Online Social Event Detection via Filtering Strategy Graph Neural Network

Lifu Chen, Junhua Fang(✉), Pingfu Chao, An Liu, and Pengpeng Zhao

School of Computer Science and Technology, Soochow University, Suzhou, China
20204227034@stu.suda.edu.cn, {jhfang,pfchao,anliu,ppzhao}@suda.edu.cn

Abstract. Nowadays, as a strongly time-dependent data type, the ubiquity of social media messages enables the detection and analysis of real-time events. Through the clustering of online posts concerning their topics, existing methods can quickly identify the current trends on social media, which helps discover marketing opportunities, prevent potential crises, etc. However, due to the diversity of social network users, the performance of current approaches is significantly affected by the long tail of random topics, which should be regarded as outliers in a clustering problem. Besides, current models are weak in detecting events that last for multiple days, which is common in real-world scenarios. Therefore, we propose the FS-GNN, a graph neural network based on a filtering strategy, for incremental social event detection in data streams. Our method uses heterogeneous information networks (HINs) to construct a social message graph, and we propose a centrality-based scoring mechanism to grade and filter noisy data before clustering. In addition, a message complement window is introduced to connect the same topic mentioned across multiple days for better clustering accuracy. Extensive experimental results demonstrate the superiority of FS-GNN over multiple baselines in both offline and online scenarios.

Keywords: Social event detection · Online clustering · Graph neural network

1 Introduction

The recent surge of social media platforms, like Twitter, Facebook and Weibo, encourages more users to share their daily lives publicly, which also causes an explosive increase of social messaging data. Such User-Generated Content (UGC) can be utilized in various data mining tasks. As one of its major applications, in recent years, researchers start to analyse online social media posts to detect real-time events [11,20].

Existing works represent a social message by the keywords it mentions, so the task of detecting events in a social message stream can be described as finding clusters of messages mentioning the same set of topics. As the traditional

T. Di Noia et al. (Eds.): ICWE 2022, LNCS 13362, pp. 66–81, 2022.
https://doi.org/10.1007/978-3-031-09917-5_5

clustering algorithms only extract topics from formally-written articles and news, there is a strong need for a new method to detect events from highly-diversified social message streams. Recent research shows that the Graph Neural Network (GNN) has strong capabilities in aggregating the structure and semantics of data, and the GNN-based social message analytic framework has shown its good performance in event detection [13]. However, two issues remain unsolved for event detection: 1) as the social media users are highly-diversified, most of their posts focus on random topics that are irrelevant to major ongoing events, so these 'noisy' posts will reduce the accuracy of our event detection task; 2) the influence of an event to the social media is usually long-lasting and scattered sometimes. For example, before a contagious disease outbreak is identified, multiple scattered posts mentioning related symptoms have appeared on social media. Due to the lack of long-term incremental clustering, the existing method cannot detect such signs before the outbreak spike kicks in, making the event detection less effective.

In this paper, we propose the FS-GNN, a model for incremental social event detection in social message streams. In our model, we use heterogeneous information networks (HINs) to map social communication elements to a unified heterogeneous graph, then we leverage GNNs to extract knowledge from the semantic and structural information contained in the social graph. Besides, we provide an effective solution to the aforementioned issues: 1) To solve the data noise problem, we propose a centrality-based scoring mechanism to score the nodes in the social graph based on their density. We then retain the nodes with higher scores, which imply high relevance to the event clustering problem. 2) To achieve better performance in long-term event detection, we design a message complement window in the incremental clustering phase to capture the correlation of messages across days. Overall, the life cycle of our model consists of an initial phase detecting new message events, and an incremental phase, which extends the knowledge of the model by resuming the training process. In the incremental phase, we employ a triplet loss set that contrasts positive message pairs with corresponding negative pairs. Meanwhile, we introduce an additional global-local pairs [1,7,15] loss function to fit the incremental context and incorporate graph structure information.

We conduct extensive experiments on the large-scale Twitter corpus [12] and the event detection corpus [17] which are publicly available. The empirical results show that FS-GNN achieves better performance over multiple baselines. Overall, we summarize our main contributions as follows:

- We design a new incremental social event detection model based on filtering strategies. Our proposed model can effectively filter data noise, and enhance the accuracy of the model for social event detection.
- We define an information complement window to capture the long-term correlation of message flows. Experiments demonstrate that our proposed information-completion scheme has higher clustering accuracy in incremental scenarios.
- Extensive experimental results on large-scale real-world datasets show that our proposed model is effective and stable in an incremental environment while guaranteeing clustering effectiveness.

2 Related Work

The social events discussed in this paper are coarse-grained, which include general events and emergent events. As a social message detection problem, existing solutions mainly fall into the following three categories:

Textual Information Extraction. As social messages are purely text messages, it is natural and common to detect social events through textual information extraction. For example, Liu et al. [10] use Word2Vec to merge identical entities and similar phrases to automatically detect events in multilingual-mixed long text streams. To further investigate deeper semantic meanings, Liu et al. [9] extract textual information into a more fine-grained representation and used this to construct story trees to represent the categories of subsequent events. However, on bigger social media platforms (such as Twitter), the detection of multilingual mixed social events becomes a major challenge. To solve this question, Hu et al. [8] use a probability distribution to represent semantic categories of multilingual mixed text divided into a fixed time slice window to detect events online. These methods focus on how to represent textual semantic information quantitatively using data structures, models or mathematical methods, but they rarely consider the data noise involved, which can strongly affect the performance of social event detection [14].

Structural Information Extraction. In addition to text content, the diversity of social connections between users and their non-textual properties have become a new source for social event detection. Specifically, some existing works focus on location, Zhang et al. [19] capture a novel authority metric of geothematic correlation between messages to extract high-quality events. Xing et al. [18] further consider the tags in the messages, they use probabilistic models to generate distributions of tags and topic text and propose the MGe-LDA model for discovering and representing sub-events. Chen et al. [3] utilize the message forwarding information (like the retweet in Twitter) to model the flow of information propagation, and they propose an RL-LDA model to discover the correlation between forwarding behaviour and events. Considering that the mentions among social messages speed up the spread of information, Adrien et al. [6] use the frequency of mentions to detect important events and estimate their level of impact on the population. Structural information can be seen as an additional attribute in social networks, but they also suffer from data noise. At the same time, the time-sensitive nature of structural information poses a greater challenge for the online detection of social events.

Hybrid Information Extraction. Some recent works consider both the textual and structural information in feature extraction and utilize deep learning models, like graph neural networks, to mine the social data. It aggregates information from the input graph data and acquires a vectorized representation of

the nodes. Cao et al. [2] is selective in retaining old knowledge in GNN to achieve online clustering. Qiu et al. [14] define a comprehensive similarity index to filter the noisy node in the social network. Cui et al. [4] use the subject labels of events to construct multi-view text messages and design an attention neural network-based model to fuse event features between multiple views. Graves et al. [5] use BiLSTM to capture the relationship between messages better. GNNs are good at aggregating semantic information in social messages, but they can not satisfy the online detection of social events.

Overall, none of the three categories we have mentioned takes into account both the data noise and the real-time nature of social messaging, which are the main focus of our paper.

3 Preliminaries

In this section, we will introduce the problem definition and the framework overview.

3.1 Problem Definition

Definition 3.1 (Social Message Stream). A social message stream S_m is a series of social messages arriving continuously and chronologically, defined as $S_m = \{m_1, m_2, ..., m_i\}$, where m_i represents a social message arriving at time i.

Definition 3.2 (Social Events). A social event set E is a set containing all social events mentioned in a social message stream, which is defined as $E = e_1, e_2, ..., e_k$ where e_k represents a social event.

Here, we specify that each social message m_a is only associated with no more than one social event e_b through mentioning.

Definition 3.3 (Representation Learning Model). Given a social message m_i, the representation learning converts each social message into a vector representing the semantic information about itself and its neighbours, depicted as $f_v(m_i, \sigma) = r_i$, where r_i denotes the vector representation of message m_i, and σ represents the set of parameters of the model f_v.

Definition 3.4 (Social Event Detection). The social event detection problem aims to identify social events mentioned by messages in a social message stream via a representation learning model. The problem can be formalised as $f_s(f_v, \theta) = E$, where f_v denotes representation learning model, θ is the parameters of f_s and E represents the detected social event set.

According to Definition 3.4, to design a better representation learning model $f_v(m_i, \sigma) = r_i$ for event detection, the main objectives of this paper are to improve the quality of m_i, i.e. social message data, as well as optimizing parameter σ to get correct results through complete information. Therefore, we propose a model named FS-GNN to address them, respectively, which will be introduced in the next section.

3.2 Framework Overview

According to Algorithm 1, our framework is divided into two parts. In part of the initial phase, we use a social message stream to construct a social graph and then clean them with our filtering strategy. And next, we use the purified graph data to train an initial model. In part of the incremental phase, we use the remaining data to construct the message blocks for model retraining. Specifically, we use the information complement window to supplement the link between cross-days. Finally, we use the vectorized representation of the model output to cluster the social events.

Algorithm 1. FS-GNN

Input: A social stream $S_m = m_0, m_1, m_2, ..., m_i,$
the number of layers L, window size w, the number of mini-batches B.
Output: Sets of social events: $e_0, e_1, e_2, ...$

/* **Start the initial phase** */
$m_{in} = 0$
for $j = 0, 1, 2, ..., 6$ **do**
  $m_{in} += j$
end for
$G_{in} \leftarrow$ Create pure initial social message graph(m_{in})
$G_f \leftarrow Flitering(G_{in})$
for $L = 1, 2, ...$ **do**
  $h_{m_i}^{(l)} = concat(h_{m_i}^{(l)} \bigoplus Aggregator_{\forall m_j \in N_{ei}(m_i)}(Extractor(h_{m_i}^{(l-1)}))$
  $e_j \leftarrow$ Clustering by h_{m_i}
end for
/* **Start the incremental phase** */
$k = 0$
for $i = 7, 8, 9, ...$ **do**
  $G_i \leftarrow$ Create incremental social message graph(m_i)
  $k += 1$
  if $i \geq 7$ **then**
    $G_i \leftarrow$ Delete obsolete nodes
    save the obsolete nodes n_i from G_i
  end if
  if $k\%w == 0 \& k! = 0$ **then**
    Completes the information of n_i to G_{i+2}
    for $L = 1, 2, ...$ **do**
      $h_{m_i}^{(l)} = concat(h_{m_i}^{(l)} \bigoplus Aggregator_{\forall m_j \in N_{ei}(m_i)}(Extractor(h_{m_i}^{(l-1)}))$
    end for
    $L_t = \sum_{t \in T} \max\left\{\|V(m_i) - V(m_i+)\|_2^2 - \|V(m_i) - V(m_i-)\|_2^2 + a, 0\right\}$
    $L_p = \frac{1}{N} \sum_i^N \left(\log P(h_{m_i}, s) + \log\left(1 - P\left(\widetilde{h}_{m_l}, s\right)\right)\right)$
    $e_i \leftarrow$ Clustering by h_{m_i}
  end if
end for
return the social event sets E

4 Methodology

As mentioned in Sect. 3.2, our FS-GNN method aims at improving the accuracy of event detection by (1) proposing the centrality-based filtering strategy in the initial phase and (2) introducing the information complement window to maintain the long-term message links in the incremental phase. In this section, we elaborate on the details of the two phases, respectively.

4.1 The Initial Phase

Figure 1 demonstrates the workflow of the initial phase. According to the figure, this phase consists of four steps: **Step 1.** A homogeneous message graph is first constructed from input social messages by modelling and extracting the correlations between them. (As shown in Fig. 1, circles represent messages, squares represent common words between message streams, diamonds represent common locations, triangles represent common friends, etc.) **Step 2.** A node filtering process is applied to the constructed graph to remove nodes that are noisy to the clustering task. **Step 3.** The filtered graph is used as input to a L-layer neural network to learn the representations. **Step 4.** The representations are finally grouped into multiple clusters representing the detected social events. Facing massive social message streams with complex attributes, our method is expected to 1) make full use of the social message data features to precisely model the correlation between messages via the message graph (Step 1), and 2) further improve the quality of the message graph by removing messages that are

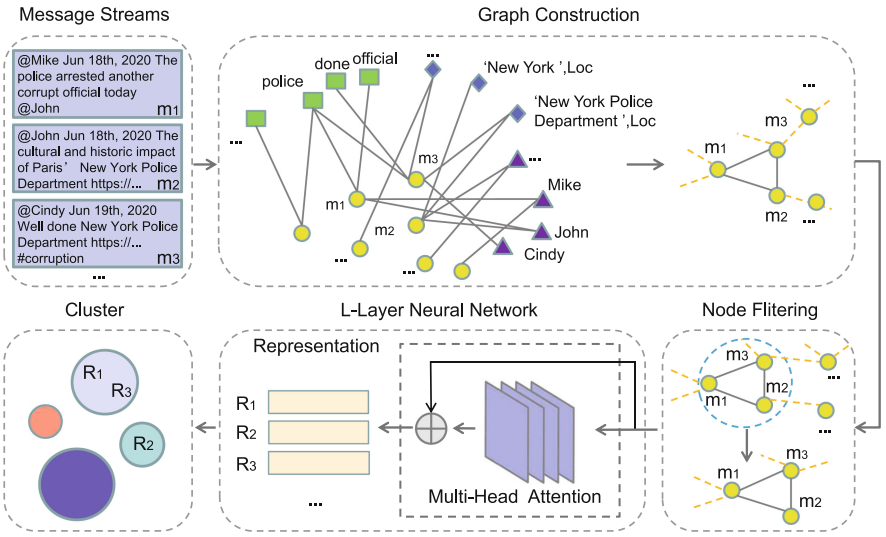


Fig. 1. Workflow of the initial phase

irrelevant to the event detection task (Step 2). To this end, we leverage the Heterogeneous Information Network (HIN) to construct the initial message graph and then filter the constructed graph by our proposed centrality-based filtering strategy.

4.1.1 Heterogeneous Social Message Modeling

The HIN is a graph that contains multiple types of nodes, which help us to represent the properties and structure of the message data better. Figure 1 shows an example, different elements in a message, such as words, locations and users, are represented as different types of nodes in the graph. And an edge between two message nodes is added when they share any element. Here, we label the node messages, words, entities and users as m, w, e, u, respectively.

Different from previous heterogeneous GNNs, FS-GNN focuses on learning correlations between messages. We use a specific method to map the original graph to a homogeneous message graph to retain only the message nodes. The mapping process is expressed as follows:

$$A_{i,j} = \min \left\{ \left[\sum_{k} W_{mk} \cdot W_{mk}^{T} \right]_{i,j}, 1 \right\}, k \in \{w, e, u\}, \tag{1}$$

where $A \in \{0,1\}^{N \times N}$ is the adjacency matrix of the homogeneous message graph, N is the total number of messages in the graph. i, j denotes the matrix element in row i column j and k denotes the node type. W_{mk} is a submatrix of the adjacency matrix of the heterogeneous message graph containing rows of type m and columns of type k. In general, if messages m_i and m_j both connect to some nodes of type k, $\left[W_{mk} \cdot W_{mk}^{T}\right]_{i,j}$ will be no less than 1 and the $A_{i,j}$ will be equal to 1.

4.1.2 Filtering Strategy

The research on network centrality originates from the study of social networks and is described as the degree to which a node is central to the network. In our task, as an event is usually followed by an extensive number of social messages discussing relevant topics, whereas other random posts are less likely to form dense social connections, we use the concept of centrality to describe the density of nodes for node filtering. Hence, we define I as the lower bound within the interval where the degree of the dense node lies. Based on this, the filtering strategy can be defined as follows:

$$I = K \cdot \frac{D_{\text{all}}}{N_{\text{node}}}, \tag{2}$$

where K represents the density factor, D_{all} denotes the total number of node degrees in the social message graph, and N_{node} represents the total number of nodes during a week. To decide an appropriate density factor K, we introduce the three most common types of centrality measures, shown in Table 1:

Table 1. Three types of centrality.

Centrality	Degree centrality	Closeness centrality	Betweenness centrality
Formulas	$D(N_i) = \frac{\sum x_{ij}}{D-1}$	$C(n_i) = \frac{1}{\sum d(n_i, n_j)}$	$B(n_i) = \sum_{s \neq t \in N} \frac{\sigma_{st}(n_i)}{\sigma_{st}}$

Degree Centrality: The degree of the node as a proportion of the total degree of all points. In the equation, D denotes the degree of all nodes in the network, x_{ij} indicates of connections of the node i to other nodes j, i.e. the number of degrees. **Closeness Centrality:** the inverse of the average of the distances from the node to all other nodes; In the equation, $d(n_i, n_j)$ denotes the shortest path of node n_i to node n_j. **Betweenness Centrality:** the ratio of the shortest path through a node and connecting two other nodes to the total number of shortest paths between those two nodes. In the equation, $\sigma_{st}(n_i)$ denotes the total number of paths passing through the node n_i in the shortest path between node s and node t, and σ_{st} denotes the total number of shortest paths between node s and node t.

4.2 The Incremental Phase

Figure 2 illustrates the procedure in the incremental phase. This phase can be divided into three steps: **Step 1.** as time goes, new messages (m_4 in the figure) are updated to the message graph as well as obsoleted messages (m_2) to be removed. **Step 2.** An information complement window is designed to aggregate the obsoleted node m_{i-2} at t_{i-2} time with message graph. **Step 3.** The new graph is input into the model for retraining and add a new event independently represented by R_4. Facing missing information between cross-day messages, our method is expected to: 1) capture information across days to complete the new messages (Step 2), and 2) learn to detect unknown events more accurately. To this end, we design an information complement window to supplement the message correlations across days, and then add two loss functions to detect events more precisely.

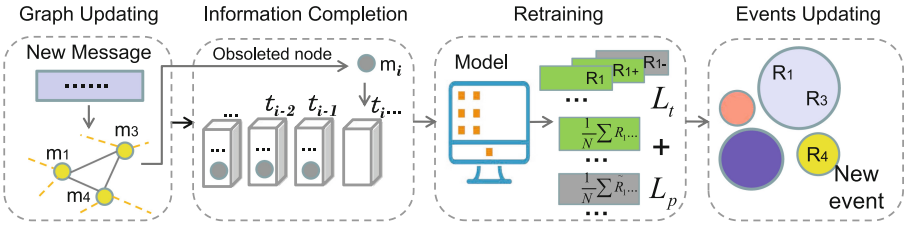


Fig. 2. Workflow of the incremental phase

4.2.1 Representation Learning

To observe the interaction between the preserved nodes and new message nodes, we use GNN to learn the message representation: $E_G : \mathbb{R}^{N\times d} \times \{0,1\}^{N\times N} \rightarrow \mathbb{R}^{N\times d^{'}}$ and $E_G(X,A) = \{h_{m_i} \in \mathbb{R}^{d^{'}} | 1 \leq i \leq N\}$ (Line 23–25 in Alg. 1). It is formalized as:

$$h_{m_i}^{(l)} = \text{concat}\left(h_{m_i}^{(l)} \oplus \text{ Aggregator }_{\forall m_j \in \text{ Nei }(m_i)}\left(\text{ Extractor }\left(h_{m_i}^{(l-1)}\right)\right)\right), \quad (3)$$

where $h_{m_i}^{l}$ is the representation of m_i at the l-th level in the GNN. $Nei(m_i)$ represents the set of neighbours of m_i obtained from the adjacency matrix A, and $\oplus$ represents the aggregation of the information contained in its two operands. $Concat(\cdot)$ represents a multi-headed cascade. $Extractor(\cdot)$ and $Aggregator(\cdot)$ denote the extraction of useful information from the representation of adjacent messages and the summarization of adjacent information, respectively. We use $h_{m_i}^{l}$ as the final representation of $h_{m_i}^{l}$, i.e. h_{m_i}. In order to incrementally perform embedding operations on message blocks, we use the graph attention mechanism [16] for neighbourhood information extraction and aggregation.

In order to complement the message correlations across days, we design a message complement window to hold the correlations between nodes joined on the day $i+2$ and nodes on the day i, which is formalized as:

$$A_{i,j}^{'} = \min\left\{\left[\sum_k W_k^i \cdot (W_k^{i+2})^T\right]_{i,j}, 1\right\}, k \in \{w, e, u\} \quad (4)$$

Similar to the mapping process of homogeneous message graph, W_k^i denote the submatrix of an adjacency matrix of node type k that was deleted at time i, W_k^{i+2} denote the submatrix of an adjacency matrix of node type k at time $i+2$. T represents the matrix transpose.

4.2.2 Contrastive Learning

To handle the raw messages, we use a triplet loss function to enable FS-GNN to learn new event categories without limiting the total number of events (Line 26 in Algorithm 1). The function can be expressed as:

$$L_t = \sum_{t\in T} \max\left\{\left\|V(m_i) - V\left(m_i^+\right)\right\|_2^2 - \left\|V(m_i) - V\left(m_i^-\right)\right\|_2^2 + a, 0\right\}, \quad (5)$$

where V denotes the vector representation of the message stream m_i and $\|\cdot\|_2^2$ denotes the Euclidean distance between the two vectors calculated. The triplet satisfies $\left\|V(m_i) - V\left(m_i^+\right)\right\|_2^2 > \left\|V(m_i) - V\left(m_i^-\right)\right\|_2^2$. t denotes a triplet as (m_i, m_i^+, m_i^-), m_i^+ is a positive message (a message stream from the same category), m_i^- is a negative message (a message stream from a different category) and T denotes the set of triplets drawn in the incremental scene.

To address the problem of updating structural information in message graphs, we construct a global-local pair loss function that enables FS-GNN to discover

and preserve features of similar local structures (Line 27 in Algorithm 1). The function can be expressed as:

$$L_p = \frac{1}{N} \sum_{i=1}^{N} \left(\log P \left(h_{m_i'} s \right) + \log \left(1 - P \left(\widetilde{h}_{m_i}, s \right) \right) \right), \tag{6}$$

where L_p represents the global-local pair loss function and $s \in \mathbb{R}^{d'}$ is the average of all message representations. P represents a bi-linear scoring function that outputs the probability of its two operands coming from the joint distribution. L_t and L_p represents the overall loss of the FS-GNN.

5 Experiments

5.1 Datasets

We use two large-scale, publicly available datasets, namely Twitter and MAVEN, for our experiments. Both datasets are preprocessed by the providers to remove duplicate and invalid data for better data quality. Table 2 shows the statistics of datasets.

Twitter [12]: the Twitter dataset contains 68,841 manually tagged tweets associated with 503 event classes, distributed over four weeks.

MAVEN [17]: the MAVEN dataset is a generic domain event detection dataset constructed from Wikipedia documents with no sentence (i.e. messages) associated with multiple event types. The dataset contains 10,242 messages with 154 associated event categories.

Table 2. Statistics of datasets.

Datasets	Nodes	Edges	Event categories
Twitter	68,841	16,358,812	503
MAVEN	10,242	24,238,110	154

5.2 Baselines and Metrics

We compare FS-GNN with general similarity metrics, offline social event detection methods, and incremental methods.

BiLSTM [5]: a model for learning bidirectional, long-term dependencies between words of a message. However, as it only focuses on words, the model does not utilize other attributes in social messages.

EventX [9]: this is a method for online event detection based on text after fine-grained text segmentation. Same as BiLSTM, it also ignores other attributes in social messages.

KPGNN [2]: it is an incremental learning approach based on knowledge preservation and shows good performance in an incremental environment, but it ignores the data noise and the information that exists between a span of days in data.

We also set up comparison experiments under different filtering strategies. To evaluate the performance of all models, we compare the similarity between the message clusters detected by the models and the ground-truth clusters using the following metrics:

NMI (Normalized Mutual Information): it measures the amount of information extracted from the distribution of ground truth labels.

$$NMI(U,V) = \frac{MI(U,V)}{F\{H(U),H(V)\}} \tag{7}$$

AMI (Adjusted Mutual Information): it is similar to NMI that measures the mutual information between two clusters.

$$AMI(U,V) = \frac{MI(U,V) - E\{MI(U,V)\}}{F\{H(U),H(V)\} - E\{MI(U,V)\}} \tag{8}$$

In the two formulas above, U denotes ground-truth vectors, V denotes predicted label vectors. H denotes information entropy. MI represents mutual information of U and V. Generally, F denotes arithmetic mean. E represents the expected value. Both two metrics are based on a mutual information approach to measuring the fit of the data distribution between the clustering results and the actual category information. NMI takes values in the range [0,1] and AMI takes values in the range $[-1, 1]$. A larger value means that the clustering result matches the real situation more closely.

5.3 Experimental Settings

We set the hyperparameters in EventX to their default values mentioned in the original paper [9]. For BiLSTM and FS-GNN, the Table 3 shows our parameters setting. In the initial phase, we use the K-Means clustering to set the total number of classes to the number of true classes after getting the message representation. And in the incremental phase, we use DBSCAN to fit the scenario that a total number of classes is unknown.

Table 3. Parameter settings.

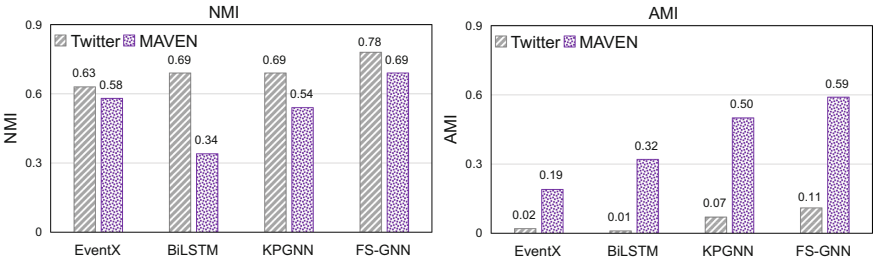
Methods	Layer	Optimizer	Embedding dimension	Retraining window size	Small batch training size	a of the triplet loss function	# of neighbours sampled per message
FS-GNN	2	*Adam*	32	3	2000	3	800
BiLSTM	2	*Adam*	32	–	–	–	–

All experiments are conducted on a 12 cores Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20 GHz and 1×NVIDIA GeForce GTX 1080 Ti GPU.

5.4 Experimental Results

5.4.1 The Initial Phase

Without loss of generality, we partitioned 70%, 20% and 10% of the two datasets for training, testing and validation respectively. Figure 3 shows the result of the experiments conducted by FS-GNN with baselines on both datasets.



(a) The NMI of two datasets. (b) The AMI of two datasets

Fig. 3. The results of initial phase.

Our proposed FS-GNN method has a significant advantage over all types of baselines. Based on the Twitter and MAVEN datasets, FS-GNN outperforms all types of baselines by 13%–102.9% (NMI) and 18%–210.5% (AMI), respectively. However, it is worth noting that the NMI value of BiLSTM is unusually low for the MAVEN dataset. This is because BiLSTM is a text-semantic information-based method and the dimensionality of the MAVEN dataset is only 8 while the Twitter is 16. Low experimental results of BiLSTM on MAVEN dataset due to the sparsity of data in the dataset. EventX and BiLSTM rely on message embedding and focus excessively on the interaction between text messages, thus ignoring the structural information in the social message stream. KPGNN and FS-GNN use a small batch training approach to partition the huge dataset, while the GNN-based approach can better aggregate semantic and structural information in social networks and abstract them into vectorized representations, which gives them an excellent performance on both datasets. Nonetheless, our proposed model FS-GNN employs a filtering strategy to deal with data noise and complements message associations across days, which allowing to achieve better results than any other baselines in the experiments.

We also compare the speed of the four methods during the training phase. In Fig. 4, we can see that EventX has a much larger time overhead than the other methods, as EventX uses a tree structure to divide the hierarchy of events. Each merge, expansion, insertion, etc. of the tree diagram requires a significant time overhead. In contrast, our FS-GNN model uses a filtering strategy to clean the data, and the time overhead in the training phase is smaller than that of the same KPGNN method. In terms of accuracy and time overhead, our proposed model FS-GNN has a very strong performance advantage.

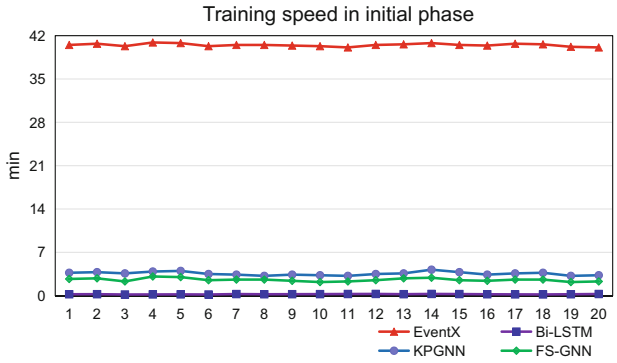


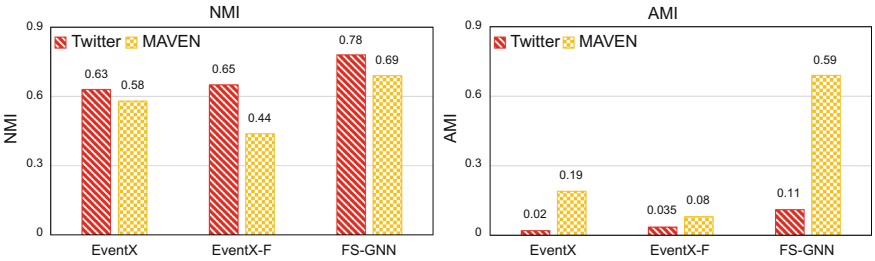
Fig. 4. The speed of 20 epochs in training.

Regarding the effectiveness of filtering strategy, Table 4 shows the improvement when applying different filtering strategies. In particular, "Degree Max" is an extra filtering strategy that indicates the proportion of the maximum degree in datasets to all degrees. Due to the negative result of the experiment with "Closeness Centrality", we also add an experiment with it removed to illustrate the effect of it on filtering strategy. The reason is that this strategy filters almost half of the nodes, including the high-density ones. In the end, the best result of these experiments will be chosen and act on the next stage.

Table 4. Results for different filtering strategies.

Metrics	Degree Max	Degree centrality	Closeness centrality	Betweenness centrality	Average of four strategies	Average of three strategies (without closeness centrality)
Value	0.02	0.12	0.45	0.01	0.15	0.05
NMI	0.73	0.73	0.68	0.75	0.77	**0.78**
AMI	0.62	0.62	0.54	0.65	0.68	**0.69**

To further prove that the performance advantage of our solution does not only come from the filtering, we apply the filtering strategy to the EventX (as other baseline methods are not applicable due to the lack of homogeneous graph structure) Fig. 5 shows the result of the experiments. We can see that the filtering strategies does not improve the performance of EventX consistently, especially in the Maven dataset, as the dataset has less semantic information when the EventX is a text-based semantic information approach. The addition of the filtering strategy filters out the noise in the MAVEN dataset and also reduces the textual semantic information needed for EventX, leading to a decrease in the experimental results. This batch of experiments demonstrates that our proposed FS-GNN still outperforms the other methods even in the same data set. The performance gains of other methods with filtering are not significant, indicating that other parts of FS-GNN also have performance advantages.



(a) The NMI of two datasets. (b) The AMI of two datasets

Fig. 5. The filtering strategy in different methods.

5.4.2 The Incremental Phase

The number of message blocks should be consistent across each baseline. First, we note the message blocks used for initial model training and exclude them. Then, we divide the remaining message streams into a total of 21 blocks to simulate the incremental scenes. Finally, we use diverse methods to detect events in different message blocks and then obtain their results separately for comparison.

Table 5. The NMI of incremental scenarios.

Blocks	M_1	M_2	M_3	M_4	M_5	M_6	M_7
EventX	0.36	0.67	0.66	0.63	0.56	0.69	0.51
BiLSTM	0.24	0.48	0.38	0.40	0.40	0.49	0.32
KPGNN	0.38	**0.78**	0.76	0.67	0.73	0.82	**0.53**
FS-GNN	**0.76**	0.65	**0.80**	**0.75**	**0.74**	**0.85**	0.22
Blocks	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}
EventX	0.71	0.66	0.68	0.64	0.60	0.56	0.57
BiLSTM	0.49	0.43	0.51	0.48	0.38	0.44	0.40
KPGNN	**0.77**	0.72	**0.79**	0.73	0.68	0.68	0.67
FS-GNN	0.49	**0.79**	0.76	**0.78**	**0.71**	**0.80**	**0.80**
Blocks	M_{15}	M_{16}	M_{17}	M_{18}	M_{19}	M_{20}	M_{21}
EventX	0.48	0.61	0.58	0.57	0.60	0.67	0.53
BiLSTM	0.38	0.50	0.48	0.46	0.48	0.44	0.38
KPGNN	0.58	0.78	0.70	0.71	0.72	0.71	0.58
FS-GNN	**0.69**	**0.79**	**0.73**	**0.78**	**0.74**	**0.73**	**0.64**

Table 5 and Table 6 show that our proposed FS-GNN performs better and more consistently in the incremental scenario than the various types of baselines which completes the missing information across days. The experimental results of BiLSTM are very poor. Due to the fact that when we divide the datasets

by days, we reduce the density of semantic information, which leads to poor experimental results. However, as we can see in the two tables, the blocks M_7 and M_8 exhibit very low precision in two evaluation metrics. The reason is explained as: when the information complement window is working in the incremental phase, there is less information across days between the two message blocks. But our information complement window mistakenly added the unnecessary nodes as missing information into the message graph, which causes a loss of accuracy.

Table 6. The AMI of incremental scenarios

Blocks	M_1	M_2	M_3	M_4	M_5	M_6	M_7
EventX	0.06	0.28	0.17	0.18	0.13	0.27	0.13
BiLSTM	0.12	0.40	0.31	0.30	0.32	0.36	0.20
KPGNN	0.36	**0.74**	0.73	0.64	0.70	0.78	**0.52**
FS-GNN	**0.76**	0.63	**0.76**	**0.72**	**0.71**	**0.79**	0.17
Blocks	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}
EventX	0.20	0.18	0.24	0.23	0.15	0.16	0.13
BiLSTM	0.34	0.32	0.38	0.36	0.30	0.31	0.34
KPGNN	**0.75**	0.71	**0.77**	0.71	0.65	0.65	0.67
FS-GNN	0.43	**0.74**	0.74	**0.77**	**0.66**	**0.76**	**0.79**
Blocks	M_{15}	M_{16}	M_{17}	M_{18}	M_{19}	M_{20}	M_{21}
EventX	0.06	0.19	0.18	0.16	0.16	0.17	0.09
BiLSTM	0.26	0.40	0.34	0.34	0.35	0.33	0.31
KPGNN	0.54	0.72	0.70	0.69	0.72	0.67	0.54
FS-GNN	**0.66**	**0.75**	**0.73**	**0.75**	**0.74**	**0.71**	**0.61**

6 Conclusions

In this paper, we proposed a new model named FS-GNN which combines a filtering strategy with GNN that incorporates the rich semantic and structural information in social message streams and filters harmful noise nodes. In incremental scenarios, the introduction of the information complement window can more effectively compensate for the lack of information between them, achieving good detection results while ensuring the stability of event detection. With the filtering strategy and information completion window in different stages, our proposed model FS-GNN achieved better performance over multiple baselines. The better values shown in metrics NMI and AMI at different stages demonstrated the superiority of FS-GNN compared to baseline.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under the grant (No. 61802273, 62102277), Postdoctoral Science

Foundation of China (No. 2020M681529), Natural Science Foundation of Jiangsu Province (BK20210703), China Science and Technology Plan Project of Suzhou (No. SYG202139), Postgraduate Research & Practice Innovation Program of Jiangsu Province (SJCX2_11342).

References

1. Belghazi, M.I., Baratin, A.: MINE: mutual information neural estimation. CoRR (2018)
2. Cao, Y., Peng, H., Wu, J.: Knowledge-preserving incremental social event detection via heterogeneous GNNs. In: WWW, pp. 3383–3395 (2021)
3. Chen, X., Zhou, X., Sellis, T., Li, X.: Social event detection with retweeting behavior correlation. Expert Syst. Appl. **114**, 516–523 (2018)
4. Cui, W., Du, J., Wang, D., Kou, F., Xue, Z.: MVGAN: multi-view graph attention network for social event detection. ACM Trans. Intell. Syst. Technol. (TIST) **12**(3), 1–24 (2021)
5. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**(5–6), 602–610 (2005)
6. Guille, A., Favre, C.: Event detection, tracking, and visualization in twitter: a mention-anomaly-based approach. CoRR (2015)
7. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S.: Learning deep representations by mutual information estimation and maximization. In: ICLR (2019)
8. Hu, L., Zhang, B., Hou, L., Li, J.: Adaptive online event detection in news streams. Knowl.-Based Syst. **138**, 105–112 (2017)
9. Liu, B., Han, F.X., Niu, D.: Story forest: extracting events and telling stories from breaking news. ACM (TKDD) **14**(3), 1–28 (2020)
10. Liu, Y., Peng, H., Li, J., Song, Y., Li, X.: Event detection and evolution in multilingual social streams. Front. Comput. Sci. **14**(5), 1–15 (2020). https://doi.org/10.1007/s11704-019-8201-6
11. Liu, Z., Yang, Y., Huang, Z., Shen, F., Zhang, D., Shen, H.T.: Event early embedding: predicting event volume dynamics at early stage, pp. 997–1000 (2017)
12. McMinn, A.J., Moshfeghi, Y., Jose, J.M.: Building a large-scale corpus for evaluating event detection on Twitter. In: ACM CIKM, pp. 409–418 (2013)
13. Peng, H., Li, J., Gong, Q.: Fine-grained event categorization with heterogeneous graph convolutional networks. In: IJCAI, pp. 3238–3245 (2019)
14. Qiu, Z., Hu, W., Wu, J., Tang, Z., Jia, X.: Noise-resilient similarity preserving network embedding for social networks. In: IJCAI, pp. 3282–3288 (2019)
15. Velickovic, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. ICLR (Poster) **2**(3), 4 (2019)
16. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J. Mach. Learn. Res. **11**, 2837–2854 (2010)
17. Wang, X., Wang, Z., Han, X.: MAVEN: a massive general domain event detection dataset. In: EMNLP, pp. 1652–1671 (2020)
18. Xing, C., Wang, Y., Liu, J., Huang, Y., Ma, W.: Hashtag-based sub-event discovery using mutually generative LDA in Twitter. In: AAAI, pp. 2666–2672 (2016)
19. Zhang, C., Zhou, G., Yuan, Q.: GeoBurst: real-time local event detection in geo-tagged tweet streams. In: SIGIR, pp. 513–522 (2016)
20. Zhou, X., Chen, L.: Event detection over twitter social media streams. VLDB J. **23**(3), 381–400 (2013). https://doi.org/10.1007/s00778-013-0320-3