Event Related Data Collection from Microblog Streams

Manoj K. Agarwal[1(✉)], Animesh Baranawal[2], Yogesh Simmhan[2], and Manish Gupta[1]

[1] Search Technology Center, Microsoft India, Hyderabad 500032, India
{agarwalm,gmanish}@microsoft.com
[2] Indian Institute of Science, Bangalore 560012, India
{animeshb,simmhan}@iisc.ac.in

Abstract. Many studies have established that microblog streams, e.g., Twitter and Weibo, are leading indicators of emerging events. However, to statistically analyze and discover the emerging trends around these events in microblog message streams, e.g., popularity, sentiments, or aspects, one must identify messages related to an event with high precision and recall. In this paper, we propose a novel problem of automatically discovering meaningful *keyword rules*, which help identify the most relevant messages in the context of a given event from fast moving and high-volume social media streams. For the specified event, such as {#trump} or {#coronavirus}, our technique automatically extracts the most relevant keyword rules to collect related messages with high precision and recall. The rule set is dynamic, and we continuously identify new rules that capture the event evolution. Experiments with millions of tweets show that the proposed rule extraction method is highly effective for event-related data collection and has precision up to 99% and up to 4.5X recall over the baseline system.

Keywords: Rules extraction · Event analysis · Streaming data · Algorithm

1 Introduction

The real-time and high-volume nature of microblog streams such as Twitter makes it a great source of information about recent or emerging events. In many existing works [22, 23], an event has been defined as 5W1H tuple – "What?", "Where?", "When?", "Who?", "Why?" and "How?". However, we use a more relaxed definition of an event as "*messages, posted by multiple users, in the same context, within a bounded time window*" [3, 10, 18, 24]. The "*context*" can be a real-world event, such as an earthquake or a football match or an abstract topic (e.g., a group of people discussing "climate change" on a particular day). Given an event, identifying the messages (e.g., tweets) related to the event can be useful for multiple applications such as understanding the sentiments towards the event [13], knowing the current hype around the event and predicting its future popularity [3], summarizing the event to obtain social news [4], predicting event location [5, 9], predicting election outcomes [12], detecting disaster events [24] and sending warnings [10], etc. For such applications, identifying the tweets or messages

C. Strauss et al. (Eds.): DEXA 2021, LNCS 12924, pp. 319–331, 2021.
https://doi.org/10.1007/978-3-030-86475-0_31

with a high recall is of utmost importance. Thus, a high-precision and high-recall system to collect event related tweets is a pre-requisite for such applications [30, 31].

Given an incoming Twitter message stream, a naïve way to obtain related tweets for an event is to query the stream using the event words or hashtags. For example, "*#brexit*" could be used to extract tweets for the event "United Kingdom withdrawal from the European Union". In many recent works [12, 24, 25], the tweets related to the events are collected using a small number of known hashtags. Similarly, most of these illustrative applications rely on hand crafted keyword rules [3, 12] (we refer to words and hashtags collectively as keywords). However, such a system would clearly have a poor recall, especially for events that are geographically and culturally diverse since events can have multiple popular representative words and/or hashtags used by potentially disjoint sections of Twitter users. Hence, such approaches are not only susceptible to bias, but also inadequate due to the dynamicity of real-world events.

Another method to collect event related tweets is to identify other keywords that co-occur frequently with the main event keyword [26, 30, 31], and fetch the tweets containing them. However, this results in low precision as frequently co-occurring words are also often part of other events. For example, in [26], the most frequently co-occurring words for the main event words {'thanksgiving, turkey'} were {'hope', 'happy'}. However, the tweets fetched based on just the keywords 'happy' or 'hope' will lead to low precision. Similarly, in [26], for main event words {'tiger', 'woods'}, the frequently co-occurring words found were 'accident' and 'crash' for a Tweet corpus from November 2009, when Tiger Woods had met an accident. Clearly, tweets fetched on just 'accident' will lead to low precision related to tweets about the event 'tiger wood car accident'. But, if we search for tweets containing the words {'tiger', 'accident'}, the collected tweets will be highly relevant. In this paper, a set of such keywords, used to fetch tweets related to an event with high precision, is called a rule. If we also use an additional rule {'tiger', 'crash'}, we get further tweets related to the main event.

If such rules related to an event can be identified automatically, the tweets related to the event can be collected with high precision and recall. Naturally, as the event evolves, the rules must also evolve accordingly.

In this paper, we propose a novel problem: For a given event, automatically identify a set of keyword rules such that these rules can be used to collect the event related tweets with high precision and high recall. The rule set is dynamic, and rules are added and removed from this rule set as the event evolves. It is natural that the keywords in the discovered rules also act as an event word cloud, and hence can be used to track the event; but the reverse is not true. Such a system is particularly useful for many applications to gauge the intensity, popularity, sentiments of an event or event summarization.

In summary, an event is represented using a disjunction of rules where each rule is a conjunction of multiple keywords. The characteristics of the rules are:

- The keywords must have high contextual frequency, i.e., they must occur in event related tweets with a high frequency.
- Keywords within a rule must be cohesive and representative of the event of interest.
- Rules should be able to capture the event evolution and the event drift, and hence must be dynamically updated.

The organization of this paper is as follows: In Sect. 2, we present the related work. In Sect. 3, we present our methodology, followed by description of the baseline system in Sect. 4. Next, we present the experimental results in Sect. 5. Finally, we present the conclusion in Sect. 6.

2 Related Work

In one of the earliest works, Li et al. [30], highlighted the need of a search system with high precision and high recall in the context of medical search, legal search, and social search, etc., and proposed a double-loop human supervised method. Similarly, in [31], authors presented a high precision, high recall system to extract tweets related to a given event and proposed a semi-supervised approach to identify the relevant keywords based on a word importance score. Finally, in a recent work [21], the authors proposed a supervised method to maximize the information coverage (i.e., high recall) for long running events and propose a human assisted method.

In summary, there exists a significant body of work to build a high precision and high recall system, to extract related traffic for the event of interest. However, the existing methods 1) mostly rely on supervised mechanism to come up with set of event related keywords, and 2) more importantly, each single keyword in this set is used as Boolean filter to extract event related traffic. Such systems either result in a low precision if a keyword is too generic, or a low recall if *only* highly informative keywords are discovered. In this paper, we propose to discover the *rules* comprising multiple keywords.

There has been significant work on tracking user specified or automatically discovered events. This work is collectively termed 'Event Tracking' [15–17, 27]. The objective of such systems is to monitor the evolving events. However, all these systems invariably identify and track the word cloud related to the source event, but their objective is not to collect event related tweets with high precision and recall. Although the word cloud evolves as the event progresses, it cannot be used to automatically create the rules unless a principled mechanism is defined to convert them into such rules. If a keyword rule is too complex (e.g., contains most of the keywords in an event word cloud), it will have a poor recall and if is it too simple, e.g., every keyword in the event word cloud becomes a rule by itself, it will have a poor precision.

Another relevant and well-studied problem is Query Expansion Techniques [7, 11, 19]. Existing works on query expansion techniques use keywords [1, 10, 13, 20], topic words [1, 8, 10], social factors [2], or visual contents [2]. Though query expansions methods over Microblog streams [6, 20, 21] appear close to our goal, the objective of such methods is to expand a query to retrieve additional semantically relevant tweets *only in the context of the given query*. On the other hand, our objective is to *continue* to retrieve the keyword rules for the entire event lifecycle. Thus, the query expansion methods cannot be trivially adapted to an event data collection system. Further, the objective of query expansion methods is to identify *semantically similar* alternate queries for improving the recall, whereas the keywords rules can be semantically different, covering different event aspects.

3 Methodology

The rule extraction problem is defined as follows.

Input: A set of one or more rules $\boldsymbol{R_0}$ representing an event $\boldsymbol{E}$ that needs to be monitored.

Output: A dynamic set of rules $\boldsymbol{R}$ for gathering the tweets related to the event $\boldsymbol{E}$ with high precision and recall during the event lifetime.

We consider $\boldsymbol{N}$ tweets at a time (called batch) over the input tweet stream.

Definition: Given a rule set $\boldsymbol{R}$, a tweet t is related to an event $\boldsymbol{E}$, if $\exists \boldsymbol{r} \in \boldsymbol{R}$ **such that** $\{\boldsymbol{r}\} \subseteq \{\boldsymbol{t}\}$. $\{r\}$ is the set of keywords in the rule r and $\{t\}$ is the set of keywords in tweet t.

Let T_{R_i} be the set of tweets filtered based on the rule set $\boldsymbol{R_i}$ in batch i. Note that, the rule set $\boldsymbol{R_0}$ (e.g., $\boldsymbol{R_0}$|'$\boldsymbol{r}$= #brexit') is defined by the user (specifying the event). Thus, the tweets in the set T_{R_0} represents the best estimate of ground truth. Similarly, we assume, tweets in batch i, matching the current rule set $\boldsymbol{R_i}$ represent the best estimate of ground truth for batch i.

We propose a two-step approach. In first step, we identify a set of candidate rules C_R. In second step, a rule $r \in C_R$ is added as the final rule if $(benefit_r/cost_r) \geq \alpha$, where the benefit and cost of a rule r is calculated over the tweets collected based on this rule. Our premise is that the keyword distribution in set T_{R_i} is the representative distribution of the event in iteration i. For a new rule r to be admitted into the rule set in iteration $i+1$, the distribution of the keywords in the tweet set T_r, filtered based on rule r, should be *similar* to this distribution. We next explain our detailed methodology.

First, we extract the set of frequent words W_i and hashtags H_i from tweets in T_{R_i} in batch i. For a given fraction threshold f, the keywords $K_{R_i} = \{W_i \cup H_i | f\}$ are considered where K_{R_i} is set of top f fraction of keywords in the batch (if $f = 0.05$, we consider top 5% most frequent keywords in set K_{R_i}). W_i represents the set of words and H_i is set of hashtags in K_{R_i}. A new rule is subset of keywords in K_{R_i}, $r = \{w | w \in K_{R_i}\}$.

Partitioning Keywords Based on Their Frequency: Next, we partition the set of keywords into buckets such that approximately *similar* frequency keywords are in the same bucket. Towards that, we sort the list of keywords in set K_{R_i} in non-increasing order of their frequency. If the frequency of two keywords is vastly different, one of them cannot be frequently co-occurring with the other in the tweets. Therefore, to identify the set of keywords with similar frequencies, we identify the change points on the frequency distribution curve of the keywords in K_{R_i}. A change point over a frequency distribution is a point where the mean frequency within a small window around it changes significantly. We use the classical CUSUM algorithm [28] to detect the change points over the frequency distribution. Keywords, corresponding to frequencies between two change points are put in the same bucket, representing a set of keywords with similar frequency.

Building Co-occurrence Graph: For each bucket of keywords, we induce a graph $G(V, E)$ over the keywords in the bucket such that each keyword is a node in G. Two keywords A and B have an edge between them *if* their co-occurrence score $\frac{|T_A \cap T_B|}{|T_A \cup T_B|} \geq \gamma$; γ $(0 < \gamma \leq 1)$ is the user specified threshold. Without loss of generality, T_A is set of tweets containing keyword A in batch i.

We identify all the cliques of size greater than one in the graph $G(V, E)$, such that the frequency of the clique is above a threshold t_h:

$$t_h = \frac{2fN\gamma}{1+\gamma} \tag{1}$$

Thus, each pair of keywords in the clique must appears in $\geq t_h$ tweets together. Note, $\frac{2\gamma}{1+\gamma} \leq 1$, and for a given fraction *f,* and a batch size of *N,* the minimum number of tweets in which a keyword appears is *f.N.* Thus, the threshold t_h ensures that the words in a clique are mostly co-appearing in the tweets. There is no limit on the clique size, however, due to limit on the tweet size, we find that cliques are no larger than six keywords.

The keywords in a clique become the candidate rules. Hence, a rule contains at least two keywords. However, if a clique of size one, with frequency greater than $2t_h$ (i.e., $2fN$) is a hashtag, it is also a rule. Note, popular hashtags for an event are likely to appear as single keyword rules (as they co-appear with many keywords).

Keyword Quality under a Rule Set: Given the i^{th} batch of N tweets, a keyword k and a rule set $\boldsymbol{R_i}$, we define the quality of keyword k under $\boldsymbol{R_i}$ as

$$Q(k|\boldsymbol{R_i}) = \frac{|T(k|\boldsymbol{R_i})|}{|T_k|} \tag{2}$$

T_k is the set of tweets containing keyword k in batch i (of N tweets) and $T(k|\boldsymbol{R_i})$ is set of tweets containing keyword k in set T_{R_i}.

Algorithm 1: Pseudo Code for Rule Extraction

```
Data: Incoming Tweets
Result: Extracted Rules
while True do
    for batch i do
        for tweet t in batch i do
            keywordList = GetHashtags(t) ∪ GetWords(t)
            /* update keywords profile */
            UpdateKeyWordsProfile(t, keywordList)          // Eqs. 2,3
            /* identify matching rules */
            rIDs = GetRuleIDs(keyWordList)
            for r ∈ rIDs do
                /* update candidate, existing Rule Profile */
                if r ∈ candidateRules then
                    /* compute cost and benefit */
                    UpdateCandidateRuleProfile(r, t, keywordList)
        /* cost-benefit analysis on candidate rules */
        AnalyzeCandidateRules()
        /* create new rules */
        CreateNewCandidateRules()
```

Tweet Quality under a Rule Set: Given the i^{th} batch of N tweets, a tweet t and a rule set $\boldsymbol{R_i}$, we define the quality of the tweet t under the rule set $\boldsymbol{R_i}$ as

$$Q(t|\boldsymbol{R_i}) = \frac{\sum_{k \in K_{R_i} \cap k \in t} Q(k|\boldsymbol{R_i})}{\left|k \in K_{\boldsymbol{R_i}} \cap k \in t\right|} \tag{3}$$

A candidate rule $\boldsymbol{r}$ is accepted as the final rule if in $(i+1)^{th}$ batch, (a) it adds many novel tweets containing hashtags and words in set K_{R_i} (high benefit). Benefit is calculated as $\sum_{t \in T_r} Q(t|\boldsymbol{R}_i)$ and (b) if the increase in frequency of hashtags/words in set K_{R_i} is more than the increase in frequency of hashtags not in K_{R_i} (low cost). Cost is calculated as $\left|T_{\boldsymbol{R}_{i+1} \cup \boldsymbol{r}} - T_{\boldsymbol{R}_{i+1}}\right|$. We define α as the ratio of benefit and cost. Thus, a rule is discovered in batch i, and if its impact is positive in batch $i+1$, it is accepted. The updated rule set is used to collect the matching tweets from the next batch of N tweets and the process continues. The overall method is illustrated in Algorithm 1.

4 Baseline

We choose the system presented in [19] as the baseline, as it comes closest to the objective of our system. The objective in [19] is to search about an event *retrospectively*, given query term(s). The twitter corpus is divided into batches of fixed time length, called timespan. A burstiness score of a word w, $b(w|T_i) = \frac{P(w|T_i)}{P(w)}$ is calculated for each word appearing in a timespan T_i where $P(w|T_i) = \frac{f_{w|T_i} + \mu \frac{f_w}{N}}{|T_i| + \mu|}$ and $P(w) = \frac{f_w + K}{N + K|V|}$. fw is the frequency of word w in the entire corpus, and $fw|Ti$ is the frequency of the word w in Ti. N is the total number of terms in the entire corpus, $|Ti|$ is the number of terms in timespan Ti. $|V|$ is the total size of the vocabulary. μ and K are the smoothing parameters. *Top-k* words are used to expand the initial

They divide the twitter timeline into batches, and consider keyword co-occurrence scores to identify correlated keywords, similar to our system. The key difference is, they consider all temporally correlated keywords. The assumption that temporally correlated keywords are related to the *same* event may not hold for disjoint events which, nonetheless, occur together. On the other hand, we consider temporally and spatially related terms (i.e., appearing in the same tweet) in the *context* of the event specific rule set. Further, unlike our system, where we dynamically discover rules, their system works on *entire* tweet corpus, and it is not a *live* system.

5 Experiments

5.1 Dataset

We consider two datasets: **DS 1**) 71.7 million tweets, from 17th Oct 2016 to 16th Dec 2016. **DS 2**) 61.5 million tweets from 5th March 2020 to 30th April 2020. Both these sets were collected using Twitter 1% traffic API. The two time periods were selected carefully, to represent tweets from two different periods, before Twitter changed its character limit from 140 to 280 characters per tweet in 2017. We used a small set of stop-words, to discard certain frequently occurring keywords.

Given a rule set $\boldsymbol{R}_i$ in i^{th} batch of tweets, the precision of a rule is defined as

$$prec(r|\boldsymbol{R}_i) = \frac{\sum_{t \in T_r} Q(t|\boldsymbol{R}_i)}{|T_r|} \tag{4}$$

Thus, each tweet contributes to the precision proportional to the quality of keywords it contains, w.r.t. rule set $\boldsymbol{R}_i$. Recall of rule r is computed as

$$rec(r|\boldsymbol{R}_i) = \frac{\sum_{t \in T_r} Q(t|\boldsymbol{R}_i)}{\sum_{t \in T_{R_i \cup r}} Q(t|\boldsymbol{R}_i)} \tag{5}$$

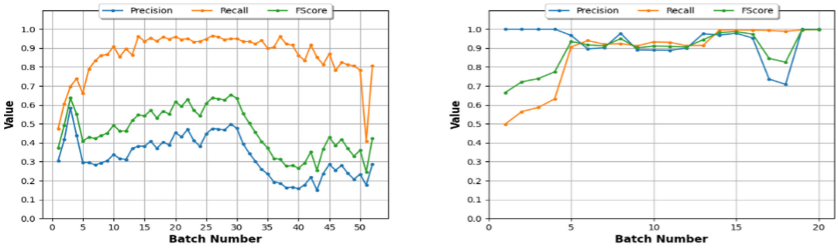
Thus, we compare the quality of the tweets collected due to a new rule r with the quality of all the tweets collected due to combined rule set of r and existing rules. As defined earlier, we consider the current rule set as the ground truth. Note, all tweets at $\boldsymbol{R}_0$ (e.g., $\boldsymbol{R}_0$|"$r = \#brexit$") are considered related to Brexit event. We considered three types of events, 1) Point events with lifespan of a day; 2) Medium term events, with lifespan of approximately a week; 3) Long term events, spread over multiple weeks.

5.2 System Configuration

We Apache Flink [14], a stream processing framework and deployed in local mode, supporting Java 11 to implement the described rule extraction framework. For each of our experiments, we set the following parameters, unless specified otherwise: $\gamma = 0.2$, $\alpha = 0.1, f = 2.5\%$ (top 2.5% keywords sorted on frequency are selected, in a batch). Note, the number of tweets collected by our system and base-line are different. Since we have a fixed batch size, the two systems may cover different number of batches (x-axis in Figs. 1, 2, 3 and 4).

5.3 Short Term Events

We evaluate a point event, *#earthday* (**DS2**). We set batch size $N = 800$ and $\gamma = 0.2$.



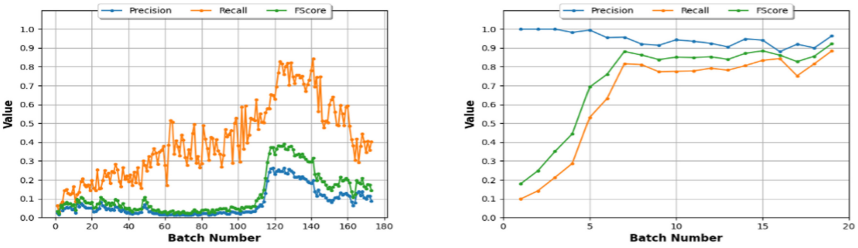
(a) Baseline (b) Our System

Fig. 1. Precision and recall for event *#earthday*

In Fig. 1, we plot the precision-recall of our system w.r.t. baseline. Note that, the first batch of tweets will always have a precision of 100% because we start with the event hashtag as the only seed rule. We see, the recall of our system improves quickly. The precision for our system drops due to addition of spurious rules, but then improves subsequently with addition of better rules. The baseline precision is low because of the presence of many unrelated keywords.

5.4 Medium Term Events

We selected the event with medium term impact, of up to seven days, namely #*blackfriday* (**DS1**). Figure 2 shows the trends due to #*blackfriday* event.



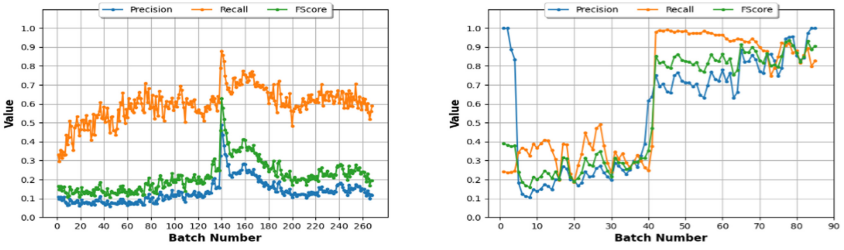
(a) Baseline (b) Our System

Fig. 2. Precision and recall for event # *blackfriday*

We see, the baseline trends are because of many keywords discovered as rules related to another event, namely, #*thanksgiving*. This leads to a significant drop in precision as well as recall. our system starts with a low recall but steadily improves. The precision of our system remains high throughout.

5.5 Long Term Events

We cover two long term events, #*trump* (during the trump election campaign) in **DS1** (Fig. 3), and Covid-19 Virus (#*coronavirus*), in **DS2** (Fig. 4).

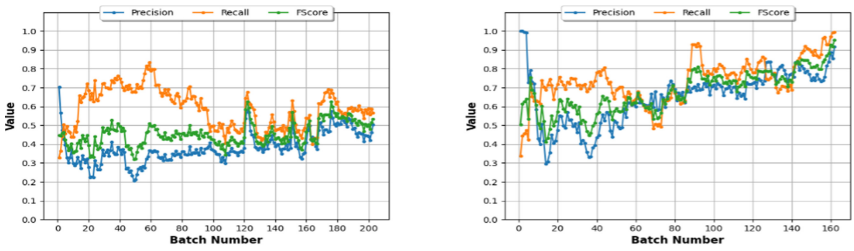


(a) Baseline (b) Our System

Fig. 3. Precision and recall for event # *trump*

We see, our system starts with low F1Score for #*coronavirus*, which improves slowly initially but subsequently adds many relevant rules.

For such long running events, determining the ground truth rules is difficult (as events change fundamentally multiple times over its lifespan), therefore, it is difficult to compute the precision-recall curve for the whole event. We, thus, zoom in around the time where they were most popular for a period of one week. Assuming that these long running events are relatively stable during the zoomed in period, we compare the baseline with our system in this time period.

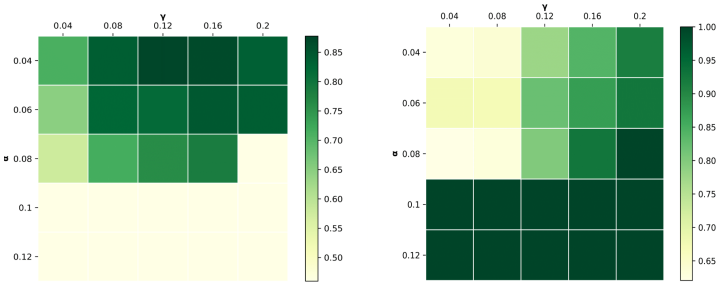


(a) Baseline (b) Our System

Fig. 4. Precision and recall for event *# coronavirus*

5.6 System Performance

We have primarily two tunable parameters, Score threshold α, and similarity threshold γ. The batch size variation did not impact the performance much, as in a short span, the event characteristics do not change much. A smaller value of γ and α implies more rules are added in the system. In the heatmap plot, in Fig. 5, we see, as α and/or γ reduce, the recall of our system improves, while reverse is true in case of precision.



(a) Recall Heatmap for #earthday (b) Precision Heatmap for #earthday

Fig. 5. System Performance under Parameter Tuning

Our system is very efficient: we could process the popular event #*trump* in 40 min. Since, the event is spread over the entire dataset, comprising ~71.7 million tweets, we get the processing speed of ≈30k tweets per second.

5.7 Human Judgements

In this experiment, the tweets retrieved by our system and the baseline system are judged by the human judges. We analyzed the rules over one day of peak period for each event. For an event, the systems discover multiple rules. We sample the rules and for each sampled rule, we uniformly randomly sampled m tweets, out of all the tweets filtered based on that rule, which were then presented to the judges. Thus, if an event has k rules discovered, up to $k.m$ tweets were judged (some rules filtered less than m tweets). The judges were trained in the task, i.e., they were briefed about the event, and asked to judge

if a tweet is related to the event or not. *m* is set to 20. We considered a rule relevant, if 80% or more tweets filtered based on the rule are considered relevant to the event by the judge. For relevant rules, we multiply the relevance score (for 80% threshold, it ranges from 0.8 to 1.0) with the numbers of tweets filtered based on that rule, to compute the final number of relevant tweets. The results are presented in Table 1 for 80% relevance threshold. We show relevant tweets vs. all tweets by two systems.

Table 1. Precision and recall (baseline vs. ours)

Event name	Rule precision (our system)	Rule precision (baseline)	Relevant/All (our system)	Relevant/All (baseline)
#earthday	45/58	2/11	5696/7148	2040/4722
#coronavirus	73/74	6/11	45216/47167	10334/14081
#trump	80/80	7/10	65043/65534	97901/121277
#blackfriday	14/16	3/11	6554/7772	3570/17616

For events #*trump*, our recall was inferior to the baseline. However, our precision at 99% was better. For #*earthday* and *#coronavirus* both the recall as well as precision are improved over baseline. For event #*coronavirus,* the recall was 4.5X over the baseline. However, as we would analyze next, even for #*trump* and *#blackfriday,* the quality of our rules is significantly better.

Table 2. Rules discovered by our system and baseline (rules in bold are relevant)

Event name	Rules – baseline	Rules – our system
#earthday	@adamcvean, @coco_who, @lowkeyel, @_mthegem, **#earthday**, **#earthday2020**	'**earth, mother'**, '**planet, protect**', '**pledge, save, water**', 'first, years', **#earthday**, **#earthday2020**, '**50th, anniversary'**, 'come, every', '**plant, trees'**
#coronavirus	**italians**, **cancellations**, **sanitizer**, **panicking**, #bestlyrics, **#coronavirus**, #kca	**'actually, people', 'house, president', 'declares, emergency', 'got, shit',** 'know, one'**, 'negative, tested, trump', 'covid19, news', #coronavirus, '#covid19, #covid'**
#trump	**elected**, **bernie**, protest, **obama**, **michelle**, #**trump**, #mannequinchallenge	**'president, states, united', 'racism, win', 'hope, win', 'make, today', 'lives, matter', 'america, great, make', #trump, #notmypresident, '#electionnight, #election2016'**
#blackfriday	moana, inner, **friday**, recount, **deals**, **#blackfriday**, #pizzagate, #thanksgiving	'chance, giveaway', 'like, people', '**black, friday'**, 'new, one' '**free, shipping**', **'#giveaway, #win', #blackfriday, '#amazongiveaway'**

In Table 2, we show some of the rules discovered by the two systems. For event, **#***trump***,** tweets were collected over Nov 9th and 10th, 2016. The baseline system collected more tweets than our system (as shown in Table 1), however, if we analyze the rules generated by the two systems, we see, most of the rules by the baseline system were *Obama, Michelle, and Bernie*. Even though, they were considered related to the event

'*Election of Donald Trump'*, but not precisely to the main intent *#trump*. Further, many unrelated hashtags such as # *mannequinchallenge*, which were co-occurring, were also identified. *#protest* was a mixed rule, with less than half of all the tweets related to election of Trump. On the other hand, the rules discovered by our system never deviated from the main intent. We identified rules related to all the major intents for this event, such as '*racism, win*' or '*hope, win*'. Even seemingly unrelated rules '*make, today*' were found to be related to the primary intent of the event.

Similarly, for *#blackfriday* (26–27 Nov 2016) many unrelated rules such as '*moana*' (about a movie), or *#pizzagate* were found by the baseline system. Similarly, unrelated rule *#thanksgiving* resulted into a huge loss of precision (Table 1). On the other hand, except two rules, all the rules by our system were related to the main event intent. Further, our system was able to remove co-occurring tweets related to *#thanksgiving*.

Finally, for *#coronavirus,* the rules discovered by our system covered varied intents. For instance, rule '*actually, people*' was about denying that anything serious called Coronavirus exists. Even a rule like '*got, shit*' that added 232 tweets was highly precise rule. '*house, president*' was a rule about 'Coronavirus bill passed by the house', etc.

6 Conclusion

We presented a novel system to automatically extract the meaningful keyword rules from live data streams, in the context of a given event with the objective of collecting event data with high precision and high recall. Ours is a first system that identifies conjunction of multiple keywords as rules. Our system could identify meaningful rules for events with varying dynamicity. The number of rules remained bounded. For long running events, the rules automatically captured the event evolution with high precision. Further, our system was highly efficient while computing the rules. Our future work direction is to automatically identify the different event facets using the rules.

References

1. Chen, C., Li, F., Ooi, B.C.: TI: an efficient indexing mechanism for real-time search on tweets. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, pp. 649–660 (2011)
2. Gao, Y., Wang, F., Luan, H.: Brand data gathering from live social media streams. In: Proceedings of International Conference on Multimedia Retrieval, pp. 169–176 (2014)
3. Gupta, M., Gao, J., Zhai, C.: Predicting future popularity trend of events in microblogging platforms. Proc. Am. Soc. Inf. Sci. Technol. **49**(1), 1–10 (2012)
4. Kwak, H., Lee, C., Park, H.: What is Twitter, a social network or a news media?. In: Proceedings of the 19th International Conference on World Wide Web, pp. 591–600 (2010)
5. Li, R., Lei, KH., Khadiwala, R.: TEDAS: a Twitter-based event detection and analysis system. In: 2012 IEEE 28th International Conference on Data Engineering, pp. 1273–1276 IEEE (2012)
6. Massoudi, K., Tsagkias, M., de Rijke, M., Weerkamp, W.: Incorporating query expansion and quality indicators in searching microblog posts. In: Clough, P., et al. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 362–367. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20161-5_36

7. Nagmoti, R., Teredesai, A., De Cock, M.: Ranking approaches for microblog search. In: 2010 IEEE/WIC/ACM WI-IAT, vol. 1, pp. 153–157. IEEE (2010)
8. O'Connor, B., Krieger, M., Ahn, D.: TweetMotif: exploratory search and topic summarization for Twitter. In: Proceedings of the ICWSM, vol. 4, no. 1 (2010)
9. Sadilek, A., Kautz, H., Bigham, JP.: Finding your friends and following them to where you are. In: Proceedings of the Fifth ACM WSDM, pp. 723–732 (2012)
10. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th WWW, pp. 851–860 (2010)
11. Teevan, J., Ramage, D., Morris, MR.: # TwitterSearch: a comparison of microblog search and web search. In: Proceedings of the Fourth ACM WSDM, pp. 35–44 (2011)
12. Tumasjan, A., Sprenger, T., Sandner, P.: Predicting elections with Twitter: what 140 characters reveal about political sentiment. In: Proceedings of ICWSM, vol. 4, no. 1 (2010)
13. Wang, H., Can, D., Kazemzadeh, A.: A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In: Proceedings of the ACL 2012 System Demonstrations, pp. 115–120 (2012)
14. Carbone, P., Katsifodimos, A., Ewen, S.: Apache flink: stream and batch processing in a single engine. In: Bulletin of the IEEE TCDE, vol. 36, no. 4 (2015)
15. Osborne, M., Moran, S., McCreadie, R.: Real-time detection, tracking, and monitoring of automatically discovered events in social media. In: Proceedings of 52nd ACL, pp. 37–42 (2014)
16. Lin, CX., Zhao, B., Mei, Q.: Pet: a statistical model for popular events tracking in social communities. In: Proceedings of the 16th ACM SIGKDD, pp. 929–938 (2010)
17. Weiler, A., Grossniklaus, M., Scholl, MH.: Event identification and tracking in social media streaming data. In: EDBT/ICDT, pp. 282–287 (2014)
18. Agarwal, MK., Ramamritham, K.: Real time contextual summarization of highly dynamic data streams. In: EDBT, pp. 168–179 (2017)
19. Metzler, D., Cai, C., Hovy, E.: Structured event retrieval over microblog archives. In: Proceedings of the Conference of NAACL-HLT, pp. 646–655 (2012)
20. Wang, Y., Huang, H., Feng, C.: Query expansion based on a feedback concept model for microblog retrieval. In: Proceedings of the 26th WWW, pp. 559–568 (2017)
21. Srikanth, M., Liu, A., Adams-Cohen, N.: Dynamic social media monitoring for fast-evolving online discussions. arXiv preprint arXiv:2102.12596 (2021)
22. Mu, L., Jin, P., Zheng, L.: Lifecycle-based event detection from microblogs. In: Companion Proceedings of the the Web Conference 2018, pp. 283–290 (2018)
23. Jin, P., Mu, L., Zheng, L.: News feature extraction for events on social network platforms. In: Proceedings of the 26th International Conference on WWW Companion, pp. 69–78 (2017)
24. Rudra, K., Goyal, P., Ganguly, N.: Identifying sub-events and summarizing disaster-related information from microblogs. In: The 41st International ACM SIGIR, pp. 265–274 (2018)
25. Phuvipadawat, S., Murata, T.: Breaking news detection and tracking in Twitter. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 3, pp. 120–123. IEEE (2010)
26. Guille, A., Favre, C.: Event detection, tracking, and visualization in twitter: a mention-anomaly-based approach. In: Proceedings of SNAM, vol. 5, no. 1 (2015)
27. Lee, P., Lakshmanan, LV., Milios, EE.: Event evolution tracking from streaming social posts. arXiv preprint arXiv:1311.5978 (2013)
28. Agarwal, MK., Gupta, M., Mann, V.: Problem determination in enterprise middleware systems using change point correlation of time series data. In: Proceedings of NOMS, pp 471–482, April 2006
29. Magdy, A., Abdelhafeez, L., Kang, Y.: Microblogs data management: a survey. VLDB J. **29**(1), 177–216 (2020)

30. Li, C., Wang, Y., Resnick, P.: ReQ-ReC: high recall retrieval with query pooling and interactive classification. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 163–172 (2014)
31. Zheng, X., Sun, A., Wang, S.: Semi-supervised event-related tweet identification with dynamic keyword generation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1619–1628 (2017)