Text is All You Need: LLM-enhanced Incremental Social Event Detection

Zitai Qiu[1], Congbo Ma[1,2], Jia Wu[1], Jian Yang[1]
[1]Macquarie University, [2]New York University Abu Dhabi
zitai.qiu@students.mq.edu.au, jia.wu@mq.edu.au

Abstract

Social event detection (SED) is the task of identifying, categorizing, and tracking events from social data sources such as social media posts, news articles, and online discussions. Existing state-of-the-art (SOTA) SED models predominantly rely on graph neural networks (GNNs), which involve complex graph construction and time-consuming training processes, limiting their practicality in real-world scenarios. In this paper, we rethink the key challenge in SED: the informal expressions and abbreviations of short texts on social media platforms, which impact clustering accuracy. We propose a novel framework, **LLM-enhanced Social Event Detection (LSED)**, which leverages the rich background knowledge of LLMs to address this challenge. Specifically, LSED utilizes LLMs to formalize and disambiguate short texts by completing abbreviations and summarizing informal expressions. Furthermore, we introduce hyperbolic space embeddings, which are more suitable for natural language sentence representations, to enhance clustering performance. Extensive experiments on two challenging real-world datasets demonstrate that LSED outperforms existing SOTA models, achieving improvements in **effectiveness**, **efficiency**, and **stability**. Our work highlights the potential of LLMs in SED and provides a practical solution for real-world applications. The code is available at GitHub[1].

1 Introduction

Social events are typically defined as unique occurrences at a specific time and location in the real world (Peng et al., 2022; Cao et al., 2024). For example, on 11 October 2012, the famous Chinese writer Mo Yan won the 2012 Nobel Prize in Literature. A defining characteristic of social events is their ability to rapidly and widely propagate through discussions on social media platforms. (Liu et al., 2015). With the increasing number of users, social media has become the primary medium for both mainstream media and individuals to publish and disseminate information. Social event detection (SED) aims to identify such events from vast amounts of user-generated content, including posts, tweets, and images (Li et al., 2022; Atefeh and Khreich, 2015). Detecting social events accurately is crucial for various real-world applications. First, it facilitates tasks such as disaster monitoring, sentiment analysis, public opinion tracking, and market regulation (Peng et al., 2021; Gaspar et al., 2016; Nisar and Yeung, 2018; Marozzo and Bessi, 2018). Second, by structuring messages into event-based formats, SED improves information organization and monitoring, enabling more effective event analysis (Wang et al., 2017; Allan, 2002).

Compared to traditional event detection tasks, SED presents additional challenges due to the brevity and informativeness of user-generated content. While such content conveys crucial information in just a few words, most machine learning techniques struggle to process and interpret it effectively (Song et al., 2014). Despite the efforts made by existing research, an **effective**, **efficient**, and **stable** SED model has not yet been fully implemented.

Specifically, SED models based on text analysis (Bollegala et al., 2018; Ramos, 2003) are **ineffective** due to the informal expression and lack of co-occurrence words of short texts. Furthermore, another factor affecting effectiveness is that most SED models embed text representations into Euclidean space, leading to distortion. Social messages, as natural language sentences, inherently exhibit a hierarchical structure (Dhingra et al., 2018), as shown in Figure 1. However, Euclidean space struggles to capture hierarchical structures effectively, while directly embedding such structures into Euclidean space can introduce embedding distortions (Ganea et al., 2018; Chami et al., 2019).

[1]https://github.com/ZITAIQIU/LSED

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4666–4680

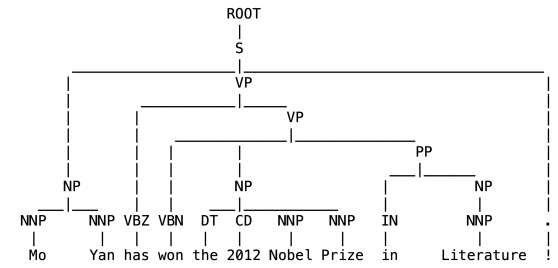


Figure 1: The constituency parse tree of a sentence. Phrases in a sentence can be arranged into a component-based parse tree, but this hierarchy is usually not annotated in text corpora.

Furthermore, SED models leveraging heterogeneous graph neural networks (Qiu et al., 2025; Cao et al., 2021, 2024; Peng et al., 2019, 2022; Yu et al., 2023; Ren et al., 2022; Li et al., 2024) can integrate multi-perspective information from social media. While this approach effectively alleviates the sparsity of co-occurring words, the complexity of graph construction and processing imposes **challenges to efficiency**.

In addition, the spread of social events is driven by user discussions on social platforms (Liu et al., 2015), distinguishing them from traditional news by the dynamic frequency and volume of events-related messages. The duration of these discussions varies based on user interest, with some lasting for days, while others dissipate within hours or even minutes. This dynamic nature of social data presents a challenge to the **stability** of existing SED models based on graph structure.

To address the above challenges, we propose LSED, a novel framework that leverages large language models (LLMs) to enhance Social Event Detection (SED) by addressing key challenges in social message processing. First, LSED leverages LLMs to expand abbreviations and standardize textual expressions in short messages, mitigating information sparsity. Next, LSED embeds text representations into hyperbolic space to better capture hierarchical structures and reduce embedding distortion, enhancing model **effectiveness**. Finally, LSED employs clustering techniques to enhance common feature representations among social messages, thereby facilitating event detection. In particular, unlike existing SED models that rely on graph structures, LSED operates independently of these dependencies, leading to improved **efficiency** and **stability**. In a nutshell, our major contributions are as follows:

- To our knowledge, LSED is the first framework to leverage LLMs for the SED task, effectively and simply addressing the challenges posed by short text in SED.
- LSED does not rely on graph structure information in social message blocks, showing a new perspective on event detection from social media without additional information from graph structures. This design enhances LSED's stability and practicality compared to SED models based on graph structure in a real-world scenario.
- Experimental results demonstrate that the hierarchical structure of natural language sentences impacts feature embedding. Using a more suitable low-dimensional space to capture these hierarchical structures (like hyperbolic space) can better reduce the distortion during embedding and improve the performance of the framework.
- Through extensive experiments on two real-world social event datasets, LSED outperforms baseline models. Further analysis, supported by ablation studies of its key components, validates the overall design choices of LSED.

2 Related Work

2.1 Social Event Detection (SED)

Early event detection models, such as LDA (Bollegala et al., 2018) and TF-IDF (Ramos, 2003), are based on text analysis and are primarily designed for traditional news manuscripts. Their approach of determining topics through word frequency analysis proves ineffective on social platforms, where short texts dominate.

Currently, the primary direction of the SED methods is based on graph neural networks (GNNs) or graph structure information between social messages. These methods combine rich semantic information and structural information on social networks to address the challenges of informal expression in social messages. For supervised models, like KPGNN (Cao et al., 2021) constructs event message graphs by incorporating users, keywords, and entity attributes; FinEvent (Peng et al., 2022) and Re-DHAN (Yu et al., 2023) combine reinforcement learning with GNN to help the model select the optimal aggregation threshold through

reinforcement learning; DAMe (Yu et al., 2024) combines federated learning with GNN to allow the model to learn more features; GraphHAM (Qiu et al., 2024) better embeds social messages by automatically selecting meta-paths; LTS (Ma et al., 2024) and DWMM (Ma et al., 2025) further optimize the efficiency and effectiveness of meta-path selection. For unsupervised models, like HISEvent (Cao et al., 2024) uses structural entropy to improve the relevance between messages and better enhance the effectiveness of SED; HyperSED (Yu et al., 2025) models social messages as semantic-based message anchors to further improve the efficiency of unsupervised SED.

However, constructing these graph structures is complex and time-consuming, requiring significant computational training resources. Moreover, due to the dynamic social messages, the models' performance relies on the graph structure, which is unstable. Therefore, we refocus on the core challenge of SED (concise texts and informal expressions), highlighting that effectively solving the short-text problem and maintain the stability of the framework.

2.2 Short Text Classification

With the development of mobile devices and social platforms, short text has become ubiquitous in various online interactions, including instant messaging, chat logs, news comments, and microblogging services like Twitter. However, short texts present challenges for classification due to their brevity, limited contextual information, and high ambiguity. Existing short-text classification methods can be broadly categorized into two approaches: **single-source** and **multi-source** methods. Single-source methods like RCNN (Lai et al., 2015) employ a recurrent convolutional neural network to capture contextual features essential for classification, while ClassiNet (Bollegala et al., 2018) leverages unlabeled data to construct word co-occurrence graphs and explore feature relationships to address word sparsity.

However, single-source approaches still suffer from data sparsity issues, leading to the development of multi-source methods, which rely on external knowledge. These methods enhance text representation by retrieving information from external sources and using attention mechanisms to determine the significance of different concepts. For example, DE-CNN (Xu and Cai, 2019) integrates contextual knowledge into convolutional neural networks for short text classification, and HGAT (Yang et al., 2021) utilizes a heterogeneous information network to incorporate additional information and relationships from open knowledge bases, helping mitigate semantic sparsity problems. Although these approaches show promising results, they highly depend on large-scale training data to effectively build models, which makes the collection of suitable training instances costly.

2.3 Prompts for Large Language Models (LLMs)

LLMs have made significant progress in various natural language tasks such as dialogue, machine translation, and reasoning (Shi et al., 2024; Wan et al., 2024). Prompts serve as a crucial bridge between users and LLMs, enabling the communication of task descriptions. By guiding LLMs to adapt to diverse downstream tasks, prompts unlock their vast potential. For example, LAMP (Shi et al., 2024) uses prompts to enable LLMs to use their rich historical knowledge and reasoning capabilities to associate the relationship between subjects and predict events; TnT-LLM (Wan et al., 2024) designs a quantifiable and traceable framework to cluster text and generate pseudo-labels using LLMs. Therefore, leveraging prompts to enhance SED with LLMs is desirable, as LLMs have been trained on extensive textual data and expressions that may not be directly accessible to SED models but could still provide valuable insights.

2.4 Hyperbolic Embeddings for Natural Language Processing (NLP)

Most real-world data exhibit hierarchical structures, either explicitly, such as in WordNet, or implicitly, as seen in social networks and natural language sentences (Dhingra et al., 2018). Recent work suggests that hyperbolic space is a promising alternative to standard Euclidean space to represent these hierarchical structures better when learning representations (Ganea et al., 2018; Chami et al., 2019).

However, there are few works exploring the use of hyperbolic space for NLP tasks, and most of them explore the hierarchical structure of graph neural networks combined with NLP tasks (Chen et al., 2021). For example, HMLC (Chen et al., 2020) uses hyperbolic space to better express the multiplication structure of labels; HypEmo (Chen et al., 2023) constructs emotions into a graph structure and uses hyperbolic space to improve the performance of multiemotion classification; Graph-

HAM (Qiu et al., 2024) and HyperSED (Yu et al., 2025) apply hyperbolic space to the graph structure constructed by social messages and achieves good performance. While these approaches improve the representation of hierarchical structures in graph structures, it is unclear whether hyperbolic spaces can improve the representation of natural language sentences.

3 Preliminaries

This section summarizes concepts related to our work context, including social message stream, social event, SED algorithm, incremental SED, and hyperbolic representation.

Definition 2.1. *Social message stream* denoted as $S = \{M_0, M_1, ..., M_n\}$ is temporal and continuous of blocks of social messages, where M_i is a *message block* that contains all social messages $\{m_0, m_1, ..., m_k\}$ that arrive during time $\{t_i, t_{i+1}\}$.

Definition 2.2. *Social event* $e_i = \{m_i, ..., m_j\}$ is a set of correlated social messages that discuss the same real-world happening. Here, we assume that each social message belongs to at most one social event.

For example, Mo Yan's winning of the 2012 Nobel Prize in Literature sparked widespread discussion on Twitter. Here *2012 Nobel Prize in Literature* can be defined as a social event.

Definition 2.3. We adopt the *Incremental SED* setting from FinEvent (Peng et al., 2022). It is indicated as $f_0, ..., f_{0+w}, ..., f_{t-w}, f_t, ...$, where w is the window size for updating the model parameters.

Definition 2.4. *Hyperbolic Representation* aims to map the features from Euclidean space to hyperbolic space via hyperbolic embedding methods. In this work, we adopt two types of hyperbolic embedding models: $\mathbb{P}$ for the Poincaré Ball model and $\mathbb{H}$ for the Hyperboloid model. For the Poincaré Ball model, we denote $E_o\mathbb{P}^{d,c}$ as the Euclidean space and $\mathbb{P}^{d,c}$ as the hyperbolic representation via the Poincaré Ball model, where o is the center of the space, d is the dimensions, and c is the curvature of this space. Thus, the mapping process from Euclidean space to hyperbolic space is $exp_o^c : E_o\mathbb{P}^{d,c} \to \mathbb{P}^{d,c}$, and the opposite mapping is $log_o^c : \mathbb{P}^{d,c} \to E_o\mathbb{P}^{d,c}$. Specifically, for a node $a \in E_o\mathbb{P}^{d,c}$ and $a' \in \mathbb{P}^{d,c}$, we have: $exp_o^c(a) = a'$ and $log_o^c(a') = a$, where

$$exp_o^c(a) = tanh(\sqrt{c}\|a\|)\frac{a}{\sqrt{c}\|a\|}, \quad (1)$$

$$log_o^c(a') = artanh(\sqrt{c}\|a'\|)\frac{a'}{\sqrt{c}\|a'\|}. \quad (2)$$

For the hyperboloid model, we denote $E_o\mathbb{H}^{d,c}$ as the Euclidean space and $\mathbb{H}^{d,c}$ as the hyperbolic representation via the hyperboloid model and the exp_o^c and log_o^c functions are defined as:

$$exp_o^c(x) = cosh\left(\frac{\|x\|}{\sqrt{c}}\right)y' + \sqrt{c}\cdot sinh\left(\frac{\|x\|}{\sqrt{c}}\right)\frac{x}{\|x\|}, \quad (3)$$

$$log_o^c(x') = d_{\mathbb{H}}^c(x', y')\frac{y' + \frac{1}{c}\langle x', y'\rangle_{\mathcal{M}}x'}{\|y' + \frac{1}{c}\langle x', y'\rangle_{\mathcal{M}}x'\|}, \quad (4)$$

where $x', y' \in \mathbb{H}^{d,c}$, $x \in E_o\mathbb{H}^{d,c}$ with $x' \neq y'$, $x \neq 0$, and $d_{\mathbb{H}}^c(\cdot)$ is the function calculates the distance between two nodes in hyperbolic space and $\langle ., . \rangle_{\mathcal{M}}$ is the Minkowski inner product.

4 Methodology

This section introduces our framework, LLM-enhanced Social Event Detection (LSED). Figure 2 gives an overview of this framework. In general, LSED contains four main steps: (1) Prompt the LLM to summarize the initial social messages based on its knowledge. (2) Vectorize the summarized social messages through the pre-trained language model. (3) Project the vectors into hyperbolic space and cluster them into events. (4) Update LSED according to window size and detect social events in changing message blocks. The process of our framework is shown in Algorithm 1.

4.1 Prompting

One of the primary goals of this work is to improve the effectiveness of the SED model on social messages. The key challenge lies in the lack of context, along with the frequent use of abbreviations and informal expressions, which can lead to ambiguity and loss of meaning. Therefore, LSED applies three state-of-the-art open-source LLMs through OLLama[2]: Meta's **Llama3.1-8B** (Llama Team, 2024), Alibaba's **Qwen2.5-7B** (Yang et al., 2024), and Google's **Gemma2-9B** (Gemma Team, 2024) (we will refer to them as Llama3.1, Qwen2.5, and Gemma2 below). To take advantage of the ability of LLMs to summarize, complete abbreviations, and provide additional context, we design and test a series of prompts on **Llama 3.1** as follows.

"Summarize" or **"Paraphrase"**: To ensure the LLM's response aligns with our expectations, we first focus on selecting keywords in the prompt,

[2]https://ollama.com/

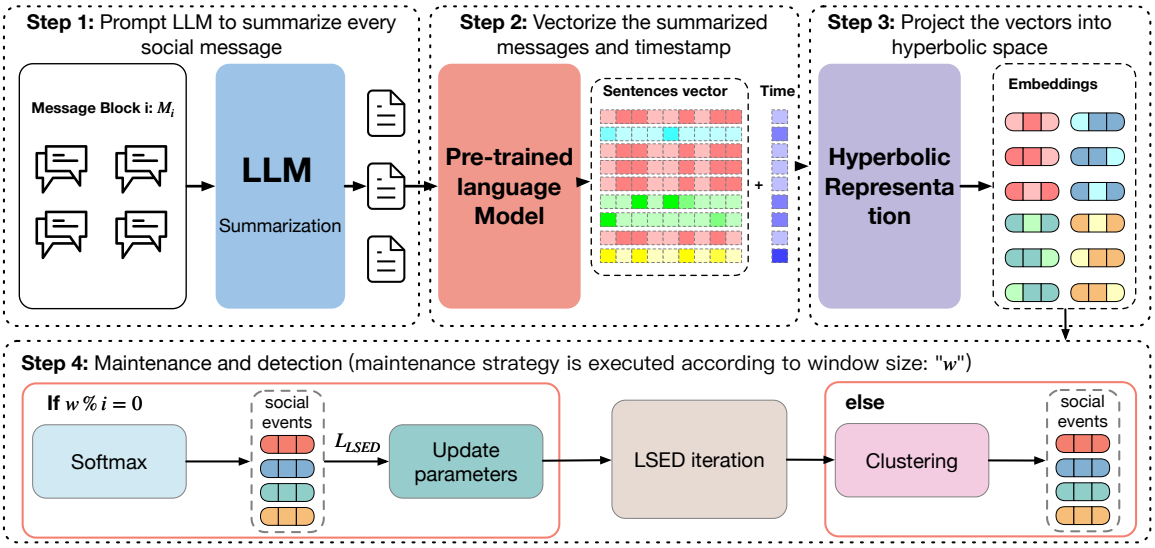
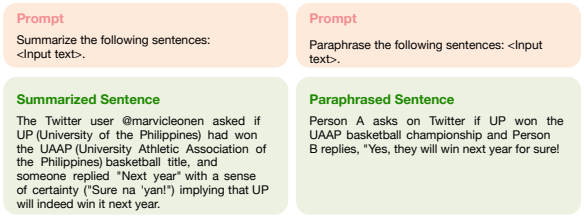
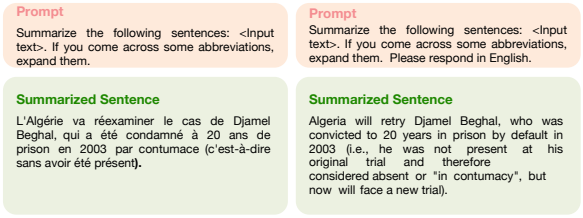


Figure 2: An overview of LSED. The LSED contains three main steps: Step (1): Prompt LLM to summarize social messages; Step (2): Vectorize the summaries and timestamps; Step (3): Hyperbolic representation; Step (4): LSED maintenance and event detection.



(a) Summarize Prompt (b) Paraphrase Prompt

Figure 3: Compare the responses to the "Summarize" and "Paraphrase" prompts.



(a) Without language restriction (b) With language restriction

Figure 4: Comparison of responses with and without language restriction.

as they play a crucial role in guiding the model's generation. Usually, "*summarize*" is used to summarize long texts, but the social message is short, so it seems that "*summarize*" is not exceptionally applicable here. "*Paraphrase*" usually replaces a sentence without changing its meaning, which seems more suitable for our situation. So we test the two prompts "*summarize*" and "*paraphrase*" on the initial message "***@marvicleonen: Is it true that UP won UAAP basketball? – Next year, Dean. Sure na 'yan!***" in Llama3.1, as shown in Figure 3. Counterintuitively, "*summarize*" can better expand the initial social message. Specifically, it can explain the meaning of the two abbreviations "***UP***" and "***UAAP***" in combination with background knowledge, as shown in Figure 3 (a). In addition, we test the impact of "summarize" and "paraphrase" on LSED (see Table 8 in Appendix D). The results show that "*summarize*" performs better than "*paraphrase*". Therefore, in this work, we uniformly use "*summarize*" as the keyword in the prompt. However, during testing, the LLM does not always expand all abbreviations in the initial message, so we improve the prompt to "***Summarize the following sentences: <Input text>. If you come across some abbreviations, expand them***". This restriction helps LLMs meet our needs well.

Multilingual: The datasets we used consist of collections of tweets in both English and French. Although we employ English prompts to interact with the LLM, the model occasionally generates responses in the original language of the tweet. For example, we test the initial tweet "***L'Algérie va rejuger Djamel Beghal, condamné à 20 ans par contumace en 2003***", as shown in Figure 4 (a), the response of Llama3.1 is still in French. We add another restriction to the prompt: "***Please respond in English***", as shown in Figure 4 (b). This setting will help LSED applies the translation ability of the

LLM to solve the problem of multiple languages. Therefore, the prompt we apply in this work is: "***Summarize the following sentences:<Input text>. If you come across some abbreviations, expand them. Please respond in English***". For the social message block M_i from a social message stream S, we have:

$$m_j^s = LLM(m_j), \tag{5}$$

where $m_j \in M_i$ and m_j^s is the summary of social message m_j via LLM.

Additionally, the hallucination problem in LLMs is another issue that warrants attention. However, in general, this problem has minimal impact on LSED. An analysis of the hallucination problem in LSED can be found in Appendix C.

4.2 Vectorization

To embed summarized social messages into hyperbolic space, we transform summarized social messages into vectors through a pre-trained language model (PLM). In this work, we select SBERT (Reimers, 2019) as PLM based on the experimental result in Section 5.3.1. Thus, for a summarised social message m_j^s, we have:

$$v_{m_j^s} = PLM(m_j^s). \tag{6}$$

In addition, the timestamp also plays a crucial role in social messages. Therefore, we adopt the time vectorization approach from KPGNN (Cao et al., 2021) to encode the timestamp. We cover each timestamp with an Object Linking and Embedding (OLE) date, and its fractional and integer parts form a two-dimensional vector. Thus, we have:

$$TIME(t) = (\frac{t_{days}}{D_{max}}, \frac{t_{seconds}}{S_{max}}), \tag{7}$$

where t_{days} and $t_{seconds}$ are the days and seconds in the timestamp, $D_{max} = 100000$ is the normalization factor for days, which controls the scaling range of days. $S_{max} = 86400$ is the number of seconds in a day. They ensure that the vector is normalized to $[0, 1]$. So, for the timestamp t_j for m_j, we have:

$$v_j^t = TIME(t_j). \tag{8}$$

Then, we add (direct addition in vector dimensions) the summarized social message vector and its timestamp vector, defined as $ADD(\mathbb{R}^n, \mathbb{R}^m) = \mathbb{R}^{n+m}$. The final vector of the summarized message m_j^s is:

$$v_j = ADD(v_{m_j^s}, v_j^t). \tag{9}$$

4.3 Hyperbolic Encoder and Clustering

LSED utilizes the hyperbolic space as a low-dimensional embedding space to capture the hierarchical structure of sentences in natural language. To achieve this, we adopt the hyperbolic encoder $H(\cdot)$ from HGCN (Chami et al., 2019). Based on Eq. 1 or Eq. 3 in Section 3. Thus, we have:

$$e_j = H(v_j). \tag{10}$$

After embedding social message vectors into the hyperbolic space, we cluster messages based on the learned message representations. In this work, we adopt distance-based clustering algorithm K-Means (MacQueen et al., 1967).

5 Experiments

We conduct extensive experiments on two real-world datasets to demonstrate the performance of LSED. First, we outline the experimental setup, including the datasets and baselines. Next, we present the overall results for both the offline and online scenarios. An ablation study also highlights the components that contribute to performance improvement, and parameter analysis validates the sensitivity of LSED to key parameters.

5.1 Experimental Setup

5.1.1 Datasets

Two datasets cover a wide range of social event classes and two languages. Events2012 (McMinn et al., 2013) is an English tweet dataset that contains 68,841 tweets belonging to 503 event classes, and Events2018 (Mazoyer et al., 2020) is a French tweet dataset that includes 64,516 tweets belonging to 257 event classes. The datasets are divided into two parts for different scenarios: offline and online. For the offline scenario, the first seven days of tweets in both datasets are collected as offline data, which is called the message block M_0. For the online scenario, all remaining data except M_0 are divided into $\{M_1, M_2, \ldots, M_n\}$ daily. Here, $n = 21$ for Events2012 and $n = 16$ for Events2018. Detailed statistical information is shown in Tabels 6 and 7 in Appendix A.

5.1.2 Baselines and Metrics

To fairly evaluate the overall performance of LSED, we compare LSED with ten **supervised** baseline

Methods	M_0 in Events2012		M_0 in Events2018	
	NMI	AMI	NMI	AMI
TwitterLDA (2011)	.26±.00	.17±.00	.22±.00	.16±.00
Word2Vec (2013)	.47±.00	.21±.00	.24±.00	.20±.00
BERT (2018)	.63±.01	.44±.00	.42±.00	.34±.00
PP-GCN (2019)	.70±.02	.56±.01	.60±.01	.49±.02
EventX (2020)	.68±.00	.29±.00	.57±.00	.56±.00
KPGNN (2021)	.76±.02	.64±.02	.66±.03	.60±.02
QSGNN (2022)	.79±.01	.68±.01	.71±.02	.64±.02
FinEvent (2022)	.86±.01	.77±.01	.78±.01	.74±.01
GraphHAM (2024)	.78±.01	.74±.00	.75±.00	.73±.00
RPLM (2024)	.87±.01	.84±.00	.92±.00	.91±.01
$LSED_{Llama3.1}$	.96±.00	.95±.00	.93±.01	.93±.01
$LSED_{Qwen2.5}$	.97±.01	**.97±.01**	**.97±.00**	**.97±.00**
$LSED_{Gemma2}$	**.97±.00**	.96±.00	.82±.00	.80±.00
Improvement	10% ↑	13% ↑	5% ↑	6% ↑

Table 1: Evaluation on the offline scenario (best results in **bold**).

Methods	Events2012 (AVG.)		Events2018 (AVG.)	
	NMI	AMI	NMI	AMI
TwitterLDA (2011)	.27±.00	.19±.00	.19±.00	.16±.00
Word2Vec (2013)	.36±.00	.26±.00	.34±.00	.31±.00
BERT (2018)	.65±.01	.62±.00	.39±.00	.34±.00
PP-GCN (2019)	.48±.01	.44±.01	.55±.01	.54±.01
EventX (2020)	.60±.00	.17±.00	.45±.00	.16±.00
KPGNN (2021)	.70±.01	.67±.01	.57±.00	.56±.00
QSGNN (2022)	.71±.00	.69±.01	.59±.02	.58±.02
FinEvent (2022)	.79±.01	.78±.01	.68±.01	.63±.01
GraphHAM (2024)	.69±.01	.63±.00	.74±.00	.72±.00
RPLM (2024)	.88±.01	.86±.00	.76±.00	.75±.01
$LSED_{Llama3.1}$	.98±.01	.98±.00	.97±.01	.96±.01
$LSED_{Qwen2.5}$	**.99±.00**	**.99±.00**	**.97±.00**	**.97±.00**
$LSED_{Gemma2}$	.98±.00	.98±.00	.97±.01	.97±.01
Improvement	11% ↑	13% ↑	21% ↑	22% ↑

Table 2: Evaluation on the online scenario. This table reports the average experimental results of the message blocks (best results in **bold**).

models ranging from traditional to state-of-the-art SED models. The baselines include: **TwitterLDA** (Zhao et al., 2011), **Word2Vec** (Mikolov et al., 2013), **BERT** (Devlin et al., 2018), **PP-GCN** (Peng et al., 2019), **EventX** (Liu et al., 2020), **KPGNN** (Cao et al., 2021), **QSGNN** (Ren et al., 2022), **FinEvent** (Peng et al., 2022), **GraphHAM** (Qiu et al., 2024), and **RPLM** (Li et al., 2024).

Normalized Mutual Information (NMI) (Estévez et al., 2009) and Adjusted Mutual Information (AMI) (Vinh et al., 2009) are applied as metrics to evaluate the performance of the models, which are widely used in previous studies (Cao et al., 2024, 2021). Details of the model implementation settings are given in Appendix B.

5.2 Overall Results

This section experiments with LSED in two scenarios and the different LLMs. The different LLMs in LSED are represented as $LSED_{Llama3.1}$, $LSED_{Qwen2.5}$, and $LSED_{Gemma2}$, respectively.

Table 1 reports the experimental results of the offline scenario. In general, LSED outperforms all baseline models on both datasets. Compared to the latest RPLM model, the NMI and AMI on the Events2012 and Events2018 datasets are improved by 10%, 13%, 5%, and 5%, respectively. It is obvious that the traditional event detection model, like TwitterLDA, performs poorly due to the influence of short texts. SED models (PP-GCN, KPGNN, QSGNN, FinEvent, and GraphHAM) that use graph structures to supplement external information perform better. However, their reliance on explicit structural relations for representation learning hinders them from performing better. In contrast, RPLM achieves the best performance among the baselines as it eliminates the reliance on explicit structures and effectively captures the relationship between structure and semantics. However, since it mitigates the impact of short texts from an external perspective rather than directly addressing the inherent challenges, its performance remains inferior to that of LSED.

In the online scenario, we apply the first week's data as the message block M_0 to train the initial framework. Then LSED maintains training or inference according to the window size w. Table 2 reports the average performance of all message blocks in the online scenario (the specific performance of each message block can be found in Tables 9 and 10 in Appendix E). LSED outperforms all baseline methods in the online scenario, consistently leading in evaluation metrics across all message blocks. The baseline models exhibit performance similar to that in the offline scenario, with RPLM remaining the best among them, further supporting our analysis in the offline scenario. However, LSED still performs better than RPLM on both datasets. On the Events2012 dataset, the average NMI and AMI increased by 11% and 12%, respectively. On the Events2018 dataset, the average NMI and AMI increased by 21% and 22%, respectively.

Furthermore, we evaluate the stability of LSED in the online scenario (see Appendix E for details). In general, text analysis-based SED models exhibit stability but struggle to effectively capture the characteristics of social messages. Graph-based SED models achieve strong performance but suffer from instability. In contrast, LSED harnesses the capabilities of LLMs for short-text processing and

Datasets	Events2012		Events2018	
Message Block M_0	NMI	AMI	NMI	AMI
$LSED^{W2V}$ w/o time	.85±.01	.82±.01	.74±.03	.72±.03
$LSED^{SBERT}$ w/o time	.94±.00	.93±.00	.92±.01	.91±.01
$LSED^{W2V}$	.87±.01	.83±.02	.76±.06	.74±.07
$LSED^{SBERT}$	**.96±.00**	**.95±.00**	**.94±.00**	**.94±.00**

Table 3: Comparions between SBERT and Word2Vec with or without time vector (best results in **bold**).

Datasets	Events2012		Events2018	
Sentence Depth	$A_d^i = 3.92 \rightarrow A_d^s = 5.17$		$A_d^i = 4.19 \rightarrow A_d^s = 6.73$	
Message Block M_0	NMI	AMI	NMI	AMI
$LSED$ w/o LLM & H	.53±.02	.47±.02	.38±.03	.35±.03
$LSED$ w/o LLM	.87±.02	.84±.02	.78±.09	.75±.09
$LSED$ w/o H	.68±.01	.64±.01	.50±.01	.48±.01
$LSED_{\mathbb{H}}$	.72±.03	.67±.04	.62±.04	.59±.05
$LSED_{\mathbb{P}}$	**.96±.00**	**.95±.00**	**.94±.00**	**.94±.00**

Table 4: The impact of hyperbolic space and sentence depth on performance (best results in **bold**). Here, "A_d^i" means the average depth of initial messages in the dataset; "A_d^s" means the average depth of the summarised messages in the dataset; "*w/o LLM & H*" means the LSED does not use LLMs and hyperbolic encoder; "*w/o LLM*" means not use LLM; "*w/o H*" means not use hyperbolic encoder; "$LSED_{\mathbb{H}}$" means LSED embeds vectors into hyperbolic space through the Hyperboloid model; "$LSED_{\mathbb{P}}$" means LSED embeds vectors into hyperbolic space through the Poincaré Ball model.

leverages hyperbolic space embeddings, ensuring both efficiency and stability while eliminating reliance on graph structures, thereby demonstrating its robustness in dynamic social message environments.

5.3 Ablation Studies

To evaluate the impact of different components in LSED, we conduct a series of ablation experiments, including comparing the choice of model during text vectorization, the hierarchical structure of social messages, and the impact of hyperbolic space. Since the performance between the three LLMs in this work is close, in ablation studies, we only report the experiments under Llama3.1.

5.3.1 Impact of Vectorization Methods

In this work, we adopt two widely used pre-trained language models, SBERT (Reimers, 2019) and Word2Vec (Mikolov et al., 2013), to vectorize social messages. The difference between SBERT and Word2Vec is that SBERT can contain more semantic information and hierarchical structures of sentences. To verify this, we experiment with the messages summarized by Llama3.1 in the message block M_0. We also study the effect of the time vector.

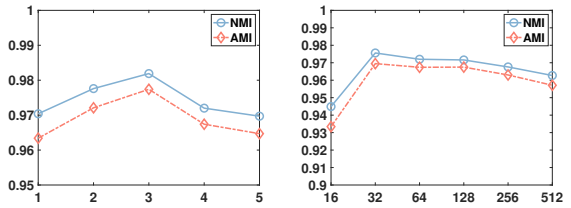
As shown in Table 3, we use "*W2V*" represent "*Word2Vec*" and "*w/o time*" to represent LSED training without time vector. LSED achieves an average performance improvement of 8% when using vectors generated by SBERT compared to those generated by Word2Vec. This demonstrates the suitability of SBERT for LSED and indirectly indicates that SBERT provides richer information and further improves the capabilities of hyperbolic space.

In addition, incorporating the time vector of each social message further enhances LSED's performance. While the improvement is not as significant as the transition from Word2Vec to SBERT, it still increases NMI and AMI by approximately 2% on both datasets.

5.3.2 Effect of Hierarchical Structure and Hyperbolic Embedding

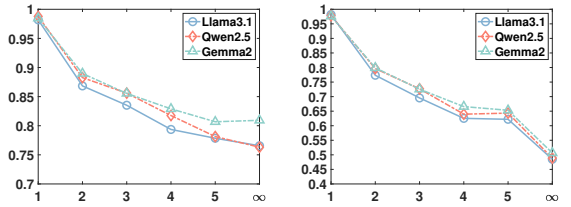
To quantify the hierarchical structure of a sentence, we use a dependency tree to calculate the depth of a sentence. The deeper the depth, the more complex the sentence. We calculate the average sentence depth of initial messages and summarised messages in the Events2012 and Events2018 datasets, defined as A_d^i and A_d^s. Table 4 shows that the average sentence depth of the datasets summarized by Llama3.1 has increased, which helps LSED's NMI increase by 19% on the Events2012 dataset and 12% on the Events2018 dataset. However, the performance gains from capturing the hierarchical structure of sentences appear to surpass those achieved through the use of LLMs. The hyperbolic embedding plays a more significant role, helping LSED's NMI to improve by 34% and 37% on Events2012 and Events2018, respectively. Therefore, we recommend prioritizing the hyperbolic space as a low-dimensional embedding space when computational resources are limited. While LLMs can mitigate limitations in short texts, selecting an appropriate embedding space is more crucial for ensuring efficiency and effectiveness.

Notably, we evaluate two hyperbolic embedding methods and find that while both effectively project features into hyperbolic space, the Poincaré Ball model better captures the hierarchical structure of sentences. This highlights the importance of selecting an appropriate embedding model when leveraging hyperbolic space for feature representation.



(a) Hidden layers (b) Hidden dimensions

Figure 5: LSED's hidden layers and hidden dimensions.



(a) Events2012 (b) Events2018

Figure 6: The performance of LSED with different window sizes. ∞ means we only train LSED on message block M_0 and directly infer all subsequent message blocks.

5.4 Parameter Sensitivity Analysis

We investigate the sensitivity of LSED concerning parameters: the number of hidden layers, hidden dimensions, and window size w. Figure 5 shows that the hidden layer has no significant effect on LSED, but LSED performs best when the hidden layers are 3. As for hidden dimensions, the framework performs best when the hidden dimensions are 32 to 128. Therefore, our strategy is to set the number of hidden layers to 3 and the hidden dimension size to 64.

Figure 6 shows the performance of LSED with different window sizes w on two datasets. We compute the mean NMI of all message blocks, excluding M_0. LSED performs best when $w = 1$, and the average performance of LSED slowly decreases as w increases. In particular, LSED performs better on the Events2012 dataset and maintains a good inference ability when $w = \infty$. In contrast, the performance of LSED on the Events2018 dataset decreases faster when w increases, which means that when the dataset is more complex, LSED needs a smaller window size to maintain its performance.

5.5 Time Efficiency

This section shows the inference time cost of using LLM in LSED, and also demonstrates that the time cost of using LLM is better than that of SED models based entirely on GNNs (such as KPGNN and FinEvent) under the same experimental environment (a V100 GPU).

Datasets	Events2012		Events2018	
Models	Inference Time	Total Time	Inference Time	Total Time
KPGNN	–	$>$ 24 h	–	$>$ 24 h
FinEvent	–	$>$ 24 h	–	$>$ 24 h
$LSED_{Llama3.1}$	19 h 07 m	19 h 10 m	17 h 55 m	17 h 58 m
$LSED_{Qwen2.5}$	12 h 25 m	12 h 28 m	11 h 39 m	11 h 42 m
$LSED_{Gemma2}$	15 h 18 m	15 h 21 m	14 h 20 m	14 h 23 m

Table 5: Inference time of LLMs in LSED.

Table 5 reports the inference time and the total training time of LSED based on different LLMs and compares with some GNN-based SED models on the Events2012 and Events2018 datasets. The LLM inference time in LSED constitutes the majority of the total time cost. Among them, LSED based on Qwen2.5 requires the least time. In contrast, GNN-based models, KPGNN and FinEvent, exceed 24 hours on both datasets. This shows that LSED has a clear advantage over SED models reliant on GNNs, particularly in terms of efficiently processing large datasets and adjusting parameters.

6 Conclusion

In this work, we propose LSED, a novel framework for social event detection (SED) that takes advantage of LLMs and hyperbolic embeddings to enhance **effectiveness**, **efficiency**, and **stability**. Unlike existing Graph Neural Network (GNN)-based or graph structure-based approaches, LSED focuses solely on textual information, rethinking the core challenge of SED. By utilizing LLMs to summarize and reformat social messages, LSED effectively mitigates the impact of abbreviations and informal expressions and improves clustering performance. Moreover, not relying on graph structures can better improve the stability of the framework in dynamic messages and make it more suitable for practical applications.

Additionally, this work demonstrates that hyperbolic space embeddings capture the implicit hierarchical structures in natural language, further enhancing event detection. Experimental results on two real-world datasets show that LSED achieves competitive and superior performance compared to SOTA SED models. Our findings highlight the potential of the LLM-driven method and hyperbolic space embedding for efficient and scalable SED.

Limitations

Our work focuses on accurately detecting social events without relying on graph structures, with LLMs serving only as an auxiliary component in LSED. Consequently, we did not investigate the relationship between prompt design and LLM hallucinations, nor the impact of hallucination levels on LSED performance. However, hallucinations remain a critical challenge in LLM applications. While we conducted a hallucination analysis on the first 100 tweets from Events2012, this evaluation does not provide a comprehensive understanding of hallucination prevalence across the entire dataset. In addition, our hallucination detection prompt is specifically designed for English tweets, and we currently lack a solution for Events2018, which contains French tweets. Future research should address these limitations to further enhance the robustness and generalizability of SED models.

Acknowledgments

This work was supported by the Australian Research Council Project with No. DP230100899, Macquarie University Data Horizons Research Centre and Applied Artificial Intelligence Centre. Corresponding author: Jia Wu.

References

James Allan. 2002. *Topic detection and tracking: event-based information organization*, volume 12. Springer Science & Business Media.

Farzindar Atefeh and Wael Khreich. 2015. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164.

Danushka Bollegala, Vincent Atanasov, Takanori Maehara, and Ken-Ichi Kawarabayashi. 2018. Classinet–predicting missing features for short-text classification. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(5):1–29.

Yuwei Cao, Hao Peng, Jia Wu, Yingtong Dou, Jianxin Li, and Philip S Yu. 2021. Knowledge-preserving incremental social event detection via heterogeneous gnns. In *Proceedings of the Web Conference 2021*, pages 3383–3395, Ljubljana, Slovenia. ACM.

Yuwei Cao, Hao Peng, Zhengtao Yu, and S Yu Philip. 2024. Hierarchical and incremental structural entropy minimization for unsupervised social event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8255–8264.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32.

Boli Chen, Yao Fu, Guangwei Xu, Pengjun Xie, Chuanqi Tan, Mosha Chen, and Liping Jing. 2021. Probing bert in hyperbolic spaces. *arXiv preprint arXiv:2104.03869*.

Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. 2020. Hyperbolic interaction model for hierarchical multi-label classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7496–7503.

Chih-Yao Chen, Tun-Min Hung, Yi-Li Hsu, and Lun-Wei Ku. 2023. Label-aware hyperbolic embeddings for fine-grained emotion classification. *arXiv preprint arXiv:2306.14822*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, MN, USA. Association for Computational Linguistics.

Bhuwan Dhingra, Christopher J Shallue, Mohammad Norouzi, Andrew M Dai, and George E Dahl. 2018. Embedding text in hyperbolic spaces. *arXiv preprint arXiv:1806.04313*.

Pablo A Estévez, Michel Tesmer, Claudio A Perez, and Jacek M Zurada. 2009. Normalized mutual information feature selection. *IEEE Transactions on neural networks*, 20(2):189–201.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic entailment cones for learning hierarchical embeddings. In *International Conference on Machine Learning*, pages 1646–1655, Stockholm, Sweden. PMLR.

Rui Gaspar, Cláudia Pedro, Panos Panagiotopoulos, and Beate Seibt. 2016. Beyond positive or negative: Qualitative sentiment analysis of social media reactions to unexpected stressful events. *Computers in Human Behavior*, 56:179–191.

Google DeepMind Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

Pu Li, Xiaoyan Yu, Hao Peng, Yantuan Xian, Linqin Wang, Li Sun, Jingyun Zhang, and Philip S Yu. 2024. Relational prompt-based pre-trained language models for social event detection. *ACM Transactions on Information Systems*.

Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, et al. 2022. A survey on deep learning event extraction: Approaches and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Bang Liu, Fred X Han, Di Niu, Linglong Kong, Kunfeng Lai, and Yu Xu. 2020. Story forest: Extracting events and telling stories from breaking news. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(3):1–28.

Chuang Liu, Xiu-Xiu Zhan, Zi-Ke Zhang, Gui-Quan Sun, and Pak Ming Hui. 2015. How events determine spreading patterns: information transmission via internal and external influences on social networks. *New Journal of Physics*, 17(11):113045.

AI Meta Llama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Congbo Ma, Zitai Qiu, Hu Wang, Jing Du, Shan Xue, Jia Wu, and Jian Yang. 2025. Enhanced social event detection through dynamically weighted meta-paths modeling. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 1184–1188.

Congbo Ma, Hu Wang, Zitai Qiu, Shan Xue, Jia Wu, Jian Yang, Preslav Nakov, and Quan Z Sheng. 2024. Learning to sample the meta-paths for social event detection. *arXiv preprint arXiv:2411.12588*.

James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Fabrizio Marozzo and Alessandro Bessi. 2018. Analyzing polarization of social media users and news sites during political campaigns. *Social Network Analysis and Mining*, 8:1–13.

Béatrice Mazoyer, Julia Cagé, Nicolas Hervé, and Céline Hudelot. 2020. A french corpus for event detection on twitter.

Andrew J McMinn, Yashar Moshfeghi, and Joemon M Jose. 2013. Building a large-scale corpus for evaluating event detection on twitter. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 409–418.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations*, Scottsdale, Arizona, USA. ICLR.

Tahir M Nisar and Man Yeung. 2018. Twitter as a tool for forecasting stock market movements: A short-window event study. *The journal of finance and data science*, 4(2):101–119.

Hao Peng, Jianxin Li, Qiran Gong, Yangqiu Song, Yuanxing Ning, Kunfeng Lai, and Philip S Yu. 2019. Fine-grained event categorization with heterogeneous graph convolutional networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 3238–3245, Macao, China. IJCAI.

Hao Peng, Jianxin Li, Yangqiu Song, Renyu Yang, Rajiv Ranjan, Philip S Yu, and Lifang He. 2021. Streaming social event detection and evolution discovery in heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(5):1–33.

Hao Peng, Ruitong Zhang, Shaoning Li, Yuwei Cao, Shirui Pan, and S Yu Philip. 2022. Reinforced, incremental and cross-lingual event detection from social messages. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):980–998.

Zitai Qiu, Congbo Ma, Jia Wu, and Jian Yang. 2024. An efficient automatic meta-path selection for social event detection via hyperbolic space. In *Proceedings of the ACM on Web Conference 2024*, pages 2519–2529.

Zitai Qiu, Jia Wu, Jian Yang, Xing Su, and Charu Aggarwal. 2025. Heterogeneous social event detection via hyperbolic graph representations. *IEEE Transactions on Big Data*, 11(1):115–129.

Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, pages 29–48. Citeseer.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Jiaqian Ren, Lei Jiang, Hao Peng, Yuwei Cao, Jia Wu, Philip S Yu, and Lifang He. 2022. From known to unknown: Quality-aware self-improving graph neural network for open set social event detection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1696–1705.

Xiaoming Shi, Siqiao Xue, Kangrui Wang, Fan Zhou, James Zhang, Jun Zhou, Chenhao Tan, and Hongyuan Mei. 2024. Language models can improve event prediction by few-shot abductive reasoning. *Advances in Neural Information Processing Systems*, 36.

Ge Song, Yunming Ye, Xiaolin Du, Xiaohui Huang, and Shifu Bie. 2014. Short text classification: a survey. *Journal of multimedia*, 9(5).

Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning*, pages 1073–1080.

Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W White, Longqi Yang, et al. 2024. Tnt-llm: Text mining at scale with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5836–5847.

Zhibo Wang, Yongquan Zhang, Yijie Li, Qian Wang, and Feng Xia. 2017. Exploiting social influence for context-aware event recommendation in event-based social networks. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE.

Jingyun Xu and Yi Cai. 2019. Incorporating context-relevant knowledge into convolutional neural networks for short text classification. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 10067–10068.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Tianchi Yang, Linmei Hu, Chuan Shi, Houye Ji, Xiaoli Li, and Liqiang Nie. 2021. Hgat: Heterogeneous graph attention networks for semi-supervised short text classification. *ACM Transactions on Information Systems (TOIS)*, 39(3):1–29.

Xiaoyan Yu, Yifan Wei, Pu Li, Shuaishuai Zhou, Hao Peng, Li Sun, Liehuang Zhu, and Philip S Yu. 2024. Dame: Personalized federated social event detection with dual aggregation mechanism. *arXiv preprint arXiv:2409.00614*.

Xiaoyan Yu, Yifan Wei, Shuaishuai Zhou, Zhiwei Yang, Li Sun, Hao Peng, Liehuang Zhu, and Philip S Yu. 2025. Towards effective, efficient and unsupervised social event detection in the hyperbolic space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13106–13114.

Yongsheng Yu, Jia Wu, and Jian Yang. 2023. Social event detection with reinforced deep heterogeneous graph attention network. In *2023 IEEE International Conference on Big Data (BigData)*, pages 463–472. IEEE.

Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval: 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings 33*, pages 338–349. Springer.

A Datasets

Two widely used real-world datasets are used to verify LSED's ability to cluster social events: Events2012 (McMinn et al., 2013), and Events2018 (Mazoyer et al., 2020). These two datasets collect a large number of tweets through the API provided by Twitter. Events2012 is an English tweet dataset that contains 68,841 tweets covering 503 social events; Events2018 is a French tweet dataset that contains 64,516 tweets covering 257 social events. To verify the ability of LSED to detect incremental social events, the datasets are divided into two parts according to the time order: offline and online. For the offline scenario, the first seven days of tweets in both datasets are collected as offline data, which is called the message block M_0. For the online scenario, all remaining data except M_0 are divided into $\{M_1, M_2, \ldots, M_n\}$ daily. Here, $n = 21$ for Events2012 and $n = 16$ for Events2018. Tables 6 and 7 present the statistics for the offline and online scenarios in both datasets.

Message Block M_0	Events2012	Events2018
No. of Messages	20,254	14,328
No. of Event Types	155	79

Table 6: Offline data statistics.

B Implementation Settings

In LSED, we implement SBERT using "**all-MiniLM-L6-v2**" from **SBERT.net**[3], and implement Word2Vec using "**en-core-web-sm**" from **spaCy**[4] to vectorize text messages. We employ the Adam optimizer for training for model configuration, with a learning rate of 0.01 and a window size of 1. Since the hyperbolic encoder of LSED is based on MLP, we set the number of hidden layers to 3 and the hidden dimensions to 64. The datasets are divided into train, validation, and test sets using a split ratio of $70\% - 10\% - 20\%$. All the experiments are conducted on 1 NVIDIA V100 GPU with 32G RAM. We run five tests to test generalizability and report the average result and standard deviation.

C Hallucination Analysis

In LSED, an LLM serves as an auxiliary component. We do not directly use LLM-generated outputs for prediction, nor do we rely on their results. Instead, LLM is employed to summarize short texts, helping to mitigate the impact of abbreviations and informal expressions without introducing new content. To further verify the LLMs' generated

[3] https://www.sbert.net/index.html
[4] https://spacy.io/models/en

Blocks	M_1		M_2		M_3		M_4		M_5		M_6		M_7	
Datasets	2012	2018	2012	2018	2012	2018	2012	2018	2012	2018	2012	2018	2012	2018
No. of Messages	8,722	5,356	1,491	3,186	1,835	2,644	2,010	3,179	1,834	2,662	1,276	4,200	5,278	3,454
No. of Event Types	41	22	30	19	33	15	38	19	30	27	44	26	57	23
Blocks	M_8		M_9		M_{10}		M_{11}		M_{12}		M_{13}		M_{14}	
Datasets	2012	2018	2012	2018	2012	2018	2012	2018	2012	2018	2012	2018	2012	2018
No. of Messages	1,560	2,257	1,363	3,669	1,096	2,385	1,232	2,802	3,237	2,927	1,972	4,884	2,956	3,065
No. of Event Types	53	25	38	31	33	32	30	31	42	29	40	28	43	26
Blocks	M_{15}		M_{16}		M_{17}		M_{18}		M_{19}		M_{20}		M_{21}	
Datasets	2012	2018	2012	2018	2012	2018	2012	2018	2012	2018	2012	2018	2012	2018
No. of Messages	2,549	2,411	910	1,107	2,676	–	1,887	–	1,399	–	893	–	2,410	–
No. of Event Types	42	25	27	26	35	–	32	–	28	–	34	–	32	–

Due to length limitations, in this table, we represent the datasets "Events2012" and "Events2018" as "2012" and "2018", respectively.

Table 7: Online data statistics.

data, we selected 100 samples from the Events2012 dataset to analyze hallucination. The results indicate that 96% of the summaries did not contain hallucinated information, suggesting that the content generated by LLM remains largely faithful to the original input. Figure 7 shows the prompt for the analysis of hallucinations in this work.

Prompt

You are a helpful and impartial assistant. You will receive an original text and a version rewritten by AI. Your task is to evaluate whether the AI-generated text contains any factual inaccuracies or irrelevant information based on the original text.

Please note that if the AI-generated text introduces new, factually correct information or elaborates on the original text in a reasonable way, it should not be considered a hallucination.

Please use the following scale to rate your evaluation:

Rating: [[2]]: The AI-generated text is mostly consistent with the original text, and any new information provided is factually correct and relevant.
Rating: [[1]]: The AI-generated text contains factual inaccuracies, irrelevant additions, or misinterpretations of the original text.

Provide your rating strictly in this format: "Rating: [[rating]]", where the rating inside the double brackets must be either 1 or 2.

Figure 7: The prompt to verify whether the summary text is hallucinated.

D Effect of Prompts on LSED Performance

We report the effect of different prompts on the LSED performance. Table 8 shows the impact of using the "*summarize*" and "*paraphrase*" prompts on LSED's NMI on Llama3.1. The experiments are conducted on the Events2012 dataset. We report the results of all the message blocks in the online scenario and the average results as shown in Table 8.

In general, these two prompts have little effect on LSED. LSED using "*summarize*" and LSED using "*paraphrase*" perform similarly in different message blocks, but the average results show that LSED using "*summarize*" performs better. Therefore, we still recommend using "*summarize*" as the keyword in the LSED prompt.

E Online Experimental Results and Stability Analysis

Tables 9 and 10 report the performance of the baseline models and the LSED for each message block on the Events2012 and Events2018 datasets in an online scenario. The GNN-based SED models (PP-GCN, KPGNN, QSGNN, FinEvent, and Graph-HAM) exhibit performance degradation in certain message blocks of Events2012 (e.g., M7, M12, M15) and Events2018 (e.g., M3, M5, M6). While RPLM mitigates the dependence on explicit graph structures, its performance remains volatile across different message blocks. In contrast, LSED maintains stable performance across all message blocks, demonstrating its robustness to structural changes in message blocks.

Blocks (#events)	M_1 (41)	M_2 (30)	M_3 (33)	M_4 (38)	M_5 (30)	M_6 (44)	M_7 (57)	M_8 (53)	M_9 (38)	M_{10} (33)	M_{11} (30)
$LSED^{Summarize}_{LLama3.1}$	**.97±.00**	**.98±.02**	**.99±.01**	.96±.01	.97±.00	**.99±.00**	**.98±.00**	.98±.01	.99±.01	**1.0±.00**	**.99±.00**
$LSED^{Paraphrase}_{LLama3.1}$	.96±.01	.97±.01	**.99±.01**	**.97±.02**	**.99±.01**	.99±.01	**.98±.00**	**.99±.00**	**.99±.00**	.99±.01	.99±.01
Blocks (#events)	M_{12} (42)	M_{13} (40)	M_{14} (43)	M_{15} (42)	M_{16} (27)	M_{17} (35)	M_{18} (32)	M_{19} (28)	M_{20} (34)	M_{21} (32)	AVG.
$LSED^{Summarize}_{LLama3.1}$	**.97±.00**	**.99±.00**	.97±.00	**.98±.01**	.99±.01	.98±.02	**.98±.01**	**.99±.01**	**.97±.01**	.96±.02	**.9819**
$LSED^{Paraphrase}_{LLama3.1}$	.97±.01	.98±.01	**.98±.00**	**.98±.01**	**1.0±.00**	**.98±.01**	.97±.00	**.99±.01**	**.97±.01**	**.98±.01**	.9809

Table 8: Effects of different prompts on LSED's NMI performance on Events2012 (best results in **bold**).

Blocks (#events)	M_1 (41)		M_2 (30)		M_3 (33)		M_4 (38)		M_5 (30)		M_6 (44)		M_7 (57)	
Metrics	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI
TwitterLDA (2011)	.11±.00	.08±.00	.27±.01	.20±.01	.28±.00	.22±.01	.25±.00	.17±.00	.26±.00	.21±.00	.32±.00	.20±.00	.18±.01	.12±.01
Word2Vec (2013)	.19±.00	.08±.00	.50±.00	.41±.00	.39±.00	.31±.00	.34±.00	.24±.00	.41±.00	.33±.00	.53±.00	.40±.00	.25±.00	.13±.00
BERT (2018)	.36±.00	.34±.00	.78±.00	.76±.00	.75±.00	.73±.00	.60±.00	.55±.00	.72±.00	71±.00	.78±.00	.74±.00	.54±.00	.50±.00
PP-GCN (2019)	.23±.00	.21±.00	.57±.02	.55±.02	.55±.01	.52±.01	.46±.01	.42±.01	.48±.01	.46±.01	.57±.01	.52±.02	.37±.00	.34±.00
EventX (2020)	.36±.00	.06±.00	.68±.00	.29±.00	.63±.00	.18±.00	.63±.00	.19±.00	.59±.00	.14±.00	.70±.00	.27±.00	.51±.00	.13±.00
KPGNN (2021)	.39±.00	.37±.00	.79±.01	.78±.01	.76±.00	.74±.00	.67±.00	.64±.01	.73±.01	.71±.01	.82±.01	.79±.01	.55±.01	.51±.01
QSGNN (2022)	.43±.01	.41±.02	.81±.02	.80±.01	.78±.01	.76±.01	.71±.02	.68±.01	.75±.00	.73±.00	.83±.01	.80±.01	.57±.01	.54±.00
FinEvent (2022)	.84±.01	.84±.01	.84±.01	.84±.01	.89±.00	.89±.01	.71±.01	.69±.00	.83±.00	.82±.00	.83±.00	.82±.02	.73±.01	.72±.00
GraphHAM (2024)	.71±.00	.68±.00	.75±.01	.71±.01	.76±.00	.70±.00	.67±.00	.61±.00	.69±.00	.60±.00	.80±.01	.74±.01	.66±.00	.60±.00
RPLM (2024)	.91±.02	.91±.01	.91±.01	.91±.00	.93±.00	.93±.00	.83±.01	.81±.01	.85±.02	.84±.02	.92±.00	.91±.00	.88±.01	.88±.01
$LSED_{Llama3.1}$	**.97±.00**	**.97±.00**	.98±.02	.98±.02	**.99±.01**	**.99±.01**	.96±.01	.95±.01	.97±.00	.97±.01	**.99±.00**	.98±.00	.98±.00	**.99±.00**
$LSED_{Qwen2.5}$	**.97±.00**	**.97±.00**	**.99±.01**	**.99±.01**	.98±.01	.98±.01	**.98±.01**	.97±.02	.99±.00	**.99±.00**	**.99±.01**	**.99±.01**	**.99±.00**	**.99±.00**
$LSED_{Gemma2}$	.97±.01	.97±.01	.98±.00	.97±.00	.98±.00	.98±.00	**.98±.01**	**.97±.01**	**1.0±.00**	**.99±.00**	**.99±.01**	.98±.01	.98±.01	.98±.01
Improvement	8% ↑	8% ↑	8% ↑	8% ↑	6% ↑	6% ↑	15% ↑	16% ↑	15% ↑	15% ↑	7% ↑	8% ↑	11% ↑	11% ↑
Blocks (#events)	M_8 (53)		M_9 (38)		M_{10} (33)		M_{11} (30)		M_{12} (42)		M_{13} (40)		M_{14} (43)	
Metrics	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI
TwitterLDA (2011)	.37±.01	.24±.01	.34±.00	.24±.00	.44±.01	.36±.01	.33±.01	.25±.01	.22±.01	.16±.01	.27±.00	.19±.00	.21±.00	.15±.01
Word2Vec (2013)	.46±.00	.33±.00	.35±.00	.24±.00	.51±.00	.39±.00	.37±.00	.26±.00	.30±.00	.23±.00	.37±.00	.23±.00	.36±.00	.26±.00
BERT (2018)	.79±.00	.75±.00	.70±.00	.66±.00	.74±.00	.70±.00	.68±.00	.65±.00	.59±.00	.56±.00	.63±.00	.59±.00	.64±.00	.61±.00
PP-GCN (2019)	.55±.02	.49±.02	.51±.02	.46±.02	.55±.02	.51±.02	.50±.01	.46±.02	.45±.01	.42±.01	.47±.01	.43±.01	.44±.01	.41±.01
EventX (2020)	.71±.00	.21±.00	.67±.00	.19±.00	.68±.00	.24±.00	.65±.00	.24±.00	.61±.00	.16±.00	.58±.00	.16±.00	.57±.00	.14±.00
KPGNN (2021)	.80±.00	.76±.01	.74±.02	.71±.02	.80±.01	.78±.01	.74±.01	.71±.01	.68±.01	.66±.01	.69±.01	.67±.01	.69±.00	.65±.00
QSGNN (2022)	.79±.01	.75±.01	.77±.02	.75±.02	.82±.02	.80±.03	.75±.01	.72±.01	.70±.00	.68±.00	.68±.02	.66±.01	.68±.01	.66±.01
FinEvent (2022)	.87±.02	.87±.01	.79±.01	.78±.01	.82±.01	.81±.00	.75±.00	.74±.00	.67±.01	.67±.02	.79±.00	.79±.00	.82±.00	.82±.01
GraphHAM (2024)	.71±.01	.60±.00	.80±.01	.74±.01	.80±.00	.73±.00	.68±.00	.60±.00	.62±.01	.56±.00	.79±.01	.74±.00	.68±.01	.65±.00
RPLM (2024)	.88±.00	.86±.00	.92±.01	.91±.00	.91±.01	.90±.01	.88±.01	.87±.01	.92±.00	.77±.00	.91±.01	.91±.00	.88±.00	.88±.01
$LSED_{Llama3.1}$	.98±.01	.96±.01	.99±.01	.99±.01	**1.0±.00**	**1.0±.00**	**.99±.00**	.98±.01	.97±.00	.97±.00	**.99±.00**	**.99±.00**	.97±.00	.97±.00
$LSED_{Qwen2.5}$	**.99±.00**	**.98±.00**	**1.0±.00**	**.99±.00**	**1.0±.00**	**1.0±.00**	.98±.01	.98±.01	.98±.01	.98±.01	.99±.01	.98±.01	**.98±.00**	**.98±.00**
$LSED_{Gemma2}$	.99±.01	.98±.01	.99±.00	.98±.00	.99±.01	.99±.01	**.99±.00**	**.99±.01**	**.99±.01**	**.99±.01**	.98±.00	.98±.00	.97±.02	.97±.03
Improvement	11% ↑	12% ↑	8% ↑	8% ↑	9% ↑	10% ↑	11% ↑	12% ↑	7% ↑	22% ↑	8% ↑	8% ↑	10% ↑	10% ↑
Blocks (#events)	M_{15} (42)		M_{16} (27)		M_{17} (35)		M_{18} (32)		M_{19} (28)		M_{20} (34)		M_{21} (32)	
Metrics	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI
TwitterLDA (2011)	.21±.00	.13±.00	.35±.01	.27±.01	.19±.00	.13±.00	.18±.00	.12±.00	.29±.01	.22±.00	.35±.00	.23±.00	.19±.00	.13±.00
Word2Vec (2013)	.27±.00	.15±.00	.49±.00	.36±.00	.33±.00	.24±.00	.29±.00	.21±.00	.37±.00	.28±.00	.38±.00	.24±.00	.31±.00	.21±.00
BERT (2018)	.54±.00	.50±.00	.75±.00	.72±.00	.63±.00	.60±.00	.57±.00	.53±.00	.66±.00	.63±.00	.68±.00	.62±.00	.59±.00	.57±.00
PP-GCN (2019)	.39±.01	.35±.01	.55±.01	.52±.01	.48±.00	.45±.00	.47±.01	.45±.01	.51±.02	.48±.02	.51±.01	.45±.02	.41±.02	.38±.02
EventX (2020)	.49±.00	.07±.00	.62±.00	.19±.00	.58±.00	.18±.00	.59±.00	.16±.00	.60±.00	.16±.00	.67±.00	.18±.00	.53±.00	.10±.00
KPGNN (2021)	.58±.00	.54±.00	.79±.01	.77±.01	.70±.01	.68±.01	.68±.02	.66±.02	.73±.01	.71±.01	.72±.02	.68±.02	.60±.00	.57±.00
QSGNN (2022)	.59±.01	.55±.01	.78±.01	.76±.02	.71±.01	.69±.01	.70±.01	.68±.01	.73±.00	.70±.01	.73±.02	.69±.02	.61±.01	.58±.00
FinEvent (2022)	.69±.01	.67±.01	.90±.01	.90±.00	.83±.00	.82±.00	.74±.01	.74±.00	.66±.01	.66±.00	.80±.00	.78±.00	.74±.01	.64±.01
GraphHAM (2024)	.53±.00	.43±.00	.95±.01	.93±.00	.36±.00	.33±.00	.54±.00	.52±.00	.70±.00	.64±.00	.77±.01	.67±.00	.61±.00	.54±.00
RPLM (2024)	.83±.01	.82±.01	.93±.01	.93±.01	.86±.01	.86±.01	.83±.00	.82±.01	.91±.00	.90±.01	.82±.01	.80±.01	.74±.02	.71±.01
$LSED_{llama3.1}$	.98±.01	.98±.01	.99±.01	.99±.01	.98±.02	.97±.03	.98±.01	.97±.01	.99±.01	.98±.01	.97±.01	.96±.01	.96±.02	.96±.02
$LSED_{Qwen2.5}$	.98±.00	**.98±.00**	**1.0±.00**	**1.0±.01**	**.98±.00**	**.98±.00**	**.99±.00**	**.99±.01**	**1.0±.00**	**1.0±.00**	**.98±.01**	.97±.02	**.99±.00**	**.99±.00**
$LSED_{Gemma2}$	**.99±.00**	**.98±.00**	1.0±.01	.99±.01	.98±.01	.98±.01	.96±.00	.95±.00	.99±.00	.99±.00	**.98±.01**	**.97±.01**	.98±.01	.98±.01
Improvement	16% ↑	16% ↑	7% ↑	7% ↑	12% ↑	12% ↑	16% ↑	17% ↑	9% ↑	10% ↑	16% ↑	17% ↑	25% ↑	28% ↑

Table 9: Online scenario evaluation for Events2012 (best results in **bold**).

Blocks (#events)	M_1 (22)		M_2 (19)		M_3 (15)		M_4 (19)		M_5 (27)		M_6 (26)		M_7 (23)		M_8 (25)	
Metrics	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI
TwitterLDA (2011)	.20±.00	.19±.00	.09±.00	.06±.00	.13±.00	.11±.00	.10±.00	.08±.00	.24±.00	.20±.00	.22±.00	.19±.00	.12±.00	.10±.00	.24±.00	.20±.00
Word2Vec (2013)	.22±.00	.21±.00	.22±.00	.21±.00	.25±.00	.23±.00	.28±.00	.27±.00	.48±.00	.46±.00	.33±.00	.31±.00	.35±.00	.33±.00	.37±.00	.34±.00
BERT (2018)	.32±.00	.28±.00	.32±.00	.31±.00	.31±.00	.32±.00	.33±.00	.30±.00	.47±.00	.44±.00	.36±.00	.33±.00	.41±.00	.36±.00	.44±.00	.38±.00
PP-GCN (2019)	.49±.01	.48±.00	.45±.00	.44±.02	.56±.03	.55±.03	.54±.03	.54±.04	.54±.02	.53±.02	.52±.02	.50±.03	.56±.04	.55±.04	.56±.03	.55±.02
EventX (2020)	.34±.00	.11±.00	.37±.00	.12±.00	.37±.00	.11±.00	.39±.00	.14±.00	.53±.00	.24±.00	.44±.00	.15±.00	.41±.00	.12±.00	.54±.00	.21±.00
KPGNN (2021)	.54±.01	.54±.01	.56±.02	.55±.01	.52±.03	.55±.02	.55±.01	.55±.01	.58±.02	57±.01	.59±.03	.57±.02	.63±.02	.61±.02	.58±.02	.57±.02
QSGNN (2022)	.57±.01	.56±.01	.58±.01	.57±.01	.57±.01	.56±.02	.58±.03	.57±.03	.61±.02	.59±.01	.60±.01	.59±.01	.64±.01	.63±.01	.57±.02	.55±.02
FinEvent (2022)	.70±.01	.70±.00	.74±.01	.74±.00	.64±.00	.64±.00	.72±.01	.67±.00	.64±.00	.64±.00	.67±.00	.57±.00	.78±.01	.67±.00	.66±.02	.62±.00
GraphHAM (2024)	.77±.01	.76±.00	.75±.00	.75±.00	.73±.00	.72±.00	.74±.01	.72±.01	.79±.02	.77±.00	.76±.00	.73±.00	.72±.00	.71±.00	.75±.00	.71±.00
RPLM (2024)	.89±.01	.89±.01	.84±.01	.84±.00	.76±.01	.77±.01	.75±.00	.75±.02	.67±.01	.67±.01	.73±.02	.72±.01	.88±.00	.87±.01	.77±.01	.76±.02
$LSED_{Llama3.1}$	.96±.02	.96±.02	**.98±.00**	**.98±.00**	.95±.01	.94±.01	.95±.00	.94±.00	.97±.02	.97±.02	.97±.00	.97±.00	.94±.00	.93±.00	.97±.01	.97±.01
$LSED_{Qwen2.5}$	**.97±.01**	**.97±.01**	.96±.02	.96±.02	**.99±.01**	**.99±.01**	.94±.00	.94±.00	**.98±.00**	**.98±.00**	**.99±.01**	**.99±.01**	**.97±.02**	**.96±.02**	.98±.02	.98±.02
$LSED_{Gemma2}$	.95±.00	.95±.00	.94±.00	.93±.00	.96±.00	.96±.00	**.97±.01**	**.97±.01**	.97±.01	.97±.01	.98±.00	.98±.00	.95±.00	.95±.00	**.99±.01**	**.99±.01**
Improvement	8% ↑	8% ↑	14% ↑	14% ↑	23% ↑	22% ↑	22% ↑	22% ↑	19% ↑	21% ↑	23% ↑	26% ↑	9% ↑	9% ↑	22% ↑	23% ↑

Blocks (#events)	M_9 (31)		M_{10} (32)		M_{11} (31)		M_{12} (29)		M_{13} (28)		M_{14} (26)		M_{15} (25)		M_{16} (14)	
Metrics	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI
TwitterLDA (2011)	.16±.00	.12±.00	.17±.00	.11±.00	.22±.00	.18±.00	.28±.00	.25±.00	.19±.00	.17±.00	.24±.00	.21±.00	.33±.00	.30±.00	.07±.00	.02±.00
Word2Vec (2013)	.33±.00	.30±.00	.46±.00	.42±.00	.41±.00	.38±.00	.40±.00	.37±.00	.22±.00	.20±.00	.36±.00	.34±.00	.41±.00	.38±.00	.28±.00	.25±.00
BERT (2018)	.38±.00	.28±.00	.42±.00	.35±.00	.45±.00	.34±.00	.48±.00	.44±.00	.31±.00	.26±.00	.43±.00	.40±.00	.39±.00	.39±.00	.34±.00	.27±.00
PP-GCN (2019)	.54±.02	.48±.03	.56±.06	.55±.04	.59±.03	.57±.02	.60±.02	.58±.02	.61±.01	.59±.02	.60±.02	.59±.01	.57±.03	.55±.03	.53±.02	.52±.02
EventX (2020)	.45±.00	.16±.00	.52±.00	.19±.00	.48±.00	.18±.00	.51±.00	.20±.00	.44±.00	.15±.00	.52±.00	.22±.00	.49±.00	.22±.00	.39±.00	.10±.00
KPGNN (2021)	.48±.02	.46±.02	.57±.01	.56±.02	.54±.01	.53±.01	.55±.04	.56±.02	.60±.02	.60±.02	.66±.01	.65±.00	.60±.01	.58±.02	.52±.02	.50±.01
QSGNN (2022)	.52±.02	.46±.02	.60±.01	.58±.01	.60±.01	.59±.02	.61±.02	.59±.02	.59±.04	.58±.03	.68±.02	.67±.02	.63±.02	.61±.00	.51±.03	.50±.03
FinEvent (2022)	.57±.01	.52±.00	.65±.01	.60±.00	.63±.00	.54±.00	.70±.01	.59±.00	.67±.00	.64±.00	.65±.00	.65±.00	.70±.01	.65±.00	.75±.01	.68±.00
GraphHAM (2024)	.71±.00	.68±.00	.54±.00	.49±.00	.76±.01	.74±.01	.80±.00	.78±.00	.59±.00	.57±.00	.75±.00	.72±.00	.85±.01	.84±.01	.81±.00	.78±.00
RPLM (2024)	.58±.02	.57±.02	.77±.01	.75±.01	.68±.01	.68±.01	.77±.01	.77±.01	.68±.00	.67±.00	.70±.00	.70±.01	.73±.01	.71±.01	.95±.02	.94±.02
$LSED_{Llama3.1}$	.95±.00	.95±.00	.97±.01	.97±.02	.98±.00	.98±.00	.98±.00	.97±.00	.94±.01	.94±.01	.95±.00	.94±.00	.98±.01	.98±.01	**1.0±.00**	**1.0±.00**
$LSED_{Qwen2.5}$	**.96±.01**	**.96±.01**	**.98±.00**	**.98±.00**	.98±.01	.98±.01	**.99±.00**	**.99±.00**	**.96±.01**	**.96±.01**	**.98±.00**	**.98±.00**	**.99±.00**	**.99±.00**	.96±.00	.96±.00
$LSED_{Gemma2}$	.95±.02	.95±.02	.97±.01	.96±.01	**.99±.01**	**.99±.01**	.99±.01	.99±.01	.93±.01	.93±.01	.96±.01	.95±.01	**.99±.00**	**.99±.00**	**1.0±.00**	**1.0±.00**
Improvement	25% ↑	28% ↑	21% ↑	23% ↑	23% ↑	25% ↑	19% ↑	21% ↑	28% ↑	29% ↑	23% ↑	26% ↑	14% ↑	15% ↑	5% ↑	6% ↑

Table 10: Online scenario evaluation for Events2018 (best results in **bold**).

Algorithm 1: LSED

Input: Social message blocks $S = \{M_0, ..., M_i\}$; Large Language Model: $LLM(\cdot)$; Pre-trained language model: $PLM(\cdot)$; Time encoder $TIME(\cdot)$; Hyperbolic encoder $H(\cdot)$; Cluster $C(\cdot)$; Softmax: $S(\cdot)$; Window size: w; Ground-truth label set $L = \{l_0, l_1, ..., l_n\}$.

Output: Predicted social event label set $L' = \{l'_0, l'_1, ..., l'_m\}$.

for $m_j \in M_i$ **do**
- A summary of social message $m_j^s \leftarrow LLM(m_j)$
- Vector of summarised message $v_{m_j^s} \leftarrow PLM(m_j^s)$
- Vector of timestamp $v_j^t \leftarrow TIME(t_j)$, where t_j is the timestamp of the social message block m_j.
- Final vector $v_j = ADD(v_{m_j^s}, v_j^t)$.
- **if** $i\%w! = 0$ **then**
  - Social message representations $e_j \leftarrow H(v_j)$;
  - $L' \leftarrow$ Predict label $l'_j \leftarrow C(e_j)$;
- **else**
  - **for** $e \in Epoch$ **do**
    - Social message representations $e_j \leftarrow H(v_j)$;
    - $L' \leftarrow$ Predict label $l'_j = S(log_o^c(e_j))$;
    - Cross-entropy loss $\mathcal{L}_{LSED} = -\sum_{i=0}^{n} l_i log l'_i$;
    - Update parameters