Overlapping Target Event and Story Line Detection of Online Newspaper Articles

Yifang Wei, Lisa Singh
Computer Science Department
Georgetown University
Email:{yw255,lisa.singh}@georgetown.edu

Brian Gallagher, David Buttler
Lawrence Livermore National Lab
Email:{gallagher23,buttler1}@llnl.gov

***Abstract*—Event detection from text data is an active area of research. While the emphasis has been on event identification and labeling using a single data source, this work considers event and story line detection when using a large number of data sources. In this setting, it is natural for different events in the same domain, e.g. violence, sports, politics, to occur at the same time and for different story lines about the same event to emerge. To capture events in this setting, we propose an algorithm that detects events and story lines about events for a target domain. Our algorithm leverages a multi-relational sentence level semantic graph and well known graph properties to identify overlapping events and story lines within the events. We evaluate our approach on two large data sets containing millions of news articles from a large number of sources. Our empirical analysis shows that our approach improves the detection precision and recall by 10% to 25%, while providing complete event summaries.**

I. Introduction

Since early 2011, online news readership has surpassed traditional newspaper readership in the US [1]. Given this transition to online news, it is not surprising that the timeliness of online news has continued to improve, also surpassing that of traditional paper sources [2]. While many services exist for finding articles that have certain keywords in them, organizing news into events helps streamline the process of finding information of interest. It can also be useful for identifying unusual events, e.g. civil unrests, or understanding the changing dynamics of topics of interest, e.g. political events/changes in a particular location of the world.

Much literature exists on event detection and story line extraction (document summarization) [3] [4] [5] [6] [7] [8] [9] [10] [11]. Two gaps in the literature that we focus on in this paper are discovering *overlapping* events and determining event story lines when there are a large number of newspaper sources. Our goal is to extract events of a particular theme/target domain (e.g. sports, violence, flu, etc) even if they occur at the same time and effectively summarize story lines associated with each event, thereby providing users with richer context. While many news events discuss a single story, some have multiple story lines (subplots). Our approach attempts to distinguish story lines when an event has more than one - differentiating our work from traditional document summarization methods. For example, suppose we identify the Super Bowl event from a newspaper collection. Different story lines related to the event may include the game summary, the effect of an injury to a key player, the half time show, etc.

For accurate event detection and understanding, it is necessary to track and reason about the connections between related event elements. We leverage a graph data representation for this purpose. Graphs are well-suited for representing complex connections between related entities, and graph analysis algorithms have been developed for reasoning about these connections. More specifically, our approach constructs a graph based on a topic and location of interest using documents in a newspaper collection (node labels are document sentences and edges are based on semantic similarity and sentence proximity between nodes), maps events to partitions of the graph using different heuristics based on well-known graph properties, and summarizes the event using high frequency node labels. This approach to event detection leads to higher precision than the state of the art, detects overlapping events accurately, and provides accurate story line descriptions of detected events.

To summarize, **our contributions to the literature are as follows:** (1) we propose a comprehensive methodology that utilizes a location ontology and a domain dictionary to identify events using relevant news articles from a large, noisy news corpus generated from multiple news sources; (2) we propose a new event detection algorithm that takes advantage of a multi-relational semantic graph to identify and summarize events and propose two additional heuristics that improve the detection quality in different situations; (3) to the best of our knowledge, our method is the first targeted event detection algorithm that detects and summarizes different events occurring at the same time; (4) an empirical evaluation on two data sets demonstrates the accuracy of our event detection when compared to the state of the art; and (5) we compare story lines generated using different event detection methods and show that subject matter experts rate our event story line synopses higher than other methods.

II. Related Literature

This section discusses recent literature in three related areas, non-targeted event detection (methods for detecting all events), targeted event detection (methods for detecting domain specific events), and event/document summarization.

Non-targeted event detection: The majority of literature related to event detection focuses on identifying events that span a broad range of themes or categories [3] [4] [5] [6] [7] [8] [9] [12] [13] [14] [15] [16]. Allan et al. [3], Yang et al. [4], and Brants et al. [5] propose variants that stem from the *TF.IDF* model. Researchers have also proposed models based on term level analysis [6] [7] [8]. Fung et al. [6] propose an algorithm that identifies groups of bursty terms by considering both document frequency of the terms and co-occurrence across documents over time. Variants have been proposed that consider burstiness by comparing to the expected frequency [7], considering spatial proximity of document streams when grouping words [8], and evaluating burstiness using wavelet transforms [9]. Segment level event detection approaches have also been proposed [12] [13] [14]. Leskovec et al. [12] track *memes*, a quoted text segment, in a news document stream, and use a group of *memes* to represent an event. Li et al. [13] divide a tweet into consecutive n-grams that represent semantically meaningful phrases from which bursty segments are selected. Sayyadi et al.'s work [14] is most similar to ours in spirit. They consider approaches for partitioning a graph (their nodes are noun phrases), so that each partition maps to an event. We will show that even though our partitioning strategies are similar, our graph construction approach leads to more accurate event detection and more interpretable story line identification.

Previous literature also considers detecting events at a latent topic level [15] [16] [17]. More recent work considers using network structure to improve event detection [18] [15] [19] [20]. Aggrawal and Subbian [18] construct a graph to represent the interactions between entities in a social stream. Recently, Wang et al. [21] proposed a dynamic hierarchical model to learn multiple aspects (opinions) of news events in Twitter. Finally, Guralnik et al. [22] detect events in numerical time series data, by capturing change points in time series.

Our work differs from all the above mentioned works in the following ways. First, we focus on targeted event detection where the target domain is prespecified. Second, we identify events having the same target theme that may occur at the same time (overlapping events). Finally, previous work generates event summaries using a set of terms, phrases, or text segments. In contrast, our event story line summaries are composed of a small number of sentences, offering readers a more comprehensive understanding of the detected events. We accomplish this trivially since we generate story line summaries using our semantic graph node labels.

Targeted Event Detection: Previous work on targeted event detection includes [23] [24] [25] [10] [11] [26] [27]. One direction of research considers using lexico-syntactic or lexico-semantic patterns to identify events [23] [24] [25]. These methods rely on the assumption that the text segments describing targeted events match one of these patterns; however in real world data, a significant part of text associated with targeted events may not match any of these patterns, resulting in a non-trivial miss rate. Wang et al. [28] propose learning patterns, as opposed to relying on pre-specified patterns, to forecast extreme weather events from spatial-temporal numerical data.

The second thread of research can be categorized as binary predictors [10] [11] [26] [27]. These methods do not detect or summarize a specific event. Instead, they detect the existence of an event within a document stream. They do not distinguish between different events of the same type or events that are overlapping. A typical domain specific approach begins with a keyword vocabulary collected by domain experts, filters the raw corpus with the domain vocabulary, and uses an increase of the number of retained documents in a time window to signify occurring of a targeted type event [10] [11] [26]. Instead of using keywords in the target domain, Muthiah et al. [27] start with a few seed patterns and use a bootstrapping strategy to learn more patterns, which are then used to identify documents (tweets) relevant to target events. Similar to these works, we use a vocabulary to represent a specific theme of interest as a component in our methodology. Our approach differs from these because we discriminate overlapping targeted events in the same and consecutive time windows, we consider simple graph properties of a dynamic semantic graph to identify events, and we trivially generate story line summaries for each detected event.

Event & Document Summarization: Most of recent work on summarizing detected events focus on events using Twitter data [29] [30] [31]. Because we are generating storylines using newspaper articles that are longer and may be discussing multiple events in a single article, we are unable to leverage these Twitter-centric methods. However, document summarization has a long research history. Here we focus on a few representative methods [32] [33] [34]. Barzilay et al. [32] generate a summary for multiple documents by identifying and synthesizing similar elements across related sentences in documents using sentence dependency trees. Shen et al. [33] summarize documents using a Conditional Random Field that labels each sentence within a document with a 1 (summary sentence) or 0 (non-summary sentence). Mihalcea and Tarau [34] build a graph with weighted edges in which nodes represent sentences within a document, and edge weights represent the textual similarities between sentences. Sentences having the highest PageRank scores are used as the summary for the document. While document summarization is relevant to our story line detection, our story lines are for events that cross multiple newspaper articles, as opposed to a summary of a single document.

III. Notation and Definitions

Here, we present definitions, assumptions, and our problem statement. An event is *something that happens at a particular time and location* [3]. We define a *targeted event* to be an event that is associated with a particular domain or topic of interest to the user, e.g. politics, violence, football, etc.

Assumptions and Notation: A newspaper collection $\mathbb{D}$ is a set of articles that occur through time. D^t denotes the set of articles that occur in a time window t. Each newspaper article $d_j \in D^t$ is decomposed into a vector that is a bag of sentences. We maintain a vector, $\mathbb{S}$, of the number of occurrences of sentences $\{s_i\}_{1 \leq i \leq N}$, where s_i is the number of occurrences of sentence i in $\mathbb{D}$, and N is the size of sentence vocabulary.

We assume the following about collection articles:

1) We know the time stamp of the article being published.
2) Each article specifies at least one location.
3) An article may be discussing zero, one, or more events.
4) Articles are composed of paragraphs.
5) A paragraph in an article discusses only one event[1].

When different news agencies describe an event, they may choose to describe different aspects of it. To capture this, we define a *story line* to be a theme or subplot of an event. We also allow an event to take place over one or more consecutive time windows, and assume that an event is reported with temporal continuity. In other words, once it begins being reported, the reporting continues until the event is completed (there are no time window skipped in the reporting).

Problem Statement: Given a newspaper collection $\mathbb{D}$ and a target domain $\mathbb{P}$, the task of overlapping target event and storyline detection has two subtasks: (1) identifying events in the target domain even if they are overlapping; (2) identifying the themes or story lines of the events that have been discovered. As we will show in Section IV, identifying storylines is trivial using our proposed data structures. Therefore, the majority of this paper will focus on subtask 1.

IV. Targeted Event and Story Line Detection

In order to identify and summarize events, we propose a methodology that contains the following steps: location identification, target domain mapping, semantic graph creation, and event detection (see figure 1). Algorithm 1 presents a high level view of our event detection method. The input to the algorithm is our document collection $\mathbb{D}$, a location ontology $\mathbb{O}$ containing major localities around the world (countries, governorates, and cities), a location $\mathbb{L}$ of interest to the user, and a domain $\mathbb{P}$ of interest to the user. Here, $\mathbb{P}$ is a small set of words and phrases that describe a topic the user wants to monitor. The output of the algorithm is a set of events $\{E_k\}$, represented as story line summaries of the documents discussing them. The algorithm begins by going through the document collection and identifying the subset $\mathbb{R}'$ of documents that include the target location (line 1) and the domain of interest (line 3). $\mathbb{R}'$ is then used to create a semantic graph G (line 4). We then look for connected components in G (line 5). These connected components are the basis of the event and story line detection. After identifying the connected components, we consider different heuristics for improving the quality of the detected events (line 7). The remainder of this section goes through the major components shown in figure 1.

[1]We have empirically validated this assumption across 1000s of articles. While articles may discuss multiple events or multiple themes of a single event, paragraphs generally focuses on a single story line in a single event.

A. Location Identification

It is not uncommon for the same event to occur in different locations, but for a user to only be interested in events in a particular location. Therefore, this step identifies the location associated with each document. There are a number of different approaches for location identification. Our approach begins by constructing an ontology $\mathbb{O}$ using open-source data (described in Section V) that contains countries, governorates, and cities. Using this ontology, we then determine the location of each document by counting the occurrences of each location and aggregating the occurrence numbers of the child locations to their parent locations iteratively. The location with the highest frequency count is considered the predominant location of the article. Ties are broken using the location in the title. If the predominant location does not map to the location of interest or the document does not contain a location, it is removed from further analysis. The processing cost of location identification is $O(|\mathbb{D}| \times |\mathbb{O}|)$, where $|\mathbb{D}|$ denotes the number of documents in $\mathbb{D}$, and $|\mathbb{O}|$ denotes the size of the location ontology $\mathbb{O}$.

Algorithm 1: Our event detection approach at high level

Input:
A document collection: $\mathbb{D}$
A location ontology: $\mathbb{O}$
A target location: $\mathbb{L}$
A target domain: $\mathbb{P}$

Output: *A set of events*: $\{E_k\}$

```
1  R = identify_geographically_relevant_documents(L, D, O)
2  T = generate_domain_dictionary(P, θ)
3  R' = identify_domain_relevant_documents(R, T)
4  G(V, E) = create_semantic_graph(R')
5  C = extract_connected_components(G)
6  for C_k ∈ C do
7      C_k = improve_component_quality(C_k)
8      E_k = identify_event(C_k)
9      E_k = generate_storyline(E_k)
10 return {E_k}
```

B. Target Domain Mapping

Since our interest is in identifying events in a particular target domain, we construct a dictionary that contains domain keywords and phrases. Beginning with a set of seed keywords in $\mathbb{P}$, we extract additional related keywords using online thesauri and ontologies. When the size of the dictionary is small or moderate, we have subject matter experts to validate the final dictionary. While a unsupervised approach may be preferred, we have found this semi-supervised approach more promising since it begins with expert knowledge, then expands the domain dictionary using online sources, and finally concludes with expert validation. In cases when the dictionary is very large, subject matter experts validate a sample of the dictionary. The validation is repeated until the accuracy is above a predefined threshold θ (Line 2 of Algorithm 1). Once the dictionary is constructed, we retain articles that contain at least one dictionary keyword in the title. As will

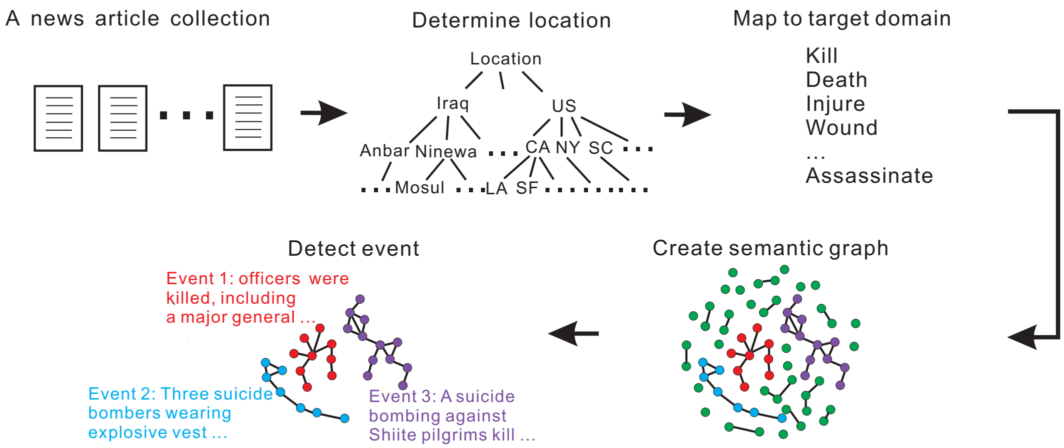


Fig. 1: The framework of our proposed approach

be discussed in Section V, we empirically find that articles themselves are noisier than titles when considering a target domain (Table II). The processing cost of target domain mapping is $O(|\mathbb{R}| \times |T|)$, where $\mathbb{R}$ denotes the documents retained after the location identification, and T denotes the constructed domain dictionary.

C. Semantic Graph Creation

As previously mentioned, graphs are well suited for representing and reasoning about entities and connections between them. While there are many different representations of text, we choose to model it in a *semantic graph*. We propose using this semantic graph G to identify and summarize events. The semantic graph we propose keeps track of sentences in relevant articles and their relationship to each other. More precisely, the semantic graph $G = (V, E)$ is composed of a set of nodes $V(G) = \{v_1, \ldots, v_n\}$ and a set of edges $E(G) = \{e_1, \ldots, e_m\}$. Each node v_i represents sentence i in the sentence vocabulary $\mathbb{S}$ of $\mathbb{R}'$. An edge (v_j, v_k) is added to G if one of the following conditions is true: (1) two sentences are consecutive in the same paragraph (proximity edge) OR (2) two sentences appear in documents that are temporally close (occur on the same day or on consecutive days) and have high semantic similarity (semantic edge).

Proximity similarity is based on the assumption presented in section III that sentences in the same paragraph of an article are discussing the same event. Therefore, an edge is added between nodes in G when the nodes represent two sentences that appear next to each other in a document. *Semantic similarity* is based on the assumption that sentences containing similar vocabulary are semantically similar. Semantic similarity can be measured in many different ways. We consider two different criteria, relative edit distance (RED) and relative common sequence length RCS, where $RED(i,j) = edit(i,j)/n_l$ and $RCS(i,j) = seq\_len(i,j)/n_s$. Here (i,j) denotes a sentence pair, $edit(i,j)$ is the edit distance between i and j, seq_len is the common sequence length between i and j, n_l is the length of the longer sentence ($|i|$ if $|i| > |j|$, otherwise $|j|$), and n_s is the length of the shorter sentence ($|i|$ if $|i| < |j|$, otherwise $|j|$). An edge is added to the semantic graph to connect the sentence pair (i,j) if the semantic similarity is high and they are temporally close. In the next section, we discuss scores that are reasonable for both of these similarity metrics.

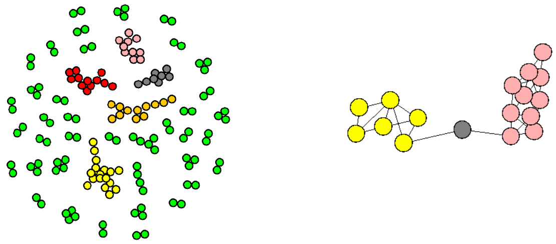
Notice that G is a multi-relational graph since it contains two different types of edges, proximity edges and semantic edges. Considering the semantics of different edges will be useful when we detect events. Finally, we pause to mention that while we could construct the semantic graph using keywords, named entities, and/or noun phrases, we will show the strengths of a sentence level semantic graph in Section V. If we assume sentence length and document length are constants, then the processing cost of semantic graph creation is $O(\mathbb{R}'^2)$, where $\mathbb{R}'$ denotes the documents retained after the target domain mapping. We will show that $|\mathbb{R}'| << |\mathbb{D}|$ in Section V, because only a small portion of documents in $\mathbb{D}$ supports the target location and maps to the target domain.

D. Event Detection

We detect events using the constructed semantic graph G. We begin by identifying the non-trivial connected components. We then consider different heuristics to improve the quality of the non-trivial connected components by pruning and separating weakly connected parts of the subgraph.

1) Connected component event detection: We define a connected component C_k to be a subgraph containing a set of nodes $V(C_k)$ and edges $E(C_k)$ such that every node in $V(C_k)$ has a path to every other node in $V(C_k)$. We define a non-trivial connected component to be a connected component whose total occurrences of its consisting sentences is *reasonably* large, where reasonable will be evaluated empirically in Section V. To provide a little intuition now, we show an example in Figure 2a of connected components identified during our analysis. Trivial connected components are shown in green. The non-trivial connected components are depicted using other colors. We then directly map an event to a non-trivial connected component.

Because some connected components contain weak connections, we also propose two heuristics that attempt to further improve the quality of the non-trivial connected components.



(a) Semantic graph (b) Subcommunity

Fig. 2: Graph examples

We refer to the first one as the *subcommunity heuristics* and the other as the *inheritance pruning heuristics*.

Subcommunity heuristics: We observe that in some cases, a connected component has clear sub-communities (see Figure 2b). This heuristics attempts to identify these subcommunities. While any reasonable community detection algorithm will work, we choose to use the edge betweenness algorithm proposed by Girvan and Newman [35], because its mechanism of detecting communities allows us to favor specific edges, and we will explore other clustering algorithms in the future work. We apply edge betweenness clustering on each non-trivial connected component in G. This algorithm removes edges that have the largest number of shortest paths going through them. Recall that G contains two types of edges, proximity edges and semantic edges. Our goal here is to maximum the chance of semantically similar sentences staying in the same connected component. Therefore, we only consider removal of proximity edges when detecting communities.

Inheritance pruning heuristics: Sometimes two nodes with a semantic edge between them contain sentences in which one sentence is clearly subsumed by the other. A shorter sentence may be connected to a number of longer sentences even though the subsumption relationship only exists between the shorter sentence and *one* of the longer sentences. For each connected component, this heuristics retains the semantic edge that maintains the inheritance relationship and removes the other semantic edges as well as all proximity edges. Intuitively, we maintain connections to 'more detailed' sentences.

E. Story Line Extraction

Each of the heuristics results in a non-trivial set of connected components, each of which maps to an event E_k. While all the sentences (nodes) in the connected component could be used to summarize the event, this leads to redundancy. We reduce redundancy by: (shown in Algorithm 2):

1) Nodes directly connected via semantic edges are reduced to one node - the node with the highest semantic similarity to the other nodes is maintained.
2) The remaining nodes within each connected component are ranked according to the number of occurrences of the sentence in $\mathbb{R}'$.
3) The top-m sentences are selected to be the story line summary of the event.

Algorithm 2: Story line extraction

Input:
A set of non-trivial connected components: C
The sentence count for the synopsis: m

Output: *Event storylines*: E

$C' = extract\_semantic\_cc(C)$
for $C'_k \in C'$ **do**
  for $v_j \in C'_k$ **do**
    $v_j.sim = semantic\_similarity(v_j, C'_k)$
  $E_k = extract\_synopsis(v, m)$
return E

V. Empirical Evaluation

We now evaluate our proposed methodology on two distinct data sets. We begin this section by describing the data sets and specific target domain event detection tasks. We then empirically evaluate different steps of the methodology, comparing our approach to other state of the art methods.

A. Data Sets & Tasks

For our empirical analysis, we consider two data sources (EOS and TREC) and four event detection tasks. The remainder of this subsection describes them.

Population Displacement Using EOS data: The EOS archive contains over 600 million publicly available open-source media articles that have been actively compiled since 2006. New articles are being added at the rate of approximately 100,000 per day from over 20,000 Internet sources in 46 languages. For this analysis, we use a subset of 5 millions English news articles published in 2013 and 2014 that are related to the Middle East, with a focus on Iraq[2].

For the task of identifying events from these 5 million Iraq-related EOS new articles, we work with subject matter experts (SMEs) studying population displacement in the Middle East. Since Iraq has been experiencing renewed security and displacement for the past decade, we are interested in identifying events and story lines related to two different topic areas or domains: violence and governance. We do not have a ground truth event catalog for the 5 million articles. Therefore, we use a restricted subset of the data to create a ground truth data set. We consider the subset of articles published from Dec 9 2013 to Dec 31 2013 reporting on Anbar, a province of Iraq, where Islamic State of Iraq and Syria (ISIS) has been active since 2011. Out of the 5 million articles, approximately 500,000 articles from over 1200 sources fall within the target time period. Subject matter experts create a detailed timeline of events in Anbar for the violence domain (39 events) and the governance domain (30 events) during this 3 week period. For this part of the evaluation, we apply our proposed event detection approach to identify events occurring in Anbar having either the target domain of violence or governance.

[2]These documents are either published by Iraqi news agencies, or contain a term or a phrase related to Iraq, e.g., Iraq, Baghdad, Gulf War.

The identified events are evaluated against the ground truth events manually by SMEs.

Civil Unrest Using TREC data: The TREC-TS-2014F data set is a news article corpus provided by NIST[3]. It includes 20 million news articles published from November 2011 to April 2013. As part of the data set, NIST provides a list of 15 ground truth incidents. Given that articles in this data set are collected during different periods (as opposed to continuously) and the ground truth incidents occur in a number of discontinuous time periods, we select two time periods to conduct our analysis: Dec 4, 2011 to Dec 25, 2011, and Jan 13, 2012 to Jan 25, 2012. During these time periods we have at least 100,000 articles per day. These two time periods contain two ground truth incidents (Russian civil unrest and Romanian civil unrest). The 2011 Russian civil unrest contains 25 events, while the 2012 Romanian civil unrest contains 13 events. In total, we have 4.4 million TREC articles in this evaluation. Our goal is to identify events related to these two incidents from the documents.

B. Location Identification

In this subsection, we explain the construction of our location ontology and test the accuracy of our approach for determining the primary location of a news article. We build our location ontology using *Wikipedia* and *Statoids* [36]. *Wikipedia* has a set of pages listing all the major cities around the world by country, while *Statoids* lists governorates and the governorates' capitals for each country. Leveraging these two sources, we construct an ontology containing approximately 7,600 locations that include countries, governorates, governorates' capitals, and other major cities. Recall that the location with the highest frequency in an article is considered the primary location the article is discussing. Due to space limitations, we do not show a complete evaluation of our ontology accuracy for determining the primary location. In general, our approach led to accuracies of over 80%. We will show in section V-E that this is sufficient accuracy for the event detection task since processing a few additional documents does not impact an event detection approach that considers both semantic content and frequency when determining events.

We pause to mention that location is important for the geographical mapping of the event AND for reducing the search space of events. For example, in our location ontology, the subtree rooted at Anbar, Russia, and Romania have 30, 229, and 78 locations, respectively. This pruning allows for the construction of considerably smaller semantic graphs (with 100s to 1000s of nodes) than if the construction was done using the complete corpus across all locations.

C. Target Domain Mapping

Using the semi-supervised methodology described in section IV, we construct three domain dictionaries with help from our subject matter experts - one for violence, one for governance, and one for civil unrest. Table I shows the number of concepts identified during each step of domain dictionary construction. We see that the thesaurus and ontology augmenting adds a large number of relevant concepts (approximately a factor of 10) and very little noise. On average, 95% of the generated concepts are considered relevant by SMEs.

TABLE I: Domain dictionary creation statistics

	# Seed Concepts	# Concepts Generated During Augmenting	# Concepts Retained after SMEs' Validation
Violence	3	28	28 (100%)
Governance	10	115	111 (97%)
Civil Unrest	3	28	25 (89%)

TABLE II: Article body vs. title domain mapping strategies

		Anbar, Violence	Anbar, Governance
Using article body	Retained	432	354
	Correct	181	162
	SNR	0.72	0.84
Using article title	Retained	166	131
	Correct	166	130
	SNR	Inf	130
	Miss Rate	8.28%	19.75%

For each event detection task, we maintain articles with titles that contain at least one concept from the corresponding domain dictionary (title domain mapping strategy). We also considered a strategy that retains articles if the concept appears in the body of the article (body domain mapping strategy). Table II shows a comparison between the two strategies. The articles identified by each approach are hand evaluated by our project team. We assess the quality of each strategy using a signal to noise ratio (SNR), where SNR is defined as the ratio between the number of documents correctly identified as relevant to the target domain and the number of documents falsely identified as relevant to the target domain. The higher the SNR, the stronger the result. We also consider the miss ratio for our method, where the miss ratio is defined as the number of documents not identified when employing the title domain mapping strategy divided by the number of documents correctly identified by the body domain mapping strategy.

The results show that the title domain mapping strategy has a high SNR compared to the body domain mapping strategy. However, we miss between 8% and 20% of the articles that are relevant. While this number seems high at first glance, we will show, that this miss rate does not result in significant deterioration of the event detection results. However, the additional noise associated with adding documents that are not relevant does lead to a reduction of accuracy for event detection in these data sets. Because of this, we use the title domain mapping strategy as part of our methodology. In future work, we will consider hybrid approaches that may lead to a reduction in the miss rate, while limiting the amount of noise added to the retained documents.

D. Semantic Graph Generation

We generated a semantic graph G for each of our tasks. Table III shows the average number of nodes and edges each day for the different cases. The proximity edges are straightforward to determine using the proposed method in

[3]Available at http://www.trec-ts.org

TABLE III: Semantic graph statistics - averages per day

	#Nodes	#Semantic Edges	#Proximity Edges
Anbar, Violence	67	17	9
Anbar, Governance	103	33	9
Russia, Civil Unrest	1,075	224	185
Romania, Civil Unrest	234	52	46

Section IV. Recall that the semantic edges are determined using two parameters, the relative edit distance (RED) and the relative common sequence length (RCS). Both of these parameters require threshold settings. This remainder of this section considers different setting values and their sensitivity.

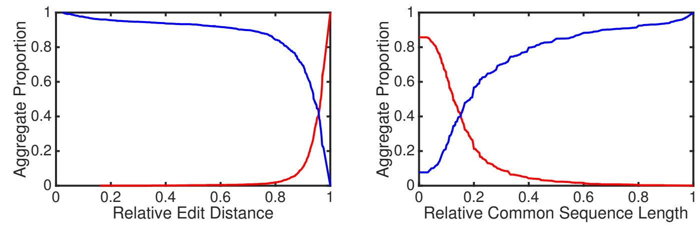
To better understand the effect of these threshold settings, we collect 23,000 random pairs of sentences, each of which consists of two sentences with different meanings, and 2,500 random pairs of sentences, each of which consists of two sentences with the same meaning. The similarity and differences in the sentences were manually determined. For each pair, we calculate the relative edit distance (RED) and the relative common sequence length (RCS). Figure 3 shows a sensitivity analysis for these two parameters. The x-axis represents the relative edit distance (left) and the relative common sequence length (right). The values are between 0 and 1. The y-axis represents the percentage or proportion of sentences that are considered similar or different for each RED or RCS value. As the plot shows, the majority of sentence pairs with the same meaning (blue line) can be identified if the threshold for the relative edit distance is below 0.8. The majority of sentence pairs with different meanings (red line) are not considered the same until the RED is larger than 0.8. For our experiments, we initially choose to be conservative and use a RED threshold of 0.2. We apply the same approach when considering relative common sequence length and find that 0.8 is a good conservative threshold. Both of these plots suggest that the sensitivity of these two thresholds is low in general. Additional extensive empirical analysis shows that the optimal threshold setting for relative edit distance and relative common sequence are 0.1-0.2 and 0.8-0.9 for our detection tasks, respectively.

E. Event Detection

We begin this subsection by comparing the event detection accuracy of our approach, Dynamic Sentence Graph (DSG), to state of the art methods. We then empirically evaluate the proposed heuristics to better understand their impact for event detection in different domains. Finally, we discuss different parameter settings, focusing on their sensitivity for the event detection task.

Event detection experiment details: We compare DSG to four state of the art event detection approaches described in section II, [12] (Meme), [6] (Bursty), [14] (KeyGraph) and [27] (Pattern). Different from others, the Pattern approach is a binary detector, i.e., it only determines whether an event in the target domain exists or not at a specific time and location, without providing a description of the event. This method also requires subject matter experts to manually construct seed patterns to detect events. For the two tasks of detecting civil unrest events in Russia and Romania, we use the same seed patterns as [27], since the types of events being detected are the same. For the two tasks related to detecting violence and governance events in Anbar, we have our subject matter experts create the same number of seed patterns as the civil unrest task. In evaluating the detected events, for the two tasks of detecting violence and governance events in Anbar, we evaluate the top eight events identified by each of the five approaches against the significant ground truth events in terms of precision and recall. For the two tasks of detecting civil unrest events in Romania and Russia, we compare the top 12 and top 20 identified events, respectively, since these two tasks involve significantly more ground truth events (refer to Table IV). We focus on this subset of events because they each have over five documents in the corpus mentioning them. We will refer to these events as 'representative' or 'popular' events. Table IV shows the number of ground truth events, representative/popular ground truth events, and the number of relevant documents for each task. The majority of other ground truth events have at most one or two documents discussing them in the document collections.

Binary event detection accuracy: We begin by evaluating binary event detection, i.e. determining whether events in the target domain exist or not, using the five event detection approaches (DSG, Meme, Bursty, KeyGraph, Pattern). A *Hit* is recorded if events in the target domain are detected and there is at least one popular ground truth event in the target



(a) (b)

Fig. 3: Plots of the aggregate proportion of sentence pairs with different meanings (Figure 3a–red line); Complement of aggregate proportion of sentence pairs with same meaning (Figure 3a–blue line); Aggregate proportion of sentence pairs with same meaning (Figure 3b–blue line); Complement of aggregate proportion of sentence pairs with different meanings (Figure 3b–red line)

TABLE IV: The number of retained documents, ground truth (GT) events, and significant GT events

	#Documents	#GT Events	# Popular GT Events
Anbar, Violence	166	39	6
Anbar, Governance	131	30	4
Russia, Unrest	675	25	15
Romania, Unrest	135	13	7

domain occurring on the same day. In this analysis, we ignore the context of the events. Table V gives the precision and recall of the detected events for each method for each task (Pattern-4 uses 4 seed patterns and Pattern-10 uses 10 seed patterns). Note that for this experiment, we do not use the subcommunity or inheritance pruning heuristics. The table shows that DSG and Pattern perform significantly better than the other methods. The Pattern approach's weakness is the lack of context about the detected event. We know that it is a civil unrest event, but details about it are unknown. The Meme approach has a lower recall because of the method's reliance on quoted text segments. The Bursty approach has the worst precision and recall. The KeyGraph method identifies most of the ground truth events; it has trouble when the ground truth events are the same type, even if the time period does not overlap.

Content-based event detection accuracy: For the next experiment, we evaluate the contents of events detected by the four non-binary event detection approaches (DSG, Meme, Bursty, KeyGraph). A *Hit* is recorded if a detected event maps to a ground truth event when considering both the content and time of the event. The results are shown in Table VI. We see that our approach (DSG) significantly outperforms the other approaches in terms of both precision and recall.

Overlapping event detection identification: Recall that one of our tasks is to detect events even if they are overlapping in time. Table VII shows the number of sufficiently represented ground truth events (over 5 supporting documents) per day, and the number of detected events mapping to each of these ground truth events in a six-day window across the four tasks. For example, the first cell tells us that there are two ground truth events on 2013-12-21, i.e., they are overlapping in time. The cell below (labeled D) indicates that at least one story line for each of the two ground truth events is detected. From the table, we can see that our approach does detect overlapping events well for all the event domains except for the Russian unrest events. We believe this is a result of the skewed frequency distribution of the documents. The data set has a strong frequency skew toward a few significant events. The other ground truth events are overwhelmed by those. So the missed overlapping events have less to do with overlaps and more to do with the highly skewed document distribution.

Event detection additional heuristics: Table VIII shows the precision and recall when incorporating the subcommunity and inheritance pruning heuristics. We see that we get an improvement in some cases, but not others. Because this initial analysis does not give us enough insight about when these heuristics are beneficial, we consider a second approach for evaluating them. We use the notion of *semantic purity*.

An event can have multiple storylines. If we do not want to separate them, then our basic approach without the added heuristics is sufficient. However, if we want to separate them, then we want each connected component to focus in on a smaller number of ideas. To measure the number of ideas in a storyline, we introduce the notion of a semantic group S_g. A

TABLE V: Event detection precision (P) and recall (R) of different algorithms working as binary detectors

		DSG	Meme	Bursty	Key-Graph	Pattern-4	Pattern-10
Anbar Violence	P	**100%**	66.7%	33.3%	75%	**100%**	**100%**
	R	60%	40%	20%	60%	**100%**	**100%**
Anbar Gover.	P	**100%**	66.7%	25%	80%	**100%**	**100%**
	R	**100%**	75%	50%	100%	50%	75%
Russia Unrest	P	**100%**	**100%**	33%	37.5%	**100%**	NA
	R	83.3%	50%	33%	**100%**	83.3%	NA
Romania Unrest	P	**100%**	**100%**	33%	50%	**100%**	NA
	R	**100%**	**100%**	25%	75%	75%	NA

TABLE VI: Event detection precision (P) and recall (R) of different algorithms when taking event content into consideration

		DSG	Meme	Bursty	KeyGraph
Anbar Violence	P	**87.5%**	62.5%	12.5%	75%
	R	**66.7%**	50%	16.7%	50%
Anbar Gover.	P	**100%**	75%	25%	87.5%
	R	**100%**	50%	50%	75%
Russia Unrest	P	**100%**	90%	35%	30%
	R	**40%**	26.7%	20%	26.7%
Romania Unrest	P	**100%**	**100%**	33%	50%
	R	**85.7%**	71.4%	42.9%	28.6%

semantic group is a group of nodes connected by semantic edges in a connected component C_i. We define semantic purity S_p as the number of semantic groups in a connected component: $S_p = |\{S_g | S_g \in C\}|$.

The lower the semantic purity, the less semantic diversity a connected component contains. Therefore, if the goal is to separate storylines, we want a lower semantic purity. Table IX shows the average semantic purity of the connected components when using the basic connected component algorithm, the sub-community heuristics, and the inheritance pruning heuristics. We find that both heuristics improve the purity of the connected component. However, neither is consistently better on different domains. We observe that connected components in the Russia civil unrest graph always have the highest semantic purity. We attribute this to the fact that the articles reporting on the Russian civil unrest are usually much longer than the articles associated with the other three tasks (the average number of words per article is 243, 297, 1214, and 895 for the four tasks, respectively). Longer articles increase the semantic diversity of a connected component, thus increasing the semantic purity. In contrast, the articles reporting on Anbar are usually much shorter, and some are reprints of other reports, thereby reducing the semantic diversity. In general, we recommend the inheritance pruning heuristics if most of supporting documents of an event derive from a few original reports, because inheritance relationships between sentences in such documents are more common than in ordinary ones. In contrast, the subcommunity heuristics may be a good option if most of supporting documents for an event are from a large number of original reports.

Determining non-trivial connected components: We now discuss the parameter setting related to determining non-

TABLE VII: The number of sufficiently represented ground truth events (G) per day, and the number of detected events (D) mapping to each of the ground truth events in a six-day window

		D1	D2	D3	D4	D5	D6
Anbar, Violence	G	2	0	1	1	0	0
13-12-21 to 13-12-26	D	2	0	1	1	0	0
Anbar, Gover.	G	0	0	1	0	1	1
13-12-26 to 13-12-31	D	0	0	1	0	1	1
Russia, Unrest	G	1	2	3	0	4	0
11-12-6 to 11-12-11	D	0	1	0	0	2	0
Romania, Unrest	G	0	1	0	0	0	4
12-1-19 to 12-1-24	D	0	1	0	0	0	3

TABLE VIII: Event detection precision (P) and recall (R) leveraging different heuristics

		Connected Components	Sub-community	Inheritance Pruning
Anbar	P	87.5%	**100%**	**100%**
Violence	R	66.7%	**83.3%**	66.7%
Anbar	P	**100%**	**100%**	**100%**
Governance	R	**100%**	**100%**	75%
Russia	P	**100%**	**100%**	**100%**
Unrest	R	**40%**	33.3%	33.3%
Romania	P	**100%**	**100%**	**100%**
Unrest	R	**85.7%**	71.4%	71.4%

trivial connected components. Using the parameter values specified in the previous section, the constructed semantic graphs for the four tasks are shown in Figure 4. The trivial connected components are green and the non-trivial connected components are other colors. To determine the cutoff between the trivial and non-trivial connected components, we plot the total occurrences of sentences in the detected events (each connected component is a detected event) in Figure 5a and Figure 5b. The x-axis represents the identified events sorted by total frequency of sentences in them and the y-axis represents the total frequency of sentences in the detected events.

Observing these plots, we see that they follow a power-law distribution. Thus, we set the threshold which determines whether a connected component is significant or not to the spot where the long tail starts (marked by the vertical black bar intercepting each line). Using this approach, both the Anbar, violence semantic graph, and the Anbar, Governance graph have 8 significant components, while the Russia civil unrest graph and the Romania civil unrest graph have 12 and 20 significant components, respectively.

Location Identification Sensitivity: To assess the effect of location identification accuracy on our event detection task, we conduct a location sensitivity analysis. As explained in Section V-B, 632 EOS documents are determined by our

TABLE IX: Semantic purity of connected components

	Connected Component	Sub-community	Inheritance Pruning
Anbar, Violence	4.5	2.1	2.3
Anbar, Governance	4.1	1.9	1.8
Russia, Unrest	8.5	3.9	4.2
Romania, Unrest	6.7	3.7	4.3

TABLE X: Event detection accuracy with varying location identification accuracy

# Noise Documents	0	30	40	80	120
Location Accuracy	100%	80%	75%	60%	50%
Event Detection Precision	87.5%	87.5%	75%	50%	50%

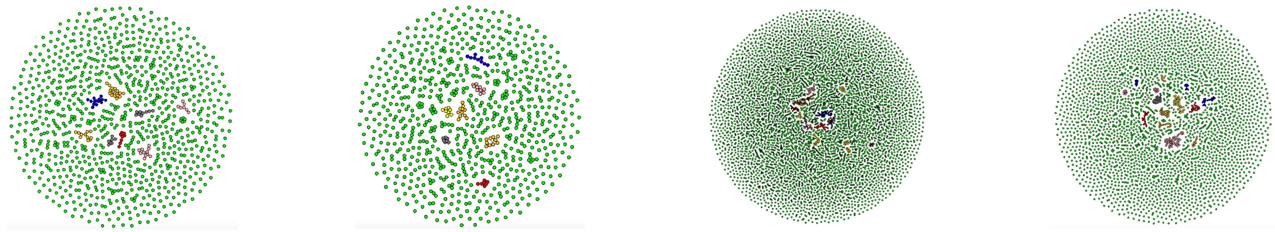
location identification approach to be discussing Anbar. The violence event in Anbar best represented in EOS is supported by 120 EOS documents. We then add a different number of EOS articles known to be discussing other locations to the 120 documents. These added articles are considered *noise*. By adding different levels of noise to the target documents, we can better understand the impact of noise on the final event detection results. In the first experiment, we add 30 noise documents (reducing the location accuracy to 80%). In the second experiment, we add 40 noise documents (reducing the location accuracy to 75%). In the third and fourth experiments, we add 80 and 120 noise documents, respectively. We apply our event detection approach to the constructed document sets to detect violence events occurring in Anbar, and then evaluate the detected events against the ground truth events. The results are shown in Table X. We see that an 80% accuracy in location identification is sufficient for target domain event detection. This makes sense since documents from other locations are not likely to have the same themes are those in the target location. However, when the amount of noise gets large, it impacts the quality of the detected events.

F. Event Story Lines

We evaluate the story line summaries generated by our approach against those generated by the other content-based event detection approaches. We also compare all the summaries to a "gold standard" summary obtained from a well known document summarization approach (PageRank) introduced by Mihalcea and Tarau [34]. For the gold standard, we use the most relevant document to summarize the event. Two SMEs rated all the summaries using a scale from 1–5, where 1 is the lowest and 5 is the highest rating based on informativeness, readability and accuracy. Since the state of the art approaches only detected 6 out of the 10 representative ground truth events associated with Anbar, this experiment focuses on those 6 events. The average ratings of the four approaches and the gold standard are shown in Table XI. We see that the SMEs almost always prefer the gold standard. However, our approach results in the highest average rating of the event detection methods. We attribute this to the sentence level nodes in the semantic graph. Table XII shows the first four lines of the event storyline summaries given by DSG, Meme, Bursty, and KeyGraph approach for one of the six events. In general, the methods using keywords resulted in less detailed summaries, while the methods using phrases were more informative, readable, and accurate to the SMEs.

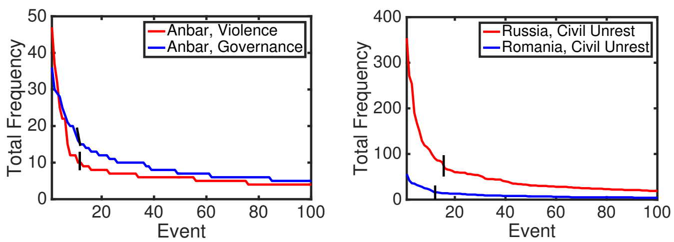
G. Depicting Event Dynamics

Our event detection approach is also helpful for identifying the evolving dynamics of events. We can accomplish this by



(a) Anbar, Violence, Dec 9 2013 - Dec 31 2013 (b) Anbar, Governance, Dec 9 2013 - Dec 31 2013 (c) Russia, Civil Unrest, Dec 4 2011 - Dec 25 2011 (d) Romania, Civil Unrest, Jan 13 2012 - Jan 25 2012

Fig. 4: Semantic graphs for different tasks. Green – trivial; other colors – non-trivial



(a) Anbar (b) Russia, Romania

Fig. 5: The total occurrences of sentences in connected components

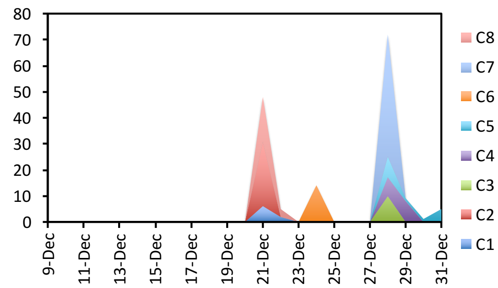
TABLE XI: The average rating of different event detection approaches compared to a gold standard for the quality of event summary (an NA means that an approach failed to detect the ground truth event.)

	DSG	Meme	Bursty	Key Graph	Gold Standard
GT Event 1	3.5	3.5	2	3.5	5
GT Event 2	4.5	1.5	1	2.5	5
GT Event 3	3	NA	NA	1	5
GT Event 4	4.5	2	1	3	4
GT Event 5	4	NA	NA	1	4
GT Event 6	2	NA	2	NA	3.5

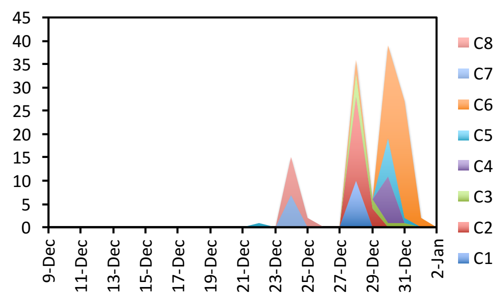
looking at the identified events and their sentence overlap through time. Figure 6 shows the total number of occurrences of sentences in detected events over multiple days. The x-axis is the date and the y-axis is the frequency of the sentences (nodes in G) associated with an event. Fluctuations in the frequency of these sentences highlights the rise and fall of an event's media popularity. While event dynamics is not the focus of this paper, this figure highlights an additional value of our event detection approach.

TABLE XII: The story line summaries given by DSG, Meme, Bursty, and KeyGraph approach for a representative ground truth event and their averaged rating (AR) by SMEs

Approach	AR	Story Line Summary
DSG	4.5	Iraqi police officials say Alwani's brother and three guards were killed after they opened fire on security forces at dawn on December 28 as they arrived to arrest him. Alwani a Sunni lawmaker who had ...
Meme	1.5	Army troops with police special forces were trying to arrest Alwani. We told him that we had a warrant for his arrest and arrested him. I call upon Sunni's protesters and sons of Ramadi to insist upon your ...
Bursty	1	amid, government, group, Maliki, prime minister city, Anbar, December
KeyGraph	2.5	Alwani's release, Anbar's provincial council, the death of his brother, a strong critic of Maliki, minority Sunni leaders, attacks that killed Iraqi soldiers, a clear violation, the core of the Iraqi constitution, its articles ...



(a) Anbar, Violence



(b) Anbar, Governance

Fig. 6: Frequency dynamics of sentences in detected events

VI. Conclusion

In this paper, we propose a comprehensive methodology that utilizes a location ontology and a domain dictionary to identify overlapping, target events using news articles from a large, noisy news corpus generated from multiple new sources. To the best of our knowledge, our method is the first targeted event detection algorithm that detects events and story lines occurring at the same time. We make use of a semantic graph constructed from sentences within articles from the corpus. We use a set of graph invariants (connected components, community structure, and node subsumption) on this semantic graph to help us identify popular events. Using this multi-relational graph allows us to capture different types of relationships between sentences in the document. We believe this type of graph is the reason we perform better than the state of the art. Extensive experiments on two large data sets demonstrate the strengths of our event detection method when compared to the state of the art. We also conduct detailed sensitivity analyses on different parameters to give researchers intuition about their settings. Finally, we show that our event synopses are effective in helping readers gain a better understanding of the detected events when compared to synopses generated by other methods. We believe that this area of research is fruitful and necessary for not only identifying and understanding events in large, noisy corpora, but also understanding the types of information people find important to discuss.

ACKNOWLEDGMENTS

We thank Abbie Taylor, Nili Yossinger, Lara Kinne, and Eleanor Swingewood for labeling data and their general subject matter expertise provided throughout the process. This work was supported in part by the National Science Foundation (NSF) Grant SMA-1338507, the Georgetown University Mass Data Institute (MDI), and the Lawrence Livermore National Laboratory (LLNL) under Contract DE-AC52-07NA27344. Any opinions, findings, conclusions, and recommendations expressed in this work are those of the authors and do not necessarily reflect the views of NSF, MDI, or LLNL.

REFERENCES

[1] "Nbcnews," http://www.nbcnews.com/technology/online-news-readership-overtakes-newspapers-124383.

[2] "Inma," http://www.inma.org/article/index.cfm/23899-credibility-of-online-newspapers.

[3] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in *SIGIR*. ACM, 1998, pp. 37–45.

[4] Y. Yang, T. Pierce, and J. Carbonell, "A study of retrospective and on-line event detection," in *SIGIR*. ACM, 1998, pp. 28–36.

[5] T. Brants, F. Chen, and A. Farahat, "A system for new event detection," in *SIGIR*. ACM, 2003, pp. 330–337.

[6] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, "Parameter free bursty events detection in text streams," in *VLDB*. VLDB Endowment, 2005, pp. 181–192.

[7] T. Lappas, B. Arai, M. Platakis, D. Kotsakos, and D. Gunopulos, "On burstiness-aware search for document sequences," in *KDD*. ACM, 2009, pp. 477–486.

[8] T. Lappas, M. R. Vieira, D. Gunopulos, and V. J. Tsotras, "On the spatiotemporal burstiness of terms," in *VLDB*, 2012, pp. 836–847.

[9] J. Weng and B.-S. Lee, "Event detection in twitter." *ICWSM*, vol. 11, pp. 401–408, 2011.

[10] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *WWW*. ACM, 2010, pp. 851–860.

[11] N. Ramakrishnan, P. Butler, S. Muthiah, N. Self, R. Khandpur, P. Saraf, W. Wang, J. Cadena, A. Vullikanti, G. Korkmaz *et al.*, "'beating the news' with embers: Forecasting civil unrest using open source indicators," in *KDD*. ACM, 2014, pp. 1799–1808.

[12] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *KDD*. ACM, 2009, pp. 497–506.

[13] C. Li, A. Sun, and A. Datta, "Twevent: segment-based event detection from tweets," in *CIKM*. ACM, 2012, pp. 155–164.

[14] H. Sayyadi, M. Hurst, and A. Maykov, "Event detection and tracking in social streams." in *ICWSM*, 2009.

[15] C. X. Lin, B. Zhao, Q. Mei, and J. Han, "Pet: a statistical model for popular events tracking in social communities," in *KDD*. ACM, 2010, pp. 929–938.

[16] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, and K. Wang, "Topicsketch: Real-time bursty topic detection from twitter," in *ICDM*. IEEE, 2013, pp. 837–846.

[17] X. Wang, C. Zhai, X. Hu, and R. Sproat, "Mining correlated bursty topic patterns from coordinated text streams," in *KDD*. ACM, 2007, pp. 784–793.

[18] C. C. Aggarwal and K. Subbian, "Event detection in social streams." in *SDM*, vol. 12. SIAM, 2012, pp. 624–635.

[19] J. Ferlez, C. Faloutsos, J. Leskovec, D. Mladenic, and M. Grobelnik, "Monitoring network evolution using mdl," in *ICDE*. IEEE, 2008, pp. 1328–1330.

[20] F. Chen and D. B. Neill, "Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs," in *KDD*. ACM, 2014, pp. 1166–1175.

[21] J. Wang, W. Tong, H. Yu, M. Li, X. Ma, H. Cai, T. Hanratty, and J. Han, "Mining multi-aspect reflection of news events in twitter: Discovery, linking and presentation," in *ICDM*. IEEE, 2015, pp. 429–438.

[22] V. Guralnik and J. Srivastava, "Event detection from time series data," in *KDD*. ACM, 1999, pp. 33–42.

[23] F. Xu, H. Uszkoreit, and H. Li, "Automatic event and relation detection with seeds of varying complexity," in *AAAI workshop event extraction and synthesis*, 2006, pp. 12–17.

[24] Y. Nishihara, K. Sato, and W. Sunayama, "Event extraction and visualization for obtaining personal experiences from blogs," in *Human Interface and the Management of Information. Information and Interaction*. Springer, 2009, pp. 315–324.

[25] A. Ritter, O. Etzioni, S. Clark *et al.*, "Open domain event extraction from twitter," in *KDD*. ACM, 2012, pp. 1104–1112.

[26] K. Lei, R. Khadiwala, and K.-C. T. Chang, "A twitter-based event detection and analysis system," in *ICDE*, 2012.

[27] S. Muthiah, B. Huang, J. Arredondo, D. Mares, L. Getoor, G. Katz, and N. Ramakrishnan, "Planned protest modeling in news and social media." in *AAAI*, 2015, pp. 3920–3927.

[28] D. Wang and W. Ding, "A hierarchical pattern learning framework for forecasting extreme weather events," in *ICDM*. IEEE, 2015, pp. 1021–1026.

[29] D. Chakrabarti and K. Punera, "Event summarization using tweets." *ICWSM*, vol. 11, pp. 66–73, 2011.

[30] J. Nichols, J. Mahmud, and C. Drews, "Summarizing sporting events using twitter," in *IUI*. ACM, 2012, pp. 189–198.

[31] F. C. T. Chua and S. Asur, "Automatic summarization of events from social media." in *ICWSM*. Citeseer, 2013.

[32] R. Barzilay, K. R. McKeown, and M. Elhadad, "Information fusion in the context of multi-document summarization," in *ACL*. Association for Computational Linguistics, 1999, pp. 550–557.

[33] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields." in *IJCAI*, vol. 7, 2007, pp. 2862–2867.

[34] R. Mihalcea and P. Tarau, "A language independent algorithm for single and multiple document summarization," in *ICJNLP*, 2005.

[35] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[36] "Statoids," http://www.statoids.com.