Real-time Adaptive Crawler for Tracking Unfolding Events on Twitter

Asmelash Teka Hadgu
L3S Research Center
Hannover, Germany
teka@L3S.de

Sallam Abualhaija
Interdisciplinary Centre for Security,
Reliability and Trust
Luxembourg, Luxembourg
sallam.abualhaija@uni.lu

Claudia Niederée
L3S Research Center
Hannover, Germany
niederee@L3S.de

v1.1.1
2018/09/10

ABSTRACT

When a major event such as a crisis situation occurs, people post messages on social media sites such as Twitter, in order to exchange information or to share emotions. These posts can provide useful information to raise situation awareness and support decision making, e.g., by aid organizations. In this paper, we propose a novel method for social media crawling, which exploits a Bayesian inference framework to keep track of keyword changes over time and uses a counter-stream to gauge the inclusion of noise and irrelevant information. In addition, we present a framework to evaluate real-time adaptive social search algorithms in a reproducible manner, which relies on a semi-automated approach for ground-truth construction. We show that our method outperforms previous methods for very large scale events.

Keywords

adaptive crawler, real-time adaptive search, event tracking, crsis communication

INTRODUCTION

Nowadays, social media platforms have become an essential communication tool since they provide an open medium on a diverse spectrum of topics. As an accessible and near real-time reflection of such events and their perception, Twitter has attracted the attention of many researchers across different domains. Twitter content analysis is useful for different applications including but not limited to detecting customers' reactions towards a product (Chen et al. 2013), analyzing public political polarities (Conover et al. 2011), predicting election results (Tumasjan et al. 2010), monitoring news (Magdy 2013), identifying and analyzing researchers (Hadgu and Jäschke 2014) and crisis management (Castillo 2016). As a foundation for those analyses, the task of adequately collecting relevant content for an event of interest, has also gained some interest in the research community recently.

Since manually checking and changing keywords as an event unfolds, requires a lot of effort and is inefficient, different automatic query keywords adaptation strategies have been proposed (Wang, Tokarchuk, Cuadrado, et al. 2013; Magdy and Elsayed 2014; Sadri et al. 2016) to monitor unfolding events on Twitter. All different adaptive search methods try to adapt the search keywords as the event evolves through out the search process. An intuitive method is to consider the most frequent hashtags that appear in a time interval and use them as the next query keywords. However, this technique has major limitations. In particular, many people exploit trending hashtags through hashtag hijacking (Hadgu, Garimella, et al. 2013) to diffuse opposing views and unrelated content, e.g., marketing companies take advantage of frequent hashtags to promote their ads. Another problem is that the same

hashtag could be used for different events that may happen at the same time. The role of the adaptive search systems is to retrieve as many relevant tweets as possible without introducing too much noise.

Therefore, more advanced techniques that consider correlation of keywords to the underlying event or those based on text content similarity measures are applied to capture promising query keywords that preserve relevant content. In this paper, we present a novel adaptive crawling approach for Twitter which can be used to track a particular event in real-time. The adaptive crawler starts with a set of seed keywords given manually by the user. In contrast to existing methods, it does not rely on the filtered stream generated based on this keyword set (and its modifications over time) alone. It complements this stream with a second stream, an unfiltered sample stream, which is used as a kind of "counter-stream" to help eliminate keywords from co-occurring events. A Bayesian framework is used to identify new keywords based on both streams.

The absence of a robust evaluation framework to compare the performance of alternative real-time adaptive algorithms is a challenge. There are no standard benchmarks to compare the performance of different adaptive algorithms. Especially, evaluating the coverage or recall of a method is difficult, since the size and speed of the Twitter stream makes it impossible to capture the whole data set for a time frame - besides the limitations enforced by Twitter on the publicly available tweets. We evaluate our approach, compare and contrast it with related work mainly on the Berlin Attack 2016 [1]. For this crisis, we systematically constructed the ground truth in order to evaluate the various adaptive crawling methods. The method can be employed for creating further ground-truth data sets in an effective, semi-automatic fashion. The evaluation considers two dimensions, the temporal aspect demands the algorithm to adapt new relevant keywords as soon as they emerge. The other dimension includes the standard metrics; precision, recall and F1-measure of the retrieved relevant tweets.

In this work, we address the real-time adaptive search task within crisis management context. Our choice is motivated by several reasons. First, the proposed system has been developed within a research project dedicated to communication concepts during crises. Hence, we were able to involve practitioners from aid organizations into our requirements analysis, development and evaluations, strengthening the evaluation results. The outcome is an application called **Sover!** short for Social media observer (Hadgu, Abualhaija, et al. 2018). The system has been tested in real scenarios. Second, crisis management is an extremely important and challenging application with a growing role for social media (Castillo 2016). The temporal aspect is very important and the reliability of crawling results is weighed by the sensitivity of critical decisions made, at least partially, based on these results.

In summary, our main contributions are:

- An effective real-time adaptive search algorithm that leverages simultaneously both the public one percent Twitter stream and the Twitter filter stream and combines them using a Bayesian framework for improving recall as well as precision.
- A simulation environment to experiment with real-time adaptive search algorithms on Twitter in a reproducible manner.
- A well-defined strategy on how to construct a ground truth specific to evaluate the performance of real-time adaptive search algorithms on Twitter.

RELATED WORK

The work in this paper builds on two distinct yet complementary tasks. The first is how to search for relevant content to a specific event in social media. The second is how to apply these methods for a real application context, crisis management in our case. Therefore, we describe the related work corresponding to each task separately.

Real-time Adaptive Search. In (Wang, Tokarchuk, Cuadrado, et al. 2013), the authors proposed an adaptive crawling algorithm, Refined Keyword Adaptation (RKwA), by developing a recall-oriented query that exploits the emerging popular hashtags in the crawling method. In their follow-up (Wang, Tokarchuk, and Poslad 2014), the authors proposed an adaptive crawling method that monitors hashtags and rates them by their similarity to the initial seeds using TF-IDF weighting. In (Magdy and Elsayed 2014; Magdy and Elsayed 2016), the authors developed an unsupervised adaptive method to track tweets for a dynamic topic. A binary classifier is trained and updated automatically and regularly to detect more relevant tweets from the stream. Another work, relevant to ours, is (Sadri et al. 2016), the Tweet Acquisition System (TAS), a system that follows an iterative process to automatically change and adapt the query representing a temporal topic aiming at maximizing the recall. The main difference between **Sover!** and these approaches is that while all these methods rely on a filter stream, to do the adaptation, **Sover!** uses simultaneously the sample stream to better estimate relevant keywords.

[1] https://en.wikipedia.org/wiki/2016_Berlin_attack

Social Search for Crisis Communication. Several tools have been developed to monitor social media for Crisis Communication. Let us have a brief look at the ones that are closest to our work. Rogstadius et al. proposed CrisisTracker, an online real-time system that tracks keywords on Twitter during disasters and creates stories using clustering. The Twitter Streaming API is used to collect tweets based on some keywords provided by the user. Twitcident (Abel et al. 2012) is a web framework that relies on the emergency broadcasting services to enable detecting an incident automatically and monitoring relevant information diffused on Twitter. XHelp in (Reuter et al. 2015) is a cross-social-media application that supports volunteers in responding to disasters. It is implemented as an embedded Facebook application. Magdy developed TweetMogaz, a news portal of tweets in the Arabic regions including Syria and Egypt. TweetMogaz provides a summary of the public response in Twitter towards ongoing events. The Emergency Analysis Identification and Management System (EAIMS), proposed by by McCreadie et al., is a crisis tracking toolkit, that provides a real-time event detection as well as some other relevant functions like sentiment analysis, information credibility estimation and automatic time-line generation. **Sover!** builds on ideas from these applications. It starts with Twitter but expands to Facebook pages, YouTube links and news pages associated with the tweets, which are significant for relief organizations.

ARCHITECTURE

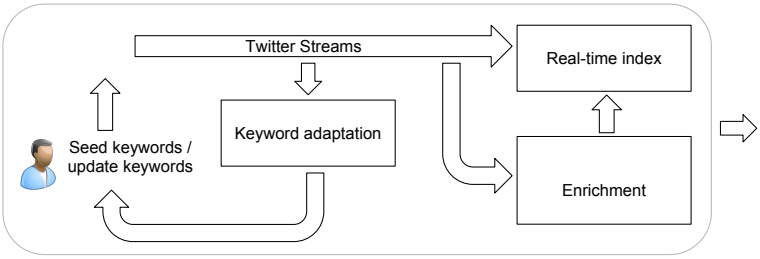


Figure 1. System Architecture of Sover!

In this section, we present the overall architecture (cf. Figure 1) and the various components of our adaptive search system **Sover!** . To start monitoring an event, a user begins the search by supplying a typically small set of initial seed 'keywords' that describe the event. E.g., to monitor social media feeds about the Berlin attack in German, one might supply 'Berlin Anschlag' ('Berlin Attack'). The system uses Twitter streams to search tweets containing these initial search terms initially. After a small time interval, the system dynamically changes the query keywords. The keyword adaptation module is used to compute these changes. A dashboard of social feeds from different platforms presents social media posts on the event to the user. A real-time indexing module and URL expansion and enrichment module support this dashboard. Each of these components is discussed below.

Twitter Streams. Sover! leverages the real-time Twitter streaming API to monitor events in real-time. This is mainly done through the **Filter Stream** [2] in combination with the **Sample Stream** [3]. The **Filter Stream** allows a searcher to supply parameters such as track to filter feeds by keywords, follow to filter by accounts or location to use bounding boxes to filter tweets from a specific area. These parameters can be changed in real-time to reflect the changing nature of an evolving event. Our system also supports searching past events. This employs data from Twitter Advanced Search [4]. The Twitter Advanced Search serves as a gateway to tap into the entire index of tweets. In order to obtain similar payload as a searcher gets from the **Filter Stream**, we combine this with the Twitter status look-up API [5]. We describe our implementation in the Evaluation Framework Section.

Keyword Adaptation. The main task of the keyword adaptation module is to take in two streams, the sample stream and filter stream and perform a statistical analysis on these collections in order to include new keywords that emerge from the collection to add in the next iteration or drop keywords from the previous query that are no longer relevant as search terms. The decision whether to update the query keywords or not takes place at discrete intervals. For instance, for a real-time event, after initialization through the user, we observe the stream for a time window, e.g., 5 min, then see if there are emerging keywords that should be included in the subsequent request to the stream. We also check if keywords in the search query are still relevant. The method behind this module is the main contribution of this paper. The keyword adaptation algorithm is presented in detail in the next Section.

[2] https://stream.twitter.com/1.1/statuses/filter.json
[3] https://dev.twitter.com/streaming/reference/get/statuses/sample
[4] https://twitter.com/search-advanced
[5] https://dev.twitter.com/rest/reference/get/statuses/lookup

URL Enrichment. Twitter posts contain URLs to original sources such as images, videos that users upload from their devices but also shared links pointing to other social media platforms, news and other websites. Most of these URLs are shortened either by Twitter for display or by users using URL shortening services. Working with these shortened URLs is challenging. The URL enrichment module helps mitigate this problem. First it performs URL unshortening by performing an HTTP HEAD request. This not only serves as a way to check if the URL is valid but also to categorize links by domain. We mainly categorize domains as Twitter, Facebook, YouTube! and News Media. We use domains harvested from the GDELT project [6] to curate a list of news media domains. This helps to present content using an intuitive social feeds dashboard when monitoring events. For valid URLs pointing to news resources, this module also enriches the URLs with meta-data: such as article title, intro, and images if any to provide a snippet.

Real-time Index. We index the filtered and enriched streams of tweets on the fly using ElasticSearch [7]. We provide a mapping for tweets that makes it easy to recognize date-time, location fields etc. This helps to provide faceted search over all the Twitter fields as they get indexed in real-time from the social feed dashboard.

Human in the Loop. As the event a user is tracking unfolds, the keyword adaptation module provides new emerging keywords or omits those that are not relevant anymore from the set of search keywords. This happens automatically. Additionally, there is the option to bring the Human in the Loop: the user can intervene at any point via the user interface, if she thinks that the system adds noisy or irrelevant keywords or drops an important keyword. **Sover!** provides a user interface for the user to monitor and modify query keywords in a running crawler as necessary.

METHODOLOGY

Event tracking on Twitter is essential for many real-time applications. The dynamic nature of Twitter however makes real-time adaptive search challenging to achieve good coverage that guarantees gathering relevant tweets, without drifting away to other events happening at the same time. In this section, we present a novel adaptive search method for tracking dynamic events on Twitter and a simulation framework to evaluate such adaptive algorithms.

Real-time Adaptive Search

Let ς be a social media stream that consists of a set of social media feeds (posts) about a variety of events. Let us further assume that we can search this stream using query keywords $\varkappa$ to retrieve posts containing any of these keywords that describe an event ε that we are interested to track in real-time.

Definition 1 Filter Stream, F : a set of posts retrieved from ς using $\varkappa$.

Definition 2 Sample Stream, S : a random sample of posts from ς.

Problem Statement. Given the sub streams F and S, and a set of query keywords $\varkappa_i$ at a given time T_i, generate a set of query keywords $\varkappa_{i+1}$ to maximize the retrieval of posts relevant to ε at the next time step T_{i+1}.

First we generate candidate keywords $\varkappa_c$ from F. The keyword generation step is discussed in the Proposed Solution below. For each candidate keyword, k, we assign a score that quantifies how good this keyword is to ε. In information retrieval, tf-idf, is a commonly used numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. In our case, the score we are trying to compute for each k as a function of S and F is a tf-idf like metric. It assigns weights to keywords in F according to their frequency of occurrence. If we let $|F_k|$ be the number of posts containing the keyword k, we can give a similar estimate as follows:

$$P(k|F) = \frac{|F_k|}{\sum_i |F_i|} \tag{1}$$

Similarly, the second component assigns a score of how important a keyword k is for all other events in ς except ε ($= \varepsilon^c$). Since we can not observe ς directly, we can use S and $\varkappa$ to estimate ε^c. Let us denote posts in S that do not contain any of the query keywords $\varkappa$ as S^o, then Similar to Equation 1:

$$P(k|S^o) = \frac{|S^o_k|}{\sum_i |S^o_i|} \tag{2}$$

[6] http://www.gdeltproject.org
[7] https://github.com/elastic/elasticsearch

Finally, in order to incorporate prior information of keywords and change the probabilities as we gain more evidence over time as the event unfolds, we apply Bayes' theorem in order to compute the likelihoods given the priors and our current estimates. In the following, we provide a walk through of the pseudo code of our proposed solution given in Algorithm 1.

Algorithm 1 Keyword Adaptation

Input: $keywords$ at time T_i, $FStream$, $Sstream$, θ, min_freq, P
Output: $keywords$ at time T_{i+1}

```
for k ∈ Fstream do
    if freq(k) > min_freq then
        add k to candidates        ▷ generate candidates
    end if
end for
for c ∈ candidates do
    estimate P(h)
    estimate P(o)
    update likelihood P(c)
    if P(c) > θ then
        add c to keywords
        update P
    end if
end for
return keywords
```

Proposed Solution. Let E denote a particular event that we are trying to track on Twitter, e.g., Berlin Attack. Initially, this event is described by the set of tweets returned from Twitter as a result of our search containing the first query keywords – seeds. If we analyze the set of keywords, i.e., hashtags and word n-grams, in the initial response, it contains predominantly the initial search terms and other potentially useful query terms for future searches. The question is how to decide, which of the new keywords are useful as potential query keywords.

The main ingredient to our solution is to consider the global context instead of dealing solely with the resulting social feeds of a particular event that we are tracking in order to update the query parameters, i.e., including possibly other major events that are happening at the same time. Our insight is that we can leverage the one percent Twitter sample to get a sample statistic of the global context. This public stream gives us a random set of tweets covering all major global events (given they are big enough). By analyzing simultaneously both the random one percent sample of the entire Twitter stream and a focused stream of tweets about the event we are interested in, we can make better decisions about which keywords pertain truly to the event we are trying to track and which just co-occur because of another event or are just noise. This will help us to avoid keywords coming from other events thereby mitigating keyword drift and identifying emerging keywords in our filter stream that are not contained in the query keywords. In practice people use hashtags and phrases to perform searches about unfolding events. In our implementation we use hashtags and word bi-grams as candidate keywords.

Suppose r is the hypothesis that a given keyword is relevant for an event and can be used as a search keyword. We can quantify how good this keyword is as a search term for the event E by computing the fraction of relevant tweets that contain this keyword, $P(r)$ as given in Equation 1. The intuition behind this is that, a keyword that generates more relevant tweets for an event when used as a query term should get a higher probability of being used as a search query than a keyword that does not bring as many relevant tweets.

For the first iteration, this can be initialized in such a way that we penalize globally most frequent keywords such as #rt, #ff etc. with low probabilities, seed query keywords with high probabilities and the rest of the keywords with some small uniform probability. We can leverage past collection of tweets from the one percent Twitter sample to estimate these probabilities.

Similarly if we let o be the hypothesis that a given keyword is relevant for all other events, other than the event E that we're tracking, then $P(o)$ as given in Equation 2 quantifies how good this keyword is as a search query for all other events but E.

Concretely, we estimate $P(r)$ using the **Filter Stream** as the fraction of tweets containing the query keyword in our search result. In the same way, we estimate $P(o)$ from the **Sample Stream**. In particular, we take the fraction of tweets containing the keyword by taking all tweets from the Sample Stream except those that contain the query keywords that are used to retrieve the tweets for the event E at the current time-step. Finally, we update these probabilities in order to make potentially useful keywords get higher probabilities and those keywords that are in the query set but are not relevant anymore receive small probabilities so that they can be dropped in subsequent queries. For a given keyword, after computing the likelihood, we compute the ratio of the likelihoods to decide if a given keyword should be included as a query keyword in the next iteration. We set a threshold, θ that defines this cut off. Algorithm 1 gives the concrete steps for the computation.

Evaluation Framework

One of the challenges in evaluating algorithms for real-time tracking of event on social media is setting up a simulation environment, where one can run experiments multiple times and in a reproducible manner. This is in contrast to ad hoc approaches where one has to anticipate and monitor the social media platform as a particular event happens and then run competing algorithms against such collection. To enable reproducible experiments, we created a "simulation" environment to re-create Twitter streams. In this section, we describe exactly how we simulate the Twitter stream for evaluating real-time adaptive crawlers.

Simulation Real-time Stream. In a typical social search problem, one prepares training and test set collections for the retrieval problem. Real-time adaptive retrieval systems are then evaluated on such setup for comparison. However, this setup is limited in the sense that one must have collected the social feeds corresponding to every event we care about as they happened beforehand. This makes reproducibility challenging. Instead of explicitly building a test collection that maintains relevant and non-relevant feeds, we create a simulation environment that can help us play back past events.

The Twitter Advanced Search is a rich query interface that exposes all Tweets through the complete Tweet index [8]. Access is done via search operations using query words (such as exact phrases, any key words, exclusion words or hashtags), people (e.g., tweets written to or from accounts and mentioning accounts), places and dates. Thus, a searcher can go back in time and filter feeds by combining these parameters, and by replaying, this essentially gives us the best way to simulate real Twitter streams.

Of course the Twitter Advanced Search is not exactly the same as the Twitter real-time stream. There are a couple of drawbacks, when using Advanced Search. First, this only gives back organic tweets, i.e., retweets are not included and some of the tweets might have been deleted. Second, there is no official API that supports this. We solve the latter problem by writing a wrapper around this interface to mimic the same way a user invokes the Twitter real-time stream through Twitter Public Stream, e.g., **Filter Stream**. Each crawler in our experiments gets its tweets from the end point we described earlier. Our implementation of this query interface is Publicly available [9] for others to reproduce our work and use it for similar social search tasks.

Optimizing the Environment. This simulation framework gives each algorithm the same end-point and is not constrained by the tweet collection procedure common in previous works, i.e., this approach is not prone to leakage of one's approach to generating the real-time stream for evaluation in the first place and an adaptive approach that may leverage techniques from the generating process. It cleanly separates the evaluation framework from competing adaptive algorithms.

In the experiments, a user initiates an algorithm to track a given event with a starting seed query, the algorithm uses the Twitter Advanced Search wrapper and its task is to modify the query as appropriate at different time windows as time goes.

In our experience, this could be slow if for instance a crawler suggests a generic term (with lots of tweets) at the early stage of the event. To overcome this, we perform two optimization measures. Early stopping and caching.

- Caching: to avoid repeating the same query to Twitter, we have a caching step, that indexes tweets and query parameters as the query progresses. Only when a query was not yet seen does the algorithm fire an actual interaction to the Twitter Advanced Search. This simple caching technique reduces the experimentation time from potentially several days to minutes.
- Early Stopping: For a given past event, a crawler e.g., can perform the following GET request to track the keyword 'berlin' from 19:20:00 to 19:24:59 on December 19 2016. Instead of going all the way from the the last minute of the day of December 19th to the beginning. We stop after retrieving tweets at 19:20:00.

EXPERIMENT

In this section, we will describe our ground-truth generation approach. Besides, we will disucss the evaluation protocol and results of our experiments to validate (i) the simulation environment to compare real-time adaptive search algorithms and (ii) how well our proposed algorithm performs compared to existing methods.

[8] https://blog.twitter.com/engineering/en_us/a/2014/building-a-complete-tweet-index.html
[9] https://github.com/asmelashteka/twitteradvancedsearch

Ground truth Data

In most previous work, the common way to evaluate different adaptive algorithms has been to run several competing algorithms and then annotate a subset of the returned tweets using crowd sourcing. This raises the question what if the algorithms miss out a large fraction of the event? In the following we propose a novel ground-truth generation approach. The idea is to leverage the flexible simulation frame work we described in the Evaluation Framework Section that allows us to look ahead and back. This gives an advantage that real-time adaptive crawlers do not have.

Approach. There are existing approaches, e.g., (McMinn et al. 2013), to build Twitter collections for events on Twitter. McMinn et al. proposed a methodology for generating Twitter corpus for event detection. This is however limited to a small number of tweets, up to 1000, per event. Our approach to building the ground-truth data mimics running a temporal adaptive crawler with relevance feedback of tweets from user annotations. Concretely, we start with seed keywords to collect the first tweets that describe the event. After this iteration, we generate candidate keywords of up to 400 keywords, a limitation imposed by Twitter, and label which keyword is relevant. In order to determine whether a given keyword is relevant or not, we look-ahead the tweets containing this keyword in the next iteration and assess it with the following two metrics.

1. Information Gain: What is the gain in terms of potential relevant tweets that are not covered by the query set in the current iteration? A potential keyword is considered a candidate keyword if it brings about at least a minimum number of tweets (e.g., 10) that wouldn't be generated with our current keywords.
2. Signal to Noise Ratio: Checks how many of the additional tweets from the candidate query, i.e., not covered by our current keywords, are relevant and how many are non-relevant? Here we use human annotators to inspect sample tweets and label the tweets as relevant or not. A potential keyword is included for the next iteration, if the relevance of the additional tweets meets a certain threshold, 70%.

Since no real-time adaptive search algorithm has the possibility to look-ahead into the future, we can guarantee that this is indeed the best choice a real-time adaptive algorithm can achieve.

Implementation. Concretely, we build a ground truth data for Berlin Attack. This has two desirable properties to test adaptive social search algorithms on. First, this is a big world wide event. This means, the algorithm has to deal with a big scale event. Second, there were other big global events happening on the same day. This creates a challenge for an algorithm to drift away from the event of interest.

Evaluation and Results

We compare the performance of our proposed approach with RKwA (Wang, Tokarchuk, Cuadrado, et al. 2013). Even though TAS (Sadri et al. 2016) would be another candidate, it is not easy to faithfully replicate as the results change drastically depending on the news article used to guide the search. RKwA and our system have the same setup in that they do not require any external dependency other than starting the search process with a set of seed query keywords. The experiment was to gather as many relevant tweets to the 2016 Berlin Attack as possible by modeling the query keyword dynamics of the event, i.e., by adding new relevant keywords and dropping obsolete ones. This is done every five minutes for the first two hours of the event. The ground truth data is constructed by manually simulating the adaptive crawler processes with the help of a crowd sourcing task. We used the seed keywords: #berlin, berlin christmas market, berlin weihnactsmarket. The ground-truth as well as the adaptive algorithms were run since: '2016-12-19 19:05:00' until: '2016-12-19 20:55:00.

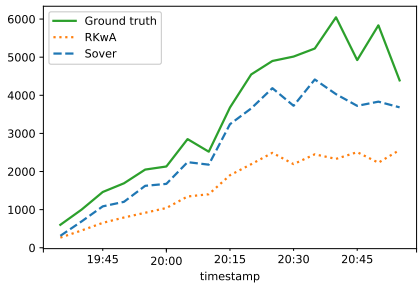
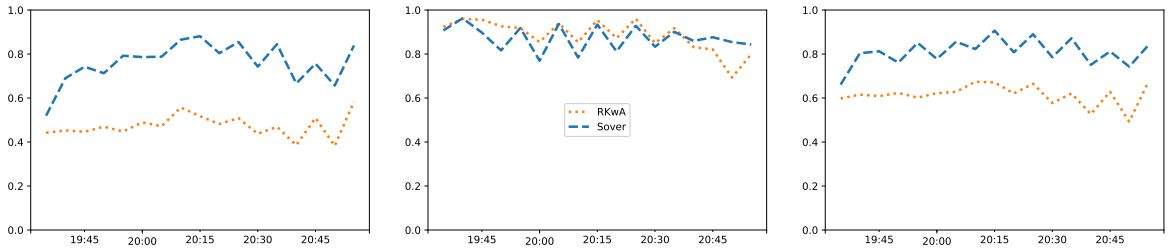


Figure 2. Number of relevant tweets gathered by (i) Sover! (ii) RkWA and (iii) the annotated ground-truth over the first two hours since the start of the event.

Relevance. This measures the total number of relevant tweets retrieved. In this case, the number of tweets that are about the Berlin Attack 2016. Figure 2 shows a comparison of the number of relevant tweets retrieved by the different algorithms. We see that **Sover!** performs better.



(a) Recall **(b) Precision** **(c) F1-score**

Figure 3. Evaluation of Real-time Adaptive Crawlers for Berlin Attack 2016 during the first two hours of the event.

Recall. is the fraction of relevant tweets that have been retrieved over the total amount of relevant tweets. Figure 3a shows that **Sover!** has significantly better recall than RKwA.

Precision. is the fraction of relevant tweets among the retrieved tweets. Figure 3b shows that **Sover!** achieves a higher recall without degrading on the precision.

F1-Score. is the weighted harmonic mean of precision and recall. Figure 3c shows that **Sover!** has overall better F1-Score than RKwA. It achieves a significant improvement with an average (over all five minute windows) F1 score of 0.81 for Sover! vs. 0.62 for RKwA.

Large-scale Crisis Simulation Exercise

Sover! was tested in a large-scale simulation exercise that was organized by the German Red Cross. The exercise was inspired by the 2013 European floods [10] crisis, in which Germany was affected. The main goal of the exercise was (1) to test communication methods, going beyond traditional means of communication and (2) to collect information from social media, mainly Twitter, using **Sover!** in order to enhance situational awareness. **Sover!** was particularly useful during the exercise to visualize the scale of the crisis through relevant posts that included images. This emphasizes the importance of visual content to enhance situational awareness. The keyword adaptation of **Sover!** was also useful. We observed that the keywords, e.g., #hochwasser ("flood") and #sturm ("storm") were added by the crawler automatically and reflected the exercise's scenario as the event was happening in the field.

CONCLUSION

In this work we presented the problem of real-time adaptive search. The challenges are: how to capture the evolution in the underlying vocabulary in order to dynamically change query keywords to track an unfolding event without adding too much noise. We proposed a method that solves this problem by leveraging simultaneously both a random sample of the Twitter stream and a focused stream about the event. The evaluation has shown that our method improves the performance of an adaptive Twitter crawler as compared to state-of-the-art methods. We also proposed a novel semi-automated ground-truth generation method in order to evaluate our proposed solution. This is also expected to bring the area forward by leading to more systematic approaches for evaluating adaptive crawling and social search technology. As a future work, we will explore how we can learn the ground-truth generation so that we can automate this process and apply it to similar events.

ACKNOWLEDGMENTS

We would like to thank Damianos P. Melidis, Negacy D. Hailu and the anonymous reviewers for their valuable feedback on the manuscript.

[10] https://en.wikipedia.org/wiki/2013_European_floods

REFERENCES

Abel, F., Hauff, C., Houben, G.-J., Stronkman, R., and Tao, K. (2012). "Twitcident: fighting fire with information from social web streams". In: *Proceedings of the 21st International Conference on World Wide Web*. ACM, pp. 305–308.

Castillo, C. (2016). *Big Crisis Data: Social Media in Disasters and Time-Critical Situations*. Cambridge University Press.

Chen, J., Cypher, A., Drews, C., and Nichols, J. (2013). "CrowdE: Filtering Tweets for Direct Customer Engagements." In: *ICWSM*. Citeseer.

Conover, M., Ratkiewicz, J., Francisco, M. R., Gonçalves, B., Menczer, F., and Flammini, A. (2011). "Political Polarization on Twitter." In: *ICWSM* 133, pp. 89–96.

Hadgu, A. T., Abualhaija, S., and Niederée, C. (2018). "Sover! Social Media Observer". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, pp. 1305–1308.

Hadgu, A. T., Garimella, K., and Weber, I. (2013). "Political hashtag hijacking in the US". In: *Proceedings of the 22nd International Conference on World Wide Web*. ACM, pp. 55–56.

Hadgu, A. T. and Jäschke, R. (2014). "Identifying and analyzing researchers on twitter". In: *Proceedings of the 2014 ACM conference on Web science*. ACM, pp. 23–32.

Magdy, W. (2013). "TweetMogaz: a news portal of tweets". In: *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 1095–1096.

Magdy, W. and Elsayed, T. (2014). "Adaptive Method for Following Dynamic Topics on Twitter." In: *ICWSM*.

Magdy, W. and Elsayed, T. (2016). "Unsupervised adaptive microblog filtering for broad dynamic topics". In: *Information Processing & Management* 52.4, pp. 513–528.

McCreadie, R., Macdonald, C., and Ounis, I. (2016). "EAIMS: Emergency Analysis Identification and Management System". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, pp. 1101–1104.

McMinn, A. J., Moshfeghi, Y., and Jose, J. M. (2013). "Building a Large-scale Corpus for Evaluating Event Detection on Twitter". In: *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*. CIKM '13. San Francisco, California, USA: ACM, pp. 409–418.

Reuter, C., Ludwig, T., Kaufhold, M.-A., and Pipek, V. (2015). "XHELP: Design of a cross-platform social-media application to support volunteer moderators in disasters". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, pp. 4093–4102.

Rogstadius, J., Vukovic, M., Teixeira, C., Kostakos, V., Karapanos, E., and Laredo, J. A. (2013). "CrisisTracker: Crowdsourced social media curation for disaster awareness". In: *IBM Journal of Research and Development* 57.5, pp. 4–1.

Sadri, M., Mehrotra, S., and Yu, Y. (2016). "Online Adaptive Topic Focused Tweet Acquisition". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, pp. 2353–2358.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., and Welpe, I. M. (2010). "Predicting elections with twitter: What 140 characters reveal about political sentiment." In: *ICWSM* 10.1, pp. 178–185.

Wang, X., Tokarchuk, L., Cuadrado, F., and Poslad, S. (2013). "Exploiting hashtags for adaptive microblog crawling". In: *Proceedings of the 2013 ieee/acm international conference on advances in social networks analysis and mining*. ACM, pp. 311–315.

Wang, X., Tokarchuk, L., and Poslad, S. (2014). "Identifying relevant event content for real-time event detection". In: *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. IEEE, pp. 395–398.