Synergizing Unsupervised Episode Detection with LLMs for Large-Scale News Events

Priyanka Kargupta **Yunyi Zhang** **Yizhu Jiao** **Siru Ouyang** **Jiawei Han**
Siebel School of Computing and Data Science
University of Illinois at Urbana-Champaign
{pk36,yzhan238,yizhuj2,siruo2,hanj}@illinois.edu

Abstract

State-of-the-art automatic event detection struggles with interpretability and adaptability to evolving large-scale key events—unlike episodic structures, which excel in these areas. Often overlooked, episodes represent cohesive clusters of core entities performing actions at a specific time and location; a partially ordered sequence of episodes can represent a key event. This paper introduces a novel task, **episode detection**, which identifies episodes within a news corpus of key event articles. Detecting episodes poses unique challenges, as they lack explicit temporal or locational markers and cannot be merged using semantic similarity alone. While large language models (LLMs) can aid with these reasoning difficulties, they suffer with long contexts typical of news corpora. To address these challenges, we introduce **EpiMine**, an unsupervised framework that identifies a key event's candidate episodes by leveraging natural episodic partitions in articles, estimated through shifts in discriminative term combinations. These candidate episodes are more cohesive and representative of true episodes, synergizing with LLMs to better interpret and refine them into final episodes. We apply EpiMine to our three diverse, real-world event datasets annotated at the episode level, where it achieves a 59.2% average gain across all metrics compared to baselines.

1 Introduction

Given the saturation of real-time news accessible at our fingertips, reading and processing a key event's critical information has become an increasingly daunting challenge. Consequently, recent work on automatic textual event detection has attempted to integrate the manner in which humans neurologically perceive/store events into textual event detection methods. Specifically, neuroscientists studying event representations in human memory find that events are stored in a top-to-bottom hierarchy, as demonstrated in Figure 1. The deeper the hierarchical event level, the more fine-grained its corresponding text granularity (Zhang et al., 2022): we consider a theme as corpus-level (all articles discussing the 2019 Hong Kong Protests), key event as document-level (an article typically discusses a full one to two day key event), episode as segment-level, and atomic action as sentence or phrase-level.

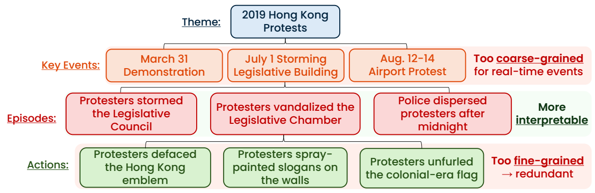
Theme:
2019 Hong Kong Protests
Key Events:
March 31 Demonstration
July 1 Storming Legislative Building
Aug. 12-14 Airport Protest
Too coarse-grained for real-time events
Episodes:
Protesters stormed the Legislative Council
Protesters vandalized the Legislative Chamber
Police dispersed protesters after midnight
More interpretable
Actions:
Protesters defaced the Hong Kong emblem
Protesters spray-painted slogans on the walls
Protesters unfurled the colonial-era flag
Too fine-grained → redundant

Figure 1: Example event structure hierarchy. Given a key event node's corpus, detect its episode children and their respective relevant text segments.

Furthermore, neurological research (Baldassano et al., 2017; Khemlani et al., 2015) indicates that events are encoded into memory as *episodic structures*. Representing events as discrete episodes helps us piece together a *coherent and concise narrative* by focusing on *meaningful* clusters of actions, reactions, and developments, rather than examining each in isolation or as a whole. Despite its strengths, **existing automatic event extraction works fail to consider the episode-level**.

For instance, key event detection focuses on identifying "a set of thematically coherent documents" for each key event (Zhang et al., 2022; Liu et al., 2023), but manually parsing large clusters of articles is inefficient and lacks interpretability. Timeline summarization (Steen and Markert, 2019; Li et al., 2021a; Gholipour Ghalandari and Ifrim, 2020; Chen et al., 2023) addresses this by providing dates and compact summaries, yet it suits historical themes better than evolving key events that require finer granularity. Event chain mining (Jiao et al., 2023) takes a fine-granularity approach by identifying temporally ordered atomic actions, but

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 29648–29663
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

its phrase-level granularity is often too fine and practically redundant for large-scale events (e.g., in Figure 1, the actions all describe the same episode). To bridge this gap, we propose the novel task of **episode detection** to pave the way for a more effective event representation.

Episode detection aims to detect episodes from a news corpus containing key event articles. An episode can be described as a cohesive cluster of potentially diverse subjects performing actions at a certain time and location, occurring as part of a larger sequence of episodes under a specific key event. We introduce **EpiMine**, which detects meaningful episodic events and their corresponding text segments in a large key event corpus, all without any level of human supervision or labeled training data. EpiMine consists of: (1) episode indicative term mining, (2) episode partitioning, (3) LLM-enhanced episode estimation, and (4) episode-segment classification. Collectively, they tackle the unique challenges of episode detection, detailed below:

Challenge 1: Episodes are not timestamped. Key event detection partitions a thematic corpus into document-level clusters by heavily relying on explicit temporal features, like publication dates, being associated with the key event articles (Zhang et al., 2022). However, this assumption fails at the episode-level, where there is no guarantee to have a distinct timestamp associated with each text segment that discusses a new episode. Fortunately, we can take advantage of the idea that journalists naturally partition news articles by sequentially discussing distinct episodes:

Example: An article likely completes its discussion of the episode A, *protesters storming the Legislative Council*, before episode B, "*protesters vandalized the Legislative Chamber*" (Figure 3).

Hence, to partition articles into distinct episode segments, EpiMine must identify whether two consecutive segments are discussing the same or different episodes— bringing us to our next challenge.

Challenge 2: Episodes contain semantically diverse actions. Each episode features a *set of unique atomic actions*, which can help determine if two segments discuss the same episode. However, for clustering actions, existing methods (Jiao et al., 2023) rely heavily on semantic similarity. This is not realistic for episode-segment clustering:

Example: "*protesters spray-painted slogans*" and "*they unfurled the colonial-era flag*" will fall under the same episode, but are semantically different and unlikely to be clustered.

Alternatively, we can identify notable (salient) terms unique to the same episode (episode A: "barriers" and "shoved"; episode B: "*defaced*" and "*walls*"), by exploiting corpus-level signals. For example, if "*defaced*" and "*walls*" are frequently mentioned together across the corpus (or their respective synonyms) and not with other terms, then they are a *discriminative co-occurrence*. When terms between two segments discriminatively co-occur, this indicates the same episode is being discussed. Conversely, if a sufficient shift in term combinations occurs, then a different episode is being discussed.

Challenge 3: Articles often do not feature all episodes. Real-time news reporting often provides an incomplete coverage of multi-day events, with individual articles potentially omitting or partially addressing key episodes. Consequently, while LLMs could assist with the first two challenges, requiring multiple articles hinders their use given their long context limitations (Li et al., 2024; Liu et al., 2024). To address this challenge, EpiMine seeks to select a minimal set of articles that maximizes both the quantity and quality of event episodes. It then merges any article partitions across these articles which likely discuss the same episode and synergizes with an LLM to provide a more fluent interpretation of the candidate episodes, accounting for the episode's core entity, actions, object, location, and time period. This allows EpiMine to finally map the remaining non-salient article segments to these episodes, pruning any candidates which are not sufficiently supported by the remaining articles. We summarize our core contributions:

- **Episode detection**: *novel* task to detect episodes & their segments from a key-event corpus.
- **EpiMine**, an unsupervised episode detection method which introduces discriminative term co-occurrence and episode partitioning.
- **Three novel datasets**, reflecting a diverse set of real-world themes and thirty global key events (no key event corpus exists for this task).
- EpiMine outperforms all baselines by, on average, a **59.2% increase across all metrics**.

Reproducibility: We provide our dataset and source code[1] to facilitate further studies.

[1] https://github.com/pkargupta/epimine

2 Related Works

2.1 Event Extraction

Event extraction has been widely studied, focusing on event detection (Liu et al., 2018a; Du and Cardie, 2020; Li et al., 2021b; Lu et al., 2021; Qi et al., 2022; Jiao et al., 2022), event relation extraction (Han et al., 2019; Wang et al., 2020; Ahmad et al., 2021), and salient event identification (Liu et al., 2018b; Jindal et al., 2020; Wilmot and Keller, 2021). Recent work has also addressed event process understanding (Zhang et al., 2020; Chen et al., 2020), though these often rely on expensive expert annotations. Some studies have introduced unsupervised methods to address annotation challenges (Weber et al., 2018; Li et al., 2020). Some overlapping work exists in topic discovery, where (Yoon et al., 2023) proposes unsupervised stream-based story discovery— computing article embeddings based on their shared temporal themes. Recently, large language models have demonstrated powerful general and event extraction-specific reasoning abilities (Pai et al., 2024; Gao et al., 2024).

However, traditional and LLM-driven methods either, (1) focus on phrase/sentence-level events (analogous to actions in Figure 1), or (2) require human-curated event ontologies, often overlooking interpretable, yet meaningful granularities and open-domain texts, which go beyond pre-defined event types. While unsupervised granular event extraction has been explored (Zhang et al., 2022; Jiao et al., 2023) at the document and phrase-level, episode detection is a more interpretable granularity that remains a largely unexplored, yet vital area.

2.2 Timeline Summarization

Timeline summarization (TLS) identifies key dates and concise descriptions for major events. Early methods were extractive, focusing on ranking events for thematic timelines (Nguyen et al., 2014) or using submodular frameworks to model temporal dimensions (Martschat and Markert, 2018). Abstractive methods later emerged, such as sentence clustering and multi-sentence compression (Steen and Markert, 2019). More recent approaches are graph-based, such as event-graph representations for salient sub-graph compression (Li et al., 2021a) and heterogeneous GATs for redundancy reduction (You et al., 2022). While they effectively summarize key events as high-level timelines, they focus on historical themes. Episode-level timelines for ongoing news remain underexplored.

3 Preliminaries

3.1 Problem Definition

Definition 1 (Episode). *An episode E_i is one of a partially ordered sequence of subevents, $\{E_1, \dots, E_i, \dots E_k\}$, of a key (major) event E, where typically $2 \leq k \leq 20$, and E_i does not overlap with E_j if $i \neq j$. Actions in the episode E_i can be either semantically similar or diverse, but typically have relatively tight time, location, and/or thematic (entities, actions, objects) proximity.*

EpiMine aims to extract episodes from a news corpus, where an episode occurs as a significant component of a larger group of episodes that fall under a specific key event. For instance, in Figure 1, without knowing Episode #1, "Protesters stormed the Legislative Council Complex", readers would not fully understand Episode #3, "Police dispersed protesters". Hence, episodes help us understand the overall key event and are especially useful for events that are currently evolving, where finer context is required for sufficiently understanding them.

Definition 2 (Episode Detection). *Given a corpus $\mathcal{D}$ about one key event, where each document $d \in \mathcal{D}$ is a news article, the task is to obtain a set of text segment clusters $\mathcal{E} = \{E_1, E_2, \dots, E_k\}$. Each episode cluster $E_i \subset \mathcal{S} = \{s_1^1, s_2^1, \dots, s_{|d|}^{|\mathcal{D}|}\}$, where $\mathcal{S}$ contains all the text segments identified in each document $d \in \mathcal{D}$, and every two clusters do not have overlapping text segments (i.e., $E_i \cap E_j = \emptyset$ for $i \neq j$).*

It is important to note that k, the number of episodes, is not known in advance and oftentimes, a news article segment may discuss either episodes of a different key event (e.g., an episode with similar aspects that occurred in a different historical key event) or multiple episodes of the current key event. Nonetheless, our goal is to detect the most relevant episodes to the current key event at hand and consequently mine the most distinctive text segments for each of these (hence our constraint of $E_i \cap E_j = \emptyset$ for $i \neq j$).

4 Methodology

To tackle episode detection, we propose a novel unsupervised framework, EpiMine. As shown in Figure 2, EpiMine consists of the following four core components: (1) **episode indicative term mining**, which identifies combinations of salient terms likely to discriminatively co-occur *within* an

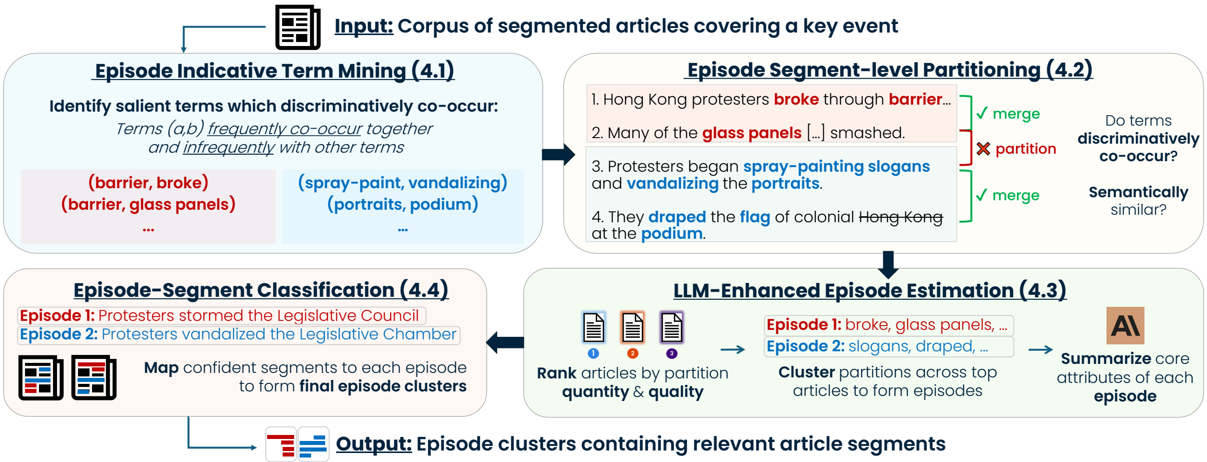
Input: Corpus of segmented articles covering a key event
Episode Indicative Term Mining (4.1)
Identify salient terms which discriminatively co-occur:
Terms (a,b) frequently co-occur together and infrequently with other terms
(barrier, broke)
(barrier, glass panels)
...
(spray-paint, vandalizing)
(portraits, podium)
...
Episode Segment-level Partitioning (4.2)
1. Hong Kong protesters broke through barrier...
2. Many of the glass panels [...] smashed.
3. Protesters began spray-painting slogans and vandalizing the portraits.
4. They draped the flag of colonial ~~Hong Kong~~ at the podium.
✓ merge
✗ partition
✓ merge
Do terms discriminatively co-occur?
Semantically similar?
Episode-Segment Classification (4.4)
Episode 1: Protesters stormed the Legislative Council
Episode 2: Protesters vandalized the Legislative Chamber
Map confident segments to each episode to form final episode clusters
LLM-Enhanced Episode Estimation (4.3)
Rank articles by partition quantity & quality
Episode 1: broke, glass panels, ...
Episode 2: slogans, draped, ...
Cluster partitions across top articles to form episodes
Summarize core attributes of each episode
Output: Episode clusters containing relevant article segments

Figure 2: We detail the overall framework of EpiMine.

episode and not *across* episodes; (2) **episode partitioning**, which partitions each article into approximate isolated episodes based on consecutive shifts in the term co-occurrence distribution, (3) **LLM-enhanced candidate episode estimation**, which clusters the top partitions into candidate episodes and utilizes LLM-based reasoning to produce fluent and meaningful episodes, and (4) **episode-segment classification**, which maps confident segments to their respective episode clusters.

4.1 Episode Indicative Term Mining

> Hong Kong protesters **broke** through the final **barrier** separating them and the interior of the city's Legislative Council Complex. Many of the **glass panels** lining the building's exterior had already been smashed in the afternoon, but it took another hour or so before protesters were able to **breach** the **metal shutters**.

> Within minutes, protesters began **spray-painting slogans** onto the **corridor walls** and **vandalizing** the **portraits** of Legislative Council presidents. They also **draped** the **flag** of colonial-era Hong Kong at the **podium** of the legislative **chamber**.

Figure 3: Natural partition between two episodes in a key event article. An episode's discriminative terms are **bolded**; salient non-discriminative terms are underlined.

Lacking supervision, our goal is to identify *potential* candidates for episodes. Episodes are often described in relation to each other and usually lack timestamps or locations consistently mentioned within their segments. For example, the phrase "police dispersed protesters" may not have a precise timestamp because it is a *response* to "protesters stormed the Legislative Council Complex," and some journalists may consider the implicit ordering adequate. Additionally, the same episode can be described using different entities and actions— journalists may *report different perspectives*. For example, both "protesters shoved against the barricades" and "the police used pepper-spray on the protesters" describe the episode "protesters stormed the Legislative Council Complex". However, they are semantically different, focused on different core entities and actions. Thus, we cannot depend on a consistent subject-action-object triple or an explicit time/location mapped to each episode in the article.

To circumvent this challenge, we exploit the idea that journalists naturally partition news articles according to episodes, forming episode fragments. For example, as shown in Figure 3, an article will likely complete its discussion of episode #1, "Protesters stormed the Legislative Council Complex" (red), before fully shifting to discussing episode #2, "protesters vandalized the Legislative Chamber" (blue). Across these episode fragments, certain *salient* terms are featured (e.g., protesters, legislative, vandalizing, podium). We adapt the idea of event salience from (Jiao et al., 2023) specifically for the task of episode detection.

Definition 3 (**Salience**). *A term is salient if it is (1)* ***distinct and significant*** *to understanding a given key event, as well as (2)* ***frequently*** *found in a key event's segments and* ***infrequently*** *in other background/general articles.*

Thus, we identify a set of salient terms for each segment within the corpus (salience score details in Appendix E).

Discriminative Co-occurrence. In Figure 3, we can see that the first episode fragment, and Episode #1 in general, features a combination of similar terms, such as "protesters", "barrier", and "breach".

Likewise, the second episode may include a combination of terms similar to "protesters", "spray-painting", and/or "flag". We note that despite some journalists choosing to only describe the protesters spray-painting, while others focus on the protesters draping the colonial-era flag, we must be able to recognize that their respective salient terms are *likely to co-occur within the same episode*.

However, we make a **novel distinction** between a co-occurrence and a ***discriminative*** co-occurrence. Salient terms a and b (e.g., "protesters" and "spray-painting") may often co-occur within an episode. However, if a also frequently co-occurs with many other terms in various episodes ("protesters broke"), a and its co-occurrences are less useful for distinguishing episodes. Thus, (a, b) is not a discriminative co-occurrence.

Definition 4 (Discriminative Co-occurrence). *A pair of terms (a, b) discriminatively co-occur if (1) they frequently appear together in episode E_i, and (2) neither a nor b appear as frequently with other terms w in other episodes $E_{\notin i}$.*

We compute the discriminative occurrence d between salient term pair (a, b) using the following:

$$\mathbf{d}(a,b) = \log\left(\frac{freq(a,b)}{\max(\bar{f}_a, \bar{f}_b)}\right) \times \log\left(\frac{|T|}{max(|F_a|, |F_b|)}\right),$$
$$\text{where } \bar{f}_a = \frac{1}{|T|}\sum_{\forall w_i \in T} freq(a, w_i), \text{ and}$$
$$F_a = \{freq(a, w_i) > 1 \; \forall \; w_i \in T\} \qquad (1)$$

The first $\log$ term ensures that the pair's co-occurrence ($freq(a, b)$) is **statistically significant** ($\geq$ the max of a and b's mean vocabulary-wide co-occurrence respectively). The second $\log$ term ensures the pair is a **discriminative match**, penalizing cases where a or b frequently co-occurs with a large portion of the salient term set T. For example, co-occurrences with "protesters" are not discriminative because "protesters" is a core entity in all episodes and thus frequently co-occurs with many terms in T. In contrast, ("slogans", "flags") is a discriminative co-occurrence since both terms frequently appear together in segments discussing episode #2 and rarely co-occur with other terms $w_i \in T$. If a and b are the same term or close synonyms (determined by statistically significant semantic similarity), they have maximum co-occurrence. By leveraging multiple articles in a large key event corpus, we have sufficient statistical support to ensure our output reflects the average realistic reporting of the key event and its episodes.

4.2 Episode Partitioning

With the ability to identify discriminative co-occurrences, we can use a key transitive property to resolve episode co-references within and across articles, where *not all combinations of an episode's discriminative terms explicitly co-occur*:

> If (a, b) and (b, c) are both discriminative co-occurrences, then (a, c) is *also likely* to be a discriminative co-occurrence.

To illustrate this, we have the following text segment excerpts of a news article (the salient and discriminative terms are *italicized*):

1. Protesters *defaced* the Hong Kong *emblem*, *spray-painted slogans*, and *unfurled* the *flag*.
2. The *portrait* of LegCo president was *defaced*.
3. A *slogan* on the *wall* reads: "The government forced us to revolt".
4. *Police* said at least 13 people had been *arrested* on *suspicion* of involvement in the pro-democracy protest.

We can naturally see that segments 1-3 all discuss the "protesters vandalized the Legislative Chamber" episode, while segment 4 discusses the "police dispersed protesters" episode. We can systematically replicate this partitioning process by considering the discriminative co-occurrence score between all pairwise combinations of terms from segments $(i - 1)$ and (i). If the average discriminative co-occurrence *and* static semantic similarity between each term a from $(i - 1)$ and b from (i) is statistically significant ($\geq \mu_d - \sigma_d$) for that specific article d (e.g., notably (slogans, defaced) for segments 1-3), we hypothesize that the **same episode** is being discussed and *merge* them into one episode fragment. If not (e.g., (slogans, arrested) for segments 3-4), this indicates that a **different episode** is being discussed, and we *partition* them into two episode fragments. Further implementation details are provided in Appendix G.

4.3 LLM-Enhanced Episode Estimation

LLMs demonstrate strong event-specific reasoning at the phrase or sentence level (Pai et al., 2024; Gao et al., 2024), but they struggle with understanding long contexts (Li et al., 2024; Liu et al., 2024). This limitation hampers their ability to process all episode fragments for detecting episodes. Additionally, noisy retrieval significantly affects reasoning performance (Shen et al., 2024). To address these challenges, we propose a synergistic approach that enhances in-context episode reasoning by reducing the number of required fragments while improving their cohesiveness and quality. We first identify the set of articles that maximizes the *quantity and quality* of potential episodes, where each article is ranked by multiplying two metrics:

1. *Quality of episode fragments*: A top article should primarily consist of episode fragments containing salient terms that discriminatively co-occur. This reduces the rank of general fragments which summarize/analyze the event. We average each episode fragment's mean inner-discriminative co-occurrence (across all pairwise combinations of its salient terms).
2. *Quantity of episode fragments*: A top article should ideally contain all ground-truth episodes. Therefore, we take the $\log$ of the number of episode fragments in the article.

After ranking all articles, we select the top $\delta\%$ and resolve potential co-references to the same episode across these top articles. We apply agglomerative clustering (Murtagh and Contreras, 2012) to the top episode fragments using a pre-computed distance matrix. The distance between two fragments (inversed) is calculated using the same discriminative and static semantic similarity score used in Section 4.2). Clusters with a statistically insignificant number of episode fragments are pruned.

Finally, we provide episode fragment clusters as a more interpretable context for the LLM to resolve two challenges: (1) missing time and location stamps in fragments, and (2) semantic inconsistencies within clusters. The LLM summarizes each cluster by identifying its core attributes— entities, actions, objects, location, and time. It then outputs the *episode attributes*, *relevant keywords* for extraction, and the top *extracted text segments* (prompt & example in Appendix H).

4.4 Episode-Segment Classification

With these core summaries of the episode clusters, we obtain a generalized description of each candidate episode. For each candidate, we encode its LLM-based core attributes and extracted segments to compute a simple episode representation. Specifically, following extremely weakly supervised text classification works (Wang et al., 2021; Kargupta et al., 2023), we take the harmonic mean of these representations—as the latter extracted segments are likely not as significant as the earlier extractions and core attributes. We similarly encode all input *segments* with the same encoder. We use these to assign an episode and confidence score to each encoded input segment.

Episode-Segment Confidence Estimation. Directly mapping a text segment to its top episode based on cosine similarity risks misclassifying episode-irrelevant segments or those discussing multiple episodes (e.g., a journalist's summary). To avoid classifying such segments and ensure non-overlapping episode clusters (as discussed in Section 3.1), we must determine the confidence of a segment discussing a single episode.

We compute segment s_i's cosine similarity to its top two episodes (e_i^0 and e_i^1). A larger gap ($e_i^0 - e_i^1$) reflects greater confidence in classifying s_i to e_i^0. Each gap is normalized by the sum of all segment-episode gaps across the corpus, ensuring confidence is relative to the key event:

$$s_{i,\text{confidence}} = \frac{e_i^0 - e_i^1}{\sum_{l=1}^{|\mathcal{S}|}(e_l^0 - e_l^1)} \tag{2}$$

Segments with statistically significant confidence in their top episode are assigned to their respective episode clusters E_i. Episodes with no assigned segments are pruned, yielding the **final detected episodes and clusters,** $\mathcal{E}$.

5 Experiments

For implementing **EpiMine**, we use the following hyperparameters across all datasets: $\delta = 25\%$, $sim\_thresh = 0.75$. We also use Claude-2.1 as our base LLM (A\). All other hyperparameters are set to their respective default values. We provide all experimental settings in Appendix A.

Table 1: Statistics of our collected datasets. The numbers are averaged per key event.

Theme	# docs	# episodes	# segments
Terrorism/Attacks	32.2	5.9	290.3
Natural Disasters	36.2	7.4	324.6
Political Events	70.2	7.5	667.7

5.1 Datasets

We conduct our experiments on three novel thematic, real-world news corpora selected from Wikipedia[2] over the last decade. For each theme, we manually collect approximately 10 key events

[2] https://en.wikipedia.org/wiki/

Table 2: Results averaged across each theme, including the mean # of episodes that EpiMine identifies per theme (in parenthesis). Results are computed on each key event corpus using the top-5 documents for each detected episode. Due to variance in LLM generation, we run it 10 times and report the average of each measure. We scale each value by 100. Bold values denote the top method; second-best method is underlined.

	Terrorism (5.36 eps)			**Natural Disasters** (7.4 eps)			**Politics** (7.5 eps)		
Methods	*5-prec*	*5-recall*	*5-F1*	*5-prec*	*5-recall*	*5-F1*	*5-prec*	*5-recall*	*5-F1*
EMiner	8.64	0.25	0.48	10.37	0.19	0.37	8.66	0.16	0.32
K-means	21.23	21.23	21.23	27.85	**28.47**	28.14	16.04	16.04	16.04
K-means + A\	28.58	14.04	18.26	37.40	16.58	22.00	27.18	17.36	18.25
EvMine	23.03	15.02	17.45	28.15	8.02	12.25	5.36	4.00	4.58
EvMine + A\	37.88	15.70	21.33	43.56	13.22	19.40	32.73	12.98	17.28
EpiMine (A\)	**71.21**	22.07	32.43	**70.98**	28.46	**34.53**	**62.67**	21.54	**29.23**
- No Confidence (A\)	61.97	**30.19**	**38.45**	43.66	20.78	27.76	60.29	**27.73**	24.77
- No LLM	37.73	21.62	24.77	37.19	14.78	17.52	30.64	23.51	19.06

composed of multiple articles and ensure that distinct *episodes* exist in each:

- **Terrorism and attacks:** *2021 Atlanta spa shootings; 2014 Montgomery County Shootings; 2021 Indianapolis FedEx shooting; 2022 Cincinnati FBI field office attack; 2019 Jersey City shooting; 2019 Naval Air Station Pensacola shooting; 2022 Greenwood Park Mall shooting, 2018 Capital Gazette shooting; 2021 Collierville Kroger shooting; 2019 Kyoto Animation arson attack*
- **Natural disasters:** *2023 Tornado outbreak sequence; 2023 Hawaii Wildfires; 2021 Western Kentucky tornado; 2017 Mocoa landslide; 2010 Haiti earthquake; 2021 Henan floods; 2019 Nyonoksa radiation accident; 2022 NA winter storm; 2011 Fukushima nuclear accident*
- **Political events:** *2020 Kyrgyz Revolution, 2019 Storming of the Hong Kong Legislative Council Complex; 2019 Siege of the Hong Kong Polytechnic University; 2017 Zimbabwean coup; 2018 Italian government formation; 2021 January 6 U.S. Capitol attack; 2018 Thai Cave Rescue Operation; 2018 Armenian Revolution; 2017 Lebanon–Saudi Arabia dispute; 2013 Tunisian political crisis*

The articles are obtained from the Wikipage references of each key event— filtered with constraints in time, language, and relevance. Furthermore, each article is segmented to match our setting (Appendix F), and each segment is automatically annotated (Appendix I) with either its corresponding episode ID or with a multiple/no episode tag ('M' or 'X'). Further details on the criteria/process, each theme, and corresponding key events are in Appendices D and I. We also conduct a human-automatic agreement analysis for segment-episode annotation, which shows substantial agreement with good reliability (Appendix D).

5.2 Baselines

We compare against the following methods using the evaluation metrics specified in Appendix C: **(1) K-means** (Likas et al., 2003): given the # of ground-truth episodes, it clusters segments using ST (Reimers and Gurevych, 2019) embeddings; **(2) EvMine** (Zhang et al., 2022): a document-level unsupervised key event detection method adapted to segment level for episode detection; **(3) EMiner** (Jiao et al., 2023): unsupervised event chain miner that clusters atomic actions, adapted to episodes; **(4) No Confidence**: an ablation that uses max cosine-similarity instead of confidence from Equation 2; **(5) No LLM**: an ablation that uses estimated episode clusters from Section 4.3 to compute our episode representations directly. We also integrate A\ into K-means and EvMine using our same prompt (Appendix H). All baseline and ablation details are in Appendix B.

5.3 Overall Results & Analysis

In Table 2, EpiMine shows an average **80.8%** increase in 5-precision, a **34.0%** increase in 5-recall, and a **62.8%** increase in 5-F1 over all baselines. Notably, despite both K-means and K-means + A\ being *given the ground-truth number of episodes*, they are **significantly outperformed by EpiMine** (both the base model and no confidence ablation). Additionally, EvMine and EMiner, originally designed for key event and atomic action levels of event granularity, **fail to address the unique challenges of episode detection**. We further analyze our results through extensive quantitative and qualitative studies, including a detailed case study on the "2019 Hong Kong Legislative Protest" (as shown in Figure 1), leading to the following takeaways:

1. LLMs require effective episode fragment clusters for synergistic episode estimation. As

Table 3: Ablation studies conducted on top 25% of article episode clusters (Section 4.3).

	Terrorism			Natural Disasters			Politics		
Ablations	*5-prec*	*5-recall*	*5-F1*	*5-prec*	*5-recall*	*5-F1*	*5-prec*	*5-recall*	*5-F1*
EpiMine-Top	**0.2292**	**0.2435**	**0.2144**	**0.3817**	**0.2232**	**0.2450**	0.1051†	**0.2233**	0.1201†
TF-IDF	0.0985	0.1403	0.1059	0.3284†	0.1919†	0.2221†	0.0907	0.1908	0.0916
No DC	0.1968†	0.1752†	0.1707†	0.2520	0.1546	0.1785	**0.1126**	0.2108†	**0.1299**

Table 4: Compares top-5 salient terms which (1) have the highest cosine-sim (CS) and (2) discriminative co-occurrence (DC), with the given keyword.

Keyword	CS	DC
broke	stormed, ransacked, dashed, occupied, rushed	glass, doors, metal, building, teargas
slogans	spray, placards, painted, defaced, pictures	reads, wall, damage, started, portraits, spray

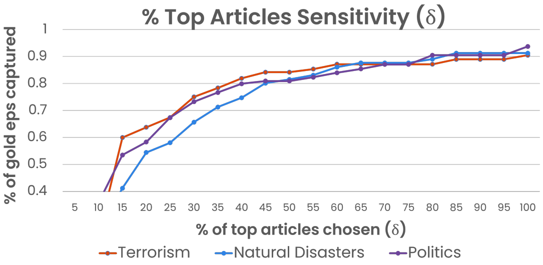

Figure 4: Percentage of key event's gold episodes captured in the $\delta\%$ top articles chosen during the candidate episode estimation. Results averaged across themes.

shown in Table 5, LLMs without any initial clusters as guidance (A\ , GPT-4[3]) fail to detect high-quality episodes, miss most ground-truth episodes, and include irrelevant atomic actions (e.g., "Brian Leung pulls off mask"). Similarly, using low-quality baseline clusters results in poor performance. EvMine detects episodes that all reflect the same event, "Protesters vandalized the Legislative Chamber". While K-means produces more distinct episodes, it does not capture the most critical, gold episodes. In contrast, EpiMine's episodes are both distinct and meaningful, attributed to its cluster quality. This is quantitatively confirmed by EpiMine-No LLM's competitive performance: using only EpiMine's episode fragment clusters to compute episode representations—without any LLM summarization—still yields significantly better performance than all baselines (without or without LLM integration) on the Terrorism and Politics datasets, and remains highly competitive on Natural Disasters. This indicates the high quality of our fragment ranking and clustering.

EpiMine's clusters also elicit the LLM to identify more meaningful temporal information. Unlike most baseline episodes which have "July 1, 2019" as the time attribute, EpiMine's episodes feature more descriptive temporal cues: "after breaking in", "after midnight", "in a news conference at 4 am on July 2". Moreover, EpiMine's "incorrect" episode #4 is a significant sub-event of the key event discussed by many articles. This *strongly demonstrates the impact of EpiMine's candidate episode clusters* as input into the LLM; ***LLMs alone cannot perform quality episode detection***.

2. The strengths of discriminative co-occurrence complement those of cosine similarity. Table 4 illustrates the qualitative strengths of our novel discriminative co-occurrence metric. Both cosine similarity (CS) and discriminative co-occurrence (DC) offer different, complementary strengths. CS identifies similar words that play a *similar role or are synonyms* within an episode (e.g., "broke", "ransacked"), while DC identifies the *key surrounding actions and objects* that co-occur within the same episode (e.g., "slogans", "wall"). This is quantitatively supported by ***TF-IDF*** and ***No DC*** in Table 3, which show a significant decrease in the quality of our top episode clusters without our salience and discriminative co-occurrence measures. We note that the politics dataset does show slight improvements in precision and F1 when discriminative co-occurrence is replaced, likely due to more term overlap across episodes (less distinct, sequential episodes).

3. Fragment ranking identifies top articles. In Figure 4, we conduct a sensitivity analysis of the top articles chosen to estimate candidate episodes (Section 4.3). We compare the gold episodes contained in the set of the top $\delta\%$ articles as we vary δ. By ranking the articles based on their likelihood of containing both high-quality and numerous episodes, we find that *EpiMine's top article selection covers the vast majority of episodes*

[3] https://chat.openai.com/

Table 5: Gold and detected episodes (a maximum of five are included for brevity) for the "2019 Hong Kong Legislative Protests" key event. We specify the gold/detected episode attributes for each episode cluster in the following semicolon-separated format: core entity; action; object; time; location. "Not detected" denotes that no more episodes were generated by the model. We note the number of detected episodes beside the model name. We also color-code attributes which clearly align to a specific episode.

Model	Episode #1	Episode #2	Episode #3	Episode #4	Episode #5
Gold (5 eps)	Activists; headed; towards the Legislative Council Complex; 1 July 2019; Hong Kong	Protesters; stormed; the Legislative Council Complex; around 9:00 pm; Hong Kong;	Protesters; damaged/defaced; portraits, furniture, emblem, etc.; 1 July 2019; Legislative Council Complex	Police; started using; tear gas to disperse protesters; 12:05 am 2 July; around the Legislative Council complex	Police; arrested; individuals in connection with the incident; between 3 July and 5 July; Hong Kong
K-means + A (4 eps)	Protesters; storm and vandalize; Legislative Council building; July 1, 2019; Legislative Council complex in Admiralty, Hong Kong	Hong Kong government; condemns; protesters storming legislative building; July 1, 2019; Hong Kong	Hong Kong protesters; express; demands for freedom and democracy; July 1, 2019; Hong Kong Legislative Council	Hong Kong police; adopt; more restrained tactics; July 1, 2019; Hong Kong Legislative Council	Not detected
EvMine + A (4 eps)	Protesters; vandalize; Hong Kong legislative building; July 1, 2019; Hong Kong legislative building	Protesters; occupy and vandalize; Hong Kong legislative chamber; July 1, 2019; Hong Kong legislative building	Protesters; spray paint; slogans and demands; July 1, 2019; Hong Kong legislative building	Protesters; deface; Hong Kong emblem; July 1, 2019; Hong Kong legislative building	Not detected
Claude (A\\) (3 eps)	Protesters; storm; Hong Kong's Legislative Council; July 1, 2019; Hong Kong's Legislative Council building	Police; retreat and avoid confrontation; protesters storming Hong Kong's Legislative Council; July 1, 2019; Hong Kong's Legislative Council building	Brian Leung Kai-ping; pulls off mask and reads protesters' demands; inside Hong Kong's Legislative Council; July 1, 2019; Legislative Council chamber	Not detected	Not detected
GPT-4 (2 eps)	Hong Kong protesters; storm Legislative Council; government and police; July 1, 2019; Legislative Council Complex, Hong Kong	Hong Kong citizens; march against extradition bill; "Carrie Lam and Chinese government; June 2019; Various locations in Hong Kong	Not detected	Not detected	
EpiMine w/ A (7 eps)	Protesters; broke into and occupied; Hong Kong's legislative building; July 1, 2019; Hong Kong	Protesters; vandalized; the legislative building; after breaking in; Hong Kong	Police; fired tear gas at; protesters; after midnight on July 1; outside the legislative building	Carrie Lam; condemned; the protesters' actions; in a news conference at 4am on July 2; Hong Kong	Police; began making arrests of; protesters involved; in the days after; Hong Kong

by $\delta = 25\%$ and more comprehensively around $\delta = 45\%$. This is significant as it helps *minimize both the noise and the amount of data* needed to accurately detect all episodes. This is further supported by the EpiMine-Top ablation (Table 3), which quantitatively shows that the top 25% of the ranked fragments alone have competitive or higher performance than the baselines.

The Role of Confidence. The confidence metric influences EpiMine toward **more conservative** episode-segment classification by pruning segments with statistically insignificant confidence scores ($\leq \mu - \sigma$), which are unlikely to map to a single episode. Reducing this threshold increases the number of mapped segments, potentially improving recall by converting false negatives into true positives, but at the cost of reduced precision due to an increase in false positives. This trade-off is quantitatively demonstrated in the "No-Confidence" ablation (Table 2), where omitting confidence leads to occasional gains in recall accompanied by declines in precision. Nevertheless, both configurations—with and without confidence—are significantly better than all baselines, allowing users to determine if including confidence aligns with their use case.

6 Conclusion

In this work, we proposed **EpiMine**, a novel, unsupervised episode detection method for large-scale news events. EpiMine performs (1) episode indicative term mining— identifying combinations of salient terms that are likely to discriminatively co-occur *within* an episode and not *across* episodes, (2) episode partitioning, which partitions each article into approximate isolated episodes, (3) LLM-enhanced episode estimation, which clusters the top partitions into candidate episodes and synergizes with an LLM to produce fluent and meaningful episodes, and (4) episode-segment classification, which maps confident segments to their respective episode clusters. EpiMine significantly outperforms all baselines on the vast majority of key events, as shown through extensive quantitative and qualitative analysis.

7 Limitations & Future Work

While EpiMine serves as an intuitive, unsupervised framework which demonstrates a more interpretable granularity for event analysis (episodes), it contains a few limitations that form the foundation for future, impactful research areas.

We note that the key event theme has an impact on EpiMine's performance. Specifically, natural disaster episodes are typically sequential and semantically distinct: disaster begins → warning → evacuation → damage/deaths → relief. As K-means is uniquely given k, the number of episodes, and relies on semantic similarity, it performs well with distinct episodes. However, we still see that its reliance on surface-level semantics leads to lower precision. Additionally, in the ablation studies shown in Table 3, the politics dataset does show slight improvements in precision and F1 when *discriminative co-occurrence is replaced*, due to more term overlap across episodes, resulting in less distinct, sequential episodes.

Further work towards the temporal analysis of episodes within articles can be explored, as well as extending our work to primarily multilingual news settings with low resources.

8 Ethics Statement

Based on our current methodology and results, we do not expect any significant ethical concerns, given that subtasks like episode detection within the news event extraction and analysis is a standard problem domain across data mining applications. Furthermore, having the method rely on zero supervision helps as a barrier to any user-inputted biases. However, one minor factor to take into account is any hidden biases that exist within the large language models used as a result of any potentially biased data that they were trained on. We used these pre-trained language models for refining the fluency of the detected episode clusters and did not observe any concerning results, as it is a low-risk consideration for the domains that we studied.

9 Acknowledgements

This work was supported by the National Science Foundation Graduate Research Fellowship. This research used the DeltaAI advanced computing and data resource, which is supported by the National Science Foundation (award OAC 2320345) and the State of Illinois. DeltaAI is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

References

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2021. GATE: graph attention transformer encoder for cross-lingual relation and event extraction. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12462–12470. AAAI Press.

Christopher Baldassano, Janice Chen, Asieh Zadbood, Jonathan W Pillow, Uri Hasson, and Kenneth A Norman. 2017. Discovering event structure in continuous narrative perception and memory. *Neuron*, 95(3):709–721.

Muhao Chen, Hongming Zhang, Haoyu Wang, and Dan Roth. 2020. What are you trying to do? semantic typing of event processes. In *Proceedings of the 24th Conference on Computational Natural Language Learning, CoNLL 2020, Online, November 19-20, 2020*, pages 531–542. Association for Computational Linguistics.

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

Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 671–683. Association for Computational Linguistics.

Jun Gao, Huan Zhao, Wei Wang, Changlong Yu, and Ruifeng Xu. 2024. Eventrl: Enhancing event extraction with outcome supervision for large language models. *arXiv preprint arXiv:2402.11430*.

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

Rujun Han, I-Hung Hsu, Mu Yang, Aram Galstyan, Ralph M. Weischedel, and Nanyun Peng. 2019. Deep structured neural network for event temporal relation extraction. In *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*, pages 666–106. Association for Computational Linguistics.

Yizhu Jiao, Sha Li, Yiqing Xie, Ming Zhong, Heng Ji, and Jiawei Han. 2022. Open-vocabulary argument role prediction for event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5404–5418. Association for Computational Linguistics.

Yizhu Jiao, Ming Zhong, Jiaming Shen, Yunyi Zhang, Chao Zhang, and Jiawei Han. 2023. Unsupervised event chain mining from multiple documents. In *Proceedings of the ACM Web Conference 2023*, pages 1948–1959.

Disha Jindal, Daniel Deutsch, and Dan Roth. 2020. Is killed more significant than fled? A contextual model for salient event detection. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 114–124. International Committee on Computational Linguistics.

Priyanka Kargupta, Tanay Komarlu, Susik Yoon, Xuan Wang, and Jiawei Han. 2023. MEGClass: Extremely weakly supervised text classification via mutually-enhancing text granularities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10543–10558, Singapore. Association for Computational Linguistics.

Sangeet S Khemlani, Anthony M Harrison, and J Gregory Trafton. 2015. Episodes, events, and models. *Frontiers in human neuroscience*, 9:590.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2024. LooGLE: Can long-context language models understand long contexts? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16304–16333, Bangkok, Thailand. Association for Computational Linguistics.

Manling Li, Tengfei Ma, Mo Yu, Lingfei Wu, Tian Gao, Heng Ji, and Kathleen McKeown. 2021a. 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*, pages 6443–6456, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare R. Voss. 2020. Connecting the dots: Event graph schema induction with path language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 684–695. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021b. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 894–908. Association for Computational Linguistics.

Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. 2003. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018a. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Zheng Liu, Yu Zhang, Yimeng Li, and Chaomurilige. 2023. Key news event detection and event context using graphic convolution, clustering, and summarizing methods. *Applied Sciences*, 13(9):5510.

Zhengzhong Liu, Chenyan Xiong, Teruko Mitamura, and Eduard H. Hovy. 2018b. Automatic event salience identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1226–1236. Association for Computational Linguistics.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Sebastian Martschat and Katja Markert. 2018. A temporally sensitive submodularity framework for timeline summarization. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 230–240, Brussels, Belgium. Association for Computational Linguistics.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Zihan Wang, Chao Zhang, Yu Zhang, and Jiawei Han. 2020. Discriminative topic mining via category-name guided text embedding. In *Proceedings of The Web Conference 2020*, pages 2121–2132.

Fionn Murtagh and Pedro Contreras. 2012. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97.

Kiem-Hieu Nguyen, Xavier Tannier, and Véronique Moriceau. 2014. Ranking multidocument event descriptions for building thematic timelines. In *COLING 2014, the 25th International Conference on Computational Linguistic*, pages 1208–1217.

Liu Pai, Wenyang Gao, Wenjie Dong, Lin Ai, Ziwei Gong, Songfang Huang, Li Zongsheng, Ehsan Hoque, Julia Hirschberg, and Yue Zhang. 2024. A survey on open information extraction from rule-based model to large language model. *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9586–9608.

Zheng Qi, Elior Sulem, Haoyu Wang, Xiaodong Yu, and Dan Roth. 2022. Capturing the content of a document through complex event identification. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 331–340, Seattle, Washington. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Xiaoyu Shen, Rexhina Blloshmi, Dawei Zhu, Jiahuan Pei, and Wei Zhang. 2024. Assessing "implicit" retrieval robustness of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8988–9003, Miami, Florida, USA. Association for Computational Linguistics.

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

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. Joint constrained learning for event-event relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 696–706. Association for Computational Linguistics.

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2021. X-class: Text classification with extremely weak supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3043–3053, Online. Association for Computational Linguistics.

Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Nate Chambers. 2018. Hierarchical quantized representations for script generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3783–3792. Association for Computational Linguistics.

David Wilmot and Frank Keller. 2021. Memory and knowledge augmented language models for inferring salience in long-form stories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 851–865. Association for Computational Linguistics.

Susik Yoon, Dongha Lee, Yunyi Zhang, and Jiawei Han. 2023. Unsupervised story discovery from continuous news streams via scalable thematic embedding. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 802–811.

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 the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4091–4104.

Hongming Zhang, Muhao Chen, Haoyu Wang, Yangqiu Song, and Dan Roth. 2020. Analogous process structure induction for sub-event sequence prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1541–1550. Association for Computational Linguistics.

Yunyi Zhang, Fang Guo, Jiaming Shen, and Jiawei Han. 2022. Unsupervised key event detection from massive text corpora. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 2535–2544, New York, NY, USA. Association for Computing Machinery.

A Experimental Settings

For implementing **EpiMine**, we use the following hyperparameters across all datasets: $\delta = 25\%$, $sim\_thresh = 0.75$. All other hyperparameters are set to their respective default values. We provide all experimental settings in Appendix A.To determine statistical significance, we check for $\geq \mu - \sigma$. For our word representations, we use `bert-base-uncased`. For our sentence representations, we use `all-mpnet-base-v2`. We choose to use `Claude-2.1`[4] for fluent candidate episode estimation due to its strong structured JSON/XML input and output formatting abilities. However, this proprietary model can be replaced with any open-source model as EpiMine is model-agnostic. We use only one NVIDIA GeForce GTX 1080 for all

[4] claude.ai/

experiments; for non-API models, we utilize two NVIDIA-RTX A6000s.

B Baselines

We compare against the following methods using the evaluation metrics specified in Appendix C.

- **K-means** (Likas et al., 2003): No. of ground-truth episodes is given; clusters segments based on semantic similarity of ST (Reimers and Gurevych, 2019) embeddings.
- **EvMine** (Zhang et al., 2022): Unsupervised framework for key event detection that leverages peak phrases and detects communities using event-indicative features. We extend the original document-level method to the segment level for episode detection.
- **EMiner** (Jiao et al., 2023): Unsupervised event chain mining that performs atomic action clustering. For episode detection, we map its final output, a list of events, back to the original sentences from which each event was extracted, treating these sentences as segments. To retrieve more episode-associated segments, we use ST (Reimers and Gurevych, 2019) to select the k most similar segments to each cluster sentence.

We also include the following full and partial ablations of EpiMine (clusters segments from all articles vs. top $\delta\%$ articles, respectively):

- **No Confidence**: A full ablation, where all input segments are classified based on the episode with max cosine similarity instead of using the confidence score from Equation 2.
- **No LLM**: We take the estimated episode clusters from Section 4.3 that normally would have been inputted into the LLM, and instead use them to compute our episode representations directly. These representations are used for our classification step (Section 4.4), run on the full dataset with confidence.
- **EpiMine-Top**: A partial ablation which directly outputs the intermediate episode clusters formed based on the top articles identified in Section 4.3 without inputting them into the LLM-based episode estimation step.
- **TF-IDF**: A partial ablation which replaces the salience and synonym expansion step (Section 4.1) with TF-IDF.
- **No DC**: A partial ablation which replaces the discriminative co-occurrence score (Equation 1) with raw pair frequency.

C Evaluation Metrics

Following a recent work on key event detection (Zhang et al., 2022), we adapt the k-prec, k-recall, and k-F1 metrics to quantitatively evaluate episode detection performance— specifically, how the model's top-k segments within each detected episode align with the ground truth episodes.

Formally, suppose there are N ground truth episodes $\mathcal{G} = \{G_1, G_2, \ldots, G_N\}$, each of which is a set of text segments related to its corresponding episode. $\mathcal{E} = \{E_1, E_2, \ldots, E_K\}$ are the model predicted episodes, each of which is a ranked list of segments, and $E_{j,k}$ means the top-k segments within E_j. Then, the k-metrics are defined as follows:

$$\textbf{k-prec} = \frac{\sum_{G_i \in \mathcal{G}} \mathbb{1}(\exists E_j \in \mathcal{E}, E_{j,k} \cap G_i \geq \frac{k}{2})}{\sum_{E_j \in \mathcal{E}} \mathbb{1}(|E_j| \geq k)}$$

$$\textbf{k-recall} = \frac{\sum_{G_i \in \mathcal{G}} \mathbb{1}(\exists E_j \in \mathcal{E}, E_{j,k} \cap G_i \geq \frac{k}{2})}{N}$$

$$\textbf{k-F1} = \frac{2 \cdot \text{k-prec} \cdot \text{k-recall}}{\text{k-prec} + \text{k-recall}}$$

D Key Event Corpus Dataset Construction & Annotation

Given that our task is novel and no large-scale key event-specific news corpus is available for this task where the key events are guaranteed to contain distinguishable episodes, we briefly discuss how we collect the input corpus from online news data. Given our set of key events (as listed in Section 5.1), we first scrape the external reference list from their corresponding Wikipedia page and select the news articles that have been published within two months of given key event's start date (e.g., all articles selected for "January 6 2021 Capitol Attack" would have been published between November 6-March 6). This is important as we want to prioritize the news articles which focus on describing the episodes of the key event and their corresponding aspects as opposed to primarily opinions or analyses. This allows us to motivate our task as one critical for currently evolving key events which required a more fine-grained episodic timeline. Furthermore, it is consequently *unlikely* for a single article to cover *all the episodes* and *exclusively* episodes under a key event. Despite this being more challenging, it is acceptable as the goal of our task is to extract only the key event-related episodes, which must be substantiated by multiple documents in either case.

During the collection process, we targeted selecting a diverse set of key events topics within a theme. For instance, we attempted to cover every type of "natural disaster", including tornados, wildfires, and etc. When selecting key events, we leave out those with less than 20 hyperlinks in the Wikipage and manually inspect at least 20 articles per event in order to ensure quality. Table 1 summarizes the statistics for these datasets. We also construct a background news corpus of approximately 4,000 long news articles using the New York Times corpus for topic categorization (Meng et al., 2020).

For the annotation process, we had each of the four individual annotators (computer science graduate students) manually identify a ground-truth description for each of the episodes under every key event. The descriptions and a one-shot annotation demonstration (assigning either an episode ID per segment, or 'M'/'X' if it describes multiple or none) are provided in our automatic annotation prompt (Appendix I).

The same annotators each manually annotated a subset of segments (25 articles, 300 segments in total) with either their corresponding episode ID or '-1' if the segment was either 'X' and 'M'. The human-automatic annotation agreement is shown in Table 6 with three versions of intra-class correlation (ICC) and Cohen's κ). The Cohen's κ indicates substantial agreement, and the all three versions of ICC indicate good reliability.

	Cohen's κ	**ICC1k**	**ICC2k**	**ICC3k**
Score	0.614	0.772	0.772	0.773

Table 6: Agreement scores between the human and LLM annotation of each article segment.

E Identifying Salient Terms for Episode Detection

We define the salience score of a term w_i within segment s as the following function, where $freq(w_i)$ is the number of key event segments that w_i is contained in, N_{bg} is the number of news articles in the background corpus we construct (using general New York Times articles), and $bgf(w_i)$ is the number of background articles that w_i is present in.

$$\textbf{Salience}(w_i) = \left(1 + \log^2\left(freq(w_i)\right)\right) \times \log\left(\frac{N_{bg}}{bgf(w_i)}\right) \tag{3}$$

Stop words and infrequent terms ($freq(w_i) < 5$) are assigned a salience score of -1. A key event's set of salient terms T is comprised of the terms with a salience score *above the mean salience* across the entire vocabulary. In the case of infrequent synonyms used by a journalist as a stylistic choice (e.g., "demonstrations", "marches"), we expand T with terms that are similar (cosine-similarity) to their static word representations (average of its contextualized word embeddings across entire key event corpus).

F Key Event News Article Pre-Processing

Given that the expected output for the episode detection task is a *cluster of text segments*, we first must segment each key event news article. We would like to ideally preserve both the primary aspects (e.g., core entities and their actions) and peripheral aspects (e.g., reactions to a core entity's action) relevant to that episode, which may be helpful for cross-document episode co-reference resolution. In order to do this, we utilize the text segmentation method, C99 (Choi, 2000). Furthermore, in order to assist with the cohesiveness of the segment, we employ entity co-reference resolution before performing segmentation, which assists with retaining the context across text segments ("They surrounded the legislative building [...]" → "The protesters surrounded the legislative building [...]"). Our core methodology is given these text segments (in their raw form, without co-references resolved) and their source articles as the primary inputs.

G Additional Details for Episode Partitioning

We note that for determining semantic similarity between the terms of two segments, we use both (1) the average cosine similarity between all unordered pairs of terms between segment (i-1) and (i), and (2) the cosine similarity between the average of static term representations in $(i-1)$ and the average of static term representations in (i). Furthermore, we filter out any non-salient segments before episode partitioning to avoid any influence of noisy segments (e.g., journalist's analysis, summary statements, historical comparisons, and other generic noise) on the quality of our episode fragments.

Finally, following (Wang et al., 2021; Kargupta et al., 2023), we take the harmonic mean of all pairwise discriminative co-occurrence scores instead of a simple average. This allows us to prioritize the more salient *and* discriminative terms when determining the episode partitions. For instance, if "protesters" consistently occurs throughout the majority of episodes and thus has a low average discriminative co-occurrence, then it is not as informative for episode partitioning.

H Claude-2 Prompt & Example for Candidate Episode Estimation

Prompt. We use the following prompt for estimating fluent candidate episodes from our input episode fragment clusters. We denote k as the number of episode fragment clusters outputted after clustering the top article episode fragments in Section 4.3.

```
Task: You are a key news event analyzer that is aiming to detect episodes (a representative subevent that reflects a critical sequence of actions performed by a subject at a certain and/or location) based on text segments from different news articles. Given the above groups of article segments, predict at least 2 and at most {k} potential episodes of the key event. Some groups may fall under the same episode. Output your answer inside the tags <answer></answer> as a JSON object where each item is also a JSON with the key "title" with the value containing the [subject, action, object, time, location] of the episode, a key "keywords" with the string value being a list of 5-10 associated keywords unique to that specific episode, and a final key "example_sentences" with a value being a list of 2-5 extracted sentences from the input segment groups. Feel free to output less than {k} episodes if you feel that any are redundant (could fit under an existing candidate episode). The title, keywords, and example sentences of a predicted episode should not be able to be placed under another different predicted episode.
```

Example. Below, we provide an example of EpiMine's candidate episode estimation step (Section 4.3). Specifically, the LLM identifies the core attributes (subject, action, object, time, location) of each unique cluster, relevant keywords, and top extracted text segments given the input clusters:

```
'title': ['Protesters', 'storm and vandalize', 'Hong Kong's Legislative Council building', 'July 1, 2019', 'Legislative Council building in Admiralty, Hong Kong']
'keywords': ['vandalism, graffiti, violence, escalation, ransacking']
'example_sentences': ['Hundreds of anti-extradition bill protesters finally broke into the legislature after many hours of attacking the public entrance and ransacked the building, including displaying the colonial Hong Kong flag in the chamber.', 'Slogans on the wall read: "Murderous regime", and "There are no rioters only a tyrannical regime."]
```

I Claude-2 Prompt for Dataset Annotation

We automatically annotate our dataset using Claude-2.1 using the prompt below (before an additional human-verification stage):

```
You are a news event analyzer that labels text segments of a news article with their matching event episode description. I will give you several text segments, and several episodes of a key event in tuples. We define an episode as the following: an episode is a set of thematically coherent text segments discussing a particular set of core entities performing actions for or towards an object(s) at a certain time and/or location during a real-world key event. The entities, actions, objects, time, and location can all be considered aspects of an episode.

[one-shot demonstration & format specification]
Please help classify the text segments under different episodes (the output value for each segment should be an
```

integer key of each episode). If you think a text segment cannot be used to describe any episodes, please use "X" in the output to indicate the lack of an episode tuple number for that segment. If a text segment is very general, does not describe the key event at hand, or can be matched to multiple episodes, then please use a "M" in the output to indicate the multiple episode mapping for that segment. There should be a value assigned to each of the len(segments) segments (segment_0, ..., segment_len(segments)-1).