Detecting and Summarizing Emergent Events in Microblogs and Social Media Streams by Dynamic Centralities

Neela Avudaiappan*, Alexander Herzog*, Sneha Kadam*, Yuheng Du*, Jason Thatcher†, Ilya Safro*‡
* *School of Computing, Clemson University*
† *Department of Management, Clemson University*
Clemson, South Carolina 29634
Email: {navudai, aherzog, skadam, yuhengd, jthatch, isafro}@clemson.edu
‡ *Corresponding author*

***Abstract*—Methods for detecting and summarizing emergent keywords have been extensively studied since social media and microblogging activities have started to play an important role in data analysis and decision making. We present a fast system for monitoring emergent keywords and summarizing a document stream based on the dynamic semantic graphs of streaming documents. We introduce the notion of dynamic eigenvector centrality for ranking emergent keywords, and present an algorithm for summarizing emergent events that is based on the minimum weight set cover. Our system is demonstrated on the streaming Twitter data related to public security.**

***Keywords*-public security; dynamic centrality; social media; microblogging; emergent events**

I. Introduction

Federal and state security organizations invest enormous efforts in detecting and monitoring public security events, such as social protests, migrant crisis, and terrorist attacks. Unfortunately, emergent security risks associated with such events are not timely detected and monitored. Recent examples include the New Year's Eve coordinated sexual assaults in Germany, the Boston marathon bombing, and terrorist attacks in France. As the details of such events begin to emerge, authorities, and citizens around the world are storming social media. This includes both the condemnation and defense of these events. After the self-proclaimed Islamic State (ISIS) claims responsibility for the violence, the social media is abuzz with both the condemnations of and solidarity with the extremists.

Another major problem with the blogosphere and social media is the enormous volume of online propaganda, suggesting that acts of terror may help the terrorist group attract supporters and conduct its recruitment. All these and many other reasons put detecting and monitoring of emergent public safety events in streaming data among the top priorities of federal and state authorities. It has been shown multiple times that relevant information can appear in Twitter much faster than the authorities begin to respond. For example, in the 2015–2016 New Year's Eve coordinated sexual assaults in multiple cities in Germany, people started to report via Twitter much earlier than the local authorities started to react. It has been discussed in media that the reporting of such incidents in one city would have been helpful in preventing similar events other cities. Emergent event detection is also important for other types of streaming data, such as search engine query monitoring, and climate data analysis.

Methods for implementing emergent event detection are often task oriented and data dependent. In search engine query monitoring, such techniques identify the rising queries that reflect the user's attention at the moment. Golbandi et al. [1], for example, use linear regression to predict future query counts. Dong et al. [2] propose a query classification framework to improve the detection of recent events. A query usually consists of short text phrases and query counts of each search term are often the most crucial feature for emergent search query detection. Both methods are based on query counts, where correlations between search terms are neglected. These methods work great for news-like search data.

In social media, it is critical to identify the changing and emerging themes from the vast amount of text streams [3], [4], [5]. Due to the fast changing nature of social data, event detection methods are often built around abrupt changes from real-time data streams. Mathioudakis and Koudas [3] built a Twitter Monitor tool which detects erupting keywords from live Twitter stream and group them into small groups based on the co-occurrences of keywords in recent tweets. A trend is then represented by a group of keywords that can be manually tagged by users for monitoring purposes.

Benhardus and Kalita's work [4] on Twitter data determines if a keyword should be selected into a topic by checking its term frequency-inverse document frequency (TF-IDF) value. The eventual trending topic being identified contains all the words that have a TF-IDF value greater than a certain threshold. Since there is no topic-word probabilistic model involved in Benhardus and Kalita's work, their trend detection approach is mainly based on word frequency and associated thresholds.

Saha and Sindhwani [5] introduce a topic-word and topic-

document association model and build a nonnegative matrix factorization (NMF) framework that separates evolving topics and emerging topics from Twitter. At the end of each time window, top K most emerging topics are generated as the trending topics. The rest of the topics are generated as the smoothly-changing evolving topics. No explicit threshold is needed for this approach and it outperforms the threshold based method used in Allan and James' [6] work in both computational time and precision. *Both* [4] *and* [5] *rely on the frequency of keywords in a given time range to recover trending keywords, which is an extremely noisy measure of keyword importance.* The process is also computationally difficult to be applied on real, high-frequency, streaming data.

Several methods have been developed in the past to extract keywords and summarize text using semantic graphs. These approaches have been used for extracting important keywords using centrality measures such as closeness centrality [7] that measures how well a keyword is connected to all other keywords in the graph by taking into account the shortest path distance between them. *However, in non-homogeneous text data, shortest path based approaches typically detect a lot of accidentally very close keywords whose closeness is meaningless with respect to mutual events.*

Another cohort of relevant centrality measures are based on the flows between nodes [8] as it models the amount of possible information content that can be transferred between nodes. In [9], a problem of text summarization is solved by finding the most important sentences, using eigenvector centralities. A connectivity matrix constructed using intra-sentence cosine similarity which is then used as the adjacency matrix of the graphical representation of sentences. In [9], the centrality measure is used in context of sentences as opposed to words in the sentences in other works. Some methods have used a hypergraph structure to maintain information regarding words [10]. *However, these methods were typically developed for static data which makes them ill-suited for dynamic data streams.*

Our contribution: We introduce a scalable, novel method to detect emergent keywords in data streams that is based on the analysis of *dynamic* semantic graphs. In contrast to many other *static* semantic graph-based approaches, we introduce a notion of node dynamic centrality to measure the emergent importance of keywords. We generalize the well known frequency, degree, and eigenvector centralities into corresponding dynamic versions and demonstrate them on streams of noisy data from Twitter related to the Boston marathon bombing as an example of a public security event. (See more experiments on Baltimore protests in [11].) We advocate our *dynamic eigenvector centrality* method to be able to capture meaningful, less noisy, and more interpretable information than frequency based measures. Furthermore, we introduce an algorithm for the data stream summarization and demonstrate it on the same data sets. The summarization approach is based on the minimum weighted set cover algorithm applied on the semantic graph of the dynamically highly ranked keywords. Our goal is to develop an extremely fast and scalable method for high-frequency streaming data. The implementation is available at https://github.com/neela23/detecting_and_summarizing_events.

II. Modeling approach

In the heart of the proposed modeling approach lies a dynamic semantic graph in which the nodes and undirected edges correspond to keywords and co-occurrences of keywords in a stream of documents, respectively. This network is used to rank and extract top-ranked emergent keywords that will also be used in summarization. Semantic graphs are among the most successful approaches that are broadly used for such tasks as keyword extraction and summarization [12], [13], disambiguation [14], and term similarities [15], [16]. Perhaps the most relevant work to our method is the SemanticRank approach [17] that is a modification of PageRank and HITS algorithms. However, to the best of our knowledge, existing work relies on *static* semantic graphs, which do no resolve the problem of extracting *emergent* information in streaming data. In dynamic semantic graphs, nodes and edges can: (a) appear when they previously have not been observed; (b) change their weights that represent the amount of keywords and connection strength, respectively, and (c) disappear when they become obsolete after a certain time.

A traditional way to detect emergent keywords (implemented in a vast majority of industrial systems) is based on different quantities that directly depend on keyword frequencies. Examples include counting bursty keywords that suddenly appear at unusually high rates [18], [19] and a variety of TF-IDF based methods. However, instead of considering the single term quantities in natural text, it is more appropriate to take into account these quantities only when the corresponding terms appear along with other high-frequency (see [20]). Thus, ranking keyword importance using the eigenvector centrality [21] in semantic graphs, we introduce the notion of *dynamic eigenvector centrality* to capture emergent keywords and summarize the trends in a dynamic setting. We also generalize the frequency and degree centralities with similar dynamic versions. However, in many cases, we observe that the eigenvector-based centrality is more illuminating because important keywords are likely to appear with other important keywords [9], [17]. This concept is reflected in our dynamic eigenvector centrality ranking in which the importance of a keyword depends on the importance of co-occurring keywords.

In the streaming setting, we discretize the time line into segments t_i and introduce the dynamic eigenvector centrality ranking that takes into account the normalized eigenvector centralities on P segments back from the current time segment t. We define the slope of an eigenvector centrality

for a keyword k at time segment t as

$$\textbf{slp-ec}_k^{(t)} = \frac{\sum_{t_i \in \{t,\ldots,t-P\}} \left(t_i - \overline{T}\right)\left(\textbf{ec}_k^{(t_i)} - \frac{1}{P}\sum_{i=0}^{P-1}\textbf{ec}_k^{(t_i)}\right)}{\sum_{t_i \in \{t,\ldots,t-P\}} (t_i - \overline{T})^2}, \tag{1}$$

where $\overline{T} = P(P+1)/2$, and $\textbf{ec}_k^{(t_i)}$ is the normalized eigenvector centrality of keyword k at time segment t_i. That is, $\textbf{slp-ec}_k^{(t)}$ is a slope of a fitted linear regression model on normalized eigenvector centralities computed on P time segments. Accordingly, we define the dynamic eigenvector centrality of keyword k at time t as

$$\textbf{dec}_k^{(t)} = \textbf{slp-ec}_k^{(t)} \cdot \textbf{ec}_k^{(t)}. \tag{2}$$

In addition to being easy to compute (which also includes a variety of methods to compute the eigenvector of a semantic graph [22]), weighting the centrality measure with the slope has several important advantages. First, it is not sensitive to missing values that could appear as a result if a keyword has not been used in a particular segment. Second, it is interpretable, which means that a domain expert user who will need to define a threshold to distinguish between emergent and regular keywords can justify the choice.

Summarization of documents in each t_i is performed by choosing a small subset of documents that contain top-ranked keywords. While this approach is not new, we demonstrate that extraction of documents that contain dynamic mutually emergent keywords provides much more relevant information than other comparable approaches. To evaluate the proposed method we compare it with the degree centrality, dynamic degree centrality (in which a similar slope is computed for the degrees of nodes that correspond to keywords), non-dynamic eigenvector centrality, simple frequency ranking for all keywords, and dynamic frequency ranking (in which a similar slope is computed).

III. Algorithms

Introducing dynamic centrality emergent keyword extraction and stream summarization, we also compare it to the dynamic degree centrality, keyword frequency count, and their corresponding non-dynamic versions. In all dynamic versions, similar to Equations (1) and (2), a regression slope is computed for the corresponding centrality indices, and then multiplied by them.

We process a data stream by discretizing it into time segments. The dynamic centralities are computed using the slopes fitted on the last P segments. At each time segment t, we maintain a semantic graph of keywords $G^{(t)} = (V^{(t)}, E^{(t)})$, where $V^{(t)}$ is a set of nodes that correspond to keywords, and $E^{(t)}$ is a set of positive weighted edges that correspond to the number of co-occurrences of two keywords in the same document, i.e., for keywords i and j, there is an edge $ij \in E^{(t)}$ with weight $w_{ij}^{(t)}$ if i and j appear together in $w_{ij}^{(t)}$ documents. If a contribution of a document d to $w_{ij}^{(t)}$ has been done K time segments ago (where K is a parameter determined by the application), the weight of ij will be decreased at time $t+1$. Accordingly, $w_{ij}^{(t)}$ can be increased at time $t+1$ if i and j appear together again. As a result, an obsolete edge can be removed from the graph if its weight becomes 0. Obsolete nodes can also be removed if they become completely disconnected. In other words, $G^{(t)}$ will contain information of $\max(K, P)$ steps back. A degree of node i at time t is denoted by $d_i^{(t)}$. A frequency of a keyword (node) i at time t is denoted be $f_i^{(t)}$.

Below we describe six algorithms we experimented with to extract keywords.

► **Degree centrality** All keywords $i \in V^{(t)}$ are ranked by normalized degrees $d_i^{(t)} / \max_i\{d_i^{(t)}\}$.

► **Dynamic degree centrality** For each keyword $k \in V^{(t)}$, we consider P values $d_k^{(t)}$, $d_k^{(t-1)}$, ... , $d_k^{(t-P-1)}$ to evaluate the slope $\textbf{slp-deg}_k^{(t)}$ (similar to Equation (1)). The dynamic degree centrality is defined as

$$\textbf{dd}_k^{(t)} = \textbf{slp-deg}_k^{(t)} \cdot d_k^{(t)}.$$

► **Frequency centrality** All keywords $i \in V^{(t)}$ are ranked by their frequencies $f_i^{(t)} / \max_i\{f_i^{(t)}\}$.

► **Dynamic frequency centrality** For each keyword $k \in V^{(t)}$, we consider P values $f_k^{(t)}$, $f_k^{(t-1)}$, ... , $f_k^{(t-P-1)}$ to evaluate the slope $\textbf{slp-freq}_k^{(t)}$ (similar to Equation (1)). The dynamic degree centrality is defined as

$$\textbf{df}_k^{(t)} = \textbf{slp-freq}_k^{(t)} \cdot f_k^{(t)}.$$

► **Eigenvector centrality** All keywords $i \in V^{(t)}$ are ranked by the entries of the eigenvector x in solving $A^{(t)}x = \lambda x$, where $A^{(t)}$ is a weighted adjacency matrix of $G^{(t)}$, x is the eigenvector that correspond to the largest eigenvalue of $A^{(t)}$. The normalized centrality index for a keyword i is then defined as $\textbf{ec}_i^{(t)} = x_i / \max_i\{x_i\}$.

► **Dynamic eigenvector centrality** See Equation (2).

In all cases, a positive slope indicates an increase in significance of the keyword while a negative slope shows an opposite trend. Hence, when multiplying by the slopes, the less important words can gradually be removed (if we set up a threshold of importance or use an appropriate insignificant outlier detection method), and there is a boost in value of keywords with high slope. By picking high value keywords, we pick the trending keywords. While optimizing the running time is not the goal of this paper, it is clear that the most computationally intensive part is a computation of the eigenvector, which is a well studied topic [22].

The pseudocode for computing dynamic eigenvector centralities is summarized in Algorithm 1. Two input parameters are the graph $G^{(t-1)}$ from step $t-1$ that will be updated with the current step data $D^{(t)}$. The $D^{(t)}$ is preprocessed with the following steps that are relevant to Twitter data:

(1) convert text to lowercase, (2) remove special characters, (3) lemmatize words, (4) remove html tags, and (5) remove stop words.

1: **procedure** DEC($G^{(t-1)}$, $D^{(t)}$)
2: Initialize $G^{(t)}$ with $G^{(t-1)}$
3: Update $V^{(t)}$ and $E^{(t)}$ by decreasing $w_{ij}^{(t)}$ and dropping obsolete edges and nodes
4: Update $V^{(t)}$ and $E^{(t)}$ by adding new edges and keywords from $D^{(t)}$
5: Compute $\mathbf{ec}_i^{(t)}$ for $G^{(t)}$
6: Rank all keywords by $\mathbf{dec}_i^{(t)}$ for $G^{(t)}$
7: $S \leftarrow$ top-ranked K keywords
8: **return** S and $G^{(t)}$
9: **end procedure**

Figure 1. Algorithm for computing $\mathbf{dec}^{(t)}$

The extracted top-ranked emergent keywords are used to summarize the data stream. The summarization is done by finding a small set of documents that cover the entire set of top keywords (see S in line 7 of Algorithm 1). We formulate the minimum size set cover problem, where S is the set to be covered, and documents are the subsets of keywords that participate in the covering. It is interesting to mention that finding the real minimum number of documents that cover S may not be informative enough because the information can be too compressed. Thus, we decided to use the greedy set cover algorithm [23] that is fast enough but does not compress the summarization too much because of the obvious reasons of poor approximation ratio. In this setting, every document i is associated with a weight

$$c_i = \sum_{k \in S} \text{tf}(k, i),$$

where $\text{tf}(k, i)$ is a frequency of keyword $k \in S$ in document i. The weight of the rest of the keywords is zero. In the greedy algorithm, we repeatedly select document i that minimizes $c_i/|S \setminus C|$, where C is the list of already covered (in previous steps of the greedy algorithm) keywords in S. There could be a situation where the emergent keyword may not be present in the documents of that time segment. In such cases, the algorithm is run until it covers top keywords in that hour. The selected documents represent a summary based on the emergent keywords.

IV. Experiments and discussion

How good is our proposed dynamic eigenvector centrality measure in detecting emergent keywords and summarization? In this section we evaluate our method with Twitter data from the 2013 Boston Marathon attacks and following manhunt for the terrorists as an example of a public safety event characterized by high volumes of Twitter activity and rapidly changing language and emergent terms used to describe unfolding events. (Additional experiments can be found in [11].)

We purchased archived tweets from Gnip, a company that provides access to the full archive of public Twitter data. We used broad search terms to collect tweets in order to create noisy data streams that cover both related and unrelated events. Our data set of 20,385,957 tweets covers seven days from April 15–21, 2013, and was collected with the following search terms: boston, marathon, bomb, blast, explosion, watertown, mit, mitshooting. A logical OR expression was used to filter the terms, which makes the data very noisy on purpose.

We coded major occurrences and changes in events from information published by news outlets. We use the timing of these events as ground truth against which we compare the algorithms discussed in Section III. Before applying each algorithm to the tweet texts (i.e., the maximum 140 character long texts), we followed standard pre-processing procedures, including the removal of stop words, numbers, URLs, and all tweets in a language other than English. We further grouped tweets into one-hour time segments.

To calculate the dynamic versions of each measure, we set $P = 5$. That is, we weighted each measure at time t with the slope of a linear regression fitted to the last five time segments.

Using actual events as ground truth, we conduct two types of experiments. First, as a proof-of-concept, we use time series plots to inspect how well different measures of keyword importance detect emergent ground-truth events. Second, we show summaries of key emergent events created by the minimum size set cover algorithm.

A. Experiment 1: Estimated keyword importance versus ground truth

We identified six key events from time lines published by two news outlets [24], [25]: (1) the detonation of the two bombs at 2:49pm on April 15, (2) the explosion of a fertilizer company at 7:50pm on April 17 in Texas, which was unrelated to the events in Boston, but was briefly thought to be another terrorist attack, (3) the publication of surveillance photos and videos of the two suspects at 5pm on April 18, (4) the death of a MIT police officer at 10:30pm on April 18, (5) the official release of Tsarnaev's name and photo at 7am on April 19, and (6) his capture by police while hiding in a boat in Watertown at around 8:45pm on April 19.

In our first experiment, we compare the six importance measures to the ground truth events. To allow for a direct comparison between the measures, we first replace negative values in the dynamic measures to zero and then normalize each measure to $[0, 1]$. Figures 2 to 5 show the ground-truth time lines together with the importance measures for six selected keywords most closely related to the actual events: "explosion", "texas", "photo", and "tsarnaev". (Additional

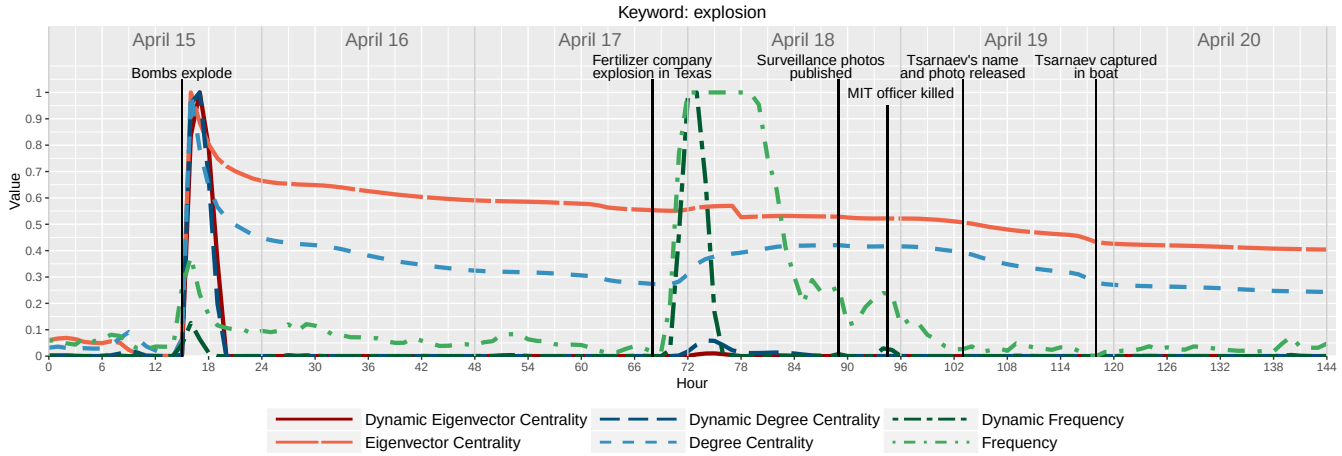

Figure 2. Timeline of actual events during Boston Marathon attacks together with importance measures for keyword "explosion".

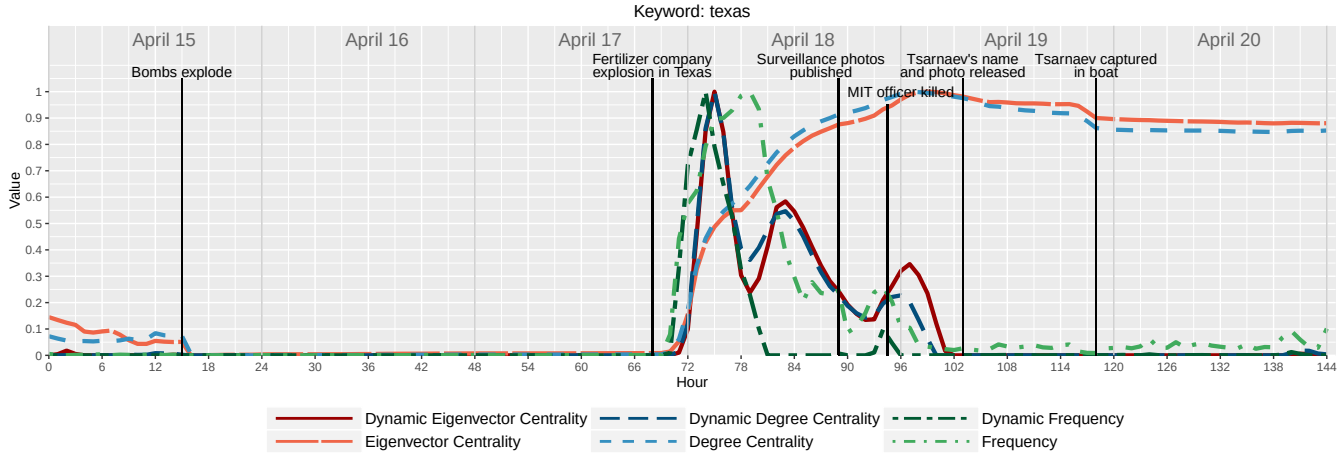

Figure 3. Timeline of actual events during Boston Marathon attacks together with importance measures for keyword "texas"'.

plots for "mit" and "boat" can be found in report [11] because of space constraints.)

The following conclusions can be drawn from the these figures: **(a)** As a proof-of-concept, the figures show that keyword-based dynamic centrality measures are extremely capable in detecting emergent events. In all cases, large spikes in the dynamic measures closely follow corresponding ground truth events; and **(b)** The dynamic measures are superior to their static versions when it comes to labeling keywords as emergent. For example, both dynamic eigenvector centrality and dynamic degree centrality for keyword "explosion" sharply increase shortly after the explosion of the two bombs, but then—and in contrast to their static counterparts—decrease keyword importance to zero.

B. Experiment 2: Tweet summary of emerging events

Having established that dynamic eigenvector centrality is a suitable measure to detect emergent events, we use it to generate a summary of the data stream. For each hour, we select the top 20 ranked keywords and then apply the greedy set cover algorithm [23] discussed in Section III to find the smallest number of tweets that cover the 20 keywords. The result is a set of documents that represent a summary of the stream based on the emergent keywords.

Table I shows the top 20 keywords and summaries for the two hours covering the explosion during the Boston Marathon at 2:49pm. The keywords and summaries for the first hour are dominated by conversations around the marathon winners Lelisa Desisa and Rita Jeptoo, with their countries and words such as "win", "won", and "winner", but also include the keyword "explosion" and a correspond-

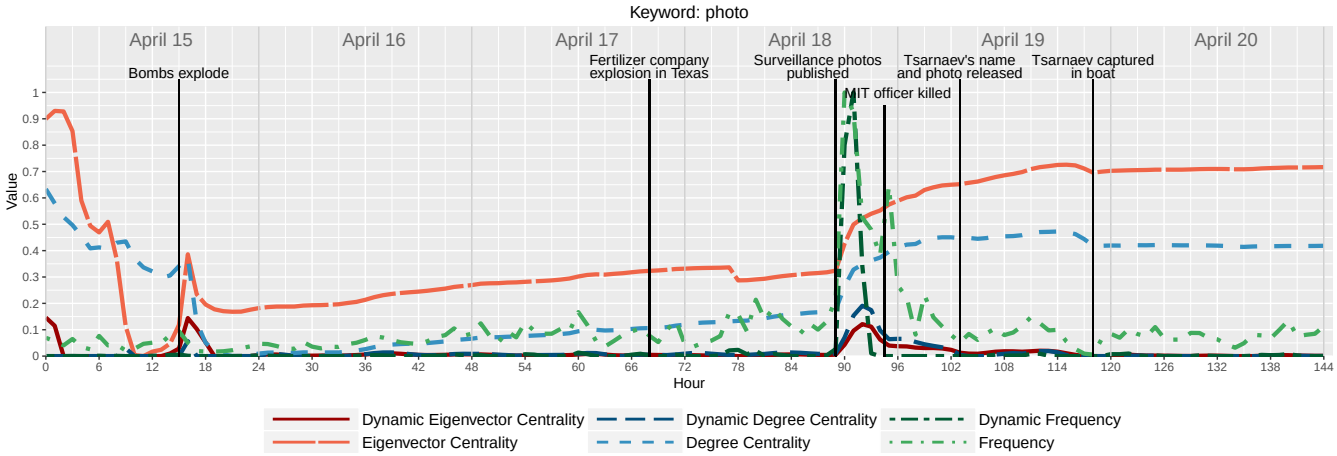

Figure 4. Timeline of actual events during Boston Marathon attacks together with importance measures for keyword "photo"'.

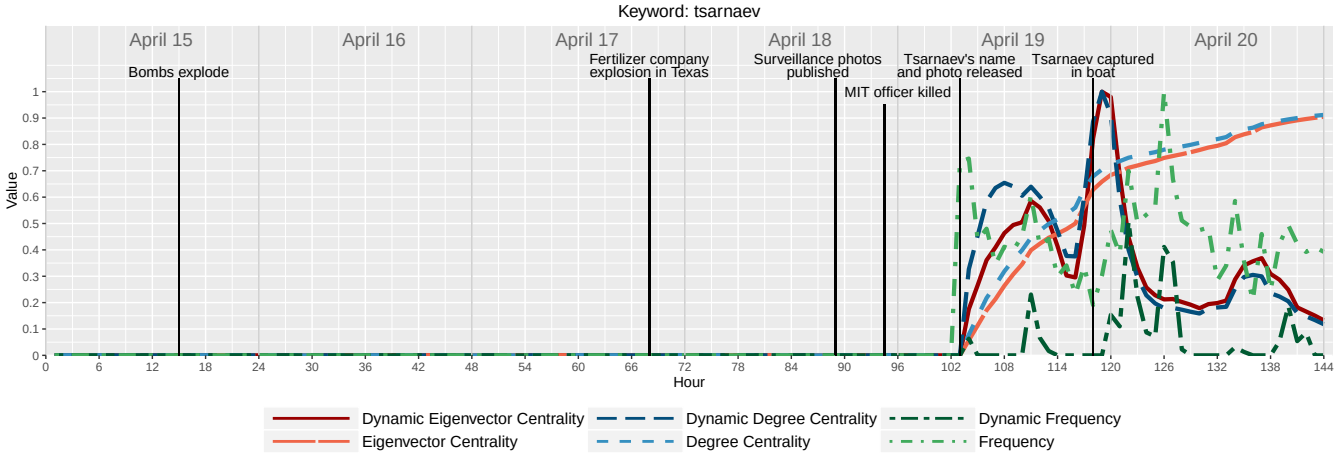

Figure 5. Timeline of actual events during Boston Marathon attacks together with importance measures for keyword "tsarnaev".

ing tweet ("Explosion! [url]"). In the following hour, the keywords and summaries shift to language describing the emergent event ("breaking", "news", "explosion", "bomb"), its location ("Boston", "finish", "line"), and offering condolence ("prayer", "praying", "thought"). (See [11] for a longer report.)

Table II provides a second example of keyword selection and stream summary for the two hours covering the capture of Tsarnaev. The tweet summary clearly captures the event, including tweets such as "'It's over' - CNN #boston" or "They finally got that boy. I know Boston feelin good now."

V. Conclusions

We introduced a dynamic eigenvector centrality measure for dynamic semantic graphs and a generalization of static semantic graph frequency and degree centrality measures into corresponding dynamic versions with applications in emergent keyword detection and data stream summarization. The proposed methods are extremely fast and scalable as graph Laplacian eigensolvers.

Using Twitter data from the 2013 Boston Marathon attacks together with coded ground-truth events, we observed that our novel dynamic centrality indices successfully detect emergent keywords and provide concise and meaningful summarization. A promising future research direction is adapting these methods into smooth stream processing and summarization (instead of discretization) in which the summary elements will not be repeated in the next few time segments if a similar information has been detected and summarized in the previous time step.

Acknowledgments

We thank the Clemson CBBS One-Year Accelerate grant program for providing funds for this research.

Table I
TOP 20 KEYWORDS AND TWEET SUMMARIES BASED ON DYNAMIC EIGENVECTOR CENTRALITY DURING AND AFTER THE EXPLOSIONS AT 2:49PM.

HOUR: 2:00–3:00PM
Top 20 keywords: boston, win, desisa, lelisa, finish, run, rt, jeptoo, men, ethiopia, won, rita, woman, mile, time, explosion, ha, kenya, winner, race **Tweet summaries:** "corrib road race marathon monday", "Friendship Blast Contest Winner's Photoshoot!! [url]", "Too much rain in Kenya, i will resume exercise tomorrow ! 'Marathon '", "Officially on the last Harry Potter film... This marathon has shown me I have emotions I didn't even know existed #hpmarathontroops", "Explosion! [url]", "First time I've worked Marathon Monday in the 21st Century. No sir, I don't like it.", "My sister just asked me to do a 26 mile marathon with her in September. Does she even know who I am", "Had a blast working on this beauty! Stay tuned for the video! Like us on FB Art Of A Woman [url]", "Rita Jespoo is proof that mat leave is awesome [url] #Bostonmarathon", "He won marathon Monday [url]", "Marathon and other victories by our athletes in many world cities are purely organic, natural. No doping... #Oromo #Oromia #Ethiopia", "We had a blast at Homecoming on the Hill 2013! Here's a great photo of all those who participated in the Men's... [url]", "RT @andrewbensonf1: BBC News - Car blast in Bahrain heightens F1 security concerns [url]", "Surround yourself w/ppl who push you. I have no desire to run a full marathon but I am inspired to push myself beyond what I think I can do.", "The fact that my aunts just got VIP passes for the marathon, see you at the finish line @DColl15, legit", "@Yusufdido @robjillo Lelisa means 'someone who desires' in Afan Oromo in Oromia (and he desired the Marathon and he got it)", "Rita Jeptoo wins in 2:26:25 #bostonmaraton. Again this would be maybe what I could do a half marathon time in. Crazy fast", "@kpfallon obvs. I mean, Lelisa Desisa and I have so many similarities, winning only our second marathon will be just one.", "I posted 7 photos on Facebook in the album '2013 Boston Marathon' [url]"
HOUR: 3:00–4:00PM
Top 20 keywords: explosion, finish, line, boston, prayer, people, thought, news, injured, rt, reported, praying, breaking, hope, happened, bomb, affected, report, bombing, safe **Tweet summaries:** "Thanking god my aunts and uncle are safe at the marathon. #prayforboston", "What do you get out of bombing a marathon someone please tell me. #PrayForBoston", "#Boston-marathon Early report in the times: [url]", "The marathon this year was dedicated to everyone affected by newtown. It's sick someone could do that.", "@murneybhoy @saintturbo if I never saw another smoke bomb, gimp mask upside down protest banner again @ our stadium i would rejoice!", "Omg and my daddy was gonna run in that marathon!!! I would have died if anything happened to him... Daddy's girl", "I hope my friend Irene is okay at the Marathon", "My aunt, who is 45 with MS is running in the marathon and it's breaking my heart that this is happening.", "#prayersforboston praying for all who's at the marathon.", "@GeorgeSandeman Casualties reported now [url] F***ing hell :(", "RT @DaveWedge: ABC reporting 2 dead at marathon; dozens hurt; source tells me FBI counterterrorism team from NYC en route to #BostonMarathon", "Pray for all those who are injured at the marathon and pray that the scum who did it rots in hell", "@HannahLuke23 heard the news about the marathon, are you alright?", "Thoughts go out to the victims in the Bodton Marathon attack.", "Terrorist attack at a marathon people raising money for good causes what's wrong with the world.", "Prayers for those at home and involved with the marathon.", "Boston...damn :(", "Its a directed blast both went off with the blast directed at the finsh line", "@ITK_AGENT_VIGO Looks like someone has set off a semtex Catherine wheel at the finish of the marathon.God will help the Yanks,he always does", "@GreenDayTilIDie lot of bad injuries. They saying more than one explosion"

Table II
TOP 20 KEYWORDS AND TWEET SUMMARIES BASED ON DYNAMIC EIGENVECTOR CENTRALITY DURING AND AFTER TSARNAEV'S CAPTURE AT 8:45PM.

HOUR: 8:00–9:00PM
Top 20 keywords: suspect, police, bombing, rt, news, watertown, ha, custody, bomber, breaking, manhunt, cnn, fbi, guy, area, officer, live, shot, terror, scanner **Tweet summaries:** "Congratulations going around on the Boston PD scanner. And well deserved.", "i wonder if the hunger strikers in solitary confinement at guantanamo bay know about the alleged terror attacks in boston... prolly not aye", "Sure is a lot of shooting going on in Boston-Bin Laden did not get shot at that much-strangethings going on in Boston-I don't buy into it.", "@Sploops Go to WCVB TV Boston's live feed. They have been very good. Also WBUR radio.", "Do they have a suicide prevention officer there to talk to him? I hope so. Boston", "'@OnlyInBOS: Tomorrow is 420, which the Boston area really, really needs after this week.' My point exactly smh", "There's boys running round East Belfast have done worse shit than this guy, don't see Belfast being locked down anytime soon #boston", "Good job, Boston PD, FBI et al. Deep breaths, deep breaths, everybody.", "'It's over' - CNN #boston", "They found the 2nd dude that bomb Boston /. #manhunt", "[ALERT] Boston Mayor Tom Menino says on twitter 'we got him' — Reuters #Breaking", "#blowthatboatup #boston #bostonmarathon #bombers #bp #america #staystrong [url]", "They have him in custody!!! #BOSTON", "@KF***INGP after all Boston has been through the past few days... Give them a break Kenny! Hahaha", "YES!!!!! Take that stupid terrorist, you where told that we will find you and we did!!! #watertown", "@FitzTheReporter Local news in Boston says so.", "RT @PrincessProbz: god bless america, god bless boston, god bless all the victims their family members. you are all in our prayers.", "Thoughts and Prayers to: the victims of the Boston Bombings and to the victims of irresponsible gun laws and policies in the U.S", "@Boston_Police amazing job", "The Boston suspect 'captured' was found ALIVE hiding in a waste container! No mames"
HOUR: 9:00–10:00PM
Top 20 keywords: suspect, police, bombing, rt, custody, watertown, news, ha, bomber, breaking, manhunt, fbi, officer, guy, terror, area, cnn, job, bostonstrong, good **Tweet summaries:** "They finally got that boy. I know Boston feelin good now.", "Tales at #Boston didn't have to invade #Iraq #Boston-Strong #USA #merica", "So many props and love for all the law enforcement in #Boston, amazing job.", "MT @josh_levin CNN's Susan Candiotti: 'Streets are empty. It's eerie. It's as though [pause, conjure metaphor] a bomb had dropped somewhere'", "Soldiers in Boston area streets are Massachusetts National Guard who proudly trace their roots to the Minutemen.", "Boston bombs: Obama lulled America into false confidence over terror threat [url]", "Something about chanting USA doesn't seem appropriate. This guy was an American student, but I sure am happy they caught the SOB. #Boston", "@WillSasso A lot of those officers were national, not from Boston", "After cheering subsides in #boston, expect some serious questions to be asked about earlier investigation of #bombingsuspect. #fbi", "Final thought: I feel like this Boston manhunt prevented us from really being able to talk about that new Daft Punk single.", "Releasing photos a risk, but pivotal in breaking #Boston case. [url] #BostonManhunt", "Glad to see they caught the Boston Bomber. Now I can go back to posting and tweeting about nonsense and less important things.", "My thoughts with the families who had lost their loved ones. Justice tonight in boston has been served", "Boston PD press conference coming up at 9:30. Tune in to CBS 13 News", "Cheers @KellyMacFarland who's been on lockdown right next to the shootout in Watertown. Go catch one of her shows and send her a drink!!!", "Dzhokhar Tsarnaev is lucky he is in Boston and not Los Angeles. LAPD could take some lessons from BPD! So glad that he is in custody alive.", "RT @msdebbieallen: My Heart goes out to our families in Boston and Texas. President Obama said it well ... [url]", "'@gherpich99: This Boston bombing story is crazy!'", "Bravo, Boston Police Department!! Bravo!!! #bostonpolice", "WashPo give the skinny on the fate of the Boston bomb suspects [url]"

REFERENCES

[1] N. G. Golbandi, L. K. Katzir, Y. K. Koren, and R. L. Lempel, "Expediting search trend detection via prediction of query counts," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, ser. WSDM '13. New York, NY, USA: ACM, 2013, pp. 295–304.

[2] A. Dong, Y. Chang, Z. Zheng, G. Mishne, J. Bai, R. Zhang, K. Buchner, C. Liao, and F. Diaz, "Towards recency ranking in web search," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 11–20.

[3] M. Mathioudakis and N. Koudas, "Twittermonitor: Trend detection over the twitter stream," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '10. New York, NY, USA: ACM, 2010, pp. 1155–1158.

[4] J. Benhardus and J. Kalita, "Streaming trend detection in Twitter," *International Journal of Web Based Communities*, vol. 9, no. 1, pp. 122–139, 2013.

[5] A. Saha and V. Sindhwani, "Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 693–702.

[6] J. Allan, *Topic detection and tracking: event-based information organization*. Springer Science & Business Media, 2012, vol. 12.

[7] Y. Matsuo, Y. Ohsawa, and M. Ishizuka, "Keyworld: Extracting keywords from document s small world," in *International Conference on Discovery Science*. Springer, 2001, pp. 271–281.

[8] U. Brandes and D. Fleischer, "Centrality measures based on current flow," in *STACS*, ser. Lecture Notes in Computer Science, V. Diekert and B. Durand, Eds., vol. 3404. Springer, 2005, pp. 533–544.

[9] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.

[10] A. Bellaachia and M. Al-Dhelaan, "Hg-rank: A hypergraph-based keyphrase extraction for short documents in dynamic genre," in *4th workshop on making sense of microposts (# Microposts2014)*. Citeseer, 2014, pp. 42–49.

[11] N. Avudaiappan, A. Herzog, S. Kadam, Y. Du, J. Thatcher, and I. Safro, "Detecting and summarizing emergent events in microblogs and social media streams by dynamic centralities," *arXiv:1610.06431*, 2016.

[12] M. Litvak and M. Last, "Graph-based keyword extraction for single-document summarization," in *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*. Association for Computational Linguistics, 2008, pp. 17–24.

[13] J.-Y. Yeh, H.-R. Ke, and W.-P. Yang, "ispreadrank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1451–1462, 2008.

[14] M. Sussna, "Word sense disambiguation for free-text indexing using a massive semantic network," in *Proceedings of the second international conference on Information and knowledge management*. ACM, 1993, pp. 67–74.

[15] A. Budanitsky and G. Hirst, "Evaluating wordnet-based measures of lexical semantic relatedness," *Computational Linguistics*, vol. 32, no. 1, pp. 13–47, 2006.

[16] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis." in *IJcAI*, vol. 7, 2007, pp. 1606–1611.

[17] G. Tsatsaronis, I. Varlamis, and K. Nørvåg, "Semanticrank: ranking keywords and sentences using semantic graphs," in *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 1074–1082.

[18] M. Mathioudakis and N. Koudas, "Twittermonitor: trend detection over the twitter stream," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 1155–1158.

[19] S. Goorha and L. Ungar, "Discovery of significant emerging trends," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 57–64.

[20] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Computational Intelligence*, vol. 31, no. 1, pp. 132–164, 2015.

[21] P. Bonacich, "Some unique properties of eigenvector centrality," *Social networks*, vol. 29, no. 4, pp. 555–564, 2007.

[22] O. E. Livne and A. Brandt, "Lean algebraic multigrid (lamg): Fast graph laplacian linear solver," *SIAM Journal on Scientific Computing*, vol. 34, no. 4, pp. B499–B522, 2012.

[23] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of operations research*, vol. 4, no. 3, pp. 233–235, 1979.

[24] Ann O'Neill, CNN, "Tsarnaev trial: Timeline of the bombings, manhunt and aftermath," http://www.cnn.com/2015/03/04/us/tsarnaev-trial-timeline/, 2015, last updated on May 15, 2015. Last accessed on October 17, 2016.

[25] S. Morrison and E. O'Leary, "Timeline of boston marathon bombing events," https://www.boston.com/news/local-news/2015/01/05/timeline-of-boston-marathon-bombing-events, 2015, last updated on January 5, 2015. Last accessed on October 17, 2016.