Table 1: The detailed dataset statistics of MidEast-TE.

λ	minimum cluster size	#clusters	%outlier doc	%outlier atomic events	#atomic events			#time span (days)		
					max	min	avr	max	min	avr
0	10	1,653	53.93	54.60	3,257	10	145.82	2,583	2	1682.16
1	5	9,126	46.13	46.82	1,318	5	30.96	1,018	1	22.74
1	10	3,750	52.52	52.52	4,541	10	67.27	2,568	2	39.25
1	30	973	60.19	59.33	10,342	34	222.08	2,568	2	85.53
1	50	525	63.18	62.31	10,268	55	381.38	2,567	11	114.18

Table 2: The detailed statistics of MidEast-TE and GDELT-TE.

Dataset	subset	#docs	#ttl atomic events	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	#CEs	#atomic events/CE			#time span (days)		
							max	min	avr	max	min	avr
MidEast-TE	train	90,252	160,953	2,641	218	3,452	112	10	46.63	78	2	30.11
	val	13,532	25,401	1,281	176	492	112	10	51.53	78	2	34.33
	test	12,333	22,440	1,183	177	453	112	10	49.54	78	2	34.92
	outlier	137,802	247,083	2,760	227	-	-	-	-	-	-	-
	total	253,836	455,877	2,794	234	4,397	112	10	47.49	78	2	31.08
GDELT-TE	train	101,487	441,120	1,546	233	3,561	306	10	123.88	78	2	29.82
	val	14,867	66,785	1,127	207	500	306	10	133.57	78	2	34.55
	test	13,302	65,433	1,122	204	474	306	10	138.04	78	2	34.85
	outlier	144,352	628,543	1,553	231	-	-	-	-	-	-	-
	total	273,845	1,201,881	1,555	239	4,535	306	10	126.43	78	2	30.87

address this problem by performing entity linking, which is also a typical step in previous TCE construction studies [12, 23]. We use the current SOTA LLM GPT-4 to conduct the entity linking since the entity set before linking is relatively small, thus the cost of using GPT-4 is marginal. Specifically, we first apply a K-means clustering over all the original entities to group them into multiple groups. Then we input the entities of each cluster as one batch and ask the GPT-4 to perform entity linking. We have 6322 different entities before entity linking, and we take K to be 32 and obtain a final entity set of size 2794. After the hierarchical extraction and entity linking, we finally obtain all atomic events to construct our dataset MidEast-TE. Even though the EE results in the original GDELT are noisy, it is still worthwhile to use it as an auxiliary dataset. Therefore, we keep the original EE results of GDELT, thus obtaining all atomic events for GDELT-TE.

2.2.3 Complex Event Identification. Previous approaches to constructing CEs typically require CE schema construction [23], *aka.* schema induction, which is non-trivial and often requires domain knowledge and extensive iterative work between human and machine [12, 38]. Recently, several efforts [4, 46] about online story discovery employ unsupervised clustering to automatically detect temporally evolving storylines, which is simpler and more flexible without the need to construct any CE schemas.

Following these approaches, we develop a time-aware document clustering pipeline as shown in the top of Figure 2. Specifically, we first use a SimCSE [13] pre-trained RoBERTa [28] to extract the document embedding. Then, we follow BERTopic [16] that uses UMAP [30] to reduce the dimensionality of embeddings and HDBSCAN [29] to cluster the documents into semantically related groups. In addition to semantic similarity, news articles within the same CE should also be temporally close to each other. Previous works [46] use a sliding window with a fixed length to split the timeline into pieces, which may not be optimal since different TEs often have varied time spans. To resolve this limitation, we propose a time-aware clustering approach that concatenates the temporal index of the news article to its semantic embedding, thus endowing the article clusters with elastic time spans. Moreover, we introduce an additional hyper-parameter λ to balance the weight of temporal affinity against the semantic similarity during clustering. After the clustering, we obtain a set of document clusters $\mathbf{D}^c$, each representing a CE. Apart from the CEs, there are quite several isolated news articles that do not belong to any cluster. We deem that they still offer precious global contextual information that is helpful to TE forecasting, so we keep the atomic events extracted from these documents as *outlier atomic events* in the global context.

Table 1 shows our parameter tuning in the clustering process. Specifically, after we reduce document semantic embedding to 200 using UMAP, we tune the hyper-parameter of *minimum cluster size* in HDBSCAN from $\{5, 10, 30, 50\}$, and tune the temporal feature weight λ from $\{0, 1\}$. We observe that when $\lambda = 0$, which means when the clustering does not consider the temporal feature and solely relies on the semantic features, the average time span of the cluster (CE) is extremely large as 1682 days (4.6 years). In contrast, under the setting of $\lambda = 1$, the average CE time span is largely reduced, showing the effectiveness of the time feature in constraining the temporal conciseness of CEs. We empirically take 10, which will result in a reasonable scale for the average cluster size, *w.r.t.* both #atomic events (67.27) and #time span (39.25).

After Clustering, there are a few extremely large clusters, for example, the max cluster has 4541 atomic events and spans 2568 days. We divide such superclusters into multiple reasonable smaller clusters by empirically setting a maximum #atomic events h_a and

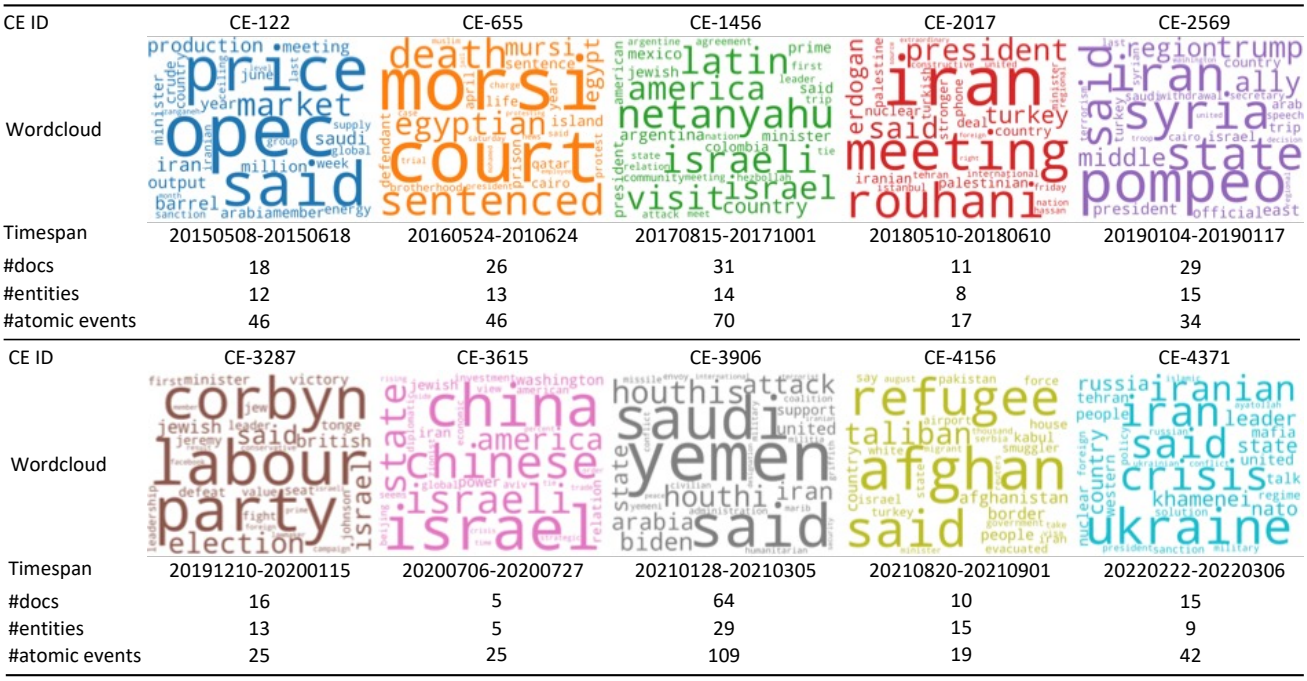

Figure 3: Illustration of ten examples of CEs, each with its CE ID, wordcloud, timespan, number of involved news articles, entities, and atomic events.

Table 3: Dataset comparison between our SCTc-TE and TCEs with schema.

Formulation	Dataset	#docs	#CEs	#atomic events
TCE w Schema	General	617	617	8,295
	IED	6,399	430	51,422
SCTc-TE (ours)	GDELT-TE	273,845	4,397	1,201,881
	MidEast-TE	274,795	4,397	455,877

a maximum #time span h_t to twice the original average. Then we remove extremely small clusters that are shorter than 2 days or have fewer than 10 atomic events. We then split CEs into training/validation/testing sets in time order to prevent information leakage in forecasting. We apply this entire pipeline to the atomic events obtained from our LLM-based EE and from GDELT. We finally obtain two large-scale SCTc-TE datasets: MidEast-TE and GDELT-TE, with the detailed statistics as shown in Table 2.

2.3 Dataset Evaluation

2.3.1 Statistics. Table 3 shows the dataset comparison between the previous formulation of TCE with Schema and our formulation of SCTc-TE. The **General** dataset is curated by LDC (LDC2020E25) and the **IED** is constructed by Li et al. [23]. Clearly, the two datasets constructed by our pipeline are much larger than the previous datasets; in particular, the number of CEs and atomic events are one magnitude larger than the previous datasets.

We compare MidEast-TE and GDELT-TE in Table 4. The events in MidEast-TE are more fine-grained due to a larger entity set. Moreover, the atomic event types defined by CAMEO are a three-layered

Table 4: Event extraction results comparison.

Dataset	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{E}\rvert$	#atomic events	% of different levels		
				1st	2nd	3rd
GDELT-TE	239	1,555	1,201,881	38.93	57.19	3.88
MidEast-TE	234	2,794	455,877	21.24	64.82	13.94

Table 5: Dataset Statistics.

Dataset	specs	train	val	test
GDELT-TE	#CEs	3,561	500	474
	#atomic events	441,120	66,785	65,433
MidEast-TE	#CEs	3,452	492	453
	#atomic events	160,953	25,401	22,440

hierarchy, and the event types at higher levels are more fine-grained. The distribution of events in different levels verifies that the events in MidEast-TE are more fine-grained. We sample 100 news articles and use GPT-3.5 to evaluate the EE performance, where the precison of MidEast-TE and GDELT-TE are 0.432 and 0.425, respectively. Event though the EE using Vicuna-13b is a little bit better than the GDELT, the overall error rate in current EE of both MidEast-TE and GDELT-TE are still high. However, we argue that the idea of using LLMs for EE is promising. If there is no budget limitation and we use more powerful LLMs, the error rate of EE could be largely reduced. We split the CEs into train/validation/testing sets according to the timestamp. Specifically, we use the CEs in the last year for

testing, the second-to-last year for validation, and the remaining of about five years for training, as shown in Table 5.

2.3.2 Evaluation. We specifically evaluate the three main steps of the dataset construction process, including clustering, event extraction, and entity linking. For the clustering, we sample some of them and plot their word clouds using tf-idf scores. The examples in Figure 3 show that the clusters are distinct with each other, and each of the cluster focus on specific complex event. For the event extraction, we adopt more powerful commercial LLMs, *i.e.,* gpt-3.5 (which is more efficient and cheap compared with the most powerful GPT-4), to evaluate the extraction precision of both MidEast-TE and the GDELT-TE (the original EE system of GDELT). We sample 100 news articles from the entire datasets and input them as well as the extracted events into the prompt, then we ask the LLMs whether the extraction results are correct or wrong. The results show that the precision of MidEast-TE is 0.432 and the precision of GDELT-TE is 0.425. Overall speaking, the extraction performances of both LLMs and GDELT are still quite low. Nevertheless, our LLM-based EE is slightly better than the original GDELT. The open sourced Vicuna-13b verifies that it is possible to be used as a zero-shot EE model. In the future, we believe the performance of EE will be largely improved when more efficient and powerful LLMs are developed. For the entity linking, since the remaining entity set is small (2794), we manually check and fix the entity linking results, ensuring the linking result is correct.

3 METHODOLOGY

Third, we propose a novel TE forecasting method, *aka.* LoGo, as shown in the bottom and right of Figure 4.

3.1 SCT Event Forecasting

Our new SCTc-TE formulation is well suited for the downstream forecasting task. The structured and time-complete properties enable SCTc-TE forecasting to be modeled by most existing TKG methods. However, previous TKG methods mix up all the atomic events and do not distinguish among different CEs, resulting in sub-optimal modeling capability. In this work, we design a novel approach named LoGo that leverages both **Lo**cal and **Glo**bal contexts for event forecasting. Specifically, given a specific query $(s, r, ?, t+1, c)$, we first obtain the entity and relation representations through the local and global context modeling modules, we then fuse these representations learned from both contexts, and finally, we use the decoder ConvTransE to predict the object o. The overall forecasting framework is shown in Figure 4.

3.1.1 Local and Global Context Modeling. Following the previous work RE-GCN [26], we design a simple module Relational-Temporal Modeling, named as RT-Mod, for the local and global context modeling. We employ a GNN model, *i.e.,* RGCN [35], to capture the relations among concurrent events at the same timestamp and an RNN model, *i.e.,* GRU [7], to preserve the temporal evolving patterns along the timeline.

Local Context Modeling. Based on the G_t^c of CE c at timestamp t, the information propagated to entity o via RGCN is $\mathbf{e}_{o,c,t}^l \in \mathbb{R}^d$, represented as:

$$\mathbf{e}_o^l = f\Big(\frac{1}{|G_t^c|} \sum_{(s,r,o)\in G_t^c} \mathbf{W}_1^l(\mathbf{e}_s^{l-1} + \mathbf{r}) + \mathbf{W}_2^l \mathbf{e}_o^{l-1}\Big), \tag{1}$$

where d is the dimensionality of the propagated information, $|G_t^c|$ represents the number of events where o is the object, $\mathbf{W}_1^l, \mathbf{W}_2^l \in \mathbb{R}^{d\times d}$ are the parameters of the GNN kernel for layer l, and $f(\cdot)$ is the RReLU activation function. It is noted that $\mathbf{e}_o^l, \mathbf{e}_s^{l-1}, \mathbf{r}, \mathbf{e}_o^{l-1}$ correspond to their representations in CE c at timestamp t, where we omit the subscript (c, t) for simplicity.

Thereafter, we aggregate the information received from multi-layer propagation and obtain the entity representation at time t under context c, defined as:

$$\mathbf{e}_o = \sum_{l=0}^{L^c} \mathbf{e}_o^l, \tag{2}$$

where $\mathbf{e}_o^0 \in \mathbb{R}^d$ is randomly initialized for each entity o. Note that the same entity share one initialized embedding over different CEs, while its representation will change differently after the graph propagation on its involved G_t^c. L^c is the number of propagation layers, which is set as the same value for all the CEs C. The representations for all entities at timestamp t are denoted as $\mathbf{E}_t^c \in \mathbb{R}^{|\mathcal{E}|\times d}$. Then, we apply a GRU to capture the temporal patterns, defined as:

$$\mathbf{E}_t^c = \mathrm{GRU}^c(\mathbf{E}_t^c, \mathbf{E}_{t-1}^c), \tag{3}$$

where GRU^c represents the GRU kernel for the local context. Following the typical implementation in TKG studies [3], we take the recent T^c steps of historical graphs to model the temporal evolving patterns for the local context. Thereby, the entity embeddings in the last GRU state retain the local contextual information for CE c, and we denote them as $\mathbf{E}^c$. All the relation embeddings in the local context are denoted as $\mathbf{R}^c \in \mathbb{R}^{|\mathcal{R}|\times d}$.

Global Context Modeling. Analogous to local context modeling, we employ a parallel branch of RT-Mod module to capture the historical patterns in the global context. Given the global context $\mathcal{G}_t$ at timestamp t, we obtain the entity representations $\mathbf{E}_t^g \in \mathbb{R}^{|\mathcal{E}|\times d}$ after L^g layers of RGCN information propagation and aggregation. Thereafter, by going through the GRU module GRU^g over the latest T^g steps in the global context $\mathcal{G}$, we achieve the entity representations $\mathbf{E}^g \in \mathbb{R}^{|\mathcal{E}|\times d}$. Note that all the entity and relation embeddings, as well as the RGCN and GRU parameters of the global context are not shared with the local context. In addition, the relation embeddings in the global context are denoted as $\mathbf{R}^g \in \mathbb{R}^{|\mathcal{R}|\times d}$.

3.1.2 Prediction and Optimization. After the local and global context modeling, we obtain two sets of representations for entities and relations, *i.e.,* $[\mathbf{E}^c, \mathbf{R}^c]$ from the local context and $[\mathbf{E}^g, \mathbf{R}^g]$ from the global context. Given a query $(s, r, ?, t+1, c)$, we lookup these embedding tables and yield the corresponding representations $[\mathbf{e}_s^c, \mathbf{r}^c]$ and $[\mathbf{e}_s^g, \mathbf{r}^g]$. We perform the element-wise sum operation to combine the representations from both contexts, resulting in the final representation $[\hat{\mathbf{e}}_s, \hat{\mathbf{r}}]$. Then, we resort to ConvTransE [36] as the decoder, formally written as:

$$\hat{\mathbf{p}}(\mathcal{E}|s, r, c, \mathbf{G}_{\le t}^c, \mathcal{G}_{\le t}) \quad = \quad \mathrm{softmax}\big(\hat{\mathbf{E}}\mathrm{ConvTransE}(\hat{\mathbf{e}}_s, \hat{\mathbf{r}})\big), \tag{4}$$