DeRiVE

ISWC2013, Sydney, Australia

Detection, Representation, and Exploitation of Events in the Semantic Web

Workshop in conjunction with the
12th International Semantic Web Conference 2013
Sydney, Australia, 21 October 2013

Edited by:
Marieke van Erp
Laura Hollink
Raphaël Troncy
Willem Robert van Hage
Piërre van de Laar
David A. Shamma
Lianli Gao

Preface

On Monday 21 October 2013, the third DeRiVE Workshop took place at the Sydney Masonic centre in Sydney, Australia. At the workshop, researchers from several communities came together to present and discuss their work on role of events the representation and organisation of knowledge and media.

We had defined questions for the two main directions that characterise current research into events on the semantic web:

1. How can events be detected and extracted for the semantic web?

2. How can events be modelled and represented in the semantic web?

The keynote by Emanuelle Della Valle and the papers presented at the workshop showed that very exciting things are happening in the research on extracting and representing events, in many different domains. However, the discussions also showed that there is still work to be done in order to really deal with events.

We would like to thank our programme committee members for reviewing and making it possible for us to put together the DeRiVE programme and the participants for their input. We hope to continue the our work at DeRiVE 2014.

February 2013
Marieke van Erp, VU University Amsterdam
Laura Hollink, VU University Amsterdam
Raphaël Troncy, EURECOM
Willem Robert van Hage, SynerScope B.V.
Piërre van de Laar, TNO
David A. Shamma, Yahoo! Research
Lianli Gao, University of Queensland

Programme Committee

The following colleagues kindly served in the workshop's program committee.

- Jans Aasman, Franz, Inc.,
- Eneko Agirre, University of the Basque Country
- Pramod Anatharam, Knoesis
- Michael Compton, CSIRO
- Christian Hirsch, University of Auckland
- Jane Hunter, University of Queensland
- Pavan Kapanipathi, Knoesis
- Azam Khan, Autodesk Research
- Jan Laarhuis, Thales
- Erik Mannens, Ghent University IBBT
- Ingrid Mason, Intersect
- Diana Maynard, University of Sheffield
- Giuseppe Rizzo, Universita' di Torino
- Matthew Rowe, Lancaster University
- Ryan Shaw, University of North Carolina at Chapel Hill
- Thomas Steiner, Google Inc
- Kerry Taylor, CSIRO & Australian National University

Contents

Keynote

Keynote: Listening to the pulse of our cities during City Scale Events

Emanuele Della Valle[1]

Politecnico di Milano

In recent years, we witnessed: the progressive instrumentation of our cities with diverse sensors; a wide adoption of smart phones and social networks that enable citizen-centric data acquisition; and a growing open release of datasets describing urban environments. As a result, nowadays it is becoming possible to make sense of all those data sources using semantic technologies together with streaming data analysis techniques. In particular, being able to feel the pulse of the city can allow delivering new services for the large number of organised and spontaneous events that take place in our cities. This talk frames the problem space, presents the challenges, proposes a solution centred on RDF stream processing that was applied to several case studies, and discusses the open problems that may be of interest of the Semantic Web community.

Domain-Independent Quality Measures for Crowd Truth Disagreement

Oana Inel[1][3], Lora Aroyo[1], Chris Welty[2], and Robert-Jan Sips[3]

[1] VU University Amsterdam
oana.inel@vu.nl, lora.aroyo@vu.nl
[2] IBM Watson Research Center, New York
cawelty@gmail.com
[3] CAS Benelux, IBM Netherlands
robert-jan.sips@nl.ibm.com

Abstract. Using crowdsourcing platforms such as CrowdFlower and Amazon Mechanical Turk for gathering human annotation data has become now a mainstream process. Such crowd involvement can reduce the time needed for solving an annotation task and with the large number of annotators can be a valuable source of annotation diversity. In order to harness this diversity across domains it is critical to establish a common ground for quality assessment of the results. In this paper we report our experiences for optimizing and adapting crowdsourcing micro-tasks across domains considering three aspects: (1) the micro-task template, (2) the quality measurements for the workers judgments and (3) the overall annotation workflow. We performed experiments in two domains, i.e. events extraction (MRP project) and medical relations extraction (Crowd-Watson project). The results confirm our main hypothesis that some aspects of the evaluation metrics can be defined in a domain-independent way for micro-tasks that assess the parameters to harness the diversity of annotations and the useful disagreement between workers. This paper focuses specifically on the parameters relevant for the 'event extraction' ground-truth data collection and demonstrates their reusability from the medical domain.

Keywords: Crowdsourcing, Ground-Truth, Event Extraction, Relation Extraction, NLP, Newspaper corpus

1 Introduction

At the basis for machine learning and information retrieval systems is the collection of ground truth data. Typically, creating such a gold standard dataset requires domain expert annotations to ensure high quality of the training and evaluation data. However, expert-annotation may result in limitedly annotated datasets, which do not capture the evolution of human expressions and the diversity in their interpretations. With its large pool of human workers, crowdsourcing became a mainstream source for higher volume and continuous collection of training and evaluation data (specifically for tasks that do not require domain expertise). Thus, the new challenge became to correctly and efficiently identifying low quality or spam contributions of the micro-workers. Research shows that micro-workers' behavior (e.g. either as intentional spam or low quality contributions) can influence the overall quality of the final results [1]. Typically, the quality is measured under the assumption that there is only one right answer for each micro-task and that it can be measured through annotators agreement [2].

Recently, however, there is evidence to support the hypothesis that harnessing diversity and disagreement between workers can improve the ground truth data [3]. Thus, it is critical to identify how much of the crowdsourced data is part of spam, low quality or actual meaningful disagreement between workers. There is an extensive body of research on spam detection through, e.g. majority decision [4], the expectation maximization [5]. Additionally, the micro-task template can impact the ability of the workers to complete the task successfully [6]. However, most of the studies have been focussing on addressing these issues as individual processes and less as part of a complete end-to-end workflow [7]. In this paper, we show that an optimal annotation workflow, which supports (1) apriori filtering of input data to maximize suitability for the workers and for the training, (2) crafting the templates to ensure proper disagreement collection and (3) defining appropriate metrics for low quality and spam filtering can impact beneficially the quality of the ground truth data, which we call *Crowd Truth* [8].

We conducted experiments in two domains - starting with **medical relation extraction** in the context of Crowd-Watson project[4] and adapting the experiences to **event extraction** in the context of DARPA's Machine Reading program (MRP)[5]. We used the same workflow in both domains: (1) *pre-processing of input data and micro-task template design*, (2) *data collection through automatic sequencing of micro-task batches*, (3) *disagreement analytics through quality metrics on workers judgments* and (4) *post-processing of the results for spam filtering and micro-task template adaptation.* The novel contribution of this work is twofold - on the one hand demonstrating a crowd truth collection workflow optimized for multiple domains; and on the other hand providing reusable disagreement-harnessing micro-task templates with the corresponding spam detection disagreement metrics.

The rest of the paper is organized as follows. Section 2 places this work in the context of crowdsourcing, evaluation metrics and event extraction. Section 3 presents the Crowd-Watson workflow and shows its adaptation for the event extraction task. Section 4 presents the experimental setup and Section 5 discusses the results. Section 6 draws the conclusions and presents the future work.

2 Related Work

The amount of knowledge that crowdsourcing platforms like CrowdFlower[6] or Amazon Mechanical Turk[7] hold fostered a great advancement in human computation [9]. Although the existing paid platforms manage to ease the human computation, it has been argued that their utility as a general-purpose computation platform still needs improvement [10]. Since the development of crowdsourcing has become more intensive, much research has been done in combining human and machine capabilities in order to obtain an automation of the crowdsourced process. Some state-of-the-art crowdsourcing frameworks are CrowdLang [10]

[4] https://github.com/laroyo/watsonc

[5] http://www.darpa.mil/OurWork/I2O/Programs/MachineReading.aspx

[6] https://crowdflower.com/

[7] https://www.mturk.com/mturk/

and CrowdMap [11]. However, CrowdLang restricts the users to work with its own internal programming language and CrowdMap solves only ontology alignment. Thus, both frameworks can be hardly adapted to another domain.

A lot of research has been focused on indentifying crowdsourced spam. Although a commonly used algorithm for removing spam workers is the majority decision [4], according to [12] it is not an optimal approach as it assumes all the workers to be equally good. Alternatively, expectation maximization [13] estimates individual error rates of workers. First, it infers the correct answer for each unit and then compares each worker answer to the one inferred to be correct. However, [14] shows that some tasks can have multiple good answers, while most spam or low quality workers typically select multiple answers. For this type of problem, some disagreement metrics [15] have been developed, based on workers annotations (e.g. agreement on the same unit, agreement over all the units) and their behavior (e.g. repetitive answers, number of annotations).

Research on events detection and extraction from medical texts [16], [17] is primarily focussed on improving the machine performance for it. In [16] the authors create an event trigger dictionary based on the original GENIA event corpus [18] and further, they apply dependency graphs for parsing the input corpus and extracting the putative events. [17] uses the Stanford Lexical Parser[8] for producing dependency graphs of the input corpus, as well as extracting the putative events. However, instead of using a dictionary for medical events, they only use the relations given by the dependency graphs.

Although there has been an extensive event extraction research using machines, the advantages of using crowdsourcing in this domain were not fully harnessed. Our new approach (fostering disagreement between annotators) [3] asks the crowd to judge the putative events and to provide event role-fillers at different granularities. The concept of harnessing disagreement in Natural Language Processing is not yet considered a mainstream process. In [19] disagreement is used as a trigger for consensus-based annotation in which all disagreeing annotators are forced to discuss and arrive at a consensus. This approach achieves κ scores above .9, but it is not clear if the forced consensus achieves anything meaningful. It is also not clear if this is practical in a crowdsourcing environment.

3 Adapting Crowd-Watson for Event Extraction

This section presents the workflow initially developed within the Crowd-Watson project (Figure 1) for creating ground truth data for medical relation extraction, that was further adapted for creating ground truth for newspaper events extraction. The resulting ground truth we refer to as *Crowd Truth*. In this paper we focus on the event extraction process for event crowd truth collection and its adaptation from the medical domain. A key point here is illustrating the reusability and optimization features of the workflow across the two domains.

The framework is designed as an end-to-end process which provides feedback loops that generate analysis for each stage of the workflow in order to improve future results. The *Pre-Processing* 3.1 component handles the adaptation of the input data for making it solving-affordable in terms of micro-tasks.

[8] http://nlp.stanford.edu/software/lex-parser.shtml

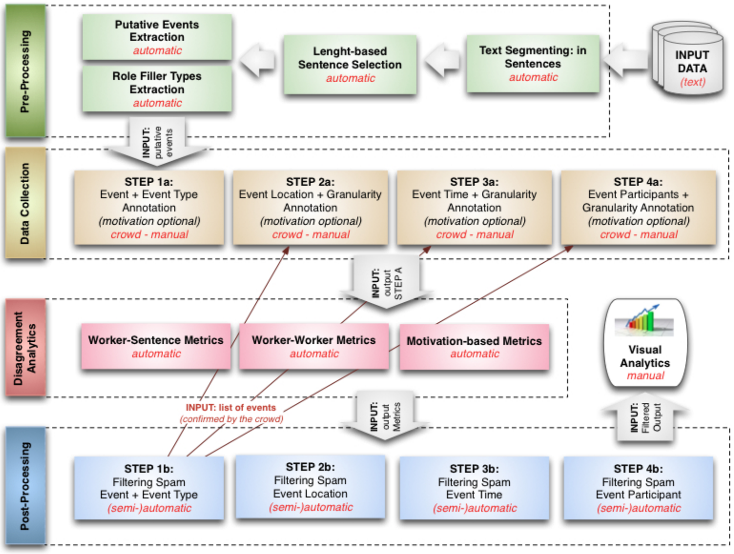


Fig. 1: Crowd-Watson Framework: Event Extraction Workflow Design

The *Data Collection* 3.2 uses CrowdFlower sequences of jobs for collecting judgments, while the *Disagreement Analysis* 3.3 component automatically handles the contributors evaluation. The *Post-Processing* 3.4 component automatically filters out the workers identified as spammers. Further, the process of collecting disagreement-based judgments can continue by reiterating each mentioned step.

3.1 Pre-Processing for Event Extraction

As, typically, the initial textual data collected from large sources, e.g. Wikipedia, newspapers first needs to be processed into smaller chunks suitable for micro-tasks (paragraphs, sentences). Further, to optimize its applicability for training, sentences that are not useful for training, e.g. too long or too short or contain specific words that increase the ambiguity need to be filtered out. The *Input Data Filtering* component clusters the input sentences based on their syntactic criteria, e.g. presence of semicolons, comma-separated lists, parentheses, etc. Each of those clusters can be either ignored or used for a specific micro-tasks. For example, sentences with putative events identified in them can be given to the crowd to confirm whether they refer to an event or not. Majority of those filters we directly reused from our medical relation extraction use case.

Input Data: For the experiments described in Section 4 we used articles from The New York Times. After their content was split into sentences (50 initial sentences), we removed the short sentences (less than 15 words). Compared with the task of medical relation extraction where the long sentences are typically difficult for the crowd, in the task of event extraction the longer the sentence the higher the chance that it will contain useful context for the event and the role fillers. This left us with 37 sentences to run the experiments with.

Putative Event Extraction: The first step in extracting events is to determine the *putative events* (verbs and nominalized verbs), i.e. word phrase that could possibly indicate an event. This component first exploits the *context-free phrase structure grammar* representation from the Stanford Parser to extract all the verbs and the nouns. Further, it follows the *typed dependencies parses* (also from the Stanford Parser) to extract word phrases that being in relation with certain verbs might trigger events. In addition to the Stanford Parser we also used NomLex[9], a dictionary of nominalizations. Thus, we extracted 205 putative events from the 37 sentences of the input data. For the crowdsourced experiments we selected only 70 putative events. Table 1 presents the putative events dataset.

Table 1: Putative Events Overview

Category	Putative Events	
	Article 1	Article 2
VB, VBD, VBG, VBN, VBZ, VBP[10]	61	57
Phrasal Verb	3	2
Verb + Direct Object	21	18
Predicate + Infinitive Verb	9	9
Adjectival Complement	2	1
Nominalized Verb	10	11
Nominalized Verb + Preposition "of"	2	0
Total: 205		

Micro-Task Template Settings In order to collect maximum diversity of answers from workers, and thus explore the disagreement space, we focus on the design of specific micro-task templates. Here again, the initial template settings were adapted from the medical relation extraction templates [8] and [14]. For the event extraction template we use a sentence with one putative event capitalized. Each template is based on conditional statements ("if clause"), which lead the worker through the template parts (see Figure 2).
Event annotation: Judge whether the capitalized word phrase refers to an event and motivate the answer. If the answer is yes, choose the type of the event.
Event role fillers: Judge whether the selected word phrase refers to an event. If the answer is yes, highlight the words referring to the attribute and choose its type(s). For participants template there is a follow-up question to choose a second participant. By allowing the worker to highlight words directly in the text instead of retyping them (in the relation extraction task much of the low contributions came from this aspect) we aim to improve the annotations collected.

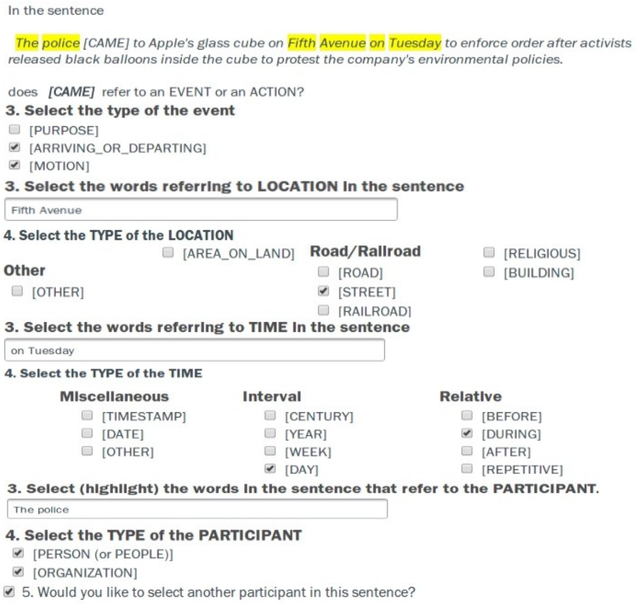
Role Fillers Taxonomies: Providing role filler selection ranges is demanding to form the annotator disagreement space. Thus, we align events, their types and role fillers to a set of simplified existing ontologies (to increase workers efficiency). For the *event type taxonomy* we used the semantic parser Semafor (Semantic Analysis of Frame Representations) [20] which identifies the frames of FrameNet 1.5[11] evoked by each English word. We set up Semafor with “strict” automatic target identification model, graph-based semi-supervised learning and AD^3 decoding type. The taxonomy includes a total of 12 top frames and grouped frames with similar semantics. The taxonomy for *event location* is based on GeoNames[12]. From each main GeoNames category we chose the most commonsense entities and various commonly used subclasses. Annotators often disagree

[9] http://nlp.cs.nyu.edu/nomlex/

[10] http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[11] https://framenet.icsi.berkeley.edu/fndrupal/

[12] http://www.geonames.org/ontology/documentation.html



In the sentence

The police [CAME] to Apple's glass cube on Fifth Avenue on Tuesday to enforce order after activists released black balloons inside the cube to protest the company's environmental policies.

does ***[CAME]*** refer to an EVENT or an ACTION?

3. Select the type of the event

□ [PURPOSE]
☑ [ARRIVING_OR_DEPARTING]
☑ [MOTION]

3. Select the words referring to LOCATION in the sentence

Fifth Avenue

4. Select the TYPE of the LOCATION

□ [AREA_ON_LAND]

Other

□ [OTHER]

Road/Railroad

□ [ROAD]
☑ [STREET]
□ [RAILROAD]

□ [RELIGIOUS]
□ [BUILDING]

3. Select the words referring to TIME in the sentence

on Tuesday

4. Select the TYPE of the TIME

Miscellaneous

□ [TIMESTAMP]
□ [DATE]
□ [OTHER]

Interval

□ [CENTURY]
□ [YEAR]
□ [WEEK]
☑ [DAY]

Relative

□ [BEFORE]
☑ [DURING]
□ [AFTER]
□ [REPETITIVE]

3. Select (highlight) the words in the sentence that refer to the PARTICIPANT.

The police

4. Select the TYPE of the PARTICIPANT

☑ [PERSON (or PEOPLE)]
☑ [ORGANIZATION]

☑ 5. Would you like to select another participant in this sentence?

Fig. 2: Event Extraction Template Design

which is the relevant level of granularity for temporal expressions. However, when gathering gold standard data for events we are interested in collecting all possible temporal expression. Thus, we combined four relative classes from Allen's time theory [21] with two time points and five time intervals from from KSL time ontology [22]. According to [23] the proper nouns strongly relate to participants in events. Thus, our *participants taxonomy* considers 5 classes that can be mostly represented by proper nouns. To foster diversity and disagreement, we added to each taxonomy the value "Other". Table 2 presents each taxonomy.

Table 2: Event Role Fillers Taxonomies

Role Filler	Taxonomy
Event Type	Purpose, Arriving or Departing, Motion, Communication, Usage, Judgment, Leadership, Success or Failure, Sending or Receiving, Action, Attack, Political, Other.
Location Type	Geographical - Continent, Country, Region, City, State, Area on Land - Valley, Island, Mountain, Beach, Forest, Park, Area on Water - Ocean, River, Lake, Sea, Road/Railroad - Road, Street, Railroad, Tunnel, Building - Educational, Government, Residence, Commercial, Industrial, Military, Religious, Other
Time Type	Before, During, After, Repetitive, Timestamp, Date, Century, Year, Week, Day, Part of Day, Other
Participants Type	Person, Organization, Geographical Region, Nation, Object, Other

Target Crowd Settings component applies context-specific restrictions on contributors, i.e. origin country, native language. After these basic conditions are applied, the *Crowdsourcing Workflow Definition* element sets the actual flow of the micro-task, i.e. judgments per unit and per worker, channels, payment.

3.2 Data Collection for Event Extraction

The Crowd-Watson[13] workflow framework [24] is targeted towards a crowd of lay workers and is developed as a micro-task platform on top of the crowdsourcing platform CrowdFlower. Additionally, Crowd-Watson supports also a gaming

[13] http://crowd-watson.nl

crowdsourcing platform[14], which targets nichesourcing with medical experts [25]. Crowd-Watson is specifically designed to stimulate the capture of disagreement and diversity in the annotator contributions. Figure 1 shows the specification of the components for event extraction.

3.3 Events Disagreement Analytics

This component assesses the quality of the workers contributions by analyzing the disagreement with *Worker Metrics* - worker agreement on a specific unit or across all the units that (s)he solved, and *Content Metrics* - the overall quality of the training data. This provides additional characteristics of the crowd truth, e.g. sentence clarity, similarity and ambiguity measures. In the event extraction task, the sentence vector is defined for each event property as the content of the aforementioned taxonomies and the "Not Applicable" value. This value is automatically added when: (1) the word phrase selected does not refer to an event, (2) there is no event property mentioned in the text snippet.

To avoid penalizing workers for contributing on difficult or ambiguous sentences, we filter sentences based on their clarity score [14]. Only then we apply the content-based worker metrics. The *worker-sentence agreement* measures the agreement between a worker annotation and the annotations of the rest of the workers for the same sentence (i.e. the averaged cosines between the worker sentence vector and the aggregated sentence vector, subtracting the worker's vector). The *worker-worker agreement* measures the agreement of the individual judgments of one worker with the judgments of all the other workers (i.e. the aggregated pairwise confusion matrix between the worker and the rest of the workers weighted by the number of sentences in common). The *number of annotations per sentence* is the average number of different types used by a worker for annotating a set of sentences.

3.4 Post-Processing for Event Extraction

The resulting analysis from these metrics provides input to *Post-Processing* to filter spam workers and spam-prone worker channels. The *Worker Spam Filtering* controls this flow. The list of spam micro-workers is sent to the *Data Collection* component to ban them from contributing to future tasks. Some statistics are also performed at the level of channels through *Crowdsourcing Channel Filtering.* Feedback is also sent to the *Pre-Processing* for improving the selection of input data, the optimization of micro-task settings and the workflow.

Finally, the *Visual-Analytics* component provides interactive visualization of (1) the workers behavior, (2) the sentence clarity and similarity. It provides a clear way to observe the dynamics in workers disagreement, completion time and the distribution of filters for spam contributions. The same visualization is used both for the relation extraction and for the event extraction tasks (Figure 3).

4 Experimental Setup

We adapted the Crowd-Watson medical relation extraction template for event extraction by constraining the workers to follow stricter rules, so that we can compare: (1) how does the new template influence the quality of the results;

[14] http://crowd-watson.nl/dr-detective-game/

and (2) and how does it effect the behavioral filters for spam and low quality contributions. We performed one preliminary experiment Exp0 3 to assess the applicability of the relation extraction template for the purposes of event extraction and established this as the baseline. We conducted four experiments (each two batches of 35 sentences for each event property, i.e 70 sentences), see Table 3. All experiments had the same settings, i.e. 15 judgments per sentence, 10 sentences allowed per worker, AMT, Vivatic, Prodege, Zoombucks and Crowdguru channels. This setting of two sequential runnings of small batches (of 35 sentences) allowed to:

Table 3: Experiments Overview

	# Sents	#Judgts per Sent	Channels	Max # Sents per Worker	#Judgts for Batch	# of Workers for batch	# Unique Workers
Exp0 Event+Type	35	15	crowdguru, zoombucks, vivatic, amt, prodege	10	525	66	66
Exp1 Event+Type	70	15	— " —	10	1050	147	141
Exp2 Event Location	70	15	— " —	10	1050	143	132
Exp3 Event Time	70	15	— " —	10	1050	146	140
Exp4 Event Participants	70	15	— " —	10	1050	141	137
Totals	*70*	*15*	— " —	*10*	*4200*	*643*	*436*
Totals (no singletons)	*70*	*15*	— " —	*10*	*4143*	*580*	*428*
Totals (no singletons, doubletons)	*70*	*15*	— " —	*10*	*4102*	*539*	*421*

– get enough judgments per sentence given there is no golden standard;

– optimize the time to finish a large sentence set

– get a quick run of the entire workflow.

However, as the split in small batches could allow spam workers to perform every batch, this could have exponential negative effect on the quality of the annotations. Thus, we optimized the spam filtering by:

– limiting the number of judgments per worker in a batch;

– applying spam filtering after each batch and blocking them from future jobs.

5 Results and Discussion

In this section we analyze the entire experimental workflow. We observe the effect of the template design on the accuracy of the crowdsourced data, and we measure the accuracy of the worker metrics compared to the accuracy of the worker metrics together with the explanation-based metrics.

The preliminary experiment (Exp0) for identifying events and event types did not use a conditional micro-task template. Nine workers submitted only one or two judgments, which did not provide evaluation relevance, and were thus excluded from the analysis. The contributions of 66 remaining workers were analyzed further. The worker metrics identified 15 spam contributors, while the explanation-based filters, described in [14] identified another 10 spam contributors. However, upon a manual evaluation of the results 5 more workers were identified with an erratic behavior, selecting either "no event" and a type different than "Not Applicable", or "yes event" and "Not Applicable" type. Such contributions could be because of intentional spamming, or negligence or misunderstanding of the task. This result guided us to a more restrictive template to improve the job outcome. The new event extraction template (Figure 2) did not allow the workers to choose simultaneously: (1) "Not Applicable" and other event property type, (2) "no event" and a type different than "Not Applicable", (3) "yes event" and "Not Applicable" type.

When adapting the taxonomies we tried to conceive different experimental cases that could give insights in the adaptability degree of the metrics:

- the list of event types, time types and participant types are similar to the number of relations provided in the medical relation extraction task;
- the taxonomy for location type is more diverse, with overlapping concepts;
- one event can have multiple participants, which can increase the number of annotation for a putative event; this is a relevant factor for evaluating the behavior the average number of annotations metric;

We performed a manual sampling-based evaluation of workers in order to determine the accuracy of the spam metrics. We examined all the workers marked as spam by the filters, as well as the ones ranked as best workers. Some workers in the gray area inbetween were also manually evaluated. Figure 4 shows the precision, recall and F-measure only for the worker metrics for each job type, i.e. event type, event location, event time and event participants. Although the worker metrics identified a high percentage of low-quality contributors, the accuracy and the precision of the metrics still need improvements. A reason for this behavior could strongly relate to event properties types distribution and similarity, which varies along the four classes of event characteristics.

For the *Event Type* task the *average number of annotations* metric was able to identify correctly a high amount of spammers. However, both the worker-sentence and worker-worker agreement had low values in terms of correctness. The distribution of event types among 35 sentences, i.e. putative events (see Figure 3) indicates high ambiguity of the event types. One reason for this could be our choice of event types, which might not be so appropriate for the workers. However, in the *Location Type* task we see a different picture. As expected, most workers chose multiple types for one identified location. For example, the highlighted location for one putative event was "Apple's glass cube on Fifth Avenue" and was classified as: [COMMERCIAL], [BUILDING], [ROAD], [STREET]. Although the worker did agree to a certain extent with the other workers solving the same sentence, he was wrongly classified as spammer based on the number of annotations. Thus, the accuracy of spam identification for location type is solely based on the worker-worker and worker-sentence agreements metrics. The F-measure for event type is equal to 0.89, while for event location is equal to 0.82. Even though the percentages of identified low-quality contributors were comparable for both experiments, the lower number of correct predictions for event location stands as a reason for a lower event location type F-measure. For the *Time Type* task the assessment of metrics behavior was the most challenging. More than half of the sentences used in the experiment did not contain any event time reference, and most of the workers chose "[NOT APPLICABLE]" as a type. This resulted in a high worker-worker and worker-sentence agreement scores. Thus, if a worker would disagree with other workers on just one sentence, (s)he would be identified as spam. Most of the spammers, however, were

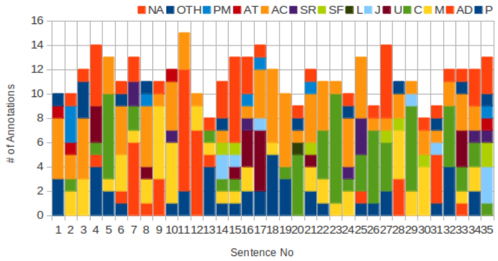


Fig. 3: Annotation Distribution - Event Types

captured by at least two worker metrics. The ones identified by less than two metrics justify the high recall and the low precision (Figure 4) (F=0.81).

One event could have multiple participants of different types. However, the highlighted participants mentions in the task were of the same type, which explains why the average number of annotations per sentence did not have an erratic behavior. The *event participants type* has the highest F-measure value (0.91). This value is a result of the high amount of spammers correctly identified as well as the high amount of spammers identified from the entire list of spammers. Thus, we can conclude that the participants taxonomy presented to the workers is concise and covers with high precision the possibilities of interpreting the participants of an event.

The high worker disagreement in the *event type* experiments gave the worker metrics an important boost of efficiency, by identifying a high amount of true spammers. However, the overall agreement was above mean expectations. Thus, the workers that did not highly agree with other workers were prone to be identified as spammers. For *event location*, however, the average number of annotations per putative event decreased the total precision of correctly identified spammers. As seen in Figure 4, the applied worker measurements had the most accurate behavior for the *event participants type* task.

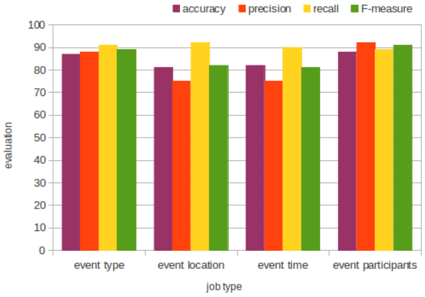


Fig. 4: Worker Metrics Evaluation

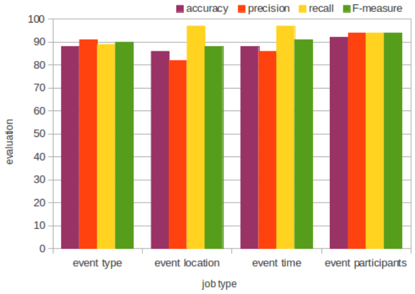


Fig. 5: Worker Metrics and Explanation-based Filters Evaluation

By looking at Figure 5 we can see that the explanations provided by the workers regarding their answers give an important boost of efficiency in identifying patterns that are associated with low-quality contributions and even spam contributions. In combination with the worker metrics, these explanation-based filters are able to increase the accuracy of detecting low-quality workers with at least 5%. This situation was possible because only a small number of workers were identified as spammer by both explanation-based filters and disagreement-based metrics (worker metrics). Thus, for each batch, not more than 2 or 3 workers were identified by both quality measurements. Hence, it seems reasonable to further use the advantages brought up by those filters. This conclusion is also underlined by the usage of all the channels, situation that is usually associated with an increased percentage of spammed results. The results presented in Figure 5 make a good case to state that the usage of both worker metrics and explanation-based filters achieves high accuracy in terms of crowdsourced data. With the results mentioned in Figure 5 we can state that we succeeded to achieve high accuracy in identifying the spam workers, but we also showed how the metrics are suitable across domains.

6 Conclusions and Future Work

This paper presents the results of our experiments on estimating the reusability and domain-dependency of crowdsourcing workflow components, such as processing of input data, micro-task templates and result assessment metrics. We demonstrated how components defined in one domain (medical relation extraction) can be easily adapted to a completely different domain (event extraction from newspapers). Results from the experiments showed that some of the metrics for workers and content can be applied with high precision in those domains. For understanding to what extent the domains can be similarly treated, we conducted different research at each step of the crowdsourced process.

By directly reusing pre-processing filters from the medical relation extraction domain we showed that the input data can be optimized using syntactic text features. Thus, we can argue that the syntactic text features are mostly domain-independent. Although the template design was adapted for stimulating diversity in worker answers, the metrics were still able to capture the low quality contributions in both domains. With the final template design for event extraction, the workers were less prone to spam the results by mistake. We have showed that especially for domains were there is no golden data known in advance, the explanations can be successfully used to identify more spam or low-quality workers. When the explanation filters are combined with the disagreement worker metrics the accuracy of detecting those low-quality contributors reaches a value greater than 92%. To sum up, the adaptation of the disagreement analysis component from the medical relation extraction to the event extraction preserved its good outcomes, and thus, these disagreement metrics are domain-independent.

As part of this research, our future work should focus on solving ambiguity-related aspects. First, our analysis showed that there is still space for improving the event properties types. The event-type taxonomy shows a lot of ambiguity when looking at the workers annotations distribution. Further experiments should clarify whether a different classification of the putative events can achieve a better performance compared to the current experiments. Also, we need to conclude how the types that are overlapping influence the results. Furthermore, each word phrase highlighted from the sentences needs to be clustered in order to determine the most appropriate structure of the event role filler.

References

1. Vuurens, J., de Vries, A.P., Eickhoff, C.: How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In: Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIRâĂŹ11). (2011) 21–26
2. Nowak, S., Rüger, S.: How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In: Proceedings of the international conference on Multimedia information retrieval, ACM (2010) 557–566
3. Aroyo, L., Welty, C.: Harnessing disagreement for event semantics. Detection, Representation, and Exploitation of Events in the Semantic Web (2012) 31
4. Hirth, M., Hoßfeld, T., Tran-Gia, P.: Cost-optimal validation mechanisms and cheat-detection for crowdsourcing platforms. In: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), IEEE (2011) 316–321
5. Ipeirotis, P.G., Provost, F., Wang, J.: Quality management on amazon mechanical turk. In: ACM SIGKDD workshop on human computation, ACM (2010) 64–67
6. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with mechanical turk. In: SIGCHI conference on human factors in computing systems, ACM (2008)
7. Kittur, A., Nickerson, J.V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Lease, M., Horton, J.: The future of crowd work. In: Proceedings of the 2013 conference on Computer supported cooperative work, ACM (2013) 1301–1318
8. Aroyo, L., Welty, C.: Crowd truth: Harnessing disagreement in crowdsourcing a relation extraction gold standard. WebSci2013. ACM (2013)
9. Quinn, A.J., Bederson, B.B.: Human computation: a survey and taxonomy of a growing field. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM (2011) 1403–1412
10. Minder, P., Bernstein, A.: Crowdlang-first steps towards programmable human computers for general computation. In: Human Computation. (2011)
11. Sarasua, C., Simperl, E., Noy, N.F.: Crowdmap: Crowdsourcing ontology alignment with microtasks. In: The Semantic Web–ISWC 2012. Springer (2012) 525–541
12. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. The Journal of Machine Learning Research **99** (2010)
13. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the em algorithm. Applied Statistics (1979) 20–28
14. Soberón, G., Aroyo, L., Welty, C., Inel, O., Lin, H., Overmeen, M.: Crowd truth metrics. Technical report, VU University Amsterdam (2013)
15. Aroyo, L., Welty, C.: Measuring crowd truth for medical relation extraction. AAAI2013 Fall Symposium on Semantics for Big Data (in print) (2013)
16. Buyko, E., Faessler, E., Wermter, J., Hahn, U.: Event extraction from trimmed dependency graphs. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task, Association for Computational Linguistics (2009) 19–27
17. Kilicoglu, H., Bergler, S.: Syntactic dependency based heuristics for biological event extraction. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task. (2009) 119–127
18. Kim, J.D., Ohta, T., Tsujii, J.: Corpus annotation for mining biomedical events from literature. BMC bioinformatics **9**(1) (2008) 10
19. Ang, J., Dhillon, R., Krupski, A., Shriberg, E., Stolcke, A.: Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In: INTERSPEECH, Citeseer (2002)
20. Das, D., Schneider, N., Chen, D., Smith, N.A.: Semafor 1.0: A probabilistic frame-semantic parser. Language Technologies Institute, School of Computer Science, Carnegie Mellon University (2010)
21. Allen, J.F., Hayes, P.J.: A common-sense theory of time. Volume 85. (1985)
22. Zhou, Q., Fikes, R.: A reusable time ontology. Technical report, KSL-00-01, Stanford University (2000)
23. Hatzivassiloglou, V., Filatova, E.: Domain-independent detection, extraction, and labeling of atomic events, Proceedings of the RANLP Conference (2003)
24. Lin, H., Inel, O., Soberón, G., Aroyo, L., Welty, C., Overmeen, M., Sips, R.J.: Crowd watson: Crowdsourced text annotations. Technical report, VU University Amsterdam (2013)
25. Dumitrache, A., Aroyo, L., Welty, C., Sips, R.J., Levas, A.: Dr. detective: combining gamification techniques and crowdsourcing to create a gold standard for the medical domain. Technical report, VU University Amsterdam (2013)

FRED as an Event Extraction Tool

Aldo Gangemi[1,2], Ehab Hassan[1], Valentina Presutti[2], Diego Reforgiato Recupero[2]

[1] LIPN, Université Paris13-CNRS-SorbonneCité, France
[2] STLab, ISTC-CNR, Rome-Catania, Italy.

Events are elusive entities; as the authors of [7] argue, even human annotators do not agree on what is an event and what is its boundary in terms of the extension of its participants, temporal and geospatial extent, etc.

More aspects of events appear when trying to recognize or extract them from text: polarity of speaker's judgment on events, negation, modality, relations (temporal, causal, declarative, etc.) to other events, etc.

For example, the text:

The Black Hand might not have decided to barbarously assassinate Franz Ferdinand after he arrived in Sarajevo on June 28th, 1914.

expresses three events (*decide, assassinate, arrive*), with *Black Hand* being a participant in two of them, *Franz Ferdinand* in the third (*arrive*), a temporal extent for the third (*June 28th, 1914*), and a relative temporal extent for the other two (given the third's extent and the past tense suffixes in the first and third), a geo-spatial extent (*Sarajevo*), a judgment with negative polarity on the second event (*barbarously*), a negation (*not*) over the modality (*might*) modifying the first event, and an explicit temporal relation between the second and third event (*after*).

Extracting, logically representing, and connecting elements from a sentence is crucial to create semantic applications that are event-aware. In addition, it's important to disambiguate as much as possible the entities and concepts expressed, in order to make the extracted model *linked*, and to exploit the full power of the Semantic Web and Linked Data.

FRED[1] [5] is a tool to automatically transform knowledge extracted from text into RDF and OWL, i.e. it is a *machine reader* [2] for the Semantic Web. It is event-centric, therefore it natively supports event extraction. In a recent landscape analysis of knowledge extraction tools [3], FRED has got .73 precision, .93 recall, and .87 accuracy, largely better than the other tools attempting event extraction.

FRED is available as a RESTful API and as a web application. In its current form, it relies upon several NLP components: Boxer[2] for the extraction of the basic logical form of text and for disambiguation of events to VerbNet, UKB[3] or IMS[4] or BabelNet API[5] for word sense disambiguation, and Apache Stanbol[6] for named entity resolution.

[1] `http://wit.istc.cnr.it/stlab-tools/fred`

[2] `http://svn.ask.it.usyd.edu.au/trac/candc/wiki/boxer`

[3] http://ixa2.si.ehu.es/ukb/

[4] http://www.comp.nus.edu.sg/~nlp/sw/

[5] http://lcl.uniroma1.it/babelnet/

[6] http://stanbol.apache.org

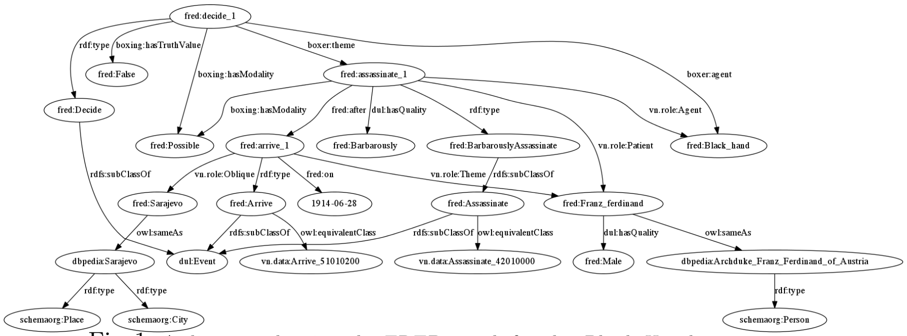


Fig. 1: A diagram showing the FRED graph for the *Black Hand* sentence.

FRED contains several functionalities for event extraction, which can be summarized according to typical subtasks:

- Event identity: FRED focuses on events expressed by *verbs*, *propositions*, *common nouns*, and *named entities* (typically proper nouns).
- Event classification: FRED uses Linked Data-oriented induction of types for the identified events, reusing e.g. VerbNet[7], WordNet[8], DBpedia[9], schema.org, and DOLCE[10] as reference ontologies.
- Event unity: FRED applies *semantic role labeling* [4] to verbs and propositions in order to detect event boundaries, and *frame detection* [1] for resolving roles against a shared event ontology.
- Event modifiers: FRED extracts *logical negation*, *basic modalities*, and *adverbial qualities*, applied to verbs and propositions, which can then be used as event judgment indicators.
- Event relations: FRED relates events via the role structure of verbs and propositions, and extracts *tense relations* between them.

The beginning and the following sentences are used as a lead example for showing FRED's functionalities:

The Renaissance was a cultural movement that spanned in Italy from the 14th to the 17th century. Some sources report that the Renaissance might have been started by Greek scholars from Constantinople.

In the diagram from Figure 2, the following events are recognized, extracted, classified, and aligned to WordNet, VerbNet, and/or DOLCE: `Renaissance` (classified as a `Movement`, and aligned to the WordNet `Motion` synset, and to the DOLCE `Situation` class), `span_1`, `report_1`, and `start_1` (classified as occurrences of the `Span`, `Report` and `Start` frames respectively, and aligned to VerbNet).

Furthermore, the events have participants (e.g. `Italy`, `scholar_1`, `source_1`, etc., also classified and linked appropriately) through some roles labelled with properties derived from VerbNet(e.g. `vn:Agent`), or from the lexicon used in the sentence (e.g. `ren:from`) In one case, a modal modifier (`Possible`) to the event `start_1` is added.

[7] http://verbs.colorado.edu/˜mpalmer/projects/verbnet.html?

[8] http://wordnet.princeton.edu

[9] http://dbpedia.org

[10] http://www.ontologydesignpatterns.org/ont/dul/DUL.owl

Finally, some relations between events are detected: `report_1 vn:Theme start_1`, and `span_1 before report_1` (through the `now_1` interval).

See also Figure 1 for the graph obtained from the beginning sentence.

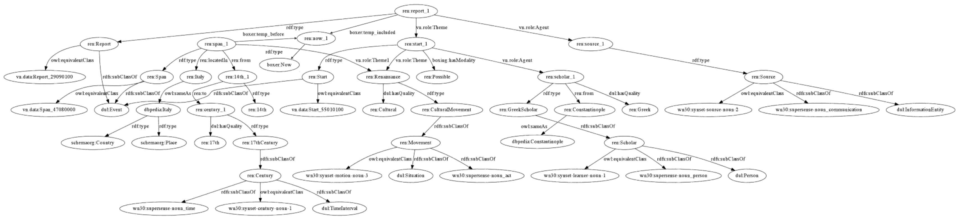


Fig. 2: A FRED graph depicting the core subset of triples representing event-related knowledge.

The triples given as output by FRED are more than those visualized, for example they include text spans and their reference to the semantic annotations, through the Earmark vocabulary [6].

FRED is therefore an intermediate component for event extraction and representation, which can be augmented with background knowledge, and whose graphs can be combined e.g. in time series for historical tasks.

FRED will be demoed as an event extractor by showing event-intensive sentences, and examples of views that focus on relevant event knowledge. RDF models can be morphed to concentrate on specific features. For example, Figure 3 semantically summarizes the model from the *Black Hand* sentence by only showing events with their relations, and their main participant, obtained by means of the following SPARQL query:

```
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
PREFIX vnrole: <http://www.ontologydesignpatterns.org/ont/vn/abox/role/>
PREFIX boxing: <http://www.ontologydesignpatterns.org/ont/boxer/boxing.owl#>
PREFIX boxer: <http://www.ontologydesignpatterns.org/ont/boxer/boxer.owl#>
PREFIX : <http://www.ontologydesignpatterns.org/ont/boxer/test.owl#>
CONSTRUCT {?e :agent ?x . ?e ?r ?e1}
WHERE {
 {{?e a boxing:Situation} UNION {?e a ?class . ?class rdfs:subClassOf+ dul:Event}}
 ?e ?p ?x
 FILTER (?p = vnrole:Agent || ?p = boxer:agent || ?p = vnrole:Experiencer || ?p = vnrole:Actor
     || ?p = vnrole:Actor1 || ?p = vnrole:Actor2 || ?p = vnrole:Theme)
 FILTER NOT EXISTS {?e vnrole:Theme ?x . ?e vnrole:Agent ?y
 FILTER (?x != ?y)}
 OPTIONAL {{{?e ?r ?e1} UNION {?e ?s ?z . ?z ?t ?e1}} {{?e1 a boxing:Situation} UNION
    {?e1 a ?class1 . ?class1 rdfs:subClassOf+ dul:Event}} FILTER (?e != ?e1)}}
```

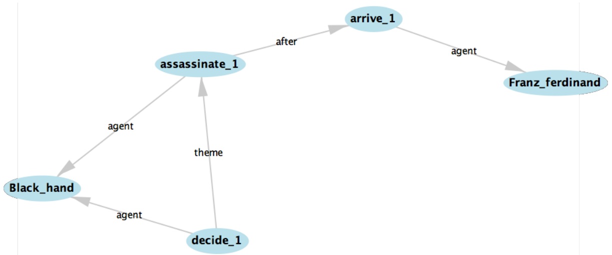


Fig. 3: A summarized FRED graph showing only event relations and agentive participants for the Black Hand sentence.

References

1. Bonaventura Coppola, Aldo Gangemi, Alfio Massimiliano Gliozzo, Davide Picca, and Valentina Presutti. Frame detection over the semantic web. In Lora Aroyo et al., editor, *ESWC*, volume 5554 of *LNCS*, pages 126–142. Springer, 2009.
2. Oren Etzioni, Michele Banko, and Michael Cafarella. Machine reading. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, 2006.
3. Aldo Gangemi. A comparison of knowledge extraction tools for the semantic web. In *Proceedings of ESWC2013*. Springer, 2013.
4. A. Moschitti, D. Pighin, and R. Basili. Tree kernels for semantic role labeling. *Computational Linguistics*, 34(2):193224, 2008.
5. Valentina Presutti, Francesco Draicchio, and Aldo Gangemi. Knowledge extraction based on discourse representation theory and linguistic frames. In *EKAW: Knowledge Engineering and Knowledge Management that matters*. Springer, 2012.
6. Peroni S., Gangemi A., and Vitali F. Dealing with markup semantics. In *Proceedings of the 7th International Conference on Semantic Systems, Graz, Austria (i-Semantics2011)*. ACM, 2011.
7. Chris Welty and Lora Aroyo. Harnessing disagreement for event semantics. In *Proc. of DERIVE Wks at ISWC2012*, 2012.

Representing Supply Chain Events on the Web of Data

Monika Solanki and Christopher Brewster

Aston Business School
Aston University, UK
[m.solanki, c.a.brewster]@aston.ac.uk

Abstract. The Electronic Product Code Information Service (EPCIS) is an EPCglobal standard, that aims to bridge the gap between the physical world of RFID[1] tagged artifacts, and information systems that enable their tracking and tracing via the Electronic Product Code (EPC). Central to the EPCIS data model are "events" that describe specific occurrences in the supply chain. EPCIS events, recorded and registered against EPC tagged artifacts, encapsulate the "what", "when", "where" and "why" of these artifacts as they flow through the supply chain. In this paper we propose an ontological model for representing EPCIS events on the Web of data. Our model provides a scalable approach for the representation, integration and sharing of EPCIS events as linked data via RESTful interfaces, thereby facilitating interoperability, collaboration and exchange of EPC related data across enterprises on a Web scale.

1 Introduction

RFID and other pervasive computing technologies empower trading partners, by enabling the capture and sharing of knowledge about the identity and location of physical items and goods as they move along the supply chain. RFID readers deployed at strategic locations on partner premises and transit points can record and register crucial information against the Electronic Product Code (EPC) [2] of items. The Electronic Product Code Information Service (EPCIS)[3] is a ratified EPCglobal[4] standard that provides a set of specifications for the syntactic capture and informal semantic interpretation of EPC based product information as it moves along the supply chain.

An observation of most existing supply chain processes highlights two crucial data sharing limitations. For any given supply chain process, a large number of RFID events are recorded at each partner's end. This leads to large volumes of event data which are inherently related but are rendered disconnected due to the

[1] We use RFID as a generic terms for all methods of tagged product identification.

[2] http://www.gs1.org/gsmp/kc/epcglobal/tds

[3] http://www.gs1.org/gsmp/kc/epcglobal/epcis

[4] http://www.gs1.org/epcglobal

design of the underlying data schemas and the curation techniques employed. EPCIS event data silos are thus created within each participating partner's EPCIS infrastructure. Further, the EPCIS XML schemas define only the structure of the event data to be recorded. The semantics of event data and data curation processes are informally defined in the specification. Their interpretation is left up to the individual EPCIS specification implementing engines, thereby highly increasing the possibility of interoperability issues arising between supporting applications, e.g., validation and discovery services built over the event repositories.

In order to enable a more meaningful representation of the event based product lifecycle as it moves along the supply chain and thereby, simplify the process of sharing EPCIS event data among partners, we propose an event model, the *EPCIS Event Model* (EEM)[5], that enables the sharing and semantic interpretation of EPCIS event data. Our model exploits Semantic Web standards/linked data technologies, and draws requirements from business processes involved in the tracking and tracing of goods. EPCIS event datasets curated and harnessed as linked data can be exploited using analysis techniques such as data mining in order to improve visibility, accuracy and automation along the supply chain. Since the recorded data is a reflection of the behaviour of the participating business processes, it can be used to derive implicit knowledge that can expose inefficiencies such as shipment delay, inventory shrinkage and out-of-stock situation.

The paper is structured as follows: Section 2 provides a brief background and highlights related work. Section 3 discusses the informal intuition behind EPCIS events. Section 4 presents EEM, the EPCIS Event Model. Section 5 provides implementation background. Section 6 illustrates an exemplifying scenario from the agri-food supply chain and finally Section 7 presents conclusions.

2 Background and Related Work

An Electronic Product Code (EPC) [6] is a universal identifier that gives a unique, serialised identity to a specific physical object. As the RFID-EPC tagged object moves through the supply chain, EPCIS implementing applications deployed at key locations record data against the EPC of the object. The EPCIS specification defines two kinds of data: event data and master data. Event data arises in the course of carrying out business processes, it grows over time and is captured through the EPCIS capture interface and made available for querying through the EPCIS Query Interfaces. An example of event data is "At Time T, Object X was observed at Location L.". Master data is additional data that provides the necessary context for interpreting the event data.

A plethora of interpretations can be derived from and assigned to the term "Event" depending on the contextual domain and the temporal dimension of its

[5] http://fispace.aston.ac.uk/ontologies/eem#

[6] http://www.gs1.org/gsmp/kc/epcglobal/tds/tds_1_6-RatifiedStd-20110922.pdf

occurrence. The representation of events has been an important aspect of linked datasets emerging from the domain of history [3], multimedia [1], geography [5], journalism [7] and cultural heritage [2]. A survey of existing models for the representation of historical events on the Semantic Web is presented in [8].

The Event ontology[8] emerged from the need of representing knowledge about events related to music. The ontology provides a minimum event model. It defines a single concept as a class `Event` and a few defined classes. The Linking Open Descriptions of Events (LODE) [9] [8] ontology is similar in spirit to the EEM in that it focuses on the four factual aspects of an event. Properties defined in this ontology are aligned with approximately equivalent properties from other models.

An extensive information model, the CIDOC-CRM[10] is an ontology for representing cultural heritage information. Classes such as `E5_Event` and `E4_Period` can be specialised for representing events. The Event-Model-F [7] is a formal model based on the DOLCE+DnS Ultralite ontology. The high level goal of the model is to represent events with explicit human participation, by modelling causal relationship between events and their varied interpretations. The Simple Event Model (SEM)[11], with weak semantics and requirements drawn from the domain of history and maritime security and safety is presented in [10]. The notion of an event here is general purpose and the model is designed with minimum semantic commitment.

Few research efforts have focused on EPCIS events. In [4], the authors present a supply chain visualisation tool for the analysis of EPCIS event data. In [6] a data model and algorithm for managing and querying event data has been proposed. A critical limitation of this model is that it is overlayed on top of relational databases and is not available in a form that can be shared and reused between organisations as linked data. In [9] the authors propose to use the InterDataNet (IDN) [12] framework for the sharing of EPCIS data. The proposed approach suffers from several critical limitations such as lack of a reusable and shared data model and the encapsulation of information as an additional IDN document layer which may significantly affect performance of querying applications.

3 EPCIS events: The Informal Intuition

The EPCIS standard defines a generic event and four different physical event types, arising from supply chain activities across a wide variety of industries.

- *EPCISEvent* represents the generic EPCIS event.
- *ObjectEvent* represents an event that occurred as a result of some action on one or more entities denoted by EPCs.

[7] `http://data.press.net/ontology/event/`

[8] `http://motools.sourceforge.net/event/event.html`

[9] `http://linkedevents.org/ontology/`

[10] `http://www.cidoc-crm.org/rdfs/cidoc_crm_v5.0.4_official_release.rdfs`

[11] `http://semanticweb.cs.vu.nl/2009/11/sem/`

[12] `http://www.interdatanet.org/`

- *AggregationEvent* represents an event that happened to one or more EPC-denoted entities that are physically aggregated (constrained to be in the same place at the same time, as when cases are aggregated to a pallet).
- *QuantityEvent* represents an event concerned with a specific number of objects all having the same type, but where the individual instances are not identified. For example a quantity event could report that an event happened to 200 boxes of widgets, without identifying specifically which boxes were involved.
- *TransactionEvent* represents an event in which one or more entities denoted by EPCs become associated or disassociated with one or more identified business transactions.

Each EPCIS event, recorded and registered against RFID tagged artifacts has four information dimensions. It encapsulate the "what", "when", "where" and "why" of these artifacts at the RFID scan point.

- *what*: indicates the central characteristic (e.g., List of EPCs for an *ObjectEvent* or EPCClass for a *QuantityEvent*) of item(s) captured by the event. This information artifact differs for each of the event types.
- *when*: indicates the date and time at which the event took place.
- *where*: indicates the business location identifiers of the place where the event took place as well as where the physical objects are expected to be following the event.
- *why*: indicates the business context of the event. In particular,
  - business step or business activity that raised the event, e.g., receiving, shipping.
  - business state (disposition) of the object after the event took place, e.g., saleable, active, transit.

EPCIS identifiers for events, products and locations are represented using URIs. Formats for the URIs have been prescribed in the GS1 EPC Tag Data Standard [13] for identifying the EPCs.

4 EEM: The EPCIS Event Model

In this section we motivate the modelling decisions we took while defining the conceptual model behind EEM and describe its structure.

4.1 Modelling Decisions

In contrast to some of the general purpose event models reviewed in Section 2, EEM is domain specific. For practical purposes, the data model underlying EEM, restricts the entities, relationship and attributes to a subset of the EPCIS specification, albeit a large subset. Our objective was to propose a model that

[13] http://www.gs1.org/gsmp/kc/epcglobal/tds/tds_1_6-RatifiedStd-20110922.pdf

provides conceptual primitives with the appropriate level of semantic abstraction required to model the various kinds of EPCIS events that can be raised and the four information dimensions they encapsulate. The design of EEM was influenced by the following decisions:

- *Level of expressivity*: Most data models for the Web of data are designed with relatively weak semantics. This is desirable if the intent is to allow the integration of cross domain datasets, described using vocabularies with multiple and differing viewpoints for similar conceptual entities. Weak semantics lead to fewer inconsistencies when reasoning over integrated/linked datasets. While designing the semantic structure of EEM, we wanted a model that could constrain the formal interpretation of EPCIS events to align with the informal intuition given by the standard. We did not want a level of expressivity that would render reasoning undecidable. We wanted our model to capture the appropriate level of formality needed to enforce the desired consequences. Although currently EEM has been represented in the OWL DL profile, in future we plan to refine it to OWL QL/RL to facilitate querying and reasoning over large event datasets.
- *Relationship with other event models*: As EEM is domain specific, we deliberately avoid a mapping of the EEM event entity with event related entities in other models. We believe EEM addresses the need of knowledge representation for a very specific class of events. The requirements, motivation and viewpoints behind the design of EEM are therefore orthogonal to those presented by other event models.
- *Extensibility*: The EPCIS standard allows extensibility of event types and event attributes. Being an ontological model, designed with modularity as one of its inherent strengths, EEM provides the flexibility required to add new entities, attributes and relationships.

The concrete implications of the above decisions in terms of choosing an expressive profile for EEM are as follows:

- Existential property restrictions have been used extensively while defining events. The various event types have mandatory or optional requirements on the features/attributes that characterise them. As an example, an `ObjectEvent` is required to have associated EPCs, an action type and the time of event occurrence. Similarly a `QuantityEvent` is required to have an `EPCClass` associated with it. We enforce these requirements by imposing existential restrictions on event properties.
- An event occurs at a unique location, it has a unique action type and is part of a singular business process. Therefore, many event properties in EEM have been declared as functional.
- The EPCIS standard defines the informal operational semantics for the "Action" attribute. EEM captures the intuition by defining SWRL rules over event types and action attribute values.

In the following sections we discuss the core classes and properties defined for EPCIS events in EEM.

4.2 EEM Classes

`EPCISEvent` is the root or super class of all events. `ObjectEvent`, `AggregationEvent`, `QuantityEvent` and `TransactionEvent` are specialised classes of `EPCISEvent`. Figure 1 illustrates the event classes in EEM.

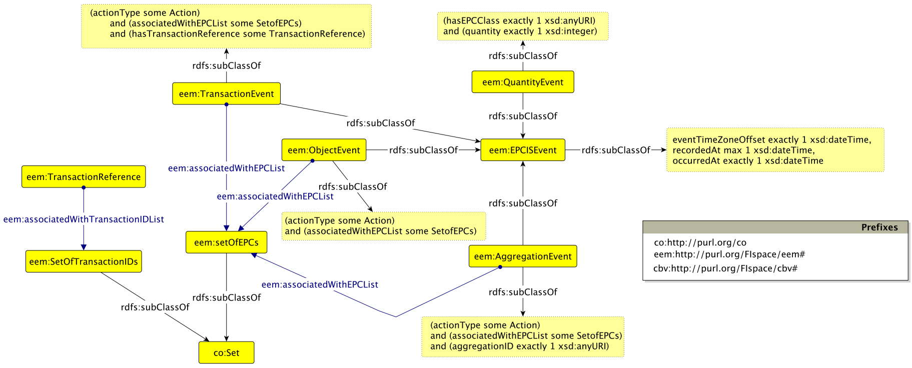

(actionType some Action)
and (associatedWithEPCList some SetofEPCs)
and (hasTransactionReference some TransactionReference)
rdfs:subClassOf
eem:TransactionEvent
(hasEPCClass exactly 1 xsd:anyURI)
and (quantity exactly 1 xsd:integer)
rdfs:subClassOf
eem:QuantityEvent
rdfs:subClassOf
rdfs:subClassOf
eem:ObjectEvent
rdfs:subClassOf
eem:EPCISEvent
rdfs:subClassOf
eventTimeZoneOffset exactly 1 xsd:dateTime,
recordedAt max 1 xsd:dateTime,
occurredAt exactly 1 xsd:dateTime
eem:TransactionReference
eem:associatedWithEPCList
eem:associatedWithEPCList
rdfs:subClassOf
eem:associatedWithTransactionIDList
(actionType some Action)
and (associatedWithEPCList some SetofEPCs)
rdfs:subClassOf
eem:setOfEPCs
eem:SetOfTransactionIDs
eem:AggregationEvent
Prefixes
co:http://purl.org/co
eem:http://purl.org/FIspace/eem#
cbv:http://purl.org/FIspace/cbv#
eem:associatedWithEPCList
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
co:Set
(actionType some Action)
and (associatedWithEPCList some SetofEPCs)
and (aggregationID exactly 1 xsd:anyURI)

Fig. 1. EPCIS event classes as represented in EEM

The class `EPC` provides a placeholder for EPCs represented using various URI schemes. The list of EPCs is represented by `SetOfEPCs`, specialising from `Set`[14].

The class `Action` denotes the activity undertaken on objects represented by `SetOfEPCs`. The set of actions [15] associated with an event are asserted with the individuals `ADD`, `OBSERVE` and `DELETE`.

The class `Transaction` encapsulates references to transactions and their types. The set of transactions associated with an event are represented by the collection class `SetOfTransactions`.

The `BusinessLocation` and `ReadPointLocation` classes capture physical location details and specialise from the `Location` class defined in the vcard [16] vocabulary. The `EPC Reader` class represents readers with physical and logical identifiers.

A companion standard to the EPCIS standard is the Core Business Vocabulary(CBV)[17] standard. The CBV standard supplements the EPCIS framework by defining vocabularies and specific data values that may populate the EPCIS data model. We provide ontological representation [18] of the vocabulary definitions as individual assertions to be used along with the EEM model.

[14] `http://purl.org/co/`. We specialise from a `Set` rather than a `List` to avoid duplicates

[15] The interested reader is referred to the EPCIS standard for details.

[16] `http://www.w3.org/2006/vcard/ns#`

[17] http://www.gs1.org/gsmp/kc/epcglobal/cbv

[18] `http://fispace.aston.ac.uk/ontologies/cbv#`

As an exemplar, the formal definition of the EPCISEvent ObjectEvent and QuantityEvent classes in EEM are presented below in the OWL Manchester syntax:

```
Class: EPCISEvent
    SubClassOf:
        eventTimeZoneOffset exactly 1 xsd:dateTime,
        recordedAt max 1 xsd:dateTime,
        occurredAt exactly 1 xsd:dateTime

Class: ObjectEvent
    SubClassOf:
        (actionType some Action)
         and (associatedWithEPCList some SetofEPCs),
        EPCISEvent

Class: QuantityEvent
    SubClassOf:
        (hasEPCClass exactly 1 xsd:anyURI)
         and (quantity exactly 1 xsd:integer),
        EPCISEvent
```

4.3 Properties

EEM defines several kinds of properties for events, in order to capture relationships between entities based on the four information dimensions.

Event specific properties EEM defines properties relating events to their business context. While many properties are common among the four event types, some are specific to certain events. For example, the hasAggregationID property is defined only for the AggregationEvent. The hasEPCClass and quantity properties have QuantityEvent as their domain. While hasTransactionReference is required to be asserted for a TransactionEvent, it is optional for the other event types.

Besides the implicit relationships described in the EPCIS specification, EEM defines a datatype property eventID. A systematic identification system assigns every event a unique eventID. This can then be used to construct URIs for events in order to publish event data as linked data and link event data with master data.

Temporal Properties An EPCIS event is associated with three types of timing properties: eventOccurredAtTime signifies the date and time at which the EPCIS capturing applications asserts the event occurred, eventRecordedAtTime captures the date and time at which this event was recorded by an EPCIS Repository (optional). Additional business context is provided through the property eventTimeZoneOffset, the time zone offset in effect at the time and place the event occurred.

Location properties The hasBusinessLocation and hasReadPointLocation object properties connect the business and read point locations respectively to an event. A business location or a read point itself is identified using the hasLocationID data type property with the property range being xsd:anyURI.

Business context properties The `BusinessStep` and `Disposition` entities relate to an event through the `hasBusinessStepID` and `hasDispositionType` property respectively. Individual assertions for these entities are provided in the CBV ontology and are used to populate the range values for the properties. Every `Transaction` entity is related to a `TransactionType` entity through the `hasTransactionType` relationship. Values for transaction types are provided by the CBV standard and asserted in the CBV ontology. Figure 2 provides an illustration of the entities, relationship and some representative individuals for the entities.

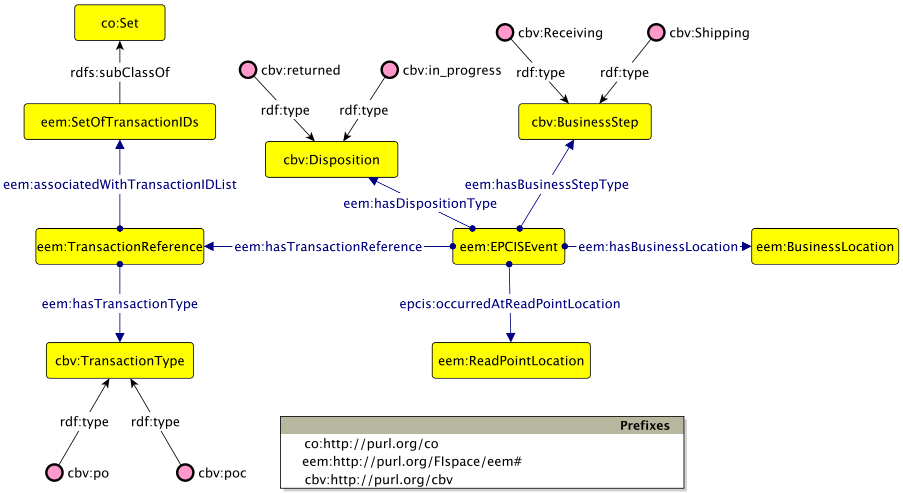


Fig. 2. Business context entities, relationships and representative individuals

4.4 Modelling the "Action" attribute

The "Action" field for an object, aggregation and transaction event occurring on an EPC tagged object or set of objects, indicates the activity that has taken place on the object(s) during the business step that generated the event. EEM declares a class entity `Action` with three class assertions: `ADD`, `OBSERVE` and `DELETE` corresponding to the values the action field can take. The `hasActionType` object property relates an event to the action type and ranges over `Action`.

For an object event the informal semantics of the action type "Add" implies that the EPC(s) named in the event have been commissioned as part of this event. We formalise this informal intuition using a SWRL rule as illustrated below:

$$\boxed{\begin{array}{l} ObjectEvent(?\,e), actionType(?\,e, ADD), \\ \quad associatedWithEPCList(?\,e, ?\,list), \\ \qquad hasBusinessStepType(?\,e, commissioning) \rightarrow commissioned(?\,e, ?\,list) \end{array}}$$

Analogous to the above, rules can be defined for aggregation and transaction events for the action types, “ADD” and “DELETE”.

5 Implementing EEM

EEM is a complex data model. It is non trivial for a user to generate class assertions and complex queries without knowing the structure of the model and nomenclature of the entities. In order to encourage the uptake of EEM among EPCIS conforming organisations and industries, ease the creation of EEM instances and facilitate querying over the instantiated datasets, we present an open source API - LinkedEPCIS[19]. The purpose of the API is to conveniently incorporate EEM in EPCIS capture and query applications.

LinkedEPCIS is a Java library for capturing EPCIS events as linked data. It has been built over the Sesame framework[20]. Every event generated using LinkedEPCIS, is systematically assigned a HTTP URI. The library provides classes, interfaces and RESTful Web services for capturing EPCIS events as linked data and curating the datasets in triple stores. Query classes encoding templated SPARQL queries for the most commonly made queries on EPCIS events are provided. Results are made available in RDF/XML, JSON and Turtle serialisations.

The most significant classes in the LinkedEPCIS library are `EPCISEvent` and `EPCISCommon`. `EPCISEvent` encapsulates the attributes and operations common to all EPCIS event types. `EPCISCommon` provides a set of operations for the internal generation and manipulation of the linked data model.

Central to the data model generated through the LinkedEPCIS library is the `Graph` interface from the Sesame API. LinkedEPCIS records data about events as triples/statements and attaches them to a `Graph`, which can be persisted as a file or dumped in a dedicated EPCIS events triple store. Besides the attributes for events predefined in the EPCIS specification, extensions are supported by retrieving the current `Graph` and attaching new triples.

EPCIS event data conforming to the EEM model can be integrated with several other linked data sources using the LinkedEPCIS library. Figure 3 illustrates some examples of such integration. EPCs defined in an EPCIS event can be linked to the product master data. Location based information from DBpedia and Geonames can be used to enrich the location attributes for read point and business locations of an EPCIS event. Finally events can be linked to party/company master data through their FOAF profiles.

6 EPCIS events in the tomato supply chain

As an exemplifier for EEM and the LinkedEPCIS library, we consider EPCIS events arising as part of the agri-food supply chain. In particular, we consider supply chains for perishable goods, e.g., tomatoes. The tomato supply chain involves thousands of farmers, hundreds of traders and few retail groups, with information infrastructure in place to record data about agricultural goods, shipments, assets and cargo.

[19] `http://code.google.com/p/linked-epcis/`

[20] http://openrdf.org

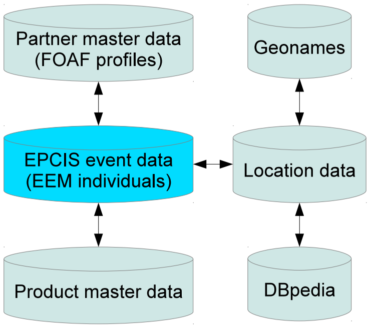
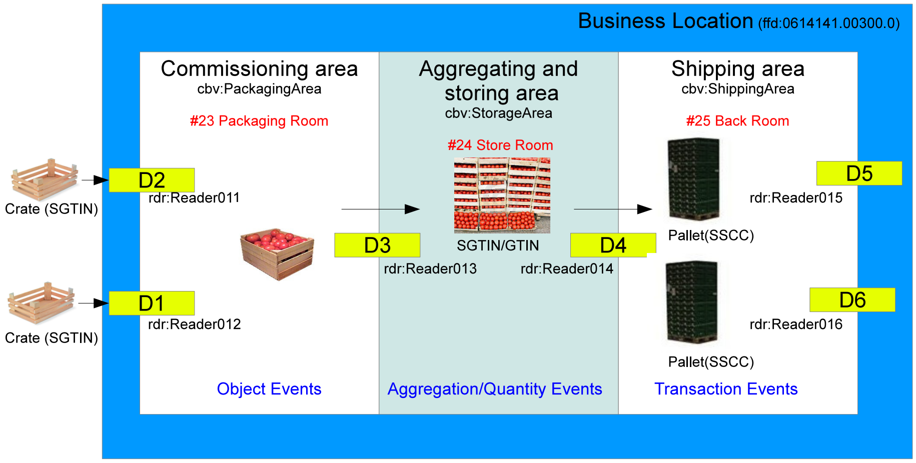


Fig. 3. Interlinking EPCIS event data

Franz is a farmer who specialises in growing tomatoes. The tomatoes are packaged and shipped to downstream traders. The packaging of tomatoes is done in crates, each of which is tagged with an RFID. Sensors installed at Franz farmer's packaging unit register the EPCs of the crates as they are being packed. Every read of the RFID tagged crate by the sensor is recorded and curated as an EPCIS event type based on the business process, the location and the supply chain operation at the point of event occurrence. A partial workflow along with possible sensor locations at Franz farmer's packaging unit is illustrated in Figure 4. Table 1 presents a subset of the EPCIS events captured in the supply chain phases.



Readers installed at Docks: D1, D2, D3, D4, D5, D6

@prefix ffd: http://data.franzfarmer.com/epcis/locations/sgln/
@prefix rdr: http://data.franzfarmer.com/epcis/reader/
@prefix cbv: http://purl.org/FIspace/cbv#

Fig. 4. EPCIS events, sensor installations and workflow

	Supply chain operation	EPCIS event type	Business Step	Disposition	Action type
1.	Commissioning crates for tomatoes	Object event	commissioning	active	ADD
2.	Storing crates	Quantity event	storing	in_progress	-
3.	Aggregating crates in pallets	Aggregation event	packing	in_progress	ADD
4.	Loading and shipping pallets	Transaction event	shipping	in_transit	ADD

Table 1. Subset of EPCIS events

Figure 5 illustrates an Object event captured at the EPCIS implementation deployed at Franz farmer's packaging utility and expressed using EEM. The Object event relates to the commissioning of crates for tomatoes.

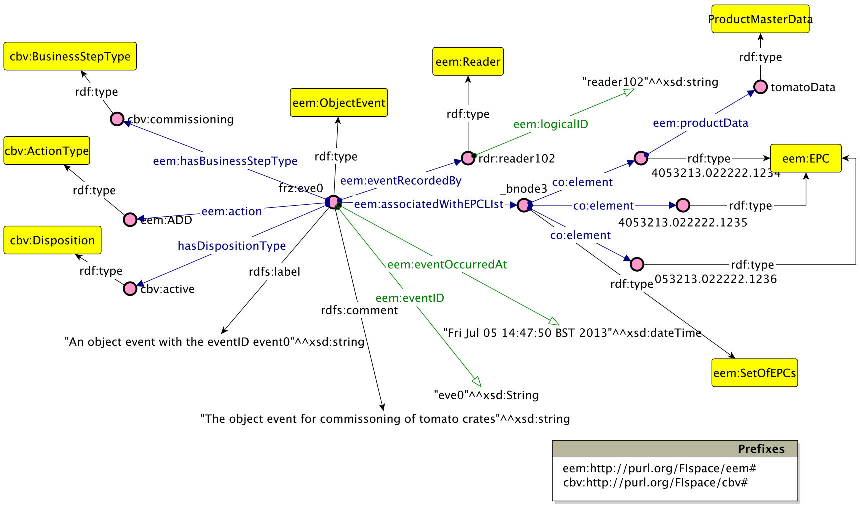


Fig. 5. An EEM object event representation from the tomato supply chain

7 Conclusions

The representation of EPCIS events on the Web of data is an important step towards achieving the objectives of sharing traceability information between trading partners and detecting inconsistencies in supply chains on a Web scale. In this paper we have proposed EEM: The EPCIS Event Model that provides the ontological primitives required to represent EPCIS events using Semantic Web standards. EEM is an OWL DL ontology and builds on foundational modelling decisions based on our requirements analysis of the supply chain sector. The capture and curation of EPCIS events linked datasets is realised using the LinkedEPCIS library implemented by us, which can be integrated with existing RFID and EPCIS implementations. We have exemplified the use of the EEM model and LinkedEPCIS library by modelling and curating events from the agri-food supply chain.

As part of our future work, we are looking into refining the EEM model to the OWL QL/RL profile in order to facilitate querying and reasoning. We have developed bespoke SWRL rules over EPC lists, actions and events, in order to materialise intuitive predicates which are currently not a part of the EPCIS specification. These will soon be implemented and integrated within the LinkedEPCIS library.

Acknowledgements

The research described in this paper has been partially supported by the EU FP7 FI PPP projects, SmartAgriFood (`http://smartagrifood.eu`) and FISpace `http://www.fispace.eu/`

References

1. A. Franois, R. Nevatia, J. Hobbs, R. Bolles, and J. Smith. VERL: an ontology framework for representing and annotating video events. *MultiMedia, IEEE*, 12(4):76–86, 2005.
2. E. Hyvönen. *Publishing and Using Cultural Heritage Linked Data on the Semantic Web*. Morgan and Claypool Publishers, 2012.
3. E. Hyvönen, T. Lindquist, J. Törnroos, and E. Mäkelä. History on the semantic web as linked data – an event gazetteer and timeline for world war i. In *Proceeedings of CIDOC 2012 - Enriching Cultural Heritage, Helsinki, Finland.* CIDOC, http://www.cidoc2012.fi/en/cidoc2012/programme, June 2012.
4. A. Ilic, T. Andersen, and F. Michahelles. Increasing supply-chain visibility with rule-based rfid data analysis. *IEEE Internet Computing*, 13(1):31–38, Jan. 2009.
5. K. Janowicz, S. Scheider, T. Pehle, and G. Hart. Geospatial semantics and linked spatiotemporal data - past, present, and future. *Semantic Web*, 3(4):321–332, 2012.
6. T. Nguyen, Y.-K. Lee, B.-S. Jeong, and S. Lee. Event query processing in epc information services. In *Proceedings of the 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System*, SITIS '07, pages 159–166, Washington, DC, USA, 2007. IEEE Computer Society.
7. A. Scherp, T. Franz, C. Saathoff, and S. Staab. F–a model of events based on the foundational ontology DOLCE+DnS ultralight. In *Proceedings of the fifth international conference on Knowledge capture*, K-CAP '09, pages 137–144, New York, NY, USA, 2009. ACM.
8. R. Shaw, R. Troncy, and L. Hardman. LODE: Linking Open Descriptions of Events. In *Proceedings of the 4th Asian Conference on The Semantic Web*, ASWC '09, pages 153–167, Berlin, Heidelberg, 2009. Springer-Verlag.
9. S. Turchi, L. Ciofi, F. Paganelli, F. Pirri, and D. Giuli. Designing epcis through linked data and rest principles. In *Software, Telecommunications and Computer Networks (SoftCOM), 2012 20th International Conference on*, pages 1–6, 2012.
10. W. R. van Hage, V. Malais, R. Segers, L. Hollink, and G. Schreiber. Design and use of the Simple Event Model (SEM). *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(2):128 – 136, 2011.

Extractivism

Extracting activist events from news articles using existing NLP tools and services

Thomas Ploeger[1], Maxine Kruijt[2], Lora Aroyo[1], Frank de Bakker[2], Iina Hellsten[2], Antske Fokkens[3], Jesper Hoeksema[1], and Serge ter Braake[4]

[1] Computer Science Department VU University Amsterdam
[2] Organization Sciences Department VU University Amsterdam
[3] Language and Communication Department VU University Amsterdam
[4] History Department VU University Amsterdam

Abstract. Activists have a significant role in shaping social views and opinions. Social scientists study the events activists are involved in order to find out how activists shape our views. Unfortunately, individual sources may present incomplete, incorrect, or biased event descriptions. We present a method where we automatically extract event mentions from different news sources that could complement, contradict, or verify each other. The method makes use of off-the-shelf NLP tools. It is therefore easy to setup and can also be applied to extract events that are not related to activism.

1 Introduction

The goal of an activist is to effect change in societal norms and standards [8]. Activists can thus both make an impact on the present and play a significant role in shaping the future. Considering the events activists are engaged in allows us to see current controversial issues, and gives social scientists the means to identify the (series of) methods through which activists are trying to achieve change in society.

The MONA[5] project is an interdisciplinary social/computer science effort which aims at producing a visual analytics suite for efficiently making sense of large amounts of activist events. Specifically, we intend to enable the discovery of event activity patterns that are 'hidden' in human-readable text, as well as provide detailed analyses of these patterns. The project currently focuses on activist organizations that have recently been protesting against petroleum exploration in the Arctic.

Social scientist are interested in finding out which activist organizations are trying to influence the oil giants, and specifically which events they are organizing to do so. This could be addressed by aggregating events that took place in this context, enabling a quantitative (e.g. "What is the common type of event

[5] Mapping Online Networks of Activism

organized?") as well as a qualitative (e.g. "Why are these types of events organized?") analysis.

In an earlier paper [12], we described initial work in the MONA project. This work primarily concerned the evolutionary explorations we performed in the activist use case to make our event modeling requirements concrete. These explorations led to the decision to use the Simple Event Model (SEM) [7], which models events as "who did what to whom, where and when". In addition, we considered how visualizations of event data could aid an end-user in answering specific types of questions about aggregated activist events.

This paper describes our approach for event extraction from human-readable text so we can aggregate them and 'feed' them to a visualization suite. Our approach repurposes off-the-shelf natural language processing software and services (primarily named entity recognizers and disambiguators) to automatically extract events from news articles with a minimal amount of domain-specific tuning. As such, the method described in this paper goes beyond the domain of activism and can be used to extract events related to other topics as well.

The output of our system are representations in the Grounded Annotation Framework (GAF) [6], which links representations in SEM to the text and linguistic analyses they are derived from. A more detailed description of GAF will be given in Section 3.2.

We use news articles because they are available from a huge variety of sources and in increasingly large numbers. Being able to tap into such a large and diverse source of event descriptions is extremely valuable in event-based research, because individual event descriptions may be incomplete, incorrect, out of context, or biased. These problems could be alleviated by using multiple sources and increasing the number of descriptions considered: Events extracted from multiple articles could complement each other in terms of completeness, serve as verification of correctness, place events in a larger context, and present multiple perspectives.

We consider both quantitative measurements and the usefulness of the extracted events in our evaluation. We quantitatively evaluate performance by calculating the traditional information retrieval metrics of precision, recall, and F1 for the recognition of events and their properties. Through examples, we give a tentative impression of the usefulness of the aggregated event data.

The rest of this paper is structured as follows. In Section 2, we give an overview of previous work in event extraction and how it relates to this work. The representation frameworks we use are explained in Section 3. In Section 5, we outline our methodology. We show how we model events, how events are typically described in text and how we use existing NLP software and services to extract them. Section 5 contains an overview of the results. We present both a quantitative evaluation as well as a detailed error analysis of the performance of our event extraction method. We go beyond performance numbers in Section 6 by discussing the usability and value of our contribution leading us to the direction future work should take.

2 Related work

In this section, we demonstrate the heterogeneous nature of the field of event extraction by giving a non-exhaustive overview of contemporary approaches from several domains. The diversity in event representations and extraction methods makes it inappropriate to make direct comparisons (e.g. in terms of performance) between our work and that of others, but we can still show how work in other domains relates to our own work.

In molecular biology, gene and protein interactions are described in human-readable language in scientific papers. Researchers have been working on methods for extracting and aggregating these events to help understand the large numbers of interactions that are published. For example, Björne [2] demonstrated a modular event extraction pipeline that uses domain-specific modules (such as a biomedical named entity recognizer) as well as general purpose NLP modules to extracted a predefined set of interaction events from a corpus of PubMed papers.

The European border security agency Frontex uses an event extraction system [1] to extract events related to border security from online news articles. Online news articles are used because they are published quickly, have information that might not be available from other sources, and facilitate cross-checking of information. This makes them valuable resources in the real-time monitoring of illegal migration and cross-border crime. The system developed for Frontex uses a combination of traditional NLP tools and pattern matching algorithms to extract a limited set of border security events such as illegal migration, smuggling, and human trafficking.

Van Oorschot et al. [11] extract game events (e.g. goals, fouls) from tweets about football matches to automatically generate match summaries. Events were detected by considering increases in tweet volume over time. The events in those tweets were classified using a machine learning approach, using the presence of certain words, hyperlinks, and user mentions as features. There is a limited set of events that can occur during a football match, so there is a pre-defined, exhaustive list of events to extract. These events have two attributes: The time at which they occurred and the football team that was responsible.

The recurring theme in event extraction across different domains is the desire to extract events from human-readable text (as opposed to structured data) to aggregate them, enabling quantitative and qualitative analysis. Our research has the same intentions, but the domain-specific nature of event representations and extraction methods in the current event extraction literature limits the reuse of methods across domains and (to our knowledge) there has been no research into extracting events for the purpose of studying activists.

Specifically, the existing work on event extraction is typically able to take advantage of an exhaustive lists of well-defined events created a priori. In our case, we cannot make any assumptions about which types of events are relevant to the end user because we intend to facilitate discovery of new event patterns, which necessitates a minimally constrained definition of ‘event’.

Ritter et al. [13] present an open-domain approach to extract events from twitter. They use supervised and semi-supervised machine learning training a model on 1,000 annotated tweets. Due to the difference in structure and language use, this corpus is not suitable for extracting events from newspaper text. Moreover, tweets will generally address only one event whereas newspaper articles can also be stories that involve sequences of events. This makes our task rather different from the one addressed in [13].

The goal of our research was to create an approach that can identify events in newspaper text while exclusively making use of off-the-shelf NLP tools. We do not make use of a predefined list of potentially interesting events like most of the approaches mentioned above. Our approach differs from Ritter et al.'s work, because there is no need to annotate events in text for training. Our approach, which will be described in the following section, can be applied for event extraction in any domain.

3 Event Representation

In this section, we describe the representations we use as output of our system. We first outline the Simple Event Model in Section 3.1. This is followed by an explanation of the Grounded Annotation Framework (GAF) [6] which forms the overall output of our extraction system in Section 3.2.

3.1 The Simple Event Model

We use the Simple Event Model (SEM) to represent events. SEM uses a graph model defined using the Resource Description Framework Schema language (RDFS) and the Web Ontology Language (OWL). SEM is designed around the following definition of *event*. "Events [..] encompass everything that happens, even fictional events. Whether there is a specic place or time or whether these are known is optional. It does not matter whether there are specic actors involved. Neither does it matter whether there is consensus about the characteristics of the event." This definition leads to a more formal specification in the form of an event ontology which models events as having actors, places and times(tamps). Each of these classes may have a type, which may be specified by a foreign type system. A unique feature of SEM is that it allows specifying multiple views on a certain event, which hold according to a certain authority. A basic example of an instantiated SEM-event can be seen in Figure 1.

3.2 The Grounded Annotation Framework

In addition to SEM, we use the Grounded Annotation Framework (GAF). The basic idea behind this framework is that it links semantic representations to *mentions* of these representations in text and semantic relations to the syntactic relations they are derived from. This provides a straight-forward way to mark the **provenance** of information using the PROV-O [10]. When presenting multiple

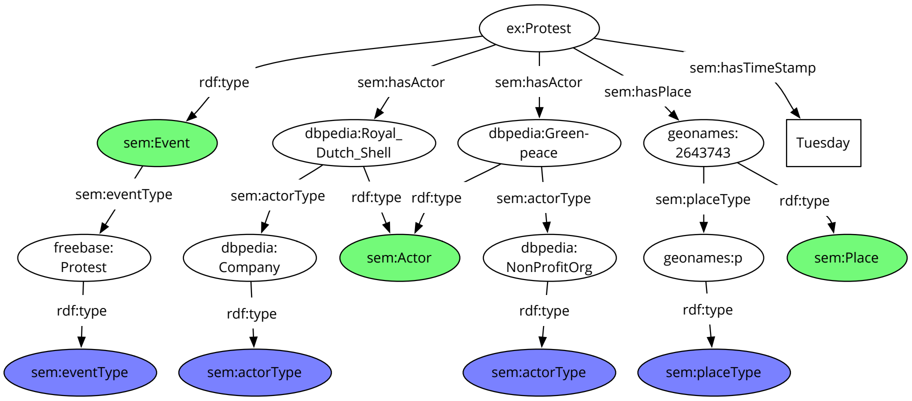


Fig. 1. Example of a SEM-event that might be instantiated for the event: "Tuesday, Greenpeace protested against Shell in London"

views next to each other, it is important to know where these views come from. Furthermore, Natural Language Processing techniques do not yield perfect results. It is thus essential that social scientists can easily verify whether extracted information was indeed expressed in the original source. Finally, insight into the derivation process can be valuable for system designers as they aim to improve their results.

4 Method

As establised in the previous section, we consider everything that *happens* an event. An event may have actors involved, a certain location, and occurs at a point in time. We use a rapidly prototyped event extraction tool which integrates several generic, off-the-shelf natural language processing software packages and Web services in a pipeline to extract this information. This section describes this pipeline which is illustrated in Figure 2.

Preprocessing & Metadata extraction The pipeline takes a news article's URL as input, with which we download the article's raw HTML. We use the Nokogiri[6] XML-parser to find time and meta tags in the HTML. These tags typically contain the article's publication date, which we need later for date normalization. Next, we use AlchemyAPI's[7] author extraction service on the raw HTML to identify the article's author, which enables us to attribute the extracted events. We then run the HTML through AlchemyAPI's text extraction service to strip any irrelevant content from the HTML, giving us just the text of the article.

[6] http://nokogiri.org/

[7] http://www.alchemyapi.com/

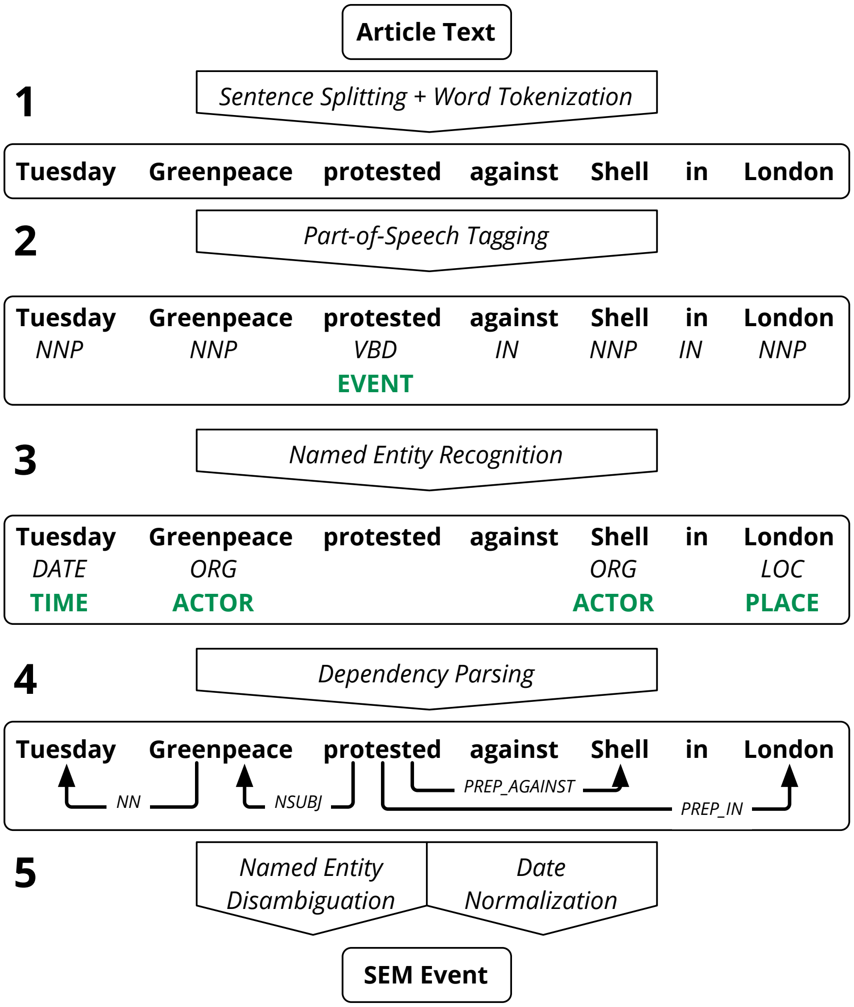


Fig. 2. Event extraction pipeline.

Processing The article's text is split into sentences and words using Stanford's sentence splitter and word tokenizer[8]. We consider each verb of the sentence to be an event, because verbs convey actions, occurrences, and states of being. This is a very greedy approach, but this is necessarily so: We do not wish to make any a priori assumptions about which types of events are relevant to the end user. We use Stanford's part-of-speech tagger [14] to spot the verbs.

Actors and places are discovered using Stanford's named entity recognizer [5]. The type (e.g. person, organization, location) of the named entity determines whether it is an Actor or a Place. Dates and times are also identified by the named entity recognizer.

The mere existence of named entities, a timestamp, and a verb in the same sentence does not immediately mean that they together form one event. One sentence may describe multiple events or a place might be mentioned without it being the direct location of the event. Therefore we only consider named entities and timestamps grammatically dependent on a specific event to be part of that event. For this we use Stanford's dependency parser [9].

Normalization & Disambiguation Using Stanford's SUTime [3], We normalize any relative timestamps (e.g. "Last Tuesday") to the publication date to transform them into full dates (e.g. "23-06-2013"). We complement Stanford's named entity recognizer with TextRazor's[9] API to disambiguate found named entities to a single canonical entity in an external data source such as DBpedia.

Storage & Export The output of the preprocessing, metadata extraction, processing, normalization, and disambiguation steps is stored in a Neo4j[10] graph database. For each article, we create a document node with metadata properties, such as the URL, author, and publication date. The document node has sentence nodes as its children, which in turn have word nodes as their children. The word nodes have the properties that were identified earlier in the pipeline, such as their part-of-speech tags, named entity tags, etc. The grammatical dependencies between words are expressed as typed edges between word nodes. We traverse the resulting graph to identify verbs with dependent named entities and timestamps. We export the event as a SEM event together with provenance in GAF.

Implementation details All of the software packages and services above are integrated using several custom Ruby scripts. We have also used several existing Ruby gems for various supporting tasks: A Ruby wrapper[11] for Stanford's NLP tools, HTTParty[12] for Web API wrappers, Chronic[13] for date parsing, and Neography[14] for interacting with Neo4j.

[8] nlp.stanford.edu/software/tokenizer.shtml

[9] http://www.textrazor.com/

[10] http://www.neo4j.org/

[11] http://github.com/louismullie/stanford-core-nlp

[12] http://github.com/jnunemaker/httparty

[13] http://github.com/mojombo/chronic

[14] http://github.com/maxdemarzi/neography

5 Evaluation

Before we present the results of our method of event extraction in Section 5.2, we describe the corpus we used for evaluation and the creation of a gold standard in Section 5.1. In Section 5.3, we describe the major issues impacting the performance of our method.

5.1 Experimental Setup

We extracted events from a corpus of 45 documents concerning arctic oil exploration activism. 15 of these documents are blog posts, the other 30 are news articles. The majority of articles are from The New York Times[15] (70%) and the Guardian[16] (15%), the rest from similar news websites.

Three domain experts manually annotated every article (each annotator individually annotated 1/3 of the corpus) to create a gold standard for evaluation. The experts were asked to annotate the articles with events, actors, places, and times and then link the actors, places, and times to the appropriate events, in such a way that the resulting events would be useful for them if aggregated and visualized. No further explicit instructions were given to the annotators. The Brat rapid annotation tool[17] was used by the experts for annotation.

Table 1 illustrates the inter-rater agreement of the annotators on a subset of the corpus that was annotated by each annotator. For each type of annotation we show the percentage of annotations that were annotated by only 1 of the annotators, by 2 of the annotators, or by all 3 annotators. For each class the majority of annotations are shared by at least 2 annotators. Events have the largest amount of single-annotator annotations, showing that inter-rater consensus is lowest for this concept.

# Annotators	Event	Actor	Place	Time
1	46%	35%	36%	28%
2	34%	32%	28%	43%
3	20%	33%	36%	29%

Table 1. Percentage of annotations that were annotated by only 1 of the annotators, 2 of the annotators, or all 3 annotators.

5.2 Results

The second and third columns of Table 2 show the amounts of events, actors, places, and times in the gold standard and the amounts extracted from the corpus. The next 3 columns show the true positives, false positives, and false

[15] http://www.nytimes.com/

[16] http://www.guardian.co.uk/

[17] http://brat.nlplab.org/

negatives. The final 3 columns show the resulting precision, recall, and F1 per class.

For each of the 1299 events correctly recognized, we checked if they were associated with the correct actors, places, and times. Table 3 shows the mean precision, recall, and F1 scores for the linking of events to the appropriate actors, places, and times.

Class	Gold	Extracted	True Pos	False Pos	False Neg	Precision	Recall	F1
Event	2241	1829	1299	530	942	0,71	0,58	0,64
Actor	2130	1609	748	861	1382	0,46	0,35	0,40
Place	508	772	276	496	232	0,36	0,54	0,43
Time	498	456	298	158	200	0,65	0,60	0,62

Table 2. Corpus-wide counts and performance metrics per class.

Link	Precision	Recall	F1
Event-Actor	0,27	0,4	0,3
Event-Place	0,2	0,2	0,2
Event-Time	0,27	0,27	0,27

Table 3. Mean precision, recall, and F1 for the linking of correctly recognized events to their actors, places, and times.

5.3 Discussion

We carried out an error analysis for each class and identified several issues that bring down performance of our system. This section describes these errors and indicates how we may improve our system in future work.

Actors masquerading as places (and vice versa) In the sentence "Shell is working with wary United States regulators.", our annotators are interested in the United States as an actor, not a location. Still, it is recognized as a location by the named entity recognizer. This is a contributor to the large number of false negatives (and false positives) for actors and places. The grammatical dependency between the verb and a named entity could give us some clues to the role an entity plays in an event. In the example, the kind of preposition ("with") makes it clear that *United States* indicates an actor, not a place.

Ambiguous times The named entity recognizer only identifies expressions that contain specific time indications as times. Relative timestamps such as "last

Tuesday" or "next winter" are resolvable by the extraction pipeline, but more ambiguous times such as "after" or "previously" and conditional times such as "if" and "when" are not detected. This contributes to the false negatives for timestamps and could be solved by hand-coding a list of such temporal expressions into the extraction process.

Unnamed actors & places The pipeline only recognizes named entities as actors and places, so any common nouns or pronouns that indicate actors are not recognized by the pipeline. This issue could be solved by relaxing the restriction that only named entities are considered for actors and places. Similar to the actors masquerading as places, looking at the grammatical dependencies could indicate whether we are dealing with an actor or a place. This may however increase the number of false positives because of the ambiguous nature of some grammatical dependencies (e.g. "about"). We propose two tactics to address this issue: coreference resolution and linking noun phrases to ontologies.

Consider the following 2 sentences: "The Kulluk Oil Rig was used for test drilling last summer. The Coast Guard flew over the rig for a visual inspection." A coreference resolver in the pipeline could indicate that "the rig" in the second sentence is a coreferent of a named entity and may thus be considered a location. Sometimes, actors or places do not refer to a specific person or location (e.g. "scientists", "an area") in which case they will not corefer to a named entity. If we link noun phrases to an ontology such as WordNet [4], we can identify whether they refer to a potential agent or location by inspecting their hyponyms. Because nouns can also refer to events (e.g. "strike"), this may also increase recall on event detection.

Gold Standard Annotations The percentages of inter-rater agreement (as shown earlier in Table 3), especially for events, indicate that the gold standard could benefit from a more rigorous annotation task description. We realize that if the task is loosely defined, human annotators may have different interpretations of what an 'event' is in natural language.

For this reason, it is interesting to compare the tool output to the three annotators individually. Table 4 shows the pipeline's F1-scores per class per individual annotator. The scores for annotator 1 and 3 are very close for all four classes. Annotator 2 differs significantly for places and times. This demonstrates the variance that annotators with different interpretations of the annotation task introduce to performance scores of the tool.

Annotator	Event	Actor	Place	Time
1	0.63	0.41	0.49	0.57
2	0.54	0.44	0.19	0.35
3	0.60	0.43	0.48	0.61

Table 4. F1-scores per class for each annotator individually.

6 Conclusion

In this paper we reported on the development and performance of our extraction method for activist events: A pipeline of existing NLP software and services with minimal domain-specific tuning. The greatest value of this contribution is the fact that it will enable further work in the MONA project. The goal of the project is to produce a visual analytics suite for efficiently making sense of large amounts of activist events. Through these visual analytics, we intend to enable the discovery and detailed analysis of patterns in event data. The extraction pipeline described in this paper (and any future revisions of it) will be able to feed our visual analytics suite with event data.

Work is already underway on the development of the visual analytics suite and details will be available in a forthcoming paper. The effectiveness of the visual analytics will be dependent on the quality of the event data our extraction pipeline produces. We already have candidate solutions for issues that negatively impact the pipeline's performance. In future work we will implement these solutions and report on their effectiveness. In the meantime, we can already get a tentative impression of the value the extracted event data has, for both discovery and more detailed analysis.

Aggregating and counting event types that a certain actor is involved in enables the discovery of the primary role of actors. Similarly, by aggregating and counting the places of events we can discover the geographical areas an actor has been active in. Filtering the events by time can give us insight into changes in active areas over time. Because we have extracted events from multiple sources, events can complement each other in terms of completeness, serve as verification of correctness, place events in a larger context, and present multiple perspectives. In future work, we intend to define measurements for these concepts (e.g. when are events complementary, when do they verify each other) in order to quantify them.

Acknowledgements

This research is partially funded by the Royal Netherlands Academy of Arts and Sciences in the context of the Network Institute research collaboration between Computer Science and other departments at VU University Amsterdam, and partially by the EU FP7-ICT-2011-8 project NewsReader (316404). We would like to thank Chun Fei Lung, Willem van Hage, and Marieke van Erp for their contributions and advice.

References

1. Atkinson, M., Piskorski, J., Goot, E., Yangarber, R.: Multilingual real-time event extraction for border security intelligence gathering. In: Wiil, U.K. (ed.) Counterterrorism and Open Source Intelligence, Lecture Notes in Social Networks, vol. 2, pp. 355–390. Springer Vienna (2011)

2. Björne, J., Van Landeghem, S., Pyysalo, S., Ohta, T., Ginter, F., Van de Peer, Y., Ananiadou, S., Salakoski, T.: Pubmed-scale event extraction for post-translational modifications, epigenetics and protein structural relations. In: Proceedings of BioNLP 2012. pp. 82–90 (2012)
3. Chang, A.X., Manning, C.: Sutime: A library for recognizing and normalizing time expressions. In: et al., N.C. (ed.) Proceedings of LREC 2012. ELRA, Istanbul, Turkey (may 2012)
4. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA (1998)
5. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd ACL. pp. 363–370. ACL '05, ACL, Stroudsburg, PA, USA (2005)
6. Fokkens, A., van Erp, M., Vossen, P., Tonelli, S., van Hage, W.R., Serafini, L., Sprugnoli, R., Hoeksema, J.: GAF: A grounded annotation framework for events. In: Proceedings of the first Workshop on Events: Definition, Dectection, Coreference and Representation. Atlanta, USA (2013)
7. van Hage, W.R., Malaisé, V., van Erp, M., Schreiber, G.: Linked Open Piracy. In: Proceedings of the sixth international conference on Knowledge capture. pp. 167–168. ACM, New York, NY, USA (June 2011)
8. den Hond, F., de Bakker, F.G.A.: Ideologically motivated activism: How activist groups influence corporate social change activities. Academy of Management Review 32(3), 901–924 (2007)
9. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st ACL. pp. 423–430. ACL '03, ACL, Stroudsburg, PA, USA (2003)
10. Moreau, L., Missier, P., Belhajjame, K., B'Far, R., Cheney, J., Coppens, S., Cresswell, S., Gil, Y., Groth, P., Klyne, G., Lebo, T., McCusker, J., Miles, S., Myers, J., Sahoo, S., Tilmes, C.: PROV-DM: The PROV Data Model. Tech. rep., W3C (2012)
11. van Oorschot, G., van Erp, M., Dijkshoorn, C.: Automatic extraction of soccer game events from twitter. In: Proceedings of the Workhop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2012). pp. 21–30 (2012)
12. Ploeger, T., Armenta, B., Aroyo, L., de Bakker, F., Hellsten, I.: Making sense of the arab revolution and occupy: Visual analytics to understand events. In: Proceedings of the Workhop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2012). pp. 61–70 (2012)
13. Ritter, A., Etzioni, O., Clark, S., et al.: Open domain event extraction from twitter. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1104–1112. ACM (2012)
14. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the NAACL and HLT 2003. pp. 173–180. NAACL '03, ACL, Stroudsburg, PA, USA (2003)

A case study on automated risk assessment of ships using newspaper-based event extraction

Jesper Hoeksema[1] and Willem Robert van Hage[2]

[1] Computer Science, Network Institute
VU University Amsterdam
De Boelelaan 1081a, 1108 HV
Amsterdam, The Netherlands
J.E.Hoeksema@vu.nl
[2] SynerScope B.V.
willem.van.hage@synerscope.com

Abstract. In this paper we describe an event-type extractor on top of a distributed search engine. We apply this event-type extractor in a case study concerned with assisting maritime security operators to assess potential risk factors of ships. Based on a corpus of maritime-related press releases we automatically investigate the history of ships as they enter an area of interest. The performance of the system is evaluated with a task-oriented focus on a set of vessels with known risk factors, and typical behaviour is evaluated by batch-processing a large set of vessels.

1 Introduction

In many safety and security-related tasks it is necessary to quickly investigate the background of an object under surveillance in order to see if its history raises any red flags. In this paper we analyse how a combination of techniques from event extraction, information retrieval, text processing and background knowledge can be used to support this task. Our aplication domain is maritime safety and security in the Dutch coastal area.

On average every thirty seconds a vessel leaves or enters the Netherlands Exclusive Economic Zone, an area of 154.011 km^2 in front of the Dutch Coast.[3] The Dutch coastguard employs 51 full-time operators who continuously monitor this area (which typically contains at any point in time around 1300 to 1400 ships) in order to predict and hopefully prevent events that threaten the law, the environment, or public safety. The current generation of naval vessel observation systems process most information retrieved from readily available sources automatically, such as vessel positions from radar and information broadcasts by the vessels themselves, and project this information on a map view to the operator. These systems, however, do not take any information from outside sources into account, such as news articles and other public information. If an operator wants to know more about a certain vessel, he or she has to search for this information manually, often on a second computer. This means that an operator is not able to fully investigate all vessels in the area of interest, as the number of ships coming and going is too large to process manually in (near) real-time. Currently, operators circumvent this problem by prioritizing the ships to investigate, using data that is immediately accessible, such as their previous port of call, bearing, name and cargo, and only further investigating the vessels with the highest priorities.

As this process of elimination is inherently incomplete due to the fact that not all available information is taken into account, potential threats could possibly slip through. We propose a full prioritization by automating parts of the initial investigation using a combination of techniques that tie into the current state-of-the-art vessel observation systems. The focus of our research is to explore the possibilities of using a combination of relevance feedback, lexical databases and domain information to perform event type detection in the context of the surveillance task assigned to a maritime security operator. This means we aim to minimize the number of false negatives detected by the system. Due to the fact that a detected threat will not result in automatic actions being taken, but rather in an alert to the operator, minimizing false positives has a lower priority, as long as the number of alerts stays within a manageable rate. It is also important that the operator is able to trace back to the sources of the information that triggered an alert. An assumption in the every day work of such operators is that ships with a record are more likely to be involved in subsequent similar situations. This can be due to many, sometimes complex, causes, possibly having to do with the motivations of the crew or the owners, but in any case the correlation between a shady history and future trouble exists.

Throughout this paper we will use the (fictional) running example of the Very Large Crude oil Carrier Sirius Star entering the Dutch coastal waters. This ship has been involved in hijacking, kidnapping of the crew, parliamentary debate about they payment of a ransom sum, and participant in a lengthy and tumultuous aftermath of these events. We choose this ship and its history as an example, because many more suitable cases touch upon sensitive information, which we want to avoid, and yet this example has a clear press coverage. This would make it easy for us to detect the event descriptions in text, and therefore it sets a good lower bound on the performance we need to demonstrate.

This paper is structured as follows. We first discuss related work in Sec. 2. Then, in Sec. 3, we describe the composition of our system and the methods used. Sec. 4 provides a description of the setup we used to evaluate our system. Sec. 5 describes the results after using the system with a sample data set. Finally, these results are discussed in Sec. 6.

2 Related Work

This work falls inside the domain of the application of computational linguistics and information retrieval to the task of structured event extraction. A lot of existing research in these domains have been done using traditional NLP pipelines, such as Gate [2] and Kyoto [9], that would require processing each document in the corpus first, before being able to say something about the history of a ship.

Atkinson *et al.* [1] state that news items, in particular from online media, are particularly interesting to exploit for gathering information about security-related events. They argue that information on certain events might not be available through other (official) sources, and even if they were, official sources often have a significant delay. They continue by presenting two approaches to extracting events related to border security events from on-line news articles: (i) a cluster-based approach looking at the title and first sentence of multiple articles at once, and (ii) an approach processing a single document at a time. These approaches both try to match specific patterns of words to the text, exploiting the fact that news articles are often written in a distinctive style. Variables inside these patterns are then filled to find the various properties of an event. Both these approaches use the articles themselves as starting point, thus requiring to pre-process all articles as they come in.

Turney *et al.* [7] stress that leveraging the respresentation of documents as term vectors, as used by many search engines, is a powerful paradigm that should be employed in many AI-related topics, such as word sense disambiguation, word clustering, spelling correction, and information extraction. The Term Saliency module in our system is an application of this paradigm, using term frequencies represented as vectors to find salient words, rather than employing natural language processing techniques.

3 Approach

Our implementation architecture, shown in Fig. 1, consists of a set of custom-built modules, an installation of WordNet, a modified ElasticSearch[3] cluster, and a database of ship names. The ElasticSearch cluster is filled with a corpus of approximately 25,000 maritime-related press releases. We will first provide an overview of the general system, and then proceed to describe its components in detail in the following subsections.

All international vessels above 300 gross tonnage, all national vessels above 500 gross tonnage, as well as all passenger ships are required to broadcast their position, name and destination using the Automatic Identification System[4] every few seconds. This, along with radar information allows the coastguard's vessel observation systems to track their position. Whenever a vessel enters the area of interest defined by the operator, an event is fired by the vessel observation system, depicted by *Mission Management* in Figure 1. This event is picked up by our risk assessment system.

From that event, a query is formulated by the *Query Builder* to retrieve relevant documents about that ship, taking special care to exclude ships with similar names. This query is fired against the ElasticSearch cluster twice by the *Term Saliency* module - once for retrieving a set of documents relevant

[3] http://www.elasticsearch.org/

[4] http://en.wikipedia.org/wiki/Automatic_Identification_System

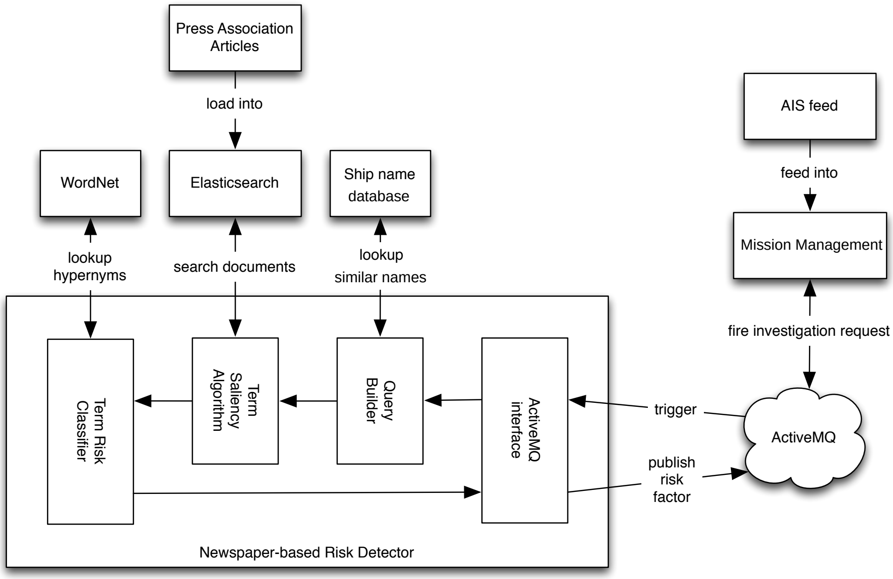


Fig. 1. System Architecture

to the query, and once to retrieve a set of documents that do not match the query. The Rocchio algorithm[6] is then executed against the term vectors of the two results, in essence calcuating a prototype document for the ship under investigation. From this prototype document, every term that on average occurs more than once in every five documents is lemmatized and matched to one or more WordNet synsets by the *Term Risk Classifier*. The selected cut-off point of five is an ad hoc choice, based on our experience with the technique and data set, and has a marginal effect on the quality of the results, but a big influence on the processing speed. All these synsets are then compared to a set of pre-defined synset trees that are each mapped to a specific event type in order to compile an evidence score for the ship for each event type detected.

3.1 Query Builder

ElasticSearch is a distributed search and analytics engine built on top of Apache Lucene. We use a modified version of ElasticSearch that allows us to retrieve the indexed term counts for each indexed document. The search cluster has been filled with approximately 25,000 maritime-related press releases from the Press Association (essentially all articles with the meta-data term "sea" of the past 10 years) and detailed records of about 40,000 ships from IHS FairPlay. These ship records contain, for example, details about ship owners and current and previous names of ships.

The name of each ship in our area of interest is broadcast by its Automatic Identification System (AIS) transmitter, which allows us to formulate a query to find all press releases in which said vessel is mentioned. Due to the fact that ship names often consist of multiple words, and names of different ships can be quite similar, we first search the detailed ship records for other ships that have names that contain the name of the ship being investigated. We can then take extra care to exclude these other names from our search. For example, this allows us to exclude documents about the *Queen Mary* when searching for articles about the ship *Mary*, which otherwise would have matched and been returned. In the case of the Sirius Star, there are no ships with a name that contains the phrase "Sirius Star" other than the Sirius Star itself. So we illustrate the query builder with the example of the Mary and the Queen Mary. The JSON ElasticSearch query constructed to fetch documents about the Mary while excluding documents about the Queen Mary is shown below.

```
query : {
  bool : {
    must : { text_phrase_prefix : { text : "mary" } },
    must_not : [ { text_phrase_prefix : { text : "queen mary" } } ]
  }
}
```

Once the query has been constructed we retrieve the term vectors of all documents that are returned by the query, and the term vectors of a sample of 100 documents that do not match the query. As an example, if we would investigate the Sirius Star, this would result in a set of term vectors about the hijacking of the Sirius Star, as well as a set of term vectors about a number of different arbitrary vessels.

3.2 Term Saliency Algorithm

The Rocchio Algorithm[6] is a relevance feedback technique. This algorithm is applied over the two sets of term vectors, in order to reshape these into one term vector that best describes the documents about the investigated vessel with respect to the other documents in the corpus. The algorithm is described in Equation 1, with $\overrightarrow{Q_m}$ being the modified query vector, $\overrightarrow{Q_o}$ the original query vector, D_r the set of term vectors of related documents, and D_{nr} the set of term vectors of non-related documents. a, b and c are weights, in this case set to 0, 1 and 1 respectively. In our Sirius Star example, D_r would contain terms such as *hijack*, *pirates*, *ship* and *captain*, while D_{nr} would contain *ship*, *captain*, *sea* and *engine*. The resulting vector $\overrightarrow{Q_m}$ would then result in *hijack*, *pirates*, *ship* and *captain*, but with significantly higher weights attached to the first two terms than the last two terms.

$$\overrightarrow{Q_m} = \left(a * \overrightarrow{Q_o}\right) + \left(b * \frac{1}{|D_r|} * \sum_{\overrightarrow{D_j} \in D_r} \overrightarrow{D_j}\right) - \left(c * \frac{1}{|D_{nr}|} * \sum_{\overrightarrow{D_k} \in D_{nr}} \overrightarrow{D_k}\right) \quad (1)$$

This essentially calculates *the query that should have been asked to the search engine in order to get the most number of relevant documents and the least number of irrelevant documents*, which essentially is a list of terms specific about the vessel being identified, along with weights. All dimensions of the $\overrightarrow{Q_m}$ that are lower than 0.2 are removed in order to speed-up subsequent processing.

3.3 Term Risk Classifier

WordNet is a large lexical database of English, with words grouped into sets of cognitive synonyms (synsets), interlinked by means of semantic and lexical relations[5].

To relate terms to event types, we have created a set of pre-defined concepts for each event type we wanted to detect. For example, the concept set for *accident* contains synsets like *hit*, *collide*, *explode*, *sink*, etc. All hyponyms of these synsets are automatically included as well.

Each term in the term vector $\overrightarrow{Q_m}$ is first lemmatized. WordNet is then searched for synsets that contain the term's lemma. If any of these synsets is a match to any of the predefined concept sets, the vessel is considered to have participated in at least one event of the matching event type.

To compensate for ambiguous words, each event type is assigned a score, consisting of the sum of the score for each matching term in the event type's concept set. This term score is in turn calculated by dividing its term frequency by the number of synsets in WordNet that contain the term's lemma. Each found event type with a score lower than 0.1 is pruned for not having enough evidence. In our example, both *hijack*, and *pirates*' lemmatized form *pirate* are a match to the *thievery and piracy related events* event type, which consequenly gets a score derived from both these terms.

3.4 Simple Event Model

To keep the output simple, we assume each matched event type with evidence (in our case *thievery and piracy*) corresponds to one distinct event, with its score as a measure of supporting evidence. These events are represented in RDF, using the Simple Event Model ontology (SEM), which is a light-weight event ontology designed with a minimum of semantic commitment to guarantee maximal interoperability[8]. Each event is modeled as an anonymous event with a *sem:eventType* type corresponding to the matched event type. The ship under investigation is linked to the event as an Actor. All other event properties (Place, Time, other actors) are left unspecified as insufficient information is available to specify these, but the schema does allow specifying them later on, either by extensions to our system or by an outside tool.

The W3C standard provenance ontology PROV[4] is used to link the event back to the documents from which it was originally derived. This allows the operator to manually read the news articles for ships that trigger an alert in order to confirm the potential threat.

The resulting events are then sent back to the vessel observation system, where they are prioritized based on the detected event types and their evidence scores.

For the moment we ignore the date of the past events (apart from the limit of 10 years in the past imposed by the coverage of the news corpus). Possible performance improvements could be obtained by investigating the order and time distribution of the events.

4 Evaluation

To evaluate our system's performance, we have compiled a gold standard, consisting of ships with known risk cases. We then let the system investigate these vessels, and evaluated at three points:

- *E1:* Given the name of a ship, does the system provide us with relevant documents that relate to the ship being investigated? This is done by manually reviewing the documents that are retrieved, checking whether they are relevant for the given ship.
- *E2:* Does the system classify the correct event types that a human annotator would also find when *only* looking at the documents retrieved in *E1*?
- *E3:* Given a ship with known risk behaviour in the past, does the system classify this ship correctly and completely? This is essentially a combination of *E1* and *E2* with a slightly different gold standard, which was constructed before running the evaluation.

Behavior for non-remarkable vessels was also evaluated qualitatively by classifying a large set of around 76000 known ship names and looking for anomalies in the results. A thorough evaluation would include a comparison to actual decisions made by coast guard personnel with and without the assistance of the tool. This remains future work for the moment.

5 Results

The evaluation results for evaluation criteria *E1*, *E2* and *E3* can be found in Table 1. After batch evaluating 76696 vessels, 3064 triggered an alert on at least one category.

6 Discussion

From Table 1 - in particular the difference in recall between *E2* and *E3* - we can see that the system performs quite well for those ships that are actually mentioned in news articles in our corpus. The drop in recall from *E2* to *E3* can for the most part be explained by the lack of news articles found for the affected vessels (see the D_F column for *E1*). A larger and more up-to-date corpus of news articles should hopefully improve these results.

Table 1. Evaluation results for evaluation criteria E1, E2 and E3 described in Section 4. D_F denotes number of documents found by the system, D_R represents which of these documents were actually relevant to the vessel. For both *E1* and *E2*, T_{TP} indicates the number of true positive classified event types, T_{FP} represents false positives, and T_{FN} denotes the number of false negatives. P and R denote Precision and Recall respectively.

	E1			E2					E3				
Vessel	D_F	D_R	P	T_{TP}	T_{FP}	T_{FN}	P	R	T_{TP}	T_{FP}	T_{FN}	P	R
Exxon Valdez	34	8	0.24	3	0	0	1.00	1.00	3	0	0	1.00	1.00
Probo Koala	0	0	1.00	0	0	0	1.00	1.00	0	0	1	1.00	0.00
Costa Concordia	0	0	1.00	0	0	0	1.00	1.00	0	0	2	1.00	0.00
Estonia	136	80	0.59	0	0	1	1.00	0.00	0	0	1	1.00	0.00
Herald of Free Enterprise	95	52	0.55	1	1	0	0.50	1.00	1	1	0	0.50	1.00
Sirius Star	46	44	0.96	2	0	0	1.00	1.00	2	0	0	1.00	1.00
Vindo	26	26	1.00	2	0	0	1.00	1.00	2	0	0	1.00	1.00
Edinburgh Castle	4	0	0.00	0	1	0	0.00	1.00	0	1	1	0.00	0.00
Zeldenrust	1	1	1.00	2	1	0	0.67	1.00	2	1	0	0.67	1.00
Scandinavian Star	9	9	1.00	1	3	0	0.25	1.00	1	3	0	0.25	1.00
Lady Azza	0	0	1.00	0	0	0	1.00	1.00	0	0	2	1.00	0.00
Ronin	2	2	1.00	2	1	0	0.67	1.00	2	1	0	0.67	1.00
Union Pluto	1	1	1.00	2	1	0	0.67	1.00	2	1	0	0.67	1.00
Achille Lauro	72	48	0.67	0	0	3	0.00	0.00	0	0	2	1.00	0.00
Viking Victor	23	22	0.96	0	1	1	0.00	0.00	0	1	1	0.00	0.00
Astree	4	4	1.00	1	2	0	0.33	1.00	1	2	0	0.33	1.00
Total	453	297	0.66	16	11	5	0.59	0.76	16	11	10	0.59	0.62

Of the ships that were mentioned in at least one news article, the system only failed to raise the correct red flags for three instances, one of which did trigger an alert but for an incorrect event type. For the other two, the system was most probably thrown off by the fact that these vessels (Estonia and Achille Lauro) were mentioned a lot in news articles about other events involving different ships. One could say that these ships might have been 'too famous' to be correctly picked up.

The false positives generated by the system seem to mostly originate from the fact that, in addition to the correct event type, sometimes additional types are triggered by the documents that describe the correct event type. For example, in the case of smuggling, the smuggled goods are often *seized* by the authorities after being discovered, which in turn triggers the *thievery and piracy* category, as the system in this case canot discern between the legal interpretation of *seizing of goods*, and the illegal one.

Out of the 76696 batch-evaluated ships, the system did not detect any risk factors for 73532. This means that, with our system in use, the operator will receive an alert and has to confirm approximately 4% of all vessels. If we assume this is a representative sample of ships, this will cause the operator to have to look at approximately 5 vessels each hour for the Netherlands Exclusive Economic Zone (compared to 120 when manually assessing all ships).

When manually reviewing the ships that trigger alerts, a considerable number of them either have names that refer to a place ('baltic sea', 'brasilia', 'brooklyn', 'casablanca'), or are named after words that have something to do with the exact threats we are looking for ('buccaneer', 'dealer', 'robin hood'). Due to the search engine only looking at words in the press releases without actually disambiguating them, the queries formed from the names of these ships most probably return articles about entirely different ships. These false positives, however, can be very quickly dismissed by the operator, as one glance at the documents should be enough to see they are not about said ship.

In this paper we wanted to focus on the statistical saliency algorithm and the term risk classification part of the entire event detection pipe line. We assume that a thorough NER tool would solve many of the cases discussed above if properly retrained with domain-specific terms such as ship vessels and port names.

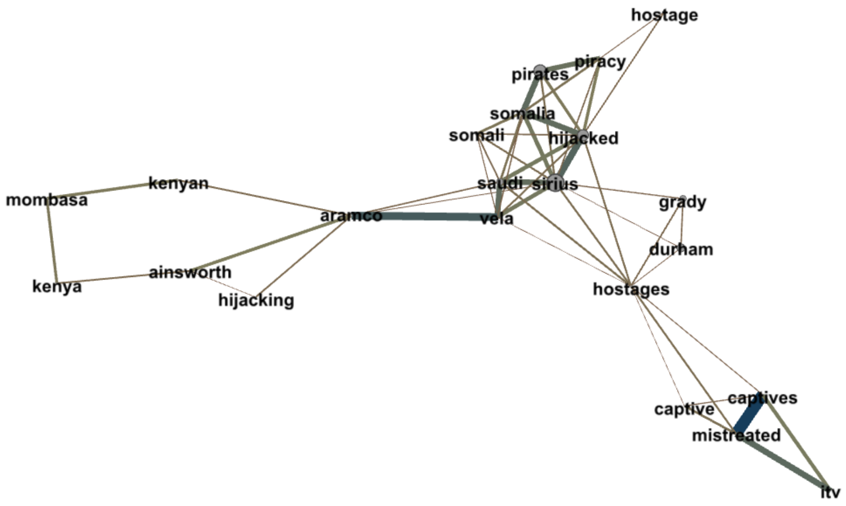


Fig. 2. Term Network for the Sirius Star

7 Conclusion

In this paper, we have described an event-type extractor on top of ElasticSearch, and applied this system in a case study concerned with assisting maritime security operators to assess potential risk factors of vessels. Our main objectives were to investigate if such a system, based on a combination of relevance feedback, lexical databases and domain information would yield results useful for the surveillance task assigned to maritime security operators.

With a task-oriented focus, we have evaluated the performance of our system using a set of vessels with known risk factors, and concluded that, given that news articles about certain events actually exist in the system's database, the system can raise red flags about ships with a suspicious history fairly accurately, and does not produce enough false negatives to overload the operator.

We will continue this line of research by using co-occurrence metrics to form term networks in order to detect clusters of terms that may point to separate distinct events. An example of such a network is shown in Figure 2. Natural Language Processing tools will then be employed to further fill in the rest of the event properties such as other actors, places and times.

Acknowledgements

We wish to thank Thomas Ploeger for his work in the initial stages of this research, and the Press Association for providing us with a set of maritime-related press releases. This publication was supported by the Dutch national program COMMIT. The research work was carried out as part of the Metis project under the responsibility of the Embedded Systems Innovation by TNO with Thales Nederland B.V. as the carrying industrial partner.

References

[1] M. Atkinson, J. Piskorski, E. Goot, and R. Yangarber. Multilingual real-time event extraction for border security intelligence gathering. In U. K. Wiil, editor, *Counterterrorism and Open Source Intelligence*, volume 2 of *Lecture Notes in Social Networks*, pages 355–390. Springer Vienna, 2011.

[2] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. Gate: an architecture for development of robust hlt applications. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 168–175. Association for Computational Linguistics, 2002.

[3] T. Hendriks and P. van de Laar. Metis: Dependable cooperative systems for public safety. *Procedia Computer Science*, 16:542–551, 2013.

[4] T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, and J. Zhao. Prov-o: The prov ontology. *W3C Recommendation, http://www. w3. org/TR/prov-o/(accessed 30 Apr 2013)*, 2013.

[5] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, 3(4):235–244, 1990.

[6] J. J. Rocchio. Relevance feedback in information retrieval. 1971.

[7] P. D. Turney, P. Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

[8] W. R. Van Hage, V. Malaisé, R. Segers, L. Hollink, and G. Schreiber. Design and use of the simple event model (sem). *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(2):128–136, 2011.

[9] P. Vossen, E. Agirre, N. Calzolari, C. Fellbaum, S.-k. Hsieh, C.-R. Huang, H. Isahara, K. Kanzaki, A. Marchetti, M. Monachini, et al. Kyoto: a system for mining, structuring and distributing knowledge across languages and cultures. In *LREC*, 2008.