A Graph is Worth a Thousand Words: Telling Event Stories using Timeline Summarization Graphs

Jeffery Ansah*
University of South Australia
jeffery.ansah@mymail.unisa.edu.au

Lin Liu
University of South Australia
lin.liu@unisa.edu.au

Wei Kang
University of South Australia
wei.kang@unisa.edu.au

Selasi Kwashie
University of South Australia
selasi.kwashie@mymail.unisa.edu.au

Jixue Liu
University of South Australia
jixue.liu@unisa.edu.au

Jiuyong Li
University of South Australia
jiuyong.li@unisa.edu.au

ABSTRACT

Story timeline summarization is widely used by analysts, law enforcement agencies, and policymakers for content presentation, story-telling, and other data-driven decision-making applications. Recent advancements in web technologies have rendered social media sites such as Twitter and Facebook as a viable platform for discovering evolving stories and trending events for story timeline summarization. However, a timeline summarization structure that models complex evolving stories by tracking event evolution to identify different themes of a story and generate a coherent structure that is easy for users to understand is yet to be explored. In this paper, we propose StoryGraph, a novel graph timeline summarization structure that is capable of identifying the different themes of a story. By using high penalty metrics that leverage user network communities, temporal proximity, and the semantic context of the events, we construct coherent paths and generate structural timeline summaries to tell the story of how events evolve over time. We performed experiments on real-world datasets to show the prowess of StoryGraph. StoryGraph outperforms existing models and produces accurate timeline summarizations. As a key finding, we discover that user network communities increase coherence leading to the generation of consistent summary structures.

CCS CONCEPTS

• **Information systems** → **Social networks**; **Summarization**;

KEYWORDS

StoryGraph, Twitter, event evolution, story timeline summarization

ACM Reference Format:
Jeffery Ansah, Lin Liu, Wei Kang, Selasi Kwashie, Jixue Liu, and Jiuyong Li. 2019. A Graph is Worth a Thousand Words: Telling Event Stories using Timeline Summarization Graphs. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313396

*Corresponding author. We acknowledge D2DCRC, Cooperative Research Centres Programme, and the University of South Australia for funding this research. The work is partially supported by ARC Discovery project DP170101306.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313396

1 INTRODUCTION

With the increasing explosion of Web 2.0 technologies, social media sites such as Twitter, Facebook and Weibo have provided a vast continuous supply of diverse information content [2, 4, 14, 16]. These sites provide functionalities for users to share and broadcast information on fashion trends, political debates, breaking news and other issues of interests [8, 17, 18, 28]. However, with the overwhelming amount of information generated from social media, it has become difficult for news analysts, law enforcement agencies and policymakers to obtain a summarative perspective of the large amount of information on trending topics, news stories and events.

A prevailing solution is to organize and present stories in a summarization timeline for easy comprehension [1, 14, 16, 23, 28]. This provides a better understanding of how events evolve throughout the life cycle of a story [8, 16, 23, 28], as well as reveal the flow and evolution of the various themes that make up a story. Such summaries help users to understand the general information of the story in a more concise and logical manner [14, 16].

Motivating Example. Figure 1 is a graph illustrating a story of the popular *"Black Lives Matter"* protest from Aug. to Nov. 2014 in the US. Each distinct colour represents a theme in the story graph. For example: all the green nodes represent related events that evolved in *Ferguson Missouri*; all orange coloured nodes are events related to the *shooting of Michael Brown.* Each node in this graph (i.e. the StoryGraph proposed in this paper) represents an event σ, and a connection between two events represents an event evolution in the story. For example, the path $\sigma_2 \rightarrow \sigma_{16}$ tells the story of the events that emerged from the *Ferguson Missouri protests* leading to the *formation of the Ferguson Committee.* As shown in Fig.1, it is of no doubt that modeling the evolutionary connections between events into a summarization timeline structure can help users to learn the main information quickly. However, developing such comprehensive summarization from the massive volume of social media posts inundated with spam and diverse range of events is like *"finding a needle in a haystack"* [9, 14, 28, 29].

In the attempts to solve timeline summarization problems, Traditional Document Summarization (TDS) [12, 21, 25] has been extensively studied with increasing popularity in many domains [6, 12]. In these lines of research [16, 21, 25, 30], the objective is to provide a curative summary of documents, news reports and articles over a period of time to tell the story of how events evolve. Despite the successes in developing novel TDS techniques to generate timeline summaries, TDS approaches are however not effective in

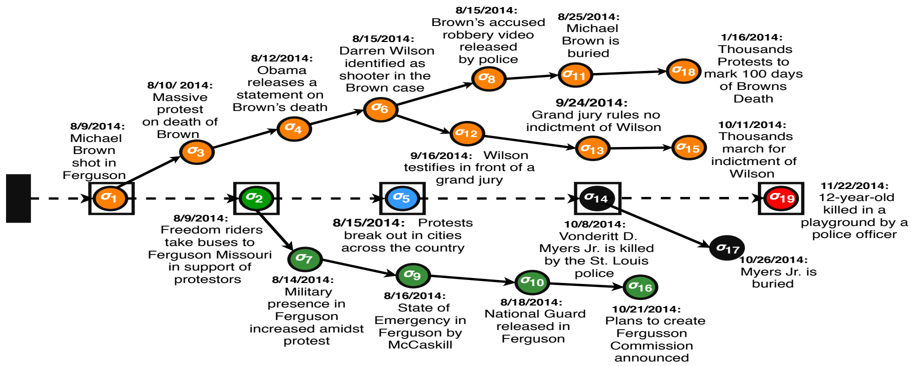

Figure 1: The StoryGraph of the *Black Lives Matter* protests from August to November 2014

handling data from social media sites such as Twitter for timeline summarization due to a number of reasons.

First and foremost, traditional documents such as news reports are well written, proof read by professionals and usually have a writing structure (i.e headlines, paragraphs, body, etc) which provides important cues for summarization [11, 22]. Tweets, on the other hand are highly unstructured. Secondly, tweets are very short, and often do not provide sufficient statistical information and context for robust similarity measures [10]. Furthermore, with the daily massive volume of tweets swamped with spam, it has become increasingly difficult to sort through the hundreds of breaking news events on Twitter and track their evolution to generate timeline summaries [11, 22]. These properties of Twitter data do not only make it difficult to apply TDS techniques, but also manifest a new research challenge for event storyline summarization.

Recent works [9, 14, 28, 29] have attempted to solve the timeline summarization problem using social media. However these models suffer from two main limitations. 1) Their output structures are usually flat timeline structures which are unable to capture the different themes in a larger story. For example, stories like the *"Black Lives Matter"* protest in Fig.1 have different themes that make up the bigger story. A flat timeline structure will connect node σ_6 to σ_7. However, these two event evolutions are totally different and have no logical connection even though they are part of the *"Black Lives Matter"* protest story. 2) The flat timeline models are incapable of presenting the complex evolution relationships between events thus resulting in incomprehensible story summaries devoid of any logical connection between the events.

Developing models to resolve these limitations will aid in the generation of comprehensible story timeline structures. Addressing these problems is non-trivial as it requires the construction of summarization timeline structures with coherent paths, while maintaining logical consistency. This will help users to understand the story in a more concise and logical manner.

To achieve this goal, we propose a novel summarization framework called StoryGraph. Given a set of tweets from the Twitter stream, StoryGraph uses an information propagation based algorithm [4, 5] to detect and extract events with high coherence from online communities in Twitter. Once these events are extracted, we carefully design high penalty metrics using temporal proximity of events, semantic contents, and user community membership. Using Fig.1 as an example, our solution identifies nodes (events) $\sigma_1, \sigma_2, \sigma_5, \sigma_{14}$ and σ_{19} as the starting events of the main themes in the story. StoryGraph further uses coherence and topic drifting metrics to build coherent timeline paths in the summary structure such that each path provides a comprehensive evolution of events related to a specific theme in the story.

While storyline summarization using social media has been attempted in previous works, to the best our knowledge, this is the first work that proposes a comprehensible summarization timeline structure that leverages: user communities, textual content and temporal metrics to model the event evolution relationships and present event summary as a graph. StoryGraph is able to identify the different aspects of a story as the story evolves.

Our main contributions can be summarized as:

- We formally define and formulate a novel graph timeline summarization structure that is capable of identifying and representing the different themes of a story.
- We design high penalty metrics to capture evolving events in communities to construct coherent paths in our summarization structure.
- We develop StoryGraph, a comprehensive summarization framework to generate summaries to tell the story of how events evolve over time. We perform extensive experiments on real-world datasets to show the prowess of our methods.

2 NOTATION AND PROBLEM DEFINITION

In this paper, we focus on timeline summarization of evolving stories using information spreading on Twitter. It is however worth mentioning that the models and formulations can be extended to other social networks such as Sina Weibo and Facebook.

We model the Twitter follower network as a graph $G_N = \langle V_N, \mathbf{E}_N \rangle$, where $V_N = \{u_1, u_2, ..., u_n\}$ represents a set of n Twitter users, and $\mathbf{E}_N \subseteq V_N \times V_N$ is the set of directed edges, and an edge $u_i \rightarrow u_j$, denotes that user u_j is a follower of user u_i on Twitter. As users post tweets, information propagates from one user to another in the network. We represent the tweet stream containing tweets as $\mathcal{D} = \{d_1, d_2, ..., d_m\}$, from which we seek to extract event set $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_k\}$ in a given time period $T_s - T_f$. Each event σ_i can be represented as $(\mathbb{W}_i, \tau_i)$, where $T_s \leq \tau_i \leq T_f$ is the timestamp of σ_i and $\mathbb{W}_i = \{w_1, w_2, ..., w_{|\mathbb{W}_i|}\}$ is the event lexis comprising of a set of words. We aim to construct a story graph to represent the development of the events in Σ.

To clearly present our problem, we firstly define some key concepts that are crucial to the proposed StoryGraph framework.

Story: A story consists of a set of events that revolve around a particular subject or concept during a specific period of time. An example of a story is the *"US presidential election"*, *"Black Lives Matter"* protest, etc. A story usually has different topic themes.
Topic Theme: A topic theme is a specific aspect/subject of a story. For example in Fig. 1, conversations about *Michael Brown* (i.e., $\sigma_1 \rightarrow \sigma_{15}, \sigma_{18}$) is one topic theme in the story graph.
Event: An event is an incident that occurs at a specific time [30], represented as $\sigma_i = (\mathbb{W}_i, \tau_i)$.

Thus, we have the hierarchy: Story →Topic Theme → Event.

More formally, we present the following definitions. Let $Sim(\sigma_i, \sigma_j)$ be a metric that returns the similarity of two events σ_i, σ_j.

*Definition 2.1 (**Related Events**).* Given a user-specified threshold $\lambda \in [0, 1]$ two events σ_i, σ_j are *related* iff. $Sim(\sigma_i, \sigma_j) \geq \lambda$. □

In our story graph, a group of related events form a topic theme. A topic theme can be thought of as the subject of a cluster of related events in the story graph.

*Definition 2.2 (**Topic Theme**).* A topic theme ϑ is a distribution of words generated by a set of *related* events. □

Intuitively, a set of related events $\Sigma' = \{\sigma_1, \sigma_2, ..., \sigma_{k'}\}$ belong to the same topic theme, and share similar words distribution.

*Definition 2.3 (**Topic Theme Node**).* Given $\Sigma' = \{\sigma_1, \sigma_2, ..., \sigma_{k'}\}$ of temporally ordered related events, the *topic theme node* is the first event σ_1 in Σ'. □

A topic theme node initiates a theme in the story graph. From Fig.1, *shooting of Michael Brown* (σ_1), is the topic theme node of the topic theme *Michael Brown*. Also, the other topic theme nodes in the story graph are $\sigma_2, \sigma_5, \sigma_{14}$ and σ_{19}.

*Definition 2.4 (**Topic Theme Path** ψ).* Given a set of related events $\Sigma' = \{\sigma_1, \sigma_2, ..., \sigma_{k'}\}$, a topic theme path is a sequence of directed edges from the topic theme node σ_1 to all events in Σ'. □

Figure 1 shows the three topic theme paths that can be found in the story graph. A topic theme path is said to be *coherent* if every pair of consecutive events are related.

*Definition 2.5 (**Coherent Score** Γ).* The coherence score of a topic theme path is a measure of the theme consistency [25] along its path of related events $\Sigma' = \{\sigma_1, \sigma_2, ..., \sigma_{k'}\}$, and is given by:

$$\Gamma(\sigma_1, \sigma_2, ..., \sigma_{k'}) = \frac{1}{|k'-1|} \sum_{i=1}^{|k'-1|} Sim(\sigma_i, \sigma_{i+1}). \quad \square$$

*Definition 2.6 (**Timeline Path**).* A timeline path is a temporally ordered sequence of topic theme nodes in a story graph. □

A timeline path connects different topic themes in a story graph. Thus, given any evolving story, a timeline path shows the various distinct topic themes that make up the story. For example, in Fig. 1, the timeline path is: $\sigma_1 \rightarrow \sigma_2 \rightarrow \sigma_5 \rightarrow \sigma_{14} \rightarrow \sigma_{19}$.

We proceed to define a story graph in Definition 2.7, following the definition of all relevant concepts.

*Definition 2.7 (**Story Graph**).* A story graph is a directed graph $\mathbf{G} = (\Sigma^G, E)$, with a set of nodes $\Sigma^G \subseteq \Sigma = \{\sigma_1, \sigma_2, ..., \sigma_k\}$ and edge set E. The node set Σ^G of a story graph is such that $\Sigma^G = \Sigma^G_T \cup \Sigma^G_E$, and $\Sigma^G_T \cap \Sigma^G_E = \emptyset$, where Σ^G_T represents the set of topic theme nodes and Σ^G_E represents the set of event evolutions. A directed edge $\sigma_i \rightarrow \sigma_j$ from node σ_i to σ_j indicates a temporal and/or a semantic relationship between two events s.t $\forall\, \sigma_i \rightarrow \sigma_j \in E, \tau_i < \tau_j$. The edge set $E \in \mathbf{G}$ is such that $E \subseteq \Sigma^G \times \Sigma^G \setminus \left(\Sigma^G_E \times \Sigma^G_T\right)$. □

Our goal is to develop a comprehensive and coherence structure that can capture the event evolution of the various themes of a bigger story. Stories contain different topic themes that evolve as new events emerge [14, 16]. For any given story, we expect our solution to identify the main topic themes and capture how the events on each topic theme evolve. For example, given the *"Black Lives Matter"* protest (Refer to Fig.1), we expect our timeline summarization structure to identify the Fergusson protest (node σ_2) as a topic theme node and capture the series of aftermath related events (nodes $\sigma_7, \sigma_9, \sigma_{10}, \sigma_{16}$). The StoryGaph proposed in this paper presents a structural timeline summarization of events evolution on a general story as shown in Fig.1

*Definition 2.8 (**Problem Statement**).* Given a tweet stream $\mathcal{D} = \{d_1, d_2, ..., d_m\}$ containing tweets published by users in the Twitter follower network $G_N = \langle V_N, \mathbf{E}_N \rangle$ over a period $T_s - T_f$, the goal of this paper is to construct a StoryGraph $\mathbf{G} = (\Sigma^G, E)$ using the event set $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_k\}$ extracted from the Tweets such that all the topic theme paths are coherent. □

3 STORY GRAPH CONSTRUCTION

In Fig. 2 is an overview of our StoryGraph framework. The framework accepts tweets over a period and the follower network relationships as input. Recall, that structural timeline summarization involves two main steps [16, 30]: 1) extracting candidate events, and 2) using the events to build a comprehensive summarization structure. To achieve this, StoryGraph constructs propagation trees and leverages bursty tree detection [5] to extract events with high coherent word-distribution from online communities. These events are used as conduits for generating our structural timeline summaries. The next subsections present details of the steps.

3.1 Event Extraction from Communities

The goal here is to extract events from the online Twitter communities. To achieve this goal, we adopt and extend SensorTree [5], an event detection method capable of extracting coherent events using burst from online communities.

Given a set of tweets over a time period as input, we construct propagation trees [4] which effectively capture information diffusion among users in a community. The trees represent communities of online social media users who are discussing events of interest. The nodes of the trees are users in the community and the edges represent the adoption of information among users in a community.

We incorporate user community membership as a feature to increase the confidence of finding topic themes of related events for our StoryGraph construction. This enriches our feature set and helps to overcome limitations posed by relying solely on text (tweet content) [5, 26].

We design a feature vector to help us get our desired output.

*Definition 3.1 (**Event Feature Vector**).* Given an extracted event σ, its event feature vector is a tuple, $\overrightarrow{\sigma} = (U_l, \mathbb{W}, \tau)$, where:

- U_l is a list of the community membership in which the event was detected;
- $\mathbb{W}$ is a word distribution describing the event;
- τ is the timestamp indicating when the event was detected.

□

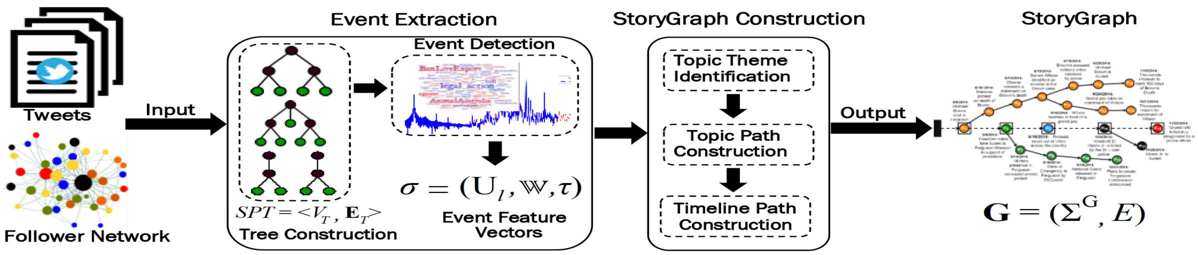

Figure 2: Framework Overview of StoryGraph

The event feature vector $\overrightarrow{\sigma}$ provides a feature rich vector that covers user community relationships, textual and temporal information on extracted events.

Given a set of tweets $\mathcal{D} = \{d_1, d_2, ..., d_m\}$ published during a given time period $T_s - T_f$, we construct semantic propagation trees [5], $SPT = \langle V_T, \mathbf{E}_T \rangle$, where $V_T = \{(u_i, [(c_i, t_i)]) \mid u_i \in V_T, [(c_i, t_i)] = [(c_{i_1}, t_{i_1}), (c_{i_2}, t_{i_2}) \cdots (c_{i_Q}, t_{i_Q})]\}$ represents users u_i, the contents c_i and the timestamps t_i of their posts, and the edge set is $\mathbf{E}_T = \{(u_i, [(c_i, t_i)]) \rightarrow (u_j, [(c_j, t_j)]) \mid u_i \rightarrow u_j \in \mathbf{E}, t_{i_1} \leq t_{j_1}\}$. We set the time window $\top = 60min$, once the burst threshold ϕ is obtained in any tree, an event description model using tensor decomposition [3] outputs word distribution and parses it into $\overrightarrow{\sigma}$. We query the node list of the community in which the event was detected to output the event feature vector $\overrightarrow{\sigma} = (U_l, \mathbb{W}, \tau)$ for that event.

The **EventExtract** block (Algorithm 1, lines 1-10) shows the pseudocode for extracting events and obtaining the event feature vector. Line 9 outputs a candidate event set $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_k\}$ which is represented by their corresponding event feature set vector $\overrightarrow{\sum} = \{\overrightarrow{\sigma_1}, \overrightarrow{\sigma_2}, ..., \overrightarrow{\sigma_k}\}$ to build a story graph. The event feature set vector $\overrightarrow{\sum}$ holds the features vectors of all the events extracted.

Algorithm 1 StoryGraph Algorithm

Input: tweets stream $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$ over $T_s - T_f$; G_N
Output: StoryGraph, $\mathbf{G} = (\Sigma^G, E)$.
1: **EventExtract:** Initialize $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$
2: $\rightarrow$ Sort $\mathcal{D}$ in temporal order
3: $\rightarrow$ Construct $SPT = \langle \boldsymbol{\mathcal{V}}_T, \mathbf{E}_T \rangle$ using extended [5]
4: $\rightarrow$ Vary $\top$ to optimum threshold and compute burst score
5: $\rightarrow$ Extract detected event set $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$
6: **for** $\sigma_i \in \Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ **do**
7: *Generate tensorized feature word-distribution* $\mathbb{W}$ [3]
8: *Extract* $\overrightarrow{\sigma} = (U_l, \mathbb{W}, \tau)$
9: *Query* $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$, $\overrightarrow{\sum} = \{\overrightarrow{\sigma_1}, \overrightarrow{\sigma_2}, \ldots, \overrightarrow{\sigma_k}\}$
10: **end for**
11: **SGConstruct** : Load $\overrightarrow{\sum} = \{\overrightarrow{\sigma_1}, \overrightarrow{\sigma_2}, \ldots, \overrightarrow{\sigma_k}\}$
12: $\rightarrow$ Sort $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ in temporal order s.t. $\tau_i \leq \tau_j$
13: $\rightarrow$ Set $\sigma_1 \in \Sigma$ as first topic theme node $\vartheta \in \Sigma_T^G$
14: **for** $\overrightarrow{\sigma} \in \overrightarrow{\sum} = \{\overrightarrow{\sigma_1}, \overrightarrow{\sigma_2}, \ldots, \overrightarrow{\sigma_k}\}$ **do**
15: *Compute* $F_{Rel}(\overrightarrow{\vartheta}, \overrightarrow{\sigma})$
16: **if** $F_{Rel}(\overrightarrow{\vartheta}, \overrightarrow{\sigma}) > \lambda$
$\rightarrow$ Append and update $\overrightarrow{\sigma}$ with $\overrightarrow{\vartheta}$
$\rightarrow$ *Check* **Node Merge Criterion** *to merge* (Definition 3.2)
$\rightarrow$ Compute *PC* (Eqn. 3) *to append new child node* and update Σ_E^G
17: **else**
Set σ *as new topic theme node* and update Σ_T^G
18: **end for**
19: $\rightarrow$ *Align topic theme nodes* Σ_T^G *on time line path in chronological order*
20: **return** $\mathbf{G} = (\Sigma^G, E)$.

3.2 StoryGraph Construction

The input for our story graph construction phase **SGConstruct** (i.e, Algorithm 1, lines 11-20) is $\overrightarrow{\sum}$. To construct the story graph, we sort the candidate events feature vectors $\overrightarrow{\sigma_1}, \overrightarrow{\sigma_2}, ..., \overrightarrow{\sigma_k}$ using increasing temporal ordering such that $\tau_i \leq \tau_j$. The first event in $\overrightarrow{\sum}$ is chosen as the first topic theme node. For every incoming node, we apply a dual-component metric to detect its topic theme.

Topic Theme Identification: As described earlier, every candidate event has its own event feature vector $\overrightarrow{\sigma}$. To quantify the relatedness between an incoming event and the existing topic theme, we compute the feature relatedness F_{Rel}, between the topic theme feature vector $\overrightarrow{\vartheta}$ and the feature vector of the incoming event $\overrightarrow{\sigma}$. The feature relatedness F_{Rel} is given by:

$$F_{Rel}\left(\overrightarrow{\vartheta}, \overrightarrow{\sigma}\right) = \frac{1}{2}\Big[D_{KL}(\mathbb{W}^{\overrightarrow{\vartheta}}||A) + D_{KL}(\mathbb{W}^{\overrightarrow{\sigma}}||A)\Big] \times \frac{2U_l^c}{U_l^{\overrightarrow{\vartheta}} + U_l^{\overrightarrow{\sigma}}} \quad (1)$$

where $A = \frac{1}{2}(\mathbb{W}^{\overrightarrow{\sigma}} + \mathbb{W}^{\overrightarrow{\vartheta}})$ is the average of the two word distributions, and D_{KL} :

$$D_{KL}(\mathbb{W}||A) = \sum_{w \in |\mathbb{W}|} p(w|\mathbb{W}) \log \frac{p(w|\mathbb{W})}{p(w|A)} \quad (2)$$

is the *Kullback-Leibler* divergence [13] which measures distribution divergence of A from $\mathbb{W}$. U_l^c is the number of the common users in the community user list of $U_l^{\overrightarrow{\vartheta}}$ and $U_l^{\overrightarrow{\sigma}}$.

Equation 1 has two parts. The first part measures the text semantic similarity to determine if an event belongs to a particular topic theme. We utilized the *Jenson-Shannon divergence (JSD)* [19] to measure relatedness directly in $F_{Rel}(\overrightarrow{\vartheta}, \overrightarrow{\sigma})$ instead of $D_{KL}(\overrightarrow{\vartheta}||\overrightarrow{\sigma})$, because the *JSD* is a symmetrical and smoothed. The second part of the equation uses the *Sørensen-Dice coefficient* [27] to measure the community similarity between the incoming event and the topic theme community. We apply this dual-component criteria because the use of only textual content is not always reliable [7, 26]. Using community membership increases the confidence of obtaining coherence. The intuition here is that users in the same community have similar interest and are likely to have discussions on the same topic theme.

Equation 1 (F_{Rel}) is used to determine whether an incoming candidate event should be added to an existing topic theme node or not. If the relatedness $F_{Rel} \geq \lambda$, we can add the event to the related topic theme. Once we determine which topic theme an incoming event belongs to, we update the topic theme feature vector $\overrightarrow{\vartheta}$. Consequently, a topic theme vector becomes the summation of all the related event feature vectors. For example, given an existing topic theme with topic theme feature vector $\overrightarrow{\vartheta_i} = \{(U_l^1, \mathbb{W}_1, \tau_1), ..., (U_l^k, \mathbb{W}_k, \tau_k)\}$. Using F_{Rel}, if an incoming event with feature vector $\overrightarrow{\sigma_j} = (U_l^j, \mathbb{W}_j, \tau_j)$ is found to be related to $\overrightarrow{\vartheta_i}$, then we append and update $\overrightarrow{\vartheta_i}$ with $\overrightarrow{\sigma_j}$. If the $F_{Rel} \geq \lambda$ criterion is not met, we create a new topic theme vector $\overrightarrow{\vartheta_j}$ with $\overrightarrow{\sigma_j}$ as the topic theme node.

Constructing the Topic Theme Path: After a candidate event has been identified to be related to a topic theme, the next step is to connect the candidate event to an existing node in the topic theme path of the related topic theme. One of our objectives is to reduce redundancy or duplication of events presented in the story graph. Thus we seek to merge nodes that discuss the same event. If two events are the same we merge their event feature vector as one event using the following criterion.

*Definition 3.2 (**Node Merge Criterion**).* Given a topic path ψ with a set of nodes $\sigma_1, \sigma_2, ..., \sigma_{k''}$, the merge operation combines an incoming node with an existing event in the topic theme path if: i) $\cos(\overrightarrow{\sigma}_i, \overrightarrow{\sigma}_j) = \frac{\mathbb{W}^i \cdot \mathbb{W}^j}{|\mathbb{W}^i| \cdot |\mathbb{W}^j|} \geq \beta$ (i.e., refers to the same event), where $\cos(\overrightarrow{\sigma}_i, \overrightarrow{\sigma}_j)$ is the similarity between the two word distributions, ii) and the two events occur within $\tau_j - \tau_i < \theta$. ◻

We set $\theta = 24hrs$. If a candidate event does not overlap with an existing node in the topic theme path, we find its parent node by computing the parent-child propensity score, *PC* in Equation 3. We design a matrix that stores the *PC* score. The *PC* score is the pairwise comparison of the incoming candidate event (potential child node) σ_i and the existing nodes (potential parent nodes) in the topic theme path. Given an incoming node σ_i that is related to $\overrightarrow{\vartheta_i}$ (i.e. has passed the F_{Rel} threshold test), and a potential parent node σ_j in the topic theme path, The *PC* score is given by:

$$PC(\sigma_i, \sigma_j) = ComSim \cdot t_d \cdot \Gamma \tag{3}$$

where $ComSim = \frac{|U_l^c|}{|U_l^{\overrightarrow{\sigma_j}}| + |U_l^{\overrightarrow{\sigma_i}}| - |U_l^c|}$ is the community similarity between the event child node and the potential parent node, the temporal distance score t_d: $t_d(\tau_i, \tau_j) = \begin{cases} \frac{1}{(\tau_j - \tau_i)}, & \forall \tau_j - \tau_i > 0 \\ 0, & otherwise \end{cases}$ measures the time proximity between the two nodes. A bigger t_d score increases the possibility that a candidate event is connected to a parent. The third component of the *PC* score is the topic path coherence $\Gamma(\sigma_1, \sigma_2, ..., \sigma_{k'}) = \frac{1}{|k'|} \sum_{i=1}^{|k'|-1} \cos(\overrightarrow{\sigma}_i, \overrightarrow{\sigma}_j)$ measuring the theme consistency [29] along the theme path. For the candidate pair with the highest score in *PC* score matrix, we create an edge from the parent node to the incoming node in the topic theme path. The topic nodes are then aligned in chronological order in the story graph.

4 EXPERIMENTS

We present a detailed description of how the experiments were conducted, evaluated as well as the results obtained in comparison with existing state-of-the-art models.

4.1 Datasets and System Settings

We use four different and geographically diverse real world Twitter data sets for our experiments. Table 1 shows the information on our datasets[1]. With the exception of the FP-Datasets that was provided by the D2DCRC[2], all other datasets used in the work as well as the user follower network were crawled through the Twitter API.

We implemented and conducted all experiments using Python 3.2 and 3.6 on a Windows machine with 8GB Memory and a 64-bit Linux virtual machine with 6GB memory all running on an Intel(R) Core(TM) i5-4310m CPU 2.70Ghz processor.

Table 1: Dataset Description

Data sets	No. of Tweets	Network Size (No. of Nodes)
FP	200,756	12,243,427
USGH	100,000	854,626
Live Exports	3,200,000	8,202,053
USJM	5,200,000	13,202,724

4.2 Gold Standard Records (GSR)

To construct a GSR for evaluation, we search through popular news portals every day and record reported civil unrest events as ground truth. Each GSR entry represents a reported protest event with attributes such as protest event date, headline description, event summary, and news source. An event is recorded if more than three major news outlets cover a story on it. Table 2 shows sample entries in the GSR. As a preprocessing step, we remove stopwords and manually select keywords tokens to obtain a GSR keywords distribution which summarizes the real-word event. For every GSR summary, we expect the summarization models to produce some word token that matches the GSR summary.

4.3 Baseline and Comparison Methods

To evaluate the performance of our methods, we compare our solution to the following models:

1. TwitInfo [20]: explores and tracks events by aggregating tweets about the event using a peak detection algorithm.

2. ConnectingDots [25]: presents a chain structure that finds hidden connections to link coherent news articles together.

3. EventGraph [30]: uses a similarity threshold to calculate connection strength for event pairs to generate an event graph.

4. LDA+Temporal Ordering: We build a single LDA [6] topic model over the datasets and temporally order the events into a timeline chain as a baseline model. This method exemplifies the naive approach to solving the timeline structure summarization problem.

It is worth mentioning that ConnectingDots [25] and EventGraph [30] were originally designed to handle news articles. To level the playground for a fair comparison we build topical tweet clusters using the methods in [24]. These clusters are used as representative news articles. We then follow strictly the authors implementation in their published work and selected the parameter settings that yield the best results. Comparing our methods with ConnectingDots [25] and EventGraph [30] will help us further assess how models that are designed to handle news articles for summarization perform with Twitter data.

4.4 Performance Evaluation

In the first part, we evaluate the quality of the abstractive summaries generated using Recall-Oriented Understudy for Gisting Evaluation (ROGUE) metric [15] which has been demonstrated to be the most consistent with human judgement and effective for evaluating such

[1]data description available at https://github.com/vanslerry/dataDescription

[2]www.d2dcrc.com.au

Table 2: Sample coded GSR event entries by news analysts

Event ID	Date	Headline Description Summary	News Sources
FP-548662	2/05/2017	Freeport to reduce Indonesian Mining Activities	Kompass, Republica, Jarkarta Post
USGM-204201	28/03/2018	Thousands fill the streets to protest new Ghana-US defence deal	Joy News, TV3, GhanaWeb
Live Exports	18/06/2018	Protesters in Adelaide rally against live animals exports	ABC, News, Advertiser

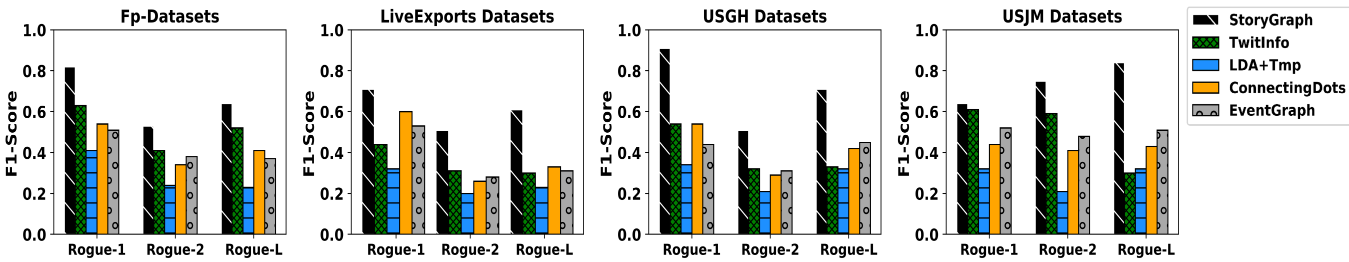

Figure 3: F1-Score on all Datasets

summaries. ROGUE measures the degree of overlap of the model's output with the ground truth. We use ROGUE-1, ROGUE-2 and ROGUE-L. For a fair comparison, we vary the vocabulary $\mathbb{W}$ size of output of the individual models, using $\mathbb{W} = 10, 20, 30, 40, 50$, to reduce bias on models that rely on word frequencies.

In the second part, we evaluate the output of the summarization structures generated by measuring the coherence between nodes in a topic path. This is achieved by computing the distance between the word vectors of the node-pairs which is given by: $C_T(\hat{c_i}, \hat{c_j}) = 1 - \frac{\hat{c_i} \cdot \hat{c_j}}{|\hat{c_i}|^2 + |\hat{c_j}|^2 - \hat{c_i} \cdot \hat{c_j}}$, where are c_i and c_j are the word vector pairs whose coherence is to be determined. This metric also indirectly measures the validity of an edge between two nodes in a path.

5 RESULTS AND DISCUSSIONS

Summary Quality: We compare the ROUGE-1, ROGUE-2 and ROGUE-L metric results of StoryGraph with 4 other comparison models. Due to space limitation, we present the averaged ROGUE metrics performance of all the model across the four datasets in Table 3. StoryGraph outperforms the comparison models in almost all cases followed by TwitInfo. LDA+Temporal is seen to be trailing in all cases. This observation is reasonable since building a single topic model over the tweet datasets will produce words distributions which are not focused on a specific subject. Even though ConnectingDots [25] and EventGraph [30] outperforms the naive approach, these models do not perform well in most cases.

Table 3: Averaged Performance comparison on all datasets

Model	ROGUE- 1		ROGUE-2		ROGUE-L	
	Precision	Recall	Precision	Recall	Precision	Recall
ConnectingDots	0.443	0.592	0.142	0.154	0.372	0.354
LDA+Temporal	0.332	0.425	0.021	0.023	0.121	0.263
EventGraph	0.522	0.53	0.258	0.253	0.254	0.248
TwitInfo	0.551	0.621	0.292	0.281	**0.513**	0.462
StoryGraph	**0.781**	**0.872**	**0.432**	**0.392**	0.503	**0.572**

This observation further affirms the claims that traditional document summarization (TDS) techniques do not perform well with Twitter data due to the unique properties of Twitter data [8, 11, 22]. We further show the F1-scores across the 4 datasets in Fig.3. StoryGraph achieves the best performance across the datasets. Recall that, StoryGraph incorporates user communities in extracting event summaries. The use of community structure contributes to this good performance. From earlier work in [4, 7], using community structure helps to extract event summaries that are coherent with fewer word intrusions. The reason is that conversations in user communities are usually focused on specific subjects of interests. Hence, there are fewer word intrusions from conversations that are not related to the topic of discussion in the community.

Structure Coherence: We evaluate three different timeline and story generation structures. A Complex Graph Structure (CGS) [30], Flat Timeline (FT) [25] and our Storygraph (SG) structure. The goal is to find out which structure presents the general information of the story in a more coherent and logical manner. We utilize the FP datasets since we have the complete evolution of events in the GSR. We present the results on the evaluation metrics in Fig.4. StoryGraph (SG) achieves the best structure coherence, followed by flat timeline (FT). Complex graph (CGS) is observed to be trailing behind in this section. While this is surprising, the possible explanation that Eventgraph [30] presents a graph structure which models the complex relationships between events in the graph. Thus, if all the nodes in the graph share some similar keywords, edges are created between them. This results in complex structures with low theme inconsistency among node pairs. A good summary structure should be simple while at the same time having the capability to model complex relationships between events, which is exactly what StoryGraph provides.

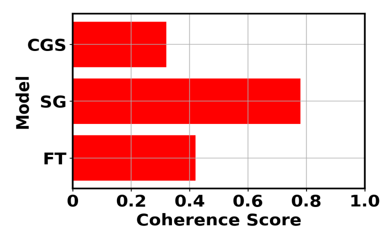

Figure 4: Coherence Comparison in Structural Summaries

6 CONCLUSION

In this paper, we have presented StoryGaph, a timeline graph summarization framework for telling the story of events that evolve over time. We formulate a novel graph timeline summarization structure capable of identifying the different themes of a story using high penalty metrics. StoryGraph constructs coherent paths to track event evolution and generates comprehensive story summaries to tell the story of how events evolve over time. Experimental results show that StoryGraph outperforms existing models.

REFERENCES

[1] Hamed Abdelhaq, Christian Sengstock, and Michael Gertz. 2013. EvenTweet: Online Localized Event Detection from Twitter. *PVLDB* 6, 12 (2013), 1326–1329.

[2] Dhekar Abhik and Durga Toshniwal. 2013. Sub-event detection during natural hazards using features of social media data. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 783–788.

[3] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. 2014. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research* 15, 1 (2014), 2773–2832.

[4] Jeffery Ansah, Wei Kang, Lin Liu, Jixue Liu, and Jiuyong Li. 2018. Information Propagation Trees for Protest Event Prediction. In *PAKDD 2018*. Springer, 777–789.

[5] Jeffery Ansah, Wei Kang, Lin Liu, Jixue Liu, and Jiuyong Li. 2018. SensorTree: Bursty Propagation Trees as Sensors for Protest Event Detection. In *Web Information Systems Engineering - WISE 2018 - 19th International Conference, Dubai, United Arab Emirates, November 12-15, 2018, Proceedings, Part I*. 281–296. https://doi.org/10.1007/978-3-030-02922-7_19

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[7] Jose Cadena, Gizem Korkmaz, Chris J Kuhlman, Achla Marathe, Naren Ramakrishnan, and Anil Vullikanti. 2015. Forecasting social unrest using activity cascades. *PloS one* 10, 6 (2015), e0128879.

[8] Deepayan Chakrabarti and Kunal Punera. 2011. Event Summarization Using Tweets. *ICWSM* 11 (2011), 66–73.

[9] Yi Chang, Jiliang Tang, Dawei Yin, Makoto Yamada, and Yan Liu. 2016. Timeline Summarization from Social Media with Life Cycle Models.. In *IJCAI*. 3698–3704.

[10] Ruifang He, Yang Liu, Guangchuan Yu, Jiliang Tang, Qinghua Hu, and Jianwu Dang. 2017. Twitter summarization with social-temporal context. *World Wide Web* 20, 2 (2017), 267–290.

[11] Karen Spärck Jones. 2007. Automatic summarising: The state of the art. *Information Processing & Management* 43, 6 (2007), 1449–1481.

[12] April Kontostathis, Leon M Galitsky, William M Pottenger, Soma Roy, and Daniel J Phelps. 2004. A survey of emerging trend detection in textual data mining. In *Survey of text mining*. Springer, 185–224.

[13] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[14] Chen Lin, Chun Lin, Jingxuan Li, Dingding Wang, Yang Chen, and Tao Li. 2012. Generating event storylines from microblogs. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 175–184.

[15] Chin-Yew Lin and Eduard H. Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *HLT-NAACL*.

[16] Bang Liu, Di Niu, Kunfeng Lai, Linglong Kong, and Yu Xu. 2017. Growing Story Forest Online from Massive Breaking News. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 777–785.

[17] Quanchao Liu, Heyan Huang, and Chong Feng. 2013. Micro-blog post topic drift detection based on LDA model. In *Behavior and Social Computing*. Springer, 106–118.

[18] Deepa Mallela, Dirk Ahlers, and Maria Soledad Pera. 2017. Mining Twitter features for event summarization and rating. In *Proceedings of the International Conference on Web Intelligence*. ACM, 615–622.

[19] Christopher D Manning, Christopher D Manning, and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.

[20] Adam Marcus, Michael S Bernstein, Osama Badar, David R Karger, Samuel Madden, and Robert C Miller. 2011. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 227–236.

[21] Ramesh Nallapati, Ao Feng, Fuchun Peng, and James Allan. 2004. Event threading within news topics. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM, 446–453.

[22] Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining text data*. Springer, 43–76.

[23] Jeffrey Nichols, Jalal Mahmud, and Clemens Drews. 2012. Summarizing sporting events using twitter. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. ACM, 189–198.

[24] Kevin Dela Rosa, Rushin Shah, Bo Lin, Anatole Gershman, and Robert Frederking. 2011. Topical clustering of tweets. *Proceedings of the ACM SIGIR: SWSM* (2011).

[25] Dafna Shahaf and Carlos Guestrin. 2010. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 623–632.

[26] Imane Tamimi, El Khadir Lamrani, and Mohamed El Kamili. 2017. Community Detection Through Topic Modeling in Social Networks. In *International Symposium on Ubiquitous Networking*. Springer, 70–80.

[27] Vikas Thada and Vivek Jaglan. 2013. Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm. *International Journal of Innovations in Engineering and Technology* 2, 4 (2013), 202–205.

[28] Han Wang and Jia-Ling Koh. 2017. Timeline Summarization for Event-Related Discussions on a Chinese Social Media Platform. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 579–594.

[29] Shize Xu, Shanshan Wang, and Yan Zhang. 2013. Summarizing complex events: a cross-modal solution of storylines extraction and reconstruction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1281–1291.

[30] Christopher C Yang, Xiaodong Shi, and Chih-Ping Wei. 2009. Discovering event evolution graphs from news corpora. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 39, 4 (2009), 850–863.