Towards the use of RDF Stream Processing for Event Enrichment from Social Media Streams

Robin Keskisärkkä and Eva Blomqvist

Linköping University, Linköping, Sweden,
firstname.lastname@liu.se

Abstract. Being able to quickly respond to incidents in an appropriate way is crucial to emergency responders, hence, good situation awareness is essential. Today, situation awareness can be developed through information from traditional information sources, coupled with domain specific knowledge, but it can also be complemented by information from other sources, such as Social Media streams. However, to handle large-scale online data streams new means for data processing are needed. In combination with Semantic Web technologies, stream processing can be used to perform Complex Event Processing over the data streams, and to enrich the event information and make it more informative for emergency responders, as well as for post-emergency analysis and investigation. In this paper we demonstrate the feasibility of using existing RDF Stream Processing (RSP) for event enrichment from Social Media streams. We exemplify this through a simple use case and discuss the potentials and limits of this approach.

Keywords: Complex Event Processing, RDF Stream Processing, event detection, social media streams

1 Introduction

The amount of data available as online streams is increasing rapidly, ranging from sensor, image, and video streams to RSS, live news, and social media. Being able to consume and analyze this data in real-time has an enormous potential in many domains, such as business analysis, disaster response, as well as emergency response and investigation.

However, the speed at which new data is generated often exceeds the capacity of traditional analysis methods, which are designed to process data at rest. In domains where data streams are rapid and data needs to be processed as it becomes available, new paradigms and algorithms need to be applied to provide sufficient scalability. For this purpose, stream processing and stream reasoning, and in particular the combination with Semantic Web technologies, so called RDF stream processing (RSP), has emerged as an active research area. RSP systems can form the basis for Complex Event Processing (CEP) over heterogeneous data streams, by leveraging the Linked Data principles, which in turn can support automated reasoning.

In order to perform CEP one needs to have sufficient information about the event in order to formulate rules to trigger, for example, transformations and abstractions about the event. An abstraction may be a classification of an event into a user-relevant category, or an aggregation of several event objects that together contribute to the overall description and understanding of the event. In this paper we explore the potential and limits of existing RSP technologies in event object enrichment. As an illustrative use case, and experimental setup, we use reports of traffic incidents, which are enriched in real-time based on social media streams in order to support situation awareness. Section 2 explains the terminology used in the paper, and Section 3 contains a discussion of related work. We then proceed to explain the overall idea behind using RSP for event enrichment in Section 4, before detailing this through a discussion of our experimental setup in Section 5. Finally, we discuss potential and limits of the approach, as well as summarize our conclusions in Section 7.

2 Preliminaries

An *event* is defined as "anything that happens, or is contemplated as happening" [7], and a *complex event* is an event that "summarizes, represents, or denotes a set of other events". Hence, *Complex Event Processing* is the process by which such an aggregation and abstraction is performed. *Event objects* are used to represent real-world events in networks of interconnected rule processors, so called Event Processing Agents (EPAs). EPAs can be used to, for example, filter, aggregate, transform, enrich, and detect *event patterns* from data streams, where a *stream* is defined as continuous (possibly unbounded) flow of time-stamped data.

3 Related Work

Successful detection and enrichment of large-scale events from analysis of social media has been reported in several domains, for example, earthquakes [10] and assessment of the spread of influenza [5]).

One study [11] investigated the possibility of detecting traffic incidents in real-time from Twitter messages using a machine learning approach. Each tweet was preprocessed to remove stopwords, correct spelling errors, apply part-of-speech tagging, and to replace temporal and spatial features based on text mentions. In a second step the features necessary for classification were extracted, before finally reaching the classification step. On average 10 tweets were reported for every identified car accident. The precision was reported at 89%, but no figures for recall were reported. Although this approach seems quite successful, solely using machine learning suffers from the inherent cold-start problem, that is, the need to train the models on the particular kind of incidents that one would like to detect, which means that typically large (and often pre-classified) datasets need to be available beforehand. This is in contrast to the rule-based approaches, where only the event patterns are needed as a starting point, and

new and edited event patterns can be entered at any time during the system runtime. Machine learning approaches are problematic in some settings since they are not transparent to human users.

The idea of detecting events from social media streams using RSP technologies has been explored in a few recent papers. In one study, city-scale events were analyzed based on Twitter streams using the Streaming Linked Data framework (SLD) [2]. SLD uses the C-SPARQL engine [3] and leverages external stream decorators for some analysis tasks. By using an event calendar as a set of ground truths, and using only those statuses that were both geotagged and contained at least one of a set of predefined keywords, the authors were able to detect the majority of the calendar events from the stream of tweets. However, since only a few percent of all tweets are geotagged this approach relies on a large set of tweets being available for each event, which means the approach is less suited for analyzing small-scale events.

Another study focused on geospatially enriching user generated text content (UGTC), and went on to provide hierarchical visualizations of these [4]. The authors tagged each UGTC with a relevance tag, calculated as a product of domain specific topical tag weights and location factors. This analysis was performed on data stored in a triple store, and it is not clear if the enrichment of data was added to the triple store directly, or if some of enrichment was performed when actually querying the data. As such, it is not possible to say which aspects of the approach would scale for real-time processing.

In general, large-scale events generate a lot of attention in social media, while small-scale events, such as traffic incidents, are short-lived and generate very little public attention. This often makes these incidents quite difficult to detect from social media streams, and we instead propose that creation of the event objects can be triggered by official communication channels, and social media streams can instead be used to enrich these event objects.

4 The RSP approach to Event Enrichment

A number of RSP engines have been developed in recent years, for example, CQELS [6], C-SPARQL [3], INSTANS [8], and EP-SPARQL [1]. These systems enable continuous querying over streaming, and static, Linked Data in ways which are both efficient and flexible. By republishing the query results as streams these engines can be used to create networks of interconnected rule processors, where each query can be viewed as a separate EPA.

The query languages alone are, however, not expressive enough for full-fledged CEP and external processing of data is often necessary. We propose that each external processing step can be viewed as a task-specific EPA, which republishes results as streams. Event enrichment can therefore take place both at the level of individual EPAs, as well as within a network of distributed EPAs.

4.1 Scenario

In the initial phases of emergency emergency response and investigation, it is of essence to gather as much relevant information as possible about an incident. In the case of traffic incidents, this may be information from people witnessing the incident and concern what actually took place, who or what caused it, exactly when it happened etc. It is rarely the case that emergency responders have the complete picture from the start, and even during post-emergency investigations and evaluations a clear picture of all the details of an incident is often not available. However, incidents, or the effects of incidents, are often witnessed (at least partially) by many people. The wide majority of these witnesses will not contact the emergency services or police themselves, but some may still share their experience of the incident on social media. An earlier study showed that the average number of tweets regarding a car accidents was around 10 [11], but for other types of traffic incidents the number is likely to be significantly lower.

The same is most likely true for other types of "minor" events, such as fights in public places, and minor crimes where bystanders or victims may not immediately feel compelled to report it to the police or emergency services. Instead such incidents may become known later, for example. when a police report is filed days later, someone makes an insurance claim, or visits a hospital for their injuries. At that time it is hard to get information, since witnesses have long since left the incident site.

The vision of CEP in this context is to be able to generate rich representations of the real-world events, containing aggregated and relevant information about the event from various sources. Many of these data sources may be high-volume streaming data sources, such as news feeds and social media streams. The goal is that the constructed event representations will be able to provide users with a more complete picture of the incident, and contribute to better situation awareness.

The scenario where our system is intended to operate, hence, starts with a report of an event through a known (trusted) channel, fr example, an emergency call or police report. While the incident is dealt with in the real-world by emergency responders, the system should work online to collect more information about the incident in real-time, and add that to the representation of the event to complement and enrich it. The enriched event object resulting from this process is intended to provide increased situation awareness, but also the opportunity for better post-event investigation and evaluation.

4.2 Solution Framework

The intended use of RSP technologies for CEP in in the aforementioned use case is illustrated in Figure 1. Given a set of input streams, each containing various event objects, the system produces an output stream consisting of user-relevant event objects, based on filtering, aggregation, and reasoning over event objects in the input stream. The complex events that are of interest to a user are defined by means of a set of event patterns, which are patterns that when detected in the

input streams will trigger the creation of complex event in the output stream. The system also has at its disposal a set of background knowledge, or static data, such as ontologies and static Linked Data.

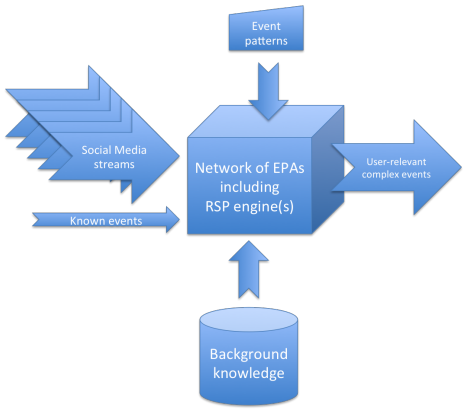

Fig. 1. Overall conceptual framework. Input streams are processed using a set of event patterns, and produces a stream of complex events in the output stream.

Event objects in the system can be modeled through the Event Processing ODP [9], which allows hierarchical relationships between event objects to be modeled using a light-weight ontology. The ontology can be used to define, fr example, simple and complex events objects [7], various types of timestamps, and relationships between events. The ontology was extended to be able to model relevance metrics between two event objects, in terms of spatial similarity, temporal similarity, and content similarity (see Section 5.3).

5 Experimental Setting

The purpose of this experiment was to evaluate the feasibility of using a current state-of-the-art RSP engine in processing streams of event and social media messages, including the calculation of basic relevance metrics on-the-fly. For the experiment our framework included the CQELS [6] engine for RSP, extended to support streams of RDF graphs, and the pipeline allowed the integration of custom EPAs, as well as the generation of new streams from RSP queries.

5.1 Event Generation

In the intended scenario, the event objects will be triggered by a stream of known events, for example, emergency calls or reports. However, for this experiment we have simulated this source through a set of event reports in the online traffic news.

The events were collected from Traffic England[1] and Traffic Scotland[2]. The collected events were manually modeled using an event ontology (see 5.3). For each traffic event the timestamp, geolocation, and nearest city was identified based on the metadata of the event. Additionally, a set of keyword phrases were manually extracted from the text description and added to the event, along with a set of synonyms. This process is fairly straightforward and could be automated for real-world scenarios.

The example below shows an incident, modelled using the event ontology, which was reported outside Stoke-On-Trent, UK, at 2015-02-17 08:00.

```
:event a eventodp:ComplexEventObject;
       eventodp:hasEventObjectTime '2015-02-17 08:00';
       eventontology:keyword 'accident', 'j15', 'j16',
            'junctions', 'delays', 'congestion", 'Stoke-On-Trent';
       rdfs:comment 'On the M6 northbound between junctions J15
            and J16, there are currently delays of 10 mins caused by
            congestion due to an earlier accident. Normal traffic
            conditions expected from 8:15 am.';
      eventontology:city 'Stoke-On-Trent';
      eventontology:country 'United Kingdom';
geo:lat '51.00';
            geo:lng '-1.00'.
```

5.2 Stream Data Collection

In this experiment we chose to use Twitter messages for enriching events, but the same strategy is applicable to many other microblogging services. Twitter's Streaming API allows tweets to be retrieved immediately as they are generated by users, based on search filter phrases, and the resulting stream can be additionally filtered geographically based on geotags. The Search API allows tweets to be queried from the set of archived tweets. The API follows roughly the same rules as the Streaming API with a few exceptions[3]. The access to the tweet stream for enriching an incident will be delayed compared to the actual event occurrence. This means that in order to collect tweets from the entire time interval of interest in real-time both the Search and Streaming API need to be used in combination.

Because only a small percentage of all tweets are geotagged, limiting the search based on geolocation risks filtering out too much of the potentially relevant tweets, this is especially important in the case of small scale events since the number of available tweets is very limited [11]. The searches did therefore not include any geolocation filter. For each event a list of twitter search phrases was generated based on: 1) the keywords associated with the event, and 2) the city and country associated with the event.

[1] http://www.trafficengland.com/
[2] https://trafficscotland.org/
[3] https://dev.twitter.com/

For each event in our experiment, the Twitter search issued generated a stream, collecting tweets back to 3 hours before the reported event occurrence and until 3 hours past the event occurrence[4]. All tweets were recorded with the associated account information and the optional place information. The details around the recording of the tweets allowed for accurate playback in the experimental setting.

The probability that a tweet is relevant with regard to a certain event is related to the spatial distance to the event [2, 11]. Only a very small number tweets are geotagged, but some tweets are additionally associated with a place, which in turn is associated with, among other things, a city/country field. Additionally, user accounts contain location information in the form of a string. These can often be used to roughly estimate a "probable" location from which a tweet was generated, typically at the city level. In comparing the geolocation of tweets and events we view geolocation at three levels of granularity; country, city, and GPS coordinate. The location information associated with tweets often identifies the geolocation to one of these levels. The most common patterns are listed in Table 1.

Pattern	*Granularity*
{city}, {state}, {country}	{city}
{city}, {country | state}	{city}
{city | state | country}	{city | country}

Table 1. The most common Twitter location description formats.

The events in this experiment were all located in England and Scotland. The 50K Gazetteer Linked Data from the Ordnance Survey[5] was used for place names, and if the country could be identified as the UK the tweet was enriched with country information.

For the geotagged tweets a conversion from WGS84 decimal system was made into OSGB36, commonly used in the UK, and the closest matching city was extracted from the gazetteer using a SPARQL query, within a limit of approximately 100 kilometres.

5.3 Ontology Extensions

Events were modeled using an extensions of the Event Processing ODP [9]. The ontology was extended to be able to model relevance between two event objects in terms of spatial similarity, temporal similarity, and content similarity (see Section 5.4). Additional properties were included to define country, city, GPS-coordinates, and keywords for event objects. Figure 2 shows the relevant

[4] This should appropriately account for possible errors in reported event time.
[5] http://data.ordnancesurvey.co.uk/datasets/50k-gazetteer

classes and properties used in the experiment. A basic schema was developed to

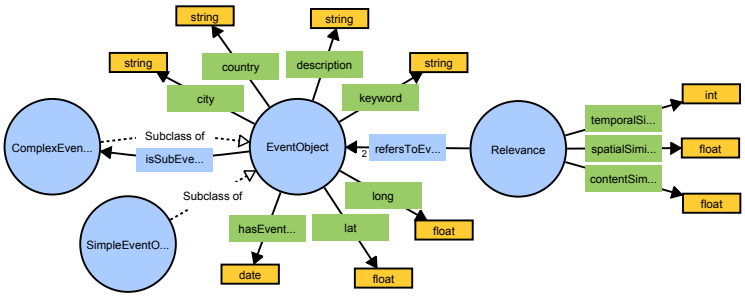

Fig. 2. The ontology used to represent events and relationships between events.

represent Twitter statuses, accounts, and places. A Twitter status was defined to be a `SimpleEventObject` in the event ontology. The Twitter schema can be viewed in Figure 3.

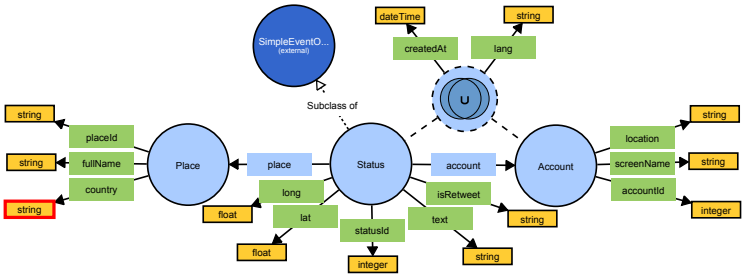

Fig. 3. Simplified version of the schema used for representing Twitter data.

5.4 Relevance metrics

Three relevance metrics were selected for the purpose of this experiment; 1) spatial similarity, 2) temporal similarity, and 3) content similarity. In the experimental setting the features 1) and 2) were calculated directly as part of a CQELS-QL query, while 3) was calculated as part of the enrichment phase.

Spatial similarity The spatial similarity between two events was based on country, city, and GPS coordinates. Since there are three levels of precision a direct comparison of geolocations is not possible. In the experiment the spatial similarity was defined as a positive real number, with lower scores indicating

higher spatial similarity. For simplification, the distance between two GPS coordinates was estimated as the sum of the absolute differences in latitude and longitude respectively.

Since the distance between events was defined as a relative metric high default values were used for non-GPS comparisons. If two events lacked GPS coordinates but were associated with the same country and city the similarity score was set to 100, if only the countries matched the similarity score was set to 1000, and otherwise the score was set to 10000.

Temporal similarity The temporal similarity between $event_1$ and $event_2$ was defined as the difference in seconds between the UNIX time values for the `hasEventObjectTime` property. Small absolute differences indicate that the events occurred near each other in time. A negative difference indicates that $event_2$ occurred before $event_1$.

Content similarity The content similarity can be assessed in many different ways. We applied a simple content similarity measure, based on the presence of event keywords. A simple way of improving this would be to provide weights for each of the content words, as was done in [4]. In the experiment workflow content similarity was calculated as part of the stream enrichment. It could also be calculated as an aggregate query using CQELS-QL, but since nested aggregation is currently not supported in CQELS it would require a separate query and the results would have to be published in a separate stream.

6 Results and Performance

The thresholds for the relevance metrics can be adjusted to restrict the amount of tweets that are added as sub-events, to filter out those which are likely to be irrelevant. Following the example in [11] we limited the temporal distance between events and tweets to 20 minutes, and filtered out re-tweets and directed messages. We also limited ourselves to tweets within the United Kingdom with a content matching of at least two, that is, at least two of the event keywords had to be present in the tweet. This restriction added as sub-events for a typical traffic incident a set of around 7 tweets, of which a majority were manually verified to be related to the incident. This number is quite close to the on identified in the machine learning approach [11]. Some events were, however, very mundane and were not covered at, such as re-curring rush-hour congestion.

In the experimental setting a stream of event objects, representing traffic incidents, and a stream of tweets were combined to produce an enriched stream of event objects. Geolocation information was added to the stream of tweets in an intermediate step, and a query generated the final stream of enriched traffic events. In an initial test all tweets were connected to all traffic incidents, regardless of the value for any of the relevance metrics. Although the workflow itself easily handles steady rates of 30–40 tweets per second on a standard notebook (i5 1.7GHz, 4GB RAM), the number of tweets added in the enrichment

step would be overwhelming for any user. By filtering out retweets and adding a threshold for the relevance metrics, the number of possibly relevant tweets was reduced drastically.

There are a few obvious bottlenecks in the current setup, particularly the geolocation of tweets, and the reversed geolocation used to identify nearest cities. Calculating the content similarity using aggregate queries and regular expressions in CQELS-QL queries also did not scale very well, and this was eventually instead embedded as a simple programmatic process in the tweet enrichment step.

7 Discussion and Future Work

Related studies have used various stream decoration techniques and external tools to detect events from social media streams. A machine learning approach to detect traffic incidents in Twitter data, while not operating on streaming data, indicated that it was likely to scale well in a streaming context [11].

The experiment in this paper tackled a similar task using rule-based interconnected EPAs. The rule-based approach makes the event detection and enrichment process tractable and transparent, where machine learning approaches are more opaque. Also, if new types of information become available the rule-based approach allows new sources to be added in a straightforward manner, where in machine learning approaches it may require the extraction of a new feature sets and a retraining of the model. However, some event patterns are likely to be inherently difficult to articulate using the rule-based approach, which means that the combination of the two could be desirable in many contexts.

A shortcoming that became apparent in the experiment was the absence of many commonly used mathematical operators in SPARQL. Calculation of geographical distances between geo-coordinates is not possible since there is no support for the necessary trigonometric functions. GeoSPARQL[6] was developed specifically to provide an extension to SPARQL for processing such geospatial Linked Data, and it would be possible to implement a similar extension for RSP engines, however, the performance of such an extension remains to be evaluated.

Complex Event Processing (CEP) depends to a great extent on the use of interconnected Event Processing Agents (EPAs) to create, such as hierarchically layered event objects, or aggregations of events. While the RSP engines comprise a set of flexible components for creating semantically enabled EPAs, they are in themselves not expressive enough to support all the types of task necessary in a full CEP architecture. Trying to solve tasks that are not suited for RSP may severely impact system performance, and EPAs targeted at specifically towards these particular tasks should be used.

In summary, our conclusion is that it is indeed feasible to use state-of-the-art RSP engines to perform event enrichment, and that quite simple preprocessing methods and relevance assessments on Social Media data can give quite reasonable results. RSP engines can provide an alternative to machine learning

[6] http://www.opengeospatial.org/standards/geosparql

approaches, avoiding the need for extensive training of models. However, there are clear limitations in the way that RSP engines handle location information, which is often essential in emergency response. Additionally, there may also be certain performance limitations, for example, related to the complexity of the event patterns that are to be detected, which leaves room for future work.

References

1. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: EP-SPARQL: A Unified Language for Event Processing and Stream Reasoning. In: Proceedings of the 20th International Conference on World Wide Web (2011)
2. Balduini, M., Della Valle, E., Dell'Aglio, D., Tsytsarau, M., Palpanas, T., Confalonieri, C.: Social Listening of City Scale Events Using the Streaming Linked Data Framework. In: Proceedings of the 12th International Semantic Web Conference, vol. 8219, pp. 1–16. Springer Berlin Heidelberg (October 2013)
3. Barbieri, D.F., Braga, D., Ceri, S., Valle, E.D., Grossniklaus, M.: Querying RDF streams with C-SPARQL. SIGMOD Record 39(1), 20–26 (2010)
4. Hobel, H., Madlberger, L., Thöni, A., Fenz, S.: Visualisation of User-Generated Event Information: Towards Geospatial Situation Awareness Using Hierarchical Granularity Levels. In: Workshop on Social Media and Linked Data for Emergency Response (SMILE2014) 11th Extended Semantic Web Conference (ESWC2014). CEUR workshop proceedings, Anissaras, Crete, Greece (May 2014)
5. Lampos, V., De Bie, T., Cristianini, N.: Flu detector: Tracking epidemics on Twitter. In: Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part III. pp. 599–602. Springer-Verlag, Berlin, Heidelberg (2010)
6. Le-Phuoc, D., Dao-Tran, M., Parreira, J.X., Hauswirth, M.: A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data. In: Proceedings of the 10th International Conference on the Semantic Web. pp. 370–388 (2011)
7. Luckham, D., Schulte, R.: Event Processing Glossary Version 2.0 (2011), http://www.complexevents.com/2011/08/23/event-processing-glossary-version-2-0/
8. Rinne, M., Abdullah, H., Törmä, S., Nuutila, E.: Processing Heterogeneous RDF Events with Standing SPARQL Update Rules. In: On the Move to Meaningful Internet Systems: OTM 2012. Lecture Notes in Computer Science, vol. 7566, pp. 797–806. Springer Berlin Heidelberg (2012)
9. Rinne, M., Blomqvist, E., Keskisärkkä, R., Nuutila, E.: Event Processing in RDF. In: ISWC 2013 Workshop: Proceedings of the 4th Workshop on Ontology and Semantic Web Patterns (WOP2013). CEUR workshop proceedings, Sydney, Australia (October 2013)
10. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World Wide Web. pp. 851–860. ACM, New York, USA (2010)
11. Schulz, A., Ristoski, P., Paulheim, H.: I See a Car Crash: Real-time Detection of Small Scale Incidents in Microblogs. In: Proceedings of Social Media and Linked Data for Emergency Response (SMILE) Co-located with the 10th Extended Semantic Web Conference. CEUR workshop proceedings, Montpellier, France (May 2013)