweets follow:
{Tweets of 10 sampled timelines}
```

E Examples

E.1 Event Timeline Summarization Examples

Timestamp	Tweet
2021-07-06 23:08:21	#Rodeo #I80 west bound before #WillowAve, truck fire on right shoulder started grass fire. Multiple lanes blocked. #KCBSTraffic photo credit: #CalTrans
2021-07-06 23:08:35	Vehicle Fire on Westbound I-80 East of Willow Ave in Hercules. Right Lanes Blocked.
2021-07-06 23:11:05	UPDATE: Vehicle Fire on Westbound I-80 East of Willow Ave in Hercules. All Lanes Blocked.
2021-07-06 23:14:29	Fire Start In #Rodeo: Putting up smoke but crews pounce on the blaze though some fencing may be burning. More crews en route. Burning along I80 and Hawthorne Dr.
2021-07-06 23:29:48	#Hercules Westbound #I80 before Willow Ave. Box truck fire sparked a grass fire. All lanes reportedly now blocked. Fire crews on the scene. Delays from Cumming Skyway in #Rodeo - Jeff Dorian -
2021-07-06 23:35:36	UPDATE: Vehicle Fire on Westbound I-80 East of Willow Ave in Hercules. The Left Lane is Open. Right and Center Lanes Remain Blocked.
2021-07-07 00:19:33	#Rodeo #I80 westbound be #WillowAve, only the left lane is open.
Method	**Summary**
Ground Truth #1	A vehicle fire on I-80 WB before Willow Ave/Rodeo led to multiple lane closures. At one point, all lanes were blocked but after some time the left lane opened back up to traffic. The vehicle fire spread to the vegetation but fire crews were able to extinguish and contain the incident. Traffic is delayed back to Cummings Skyway.
Ground Truth #2	On Rodeo I80 west bound before Willow Ave, a truck fire on right shoulder started grass fire. on June 7th, this resulted to traffic being blocked on the right and middle lane as only the left lane was left open as firefighters quickly pounce on the fire and was able to contain both fire quickly.
Llama2-13B	A truck fire on the right shoulder of I-80 westbound before Willow Ave in Hercules sparked a grass fire. The left lane is open, but the right and center lanes remain blocked. The truck fire is out and the grass fire is contained.

E.2 Topic Timeline Summarization Examples

Partial timeline of "Steve Jobs" generated by LLM-TLS.

Date	Event
2005-01-11	Apple CEO Steve Jobs unveils the iPod Shuffle and Mac Mini.
2006-09-12	Apple announces new iPod, iTunes movie downloads, and iTV.
2007-01-09	Apple CEO Steve Jobs unveils the iPhone, a touchscreen mobile phone with a built-in iPod and web access.
...	...
2009-01-05	Apple CEO Steve Jobs admits to a "hormone imbalance" that has caused him to lose weight.
2009-01-14	Apple CEO Steve Jobs takes medical leave of absence due to health issues.
2010-01-27	Apple unveils the iPad, a tablet computer with a 10-inch touch-sensitive screen.
2011-03-02	Apple launches iPad 2.
2011-06-06	Apple Inc. launches iCloud, a cloud computing service that allows users to store music, photos, documents, and other files online.
2011-08-24	Steve Jobs resigns as CEO of Apple.
2011-10-05	Steve Jobs, the founder of Apple, dies of cancer at the age of 56.

Partially matched timeline of "Steve Jobs" from the ground-truth timeline for comparison.

Date	Event
...	...
2007-01-09	Jobs unveils the iPhone at the Macworld conference.
...	...
2009-01-05	Writes an open letter to the public dismissing rumors about his health, claiming that his weight loss in the past year is due to a "hormone imbalance."
2009-01-14	Announces he will take a medical leave of absence until the end of June 2009. Jobs gives no details on his health issues other than that they are "more complex" than originally thought.
2010-01-27	Apple unveils the iPad, a tablet computer with a 10-inch touch-sensitive screen.
2011-03-02	Jobs receives a standing ovation when he takes the stage to unveil the iPad 2.
2011-06-06	At the Worldwide Developers Conference (WWDC) Jobs introduces iCloud the new online media storage system. Other Apple officials demo the new operating systems OS-X Lion and iOS-5.
2011-08-24	Resigns as CEO of Apple, but announces he will stay on as chairman. Tim Cook is promoted to CEO.
2011-10-05	Steve Jobs dies at the age of 56.