Text Stream to Temporal Network - A Dynamic Heartbeat Graph to Detect Emerging Events on Twitter

No Author Given

No Institute Given

Abstract. Huge mounds of data are generated every second on the Internet. People around the globe publish and share information related to real-life events they experience every day. This provides a valuable opportunity to analyze the content of this information to detect real-life happenings, however, it is quite challenging task. Most of the existing methods focus on bursty features to highlight the significance of data entities, but ignore the fact that burstiness often dominates the other minor details which, sometimes, can be very important. Based on this fact, in this work, we propose a novel graph-based approach named the Dynamic Heartbeat Graph (DHG) that not only detects the events at an early stage, but also suppresses them in the upcoming adjacent data stream in order to highlight new emerging events. This characteristic makes the proposed method interesting and efficient in finding emerging events and related topics. The experiment results on real-life datasets (i.e. FA Cup Final and Super Tuesday 2012) show a considerable improvement in most cases, while time complexity remains very attractive.

Keywords: Dynamic graph, time series analysis, event detection, text stream, big data, emerging trend

1 Introduction

In recent years, with the unprecedented growth of social media and blog networks, huge amounts of diverse types of data are being generated every day. The information that is collectively generated on such platforms is of great value. In addition to its huge volume and diversity, much of the data is inter-dependent in nature. The analysis of such data is quite important and helps to successfully detect meaningful information that could be used for searching, discovering patterns and sensing trends. The detection of emerging trends from social media text streams has recently become a research area of great interest. However, real-time streaming data is quite complicated to analyze. Recent work mainly focuses on event detection using bursty features or graph similarity patterns using subgraph matching [5, 6], however, there is a need for a more scalable and localized pattern analysis approach to detect emerging events in text streams.

Analyzing large, diverse and noisy data, especially social media, requires addressing scalability, accuracy as well as complexity challenges. Documents describing the same event and story have a similar set of collocated keywords

that could be used to identify time and its description. In order to identify significant/unusual patterns, recently, graph-based methods have been extensively applied to deal real-life data efficiently [10–12].

Graph mining has received considerable attention in the data analytic community. Most of the time, data is gathered as a stream of time, thus traditional graph-based algorithms are not efficient to process data of such complex nature (i.e. dynamic and non-stationary). Most existing graph-based methods focus on frequent, co-occurrent, and highly weighted patterns to highlight the significance of data entities, but ignores the fact that burstiness often dominates the other related details that exist in the data which, sometimes, can be very important. In this work, we present a novel graph-based approach named Dynamic Heartbeat Graph (DHG) based on the differences between temporal graphs. The proposed DHG approach not only detects events at an early stage but also suppresses the burstiness of event related topics in the upcoming data stream for a certain time interval in order to highlight new emerging events. This characteristic makes the DHG approach unique and efficient in finding new emerging events and related topics.

Our approach specifically focuses on micro-sized documents, such as those published on micro-blogging services like Twitter and Facebook. We formulate the text stream as a series of disjoint temporal graphs. These disjoint graphs are further processed to generate heartbeats within each time window of fixed temporal length. We design three features *growth factor*, *trend probability*, and *topic centrality*. Based on these features, we use a binary classifier to detect emerging events in data stream.

The **goal** of this paper is to address the key aforementioned problem of time series data analysis to detect emerging events. By employing the proposed DHG approach which analyzes the patterns in adjacent time windows, we can overcome the limitations of the state-of-the-art work by identifying key occurrences efficiently. We describe the theoretical and empirical **key contributions** of this work as follows:

- A novel graph-based approach named Dynamic Heartbeat Graph (DHG) which is efficient in the detection of events.
- Low computational complexity of proposed method, which generates a series of DHGs in $O(K|V|^2)$, where K is total number of DHGs which is considerably small in value, and classifies all DHGs in $O(N^2)$.
- The latter method is evaluated empirically on the FA Cup and Super Tuesday datasets. The experiment results on data show that the DHG outperforms state-of-the-art methods.

2 Related Work

Graph-based methods have become extremely popular for analyzing real-life data. Graph kernels have been successfully used to compare graph structure [4–8, 10–12].

Most of the existing graph kernels [10, 11] compare specific substructures of graphs locally that correspond to small subgraphs or to the relationships between small subsets of vertices. A comparison of a specific substructure focuses on local properties which results in the global structure of a graph being ignored. Only considering substructure and ignoring global structure could work for a small graph but is not suitable for large-sized graphs, as several properties may not be utilized in locally based approaches.

To overcome this issue, recent developments considered two graph kernels [5]. However, in this approach, kernels are based on unlabeled graphs. In other work, Johansson and Dubhashi compared pairs of graphs by computing the optimal matching between vertices generated from a graph-related adjacency matrix [6]. Nikolentoes et al. considered the features that best describe the global properties of a graph [4]. A set of vectors (graphs) corresponds to the embedding of a graph's vertices which is computed by eigenvalue decomposition of the vector adjacency matrix. The similarity between two graphs is computed using the Earth Movers distance metric. Eigenvectors capture global properties thus, it works for both labeled and unlabeled graph. However, it finds the similarity between two graphs hence this technique may not work well on graphs based on text stream. It may fail to identify change patterns that are more likely to appear in social text stream. Moreover, it considers global features based on eigenvectors thus, it is more biased towards similarity patterns.

Velampalli and Eberle proposed graph-based anomaly detection method by adding background knowledge (in the form of rule coverage) to the evaluation metrics, and as well as biasness by assigning negative weights to the substructure of the graph [9]. An anomalous structure is identified based on the negative weight to the rule coverage that represents the final graph covered by the instances of the substructure. Similar to [4], Velampallli and Eberle's approach is suitable to identify similar patterns in a graph whereas a common substructure with fewer nodes is considered as anomaly. Since content shows great diversity when an event occurs, therefore, this method may not be efficient to detect events in the text stream. A large-sized substructure of a graph would be effected whenever a real-life event is reported in the data. It is more likely that substructure of a graph which is considered an anomaly may not reveal accurate topics to identify the underlying event due to the diverse nature of the text stream over time.

Recent graph-based approaches rely on subgraph structures. However, these methods are not efficient for text stream data due to the diversity of content and dynamic nature which exist in their relationship. Thus, due to the rapid changes in the graph, these methods may not detect upcoming emerging events. Generally, a graph structure cannot be patronized due to the diverse nature of data; thus, Unlike the existing approaches that work on graph substructure, we propose a noval graph-based approach that transform and map two adjacent temporal network on to a new graph called DHG. The DHG is implicitly resistant to the burstiness based on the change in the weights of nodes and edges in preceding graph with respect to the time. Furthermore, node centrality charac-

teristic inherited from temporal network makes the DHG robust against existing approaches (see Section 4 for details).

3 Preliminaries

A ***Micro-document*** is short textual content consisting of words that are published online through some micro-blog. It is defined as 3-tuple $d_i = (t, u, W)$, where u is a user who publishes a micro-document d_i with some set of words W at a specific time instance t. A ***Text Stream*** is a set of micro-documents $\mathcal{D} = \{d_1, d_2, d_3, ..., d_n\}$, where d_i and $d_{(i-1)}$ are the i^{th} and $(i-1)^{th}$ micro-documents published at time $\pi_1(d_i)$ and $\pi_1(d_{i-1})$ respectively, such that $\pi_1(d_i) \geq \pi_1(d_{(i-1)})$. The lengths of micro-documents are usually short hence, the measures based on burstiness, similarity as well as distance may not yield good results, however this issue could be resolved by creating a super-document. Let $\mathcal{D}$ be the set of all micro-documents available in a text stream, then a ***Super-document*** d_i^ρ is a continuous temporal accumulation of each $d_i \in \mathcal{D}$ separated at t_a and $t_{(a+b)}$ time intervals (we refer as t_i later in the paper). To create a super-document, instead of merging the micro-documents into one core document, we create k partitions in text stream $\mathcal{D}^\rho = \{\{d_1, d_2, ..., d_p\}, \{d_{p+1}, ..., d_{p+q}\}, ..., \{..., d_n\}\}$. By doing so, we are able to retain the identity of each micro-document that we use later to generate a network series (See Section 4.1 for details) which increases the cohesiveness among the topics and keywords. Thus, this super-document can be defined as k number of mutually exclusive partitions i.e. $\bigcap_{i=1}^{|\mathcal{D}^\rho|} d_i^\rho = \varnothing$. A ***Sliding Window*** is a set of super-documents (chunk of data) whose temporal length Δt. Each sliding window is processed independently in each sliding window to detect event related information. A set of word(s) in a text stream may refer to a *topic*. When more people are using specific topic in their micro-document, it becomes a trend, often called a trending topic. Similar to other research studies, we use the terms "trend" and "event" interchangeably and also the terms "word(s)" and "topic(s)" [1–3, 7, 13, 14].

4 Dynamic Heartbeat Graph (DHG)

Using text stream, we devise a technique that creates a series of dynamic disjoint graphs and then maps each adjacent pair of graphs in a network series on to another DHG series. In order to classify DHGs as candidates for events, we design and use trend probability, change in burstiness, and normalized degree centrality in the DHGs as key features. This section defines all the components involved in the transformation of text stream into series of temporal networks.

4.1 Network Series

Against each super-document $d_i^\rho \in D^\rho$, a network G_i is created in such a way that each node is a "word" and an edge between two nodes represents

co-occurrence relationship. A network series is a set of disjoint graphs $\mathcal{G} = \{G_1, G_2, G_3, ..., G_{|\mathcal{D}^\rho|}\}$, where each $G_i \in \mathcal{G}$ is built against $d_i^\rho \in \mathcal{D}^\rho$ such that G_i is a labeled graph, i.e. $G_i = (V, E, \mathscr{W}, \mathscr{S})$, where V is a set of nodes such that $\forall v_i \in V$ are the unique words which appear in d_i^ρ, and $E \subseteq V \times V$ is a set of edges such that $e_k = (v_k, v_k^{'}) \wedge v_k \neq v_k^{'}$. $\mathscr{W} : V \to \mathbb{R}$ and $\mathscr{S} : E \to \mathbb{R}$ are the functions that assign weights to each node and edge in the graph G_i as shown in Equation 1 and 2, where $|d_i^\rho(v_k)|$ is the term-frequency of v_k and $|d_i^\rho(v_k, v_k^{'})|$ is the frequency of co-occurrence of nodes v_k and $v_k^{'}$ in super-document d_i^ρ.

$$\mathscr{W}(v_k) = |d_i^\rho(v_k)| \tag{1}$$

$$\mathscr{S}(e_k) = |d_i^\rho(v_k, v_k^{'})| \tag{2}$$

The network is created in such way that it retains the coherence among the words of each micro-document d_i participating in the building of the network. The coherence is enhanced by creating a clique among the words of each d_i. *Clique* — each node $v_k \in d_i$ is connected to every other node $v_k^{'} \in d_i$. This results in an increase in the central tendency of topics within the large network of diverse words.

4.2 DHG Series

To create a DHG series, Algorithm 1 linearly combines and maps every pair of adjacent graphs G_i and G_{i-1} on to a new DHG G_i^h which is used further for emerging trend detection. The goal of generating a set of DHG $\mathcal{G}^h$ is to discriminate among topics and the drift in their popularity within each subsequent graph.

Algorithm 1: Generate Set of Dynamic Heartbeat Graphs

input : $\mathcal{G} = \{G_1, G_2, G_3, ..., G_{|P|}\}$ set of a graph series
where $\exists G_i \in \mathcal{G}$ is generated against $\exists d_i^\rho \in D^\rho$

output: $\mathcal{G}^h = \{G_1^h, G_2^h, ..., G_{|\mathcal{G}|-1}^h\}$
$\varepsilon = \{\epsilon_1, \epsilon_2, \epsilon_3, ..., \epsilon_{|\mathcal{G}|-1}\}$

for $i \leftarrow 1$ **to** $|\mathcal{G}| - 1$ **do**
  $U \leftarrow$ `Join`$(V^{G_i}, V^{G_{i+1}})$
  $A \leftarrow$ `RegenerateMatrix`$(G_i\ ,\ U)$
  $B \leftarrow$ `RegenerateMatrix`$(G_{i+1}\ ,\ U)$
  $\varepsilon[i] \leftarrow$ `EstimateHeartbeat` $A, B, V^{G_i}, V^{G_{i+1}}$
end

The DHG algorithm takes network series G as input and generates another series of networks which we call Dynamic Heartbeat Graph (DHG) series. For every adjacent pair of graphs G_{i-1} and G_i, the algorithm aligns the dimensions of the adjacency matrices by taking union of the vertices in both graphs and

then reorders them canonically. In later step (at Line 5), the algorithm estimates the change in the node and edge weights (see Algorithm 2) and stores it in an indexed vector $\varepsilon[i] \in \mathbb{R}^{n\times 3}$. The step-by-step implementation detail is given in Algorithm 1. An example in Figure 1 shows how the DHG between two networks is calculated, where node weights (given by "()"), and edge weights can be seen in the graphs as well as in the adjacency matrices. Reordering each graph G_i canonically and transforming the DHG into vector-space reduce the computational complexity significantly from $O(K|V|^4)$ to $O(K|V|^2)$, where K is considerably a very small value, i.e. $K = |\mathcal{G}^h|$.

Algorithm 2: Estimate change in burstiness

input : A, B are adjacency matrices that represent G_{i-1} and G_i respectively.
V^A and V^B are lists of vertices in G_{i-1}, G_i respectively

output: e vector that represents a DHG against G_{i-1} and G_i
V^H list of vertices in DHG

```
1  for k ← 1 to |V^B| do
2  |  V^H[k] ← V^B[k] − V^A[k]
3  end
4  for x ← 1 to |V^B| do
5  |  for y ← 1 to x do
6  |  |  edgeWt ← B[x, y] − A[x, y]
7  |  |  if edgeWt! = 0 then
8  |  |  |  e.Add( x,y, edgeWt)
9  |  |  end
10 |  end
11 end
```

The DHG approach implicitly suppresses and handles the dominance of bursty topics by calculating the change in the weights of each node and edge between each pair of adjacent graphs $G_{(i-1)}$ and G_i in order to highlight other details which are less frequent. Algorithm 2 estimates and labels all the corresponding nodes and edges with new weights in DHG G_i^h. The DHG series is a set of disjoint graphs, generated in a streaming fashion; therefore, it is temporally well aligned with the text stream. Furthermore, these DHGs are used to detect emerging events. The detail of the detection method is given in next section.

4.3 Event Detection Method

In the following section, we present our event detection method using DHGs. The event detection method works on the following assumptions:

– The text stream has diverse contents, but an emerging event may only occur in a text stream whenever there is a significant change in either burstiness

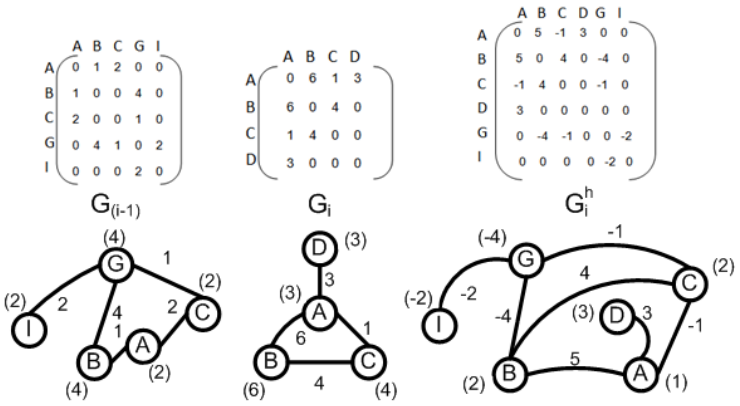

Fig. 1. Example for the creation of DHG from two subsequent time intervals

displacement of existing topics or the appearance of new topics in the text stream between two adjacent time intervals at $t_{(i-1)}$ and t_i.

– The significant change is not only dependent upon the burstiness of topics, but also the change in their probability distribution and central tendencies within the network.

In all of the following equations, for the simplification of the notations, let $\psi = G_i^h$ where G_i^h is i^{th} heartbeat graph. The detection method uses the fusion of three key features *GrowthFactor*, *TrendProbability*, and *TopicCentrality* as shown in equations 3,4, and 6, respectively and calculates *HeartbeatScore* $\mathscr{H}(\psi)$ as shown in equation 7. Whereas *GrowthFactor* $Gr_{fact}(\psi)$, *TrendProbability* $Tr_{Prob}(\psi)$, and *TopicCentrality* $\Sigma\mathscr{C}(v^{\psi})$ represents the significance of the change in the burstiness of topics, possibility of an emerging event at time interval t_i, and central tendency and coherence among different topics in DHG ψ, respectively.

The *GrowthFactor* of DHG ψ is calculated as shown in Equation 3 where $\mathscr{W}(v_k^{\psi})$ is the k^{th} node weight that represents a change in burstiness of a topic between G_i and G_{i-1} (see Algorithm 2). A higher score of *GrowthFactor* shows that the topics are appearing with high frequency in sliding window $k\Delta t$.

$$Gr_{fact}(\psi) = \sum_{k=0}^{|V^{\psi}|} \mathscr{W}(v_k^{\psi}) \tag{3}$$

A node in DHG ψ can have negative and positive weights. To calculate *TrendProbability* $Tr_{Prob}(\psi)$, the probability distribution against positive $\mathscr{W}(v_k^{\psi+})$ and negative $\mathscr{W}(v_k^{\psi-})$ weights of each word are calculated within the DHG ψ. The probability distribution over positive and negative weights are then linearly combined, as shown in Equation 4, which shows the convergence of DHG ψ towards trending topics, where β_1 and β_2 are 1 and -1 respectively. $Tr_{Prob}(\psi) > 0$ indicates that topics are gaining popularity, thus, denoting the possibility of emerging event(s) in sliding window $k\Delta t$.

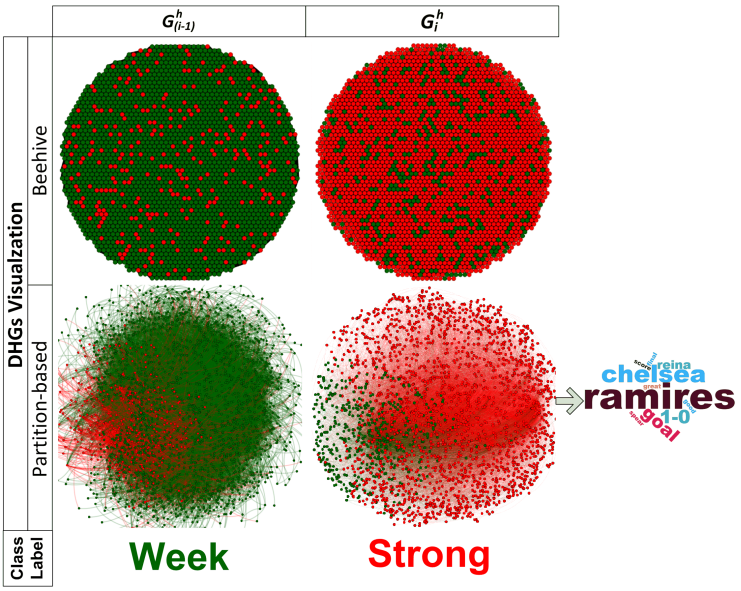

Fig. 2. Graph visualization (Beehive and Partition-based) of two subsequent DHGs $G^h_{(i-1)}$ and G^h_i at time $t_{(i-1)}$ and t_i respectively, from the FA Cup 2012 dataset. At t_i a significant event "Goal" occurs. Red and green are the nodes with positive and negative weights respectively. The DHG approach shows hyper-sensitivity to burstiness and as well as newly emerging topics.

$$Tr_{Prob}(\psi) = \beta_1 \sum_{k=0}^{|V^{\psi+}|} \frac{\mathscr{W}(v_k^{\psi+})}{\sum_{l=0}^{|V^{\psi}|} |\mathscr{W}(v_l)|} + \beta_2 \sum_{k=0}^{|V^{\psi-}|} \frac{|\mathscr{W}(v_k^{\psi-})|}{\sum_{l=0}^{|V^{\psi}|} |\mathscr{W}(v_l)|} \tag{4}$$

TopicCentrality $\mathscr{C}(v_k^{\psi})$ is then calculated to highlight the central tendency of topics in each DHG ψ, as shown in equation 6 where v_k^{ψ}, $\epsilon_i \in \varepsilon$, $\pi_3(e_i)$, and $|V^{\psi}|$ represent the topic, indexed edge vector, weight of edge e_i connected to v_k^{ψ}, and the total number of topics in DHG ψ, respectively. In the calculation of *TopicCentrality*, all the edges with negative weights are dropped because of the initial assumption (see Section 4), which positively influences the centrality of newly emerging topics with respect to the existing ones. A higher aggregated centrality score shows that the emerging topics are coherent and concurrently appearing in text stream in sliding window $k\Delta t$. The detection method comprises two steps:

1. If $Tr_{Prob}(\psi) \leq 0$ then DHG is assigned to the *"Weak"* class. Once the highly frequent topics reach their peak, they start to lose their importance because of the decay in their burstiness. If the weights of certain topics are reduced at time t_i compared to t_{i-1} and there is no significant increase in the weights of the other topics, then the *TrendProbability* score is always negative, therefore indicating the fact that the heartbeat between G_{i-1} and G_i is not significant.

2. Otherwise calculate heartbeat score $\mathscr{H}(\psi)$ which is the product of *Growth-Factor* $Gr_{fact}(\psi)$, *TrendProbability* $Tr_{Prob}(\psi)$, and aggregated *TopicCentrality* $\Sigma\mathscr{C}(v^{\psi})$ in DHG ψ (as shown in equation 7).

To assign a binary class membership (i.e. $[Strong, Weak]$) to each DHG ψ, where *"Strong"* means DHG ψ contains emerging trends and *"Weak"* means an insignificant heartbeat, a classification function $Est(\psi)$ (as shown in equation 8) estimates and assigns two-class labels to each DHG $\psi \in \mathcal{G}^h$. Here, θ is an adoptive measure that finds the local optimum value in each sliding window $k\Delta t$ to set a threshold for classification function $Est(\psi)$ as shown in Equation 5, where Δt, τ are the temporal length of each sliding window and super-document d_i^{ρ} respectively such that $\Delta t (\text{mod } \tau) = 0$, i is the index of the first DHG in the sliding window under consideration, $Gr_{fact}(\psi)$ is the *GrowthFactor* (as Equation 3) of each DHG ψ, and ω is the adjustment parameter. We set ω as 1 and 0.6 for the FA Cup and Super Tuesday dataset, respectively.

$$\theta_{(k\Delta t)} = \frac{\tau \sum_{i}^{i+\frac{\Delta t}{\tau}} (Gr_{fact}(\psi))}{\Delta t} + \omega \sqrt{\frac{\tau (Gr_{fact}(\psi) - \frac{\tau \sum_{i}^{i+\frac{\Delta t}{\tau}} (Gr_{fact}(\psi))}{\Delta t})^2}{\Delta t}} \tag{5}$$

$$\mathscr{C}(v_k^{\psi}) = \frac{\sum^{\forall i:(\pi_1(\epsilon_i^{\psi})=k) \vee (\pi_2(\epsilon_i^{\psi})=k) \wedge (\pi_3(\epsilon_i^{\psi})>0)} \pi_3(\epsilon_i^{\psi})}{|V^{\psi}|} \tag{6}$$

$$\mathscr{H}(\psi) = Gr_{fact}(\psi) \times Tr_{Prob}(\psi) \times \sum_{k=0}^{|V^{\psi}|} \mathscr{C}(v_k^{\psi}) \tag{7}$$

$$Est(\psi) = \begin{cases} \text{'Strong'}, & \text{if } \mathscr{H}(\psi) \geq \theta_{(k\Delta t)} \\ \text{'Weak'}, & \text{otherwise} \end{cases} \tag{8}$$

The transformation of the DHG series into vector-space ε (see Section 4.2) results in reducing the computational complexity of binary classification from $O(|V|^2)$ to $O(N^2)$, where $V = Max(|V^{\psi}|)$and $N = Max(|\epsilon_i|)$. Here, the value of $N^2 \ll |V|^2$. In the worst case scenario, $O(|V|^2) = O(N^2)$ if and only if all DHG ψ are complete graphs, however, the occurrence of such scenarios is quite rare as we know that each DHG $\psi \in \mathcal{G}^h$ is sparse due to the diverse content of the text stream. Later, in each sliding window $k\Delta t$ a ranked topic list in the candidate DHGs that are classified as *"Strong"* is generated by calculating the score of each topic, as shown in Equation 9. Figure 2 shows the heartbeats of two subsequent DHGs and their class labels using classification function $Est(\psi)$ with top ten trending topics.

$$Rank(v_k^{\psi}) = \mathscr{C}(v_k^{\psi}) \times \mathscr{W}(v_k^{\psi}) \tag{9}$$

5 Experiment and Results

In this section, we evaluated the performance of proposed dynamic heartbeat graph as discussed in table 1.

5.1 Evaluation

To evaluate the performance of proposed DHG method, two benchmark datasets "FA Cup" and "Super Tuesday", and the framework introduced by [1] are used for the comparison. We create partitions for the super-document accumulation as one minute and five minutes for the "FA Cup" and "Super Tuesday" datasets, respectively. The DHG method takes input data in a streaming fashion and create time series disjoint network and then DHG series based on the temporal length of accumulation.

Two evaluation measures: *Topic-Recall@K* (T-Rec), which is the percentage of ground truth topics detected correctly at top K retrieved topics; and *Keyword-Precision@K* (K-Pre), which is the percentage of keywords detected correctly out of the top K number of keywords are used. *T-Rec* and *K-Prec* are calculated by micro-averaging the individual *T-Rec* and *K-Prec* scores.

5.2 Dataset

We conducted our experiments on a well-known benchmark datasets ("FA Cup final" and Super Tuesday" [1]). The "FA Cup" is one of the oldest knock-out football competition and very popular among the fans around the world. The dataset consists of a text stream of the final match held on May 5^{th}, 2012, between the Chelsea and Liverpool teams. The ground truth comprised 13 topics, including goals, bookings and fouls, kick-off, half-time and match ending. The "Super Tuesday" dataset is the US presidential primary election held on Tuesday 6 March 2012, the key moment when it is likely that the party nominee is elected as presidential candidate. The ground truth comprised 22 topics covering stories related to the projection and success of nominees in particular states and their speeches. For evaluation purposes, the temporal length Δt of each sliding window is set to one minute and one hour for the FA Cup and Super Tuesday datasets, respectively. The ground truth contains topics with respect to each sliding window. To reduce noise, the datasets are pre-processed. To improve data quality, retweets, tweets containing URLs or those containing less than three (3) words are removed. Furthermore, common words, stop words, the words which have less than three letters, and punctuation are removed.

5.3 Results

We present the results for *Topic-Recall* (T-Rec) at $K = 2, 4, 6, ..., 20$ in Figure 3, and *Keyword-Precision* (K-Prec) in Table 1 to compare the six different event detection methods including DHG. Our method gives best results on FA Cup, because of the users those publishing contents on micro-blogs, are very focused, consistent, and to the point due to the popularity and limited time of the underlying event. Therefore, the topics reported in the text stream are less diverse, making them easier to detect compared to the Super Tuesday dataset. The topics which are reported by the ground truth are taken from the mainstream media and cover a broader semantic prospective. For instance at time

window 17 : 56 in the FA Cup, ground truth marked "Andy, Carroll" as topic, whereas "header,cech,over,claim,equalize" are among the other keywords therefore, it is more likely that the topics are among the top trends but they do not necessarily appear in the top most position every time. The DHG method has comparable T-Rec at $K = 2, 4, 6$, and 8. It eventually achieves the maximum possible T-Rec at $K \geq 10$ for the FA Cup dataset. Similarly, the DHG method outperforms the other detection methods after $N > 30$ for the Super Tuesday dataset. The results for T-Rec are shown in Figure 3

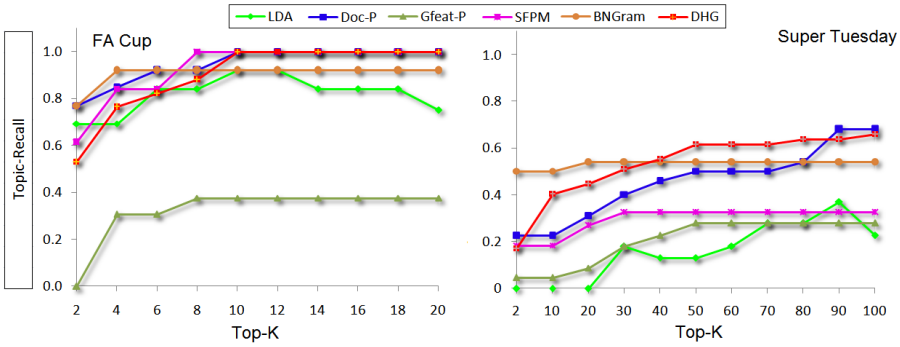

Fig. 3. Topic-Recall@K for six (6) different well-known methods for the FA Cup (left) and Super Tuesday (right) datasets.

Conversely, the DHG method combines the scores of change in topics' burstiness and central tendency in the graph, therefore it is able to detect relevant keywords with high precision compared to the other methods for both datasets, as shown in Table 1. Hence, DHG exhibits one of the effective detection method in terms of performance and accuracy. Figure 4 shows the heartbeat pattern on the FA Cup 2012 data and the identification of emerging trends across its temporal length.

Table 1. DHG outperforms all other detection methods for K-Prec@2 for both the FA Cup and Super Tuesday datasets

Method	FA Cup	Super Tuesday
LDA	0.164	0.000
Doc-P	0.337	0.511
Gfeat-P	0.000	0.375
SFPM	0.233	0.471
BNGram	0.299	0.628
DHG	**0.682**	**0.875**

We observe an interesting correlation among user participation, network size, and the heartbeat score of DHGs across the temporal data, where user participation is the total number of unique users who published at least one micro-document, and network size is the total number of unique words in the DHG ψ at time t_i. It is observed that the DHG method detects emerging events at an early stage. Whenever an event occurs on a particular time interval, our method detects the related topics and keywords before the diversity in the text

stream increases. On the other hand, we also observe that user participation also increases whenever an event occurs.

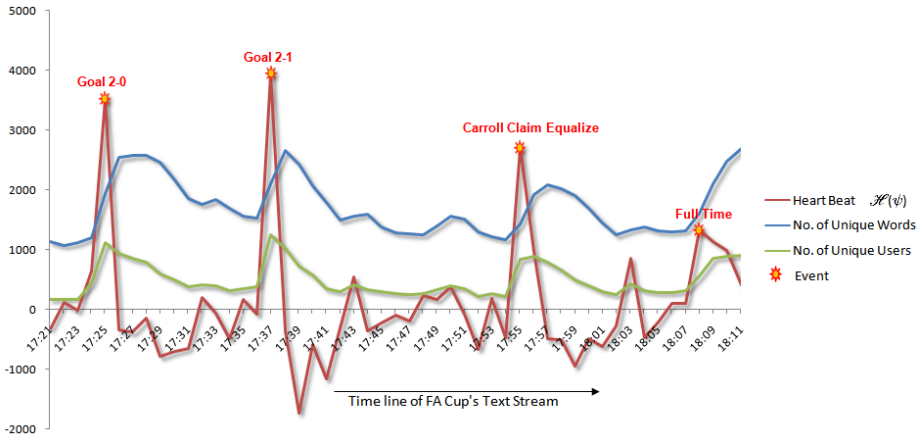

Fig. 4. Detected events with respect to the heartbeat pattern. The figure also shows the variations in the number of unique words and user participation across different time intervals

The heartbeat pattern, user participation, and the number of unique words in each time interval are shown in Figure 4.

6 Conclusion

In this paper, a novel Dynamic Heartbeat Graph (DHG)-based method is developed that is efficient for text streams such as Twitter. We formulated the text stream as a series of disjoint temporal graphs that are further processed to generate heartbeats within each time interval of fixed temporal length. Furthermore, we have designed three unique features growth factor, trend probability and topic centrality to identify the emerging events using DHG. In order to evaluate the performance of DHG, we have used two publicly available benchmark datasets (the FA Cup Final 2012 and Super Tuesday 2012). The quantitative evaluation shows that the DHG method is sensitive to the dynamic nature of text streams and detected emerging events with high precision compared to the state-of-the-art methods. Empirical evaluation showed that DHG method is robust in terms of computational complexity and scalability thus, it could be used for live streaming as well. In future, our study will focus on exploring the user participation and network-based features, and evaluating the proposed DHG approach on live data streams containing number of diverse events.

References

1. Luca Maria Aiello, Georgios Petkos, Carlos Martin, David Corney, Symeon Papadopoulos, Ryan Skraba, Ayse Göker, Ioannis Kompatsiaris, and Alejandro

Jaimes. Sensing trending topics in twitter. *IEEE Transactions on Multimedia*, 15(6):1268–1282, 2013.

2. James Benhardus and Jugal Kalita. Streaming trend detection in twitter. *International Journal of Web Based Communities*, 9(1):122–139, 2013.
3. Cody Buntain. Discovering credible events in near real time from social media streams. In *Proceedings of the 24th International Conference on World Wide Web*, pages 481–485, New York, NY, USA, 2015. ACM, ACM.
4. Polykarpos Meladianos andMichalis Vazirgiannis Giannis Nikolentzos. Matching node embedding for graph similarity. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 2429–2435, 2017.
5. Fredrik Johansson, Vinay Jethava, Devdatt Dubhashi, and Chiranjib Bhattacharyya. Global graph kernels using geometric embeddings. In *Proceedings of the 31st International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 2014.
6. Fredrik D Johansson and Devdatt Dubhashi. Learning with similarity functions on graphs using matchings of geometric embeddings. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 467–476. ACM, 2015.
7. Duc T Nguyen and Jai E Jung. Real-time event detection for online behavioral analysis of big social data. *Future Generation Computer Systems*, 66:137–145, 2017.
8. Tegjyot Singh Sethi and Mehmed Kantardzic. On the reliable detection of concept drift from streaming unlabeled data. *Expert Systems with Applications*, 82:77–99, 2017.
9. Sirisha Velampalli and William Eberle. Novel graph based anomaly detection using background knowledge. In *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference*, pages 538–543, 2017.
10. Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1365–1374. ACM, 2015.
11. Pinar Yanardag and SVN Vishwanathan. A structural smoothing framework for robust graph comparison. In *Advances in Neural Information Processing Systems*, pages 2134–2142, 2015.
12. Yibo Yao and Lawrence B Holder. Detecting concept drift in classification over streaming graphs. In *KDD Workshop on Mining and Learning with Graphs (MLG). August 14, 2016. San Francisco, CA*, pages 2134–2142, 2016.
13. Deyu Zhou, Liangyu Chen, and Yulan He. An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In *Proceedings of 29th AAAI Conference on Artificial Intelligence*, pages 2468–2475, USA, 2015. AAAI.
14. Xiangmin Zhou and Lei Chen. Event detection over twitter social media streams. *The VLDB JournalThe International Journal on Very Large Data Bases*, 23(3):381–400, 2014.