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

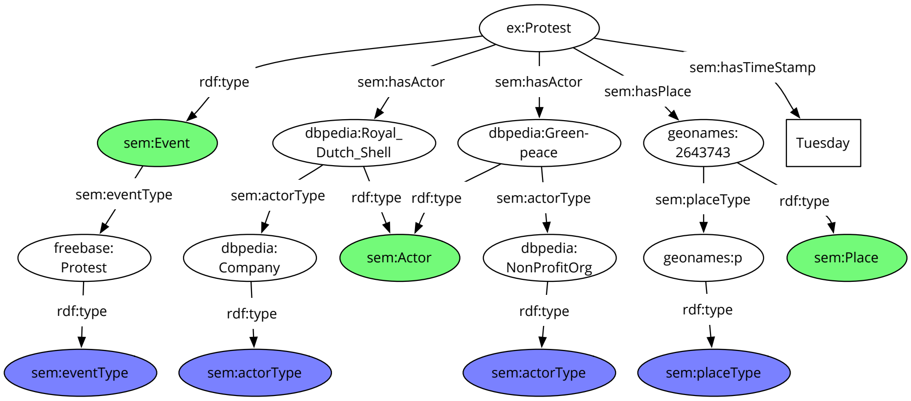

Fig. 1. Example of a SEM-event that might be instantiated for the event: “Tuesday, Greenpeace protested against Shell in London”

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/

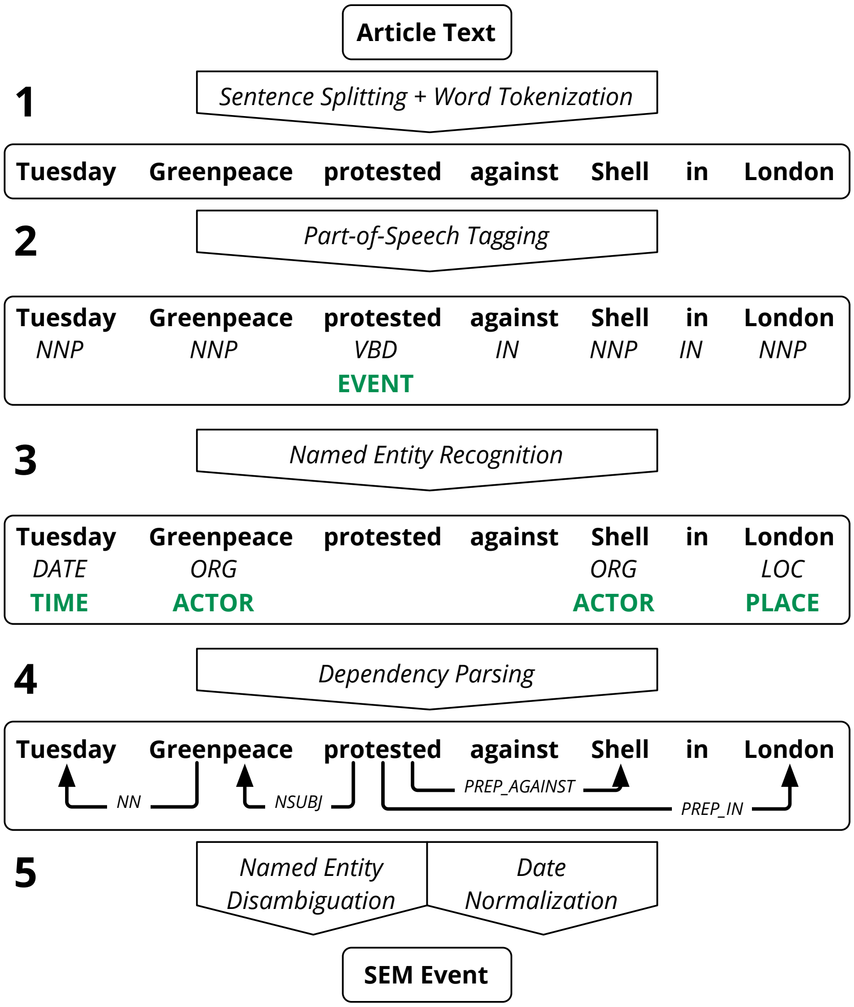
Article Text
1
Sentence Splitting + Word Tokenization
Tuesday Greenpeace protested against Shell in London
2
Part-of-Speech Tagging
Tuesday Greenpeace protested against Shell in London
NNP NNP VBD IN NNP IN NNP
EVENT
3
Named Entity Recognition
Tuesday Greenpeace protested against Shell in London
DATE ORG ORG LOC
TIME ACTOR ACTOR PLACE
4
Dependency Parsing
Tuesday Greenpeace protested against Shell in London
NN
NSUBJ
PREP_AGAINST
PREP_IN
5
Named Entity Disambiguation
Date Normalization
SEM Event

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)