Towards Accurate Event Detection in Social Media: A Weakly Supervised Approach for Learning Implicit Event Indicators

Ajit Jain, Girish Kasiviswanathan, Ruihong Huang
Department of Computer Science and Engineering
Texas A&M University
College Station, TX 77843
{ajitjain,girishk14,huangrh}@cse.tamu.edu

Abstract

Accurate event detection in social media is very challenging because user generated contents are extremely noisy and sparse in content. Event indicators are generally words or phrases that act as a trigger that help us understand the semantics of the context they occur in. We present a weakly supervised approach that relies on using a single strong event indicator phrase as a seed to acquire a variety of additional event cues. We propose to leverage various types of implicit event indicators, such as props, actors and precursor events, to achieve precise event detection. We experimented with civil unrest events and show that the automatically learnt event indicators are effective in identifying specific types of events.

1 Introduction

Social media data has today evolved into a crowd-sourced knowledge base of real-time happenings, sentiments, and future events. For instance, in a recent natural disaster, rescue volunteers used social media to coordinate their actions and identify survivors (Starbird and Palen, 2011). Accurately identifying a particular class of events in social media will benefit many downstream applications such as event tracking, event time-line generation, and event summarization. However, some critical information is buried within piles of mundane tweets and is hard to detect precisely. In Table 1, we provide some examples of potentially critical content embedded within tweets that our system was able to extract.

Two intuitive approaches have been widely applied to event detection in tweets, first the event keyword matching approach that is far from being perfect with our experiments yielding accuracies as low as 14%, even while using keywords very relevant to a domain. Second, unique to tweets, hashtags, for instance,

Tweet	Critical Content
Public transport suspended in Istanbul as part of authorities precaution to curb #MayDay gatherings in Taksim Square	Real-time warning
Extreme violence of police in Besiktas, Istanbul now! Let the world know	Pleas for help
1km chain of citizens to build a barricade in besiktas, akaretler.	Potential risk in future
An innocent young girl was put into a coma due to tear gas attacks and was later labeled by the government as a terrorist	Consequence of event
This is the craziest I've seen Cairo in the two years I've been here. So many clashes, tear gas everywhere. Insane	Sentiment about event
Dear Policemen, Thank you so much for using bean bag rounds and rubber bullets in Brisbane. I really appreciate your protection of life	Sarcasm

Table 1: A few examples of highly critical tweets that we have detected using the bootstrapping approach and a single high quality event indicator seed "tear gas".

Proceedings of the 2nd Workshop on Noisy User-generated Text,
pages 70–77, Osaka, Japan, December 11 2016.

Trigger Type	Examples
Instruments/Props	barricades, rubber bullets
Sub-events	police pushing, crowd shouting
Precursor events	government bans, arrest of prisoner
Consequences	injury, damaged
Locations	Taksim Square, Ankara
Actors	angry employees

Table 2: Civil unrest event indicators types with examples of extracted phrases. Clustering algorithm is useful towards understanding the semantic types of the learned event indicators.

"#RiseUpOctober" can be used to accurately identify tweets describing the October New York protest in 2015. However, this approach is heavily biased towards already known trending events that are more likely to be "hashtagged", and is often not reliable even to capture tweets referring to a single event.

Tweets belonging to a particular event domain (eg. civil unrest, disaster, presidential election) can be identified by learning various kinds of event indicators or sensors across contexts. In spite of their highly informal and ambiguous nature (Ritter et al., 2011), tweets often mention multiple event characteristic properties and features that act as implicit event cues. An event indicator is defined as any word or phrase that can act as reliable evidence towards detecting an event mention in text. A strong indicator is one that is almost exclusively relevant to the event under consideration (eg. *touchdown* is strong evidence of a sports event). A weak indicator tends to occur in more generic textual contexts and is therefore less useful.

We also observe that these event indicators can be categorized into sub-classes, depending on how they influence the event. Table 2 presents a summary of event indicator types that were prevalent for the civil unrest domain, and corresponding examples of the extracted indicator phrases.

For example, the following tweet,

> *"Police attacking us with plastic bullets and tears gaz in istanbul! 5people died, hundreds injured! Help"*

contains multiple indicators including a sub-event (*police attacking*), instruments (*plastic bullets* and *tears gaz*) and consequential events (*people died* and *hundreds injured*). Some event indicators are so strong that tweets containing one such phrase are almost certainly relevant, for instance *"tear gas"* is a unique type of instrument used in civil unrest events.

Following these observations, we propose a weakly supervised approach to automatically learn such event indicators. The system starts with a single high quality event indicator seed, and extracts an initial set of highly relevant tweets, which are likely to contain other indicators. These potential indicators are then combined with the seed set to grow a collection of indicators. This way, the tweet extraction and collection expansion phases augment each other mutually, the whole process iterates while using empirical thresholds to monitor quality of extraction (Riloff et al., 1999). We then proceed to cluster these indicators based on their contextual similarities, which yield some naturally occurring categories of event triggers as seen in Table 2. Our approach is able to outperform naive keyword matching even without having any prior knowledge or hand labeled data, other than a single seed word. Furthermore, our indicators were able to identify tweets that did not contain a single event keyword.

2 Prior Work

Our research closely follows the multi-faceted event recognition approach (Huang and Riloff, 2013) for news articles which suggests that event facets, namely agents and purposes, supplement event expressions. However, event indicators in tweets are loosely defined and lack phrasal dependencies, hence posing a more challenging problem.

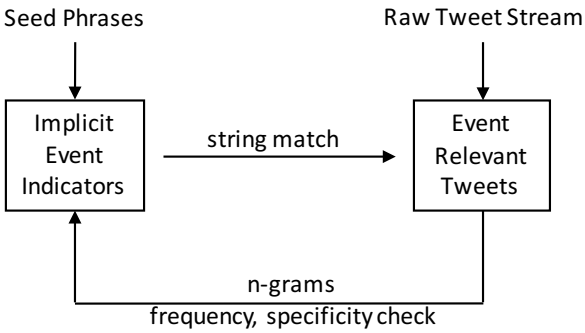

Figure 1: The bootstrapping process iteratively adds event relevant tweets and implicit event indicators.

First, in contrast to event facets, event indicators are loosely defined event properties and features which cover a broader range of implicit event cues. Second, in their proposed bootstrapping system, strict phrasal structures for event facet phrases and dependency relations between event facet phrases and the main event expression are required. Instead, our system is well tailored to event indicator learning in noisy tweets, where learning targets are simply n-grams[1] and we only consider n-grams that strongly correlate with domain-specific tweets as generated through the bootstrapping process.

(Atefeh and Khreich, 2015) summarize the key literature in the spectrum of event detection in Twitter, and establish a taxonomy of these techniques based on type of event, detection task, and detection method. Our weakly supervised approach is capable of detecting both new and retrospective unspecified event types.

(Becker et al., 2011), together with (Sankaranarayanan et al., 2009) and (Petrović et al., 2010), discuss an approach that collects tweets in temporal bins, forms event clusters, and trains a classifier to identify real world events. This approach however is heavily dependent on the incremental clustering criteria, training set of events, and sparse TF-IDF vectors. In this work, we stress on the broader task of extracting tweets that are relevant to a given class of event.

The event extraction task itself most closely resembles that of (Ritter et al., 2012), which presents a hidden variable model for event type discovery on a collection of tweets. Tweets are segregated into discrete event types by first training a sequence model that can identify event triggers, and then using a latent variable model to discover the types implicit within the tweets, and segregating them accordingly. Instead, we focus on improving accuracy of detecting individual tweets that describe a particular type of event by assuming prior knowledge about the event type in the form of a single seed.

This, to the best of our knowledge, is the first attempt to extract events from social media data using a weakly supervised approach.

3 System Overview

Our system broadly comprises of (1) a bootstrapping module for unsupervised learning of event indicators and (2) a clustering module for understanding context of occurrence of these event indicators.

3.1 Bootstrapping

Figure 1 shows the bootstrapping process. We start by selection of initial seed phrases, based on inspection of civil unrest event indicators. Generally, any word that is deemed by a domain expert as being highly relevant to an event, can be used as the seed. We add the seed phrases to the set of implicit event indicators, which are then searched over the raw tweet stream to identify event relevant tweets. From these event relevant tweets, we select n-grams as implicit event indicators for the next iteration, based on frequency and specificity checks.

For our experiment, we used "*tear gas*" as the initial seed word. We used about 200 million tweets that were collected over a period of 6 months in 2013 for bootstrapping implicit event indicators and event

[1]We consider 1, 2, 3, 4-grams as candidate event indicators.

relevant tweets. We stop after a couple of iterations as our study is currently in initial phase. We discuss generation of event relevant tweets and implicit event indicators, including filtering steps, redundancy handling, and selection criteria. We also discuss the phrase specificity parameters used for monitoring the quality of extraction.

3.1.1 Event Relevant Tweets

The bootstrapping algorithm first finds all the tweets containing the seed phrases. Redundant tweets drastically impact frequencies and thus selection of event indicators, therefore the algorithm performs a redundancy check for removal of any duplicate tweet texts. This includes removing tweets that are exact matches, substrings, and differing only in use of RTs, preceding and succeeding punctuations, @ and # associations, and web urls. A recursive check is performed as such differences can occur in any order. We then use the non-redundant tweets for determining implicit event indicators for the next iteration. We additionally base selection of tweets on *specificity factor*, which we describe next to generation of implicit event indicators.

3.1.2 Implicit Event Indicators

Our algorithm generates candidate implicit event indicators from newly learned event relevant tweets. For this purpose, we extract n-grams (n=1,2,3,4) from tweets. We use pre-filtering where we discard any n-grams that have frequency less than 5 in the event relevant tweets. This helps us in removal of n-grams less likely to be event indicators. However, this way, we also get a large number of n-grams that are general, i.e., which can occur in any set of random tweets. To handle this, we use the specificity factor, which we describe next.

3.1.3 Specificity Factor

We select both the event relevant tweets and implicit event indicators based on specificity checks. We define specificity factor (sf) as the normalized ratio of frequency of an event indicator occurs in the event relevant tweets and frequency in a random tweet set. To calculate the specificity factor, we use the formula:

$$sf = \frac{(h_c/c_c)}{(h_r/c_r)}$$

where h_c = event indicator hit count in event relevant tweets, c_c = count of tweets in tweet collection, h_r = event indicator hit count in event relevant tweets, c_r = count of tweets in random tweet set

We formed the random tweet set by sampling about 1% i.e. 2 million tweets from the original dataset used for bootstrapping. A general phrase occurs frequently in both the random set and tweet collection. Hence, its specificity factor is low. On the other hand, an event indicator occurs more frequently in the tweet collection as compared to the random set. Only phrases above the set specificity factor thresholds (1000 for unigrams, 5000 for 2,3,4-grams) are retained as event indicators for the next iteration, i.e., for finding event relevant tweets. We also distinguish strong event indicators from relatively weak ones.

Strength	Examples of Event Indicator
Strong Indicators	gas shells, firing tear, clouds fill, disperse protesters, gas canisters, during clashes, water canons, throwing tear, gas attack, gas shot, shooting tear, stun grenades, police disperse, police firing, tear gas, gas clouds, fired by police
Weak Indicators	forces fire, pressurized water, police use, against peaceful, riot police, against protesters, gas bombs, rubber bullets, pressurized, barricades, grenades, anti-government, mercenaries, protesters, demonstrators, clashes, #occupygezi, quell, #bahrain

Table 3: Top event indicators resulting from the bootstrapping approach in descending order of phrase specificities. Different semantic types could be learned starting with a single high quality seed phrase.

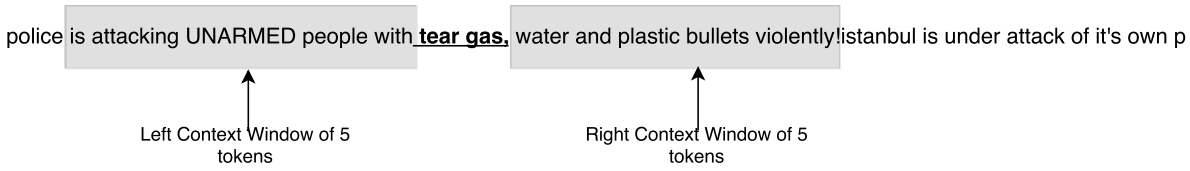

Figure 2: Extraction of context windows from one the tweets containing the learnt phrase 'tear gas'. The tweets are padded on both sides with UNK tokens and a frequency threshold is used to discard unimportant tokens

Strong event indicators have specificity greater than 20000. A tweet is determined to be a relevant tweet if it contains one strong event indicator or two weak event indicators. These thresholds were identified by inspecting specificity factor scores of the n-grams generated from initial couple of iterations. Note that we expect the concrete parameters to change when applying the same system but to learn event indicators for a different event domain, however, tuning the parameters in our experiments based on the generated candidate event indicators from the first one or two iterations is quite manageable. Table 3 shows the top event indicators organized in these two categories.

3.2 Clustering

Upon building a collection of event indicators, we attempt to identify contextual similarities, *i.e* how each phrase acts as a cue for an event, as discussed in Table 2. For each phrase, we accumulate its contexts using a window of 5 words on either side of each tweet in which the phrase occurs, as shown in Figure 2. Next, we build a feature vector to represent the context words of each indicator using two approaches - (1) frequency bag-of-words (2) sum of word embeddings ("tweet-token embedding" to be precise) while clipping stop words and infrequent words. We use the 200-dimensional global vectors (Pennington et al., 2014), pre-trained on 2 billion tweets, covering over 27-billion tokens. We then perform clustering using the affinity-propagation algorithm (Frey and Dueck, 2007).

4 Evaluation

We evaluate the quality of tweets identified by using the automatically learned event indicators over the successive iterations of the bootstrapping phase, as described in Section 3. Table 4 shows the number of event indicator phrases and civil unrest tweets that were collected after each iteration of bootstrapping. Table 5 shows examples of correct and incorrect tweets.

To measure the relevance of the tweets as identified by the event indicators, we sampled 500 tweets from the final set collected after 4 iterations. Further, to verify the capability of our system to learn less prominent indicators such as "cops rounded", "spark clashes", "made demands", we collect an additional 500 tweet sample after filtering out tweets that contain a common event keyword like "protest", "riot", etc.

We also separately collected another random sample of 1000 tweets from the original corpus of 220

Iteration	Event Cues	Relevant Tweets
1	1	1445
2	309	4503
3	719	7555
4	1037	8521

Table 4: Event indicators and relevant tweets accumulate with each iteration. Semantic drift occurs in this process.

Correct:
Protesters chant 'Stop the IRS' in Cincinnati: CINCINNATI (AP) Tea party activists waving flags and signs, singing pat
Protestors have attacked Al Jazeera TV headquarters in Egypt and have set fire to the building
ISTANBUL - Riot police used tear gas and pressurized water in a dawn raid on Friday to rout a peaceful demonstration by hundreds of peopl
NYC protest at Zuccotti Park in solidarity with #OccupyGezi starting now
CNN: 25 people killed, 70 others wounded as clashes erupted between Iraqi Security Forces and protesters in Hawija, Iraq
Incorrect:
Saudi Arabia postpones crucifixion, firing squad executions: Seven juveniles were arrested for armed robbery
ICYMI: Watch video of my comments on @CTVnews on #Syria and #sarin gas. Iraq WMD deja vu? Or #Assad war crime

Table 5: Examples of correct and incorrect tweets learned through the bootstrapping process.

million tweets, so as to train a supervised baseline classifier.

Each sample was then annotated by two-annotators to determine if each tweet was actually relevant to civil unrest, with an inter-annotator agreement, $\kappa = 0.81$ (Cohen, 1968), suggesting that our task guidelines are well-defined and unbiased.

We next show the event detection performance of the supervised system and the performance of our event indicator matching approach. In addition, we explain the semantic drift in the bootstrapping process, and demonstrate the results of event indicator clustering.

4.1 Supervised Baseline

To train the baseline model, we used a simple bi-gram model to extract features from each tweet, and trained a Support Vector Machine classifier using a a linear kernel, on the sample of annotated tweets held out exclusively for training purpose.The sample contained 350 relevant tweets.

We then tested this model on the sample set extracted after 4 iterations of our algorithm. The supervised model was able to achieve a precision of 95%, but only yielded a recall of 4%, leading to a poor F-Score of 8%.

The main point to note here is that supervised algorithms are restricted to learning only features that they have been exposed to, and are vulnerable and limited when abundant annotated data is not available.

4.2 Precision and Recall

Table 6 shows accuracies of the event keyword matched tweets and the two tweet samples identified by using learned event indicators. We can see that event keyword alone is very unreliable in detecting tweets that describe a particular type of event and the accuracy is as low as 12%. Using event indicators, we can greatly improve the accuracy of event detection to 66%. The automatically learned event indicators

Methods	Accuracy
Event keyword Matching	12%
Bootstrapped (contain a keyword)	66%
Bootstrapped (contain no keyword)	35%

Table 6: Accuracy of bootstrapping on collected tweets. Our system outperforms the baseline event keyword matching. The automatically learned event indicators is even able to identify the relevant tweets that do not contain an event keyword.

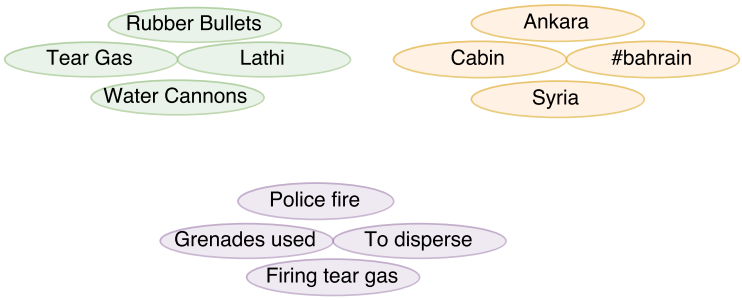

Figure 3: A few examples of the different categories of event indicators our clustering algorithm was able to identify, namely props, locations and actions

can even discover relevant tweets that do not contain a common event keyword, achieving an accuracy of 35%, which is still much better than using the method of keyword matching.

However, we do not analyze recall at this point since it is hard to define a global notion of what subset of the tweets of the entire stream are true positives that our system is expected to detect. We will look into more doable laboratory settings in the future.

4.3 Semantic Drift

While the automatically learned event indicators have significantly improved the accuracy of event detection, the performance is not perfect and many irrelevant tweets were wrongly labeled. Table 5 shows examples of tweets that are truly event relevant tweets and the ones that are errors. Our analysis of the errors shows that the current bootstrapping algorithm suffers from a major drawback, *semantic drift* (Davenport and Cronin, 2000). After each iteration, a conceptual drift in the target collection is triggered, with a few poor indicator phrases propagating error and causing a gradual shift in domain. For instance, there was some noise induced from tweets on police arrests and war crimes. In the future, we are going to investigate techniques for reducing the semantic drift.

4.4 Clustering

While the sparsity and inconsistency of the twitter vocabulary tend to ambiguate the notion of contextual similarity, the final clusters obtained from either feature set (BoW or GloVe), reflected a few interesting patterns, similar to Table 1. For instance, *'tear gas', 'lathi', 'water cannons' and 'pepper spray'* were clustered together, and similarly were *'dispersed', 'violence on', 'sprayed on'*. Hence, clustering can prove useful towards understanding the latent semantic categories of the learnt event indicators. However, we reserve strict evaluation of these clusters for future work, as this task is heavily contingent on our primary task of learning high quality event indicators. Figure 3 shows a few examples of the preliminary clusters we were able to isolate in this setting.

Another interesting observation is that most of the domain-drift inducing indicators discussed in Section 5.2 were grouped together, suggesting that we can filter them out in the future.

5 Future Work

Extracting meaning from tweets is task far more challenging beyond the perspective we have adopted, and requires many more standpoints. The next level of our work is to investigate more event domains as well as to suggest means of eliminating noise propagation in the bootstrapping process. We look to analyze recall using feasible methods. We are also going to further investigate understanding of semantic types through means of clustering.

References

Farzindar Atefeh and Wael Khreich. 2015. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164.

Hila Becker, Mor Naaman, and Luis Gravano. 2011. Beyond trending topics: Real-world event identification on twitter. *ICWSM*, 11:438–441.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Elisabeth Davenport and Blaise Cronin. 2000. Knowledge management: semantic drift or conceptual shift? *Journal of Education for library and information Science*, pages 294–306.

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science*, 315(5814):972–976.

Ruihong Huang and Ellen Riloff. 2013. Multi-faceted event recognition with bootstrapped dictionaries. In *HLT-NAACL*, pages 41–51.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189. Association for Computational Linguistics.

Ellen Riloff, Rosie Jones, et al. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics.

Alan Ritter, Oren Etzioni, Sam Clark, et al. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM.

Jagan Sankaranarayanan, Hanan Samet, Benjamin E Teitler, Michael D Lieberman, and Jon Sperling. 2009. Twitterstand: news in tweets. In *Proceedings of the 17th acm sigspatial international conference on advances in geographic information systems*, pages 42–51. ACM.

Kate Starbird and Leysia Palen. 2011. Voluntweeters: Self-organizing by digital volunteers in times of crisis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1071–1080. ACM.