

# Event Detection and Summarization using Phrase Network

Sara Melvin<sup>1</sup>, Wenchao Yu<sup>1\*</sup>, Peng Ju<sup>1</sup>, Sean Young<sup>2</sup>, and Wei Wang<sup>1\*</sup>

<sup>1</sup>Department of Computer Science, UCLA

<sup>2</sup>University of California Institute for Prediction Technology, UCLA

**Abstract.** Identifying events in real-time data streams such as Twitter is crucial for many occupations to make timely, actionable decisions. It is however extremely challenging because of the subtle difference between “events” and trending topics, the definitive rarity of these events, and the complexity of modern Internet’s text data. Existing approaches often utilize topic modeling technique and keywords frequency to detect events on Twitter, which have three main limitations: 1) supervised and semi-supervised methods run the risk of missing important, breaking news events; 2) existing topic/event detection models are base on words, while the correlations among phrases are ignored; 3) many previous methods identify trending topics as events. To address these limitations, we propose the model, PhraseNet, an algorithm to detect and summarize events from tweets. To begin, all topics are defined as a clustering of high-frequency phrases extracted from text. All trending topics are then identified based on temporal spikes of the phrase cluster frequencies. PhraseNet thus filters out high-confidence events from other trending topics using number of peaks and variance of peak intensity. We evaluate PhraseNet on a three month duration of Twitter data and show the both the efficiency and the effectiveness of our approach.

**Keywords:** Event Detection, Phrase Network, Event Summarization

## 1 Introduction

It has been of interest for many years to have an automated tool to alert and summarize newsworthy events in real-time. Identifying events in real-time is crucial for many occupations to make timely, actionable decisions. It is shown to be extremely challenging to identify these events because of the subtle difference between “events” and trending topics, the definitive rarity of these events, and the complexity of modern Internet’s text data. Existing approaches often utilize topic modeling technique and keywords frequency to detect events on Twitter, which have three main limitations:

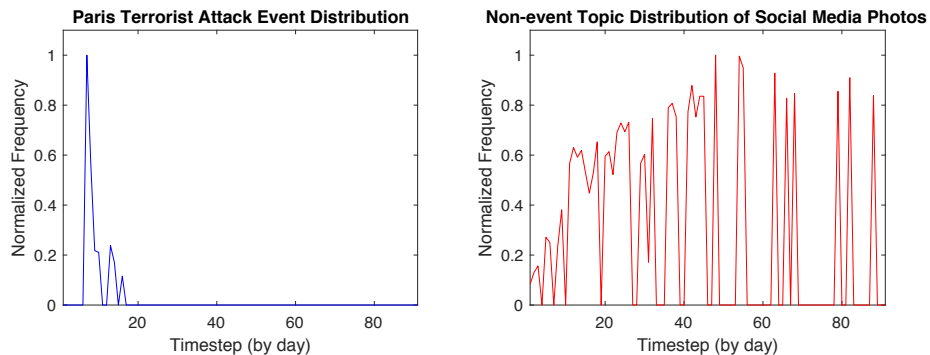
1. Supervised and semi-supervised methods run the risk of missing important, breaking news events [3, 5, 10, 12–14]. These methods share one common

---

\* Corresponding authors.

weakness, they rely on the seeding of keywords for their tool or human labeling of tweets to train their models. This approach runs the risk of missing some events since their model is scoped to identify only events that fall under their static list of keywords.

2. Many previous methods mistakenly identify trending topics as events [8, 11], however the description of an “event” is a unique sub-component to all “topics”. Figure 1 shows the difference between event distribution (Paris terrorist attack) and topic distribution (discussion of social media photos).
3. Existing methods [1, 19] summarize their results with a small grouping of keywords that do not convey enough information for a user to know in real-time what occurred. These models are also base on unigram words, while the correlations among phrases are ignored.



**Fig. 1.** A comparison between the distributions of an event and a topic. This figure shows the normalized frequency distribution between a non-event topic discussion of social media photos (right) and the event distribution describing the Paris Terrorist Attack (left) at the offices of Charlie Hebdo.

To address the above limitations, we propose PhraseNet, a model for event detection using phrase network. Our method begins by extracting the high-frequency phrases from tweets. Each frequent phrase and relationship between phrases are then represented in a phrase network. A community detection algorithm is applied to the phrase network to identify a grouping of phases which we define as event candidates. Finally, the high-confidence events can be identified by three criteria extracted from the event candidate distributions over time: 1) number of peaks in distribution, 2) intensity of peaks and 3) variance of the distribution.

Defining the unique features of an event is key in designing an event detection model. Consider an event such as the Paris terrorist attack on the offices of Charlie Hebdo. As you can see in Figure 1, the words to describe the event spike in a collective frequency on the day of the attack with only a couple of peaks post event. In contrast, words used in the discussion of the non-event topic of

social media photo opinions spike in frequency during several different time steps throughout the data. Therefore, the characteristic of an event’s distribution is defined to have very few peaks because an event description is usually unique; not normally shared by many other events.

In addition, non-event topics are discussed by the masses rises and falls with similar frequency throughout time because of the common interest in such topics stays fairly consistent. However, events are discussed during the occurrence and post-event to discuss opinions about the event or, if the events are planned, events can be discussed prior in anticipation. These event peaks that occur prior- and post-event will be small in frequency compared to the moment the event occurs, therefore the standard deviation of an event’s peak intensity will be larger than a non-event topic because of the varied interest in discussing the event. As you can see in Figure 1, the Paris attack was not planned, there will be no peaks prior to the event occurrence.

Finally, our method, PhraseNet, leverages phrases and graph clustering to group correlated phrases together and help give more context to the identified event. You will see in Section 4.3 how PhraseNet summarizes compared to Twevent.

In summary, our contributions in this paper are:

1. *Event detection using phrase network*: We proposed the PhraseNet model to detect and summarize events on Twitter stream which includes three steps: 1) building phrase network using high-frequency phrases extracted from tweets, 2) detecting event candidates using community detection algorithm on phrase network, 3) identifying high-confidence events from candidate set using criteria such as number of peaks and variance of peak intensity in the event candidate distributions.
2. *Event summarization with phrases*: The proposed model summarizes events with phrases to give an interested user a short description and time duration of the detected event.
3. *Empirical improvements over Twevent*: We evaluate the PhraseNet model on a three month duration of Twitter data, and show that PhraseNet outperforms the baseline Twevent [9] by a large margin, which demonstrates the effectiveness of our model.

## 2 Problem Definition

In this section, we formally define a *phrase* as a sequence of contiguous tokens [6]:

$$p_m = \{w_{d,i}, \dots, w_{d,i+n}\}, i + n \leq \mathcal{N}_d \quad (1)$$

where  $w_{d,i}$  is a word (a.k.a. token) in the  $i$ -th place of the document  $d$ ;  $n \geq 0$ . The  $d$ -th document is a sequence of  $\mathcal{N}_d$  tokens. A *topic* consists of a set of phrases  $\mathcal{P} = \{p_1, \dots, p_k\}$  where  $p_m$  is a phrase and  $k$  is the total number of phrases in the set ( $m \in [1, k]$ ).

A *sliding window*,  $T$ , consists of  $\tau$  amount of *time steps*,  $t$ . As the sliding window moves along, a sliding window *mean*,  $\mu_T$ , and the sliding window *standard deviation*,  $\sigma_T$  are calculated as follows:

$$\mu_T = \frac{1}{\tau} \sum_{t=1}^{\tau} \left( \sum_{m=1}^k \mathcal{F}(p_m^{(t)}) \right) \quad (2)$$

$$\sigma_T = \frac{1}{\tau} \sum_{t=1}^{\tau} \left( \sum_{m=1}^k \mathcal{F}(p_m^{(t)}) - \mu_T \right)^2 \quad (3)$$

where  $\tau$  is the number of time steps within the sliding window and  $\mathcal{F}(p_m^{(t)})$  is the frequency of phrase  $p_m$  at time step  $t$  in the sliding window  $T$ .

A *trending topic*, or an *event candidate*, is identified by a peak in topic phrase frequency above a certain standard deviations from the topic’s mean. Therefore, the peak is defined as:

$$\frac{\sum_{m=1}^k (\mathcal{F}(p_m^{(t)})) - \mu_T}{\sigma_T} > \theta \quad (4)$$

where  $\theta$  is user-specified threshold. Therefore, an *event* is an unique subset of *trending topics*, or *event candidates*, that is formally defined in this method as a phrase cluster with very few peaks ( $\leq \alpha$ ), a high frequency intensity of a peak ( $\geq \beta$ ), and the largest standard deviation in peak height ( $\geq \chi$ ).

### 3 Approach

#### 3.1 Creating the Phrase Network

As mentioned in Section 2, to identify these phrases, the ToPMine algorithm [6] was used to identify the frequent phrases for a certain unit of time (e.g. an hour)  $t$  and to partition each tweet into a combination of frequent. ToPMine algorithm includes two phases: 1) parse all the words into text segments; 2) create a hashmap of phrases and recursively merge if phrases appear frequently enough together.

The second phase is a bottom-up process that results in a partition on the original document that, when completed, creates a “bag-of-phrases.” For example, the following tweet: *american sniper wins for putting bradley in that body #oscars2015*. Would be partitioned with the following phrases with a minimum support of 50: *american sniper*, *bradley*.

Now each frequent phrase found is considered a node in a graph. The edges between each frequent phrase reflect the co-occurrence of the phrases in the same tweet. The weight to the edge,  $w_e$ , is the Jaccard coefficient defined as  $w_e = \frac{\mathcal{F}(p_a \wedge p_b)}{\mathcal{F}(p_a) + \mathcal{F}(p_b)}$ , where the edge connects the phrases  $p_a$  and  $p_b$ .

To calculate the most frequent co-occurring phrase pairs efficiently, the FP-Growth algorithm [7] was used. In this research, brute force scanning and tallying up co-occurrences became a bottleneck in PhraseNet, however, the FP-Growth exhibited the speed necessary to keep PhraseNet a real-time algorithm.

### 3.2 Phrases Clustering

After the graph is constructed, it is clustered into communities of phrases using the Louvain community detection method [2], which maximizes the modularity. The clusters identified by this method are event candidates. Hence, output for this stage is the set of event candidates  $\Xi = \{\mathcal{P}_1, \dots, \mathcal{P}_c\}$  where  $c$  is number of event candidates in all time steps. The details are shown in Algorithm 1.

---

#### ALGORITHM 1: Phrase network construction and event candidate detection

---

**Data:** Frequent patterns of phrases  $P = \{p_i, \mathcal{F}(p_i)\}$

**Result:** List of event candidates,  $\Xi = \{\mathcal{P}_1, \dots, \mathcal{P}_c\}$

```

1 Graph G=(V,E)
2 for  $p_i, \mathcal{F}(p_i)$  in  $P$  do
3   for  $p_a, p_b$  in  $p_i$  where  $a \neq b$  do
4     if  $p_a \notin V$  then
5        $V = V \cup p_a$ 
6     if  $p_b \notin V$  then
7        $V = V \cup p_b$ 
8      $e = (p_a, p_b)$ 
9      $e.\text{weight} = \mathcal{F}(p_a \wedge p_b) / (\mathcal{F}(p_a) + \mathcal{F}(p_b))$ 
10     $E = E \cup e$ 
11  $\Xi = \text{LouvainClustering}(G)$ 
12 return  $\Xi$ 

```

---

### 3.3 Merging Event Candidates Across Time Steps

Since events could potentially carry on beyond the set time interval, each event candidate  $\mathcal{P}_i$  is measured against the other event candidates of the next time step to measure whether the two event candidates should merge. The criteria used to determine the merge is the similarity score defined by Eq.(5). If the two event candidates with the highest score have a score greater than a threshold (we set 0.5 in this paper), then the event candidates will merge.

$$\text{similarity} = \max \left( \frac{\sum_{p_s \in (\mathcal{P}_{i,t} \cap \mathcal{P}_{i,t+1})} w_s}{\sum_{p_r \in \mathcal{P}_{i,t}} w_r}, \frac{\sum_{p_s \in (\mathcal{P}_{i,t} \cap \mathcal{P}_{i,t+1})} w_s}{\sum_{p_j \in \mathcal{P}_{i,t+1}} w_j} \right) \quad (5)$$

For each time interval there is a set of phrase,  $\mathcal{P}$  at time step  $t$ . Each phrase,  $p_m$  has a weight,  $w_m$  associated with it that will be normalized by the total number of phrases in the time interval  $t$ , denoted as  $n$  in the equation below.

$$w_m = \frac{\mathcal{F}(p_m)}{\sum_{i=1}^n \mathcal{F}(p_i)} \quad (6)$$

On completion of merging there remains a set of unique event candidates are maintained through all time steps. The event candidate distribution over

time is created by defining the frequency of the phrase cluster over each time step. The frequency of a phrase cluster  $\mathcal{P}$  will be denoted as  $\mathcal{F}(\mathcal{P})$ . Therefore,  $\mathcal{F}(\mathcal{P}) = \sum_{m=1}^k w_m$  which is the sum of all phrase weights contained in the phrase cluster that make up  $\mathcal{P}$ .

### 3.4 Peak Detection

PhraseNet identifies potential events by first identifying the trending topics. Trending topics are discussions on a subject that becomes, all of a sudden, popular. To define “all of a sudden,” the z-score was used to calculate the phrase cluster frequency,  $\mathcal{F}(\mathcal{P})$ , is  $\theta$  standard deviations above the sliding window mean,  $\mu_t$ . The z-score was used to better identify peaks in a noisy environment. For example, a planned event may be discussed in advance thus showing a  $\mathcal{F}(\mathcal{P}) > \mu_t$ , however, these discussions are only small bumps compared to the height of the phrase community on the day of the planned event. To clarify the day and the duration of the event, whether planned or not planned, z-score helps filter the larger spikes in frequency compared to the small bumps.

Some events last longer than a time step, therefore, the sliding window average is updated as it slides, however a damping coefficient,  $\omega_t$ , is used to weight the phrase communities’ peak. Therefore, the sliding window average shown in Eq.(2) is updated as follows:

$$\mu_T = \frac{1}{\tau} \sum_{t=1}^{\tau} \omega_t \left( \sum_{m=1}^k \mathcal{F}(p_m^{(t)}) \right) \quad (7)$$

where  $\omega_t$  is zero for non-peak topic time steps and during peak time intervals of a topic the coefficient is  $0 \leq \omega_t \leq 1$  where  $\omega_t \in \mathbb{R}$ . The exact definition of  $\omega_t$  is a parameter for the user to define.

Finally, to focus on event candidate peaks, all time steps where the phrase community did not show a peak, their phrase community frequency is lowered to zero, however, all peak identified time steps maintain the phrase community frequency,  $\sum_{m=1}^k \mathcal{F}(p_m^{(t)})$ . This filtering is shown in Figure 1.

Lastly, all event candidates are held to a certain threshold of key features and then sorted: the least number of peaks ( $\alpha_i > \alpha_j$  where  $i \neq j$ ), the largest standard deviation of peak heights ( $\beta_i < \beta_j$  where  $i \neq j$ ), and the highest peak intensity ( $\chi_i < \chi_j$  where  $i \neq j$ ). The last feature ( $\chi$ ) is used to merely sort between the most popular phrase groups to aid in identifying the most urgent events. The first  $\gamma$  of the event candidates are considered **events**. Each event that has a peak on the same day as another event are joined together for a total summary of the time step occurrences.

## 4 Results

It will be shown in this section how accurate and quick PhraseNet identifies events in comparison with Twevent.

## 4.1 Data and Parameters

Data was collected using Twitter’s REST API<sup>1</sup> for the time period of January 1, 2015 to March 31, 2015. The sliding window for each time step was set for 24 hours, from midnight to midnight. The experiment dataset only used English tweets thus using a total of 2,747,808 tweets. Each tweet was preprocessed to expand all contractions, all non-English characters were removed, and all stop words were removed.

The ToPMine algorithm uses the minimum support of 40 to find all frequent phrases and phrases were given a limit to search no more than 5-gram. In addition, the FP-Growth algorithm used a minimum support of 8. The  $\theta$  value was placed at a 3, which means all event candidate peaks are identified as more than 3 standard deviations above the sliding window mean. The dampening coefficient,  $\omega_t$ , weight was defined as 0.1 and the allowed window of time for a true positive event peak to occur consisted of the true event date  $\pm 5$  days. Lastly, the event key feature thresholds are the following:  $\alpha = 10$ ,  $\beta = .05$ , and  $\chi = .5$ .

## 4.2 Experiment and Evaluation

Since ground truth was not available for this dataset, ground truth was defined from the “On This Day” website<sup>2</sup> and by various other reliable news sources. From the “On This Day” website, all events were filtered to only include English speaking country events (i.e. United States, England, Australia, Canada, and New Zealand) and terrorist attacks. In addition, all national holidays celebrated by the United States, U.K., Australia, Canada, and New Zealand identified in Wikipedia were added to the ground truth. Lastly, all sports related events were found via ESPN, BBC Sport, or NFL websites. Under this definition of ground truth, there are 102 events in total.

A sampling of true positives found by PhraseNet are listed in the table found in Table 1. This table exhibits the correlation of sub-events identified by peaks within the same time step. For example, the Grammy Awards are described by PhraseNet with some of the winners’ names and included the word “Kanye” and “Beyonce” to note the fact that Kanye, again, interrupted a Grammy winner’s speech to stick up for his friend Beyonce.

Considering the ground truth for identifying and labeling all true positives, false positives, and false negatives, it is impossible to determine every event that occurred within the data time frame, therefore this research uses the metrics of precision and recall. To show the trade off between precision and recall, the  $F_1$  score is also provided for comparison. Precision is defined as the number of event candidates that correlate to known events divided by total number of event candidates. Recall is defined as the number of unique events detected divided by the total number of events possible listed in the ground truth. The final performance of PhraseNet is shown in Figure 2 and detailed further in Table 2.

<sup>1</sup> <https://dev.twitter.com/rest/public>

<sup>2</sup> <http://www.onthisday.com/events/date/2015>

**Table 1.** A sampling of events identified and summarized by PhraseNet.

Date of Event Detected	Description of Event	PhraseNet Phrase Set
January 1, 2015	Steven Gerrard announced he will be leaving the Liverpool soccer team at the end of the season	gerrard, steven
January 5, 2015	ESPN longtime host, Scott Stuart, died at the age of 49	espn, sportscenter, stuart scott, rip
January 7, 2015	Terrorist attack at a newspaper office, Charlie Hebdo, in Paris, France	charlie, hebdo, paris, attack, jesuischarlie, charliehebdo
January 11, 2015	72nd Golden Globes where George Clooney won a lifetime achievement award	clooney, george
January 15, 2015	Oscar Nominations are Announced	oscar, nominations
January 15-19, 2015	Pope Francis visits the Philippines for the first time in 20 years	francis, philippines, pope
January 21, 2015	Barack Obama gives the State of the Union (sotu) speech	union, state, sotu, address, president, barackobama, obama
January 24, 2015	Golden State Warrior scores the most NBA points and the most 3-pointers in a quarter	quarter, point
January 24, 2015	FA Cup in the 4th Round	cup, fa
January 25, 2015	WWE Royal Rumble	rumble, royal, royalrumble, wwe
January 31, 2015	Anderson Silva vs. Nick Diaz UFC 183 Fight	silva, diaz
February 1, 2015	103rd Men's Australian Open where Novak Dokovic defeats Andy Murray	murray, andy
February 8, 2015	Grammy Awards ceremony where "Stay With Me" by Sam Smith won best song, Beck was given Album of the Year, and Kanye West almost interrupts Beck's speech to argue that Beck's award should go to Beyonce	grammys, give, win, year, brits, kanye, west, beyonce, congrats, show, live, performance, watch, beck, awards, ago, pharrell, sam smith, night, samsmithworld, artist, nominated, tonight, shit, won, enter, album, con-norfranta
February 14, 2015	Valentine's Day	ago, gift, house, valentine, birthday, card, year, cards, red, blue, blackhawks, art, gift, violets, valentine day, roses, special, tomorrow, red, carpet
February 25, 2015	BRIT Awards	brits, awards, brit, awards
March 5, 2015	Harrison Ford crash lands his plane	ford, harrison
March 10, 2015	The family of Marvin Gaye win a record \$7.3 million lawsuit for music copyright infringement (song: "Blurred Lines")	lines, blurred
March 12, 2015	Sir Terence "Terry" Pratchett dies	terry, pratchett
March 16, 2015	Two police officers were shot in Ferguson	ferguson, shot, police
March 24, 2015	Co-pilot commits suicide by crashing Germanwings flight in the French Alps	germanwings, cockpit, pilot, locked, crash, plane french, alps, crash, plane
March 26, 2016	U.S.A. Indiana Religious Freedom Act Protest	indiana, law, religious, freedom

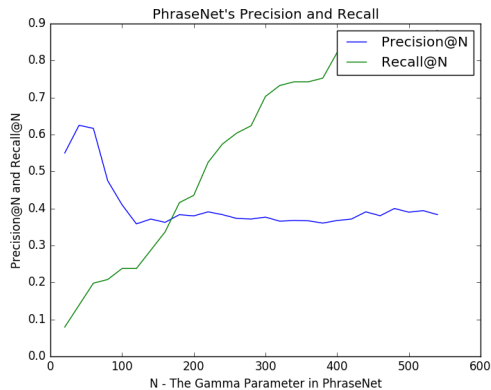
It is seen in the table that the best trade off between precision and recall is when  $\gamma$  is 480 giving an  $F_1$  score of .54.

**Table 2.** This table shows the Precision@N, Recall@N, and  $F_1$  Score of PhraseNet. As you can see, the best precision occurs when  $\gamma$  is set to 40, however, the recall becomes the best when  $\gamma$  is set to 520. To determine the best trade off between precision and recall, is shown by the the  $F_1$  of .54 when  $\gamma$  is 480.

	$\gamma = 20$	$\gamma = 40$	$\gamma = 80$	$\gamma = 260$	$\gamma = 400$	$\gamma = 480$	$\gamma = 500$	$\gamma = 520$
<b>Precision@N</b>	55%	<b>63%</b>	48%	36%	36%	40%	39%	39%
<b>Recall@N</b>	8%	14%	21%	60%	78%	84%	85%	<b>86%</b>
<b><math>F_1</math> Score</b>	.14	.23	.29	.46	.491	<b>.542</b>	.537	.541

For comparison, Twevent was used since it is the most similar state-of-the-art phrase event detection method. Twevent's source code was provided by the authors without the segmentation source code, therefore, the PhraseNet ToPMine output was used to create the necessary segments. In addition, the authors of Twevent specified to set the prior probability of segments to 0.01 based upon





**Fig. 2.** This figure portrays the Precision@N and Recall@N where the N refers to the PhraseNet parameter  $\gamma$ . As you can see from this graph, as more event candidates are considered as events, the recall increases to almost 100%, however, with the increase in recall the precision of PhraseNet begins to slightly decrease.

their previous calculations from Wikipedia and Microsoft N-Gram Web, however, it was found that the prior probability that gave the best  $F_1$  score was .001, therefore, it was used for the comparison.

**Table 3.** Precision and Recall for the best  $F_1$  Score of both PhraseNet and Twevent.

	Precision	Recall	$F_1$ Score
<b>PhraseNet</b>	40%	84%	.54
<b>Twevent</b>	2%	15%	.04

As you can see in Table 3, PhraseNet shows a distinct strength in discovering events compared to Twevent. In total Twevent identified 694 potential events for the three months of data, however, only 22 of those were confirmed true positives. In addition, Twevent identified 11 distinct events out of 102. In comparison when PhraseNet returned 480 potential events, 86 distinct events were correctly identified. These results were determined with the same ground truth list and with the all true positives were identified if found within  $\pm 5$  days of the true event date.

Figure 1 showed an example displaying the key differences between an event distribution and a non-event topic distribution. Twevent identified the non-event topic of social media photos as an event and the Paris attack was not even identified, however, both of these cases were identified correctly by PhraseNet.

One reason for Twevent’s performance is the mistake of identifying a non-event topic as an event. This is due to the mechanism that determines a “bursty” segment. Some words are frequent, however, their popularity in usage tends to

rise and fall in its frequency throughout time. PhraseNet can find these groups of phrase segments and recognizes these multiple rises and falls as a characteristic of a non-event topic.

There was one common weakness made by Twevent and PhraseNet. They both mistakenly identified some non-event topics as events because these particular non-event topics showed event-like characteristics. For example, some artists have an army of users spreading a marketing campaign across social media to pre-order their new album. These types of discussions do not continue after the initial push from the artist’s publicist, therefore, there shows a single high frequency peak on the day of the marketing campaign, yet no other frequency throughout the rest of the data.

### 4.3 Event Summarization: A Case Study

PhraseNet gives a more holistic picture about an event by leveraging phrases and graph clustering than other phrase focused event detection methods. For example, the Super Bowl event detected by PhraseNet consists of the following set of phrases: *superbowl, super bowl, pats, watch, year, vote, superbowlclix, seattle, end, patriotswin, patriots, fans, call, katy perry, music, play, hase, commercial, f\*\*k, s\*\*t, depressing, game, seahawks, win, ago, nfl, chance, team, sb, halftime show, win sb, mousetrapspellingbee, video, youtube, kianlawley*. This description, correlated, aggregated, and produced by PhraseNet, explains that the Seattle Seahawks and the Patriots played in the NFL Super Bowl XLII and, from the “patriotswin” hashtag, the Super Bowl was won by the Patriots. In addition, PhraseNet unveils that the Super Bowl half time show starred Katy Perry.

However, Twevent [9] gives a description of the same event with the following keywords: *rt, superbowl, ve, super bowl, ll, commercial, watch, game, seahawks, time, patriots*. This description of the Super Bowl leaves out the half time show description and who eventually won the game.

### 4.4 Scalability and Efficiency

PhraseNet can be implemented in real-time. PhraseNet has a complexity of  $O(\tau n)$  where  $\tau$  is the number of intervals of the sliding window (i.e. number of documents) and  $n$  is the number of phrases within each sliding window, therefore, it scales to be a suitable algorithm for real-time. Under the experiment setting described in Section 4.1, the running time of PhraseNet is 8.12 seconds per time step where the experiment was run on a Macbook Pro 2.2GHz Intel Core i7 with 16GB of memory. It takes Twevent 45.95 seconds under the same setting.

## 5 Related Work

Twitter opens up doors to a faster way to gain information and to connect. People became a form of social “sensors” [16]. Many event detection algorithms have been proposed, both supervised and unsupervised, based on this platform.

**Supervised Methods.** Supervised methods focus on a certain set of seed keywords or hashtags which causes the method to miss events that have never been seen before or other important, unique, and rare events. This limits the ability of the system to rapidly evolving with its users and the evolving environment the users interact and live [3, 5, 10, 12, 13]. Thelwall et. al. [18] showed evidence that strong negative or positive sentiment about a subject would separate out the events. However, the sentiment was found of a specific set of seeded keywords and hashtags used for tweet correlation which biases the detections to past data and recurring events.

**Unsupervised Methods.** Some event detection papers, such as Twevent, [9], consider trending (aka “bursty”) topics as synonymous to events, however, not all topics are events [11, 8, 20]. Other methods are more semi-supervised methods since they need seeded events to learn from to identify events in the midst of other topics. FRED [14] use training data labeled as “newsworthy” to aid in seeding the model. In addition, GDTM [4] explores a graphical model approach which relies on keywords to seed their unsupervised topic modeling. Ritter et. al. [15] developed a semi-supervised method which makes use of text annotation, however, in the midst of an informal environment such as Twitter, annotations could easily be mistaken. HIML [21] and EMBERS [17] methods required an already established taxonomy to find complex events. The taxonomy focuses on location information given in the text, which is hardly ever the case for Twitter data. TopicSketch [19] identifies “bursty topics” in real time where topics are defined as a word used more frequently at a rate greater than a threshold and does so uniquely. Agarwal et. al. [1] similarly use keywords that occur together in the same tweet appearing in a short sliding window (“burstiness” of a keyword) to identify potential events. In addition, this method uses a greedy clique clustering method to incrementally find small, dense clusters which limits the final description of the event.

## 6 Conclusion

PhraseNet has exhibited to be an unsupervised, real-time Twitter event detection algorithm that summarizes events with a grouping of phrases. PhraseNet showed to have no bias towards certain types of events by being unsupervised, PhraseNet distinguished out non-event topics from events, and gave a short description of the events with a short keyword description. For potential future work, we want to identify dependencies between events and calculate the probability of influence unsupervised.

## Acknowledgement

The work is partially supported by NIH U01HG008488, NIH R01GM115833, NIH U54GM114833, and NSF IIS-1313606. We thank the anonymous reviewers for their careful reading and insightful comments on our manuscript.

## References

1. AGARWAL, M. K., RAMAMRITHAM, K., AND BHIDE, M. Real time discovery of dense clusters in highly dynamic graphs: identifying real world events in highly dynamic environments. *VLDB* 5, 10 (2012), 980–991.
2. BLONDEL, V. D., GUILLAUME, J.-L., LAMBIOTTE, R., AND LEFEBVRE, E. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.
3. BOLLEN, J., MAO, H., AND ZENG, X. Twitter mood predicts the stock market. *Journal of computational science* 2, 1 (2011), 1–8.
4. CHUA, F. C. T., AND ASUR, S. Automatic summarization of events from social media. In *ICWSM* (2013).
5. DU, N., DAI, H., TRIVEDI, R., UPADHYAY, U., GOMEZ-RODRIGUEZ, M., AND SONG, L. Recurrent marked temporal point processes: Embedding event history to vector. In *KDD* (2016), ACM, pp. 1555–1564.
6. EL-KISHKY, A., SONG, Y., WANG, C., VOSS, C. R., AND HAN, J. Scalable topical phrase mining from text corpora. *VLDB* 8, 3 (2014), 305–316.
7. HAN, J., PEI, J., AND YIN, Y. Mining frequent patterns without candidate generation. In *SIGMOD* (2000), vol. 29, ACM, pp. 1–12.
8. KWAK, H., LEE, C., PARK, H., AND MOON, S. What is twitter, a social network or a news media? In *WWW* (2010), ACM, pp. 591–600.
9. LI, C., SUN, A., AND DATTA, A. Twevent: segment-based event detection from tweets. In *CIKM* (2012), ACM, pp. 155–164.
10. LIN, C. X., ZHAO, B., MEI, Q., AND HAN, J. Pet: a statistical model for popular events tracking in social communities. In *KDD* (2010), ACM, pp. 929–938.
11. MATHIOUDAKIS, M., AND KOUDAS, N. Twittermonitor: trend detection over the twitter stream. In *SIGMOD* (2010), ACM, pp. 1155–1158.
12. POPESCU, A.-M., AND PENNACCHIOTTI, M. Detecting controversial events from twitter. In *CIKM* (2010), ACM, pp. 1873–1876.
13. POPESCU, A.-M., PENNACCHIOTTI, M., AND PARANJPE, D. Extracting events and event descriptions from twitter. In *WWW* (2011), ACM, pp. 105–106.
14. QIN, Y., ZHANG, Y., ZHANG, M., AND ZHENG, D. Feature-rich segment-based news event detection on twitter. In *IJCNLP* (2013), pp. 302–310.
15. RITTER, A., ETZIONI, O., CLARK, S., ET AL. Open domain event extraction from twitter. In *KDD* (2012), ACM, pp. 1104–1112.
16. SAKAKI, T., OKAZAKI, M., AND MATSUO, Y. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW* (2010), ACM, pp. 851–860.
17. SARAF, P., AND RAMAKRISHNAN, N. Embers autogs: Automated coding of civil unrest events. In *KDD* (2016), ACM, pp. 599–608.
18. THELWALL, M., BUCKLEY, K., AND PALTOGLOU, G. Sentiment in twitter events. *Journal of the Association for Information Science and Technology* 62, 2 (2011), 406–418.
19. XIE, W., ZHU, F., JIANG, J., LIM, E.-P., AND WANG, K. Topicsketch: Real-time bursty topic detection from twitter. *TKDE* 28, 8 (2016), 2216–2229.
20. YU, W., AGGARWAL, C. C., AND WANG, W. Temporally factorized network modeling for evolutionary network analysis. In *WSDM* (2017), ACM, pp. 455–464.
21. ZHAO, L., YE, J., CHEN, F., LU, C.-T., AND RAMAKRISHNAN, N. Hierarchical incomplete multi-source feature learning for spatiotemporal event forecasting. In *KDD* (2016), ACM, pp. 2085–2094.