Event Detection on Microposts: a Comparison of Four Approaches

Akansha Bhardwaj, Albert Blarer, Philippe Cudré-Mauroux, Vincent Lenders, Boris Motik, Axel Tanner, and Alberto Tonon

Abstract—Microblogging services such as Twitter are important, up-to-date, and live sources of information on a multitude of topics and events. An increasing number of systems use such services to detect and analyze events in real-time as they unfold. In this context, we recently proposed *ArmaTweet*—a system developed in collaboration among armasuisse and the Universities of Oxford and Fribourg to support semantic event detection on Twitter streams. Our experiments have shown that *ArmaTweet* is successful at detecting many complex events that cannot be detected by simple keyword-based search methods alone. Building up on this work, we explore in this paper several approaches for event detection on microposts. In particular, we describe and compare four different approaches based on keyword search (*Plain-Seed-Query*), information retrieval (*Temporal Query Expansion*), Word2Vec word embeddings (*Embedding*), and semantic retrieval (*ArmaTweet*). We provide an extensive empirical evaluation of these techniques using a benchmark dataset of about 200 million tweets on six event categories that we collected. While the performance of individual systems varies depending on the event category, our results show that *ArmaTweet* outperforms the other approaches on five out of six categories, and that a combined approach offers highest recall without adversely affecting precision of event detection.

Index Terms—Event Detection, Microposts, Query expansion, Word Embeddings, Temporal query expansion, Semantic Inference

1 Introduction

Twitter is a popular microblogging service. Tweets typically contain up to 140 characters[1] and form an important source of instant information on any topic including celebrity gossip, entertainment, news, and more. Although the information present in tweets is often fragmented and noisy, Twitter users often provide live updates on important events; for example, more than 3.4 million tweets were sent in the first 24 hours after the *Charlie Hebdo* terror attacks.[2] Most tweets can be read by unregistered users, so Twitter can potentially provide a real-time source of information for detecting newsworthy events before these are covered by conventional broadcast media channels. Consequently, the development of techniques for tweet analysis and event detection has attracted considerable attention [1].

Armasuisse Science and Technology, the R&D agency of the Swiss Armed Forces, is developing a *Social Media Analytics* system that aims to help analysts detect security-related events. Initially, the armasuisse research team used standard Information Retrieval (IR) techniques [2], [3] for event detection, but these failed to detect many relevant complex events. For example, to detect deaths of politicians, an analyst would query the system using keywords such as "politician" and "die", but this resulted in low precision and low recall. Reliable detection of such events seems to require understanding the intended meaning of the query, knowing which individuals can be classified as "politicians", and identifying tweets that mention such individuals as a subject of a verb that describes the act of dying.

ArmaTweet [4] is a semantic event detection system developed in a collaboration among armasuisse and the Universities of Fribourg and Oxford. The system leverages advanced NLP techniques to generate structured representations of tweets that are integrated with external knowledge bases (DBpedia and WordNet) to create a RDF knowledge graph. Users can describe the relevant event categories as semantic queries over the generated knowledge graph. *ArmaTweet* then evaluates such queries to retrieve the relevant tweets and uses statistical anomaly detection to determine whether these tweets correspond to actual events.

While *ArmaTweet* showed superior results compared to standard keyword search, it was unclear how the system compares against more advanced techniques. To answer that question, we embarked on a rather ambitious project of identifying and empirically comparing several micropost event detection techniques on a large benchmark dataset. This dataset contains about 200 M tweets obtained using Twitter's public API. In this paper we summarize our results by presenting the four different event detection methods we considered and comparing their performance. Specifically, we compare *ArmaTweet* against a keyword-based *Plain-Seed-Query* (*PSQ*) approach, and two advanced query expansion techniques. In particular, we consider an adapted and extended version of *Temporal Query Expansion* (*TQE*) by Metzler et al. [5], which exploits co-occurrences of terms to expand a seed query to a larger query that retrieves the relevant tweets. Furthermore, we developed a new *Embedding* query expansion approach, which uses the well-known Word2Vec word embeddings by Mikolov et al. [6] to map vocabulary terms into vectors in a high dimensional space so that terms appearing in similar contexts are mapped to similar vectors. In this paper, we show how these four techniques compare on the task of detecting events from six different event categories described in Section 3.2. Our contributions can be summarized as follows.

1. Twitter raised this limit to 280 in 2017.
2. https://www.theguardian.com/media/2015/jan/11/charlie-hebdo-social-media-news-readers

1) We confirm that Twitter is a valuable source of information about security-related events.
2) We describe *ArmaTweet*, a novel event detection system that integrates several knowledge sources in an RDF knowledge graph, which is then used to identify events by means of semantic queries.
3) We describe two event detection methods that use query expansion techniques based on temporal co-occurrence and word embeddings.
4) We present the results of an extensive empirical evaluation of four very different event detection techniques on a large corpus of tweets.

2 Related Work

A recent survey of event detection methods on Twitter [1] identified three broad groups of techniques that extract *unspecified events*, *predetermined events*, and *specific events*.

The approaches from the first group target unspecified events—that is, events of general interest with no a priori description. These approaches typically extract features from (clusters of) tweets that reflect the tweets' topic and then analyze trends in the features to identify events [7], [8], [9]. Several systems detect trending news topics [10], [11], [12], and one additionally classifies events into predefined categories such as "sports", "death", or "fashion" [13].

The approaches from the second group detect events of predefined and fixed categories, such as concerts [14], controversial events [15], local festivals [16], earthquakes [17], crime and disaster events [18], or disease progression [19]. The EMBERS system [20] goes a step further by aggregating many sources of information (Twitter, Web searches, news, blogs, Internet traffic, etc.) to detect and predict instances of civil unrest. Such approaches typically involve training a classifier on manually annotated tweets to learn the correlations of features identifying tweets related to the topic of interest. Extending such systems to a new event category thus typically requires significant effort since one must produce a new training set and retrain the classifier.

The approaches from the third group detect events that match an explicit description of the relevant event types, and so they can usually be extended more easily to new event categories. They typically use Information Retrieval (IR) methods to match a term (aka keyword) query to a database of tweets. Queries are either provided by the user or are learned from the context [21]. These techniques have been combined with geographical proximity analysis to detect civil unrest and model events in Twitter streams. The work we present in this paper broadly falls into this category. In particular, we see *ArmaTweet* [4] as belonging to this group, even though it uses semantic instead of term queries to describe the relevant event types.

We next discuss in more detail various *query expansion* [22] techniques, which form the basis for *TQE* and *Embedding*. The main objective of query expansion is to reformulate a seed query into a new query that overcomes the word mismatch of keyword retrieval models and thus improves recall. Query reformulation typically involves extending the query with new terms, and it can be automatic, manual, or user-assisted. A recent survey [23] classifies query expansion techniques into three groups based on whether the reformulation uses corpus-dependent knowledge models, relevance feedback, or language models.

Corpus-dependent language models assume that pairs of words that often occur together in corpus documents talk about the same topic [24]. Roughly speaking, corpus documents are first clustered based on their similarity; then, to expand a query, query terms are mapped onto one or more clusters, and the terms of these clusters are used to expand the original query. Relevance feedback is another well-established query expansion approach, where a query is first evaluated against a corpus, and then the terms from the retrieved documents are used to expand the original query [25]. Another prominent approach builds a statistical language model in form of a probability distribution over terms [26], and uses this model to select the terms for query expansion. The *TQE* approach uses ad hoc relevance feedback, and our *Embedding* approach can be roughly understood as using a language model.

Finally, query expansion techniques can be further divided into two categories based on what data is used for query expansion. Global query expansion methods reformulate query terms independently of the query and its results, but they may include external sources of information such as WordNet. In contrast, local methods expand a query by taking into account the documents that match the query. Methods involving relevance feedback fall under the category of local methods. The *TQE* approach uses relevance feedback for query expansion, and the *Embedding* approach uses a Word2Vec model learned on Twitter data; thus, both techniques belong to the group of local methods.

3 Datasets & Event Categories

We are unaware of any publicly available, large collection of tweets suitable for the evaluation of event detection methods. Thus, we created our own datasets for training and evaluating our techniques, as well as the event categories we considered in our evaluation, which we describe next. We used the same training (if applicable) and evaluation datasets in all four approaches. Following Twitter's content redistribution terms,[3] we published the tweet IDs,[4] from which the Twitter's API can retrieve full tweets.

3.1 Training & Evaluation Datasets

Both *TQE* and *Embedding* include a training phase, although they do not require any labels—that is, they use unsupervised learning. We used a collection of about 500 M tweets in English collected in 2014 using Twitter's streaming API that returns 1% of all posted tweets. Moreover, we evaluated our techniques on a collection of about 200 M tweets in English collected in the same way during the first half of 2015.

3.2 Event Categories

We evaluated our approaches using the following six event categories stemming from our work on *ArmaTweet* [4].

- The "Aviation accident" category aims to detect crashes of airplanes. We observed in our evaluation

3. https://developer.twitter.com/en/developer-terms/more-on-restricted-use-cases.html
4. https://github.com/eXascaleInfolab/Event-Detection-Twitter

that *ArmaTweet* detected only incidents involving an airline, whereas other approaches also detected incidents involving small planes not related to civil aviation. To analyze this more closely, we introduced an "airline only" subcategory of this category.
- The "Cyber-attack on a company" category aims to detect cyber-attacks against known organizations.
- The "Capital punishment in a country" category aims to detect executions of criminals by a state.
- The "Militia terror act" category aims to detect terror attacks carried out by known militia or terrorist organizations.
- The "Politician dying" and "Politician visiting a country" categories aim to detect a known politician either dying or participating in a state visit.

As we have described in more detail in our previous work [4], these categories were identified in a workshop with armasuisse as relevant to the security context, as well as covering different types of event descriptions. In this paper, we do not consider the "Unrest in a country" category from our earlier work: it is defined by a predicative complement and is thus analogous to the "Capital punishment in a country" category, and its events overlap significantly with those of the "Militia terror act" category.

4 Event Detection Approaches

In this section, we describe the four event detection approaches that we consider in this paper: *PSQ*, *TQE*, *Embedding*, and *ArmaTweet*. All four approaches are realized as processing pipelines shown in Figure 1. For each event category, each of the four approaches outputs zero or more tweet time series—that is, a set of tweets identified as relevant to the event category grouped by the occurrence date. In all four cases, these time series are further passed to an event detection component that extracts zero or more events from each time series. This step is exactly the same in all four cases, and is described in more detail in Section 4.5. In contrast, the four approaches extract the time series from the evaluation dataset in radically different ways, which we describe in more detail in Sections 4.1 to 4.4.

4.1 Baseline: Plain Seed Query (*PSQ*)

In our baseline approach, an event category is manually described using a seed query expressed as a Boolean expression over terms (henceforth, referred to as *Boolean seed query*) that aims to identify tweets relevant to the event category. All seed queries used in our evaluation are shown in Appendix A.1. To detect the events matching a category, the corresponding term query is evaluated on tweets in the evaluation dataset and all matching tweets are retrieved. A single time series is then produced by grouping tweets by days of their occurrence. Since tweets are quite short, we do not believe that using elaborate IR ranking functions (such as BM25 [27]) would yield significantly better performance.

4.2 Temporal Query Expansion (*TQE*)

Query expansion is a common IR method that reformulates a term query by introducing additional terms that improve the recall of the retrieval. This task can be very challenging for microposts, which are short and often use informal language that does not contain the query terms; this is known as the *vocabulary mismatch problem* [28]. This problem can be addressed by query expansion with pseudo-relevance feedback. The main idea is to augment the seed query with terms that appear in the initial top-k retrieved documents, and it can be further extended to also consider the temporal dimension. In particular, Efron et al. [29] hypothesize that, in search tasks where time plays an important role, relevant documents tend to be clustered in time. By taking this idea into account, Metzler et al. [5] recently presented a temporal query expansion approach suitable for event detection from micropost archives. The algorithm includes two basic steps. First, it computes an expanded set of query terms that co-occur in many tweets clustered in time. The main idea is to use term co-occurrences to derive additional related terms, including shortened words, slang words, or hashtags, that might be relevant. Second, the algorithm uses this expanded set of terms to identify events and associate them with structured and meaningful summaries.

Our *Temporal Query Expansion* (*TQE*) approach refines the approach by Metzler et al. [5]. For each event category, we start with the same *Boolean seed query* as in *PSQ* (see Appendix A.1). The query is first expanded using the training dataset into a set of weighted terms using the following steps, which broadly follow the first part of the algorithm by Metzler et al. [5].

1) Use the terms of the seed query to generate a scoring for each hourly timespan in the training dataset.
2) Select the top-scoring *pseudo-relevant* time-spans. Metzler et al. [5] do not specify how many time-spans to select, but rather indicate that this depends on the event type. In our work, we automated this step using the Tukey outlier test [30].
3) For *all* terms occurring in tweets that belong to these pseudo-relevant time-spans, calculate the *burstiness score*. This produces a ranked list of terms that temporally co-occur with the seed query terms.
4) Select a set of top-scoring terms as the new, expanded query terms (with their weights). We select the top 15 terms as a reasonable compromise across the different event types, as there is no natural cutoff and none was given in the original paper. We show the resulting terms in Appendix A.2.

After deriving the expanded set of terms with weights, we depart from the approach by Metzler et al. [5] and proceed as follows.

5) Use the 15 top-scoring terms, including the relative weights, to query the evaluation dataset and thus derive a single time series of weighted daily counts of relevant tweets.
6) Use the event detection technique described in Section 4.5 to identify zero or more events from this tweet time series.

This process computes a set of weighted expanded query terms without any information about the Boolean connections between the terms. This is an unfortunate restriction of the original algorithm. Our experiments showed that using

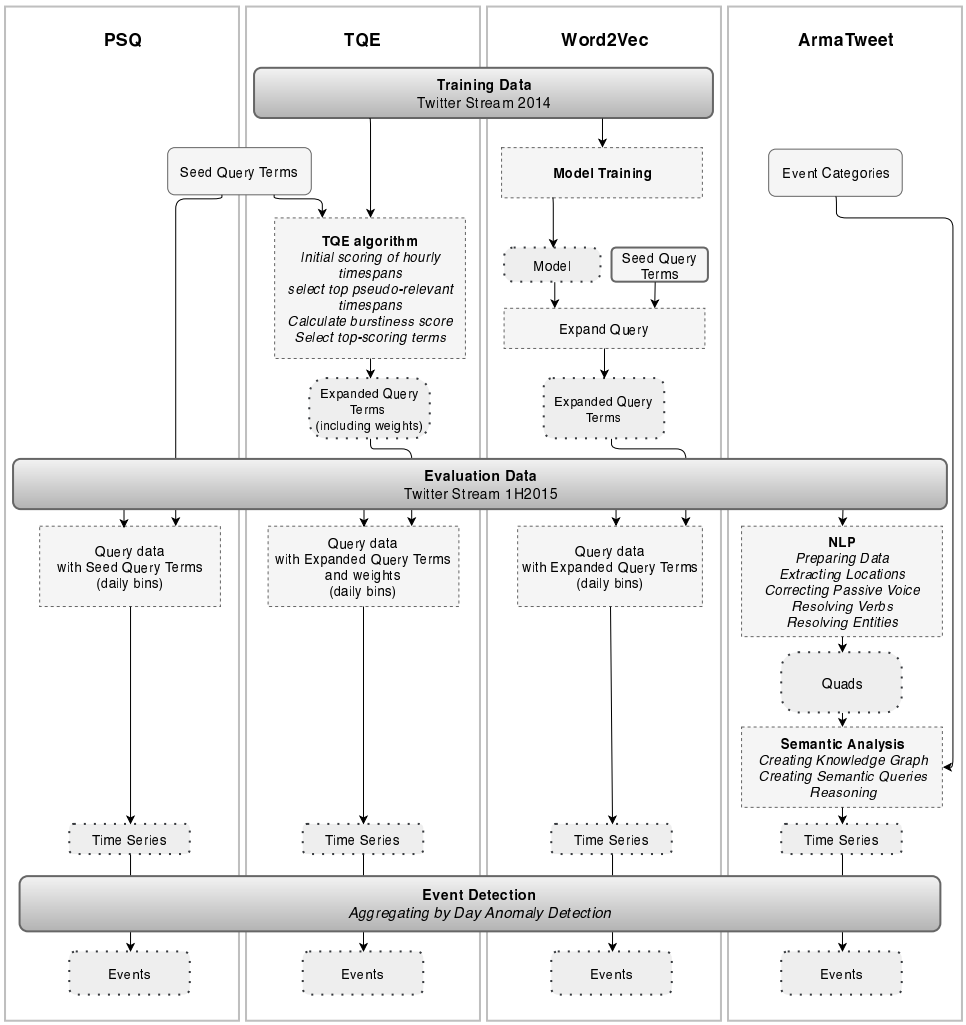


Fig. 1. Common pipeline for event detection approaches. The training (where applicable) and the evaluation datasets were collected in 2014 and the first half of 2015, respectively, using Twitter's API providing access to 1% of all tweets. Each approach generates a time series which is fed into the same event detection component to identify important events.

disjunctive seed queries as starting points for the algorithm leads to better results than using conjunctive queries, so we followed that approach in our evaluation.

4.3 The *Embedding* Approach

Word embedding techniques use statistics to attribute semantics to terms based on the context in which the terms appear. In particular, they map each term to a multidimensional space of a fixed dimension so that words sharing common contexts are placed close to each other. The Word2Vec model by Mikolov et al. [6] is based on neural networks and has gained a lot of attention lately. It is obtained by training a two-layer neural network that contains the word embeddings in its input layer after training finishes. There are several variants of this approach, and in our work we use the *skip-gram model* which we briefly summarize next.

The training set for the skip-gram variant of Word2Vec is a sequence of words, and objective of the model is to predict for each input word its context—that is, words that are likely to occur around the input word in a window of size c. The window size c is thus one of the parameters that must be selected before training. To formalize the equations of the model, let us assume that our input contains V distinct words $w_1, w_2, \ldots, w_V$—that is, we identify each distinct word w_i using an index i with $1 \leq i \leq V$. The model is trained on a sequence $w_{t_1}, w_{t_2}, \ldots, w_{t_T}$ of T words—that is, each t_j is a word index and so it satisfies $1 \leq t_j \leq V$, whereas j satisfies $1 \leq j \leq T$. Note that the input sequence can contain repeated words. The neural network consists of two layers, and each distinct word w_i is assigned an *input* vector $\mathbf{e}_i$ in the input layer and an *output* vector $\mathbf{e}'_i$ in the output layer. All of these vectors are of the same dimension, which is selected as a parameter of the model; for our model, this parameter was set to 200. After the network is trained, the input vectors $\mathbf{e}_i$ provide the desired word embeddings, whereas the output vectors $\mathbf{e}'_i$ are discarded. The network is trained to maximize the average log-probability of a word $w_{i'}$ occurring in a context of a word w_i. Formally, this corresponds to selecting $\mathbf{e}_i$ and $\mathbf{e}'_i$ such that the following expression is maximized:

$$\frac{1}{T}\sum_{j=1}^{T} \sum_{\substack{\max(1,j-c)\leq k,\\ k\leq \min(T,j+c),\\ k\neq i}} \frac{\exp({\mathbf{e}'_{t_k}}^T \cdot \mathbf{e}_{t_j})}{\sum_{i=1}^{V} \exp({\mathbf{e}'_i}^T \cdot \mathbf{e}_{t_j})}. \quad (1)$$

Several skip-gram Word2Vec models have been pre-trained on large text corpora and are freely available. However, we expected a model trained on Twitter data to perform better than the models trained on general news media or text, so we produced a new Word2Vec model using our training dataset. To empirically verify the expectation that our model performs better, we run our experiments using our model and the model trained on Google News.[5] For the "Politician dying" event category, our model detected 14 events, compared with 11 events detected using the Google News model; out of these, seven events were detected by both models. For the "Cyber-attack on a company" category, the two models detected five event and four events, respectively. We believe that these results can generalize to most of our events as our Word2Vec model is able to detect relevant hashtags which would otherwise be ignored by the Google News model. However, we have not tested all event categories because this would require us to manually annotate a very large amount of data

Our *Embedding* approach uses the Word2Vec model for query expansion as follows. For each event category, it takes the *Boolean seed query* (see Appendix A.1) and reformulates it into another Boolean query by adding additional relevant terms. Unlike *TQE*, the expansion preserves the query structure—that is, the result is a conjunction of disjunctions. Moreover, we follow an interactive query expansion (IQE) [31] approach where a user guides the first stage of query expansion: automatic query expansion can be difficult when the Boolean seed query does not reflect well the search intent, whereas IQE can still be very effective in such cases. The input query is expanded by applying the following steps to each term t of the input query.

1) Select the top ten terms $t_1, \ldots, t_n$ most similar to t from Word2Vec vector space, where vector similarity is measured using the cosine metric.
2) Show t and $t_1, \ldots, t_n$ as a suggestion to a user, and let the user decide which terms (if any) are relevant to the input query.
3) For each term t_i selected in the previous step, select further top k terms $t_i^1, \ldots, t_i^k$ most similar to t_i from the Word2Vec vector space, where the maximum possible value of k is 10.
4) Replace t in the input query with a disjunction containing t OR t_i OR t_i^1 OR ... OR t_i^k for each t_i selected in step 2.

We explain how this algorithm handled the seed query "(politician) AND (visit OR visits OR visited OR travel OR country)" for the "Politician visiting a country" category. The algorithm first processed the term "politician". In step 1, terms "politicians", "bureaucrat", "journo", "journalist", "parliamentarian", "democrat", "ideologue", "dictator", "technocrat", and "minister" were selected as most similar to "politician". In step 2, our test user identified only "politician", "bureaucrat", and "minister" as relevant. In step 3, terms "bureaucrats", "politician", "technocrat", and "govnt" were selected as most similar to "bureaucrat"; moreover, "ministers", "ministe", "minist", "minister's", "minster", "ministerial", "ministry", "secretary", "secy", and "minsiter" were selected as most similar to "Minister". Thus, in step 4, term "politician" in the original query was replaced with a disjunction of all similar terms—that is, the query was modified to "(politician OR politicians OR bureaucrat OR journo OR journalist OR parliamentarian OR democrat OR ideologue OR dictator OR technocrat OR bureaucrats OR govnt OR minister OR ministers OR ministe OR minist OR minister's OR minster OR ministerial OR ministry OR secretary OR secy OR minsiter)." The same process was repeated for all remaining terms (i.e., "visit", "visits", "visited", "travel", and "country") from the seed query. All reformulated queries are given in the supplementary material (see Appendix A.3).

The reformulated query is then answered on the evaluation dataset. This produces a single tweet time series, which is passed to the event detection technique described in Section 4.5 to extract zero or more events.

5. https://code.google.com/archive/p/word2vec/

4.4 Semantic Approach: *ArmaTweet*

ArmaTweet allows users to describe the relevant event categories *semantically* so, instead of statistical methods, it uses semantic search [32]. In particular, the system uses NLP techniques to extract a semantic representation of the tweet content and link it with information sources such as DBpedia and WordNet in a *knowledge graph* that represents the meaning of the tweets. Figure 1 shows the processing pipeline of *ArmaTweet*, which can be broadly divided into three phases: NLP, Semantic Analysis, and Event Detection. The first two phases involve considerable processing that we cannot present in detail in this paper; please refer to our earlier work for full details [4].

The *Natural Language Processing (NLP)* phase analyses the tweets' texts and extracts from it the entities and structured data. Roughly speaking, the text of each tweet is processed as follows. First, emoticons and special characters are removed as these typically cannot be matched to entities or relations. Next, using the OpenIE annotator from Stanford CoreNLP [33], the text is transformed into *text triples* consisting of a subject, a predicate, and an object; intuitively, each text triple states that a subject performs an action described by the predicate on the object. Each text triple is further processed by DBpedia Spotlight [34], which identifies parts of subject and/or object that mention a DBpedia resource. Next, to support location-aware event queries, the text triple is extended into a *quad* by identifying the location of the action described in the triple. A further step aims to improve the quality of quads by converting quads in passive voice into active voice. Since DBpedia generally does not contain information about verbs, *ArmaTweet* uses its own approach for verb resolution. In particular, it identifies all verb occurrence using a POS tagger, it lemmatizes each verb occurrence, possibly links each verb with its phrasal verb particles, and then matches each verb to a WordNet synset. As a result of this process, each tweet is converted into a set of quads consisting of a subject, predicate, object, and a location, whose components refer to DBpedia entities and WordNet verbs. Note that a quad can be *partial* and omit some of its components. Applying the NLP component to our evaluation dataset produced 14.5 M quads from 12.8 M tweets. Most quads had two components: 6.2 M quads

contained a predicate and an object, 5 M quads contained a subject and a predicate, the remaining 0.7 M quads had three components, and no quad had four components. About 0.5 M quads contained location information.

The *Semantic Analysis* phase further converts quads into a knowledge graph. Each tweet is represented in the graph as a resource obtained from the tweet's ID. This resource is made an instance of the aso:Tweet class, and information such as the time of creation or the tweet's content is associated using properties such as aso:createdAt and aso:tweetText. In addition, a tweet is associated with its quads, each of which is represented as an RDF resource. The relevant information is attached to the quads using properties aso:quadSubject, aso:quadPredicate, aso:quadObject, and aso:quadLocation, which connect quad resources with resources from DBpedia and WordNet. Integrating the quad information with DBpedia and WordNet produced a knowledge graph containing a total of 725.8 M triples, an excerpt of which is shown in Figure 2.

The knowledge graph allows users to describe the relevant events categories declaratively as conjunctive SPARQL queries. In particular, these queries can refer to DBpedia and WordNet entities, which lends *ArmaTweet* its power. For example, the query for the "Politician dying" event category refers to the DBpedia "Politician" type; thus, even if a tweet does not mention the term "politician", the semantic query can identify tweets mentioning a politician whenever the person in question is present in DBpedia and is classified as an instance of the yago:Politician110451263 class (assuming that Spotlight correctly linked the tweet to the DBpedia resource). The queries describing event categories are constructed manually, allowing users to precisely describe their information needs. In our experiments, creating the queries for all of our categories took about four person-days of an expert in semantic technologies. Depending on which quad parts are used in a query, each query was classified as a subject–predicate (SP), predicate–object (PO), subject–country (SC), predicate–country (PC), or subject–predicate–country (SPC) query. These queries were converted into datalog rules that identify a distinct tweet time series for each potentially relevant event, as well as create a summary for each event. For example, the query for the "Politician dying" event category was converted into rules that extract a time series *for each distinct politician matched*: the system extracted one time series for Edward Brooke, another time series for Mario Cuomo, and so on. This is unlike all other approaches, which produce a single time series per query. Moreover, each time series produced by *ArmaTweet* is associated with a summary describing the event; for example, a summary of a time series for the events in the "Politician dying" category consist of a reference of the DBpedia resource for the person in question and the WordNet synset describing the act of dying. Furthermore, each tweet in a time series is assigned a high or low confidence status, which is used to increase the system's precision [4]. Applying the rules for all our categories increased the size of our knowledge graph to 800 M triples, and the rules were evaluated using the state of the art reasoning system RDFox.[6]

The *Event Detection* phase extracts from each time series zero or more events as described in Section 4.5.

4.5 Event Detection

All approaches described in Sections 4.1– 4.4 generate zero or more tweet time series. The Event Detection component is the final step in all four approaches, and it identifies zero or more events from each time series. This is achieved using the Seasonal Hybrid ESD (S-H-ESD) test [35] that was tailored to Twitter data. The standard S-H-ESD algorithm takes as input a real number p between 0 and 1, a set of time points T, and a real-valued function $x : T \to \mathbb{R}$ that can be seen as a sequence of observations of some value on T, where $x(t)$ is the value observed at time $t \in T$. The algorithm identifies a subset T_a of T of time points at which the value of x is considered anomalous, while satisfying $|T_a| \leq p \cdot T$; thus, p is the maximal proportion of the time points that can be deemed anomalous. Roughly speaking, the S-H-ESD test first determines the periodicity/seasonality of the input data; it splits the data into disjoint windows each containing at least two weeks of data; and, for each window, it subtracts from x the seasonal and the median component and applies to the result the Extreme Student Derivative (ESD) test—a well-known anomaly detection technique. We used the open-source implementation of S-H-ESD from the R statistical platform.[7]

To apply the S-H-ESD test in our setting, each tweet time series is converted into a sequence of temporal observations by aggregating the tweets by day—that is, the set T corresponds to the set of all days with at least one tweet, and, for each day $t \in T$, the value of $x(t)$ is the number of all tweets occurring on day t. We also evaluated other aggregation level such as hours, but they did not improve the anomaly detection for the event categories we consider in this work. We then run the S-H-ESD test and configured the algorithm to detect only positive anomalies (i.e., cases where the number of tweets is above the expected value), which is natural for event detection. Each day on which the S-H-ESD test detected an anomaly was extracted as an event from the tweet time series, and it was associated with the tweets occurring on the day of the event.

5 Empirical Evaluation

We now discuss our experimental setup, introduce the evaluation metrics, and discuss our findings. Our main objective is to provide an insight into the strengths and weaknesses of different event detection techniques. All our systems are research prototypes whose performance can be considerably improved through further engineering, which is out of the scope of this paper. We leave a comparison of the systems' scalability and efficiency, as well as developing ways for streaming event detection for our future work.

5.1 Relevance judgments

We used a standard IR methodology for assessing the relevance of the detected events. In particular, the relevance of

6. https://www.cs.ox.ac.uk/isg/tools/RDFox/

7. https://github.com/twitter/AnomalyDetection

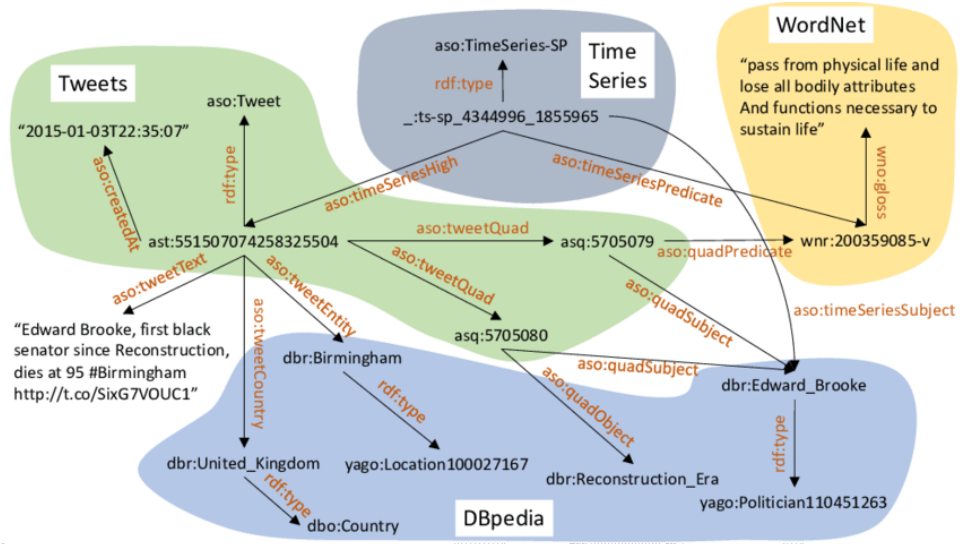


Fig. 2. A fragment of the RDF Knowledge Graph showing how DBpedia and WordNet provide us with a vocabulary and background knowledge for describing complex events. ast:551507074258325504 is an instance of the aso:tweet class. Text of the tweet is stored in aso:tweetText data property. This tweet is associated with two quads. aso:quadsubject of both quads identifies dbr:Edward_Brooke entry in DBpedia which is classified as a yago:Politician110451263 type. Predicate of the quad asq:5705079, wnr:200359085 identifies the synset 'to die' in WordNet.

the events identified by our systems was manually scored using the following scale:

1) R3: clear positive instances of the event category;
2) R2: positive instances where the entity resolution or the subject–object relationships in the event summary are incorrect (e.g., an event was linked to dbr:British_Raj instead of dbr:India);
3) R1: events with a fuzzy relationship to the category (e.g., "ISIS kills X" for the "Unrest in a country" category); and
4) R0: events with no relevance to the category (i.e., false positives).

We categorized each event extracted from our systems into the above categories by manually inspecting the tweets associated with the event and judging how well they reflect the intent of the event category. The time series produced by *PSQ*, *TQE*, and *Embedding* approach were obtained by evaluating Boolean term queries and so the number of tweets associated with each event was generally quite large. To streamline the inspection process, we ranked the tweets using the standard ranking measures (provided by SOLR[8]) and inspected only the top 50 tweets. We did not consider advanced ranking algorithms such as NDCG (Normalized Discounted Cumulative Gain) since tweet ranking is not of core importance in our approach and is used solely to reduce the amount of tweets to be inspected. No such ranking was available for *ArmaTweet*, but the system generally produced fewer tweets per event, and it also produced a summary that streamlined the event categorization.

5.2 Evaluation Metrics

We next describe the metrics that we used to present and analyze our results.

8. http://lucene.apache.org/solr/

Precision is the ratio of the number of the relevant events and the total number of retrieved events. Since events were assigned different degrees of relevance, for each event category and system we computed three precision values by considering events of relevance R3 only, of relevance R3 and R2, and of all relevance categories apart from R0.

Recall is the ratio of the number of the relevant events and the total number of events in the evaluation dataset. To calculate recall, we must know not only which events have been retrieved, but also which events have not been retrieved. The latter, however, is impossible in our case because there is no comprehensive list of all the events relevant to our event categories. Therefore, we calculated the *relative recall* [36] of our four approaches. Specifically, the relative recall of method i is given by N_i/N_{all}, where N_i is the number of distinct events of relevance R3 detected by method i, and N_{all} is total number of distinct events detected by at least one method. Events whose relevance was categorized as R2 or R1 do not constitute clear positive instances of a category in question, so we deemed it unfair to penalize one method for the mistakes of another. For example, although *Embedding* detected many events in the "Politician dying" category, most events were assigned relevance R1 as they are concerned with deaths of people who are not politicians. Moreover, the event detection step (see Section 4.5) operates at day granularity, so the same events were sometimes detected on several days. To compute N_i, we eliminated all repeated occurrences of the same event since counting duplicates would give an unfair advantage to method i. Similarly, to compute N_{all}, we counted just once each event that was detected by more than one method.

5.3 Results

In this section, we discuss the precision and relative recall of our systems on the evaluation dataset.

5.3.1 Precision

Table 1 shows the total number of detected events per category for each approach and relevance score, as well as the corresponding precision percentages. To allow an easier comparison of our approaches, the precision of our systems is summarized graphically in Figure 3. The results can vary considerably depending on the event category, so we next discuss each category separately.

Events from the "Aviation accident" category are typically unambiguous and tend to be widely discussed. *PSQ* and *Embedding* detected a fairly large number of such events covering all kinds of plane crashes. In contrast, *ArmaTweet* detected only events involving a plane belonging to an airline and thus returned a smaller number of events. To investigate this further, we manually selected the events detected by *PSQ*, *TQE*, and *Embedding* referring to an aviation incident involving a plane of a commercial airline. This revealed that accidents involving small planes are much more frequent than the ones involving commercial airliners; for example, *Embedding* detected 227 events in total, but only 70 of these involved an airline. We report the precision for this subcategory in Table 1 as well. As one can see, the "Aviation accident" event category was the only one where *ArmaTweet* was outperformed by other systems.

In the "Capital punishment" category, *ArmaTweet* clearly outperformed all systems both in terms of the total number of detected events and the number of events in the R3 category. *TQE* was the next best approach in terms of the total number of events, but only one of those belonged to the R3 category. In contrast, *Embedding* detected fewer events, but with much higher accuracy.

The "Cyber-attack on a company" category exhibited similar results. By exploiting information from DBpedia, *ArmaTweet* managed to select a large number of events, of which 15.5% were clearly relevant. All other approaches selected a much smaller number of events, and an even smaller number of relevant events. Surprisingly, *PSQ* produced much better results than the two query expansion methods. *Embedding* selected very ambiguous tweets: the most similar terms identified by word embeddings were "#hacking", "#cybersecurity", and so on, which were often used in tweets talking about seminars or workshops in this field, rather than cyber-attacks. *TQE* introduced terms such as "north", "company", "interview", "update", and so on, which were completely unrelated to the query. Thus, both *TQE* and *Embedding* lost the context of the initial seed query.

The "Politician dying" and "Politician visiting a country" categories were handled similarly to the "Cyber-attack on a company" category. The word "politician" was surprisingly close to the word "journalist" in our Word2Vec model, which prevented the *Embedding* approach from distinguishing deaths of journalists from the deaths of politicians. Similarly, *TQE* lost context during query expansion and did not select any R3-relevant events for these two categories.

The "Militia terror act" category was somewhat similar to the "Aviation accident" category in that it specifically looked for terror acts claimed by a known terrorist organization. *PSQ* again exhibited the best results in terms of the R3-rated events, while the performance of the query expansion approaches lagged behind. For example, *TQE* introduced terms specific to events of the training data set such as "2014", which were clearly not applicable to the evaluation dataset that covered only the year 2015.

To summarize, *ArmaTweet* performed much better than the other approaches as the query got more complex and specific. The *Embedding* approach offered good precision, but not as good recall. *TQE* did not produce complex Boolean queries and thus tended to perform poorly on complex event definitions. Our baseline, *PSQ*, exhibited surprisingly good precision, particularly in cases where *TQE* and *Embedding* lost context of the initial terms. This, we believe, is due to the presence of few but meaningful query terms that can generate meaningful time series. Although the precision of *PSQ* was better than *TQE* and *Embedding*, it detected much fewer events than *ArmaTweet*.

5.3.2 Relative recall

Table 2 presents the relative recall of our systems, which we computed as explained in Section 5.2: we considered only events that were assigned relevance R3, and we eliminated duplicate events. For the "Aviation accidents" category, the table also shows the recall for the subcategory involving planes of a commercial airline. The "Total events" column shows the total number of unique events identified using all four approaches. Figure 4 summarizes these results graphically, and Figure 5 breaks down the results by showing how many events were detected by a combination of systems.

For all event categories apart from the general "Aviation accidents" one, *ArmaTweet* outperforms the other three systems. *PSQ* is surprisingly the second-best approach. The performance of query expansion approaches lags behind the other two, mainly because they generally selected few clearly relevant events (i.e., with relevance R3).

5.4 Discussion

Our event categories differ considerably in terms of event type popularity. For example, the "Aviation accident" category is very popular, whereas "Politician Drying" is not: *Embedding* returned about 31500 tweets for the former, and only about 3100 tweets for the latter. The categories also differ widely in semantic complexity. For example, a tweet about an aviation accident is likely to contain a categorical word "plane"; in contrast, a tweet reporting on a politician's visit is likely to contain just a person's name, without explicitly stating that the person in question is a politician.

As our results show, complex events are quite challenging for query expansion techniques. The semantic approach generally performs much better in such cases because it allows users to describe the event more precisely in terms of restrictions on the subjects, predicate, object, and/or location of an action. For example, in the "Aviation accident" category, *ArmaTweet* could identify crashes involving a plane of an airline, whereas information retrieval approaches also detected crashes involving small planes; while different use cases may prefer one or the other method, this clearly illustrates the kind of precision that can be attained using semantic search. Similarly, detecting terrorist attacks performed by a known terror group was much more easily described using a semantic approach. The main obstacle to the performance of *TQE* is the loss of the query structure (i.e., query expansion produces a disjunction of terms), which seems to be critical for detection of complex events.

TABLE 1
Precision of event detection with different relevance scores

Event category	Method	Total events	R3 (%)	R3+R2 (%)	R3+R2+R1 (%)
AviationAccident	*ArmaTweet*	84	44 (52.4)	51 (60.7)	64 (76.2)
	PSQ	203	173 (85.2)	186 (91.6)	202 (99.5)
	TQE	24	22 (91.7)	24 (100.0)	24 (100.0)
	Embedding	227	203 (89.4)	214 (94.3)	227 (100.0)
AviationAccident (only airlines)	*PSQ*	76	62 (81.6)	64 (84.2)	66 (86.8)
	TQE	12	11 (91.7)	12 (100.0)	12 (100.0)
	Embedding	70	68 (97.1)	69 (98.6)	70 (100.0)
CapitalPunishment	*ArmaTweet*	153	47 (30.7)	67 (43.8)	92 (60.1)
	PSQ	34	6 (17.6)	6 (17.6)	28 (82.4)
	TQE	48	1 (2.1)	1 (2.1)	48 (100.0)
	Embedding	36	31 (86.1)	31 (86.1)	36 (100.0)
CyberAttackCompany	*ArmaTweet*	129	20 (15.5)	42 (32.6)	58 (45.0)
	PSQ	18	13 (72.2)	17 (94.4)	17 (94.4)
	TQE	1	0 (0.0)	0 (0.0)	0 (0.0)
	Embedding	10	6 (60.0)	6 (60.0)	10 (100.0)
MilitiaTerrorAct	*ArmaTweet*	220	92 (41.8)	125 (56.8)	141 (64.1)
	PSQ	55	42 (76.4)	48 (87.3)	55 (100.0)
	TQE	25	11 (44.0)	15 (60.0)	17 (68.0)
	Embedding	61	34 (55.7)	45 (73.8)	61 (100.0)
PoliticianDying	*ArmaTweet*	111	76 (68.5)	80 (72.1)	85 (76.6)
	PSQ	53	15 (28.3)	21 (39.6)	21 (39.6)
	TQE	2	0 (0.0)	0 (0.0)	0 (0.0)
	Embedding	54	18 (33.3)	21 (38.9)	54 (100.0)
PoliticianVisits	*ArmaTweet*	44	29 (65.9)	36 (81.8)	44 (100.0)
	PSQ	88	2 (2.3)	3 (3.4)	3 (3.4)
	TQE	3	0 (0.0)	0 (0.0)	0 (0.0)
	Embedding	2	2 (100.0)	2 (100.0)	2 (100.0)

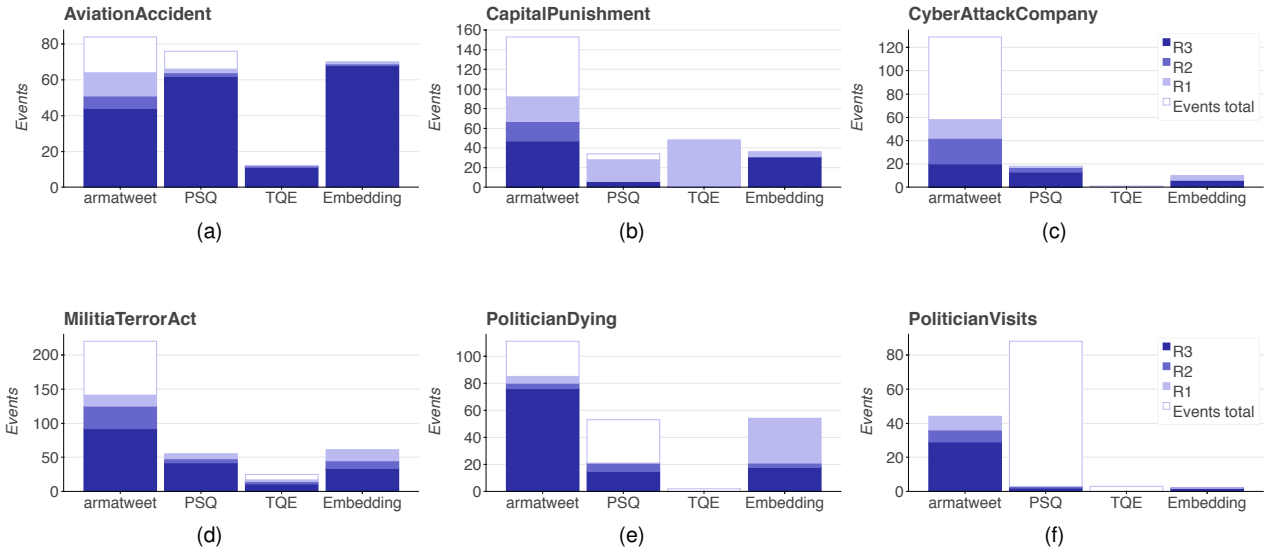


Fig. 3. Precision of *ArmaTweet*, *PSQ*, *TQE*, and *Embedding* from left to right per event category: (a) Aviation accident (airline-related), (b) Capital punishment, (c) Cyber-attack on a company, (d) Militia terror acts, (e) Politician dying (f), Politician visiting a country.

TABLE 2
Relative recall

Event category	Total events	*ArmaTweet* (%)	*PSQ* (%)	*TQE* (%)	*Embedding* %
AviationAccident	158	24 (15.2)	86 (54.4)	16 (10.1)	96 (**60.8**)
AviationAccident (airline-related)	40	24 (**60.0**)	15 (37.5)	6 (15.0)	18 (45.0)
CapitalPunishment	42	29 (**69.0**)	6 (14.3)	1 (2.4)	20 (47.6)
CyberAttackCompany	18	11 (**61.1**)	9 (50.0)	0 (0.0)	2 (11.1)
MilitiaTerrorAct	116	75 (**64.7**)	30 (25.9)	6 (5.2)	17 (14.7)
PoliticianDying	94	76 (**80.9**)	14 (14.9)	0 (0.0)	12 (12.8)
PoliticianVisits	31	29 (**93.5**)	2 (6.5)	0 (0.0)	2 (6.5)

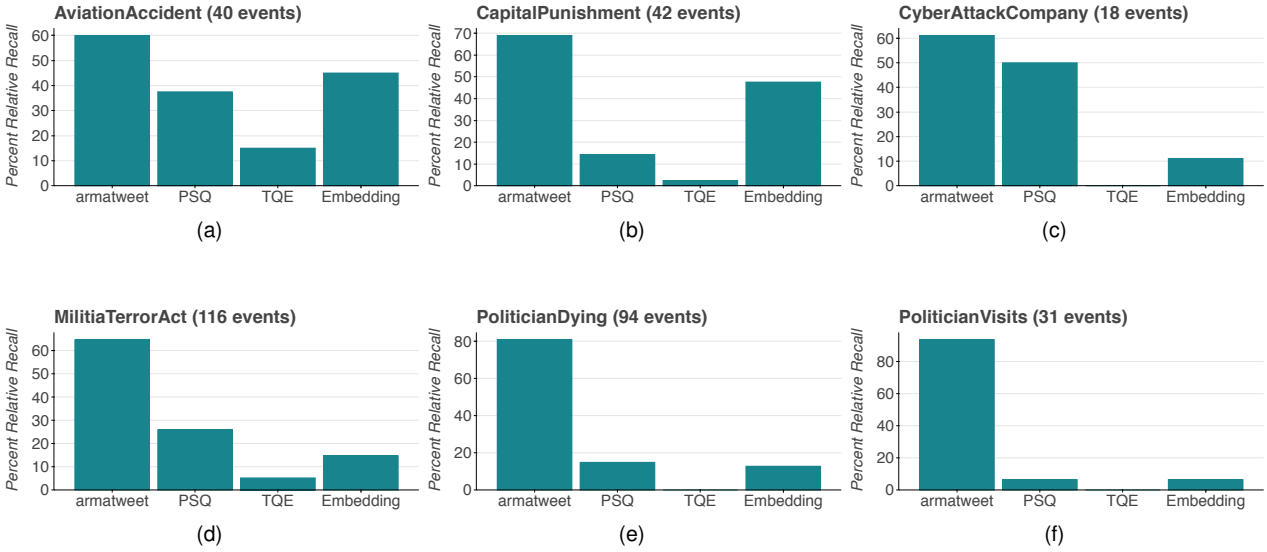


Fig. 4. Relative recall of *ArmaTweet*, *PSQ*, *TQE*, and *Embedding* from left to right per event category: (a) Aviation accident (airline-related), (b) Capital punishment, (c) Cyber-attack on a company, (d) Militia terror acts, (e) Politician dying ,(f) Politician visiting a country.

While all approaches require users to describe the event category using a query, one can argue that constructing a semantic query necessary for *ArmaTweet* is much more complex and time-consuming than constructing a Boolean term query that is used by the other three approaches. We believe, however, that the cost of query construction can be reduced by producing a suitable user interface allowing users to explore the knowledge graph during query construction.

The breakdown of the relative recall by systems in Figure 5 shows that each method provides an advantage in each event category. For example, a total of 117 events with relevance R3 were detected in the "Militia terror attack" category (see Table 2). Although *ArmaTweet* detected 75 of these events, additional 22 and 14 events were detected solely by *PSQ* and *Embedding*, respectively. The overlap among the four systems is limited, but not insignificant; for example, 27.4% of the events detected by *Embedding* were detected by at least one other method, and 40% of the events were detected by *ArmaTweet*. The overlap was lowest for *TQE*, with only 18.5% of its events detected by at least one other method. Thus, an approach combining elements of both semantic and keyword search might be needed in use cases where high recall is important and missing even a single event can have significant societal consequences.

Interestingly, combining the results of the four systems does not negatively impact the precision: cumulative precision is always more than the precision of at least one of the approaches (see Appendix A.4). Specifically, the "Politician visiting a country" and "Cyber-attack on a company" event categories exhibit the lowest cumulative precision of 35.8% and 53.9%, respectively. For all other event categories, cumulative precision always exceeds more than 70%.

6 Conclusion

In this paper, we presented an extensive evaluation of four different techniques for event detection on Twitter: the *PSQ* baseline approach, the *TQE* query expansion technique exploiting temporal co-occurrence of words, the *Embedding* approach that aims to identify word contexts, and *ArmaTweet* using a semantic search. While *ArmaTweet* generally exhibited the best performance, a combination of all of these techniques might be most suitable as it delivers high recall that is typically needed in security-related applications. In future work, we plan to develop and evaluate such a combined system. Moreover, to simplify the development of event queries in *ArmaTweet*, we plan to develop a subsystem that can visualize the knowledge graph and provide a simple ad hoc querying interface to the end user.

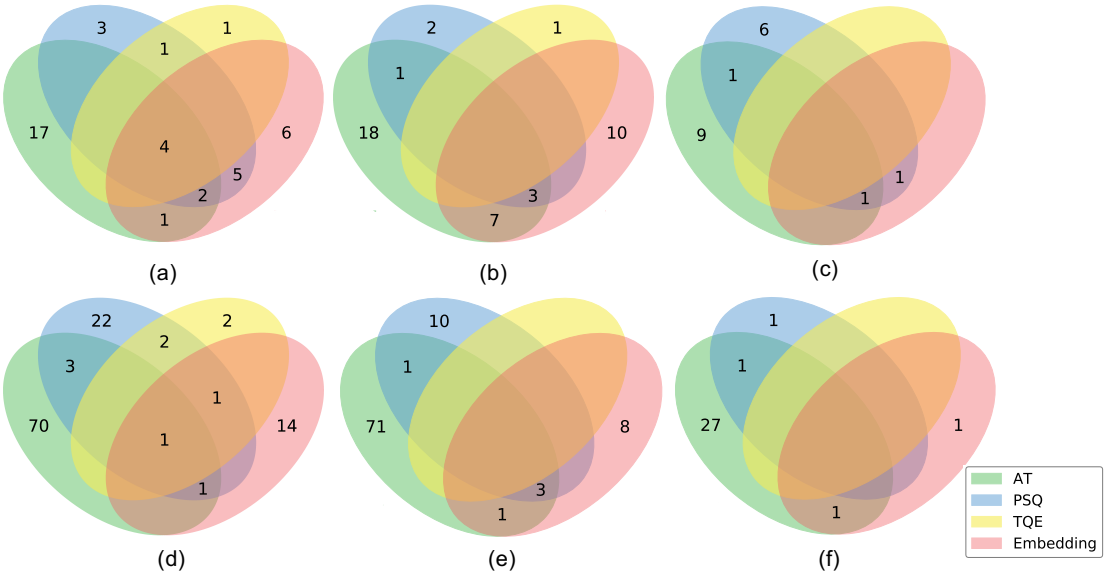


Fig. 5. Breakdown of the recall per approach and event category: (a) Aviation accident (airline-related), (b) Capital punishment, (c) Cyber-attack on a company, (d) Militia terror act, (e) Politician dying, (f) Politician visiting a country.

Acknowledgments

This work was supported (in part) by the Swiss National Science Foundation under grant number 407540_167320, and the EPSRC fellowship AnaLOG (EP/P025943/1).

References

[1] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Comput. Intell.*, vol. 31, no. 1, pp. 132–164, Feb. 2015. [Online]. Available: http://dx.doi.org/10.1111/coin.12017

[2] H. Becker, F. Chen, D. Iter, M. Naaman, and L. Gravano, "Automatic identification and presentation of twitter content for planned events," in *ICWSM*, 2011.

[3] K. Massoudi, M. Tsagkias, M. De Rijke, and W. Weerkamp, "Incorporating query expansion and quality indicators in searching microblog posts," in *European Conference on Information Retrieval*. Springer, 2011, pp. 362–367.

[4] A. Tonon, P. Cudré-Mauroux, A. Blarer, V. Lenders, and B. Motik, "ArmaTweet: Detecting Events by Semantic Tweet Analysis," in *ESWC 2017*, ser. LNCS, E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler, and O. Hartig, Eds., vol. 10250, Portorož, Slovenia, May 28–June 1 2017, pp. 138–153.

[5] D. Metzler, C. Cai, and E. Hovy, "Structured event retrieval over microblog archives," in *NAACL*, ser. HLT '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 646–655. [Online]. Available: http://dl.acm.org/citation.cfm?id=2382029.2382138

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[7] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on twitter," *Icwsm*, vol. 11, no. 2011, pp. 438–441, 2011.

[8] R. Long, H. Wang, Y. Chen, O. Jin, and Y. Yu, "Towards effective event detection, tracking and summarization on microblog data," in *International Conference on Web-Age Information Management*. Springer, 2011, pp. 652–663.

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

[10] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling, "Twitterstand: news in tweets," in *Proceedings of the 17th acm sigspatial international conference on advances in geographic information systems*. ACM, 2009, pp. 42–51.

[11] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in twitter," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, vol. 3. IEEE, 2010, pp. 120–123.

[12] S. Petrović, M. Osborne, and V. Lavrenko, "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*. Association for Computational Linguistics, 2010, pp. 181–189.

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

[14] E. Benson, A. Haghighi, and R. Barzilay, "Event discovery in social media feeds," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 389–398.

[15] A.-M. Popescu and M. Pennacchiotti, "Detecting controversial events from twitter," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 1873–1876.

[16] R. Lee and K. Sumiya, "Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection," in *Proceedings of the 2nd ACM SIGSPATIAL international workshop on location based social networks*. ACM, 2010, pp. 1–10.

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

[18] R. Li, K. H. Lei, R. Khadiwala, and K. C.-C. Chang, "Tedas: A twitter-based event detection and analysis system," in *ICDE)*. IEEE, 2012, pp. 1273–1276.

[19] A. Signorini, A. M. Segre, and P. M. Polgreen, "The use of twitter to track levels of disease activity and public concern in the us during the influenza a h1n1 pandemic," *PloS one*, vol. 6, no. 5, p. e19467, 2011.

[20] N. Ramakrishnan, P. Butler, S. Muthiah, N. Self, R. Khandpur, P. Saraf, W. Wang, J. Cadena, A. Vullikanti, G. Korkmaz *et al.*, "'beating the news' with embers: forecasting civil unrest using open source indicators," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1799–1808.

[21] H. Becker, F. Chen, D. Iter, M. Naaman, and L. Gravano, "Automatic identification and presentation of twitter content for planned events," in *ICWSM*, 2011.

[22] K. Massoudi, M. Tsagkias, M. De Rijke, and W. Weerkamp, "Incorporating query expansion and quality indicators in searching microblog posts," in *European Conference on Information Retrieval*. Springer, 2011, pp. 362–367.

[23] J. Ooi, X. Ma, H. Qin, and S. C. Liew, "A survey of query expansion, query suggestion and query refinement techniques," in *ICSECS, 2015 4th International Conference*. IEEE, 2015, pp. 112–117.

[24] H. J. Peat and P. Willett, "The limitations of term co-occurrence data for query expansion in document retrieval systems," *Journal of the american society for information science*, vol. 42, no. 5, p. 378, 1991.

[25] K. S. Jones, *Readings in information retrieval*. Morgan Kaufmann, 1997.

[26] C. Carpineto and G. Romano, "A survey of automatic query expansion in information retrieval," *ACM Computing Surveys (CSUR)*, vol. 44, no. 1, p. 1, 2012.

[27] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: Bm25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, Apr. 2009.

[28] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais, "The vocabulary problem in human-system communication," *Communications of the ACM*, vol. 30, no. 11, pp. 964–971, 1987.

[29] M. Efron, J. Lin, J. He, and A. De Vries, "Temporal feedback for tweet search with non-parametric density estimation," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014, pp. 33–42.

[30] J. W. Tukey, *Exploratory data analysis*. Reading, MA, USA, 1977, vol. 2.

[31] D. Harman, "Towards interactive query expansion," in *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1988, pp. 321–331.

[32] P. Cudre-Mauroux, "Semantic search," *Encyclopedia of Big Data Technologies*, 2018.

[33] G. Angeli, M. J. J. Premkumar, and C. D. Manning, "Leveraging linguistic structure for open domain information extraction," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 344–354.

[34] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes, "Improving efficiency and accuracy in multilingual entity extraction," in *Proceedings of the 9th International Conference on Semantic Systems*. ACM, 2013, pp. 121–124.

[35] O. Vallis, J. Hochenbaum, and A. Kejariwal, "A novel technique for long-term anomaly detection in the cloud," in *HotCloud*, 2014.

[36] S. J. Clarke and P. Willett, "Estimating the recall performance of web search engines," in *Aslib Proceedings*, vol. 49, no. 7. MCB UP Ltd, 1997, pp. 184–189.

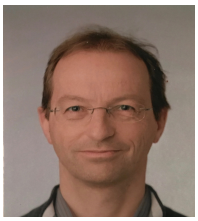
Akansha Bhardwaj is a PhD candidate at eXascale Infolab, University of Fribourg in Switzerland supported by the Swiss National Foundation (SNF). Her current research focuses on detection of important events in different kinds of media, and she works with Prof. Dr. Philippe Cudre-Mauroux. Akansha has a M.Sc in Computer Science (*Intelligent Systems*) from Technical University, Kaiserslautern in Germany.

Albert Blarer is scientific project manager in the department of C4I at armasuisse Science and Technology. He originally studied biology at the University of Basel, Switzerland, and received his PhD at the same University in 1999, working on bio-mathematical topics of dynamical systems. Albert Blarer's main focus at work is on data analysis. He is currently involved in different Deep Learning projects and focuses on the technical and scientific expertise of Machine Learning systems which might have implications for the Swiss Armed Forces and related organizations. He is also teaching Data Science Analytics as part of Certified Advanced Studies (CAS) at the University of Lucerne.

Philippe Cudré-Mauroux is a Full Professor and the director of the eXascale Infolab at the University of Fribourg in Switzerland. Previously, he was a postdoctoral associate working in the Database Systems group at MIT. He received his PhD from the Swiss Federal Institute of Technology EPFL, where he won both the Doctorate Award and the EPFL Press Mention in 2007. Before joining the University of Fribourg, he worked on distributed information and media management for HP, IBM Watson Research (NY), and Microsoft Research Asia. He was Program Chair of the International Semantic Web Conference in 2012 and General Chair of the International Symposium on DataDriven Process Discovery and Analysis in 2012 and 2013. He recently won a Google Faculty Award, the Verisign Internet Infrastructures Award, as well as a € 2 M grant from the European Research Council. His research interests are in next-generation Big Data management infrastructures for non-relational data. Webpage: http://exascale.info/phil

Vincent Lenders is a Director of research for *Cyberspace and Information* at armasuisse Science and Technology. He is the co-founder and chairman of the executive boards of the OpenSky Network and Electrosense associations. Vincent Lenders earned his PhD degree (2006) and Msc (2001) in electrical engineering and information technology both at ETH Zurich. He was postdoctoral research faculty at Princeton University. He is the author of more than 100 publications that appeared in peer-reviewed international conferences and journals. He has received best paper awards at IEEE WONS 2012, DFRWS EU 2015, ACM CPSS 2015, and DASC 2015, and the Security Award in 2011 from the Swiss Federal Department of Defense. His main research interests are in the fields of cyber defense, data science, and digital avionics.

Boris Motik is a Professor of Computer Science at the University of Oxford. His research interests include knowledge representation and reasoning, databases, ontologies and the Semantic Web, and their applications. He is the lead editor of the OWL 2 standard. He received the 2007 Cor Baayen Award, was one of "AI's 10 to Watch" in IEEE Intelligent Systems in 2007, received an EPSRC Early Career Fellowship in 2012, and won the 2013 Roger Needham Award.



Axel Tanner is working in the Security & Privacy group of the IBM Zurich Research Lab. Since 2002, he has conducted research in areas of big data, computer security, services and systems management. His current interests include social media analytics using big data tools, privacy and desensitization, as well as computer security and risk assessment in socio-technical systems. He joined IBM Research in 1993, first as part of the IT group of the lab, later serving as the manager of the IT department until 2002. He received a PhD in physics from the University of Zürich, Switzerland, in 1993. Contact him at axs@zurich.ibm.com.

Alberto Tonon was formerly a member of the eXascale Infolab, led by prof. Philippe Cudr-Mauroux, where he did research on how to build and exploit Knowledge Graphs to help users satisfying their information needs, performing transactional tasks, and providing them with serendipitous information about the content they are consuming. Alberto obtained his Ph.D. in June 2017 and since then is working as a data scientist at Swisscom AG, where he broadened his knowledge by working on topics related to Natural Language Processing and Machine Learning.

APPENDIX A

A.1 Query Terms used in *PSQ*

Table 3 shows the seed queries for our event categories.

A.2 Expanded Query Terms in *TQE*

Table 4 shows the expanded queries that *TQE* produced from the seed queries in Table 3. All queries are disjunctions of the terms shown, the numbers in parentheses indicate the relevance of each term, and #PRT is the number of pseudo-relevant timestamps.

A.3 Expanded Query Terms in *Embedding*

Table 5 shows the expanded queries that *Embedding* produced from the seed queries in Table 3.

A.4 Results of Combined system

Table 6 shows the precision obtained by combining all four approaches into one system.

TABLE 3
Query terms used in *PSQ*

Event Type	Boolean Query
AviationAccident	(plane OR aircraft OR airline) AND (crash OR crashed)
CyberAttackCompany	company AND (hack OR hacked OR hacking OR attack OR cyber)
CapitalPunishment	(kill OR killed OR punish OR punished OR punishment OR execute OR executed)
MilitiaTerrorAct	(bomb OR bombs OR kill OR killed OR die OR dies OR died OR attack OR assault OR destruction OR torment OR hijack OR hijacking OR kidnap OR kidnapping OR abduction OR terror)
PoliticianDying	politician AND (die OR dies OR died OR rip)
PoliticianVisits	politician AND (visit OR visits OR visited OR travel OR country)

TABLE 4
Expanded query terms derived by *TQE* from the seed query

Event Type	#PRT	Expanded Query Terms with weights			
Aviation Accident	735	crash (1.78)	plane (1.739)	flight (1.466)	allah (1.333)
		jet (1.313)	#news (1.26)	victims (1.256)	children (1.256)
		international (1.254)	muslim (1.231)	air (1.227)	visit (1.225)
		crashed (1.224)	pray (1.224)	indian (1.223)	
Capital Punishment	400	killed (2.21)	children (1.474)	innocent (1.444)	murdered (1.438)
		seconds (1.428)	direction (1.428)	kill (1.406)	israel (1.404)
		killing (1.389)	police (1.353)	war (1.289)	black (1.288)
		humanity (1.286)	2014 (1.284)	death (1.279)	
Cyber Attack Company	651	north (1.205)	company (1.203)	interview (1.199)	update (1.195)
		online (1.193)	gift (1.192)	giveaway (1.187)	prices (1.181)
		stock (1.18)	price (1.179)	2014 (1.177)	beauty (1.174)
		list (1.173)	present (1.173)	free (1.173)	
Militia Terror Act	318	killed (1.778)	died (1.639)	rip (1.494)	children (1.388)
		2014 (1.351)	die (1.3)	year (1.299)	kill (1.298)
		innocent (1.288)	dead (1.286)	years (1.282)	peace (1.279)
		lives (1.277)	death (1.27)	killing (1.263)	
Politician Dying	157	politician (1.648)	indian (1.258)	culture (1.255)	london (1.249)
		market (1.245)	minister (1.236)	latest (1.232)	news (1.231)
		media (1.231)	event (1.229)	popular (1.229)	data (1.229)
		staff (1.22)	fab (1.219)	prices (1.217)	
Politician Visits	412	politician (1.436)	#nowplaying (1.179)	tea (1.173)	daily (1.17)
		stories (1.167)	pic (1.167)	photo (1.167)	busy (1.163)
		coins (1.162)	power (1.16)	fear (1.16)	free (1.158)
		english (1.158)	country (1.157)	news (1.157)	

TABLE 5
Query terms obtained using *Embedding* approach

Event Type	Boolean Query
AviationAccident	(airline OR airlines OR airline's OR airways OR #airlines OR jetstar OR #airline OR jetblue OR aircraft OR aircrafts OR boeing OR #aircraft OR unmanned OR aviation OR f-16 OR aircraft's OR flight OR flight's OR plane OR flig OR sfo OR flts OR yyz OR plane OR flight OR planes OR airliner OR airplane OR helicopter) AND (crash OR crashes OR collision OR pileup OR accident OR crashed OR accid OR crashed OR crashing OR collided OR collapsed OR exploded OR accidents OR acciden OR hashtags:accident OR incident)
CyberAttackCompany	company AND (cyber OR hashtags:cyber OR hashtags:ccureit OR hashtags:infosec OR hashtags:cybersecurity OR hashtags:cybercrime OR hashtags:cyberwarfare OR hashtags:securityaffairs OR hashtags:hacksurfer OR hashtags:hacking OR hack OR hacking OR hashtags:hack OR hacker OR hackers OR hacks OR ifunbox OR hashtags:kaminfo OR hashtags:hacksurfer OR hacker's)
CapitalPunishment	(execution OR executions OR executed OR executio OR prosecution OR hashtags:deathpenalty OR executing OR execute OR hashtags:execution OR prosecute OR prosecuted OR prosecuting OR punish OR convict OR investigate OR execution OR executing OR prosecute OR punish OR convict OR investigate OR extradite)
MilitiaTerrorAct	(terror OR terrorism OR terrorist OR islamist OR hashtags:terror OR terrorists OR jihadist OR terroris OR isis OR militant OR extremist OR qaeda OR jihadi OR islamists OR extremists OR qaida OR extremism OR corruption OR terroris OR radicalization OR violence OR extremists OR isil)
PoliticianDying	(politician OR politicians OR bureaucrat OR journo OR journalist OR parliamentarian OR democrat OR ideologue OR dictator OR technocrat OR bureaucrats OR govnt OR minister OR ministers OR ministe OR minist OR minister's OR minster OR ministerial OR ministry OR secretary OR secy OR minsiter) AND (rip OR r.i.p. OR r.i.p OR hashtags:rip OR hashtags:restinpeace OR hashtags:r.i.p OR died OR death OR deat OR hashtags:death OR demise OR deaths OR dealth OR murder)
PoliticianVisits	(politician OR politicians OR bureaucrat OR journo OR journalist OR parliamentarian OR hashtags:politician OR democrat OR ideologue OR dictator OR technocrat OR hashtags:bureaucrat OR govnt OR minister OR ministers OR ministe OR minist OR minister's OR minster OR ministerial OR ministry OR secretary OR secy OR minsiter) AND (visit OR visiting OR vist OR vsit OR join OR check OR itineraries OR visist OR register OR h34-official OR hashtags:visit OR visited OR visting OR volunteering OR welcoming OR visitng OR relocated OR meeting OR attending OR traveling OR travel OR travelers OR travelling OR explore OR hashtags:travel OR traveller OR traveler OR backpacking OR travellers)

TABLE 6
Combined precision of all the systems

Event Type	Combined precision
AviationAccident	96.09
AviationAccident (only airlines)	87.59
CapitalPunishment	75.26
CyberAttackCompany	53.82
MilitiaTerrorAct	75.90
PoliticianDying	72.73
PoliticianVisits	35.76