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

Workshop in conjunction with the
11th International Semantic Web Conference 2012
Boston Massachussets, USA, 12 November 2012

Edited by:
Marieke van Erp
Laura Hollink
Willem Robert van Hage
Raphaël Troncy
David A. Shamma

Preface

In recent years, researchers in several communities involved in aspects of the web have begun to realise the potential benefits of assigning an important role to events in the representation and organisation of knowledge and media – benefits which can be compared to those of representing entities such as persons or locations instead of just dealing with more superficial objects such as proper names and geographical coordinates. While a good deal of relevant research – for example, on the modelling of events – has been done in the semantic web community, much complementary research has been done in other, partially overlapping communities, such as those involved in multimedia processing and information retrieval.

However, there is a shift in semantics in multimedia research, one that moves away from content semantics towards conversation semantics that is contained in social media. With respect to events and information, what happens in an event becomes secondary to how people react and/or what they talk about. The attendance of DeRiVE 2011 proved that there is a great interest from many different communities in the role of events.

The goal of DeRiVE 2012 is to further strengthen and expand on the results from DeRiVE 2011 and to strengthen the participation of the semantic web community in the recent surge of research on the use of events as a key concept for representing knowledge and organising and structuring media on the web. The workshop invited contributions to three central questions, with the goal to formulate answers to these questions that advance and reflect the current state of understanding. Each submission was be expected to address at least two questions explicitly, if possible including a system demonstration. This year, we also specifically focused on event and conversation semantics in multimedia and social media.

The questions we aim to address are the following:

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

- How can events be recognised in particular types of material on the web, such as calendars of public events, social networks, microblogging sites, semantic wikis, and regular web pages?
- How can events be summarised, segmented and described using social media?
- How can the quality and veracity of the events mentioned in noisy microblogging sites such as Twitter be verified?
- How can a system recognise a complex event that comprises separately recognisable subevents?
- How can a system recognise when a newly detected event is the same as a previously detected and represented event?

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

- How can we improve the interoperability of the various event vocabularies such as EVENT, LODE, SEM, or F to name a few?
- How deployed is the schema.org Event class on the web?
- To what extent can the many different event infoboxes of Wikipedia be reconciled for Wikidata?
- What are the requirements for event representations for qualitatively different types of events (e.g., historical events such as wars; cultural events such as upcoming concerts; personal events such as family vacations)?
- How can aspects of existing event representations developed in other communities be adapted to the needs of the semantic web?
- To what extent can/should a unified event model be employed for such different types of events?
- How do social contexts (Facebook, Twitter, etc.) change the implicit content semantics?

Question 3: What is the relationship between events, data, and applications?

- How can events be represented in a way to support conversation semantics, search, or enhanced browsing?
- How do tools for event annotation and consumption alter or change the content semantics of the event itself?
- How can we improve existing methods for visualising event representations and enabling users to interact with them in semantic web user interfaces?
- What are the requirements for event detection, representation, and systems creation implicitly or explicitly defined by these three questions?

Contributions of the Workshop Papers

In each of the seven accepted papers for DeRiVE 2012, two of the workshop topics are addressed. The first, fourth and fifth contributions to be presented, *Automatic Classification and Relationship Extraction for Multi-Lingual and Multi-Granular Events from Wikipedia* by Daniel Hienert, Dennis Wegener and Heiko Paulheim, *Harnessing Disagreement for Events Semantics* by Lora Aroyo and Chris Welty and *Using Syntactic Dependencies and WordNet Classes for Noun Event Recognition* by Yoonjae Jeong and Sung-Hyon Myaeng, present experiments for the extraction and (re)presentation of events on the Semantic Web.

The second and third contributions, *Hyperlocal Event Extraction of future Events* by Tobias Arrskog, Peter Exner, Håkan Jonsson, Peter Norlander, and Pierre Nugues and *Automatic Extraction of Soccer Game Events from Twitter* by Guido van Oorschot, Marieke van Erp and Chris Dijkshoorn primarily focus on extraction of events from real-world data but also explore how wide deployment of their techniques would alter current methods of information processing around events.

The focus on detection in a majority of the submissions shows that this is still a topic that deserves much attention, but the fact that there is already a significant amount of (semi-)structured event-data available and that the results of event detection are reaching acceptable levels have opened up interesting avenues for starting to use event-data in real world settings. This is showcased by the the sixth and seventh contributions accepted for presentation, *Bringing parliamentary debates to the Semantic Web* by Damir Juric, Laura Hollink and Geert-Jan Houben and *Making Sense of the Arab Revolution and Occupy: Visual Analytics to Understand Events* by Thomas Ploeger, Bibiana Armenta, Lora Aroyo, Frank de Bakker and Iina Hellsten. These contributions show what issues are encountered in working with event-based and how these are being addressed by use of various (inter)disciplinary methods.

We hope that in compiling the programme and proceedings for DeRiVE 2012 we have succeeded in presenting various perspectives and discussion points on the problems around detection, representation and exploitation of events and that the workshop contributed to yet another step closer to getting to understand events and their uses better.

September 2012
Marieke van Erp, VU University Amsterdam
Laura Hollink, VU University Amsterdam
Willem Robert van Hage, VU University Amsterdam
Raphaël Troncy, EURECOM
David A. Shamma, Yahoo! Research

Programme Committee

The following colleagues kindly served in the workshop's program committee. Their joint expertise covers all of the questions addressed in the workshop, and they reflect the range of relevant scientific communities.

- Jans Aasman, Franz, Inc.,
- Klaus Berberich, Max-Planck Institute for Informatics
- Fausto Giunchiglia, University of Trento
- Christian Hirsch, The University of Auckland

- Diana Maynard, University of Sheffield
- Vasileios Mezaris, CERTH/ITI
- Yves Raimond, BBC
- Matthew Rowe, Knowledge Media Institute
- Ansgar Scherp, University of Koblenz-Landau
- Nicu Sebe, University of Trento
- Ryan Shaw, University of North Carolina
- Thomas Steiner, Google
- Denis Teyssou, AFP
- Sarah Vieweg, University of Colorado Boulder

Contents

Automatic Classification and Relationship Extraction for Multi-Lingual and Multi-Granular Events from Wikipedia

Daniel Hienert[1], Dennis Wegener[1] and Heiko Paulheim[2]

[1] GESIS – Leibniz Institute for the Social Sciences
Unter Sachsenhausen 6-8, 50667 Cologne, Germany
{daniel.hienert, dennis.wegener}@gesis.org

[2] Technische Universität Darmstadt
Knowledge Engineering Group
Hochschulstraße 10, 64283 Darmstadt, Germany
paulheim@ke.tu-darmstadt.de

Abstract. Wikipedia is a rich data source for knowledge from all domains. As part of this knowledge, historical and daily events (news) are collected for different languages on special pages and in event portals. As only a small amount of events is available in structured form in DBpedia, we extract these events with a rule-based approach from Wikipedia pages. In this paper we focus on three aspects: (1) extending our prior method for extracting events for a daily granularity, (2) the automatic classification of events and (3) finding relationships between events. As a result, we have extracted a data set of about 170,000 events covering different languages and granularities. On the basis of one language set, we have automatically built categories for about 70% of the events of another language set. For nearly every event, we have been able to find related events.

Keywords: Historical Events, News, Wikipedia, DBpedia

1 Introduction

Wikipedia is an extensive resource for different types of events like historical events or news that are user-contributed and quality-proven. Although there is plenty of information on historical events in Wikipedia, only a small fraction of these events is available in a structured form in DBpedia. In prior work we have focused on extracting and publishing these events for the use in the semantic web and other applications [6]. In this paper, we focus on how the dataset can be enriched and its quality can be further improved. We address this question with two approaches: to find categories for events and to extract relationships between events. These features can later be used in end-user applications to list related events, browse between events or filter events from the same category.

The remainder of this paper is as follows: Section 2 presents related work. In Section 3, we address the question on how events can be detected, extracted,

processed and presented in different forms for the semantic web (Workshop questions 1, 2 and 3). In Section 4 we present an approach on how events can be automatically classified with categories (Question 1). In Section 5 we show how relationships between events from different languages and granularities can be found (Question 1).

2 Related Work

There is a range of systems specialized for the extraction of events and temporal relations from free text. The TARSQI toolkit [16] can detect events, times and their temporal relations by temporal expressions in news articles. HeidelTime [14] is a rule-based system for the extraction and normalization of temporal expressions using mainly regular expressions. The TIE system [9] is an information extraction system that extracts facts from text with as much temporal information as possible and bounding start and end times.

Some work has been done for the extraction of events from Wikipedia articles with machine learning or rule-based approaches and the presentation for the end user in user interfaces with timelines and maps. The approach of Bhole [2] for example first classifies Wikipedia articles as persons, places or organizations on the basis of Support Vector Machines (SVM). Then text mining is used to extract links and event information for these entities. Entities and their events can be shown on a timeline. In another system [3] major events are extracted and classified for a historical Wikipedia article and shown in a user interface with a timeline, map for event locations and named entities for each event.

Other work concentrates on the extension of knowledge bases like DBpedia [1] or YAGO [15] with temporal facts. Exner and Nugues [4] have extracted events based on semantic parsing from Wikipedia text and converted them into the LODE model. They applied their system to 10% of the English Wikipedia and extracted 27,500 events with links to external resources like DBpedia and GeoNames. Since facts in knowledge bases evolve over time the system T-YAGO [17] extends the knowledge base YAGO with temporal facts, so that they can be queried with a SPARQL-style language. As a subsequent technology, Kuzey & Weikum [8] presented a complete information extraction framework on the base of T-YAGO that extracts more than one million temporal facts from Wikipedia resources like semi-structured data (infoboxes, categories, lists and article titles) and free text of Wikipedia articles with a precision over 90% for semi-structured and 70% for full text extraction. Alternatively, the YAGO2 system [7] extends the YAGO knowledge base with temporal and spatial components. This information is extracted from infoboxes and other resources like GeoNames.

There is a collection of ontologies for the modeling of events in RDF like EVENT[1], LODE [13], SEM [5], EventsML[2] and F [12], a comparison can be found in [5].

However, most related work in this field is about the *extraction* of events from free text or knowledge bases like Wikipedia or YAGO and the *enrichment* of entities from text or knowledge bases with temporal information. Not much work has been done on

[1] http://motools.sourceforge.net/event/event.html
[2] http://www.iptc.org/EventsML/

the further enrichment of event datasets such as adding relations or additional information like categorizations.

3 Events from Wikipedia

Wikipedia is a rich data source for events of different topics, languages and granularity. Most research focuses on the extraction of events from the full text of Wikipedia articles and on relating it to the appropriate entities. Major historical events have their own article, or events are collected in articles for a special topic. Events are also collected in time units of different granularity (i.e. years or months) available for different languages. These articles contain lists with events, whose structure is relatively stable. In prior work we have focused on the extraction of events from year-based articles, which include information on individual years for different languages [6]. Table 1 gives an overview over the extracted events for different languages and their extraction quotients. The number of possible events for each language is based on the assumption that every event line in the Wiki markup starts with an enumeration sign. The extracted dataset has several unique characteristics: (1) it has a wide temporal coverage from 300 BC to today, (2) it is available for a lot of different languages, (3) different granularities (year or month) are available, (4) Wikipedia users already have chosen which events are important for different granularities, (5) events already contain links to entities, (6) events have categorizations or can be enriched with categorization and relationships among each other.

Table 1. Number of extracted events for language/granularity and the extraction quotients

Language/Granularity	Possible Events	Extracted Events	Extraction Quotient
German/Year	36,713	36,349	99.01%
English/Year	39,739	34,938	87.92%
Spanish/Year	20,548	19,697	95.86%
Romanian/Year	13,991	10,633	76.00%
Italian/Year	14,513	10,339	71.24%
Portuguese/Year	8,219	7,395	89.97%
Catalan/Year	7,759	6,754	87.05%
Turkish/Year	3,596	3,327	92.52%
Indonesian/Year	2,406	1,963	81.59%
English/Month	38,433	35,633	92.71%
German/Month	11,660	11,474	98.40%
Total		**178,502**	

3.1 Extraction, processing and provision

Figure 1 shows the overall extraction and processing pipeline. Our software crawls Wikipedia articles for different granularities (years and months) and different languages. For year-based articles, German, English, Spanish, Romanian, Italian, Portuguese, Catalan, Turkish and Indonesian with a temporal coverage from 300BC to today are crawled. For daily events, German and English articles from the year 2000 to today are collected. In the next step, the events are extracted from Wiki

markup. We use a set of language-dependent regular expressions for the identification of the event section in the article, the identification of events in the event section and the separation of date, description and links for each event. Events can be further described by categories that result from headings in the markup. Events and links are then stored in a MySQL database.

The resulting data set is then further processed. For the automatic classification see Section 4, for the finding of relationships between events see Section 5. We also crawl the Wikipedia API to add an individual image to each event for the use in the timeline.

We provide access to the extracted events via the Web-API, SPARQL endpoint, Linked Data Interface and in a timeline. The Web-API[3] gives lightweight and fast access to the events. Events can be queried by several URL parameters like begin_date, end_date, lang, query, format, html, links, limit, order, category, granularity and related. Users can query for keywords or time periods, and results are returned in XML or JSON format. The Linked Data Interface[4] holds a representation of the yearly English dataset in the LODE ontology [13]. Each event contains links to DBpedia entities. Users can query the dataset via the SPARQL endpoint (http://lod.gesis.org/historicalevents/sparql). Additionally, yearly events for the English, German and Italian dataset are shown in a Flash timeline (http://www.vizgr.org/historical-events/timeline/) with added images and links to Wikipedia articles. Users can search for years, scroll and scan the events and navigate to Wikipedia articles.

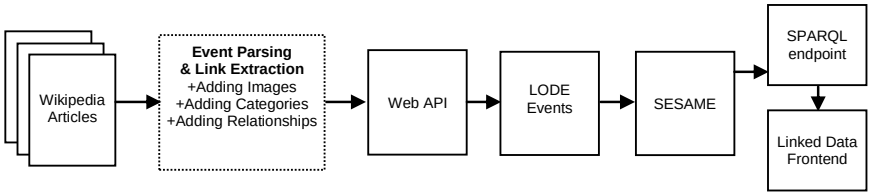

Fig. 1. Processing, extraction and provision pipeline.

3.2 Extraction of daily events

In addition to the extraction of yearly events presented in [6], we have extracted daily events from the German and English Wikipedia version. The German version provides events on a daily basis in articles of months (i.e. http://de.wikipedia.org/wiki/Juni_2011) from the year 2000 to today. The English structure is quite more complicated and daily events are distributed in three different site structures: (1) most daily events are collected in the *Portal:Current events* (http://en.wikipedia.org/wiki/Portal:Current_events), (2) some events are collected in the *Portal:Events* (before July 2006) and (3) other events are collected in month collections similar to the German version. English daily events are also available for the years 2000 to today. First, we have extended the extraction software to query

[3] http://www.vizgr.org/historical-events/
[4] http://lod.gesis.org/pubby/page/historicalevents/

these site structures. Then, regular expressions for the identification of event section and for the individual events have been added. The extraction algorithm had to be slightly modified to handle new structures specific for daily events. As a result, the software could extract 35,633 English daily events (extraction quotient: 92.17%) and 11,747 German daily events (extraction quotient: 98.40%).

3.3 Analyzing the data set

The overall data set has been analyzed as a prerequisite to the automatic classification and the search for relationships between events. The number of extracted events and extraction quotients for different languages and granularity are shown in Table 1. The categories in German events are created from subheadings on the corresponding Wikipedia page. Yearly German events are categorized with one or two categories by headings of rank 2 or 3, which can be used for the automatic classification of events. Table 2 shows the ten most used categorizations for German events. In English or other languages categorizations are rarely used. The number of links and entities per event can be seen in Table 3. In the German and English dataset most events have between one and four links.

Table 2. Categories (translated) and their counts for yearly German events

Category	Count
Politics and world events	18,887
Culture	4,135
Science and technology	3,096
Religion	2,180
Economy	2,011
Sports	1,434
Disasters	1,351
Politics	613
Culture and Society	309
Society	286

Table 3. Distribution of links to entities within the German and English yearly dataset

Count of entities	English	German
No entity	6,371	1,489
One entity	5,773	7,815
Two entities	10,143	9,969
Three entities	8,405	8,086
Four entities	4,499	4,606
Five entities	2,376	2,457
Six entities	1,271	1,234
Seven or more entities	901	693

4 Automatic Classification of Events

To provide a useful semantic description of events, it is necessary to have types attached to these events. Possible types could be "Political Event", "Sports Event", etc. In the crawled datasets, some events already have types extracted from the Wikipedia pages, while others do not. Therefore, we use machine learning to add the types where they are not present.

The datasets we have crawled already contain links to Wikipedia articles. In order to generate useful machine learning features, we have transformed these links to DBpedia entities. For inferring event types, we have enhanced our datasets consisting of events and their descriptions by more features: the direct types (rdf:type) and the categories (dcterms:subject) of the entities linked to an event, both including their

transitive closures (regarding rdfs:subClassOf and skos:broader, respectively). For enhancing the datasets, we have used our framework FeGeLOD [11], which adds such machine learning features from Linked Open Data to datasets in an automatic fashion. The rationale of adding those features is that the type of an event can be inferred from the types of the entities involved in the event. For example, if an entity of type SoccerPlayer is involved in an event, it is likely that the event is a sports event.

As discussed above, the majority of events in our datasets comprises between one and four links to entities. Therefore, we have concentrated on such events in our analysis. We have conducted two experiments: first, we have inferred the event types on events from the German dataset, using cross validation for evaluation. Second, we have learned models on the German datasets and used these models to classify events from the English dataset, where types are not present. In the second experiment, we have evaluated the results manually on random subsets of the English dataset.

Figure 2 depicts the classification accuracy achieved in the first experiment, using 10-fold cross validation on the German dataset. We have used four random subsets of 1,000 events which we have processed by adding features and classifying them with three different commonly used machine learning algorithms: i.e., Naïve Bayes, Ripper (in the JRip implementation), and Support Vector Machines (using the Weka SMO implementation, treating the multi-class problem by using 1 vs. 1 classification with voting). As a baseline, we have predicted the largest class of the sample. It can be observed that the categories of related entities are more discriminative than the direct types. The best results (around 80% accuracy) are achieved with Support Vector Machines.

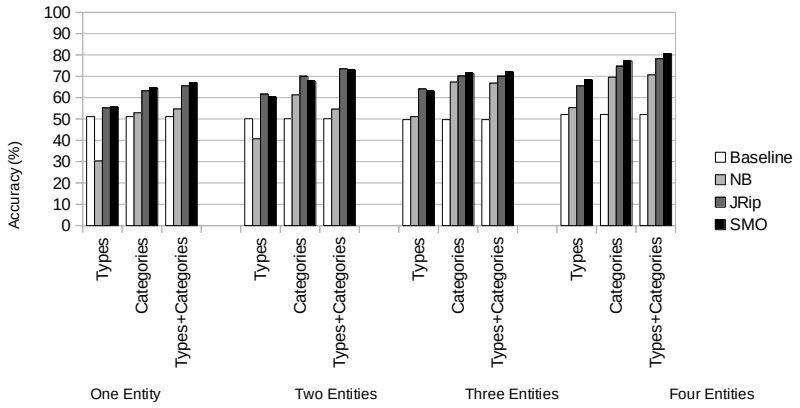

Fig. 2. Classification accuracy on the German dataset, using ten-fold cross validation for evaluation

Since Support Vector Machines have yielded the best results in the first experiment, we have trained four SVMs for the second experiment, one for each number of related entities (one through four), using the subsets of 1,000 events. We

have then used these models to classify four subsets of the English dataset, consisting of 50 events each. The results of that classification have been evaluated manually.

The results are shown in Figure 3. First, we have tested the best performing combination of the first experiment, using both categories and direct types of the related entities. Since the results were not satisfying, we have conducted a second evaluation using only direct types, which yielded better results. The most likely reason why categories work less well as features than classes is that the German and the English DBpedia use the same set of classes (i.e., DBpedia and YAGO ontology classes, among others), but different categories. In our experiments, we have observed that only a subset of the categories used in the German DBpedia have a corresponding category in the English DBpedia. Thus, categories, despite their discriminative power in a single-language scenario, are less suitable for training cross-language models.

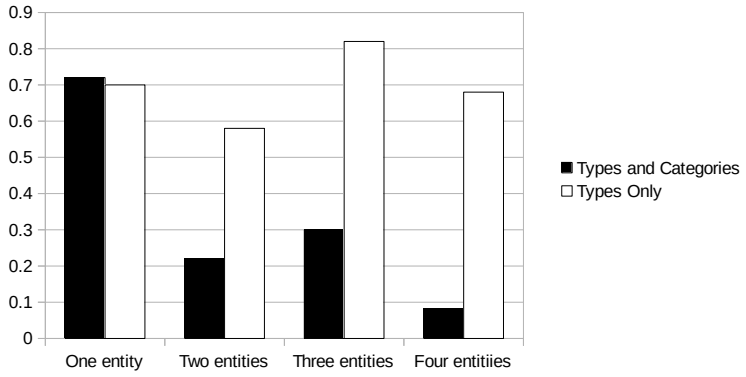

Fig. 3. Classification accuracy achieved on English dataset, using Support Vector Machines trained on the German dataset

In summary, we have been able to achieve a classification accuracy of around 70% for the English dataset, using a model trained on the German dataset. The results of both experiments show that machine learning with features from DBpedia is a feasible way to achieve an automatic classification of the extracted events.

5 Relationships Between Events

With a dataset of events for different languages and granularities it is interesting to know which relations between these events exist. To find relationships, different features of the events could be used: (1) time, (2) categories, (3) topic/content or (4) links. Time as a single criterion is not by far enough. The category is too simplistic and there are only a few categories. Relationships based on the topic/content of the event are not easy to find as the events only include micro-text with a few words or sentences. Taking links as a criterion, we have to consider which links to take and

how many links. In our approach we use a combination of the features time and links for extracting relationships between events.

As described in Section 3.3, we have extracted 178,502 events in total. From these, 172,189 events include links. As a preprocessing step, we transform every non-English link to the English equivalent by querying the inter-language link from the Wikipedia API. As a result, every event from different languages contains links to English Wikipedia/DBpedia entities.

In the following, we analyze this set of events. As first step we vary the number of links that two events have to share and count the events that share this number of links with at least one other event (see Table 4). In detail, we consider two events to *share a link* if these events contain a link to the same DBpedia entity. From our analysis results it can be seen that 95.8 % of the events (that include links) share at least one link with at least one other event. As we are dealing with a multi-lingual set of events, it is interesting to know how many events share one link with at least one event of a different language. In our set of events, 155,769 events share at least one link with at least one other event of a different language, which is 90.5 % of the events in the set. 75.7% of the events include a link to another granularity, i.e. from year to month or vice versa.

Table 4. Analysis of the number of shared links between events

# shared links	# events that share the number of links with at least one other event	in % (# total events = 172,189)
1	165,014	95.8 %
2	100,401	58.3 %
3	35,456	20.6%
4	9,900	5.7%

So far, we have looked for events that share one link in the overall database. In the following, we vary the time interval in which we search for these events (see Table 5). In detail, if we look at an event at time x, an interval of one month means that we search for events in the time interval [x-15 days : x + 15 days]. For the time-based analysis, we can only consider events where the date includes information on the day (and not only on the month and year). In our set these are 109,510 events.

Table 5. Analysis of the number of events that hold shared links in a given time interval

Time interval	Number of events that share one link with at least one other event in the time interval	In % (number of total events with exact date = 109,510)
Overall	105,042	95,9 %
Year [x-182 days : x+182 days]	90,193	82,4 %
Month [x-15 days : x+15 days]	74,499	68,0 %
Week [x-3 days : x+3 days]	61,246	55,9 %

Based on this analysis we have been able to define the *relatedness* between two events A and B with the time interval minimal and the number of shared links

maximal between these events. Whereby we have found that in our dataset, a large part has at least one link in common (95.8%) within a time interval of a year (82.4%) and we can also find links to other languages (90.5%) and granularities (75.7%).We have implemented the relatedness feature in the Web-API. To compute related events for an individual event, we query for events that have at least one link in common within a time interval of plus/minus ten years and then sort results first by number of shared links and then by time distance to the original event.

For example, the query for *Arab Spring*[5] finds eleven events from the yearly English dataset and related events from other languages and granularities. For example, the event of 2011/01/14: "Arab Spring: The Tunisian government falls after a month of increasingly violent protests President Zine El Abidine Ben Ali flees to Saudi Arabia after 23 years in power." lists equivalent events from different languages, i.e. Italian: "In Tunisia, dopo violente proteste…", Spanish: "en Túnez el presidente Zine El Abidine Ben…", German: "Tunis/Tunesien: Nach den schweren Unruhen der Vortage verhängt Präsident Zine el-Abidine…" and from a month/news view: "Thousands of people protest across the country demanding the resignation of President Zine El Abidine Ben Ali. [Link] (BBC)"

As a final step we have compiled an evaluation set with 100 events and 5 related events for each and analyzed them manually. We have found that the perceived relatedness between two events (1) depends on the time interval between events and (2) depends on the count (1 vs. 4), type (general types like *Consul* vs. finer types like *Julius Caesar*) and position (at the beginning or the end of the description) of shared links.

In summary, we have been able to find a related event for nearly every event in the dataset, also for events from other languages and granularities.

6 Conclusion

We have extracted an event dataset from Wikipedia with about 170,000 events for different languages and granularities. A part of these events includes categories which can be used to automatically build categories for about 70% of another language set on the basis of links to other Wikipedia/DBpedia entities. The same linking base is used together with a time interval to extract related events for nearly every event, also for different languages and granularities.

At the moment, we only use Wikipedia/DBpedia links that are already included in the events' descriptive texts. However, those links are not always complete or available in other data sets. Using automatic tools such as DBpedia spotlight [10] would help increasing the result quality and allow us to process text fragments without hyperlinks as well.

At the end of Section 5 we have shown that the perceived quality of events depends also on the abstractness of links. The analysis on how the abstractness of links can be modeled and used as an additional feature for the ranking of related events remains to future work.

[5] http://www.vizgr.org/historical-events/search.php?query=arab%20spring&related=true

References

1. Auer, S. et al.: DBpedia: A Nucleus for a Web of Open Data. In 6th Int'l Semantic Web Conference, Busan, Korea. pp. 11–15 Springer (2007).
2. Bhole, A. et al.: Extracting Named Entities and Relating Them over Time Based on Wikipedia. Informatica (Slovenia). 31, 4, 463–468 (2007).
3. Chasin, R.: Event and Temporal Information Extraction towards Timelines of Wikipedia Articles. Simile. 1–9 (2010).
4. Exner, P., Nugues, P.: Using semantic role labeling to extract events from Wikipedia. Proceedings of the Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011). Workshop in conjunction with the 10th International Semantic Web Conference 2011 (ISWC 2011). , Bonn (2011).
5. Hage, W.R. van et al.: Design and use of the Simple Event Model (SEM). Web Semantics: Science, Services and Agents on the World Wide Web. 9, 2, (2011).
6. Hienert, D., Luciano, F.: Extraction of Historical Events from Wikipedia. Proceedings of the First International Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data (KNOW@LOD 2012). , Heraklion, Greece (2012).
7. Hoffart, J. et al.: YAGO2: exploring and querying world knowledge in time, space, context, and many languages. Proceedings of the 20th international conference companion on World wide web. pp. 229–232 ACM, New York, NY, USA (2011).
8. Kuzey, E., Weikum, G.: Extraction of temporal facts and events from Wikipedia. Proceedings of the 2nd Temporal Web Analytics Workshop. pp. 25–32 ACM, New York, NY, USA (2012).
9. Ling, X., Weld, D.S.: Temporal Information Extraction. In: Fox, M. and Poole, D. (eds.) AAAI. AAAI Press (2010).
10. Mendes, P. et al.: DBpedia Spotlight: Shedding Light on the Web of Documents. In the Proceedings of the 7th International Conference on Semantic Systems (I-Semantics). (2011).
11. Paulheim, H., Fürnkranz, J.: Unsupervised Generation of Data Mining Features from Linked Open Data. International Conference on Web Intelligence and Semantics (WIMS'12). (2012).
12. Scherp, A. et al.: F–a model of events based on the foundational ontology dolce+DnS ultralight. Proceedings of the fifth international conference on Knowledge capture. pp. 137–144 ACM, New York, NY, USA (2009).
13. Shaw, R. et al.: LODE: Linking Open Descriptions of Events. Proceedings of the 4th Asian Conference on The Semantic Web. pp. 153–167 Springer-Verlag, Berlin, Heidelberg (2009).
14. Strötgen, J., Gertz, M.: HeidelTime: High quality rule-based extraction and normalization of temporal expressions. Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 321–324 Association for Computational Linguistics, Stroudsburg, PA, USA (2010).
15. Suchanek, F.M. et al.: Yago: a core of semantic knowledge. Proceedings of the 16th international conference on World Wide Web. pp. 697–706 ACM, New York, NY, USA (2007).
16. Verhagen, M., Pustejovsky, J.: Temporal processing with the TARSQI toolkit. 22nd International Conference on on Computational Linguistics: Demonstration Papers. pp. 189–192 Association for Computational Linguistics, Stroudsburg, PA, USA (2008).
17. Wang, Y. et al.: Timely YAGO: Harvesting, Querying, and Visualizing Temporal Knowledge from Wikipedia. Proceedings of the 13th International Conference on Extending Database Technology (EDBT), Lausanne, Switzerland, March 22-26. pp. 697–700 (2010).

Hyperlocal Event Extraction of Future Events

Tobias Arrskog, Peter Exner, Håkan Jonsson, Peter Norlander, and Pierre Nugues

Department of Computer Science, Lund University
Advanced Application Labs, Sony Mobile Communications
{tobias.arrskog,peter.norlander}@gmail.com
hakan.jonsson@sonymobile.com
{peter.exner,pierre.nugues}@cs.lth.se

Abstract. From metropolitan areas to tiny villages, there is a wide variety of organizers of cultural, business, entertainment, and social events. These organizers publish such information to an equally wide variety of sources. Every source of published events uses its own document structure and provides different sets of information. This raises significant customization issues. This paper explores the possibilities of extracting future events from a wide range of web sources, to determine if the document structure and content can be exploited for time-efficient hyperlocal event scraping. We report on two experimental knowledge-driven, pattern-based programs that scrape events from web pages using both their content and structure.

1 Introduction

There has been considerable work on extracting events from text available from the web; see [1] for a collection of recent works. A variety of techniques have been reported: [2] used successfully data-driven approaches for the extraction of news events while knowledge-driven approaches have been applied to extract biomedical [3], historical [4], or financial events [5] among others.

Much previous research focuses on using the body text of the document, while some authors also use the document structure. For example, [4] apply semantic role labelling to unstructured Wikipedia text while [6] use both the document structure and body text to extract events from the same source.

The focus of this paper is on extracting future events using the body text of web pages as well as their DOM structure when the content has multiple levels of structure. We naturally use the body text from the web page as itthat contains essential information, e.g. time, date, and location instances. We also exploit the DOM structure as a source of information. Although HTML embeds some sort of structure, the actual structure is not homogeneous across websites. We report on the problem of extracting event information from a variety of web pages and we describe two systems we implemented and the results we obtained. .

1.1 Properties of Local Events

The events we are interested in are those that typically appear in calendars and listings, such as cultural, entertainment, educational, social, business (exhibitions, conferences), and sport events, that attract athe general and large public may have an interest in.

The end goal of this project is to be able to serve users with information about events that match their current interest and context, e.g. using location-based search, by aggregating these events from hyperlocal sources.

Event aggregators already exist, e.g. *Eventful* and *Upcoming*, that collect and publish event information, but they tend to only gather information about major events in cooperation with organizers or publishers. By contrast, we want to extract existing information directly from the publisher.

The main challenge is time-efficient scaling since there is a great number of hyperlocal organizers and sources as well as variations in the formats and DOM structure of the sources and ambiguity. We may also have to deal with missing, ambiguous, or contradictory information. For example, locations can appear in the title:

Concert – Bruce Springsteen (This time in the new arena),

and contradict the location indicated elsewhere. Another example is a title:

Outdoor dining now every Friday and Saturday

containing date information which narrows or sometimes contradicts the dates indicated elsewhere on the page.

FThe domain we are interested deals with future events form. This is a very wide area, where only few historically-annotated data is available. This makes a statistical approach problematic, at least initially. Instead, we chose a knowledge-driven, pattern-based approach, where we process both the structure of HTML documents and their content. We analyzse the content using knowledge of the event domain, e.g. event keywords.

In this paper, we report on the problem of extracting event information from given web pages and we describe two systems we implemented and the results we obtained.

1.2 Applications and Requirements for Event Structures

From the possible properties of an event, wWe chose to extract the *title, date, time, location, event reference (source)* and *publisher* which answers the *wWhen, where*, and *what* questions aboutof thean event. These are however the most basic attributes, and for a useful application, further information could be extracted, including topic, organizer, cost and target audience.

We set aside In this paper, we do not cover the semantic representation of event data, but future research may need to address representing the above attributes in existing event data models.

2 System Architecture

2.1 Designing a Simple Scraper

For each site in the list, we created a unique script. These scripts contained a hand-crafted set of rules to extract the correct information for that specific site. This may require a good deal of manual effort as we naturally have toTo expand the list of additional hand-crafted scripts is required, which leads to high costs when scaling to multiplemany sources..

In order to limit scaling costs, the scripts need to be simplistic. For this reason, we decided to A chosen limit ation was that the internal structure of the information in the events needs to be the same between each other, so that a small set of rules can extract the information from all the events.

2.2 Designing a Generic Scraper

We investigated if it would be possible to create a generic scraper which could handle all websites without manual labour.

The first step to generically scrape a website is to find all the pages that contain events. This is currently done using domain knowledge, i.e. the system is given only pages which are known to contain events. The possibilities to find pages without manual labour is further discussed in Sect. 5. The system uses six steps to scrape the events from a given web page. Figure 1 shows the system architecture. We implemented the first three steps using the ad-hoc scripts of Sect. 2.1.

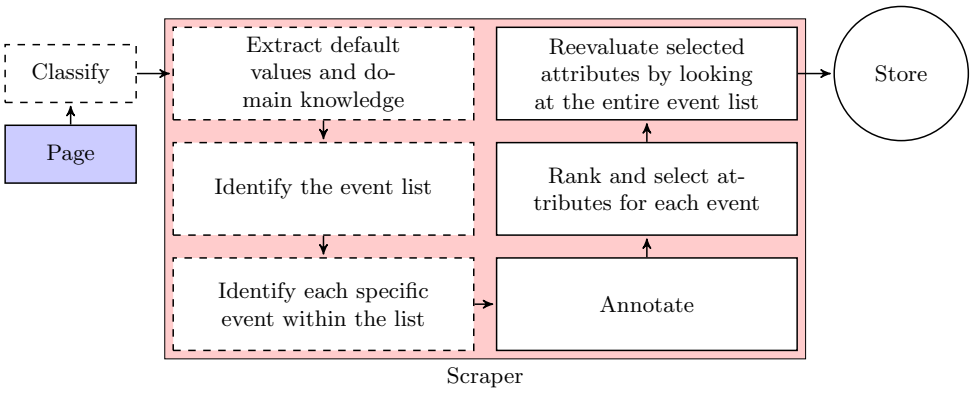

Fig. 1. The implemented generic scraper. Dashed boxes use manually written, site-dependent scripts.

2.3 Attribute Annotation and Interpretation

The system uses rules to annotate and interpret text. The benefit of a rule-based system is that it can both parse the text and create structured data. As previous work suggests, extracting the time and date of events can be solved through rules. While problematic, the system is able to extract named entities, for example named locations as well. To do this, the system uses three major rules:

1. Keyword detection preceding a named location, e.g looking for *location:* or *arena:*
2. Keyword detection succeeding a named location, for example a city
3. Structured keyword detection preceding a named location. e.g. look for *location* or *arena* when isolated in a separate structure. As an example: ***location*** *Boston* which corresponds to "`<b>location</b> Boston`" using HTML tags.

When the rules above return a named location, we query it against a named location database. Using these rules and a database lookup, we can minimize the false positives.

2.4 Attribute Ranking and Selection

The system uses domain knowledge to choose what data to extract:

- The system extracts only *one* title and chooses the most visually distinguished text it can, implied by the DOM structure
- Dates and times are following a hierarchy of complexity, where it takes those of highest complexity first. Some sites used a structure where event structures were grouped by date. To avoid false positives with dates in these event structures, the scraper choose dates between the event structures if less than half of the event structures contained dates.
- The extraction of the location for the event was done in the following order: If the event structure contained a location coordinate, choose it. Otherwise use a default location. If the event site had no default location, use the most commonly referred city in the event structure.

3 Evaluation

3.1 Scoring

We evaluated the performances of the simple and generic scrapers and we compared them with a scoring defined in Table 1.

Table 1. Criteria for full and partial scoring for the test set.

	Full match
Title	Lexicographic distance to correct = 0
Date	Resulting date(s) equal to correct date(s)
Time	Resulting start time equals correct start time (minute)
Location	Result within 1000 m of correct

	Partial match
Title	Result contains correct title
Date	Full match or if result contains at least one of correct date(s)
Time	Full match or if result contains at least one of correct start time(s)
Location	Result within 5000 m of correct

3.2 Training

At the start of the project, we gathered a training set composed of nine different event sites found in the Lund and Malmö area, Sweden. With the help of the training set, we could change the rules or add new ones and easily monitor their overall effect. This concerned both the rules of the annotator, scraper, and the location lookup.

3.3 Evaluation

In order to evaluate the system, we gathered a test set of nine, previously unseen, event web sites. The goal was to extract information about all (max. 30) events. The tests were conducted in three parts.

1. In the first part, we used the generic scraper (Sect. 2.2);
2. In the second one, we built simple scrapers (Sect. 2.1) for each of the test sites.
3. We extracted the events manually by hand in the third part.

The results from the first two parts were then compared against the third.

The generic scraper and the simple scrapers were compared in how accurately they extracted the title, date, time, and location of the event. The time of the setup was also compared for both the generic and simple scrapers.

We built a simple scraper for each site specifically to extract the text containing the title, date, time, and the location. The text strings containing the dates and times were then sent to the same algorithm that the generic scraper uses to parse the date and time. Once the text containing the location is extracted, we use the same location lookup in all the scrapers.

3.4 Bias Between the Training and Test Sets

The sites in the training set were all composed of a list with events where all the necessary information (title, date, time, location) could be found. In the

Table 2. F_1 score for full and partial match on test data for the generic scraper.

	Full					Partial				
Site	Title	Date	Time	Location	Average	Title	Date	Time	Location	Average
lu	0.0	0.967	0.767	0.433	0.542	0.4	0.967	0.933	0.633	0.733
mah	0.068	1.0	0.0	0.6	0.417	0.915	1.0	1.0	1.0	0.979
babel	0.0	0.818	0.0	1.0	0.830	1.0	0.909	0.818	1.0	0.932
lund.cc	1.0	0.667	1.0	0.652	0.714	1.0	0.967	1.0	0.652	0.905
möllan	0.0	0.857	1.0	1.0	0.75	0.0	0.857	1.0	1.0	0.714
nsf	1.0	1.0	1.0	0.0	0.673	1.0	1.0	1.0	0.286	0.822
malmö.com	1.0	1.0	0	0.691	0.543	1.0	1.0	0	0.963	0.741
burlöv	0.889	0.75	0.333	0.2	0.369	1.0	0.875	0.333	0.2	0.602
dsek	0.0	0.2	0.444	0.833	0.588	1.0	0.2	1.0	0.833	0.758
Average F_1	0.440	0.807	0.505	0.601	0.603	0.813	0.864	0.787	0.730	0.799

Table 3. F_1 score for full match on test data for the generic scraper without loading the event details page.

	Full				Partial			
Site	Title	Date	Time	Location	Title	Date	Time	Location
lu	1.0	1.0	0.967	N/A	1.0	1.0	0.967	N/A
mah	0.967	0.929	1.0	N/A	0.967	0.929	1.0	N/A
babel	0.0	0.0	N/A	1.0	1.0	0.0	N/A	1.0

Table 4. F_1 score for full and partial match on test data for the simple scraper.

	Full					Partial				
Site	Title	Date	Time	Location	Average	Title	Date	Time	Location	Average
lu	1.0	0.967	0.967	0.267	0.800	1.0	1.0	1.0	0.667	0.917
mah	1.0	1.0	0.0	0.7	0.675	1.0	1.0	1.0	1.0	1.0
babel	0.0	0.7	0.211	1.0	0.478	1.0	0.7	0.632	1.0	0.833
lund.cc	1.0	0.667	1.0	0.622	0.822	1.0	0.967	1.0	0.622	0.897
möllan	0.857	0.667	1.0	1.0	0.881	1.0	0.833	1.0	1.0	0.959
nsf	1.0	1.0	1.0	0.0	0.75	1.0	1.0	1.0	0.0	0.75
malmö.com	1.0	1.0	0.0	0.823	0.706	1.0	1.0	0	0.912	0.728
burlöv	1.0	1.0	0.0	0.0	0.5	1.0	1.0	0.0	0.0	0.5
dsek	0.952	0.706	0.778	1.0	0.859	0.952	0.706	0.889	1.0	0.887
Average F_1	0.868	0.856	0.551	0.601	0.719	0.995	0.912	0.725	0.689	0.83

Table 5. Time taken for the setup for the test sites.

Site	Generic	Simple	Manual
lu	23 min	83 min	60 min
mah	7 min	24 min	68 min
babel	11 min	59 min	15 min
lund.cc	9 min	13 min	60 min
möllan	2 min	31 min	13 min
nsf	5 min	24 min	15 min
malmö.com	31 min	63 min	35 min
burlöv	10 min	30 min	22 min
dsek	11 min	23 min	21 min
Average	12 min	39 min	34 min

test set, most of the sites had a structure that did not have all the required information: Each event had a separate page with all the information, the event details page. The information on the event details page was not composed of the typical compact structured form but rather had more body text. Of the nine sites in the test set, three sites (lund.cc, nsf, dsek) did not require an event details page for the necessary information. But the information on the sites nsf and dsek were in their structure more comparable to a body text. A concept to handle this is presented in Sect. 4.1 that concerns the extraction of the title.

4 Conclusion

The setup for the generic scraper took on average 12 minutes, compared to creating a simple scraper for each site that took on average 39 minutes (Table 5). The setup for the generic scraper is more than three times faster than creating a simple scraper for each site. This can be compared to the pure manual labor which took on average 34 minutes per site, thus both scrapers essentially have a pay back time of one pass.

4.1 Title

The generic scraper performs rather poorly on the test set while it shows better results on the training set. This is either due to a training overfit or a significant mismatch between the training and test sites. Sect. 3.4 analyzes the mistakes and discusses this problem. When using the system on these pages without loading, they do yield better results, as shown in Table 3. The rest of the failing test sites failed because the system looked to much in the structure where it should have analyzed the layout instead, i.e. it chose links when it should have chosen the ones which were more visually prominent.

4.2 Date

The simple scraper is 5% better on the date identification than the generic scraper on average for both the full and partial matches. Examining the scores

for the full match more closely, (Tables 2 and 4), the score for the generic is the same or better than the score for the simple scraper for every site except burlöv and dsek. We even observe a complete failure for dsek. We investigated it and we discovered that dsek expressed the dates relative to the current date e.g. *today*, *tomorrow*. This wasn't implemented yet which made the generic scraper pick another strategy for picking dates, as a result the correct dates were forfeited.

4.3 Time

The average scores for the time extraction between the generic and the simple scrapers are rather similar. The system does finds the correct times but does report many false positives, which according to the scoring set in Sect. 3.1 yields only a partial match. The system tends to over detect times. We programmed it to prefer times coupled with dates over solitary times but in the test set, it seems it was rather common to have time and dates further apart. This makes the system choose all times, where it should have chosen a subset. Another pattern was also found: for some sites, the system returned both start and end time separately which shows that the system is lacking rules to bind start and end times together.

4.4 Location

The difference between simple and generic scraper is negligible and the problem of location is less about selection and more about actually find and understand the named locations (Tables 2 and 4). The system uses assumed knowledge to fill in what is left out of the events, i.e. knows city, region or location which it can use to fallback to or base the search around. Using this assumed knowledge has proved useful when looking at babel, möllan, dsek, lu and mah and this should hold true on all hyperlocal websites. Even if the system has some basic knowledge about the web page, the location annotation and selection still has problems with disambiguation. This disambiguation problem is partly rooted in the fact that the named locations are within the domain knowledge of the site. As an example, a university website might write lecture halls or class rooms as the location of the event. These named locations could have the same name as pub in another city, a scientist or simply nonexistent in any named location database.

4.5 Final Words

At the end of the test cycle, however, we considered that an generic scraper is not only possible to do, but in some cases even better than a simple one. The hardest problem with scraping sites is not necessarily to understand the structure, even if vague. The problem for a scraper is rather to understand what can only be described as domain knowledge. Sites uses a lot of assumed knowledge which can be hard to understand for a machine or even if its understanding could be

completely wrong in the context. For example, lecture halls can be named the same as a pub in the same region, making it hard for a system to determine if the location is correct or not. This might be attainable with better heuristics, e.g. if the location lookup can be made with some hierarchical solution and domain knowledge can be extracted from the sites prior to the extraction of events.

5 Future Work

5.1 Page Classification

On the Internet, sites show a significant variation and most of them do not contain entertainment events. Therefore a first step in a generic system, the dashed box "Classify" in Figure 1, would be to identify *if* the input web page contains events. If it does not, it makes no sense to scrape it and doing so could even lead to false positives. If web pages could be classified with reasonable certainty, it could also be used with a crawler to create an endless supply of event pages to scrape.

5.2 Exploring Repetitiveness

To solve the dashed box "Identify the event list" shown in Figure 1, we investigated the repetitiveness of the event list. With the help of weighing in structural elements, e.g. P, STRONG, H3, it yielded some interesting results on small sites. This technique can potentially be further refined by calibrating weights if the page is annotated using what is described in Sect. 2.3.

5.3 Rank and Select with Help of Layout

While the system uses a very limited rank and selection based on an implied layout for title (prefer H3, H2 etc. over raw text), it would be interesting to have the selection fully use layouts. To attract attention and to create desire, the vital information about an event are among the first things the reader is supposed to notice and comprehend. Thus it is usually presented in a visually distinguishing way. This can be achieved by coloring the text differently, making it larger, or simply in a different font or typing. This layout is bundled within the HTML document, possibly modified by the CSS, thus looking at these clues with some heuristics allows to find the visually distinguishing sentences [7]. As an example, an event might use a H3 element for the title, bold for the location, or it might have another background color for the date. If the entire system would use layout to aid the selection we believe that the system will perform better and will yield less false positives.

References

1. Hogenboom, F., Frasincar, F., Kaymak, U., de Jong, F.: An Overview of Event Extraction from Text. In van Erp, M., van Hage, W.R., Hollink, L., Jameson, A., Troncy, R., eds.: Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at Tenth International Semantic Web Conference (ISWC 2011). Volume 779 of CEUR Workshop Proceedings., CEUR-WS.org (2011) 48–57
2. Liu, M., Liu, Y., Xiang, L., Chen, X., Yang, Q.: Extracting key entities and significant events from online daily news. In Fyfe, C., Kim, D., Lee, S.Y., Yin, H., eds.: Intelligent Data Engineering and Automated Learning - IDEAL 2008. Volume 5326 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2008) 201–209
3. Chun, H.w., Hwang, Y.s., Rim, H.C.: Unsupervised event extraction from biomedical literature using co-occurrence information and basic patterns. In: Proceedings of the First international joint conference on Natural Language Processing. IJCNLP'04, Berlin, Heidelberg, Springer-Verlag (2005) 777–786
4. Exner, P., Nugues, P.: Using Semantic Role Labeling to Extract Events from Wikipedia. In van Erp, M., van Hage, W.R., Hollink, L., Jameson, A., Troncy, R., eds.: Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at Tenth International Semantic Web Conference (ISWC 2011). Volume 779 of CEUR Workshop Proceedings., CEUR-WS.org (2011) 38–47
5. Borsje, J., Hogenboom, F., Frasincar, F.: Semi-automatic financial events discovery based on lexico-semantic patterns. Int. J. Web Eng. Technol. **6**(2) (January 2010) 115–140
6. Hienert, D., Luciano, F.: Extraction of historical events from wikipedia. In: Proceedings of the First International Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data, CEUR-WS.org (2012)
7. Cai, D., Yu, S., Wen, J.R., Ma, W.Y.: Extracting content structure for web pages based on visual representation. In: Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications. APWeb'03, Berlin, Heidelberg, Springer-Verlag (2003) 406–417

Automatic Extraction of Soccer Game Events from Twitter

Guido van Oorschot, Marieke van Erp[1], and Chris Dijkshoorn[1]

The Network Institute
Department of Computer Science
VU University Amsterdam
{marieke.van.erp,c.r.dijkshoorn}@vu.nl

Abstract. Sports events data is often compiled manually by companies who rarely make it available for free to third parties. However, social media provide us with large amounts of data that discuss these very same matches for free. In this study, we investigate to what extent we can accurately extract sports data from tweets talking about soccer matches. We collected and analyzed tweets about 61 Dutch premier league soccer matches. For each of these matches we 1) extracted the minutes in which an event occurs, 2) classified the event type and 3) assigned events to either the home or away team. Our results show that the aggregation of tweets is a promising resource for extracting game summaries, but further research is needed to overcome data messiness and sparsity problems.

1 Introduction

Soccer is a highly popular game, and with it information about soccer matches played. Many soccer fans try to keep track of their favorite teams by reading or watching game summaries. Generally, these summaries provide an overview of the minutes in which game highlights as goals, cards, and substitutions happen for both teams. This type of data is often created manually, a time-consuming and expensive process. Companies make good money off selling these data to third parties. However, the rise of vast amounts of data on social media platforms such as Twitter[1] is drawing the attention of the research community. [1] for example, mine Twitter to detect earthquakes and notify people more quickly and accurately than conventional methods are able to. [2] predict stock prices from analysing sentiment in tweets about stock tickers. Twitter is also a beloved medium for sports fans, during matches they often tweet about their teams and what is happening in the match. Preliminary work to extract useful information about sport matches from tweets has been carried out by [3]; they were able to successfully extract certain types of events from soccer and rugby games by analysing the number of tweets per minute. In this contribution, we build upon this work and present an approach to construct soccer match summaries from

[1] http://www.twitter.com

tweets by detecting 'event minutes' from the Twitter stream, classifying them and assigning them to a team.

Although individual tweets are rather short, our experiments show that there is enough information contained in the aggregate of tweets around soccer matches to extract game highlights. Our results are not perfect yet, but we show that this 'free' community generated data has the potential for use in automatically generated game summaries instead of relying on expensive third parties. In the remainder of this contribution, we first present related work in Section 2, a description of our data set in Section 3, our game event detection experiments in Section 4, game event classification in Section 5, and the assignment of teams to events in Section 6. We conclude with a discussion of our findings and pointers for future work (Section 7).

2 Related Work

The majority of research into automated event detection in sports games is aimed at extracting highlights from audio and video content. In [4,5] the level of excitement of the sports commentator and game-specific sounds are used to extract highlights. Video analysis to extract highlights has been performed for soccer [6], tennis [7] and other sports [8] with varying success. Audio and video analysis are computationally expensive and often an event can be detected, but not classified. Some approaches pair the video signal with a textual source such as a minute-by-minute report [9], but such reports still require human input.

Recently, crowdsourced data has gained interest to leverage this problem. [10] present a mobile application in which users could annotate events in a soccer game. Results showed that the number of annotations increased around important events in a game, but people still needed to make a special effort to use the application. On Twitter, people are already discussing the game. Realising this, [3] set out to use this data to mine tweets for highlight detection in soccer and rugby matches. They employed a fairly simple approach detecting 'interesting minutes' by looking at the peaks in the Twitter stream. Their results are comparable to highlight detection from audio and video signals, but still suffer from a high number of false positives. We aim to improve on this work by employing smarter peak detection, machine learning to classify the events and enriching the event information by assigning it to a team. Doing this, we aim to leverage the information embedded in tweets and develop a automatic system that can extract cheap, crowdsourced soccer event data with accuracies that rival expensive, manually created data.

3 Data

In this section, we detail the data collection and preprocessing steps.

3.1 Collecting Soccer Tweets

For this study, we considered two approaches in collecting data about a certain topic using Twitter. The first method of collecting tweets focuses on tweets by people who are knowledgeable about a certain topic, while the second method focuses on tweets explicitly related to a certain topic by collecting them based on keyword occurrence. In [11] both methods were investigated and showed that collecting tweets by people knowledgeable about a certain topic proved to be susceptible to external noise. Their research demonstrated, in line with findings in [12], that using hashtags was the most effective way to gather tweets around a particular topic.

In our domain, we found that by convention hashtags are created that consist of abbreviations of club names for each soccer game, starting with the home team. Tweets about the game of Ajax against Feyenoord for example will thus contain the hashtag $\#ajafey$. This convention enabled us to easily develop a scraper that would search for the game hashtags. We ran the scraper from the beginning of the Dutch premier soccer league in December 2011 to the end of the season in May 2012. The scraper was written in Python using the Tweetstream library[2] and embedded in a Django application using a PostgreSQL database.

3.2 Gold Standard

We also collected the official Dutch Premier League sports data for each game. The format of this official soccer game data is a report of the minutes in which events in a game happened and contains the following 5 classes of events: goal scored, own goal scored, yellow card, red card, and player substitution. In this data, we found a total number of 700 events. In 39 minutes, multiple events occurred at the same time. For simplicity's sake, we only want each minute to belong to one class of event. To this end, we devised the following hierarchy of importance of events (from important to less important): goals, own goals, red card, yellow card, substitution. For example, if in a minute a goal is scored and a yellow card given, this minute will be of class goal. This resulted in 169 goal minutes, 2 own goal minutes, 18 red card minutes, 187 yellow card minutes, and 285 substitution minutes in our gold standard.

3.3 Data Preprocessing

In the period the scraper was deployed, 156 games have been put into the database to be scraped. From these games, 18 games could not be tracked due to unavailability of the scraper. Of the remaining 138 games that were tracked, 2 games turned out to have an erroneous hashtag. For further analysis we used the remaining 136 games and a total of 1,050,434 associated tweets.

Tokenization We removed punctuation except for the # which indicates a hashtag. All letters were transformed to lowercase and words were separated

[2] `http://pypi.python.org/pypi/tweetstream/`

based on whitespace. Common Dutch words were removed using the stopword list from the Python Natural Language Toolkit[3]. Hyperlinks, mentions of other Twitter users, and the presence of a score (e.g., 1-0) were converted to a binary value indicating their presence in the tweet.

Outlier Detection We calculated the average number of tweets per game for each team. It is no surprise that Ajax Amsterdam is the most popular team. A low number of tweets for Roda JC Kerkrade can be explained by the fact that people started using a different abbreviation for the Roda JC games halfway during the competition: from *rjc* to *rod*.

Missing Values Due to the limitations of the Twitter streaming API, our scraper had problems processing large numbers of tweets coming in when games between large teams were being played and many people were tweeting at the same time. We manually analyzed every game by looking at the number of tweets per minute figures and decided to leave out the games in which gaps were visible. The removal of these corrupted games left 63 games and 326,487 tweets included for further analysis.

Multiple Game Hashtags Oftentimes tweets refer to all the games played during the day or weekend or summarize the results of different games, as the following example translated from Dutch shows: *"Enjoying a nice day of football. #psvhee #twefey #adoaja #aznac"*.[4] We excluded tweets containing multiple game hashtags in a four-day window around a game from our analysis, as these are most likely a week summary. This resulted in the removal of 10.643 tweets.

Aligning Start Times Many games do not start exactly on time but a few minutes late. Our scraper would already start collecting tweets 15 mins prior to each game, and we tried to identify the actual starting times by looking for tweets with the terms "has started"[5] in a 10-minute window around the officially designated starting time. After some experimenting we decided to select the first minute with a tweet count higher than 50% of the peak amount as the first minute of a half. We analyzed the results for both halves of 10 randomly selected games. The starting minutes of 14 halves were correct, for 5 halves the actual time is 1 minute later or earlier and for only 1 half no starting time could be extracted.

4 Game Event Detection

As our gold standard reports events by the minute in which they occurred, we also study tweet volume in one-minute intervals. In our case, the signal in which we want to detect peaks is the number of tweets per minute for each minute in a soccer match.

As the tweet volume differs per game, it is not possible to set a threshold to detect event minutes. We found that automated Twitter accounts and spam bots talk about matches without regarding the specific events in a game, creating a

[3] `http://nltk.org`
[4] Tweet: "Genieten van een mooi dagje voetbal. #psvhee #twefey #adoaja #aznac"
[5] In Dutch: "is begonnen"

Table 1. Overall average number of event minutes selected per game (# min), precision and recall for different peak selection methods. Recall per event class (goal, own goal, red card, yellow card, and player substition)

		Overall			Per Event Class				
		# min	precision	recall	goal	own	red	yellow	sub
LocMax-NoBase-LineCorr	peak	15.46	0.180	0.257	0.408	0.500	0.333	0.176	0.214
	peak +/- 1	46.38	0.145	0.619	0.805	1.000	0.722	0.487	0.586
	peak +/- 2	74.15	0.122	0.832	0.935	1.000	0.889	0.775	0.804
	peak +/- 3	88.46	0.113	0.921	0.976	1.000	0.944	0.893	0.905
IntThresh-NoBase-LineCorr	peak	10.02	0.268	0.248	0.586	0.500	0.389	0.112	0.126
	peak +/- 1	29.03	0.181	0.486	0.917	1.000	0.833	0.283	0.337
	peak +/- 2	42.90	0.154	0.610	0.970	1.000	0.889	0.471	0.467
	peak +/- 3	52.90	0.137	0.667	0.970	1.000	0.889	0.519	0.568
IntTresh-WithBase-LineCorr	peak	8.00	0.291	0.215	0.580	0.500	0.222	0.080	0.084
	peak +/- 1	23.23	0.188	0.402	0.888	1.000	0.722	0.193	0.228
	peak +/- 2	34.62	0.158	0.504	0.948	1.000	0.778	0.332	0.333
	peak +/- 3	43.64	0.144	0.554	0.948	1.000	0.778	0.385	0.414

certain baseline noise of tweets around a match. Also, as [3] found, towards the end of a game overall tweet activity was higher, making peak selection more difficult - baseline correction might help us avoid this problem too. We therefore investigated three different peak detection methods.

The first setting of peak detection will use no baseline correction and takes local maxima selection as peak picking method (LocMaxNoBaseLineCorr). For every minute we check if it is a local maximum of a window of two neighboring minutes. If so, we will select this minute as being a peak. The second setting also has no baseline correction and uses the intensity threshold method of peak picking (IntThreshNoBaseLineCorr). This method looks at the difference in levels between different minutes and decides a minute is a peak when its change in volume compared to the previous minute(s) is higher than a certain threshold. In the third setting we also use this intensity threshold measure for picking peaks, but we also apply baseline correction to the tweet volume per minute signal (IntThreshWithBaseLineCorr).

Additionally, taking only the peak minute as instance might be inaccurate. If for example a goal is scored at the end of a game minute, the peak in tweets about that goal will be apparent in the minute after which the goal is officially scored. We therefore also experiment with a window of 1 to 3 minutes around the peak candidate to correct for a lag in the Twitter stream.

The baseline to which we compare our peak detection settings is the recall and precision levels for taking all minutes of a game. An analysis of our gold standard data shows that in that case, recall is 1 and precision is 0.108. This means that on average in 10.8% of the minutes of each games an event happens (σ=2.59).

Table 1 shows the average number of minutes selected per game, overall precision, overall recall, and recall per class measures for our peak selection method

with varying windows of extra minutes around the peaks that are selected. The results show a clear trade-off between recall and precision. For all three peak selection methods and windows, higher recall gives lower precision. The goal of our analysis is to select minutes in which an event occurs, so we want to increase precision while still keeping acceptable levels of recall compared to the baseline of taking all minutes (recall > 0.9).

From an in-depth analysis of five randomly selected games, we found that there are four main reasons for the peak selection to not achieve perfect precision. The first is that there are some errors in the starting time alignment (see also Subsection 3.3). For example, in the FEY-NAC game, four goals were scored, and because our start time selection is off one minute, our peak selection is also off exactly one minute. Proliferation of errors is a typical problem for any sequential approach, and it shows that our peak selection method can only work if the starting time is known. The second problem arises in games that are not very popular to tweet about, this indicates that a certain amount of data is needed for the approaches to work reliably. The third problem is that there is a lag between when an event happens and when Twitter messages are sent.The fourth problem for our peak detection method is that the Twitter users do not only comment on the five classes of events we have defined, but also on other events. In 29% of the selected peak minutes no event from our five classes occurs in a 5-minute window around the peak. In these minutes, people often comment on the ending of a half or exceptional events such as particularly nice shots or abnormal supporter activity. Although these events do not occur in our classification, they are legitimate events that may be interesting to include in a match summary, for now they slightly taint our results.

5 Game Event Classification

After finding out when events happened in a game, we also want to know what kind of event it is. As mentioned in Subsection 3.2, we distinguish between 5 types of events: goal scored, own goal scored, yellow card, red card, and player substitution.

To classify the type of event, we used a machine learning approach. Per game, we created feature vectors for each minute, using the words occurring in the tweets, as well as presence of hyperlinks, mentions of other Twitter users, and score patterns (see also Subsection 3.3). As this resulted in over 18,000 features, for a total of 6100 game minutes, we experimented with several different feature selection methods based on word frequency, information gain, and gain ratio. From our preliminary experiments, we found that the feature set based on gain ratio produced the best results. In this setting we included the 50 words with the highest normalized information gain.

As in only 10% of the game minutes an event takes place, we also investigated how the different feature sets perform if we include all game minutes or a pre-selected set of minutes in which the proportion of minutes in which nothing happens is smaller. We experimented with three sets of game minutes: All

MINUTES no filtering is performed (Baseline), PEAK MINUTES only minutes from the best performing peak selection method from Subsection 3.3 are included and EVENT MINUTES only minutes in which an event takes place are included.

As we did not know which algorithm would perform best on our data set, we started our exploratory search with a set of different types of algorithms using the Weka toolkit[6] and the LibSVM library[7]. All experiments were carried out using 10-fold cross-validation. Due to lack of space we will only discuss the best performing settings below[8].

Table 2. F-scores for SVM classifier on All Minutes, Peak Minutes selection, and gold standard Event Minutes using Top50GainRatio feature set

	Goal	Own Goal	Red	Yellow	Sub	Nothing	Overall
All Minutes	0.466	0.000	0.052	0.000	0.000	0.948	0.859
Peak Minutes	0.696	0.000	0.444	0.000	0.000	0.877	0.759
Event Minutes	0.841	0.000	0.848	0.785	0.839	n/a	0.822

In Table 2 we show the results on the different feature sets of the best classifier, SVM. As can be seen from the comparison with the ALL MINUTES setting, only typing the event minutes gives a fair boost in the results. As expected, we encountered the problem of dealing with imbalanced data in our ALL MINUTES baseline. In our experiment we explored a way of dealing with this problem: by more accurately selecting event minutes with the PEAK MINUTES setting, the performance of the classifiers increased.

Narrowing the instance selection to only include event minutes, we found that the classifiers could rather accurately classify the event minutes. Overall, goals, red cards, and substitutions could be classified best, followed by yellow card events. Own goals were not classified accurately because only two instances of such events existed in our data set (in combination with the large overlap in words describing both goals and own goals).

Inspecting the feature sets these methods created validated our expectations. A large number of words found in these sets are words such as "goal", "yellow card", "red card", "substitution" or synonyms and variations on those. Additionally, a considerable number of words in these sets are either curse words or words expressing positive excitement.

6 Team Assignment

In the previous experiment, we were able to classify with fair accuracy what type of event took place in a certain minute in a game, but it is yet unknown

[6] `http://www.cs.waikato.ac.nz/ml/weka/`

[7] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[8] An overview of all experiment settings and results, as well as our tweet corpus can be found at `http://semanticweb.cs.vu.nl/soccertweet`

which team scored the goal or received the card. In this third experiment, we investigate assigning a team to an event based on the team's fanbase.

In studying tweets around American football matches [13] found that during a match the tweet activity from the home and away teams differed; sports game are experienced differently for fans of each team. Tang and Boring assumed that a tweet with only one team name in it indicated that the writer was a fan of this team, however, we believe that this is too simplistic. All of our tweets for example contain two team names, and [14] found that in about 10% of the tweets a negative sentiment is expressed towards a team. Therefore, to determine a user's favorite team, we assume that this person tweets about this team over the course of multiple matches.

There are 147,326 Twitter users who have contributed to our soccer tweet collection, but in order to decide if someone is a fan of a particular soccer team, we only consider users of whom we have at least four tweets in our collection. This leaves us with 44,940 users (31%). For each of these, we extracted all the unique matches they tweeted about, and we counted how often they tweeted about a match of a particular team. We then ranked the teams based on the number of times they were mentioned by the user. To evaluate our approach, we annotated a stratified sample of 360 Twitter users (20 per team) with their real favorite teams as based on manual inspection of their tweets. By this approach, we can correctly assign the Twitter user's favorite team in 88% of the cases, in 10% of the cases we could not make out the favorite team of the Twitter user and in 2% of the cases our approach was incorrect.

We can now use the fan distribution to try to assign events to a team. We do this by testing a logistic regression model. Table 3 shows that we can accurately assign goals and red cards to the home or away team. Yellow cards and substitutions are more difficult, this is not surprising as there is less Twitter activity around these events.

Table 3. Assigning teams to events overall and per-class performance. Baseline percentage and performance as % correctly classified.

	Goal	Red	Yellow	Substition	Overall
Baseline	58	50	63	52	52
Regression	68.73	61.13	63.23	56.54	57.76

The results of this experiment indicate that our method of extracting users' favorite teams can be used to learn what team an event belongs to. We found that fans to some extent will tweet more in the minute of an event from their team. However, the results are not accurate enough to reliably assign events to teams. One reason can be that we only checked the ratio in one particular minute. Accuracy might be improved by including some minutes around the event minute. Our set of fans is also not complete as we could only assign teams to 26% of the Twitter users in our corpus. By increasing the number of tweets we can assign to fans, the difference between events may become clearer. People

from the team an event does not belong to still tweet about it, but in a more negative way, which also affects the accuracy. As we found a fair portion of sentiment (both positive but also negative, as in curse words) it is fair to assume that users with a fair dislike of a particular team may taint our results. Including sentiment analysis in our approach is therefore at the top of our priority list.

7 Discussion and Future Work

In this contribution, we presented experiments and results for detecting the most important events occurring in soccer games through mining soccer tweets. We take a three step approach in which we first try to detect interesting minutes, then classify the type of event that occurs in this minute, and finally we assign the team. Our approach is novel in that it relies entirely on user created data and takes it a step further than previous work by assigning teams to events. Our approach does have its drawbacks, for example in less popular games there were too few tweets to reliable detect the interesting event minutes. Our approach is also limited in that it cannot detect two events happening in the same minute, and that the lag of tweets is not taken into account, thus if an event took place at the end of a minute, we only detect it in the next minute. Furthermore, we cannot yet determine the favourite team of the majority of Twitter users, which makes our data to assign events to teams too sparse.

In future work, we will focus on mitigating these problems by integrating the different steps in the approach more; currently, we have taken a very sequential approach. The event detection task for example relies solely on changes in tweet volume, while it may benefit from knowing what kind of words occur in tweets around events. To better classify favourite teams of Twitter users we are planning to look into sentiment analysis. This may also give us some help in classifying events that are currently out of our scope as they do not occur in our gold standard, but nonetheless may be interesting, such as particularly stunning passes or curious supporter activity. Additionally, future research should investigate the applicability and generalizability of this study's methods in other languages and soccer competitions as well as other sports.

On the longer term, we aim to integrate our automatically extracted events into applications that currently have to rely on expensive manually created data, such as websites simply showing textual information about a game or systems that automatically generate personalized visual summaries based on a video feed in combination with user preferences. With the advent of social media, everyone can be a content provider. As our techniques for mining this user-generated content improve and the extracted information approaches the quality of commercially available data sets, we may see a change in how sports and other once proprietary data is provided. The rules of the game have changed.

Acknowledgements

This work has been carried out as a part of the Agora project and the SEALINCMedia project. The Agora project is funded by NWO in the CATCH programme, grant 640.004.801. The SEALINCMedia project is funded by the Dutch national program COMMIT.

References

1. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web. WWW '10, New York, NY, USA, ACM (2010) 851–860
2. Bollen, J., Mao, H., Zeng, X.: Twitter mood predicts the stock market. Journal of Computational Science **2**(1) (2011) 1 – 8
3. Lanagan, J., Smeaton, A.F.: Using twitter to detect and tag important events in live sports. In: Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM'11), Barcelona, Spain (2011) 542–545
4. Rui, Y., Gupta, A., Acero, A.: Automatically extracting highlights for tv baseball programs. In: Proceedings of ACM Multimedia. (2000)
5. Xu, M., Maddage, N.C., Xu, C., Kankanhalli, M., Tian, Q.: Creating audio keywords for event detection in soccer video. In: Proceedings of the 2003 International Conference on Multimedia and Expo (ICME'03). (2003) 281–284
6. Qian, X., Liu, G., Wang, H., Li, Z., Wang, Z.: Soccer video event detection by fusing middle level visual semantics of an event clip. In: Advances in Multimedia Information Processing (PCM 2010). (2010) 439–451
7. Kijak, E., Gravier, G., Gros, P., Oisel, L., Bimbot, F.: HMM based structuring of tennis videos using visual and audio cues. In: Proceedings of the 2003 International Conference on Multimedia and Expo (ICME'03). (2003)
8. Sadlier, D.A., O'Connor, N.E.: Event detection in field sports video using audio-visual features and a support vector machine. IEEE Transactions on Circuits and Systems for Video Technology **92**(2-3) (2005) 285–305
9. Xu, C., Wang, J., Wan, K., Li, Y., Duan, L.: Live sports event detection based on broadcast video and web-casting text. In: Proceedings of the 14th annual ACM international conference on Multimedia (MULTIMEDIA'06). (2006) 221–230
10. Shirazi, A.S., Rons, M., Schleicher, R., Kratz, S., Müller, A., Schmidt, A.: Real-time nonverbal opinion sharing through mobile phones during sports events,. In: Proceedings of the 2011 annual conference on Human factors in computing systems (CHI'11). (2011) 307–310
11. Wagner, C., Strohmaier, M.: The wisdom in tweetonomies. In: Proceedings of the 3rd International Semantic Search Workshop on - SEMSEARCH '10, ACM Press (2010)
12. Laniado, D., Mika, P.: Making sense of twitter. In: Proceedings of the 9th International Semantic web Conference (ISWC 2010), Shanghai, China (November 2010)
13. Tang, A., Boring, S.: #EpicPlay: Crowd-sourcing sports video highlights. In: Proceedings of SIGCHI Conference on Human-Factors in Computing Systems (CHI 2012), Austin, TX, USA (May 2012) 1569–1573
14. Zhao, L.Z., Wickramasuriya, J., Vasudevan, V.: Analyzing twitter for social tv: Sentiment extraction for sports. In: Proceedings of the 2nd International Workshop on Future of Television. (2011)

Harnessing Disagreement for Event Semantics

Lora Aroyo[1,2], Chris Welty[2]

[1] VU University Amsterdam
lora.aroyo@vu.nl
[2] IBM Watson Research Center
cawelty@gmail.com

Abstract. The focus of this paper is on how events can be detected & extracted from natural language text, and how those are represented for use on the semantic web. We draw an inspiration from the similarity between crowdsourcing approaches for tagging and text annotation task for ground truth of events. Thus, we propose a novel approach that harnesses the disagreement between the human annotators by defining a framework to capture and analyze the nature of the disagreement. We expect two novel results from this approach. On the one hand, achieving a new way of measuring ground truth (performance), and on the other hand identifying a new set of semantic features for learning in event extraction.

1 Introduction

Events play an important role in human communication. Our understanding of the world is transferred to others through stories, in which objects and abstract notions are grounded in space and time through their participation in events. In conventional narrative, these events unfold sequentially in a timeline. Upon inspection, however, our understanding of events is quite difficult to pin down. This can be seen in metaphysics, where theories range from events as the most basic kind of entity in the universe to events as an unreal fiction [1], and in Natural Language Processing (NLP), where the few annotation tasks for events that have been performed have shown very low inter-annotator agreement.

One of the simplest and most prevalent ontological views of the universe is that there are two basic kinds of entities, objects and events. They are distinguished in that events *perdure* (their parts exist at different time points) and objects *endure* (they have all their parts at all points in time) [2]. The distinction is sometimes phrased ”objects are wholly present at any point in time, events unfold over time.” This definition and distinction is not universally held, but it serves us here as a convenient reference point; we believe the conclusion holds regardless of the ontological status of events.

The importance of events and their interpretation is widely recognized in NLP, but solutions remain elusive, whereas NLP technology for detecting objects (such as people, places, organizations, etc.) in text has reached ”off the shelf” levels of maturity. In addition, there is comparatively little annotated data for training and evaluation of event detection systems, and the bulk of what is available is difficult to reproduce. Annotator disagreement is quite high in most cases, and since many believe this is a sign of a poorly defined problem, guidelines for these event annotation tasks are very precise in

order to address and resolve specific kinds of disagreement. This leads to brittleness or over generality, making it difficult to transfer annotated data across domains or to use the results for anything practical.

One of the reasons for annotator disagreement is that events are highly compositional in the way they are described in language. Objects are compositional, too, but only in reality – in language we rarely refer to the parts of the object, only to the object itself. For events, we often describe where and when they take place, who or what the participants were, what the causes or results of the event were, and what type of event it was. More importantly events are usually referred to through their parts, e.g. we might talk about a terrorist event by using the word "explosion", which literally refers to only a small part of the overall event, making it sometimes difficult to determine whether two parts of one event refer to the same thing.

This highly compositional nature means that there are more potential ways in which two human annotators can disagree about a single event. Since agreement is never perfect for any annotation task, the agreement for a composite annotation task will necessarily degrade as the product of the agreement for the sub-tasks. In other words, if events are taken to be a time, place, actor, patient, and type, the agreement for the event task will be the product of the agreement on the five sub-tasks, which would be low since agreement for any task is between 0 and 1.

In our efforts to study the annotator disagreement problem for events, we began to realize that the disagreement didn't really change people's understanding of a news story or historical description. People seem to live with the vagueness of events perfectly well; the lack of precision and identity in event detection began to seem like artificial problems. This led us to the hypothesis of this paper, that the kind of annotator disagreement we see is a natural state, and that event semantics, both individual and social, is by its very nature imprecise and varied. We propose to harness this by incorporating disagreement as parameter of the annotated meaning of events using a crowdsourcing approach, which allows for capturing the wide range of interpretations of events with a minimal requirement for agreement (only for e.g. spam detection). We can then use a form of semantic clustering by defining a similarity space not of lexical features of language, but of dimensions that come from a classification of human disagreement on event interpretation.

In this preliminary work we present the classification framework and annotation task, and describe how it will be used for event detection. This work is performed in the context of the DARPA's Machine Reading program (MRP)[3]

2 Classification Framework

Our classification of the multitude of event perspectives derives from, and forms the basis for understanding, the disagreement in the crowd-sourced event annotation task, and we use it further to define similarity between events identified by the annotators. Methodologically, the initial set of classifications in the framework were produced by observing disagreement in previous annotation tasks, and we expect to further extend and refine the set as we conduct new annotation tasks.

[3] http://www.darpa.mil/Our_Work/I2O/Programs/Machine_Reading.aspx

We identify three high-level views to disagree on the annotation of events:

- *ontology*: disagreements on the basic status of events themselves as referents of linguistic utterances, for example are people events or do events exist at all.
- *granularity*: disagreements that result from issues of granularity, such as the location being a country, region, or city, the time being a day, week, month, etc.
- *interpretation*: disagreements that result from (non-granular) ambiguity, differences in perspective, or error in interpreting an expression, for example classifying a person as a terrorist or hero, the "October Revolution" took place in September, etc.

2.1 Ontology Disagreements

We do not address ontological disagreements on events in this paper, and we assume annotation tasks to be defined by a particular ontology. The literature and history of event ontology is vast, see [1] for a good start. We assume for the purposes of this framework that events do exist (it is a particular ontological position that they don't), that they are located in space, occur over some time, have a prescribed type, have temporal parts, and have participants. This gives us five dimensions in which to classify possible annotator disagreements (space, time, classification, composition, and participation).

2.2 Granularity Disagreements

We consider disagreements on levels of granularity to be, for the most part, agreement about what the event refers to but disagreement about what level of detail is important to extract and identify the event.

- *Spatial* granularity disagreements occur when the location can be specified at sizes within some regional containment. If a sentence said, "...a bombing in a downtown Beirut market..." the event might have taken place in "downtown Beirut", "Beirut", even "Lebanon" or "Middle East". Each is correct, but typical gold standards define only one to be.
- *Temporal* granularity disagreements occur when the time can be specified at different durations of temporal containment. If a sentence said, "...a bombing last Wednesday during the busy lunch hour..." might have taken place at "lunch hour", "last Wednesday", even "last week", "2001", etc.
- *Compositional* granularity disagreements occur when events are referred to by their parts at different levels of composition. Events are infinitely decomposable, and while this won't be reflected explicitly in a textual description, the compositionality does manifest as an abundance of ways of referring to what happened. If a sentence said, "...a bombing took place last week, the explosion rocked the central marketplace..." we might say the event "explosion" is part of the event "bombing" and that the "explosion" event is not the one of interest. There are many types of compositional disagreement (see section 2.3 below), here we refer only to disagreements in labeling the events in a way that affects *counting*, e.g. are there two events in the sentence or one? This category includes aggregate event mentions, such as "5 bombings in Beirut", for which annotators may disagree on whether the "5 bombings" is one event with 5 parts, or 5 events.

- *Classificational* granularity disagreements occur when events are classified at different places in a given taxonomy, such that one class subsumes the other. If the annotators were provided with a taxonomy of events that specified bombing $\ll$ attack $\ll$ event, they may disagree on whether a particular event is a "bombing" or "attack".
- *Participant* granularity disagreements occur when event participants are part of some group that can be identified at different levels. If a sentence said, "... a shooting by Israeli soldiers ..." we might say the participants are "soldiers", "Israeli soldiers", "Israeli Army", or "Israel".

Thus, the identification of an event by human annotators can disagree in any of these granular dimensions with respect to the words used in the annotated text, while still representing a general agreement about the event itself. It is a peculiarity of NLP annotation tasks that this would be considered disagreement at all.

Often we observe disagreement in granularity when different levels of detail are needed to distinguish different events that share some property at some level. For example, if there were two bombings in Beirut on September 5th, some annotators would consider it more important to fix the time of day for each bombing or the participants mentioned by their role and name.

In previous attempts to define event annotation tasks, researchers have typically "perfumed" annotator disagreement on granularity by forcing one choice in particular contexts. Examples include fixing the granularity for all events to a day, if a day is unavailable, the week, then month, then year, then decade. This is regardless of whether that choice is believed by the annotator to be the most relevant level of detail, or even correct. These choices may reduce disagreement according to some measure, but we argue that they do not fix the problem, they simply cover it up: they are brittle in that they cannot be reused for applications requiring a different granularity, they make the task harder to learn (for machines) as they force an interpretation that people may not consistently have, and they occasionally force annotators to make the wrong choices in certain situations, even when they know its wrong.

2.3 Interpretation Disagreements

Disagreements on interpretation reflect genuine disagreement about what the event refers to. As with granularity, the disagreement can come from an event's relation to other entities, and we break interpretation disagreements into the same five dimensions. Interpretation disagreements also include errors and misunderstandings by the annotators.

- *Spatial* interpretation disagreements occur when the location is vague, controversial, has some context that may change the coordinates, or perspectives that change some element of the spatial containment across annotators. For example, the location of a bombing could be "the front lines", which may be shifting and difficult to pin down latitude and longitude, or "Prussia" which is still the name of a region but once also the name of a much larger country. A location, such as Taiwan, may be considered by one annotator to be part of the People's Republic of China, and by another not to be.

- *Temporal* interpretation disagreements, similar to spatial, may occur when the time is vague or has some context that changes the actual time points. For example, the time of a bombing may be reported in a country whose time zone makes the time or even the day of the event different, or expressions like ”the past couple days” in which one annotator may take it to be a duration of two days, and another may take as a different duration. Relative dates like ”the end of world war II” or ”the October Revolution” (which took place in September) can also cause genuine disagreement among annotators if required to normalize the date to a specific year, month and day.
- *Compositional* interpretation disagreements occur when events are referred to by their parts and the annotators disagree on what the parts are. This includes the direction of the composition, e.g. ”bombing” is part of the ”explosion”, or ”explosion” is part of the ”bombing” in the previous example. This also includes the placement by annotators of implied events that contain, or are contained by, the mentioned ones.
- *Classificational* interpretation disagreements occur when events are classified under different classes, and one class does not imply the other (as opposed to granularity). This includes cases where the two classes are logically disjoint, and cases where they are not disjoint but in different branches of the taxonomy.
- *Participant* interpretation disagreements occur when the participants are vague (e.g. ”Western Authorities”), or controversial (e.g. ”Pakistan denied responsibility for the bombing”), or has some context that causes an annotator to differ from others. For example, in ”Saddam Hussein’s top advisor called the bombing an outrage” an annotator might assume that the advisor would not have spoken unless it was what he was told to say, and attribute ”Saddam Hussein” as the participant in the ”called” event, whereas a stricter reading would have the advisor as the participant.

The most common form of interpretation disagreements are ones that stem from misreadings of the text. It is important to note that most of the time, human readers are very tolerant of these kinds of errors in forming their understanding of what happened. It is more reasonable to try and ”correct” these errors to reduce disagreement, but we claim that if annotation is to scale, we need to be tolerant of them.

Interpretation disagreements are more difficult to account for than the granularity disagreements. Thus, we start with the first version of this crowdsourced annotation experiment by focussing on granularity disagreements only.

3 Annotation Task

NLP systems typically use the ground truth of an annotated corpus in order to learn and evaluate their output. Traditionally, the ground truth is determined by humans annotating a sample of the text corpus with the target events and entities, with the aim to optimize the inter-annotator agreement by restricting the definition of events and providing annotators with very precise guidelines. In this paper, we propose an alternative approach for the event annotation, which introduces a novel setting and different perspective on the overall goal.

Table 1. Annotation Matrix for Putative Event$_i$

Event$_i$	Temporal						Spatial						Participants						Compositional						Classificational					
	1	2	3	4	5	ø	1	2	3	4	5	ø	1	2	3	4	5	ø	1	2	3	4	5	ø	1	2	3	4	5	ø
ann$_1$	1	0	1	1	0	1	0	1	1	0	1	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	1	1	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
ann$_N$	1	1	1	1	0	1	0	1	1	0	1	0	1	1	0	0	1	1	1	0	0	0	0	0	1	0	1	1	1	0

By analogy to image and video tagging crowdsourcing games, e.g. Your Paintings Tagger [4] and Yahoo! Video Tag Game [3], we envision that a crowdsourcing setting could be a good candidate to the problem of insufficient annotation data. However, we do not exploit the typical crowdsourcing agreement between two or more independent taggers, but on the contrary, we harness their disagreement. Our goal is to allow for a maximum disagreement between the annotators in order to capture a maximum diversity in the event expressions.

Annotation Matrix: In section 2 we introduced a classification framework to understand the disagreement between annotators. In our annotation task we only consider the granularity-based disagreement – with five axes and five levels of granularity for each axis. Following this, for each putative event (a marked verb or nominalized verb), we build an *Annotation Matrix* (Table 1) from the input of all annotators. We can then subsequently use these annotation matrices for an analysis over the whole collection of events, e.g. for determining similarity between different events and thus recognizing missed coreferences. We can also use the matrices for an analysis of the annotation space of each individual event. For example, the highest agreement in each axis level could indicate the most likely granularity for this event, while still giving a sense of the range of acceptable granularities in each dimension. Such in-depth analysis of the annotations can allow us to identify a new set of features that can help to improve the event extraction. For example, we could thus expect to find dependencies between the type of events and the level of granularity for its spatial or temporal entities.

Annotation Setting: For the proposed annotation task we plan to use a sample of the $10,000$ documents taken from the Gigaword corpus (used in the context of the DARPA's Machine Reading program (MRP)[5]) together with several sources for background knowledge. The background knowledge includes, for example, the IC++ Domain Ontology for Violent Events (identifying event types and binary relations), geographical and temporal resources as well as general lexical resources such as WordNet and DBpedia.

A pre-annotation is performed by automatically marking all the verbs and nominalized verbs as putative events (Fig. 2): this would include both events from the IC++ ontology, as well as reporting and other communication events. The IBM Human Annotation Tool (HAT) was used as an initial annotation interface. Our background knowl-

[4] http://tagger.thepcf.org.uk/

[5] http://www.darpa.mil/Our_Work/I2O/Programs/Machine_Reading.aspx

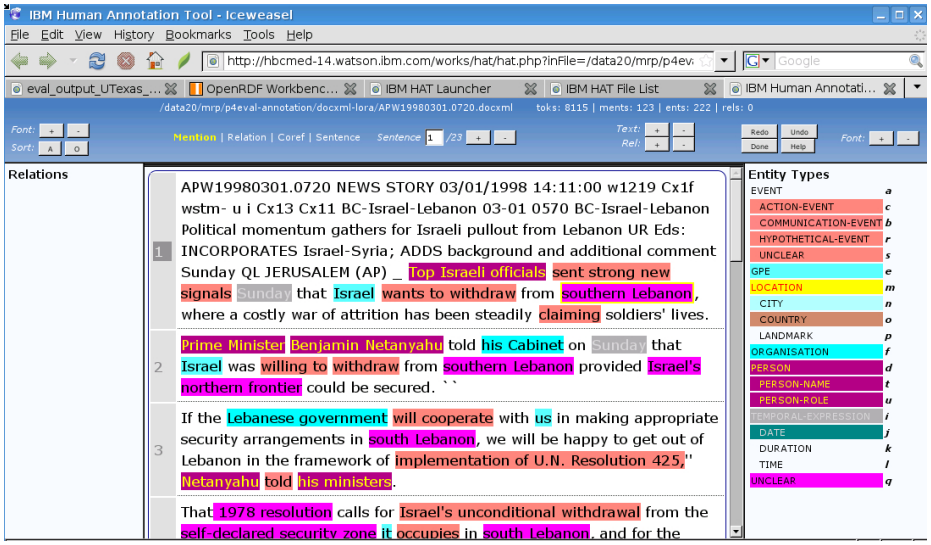

Fig. 1. Annotation interface

edge base allows us to pre-label temporal, spatial, and participant entities with granularities (e.g. city, region, country), and we provide an *a-priori* mapping from these to the numbers in the annotation matrix. The annotators do not need to know the granularity level, they are presented with all the possible choices and they select one (or more), and their choices are automatically mapped into the matrix. For example, for the sentence, "A bomb exploded in Beiruit, Lebanon last Friday," the annotator would be presented with "exploded" as the putative event, and could select between Beirut and Lebanon (or both) as the location. Since our background knowledge includes that Beirut is a city and Lebanon a country, if selected as a location for the event these are mapped to granularity levels 2 and 3, resp.

We ran explorative annotation experiments with the IBM Human Annotation Tool (Fig. 1), and proceeded further with using larger annotator pool at Amazon Mechanical Turk and CrowdFlower. Annotation data was collected according to the stages sketched in Fig. 2. As presented in the figure, the process comprises of four Phases (I-IV). Each Phase is split in two main steps: (A) collecting initial set of annotations (in each Phase different types of annotations) and (B) performing spam filtering step. In each phase we select from the A results items that can be used as Gold Standard items in step B.

4 Related Work

This work derives directly from our efforts in the Machine Reading Program (MRP) to define an annotation task for event coreference. The process of developing guidelines is very iterative - starting with an initial set of requirements from simple examples, the guidelines are then applied by a small group and the disagreements, in particular, are studied and the guidelines modified to address them. The process is repeated until the

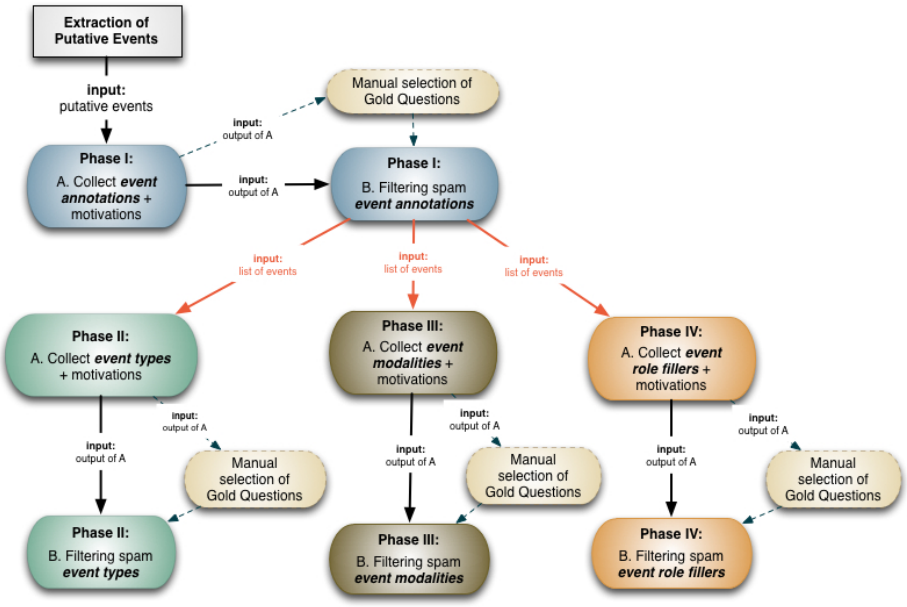

Fig. 2. Crowdsourcing Annotation Process

agreement (typically a κ score) reaches an acceptable threshold, and then is distributed to the actual annotators. Developing the guidelines usually takes several months and requires language experts.

The idea of analyzing and classifying annotator disagreement on a task is therefore not new, but part of the standard practice in developing guidelines, which are widely viewed as necessary for human annotation tasks. However, the goals of classifying disagreement, in most previous efforts, has been designed to eliminate it, not to exploit it. This can be seen in most annotation guidelines for NLP tasks. For example, in [4], the instructions include that all modality annotations should, "ignore temporal components of meaning. For example, a belief stated in the future tense (Mary will meet the president tomorrow) should be annotated with the modality 'firmly believes' not 'intends' or 'is trying'." [4]. Here the guideline authors repeat that these choices should be made, "even though other interpretations can be argued."

Similarly, in the annotator guidelines for the MRP Event Extraction Experiment (aiming to determine a baseline measure for how well machine reading systems extract attacking, injuring, killing, and bombing events) [5] show examples of restricting humans to follow one interpretation, for example for location, in order to ensure higher chance for the inter-annotator agreement. In this case, the spatial information is restricted only to "country", even though other more specific location indicators might be present in the text, e.g. the Pentagon.

There are many annotation guidelines available on the web and they all have examples of "perfuming" the annotation process by forcing constraints to reduce disagree-

ment (with a few exceptions). In [6] and subsequent work in emotion [7], 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 acheives very high κ scores (above .9), but it is not clear if the forced consensus really achieves anything meaningful. It is also not clear if this is practical in a crowdsourcing environment.

A good survey and set of experiments using disagreement based semi-supervised learning can be found in [8]. However, they use disagreement to describe a set of techniques based on bootsrapping, not collecting and exploiting the disagreement between human annotators. The bootstrapping idea is that small amounts of labelled data can be exploited with unlabelled data in an iterative process [9], with some user-relevance feedback (aka active learning).

Disagreement harnessing and crowdsourcing has previously been used by [10] for the purpose of word sense disambiguation, and we will explore similar strategies in our experiments for event modeling. As in our approach, they form a confusion matrix from the disagreement between annotators, and then use this to form a similarity cluster. In addition to applying this technique to events, our work adds a novel classification scheme for annotator disagreement that provides a more meaningful feature space for the confusion matrix; it remains to be demonstrated whether this will have impact.

The key idea behind our work is that harnessing disagreement brings in multiple perspectives on data, beyond what experts may believe is salient or correct. This concept has been demonstrated previously in the Waisda? video tagging game [11], in which lay (non-expert) users provided tags for videos in a crowdsourcing game. The Wasida? study showed that only 14% of tags provided by lay users could be found in the professional video annotating vocabulary (GTAA), which indicates a huge gap between the professional and lay users' views on what is important in a video. The study showed the lay user tags were meaningful (as opposed to useless or erroneous), and the mere quantity of tags was a success factor in retrieval systems for these multimedia objects. Similarly, the steve.museum project [12] studied the link between a crowdsourced user tags folksonomy and the professionally created museum documentation. The results showed that users tag artworks from a different perspective than that of museum professionals: again in this seperate study only 14% of lay user tags were found in the expert-curated collection documentation.

5 Conclusions

When considering approaches for detecting and extracting events in natural language text and representing those extracted events for use in the Semantic Web, we see the implications of what differentiates events from objects. When it comes to annotation tasks, the compositional nature of events plays an important role in the way in which annotators perceive the events, annotate them and agree in their existence.

For the goal of improving event detection, we have chosen to leverage the annotator disagreement in order to obtain an event description that allows machine readers to better identify and detect events. In this way, we do not aim for annotator agreement (as in many tagging scenarios where similarity is an indicator for success), but on the contrary we hypothesized that annotator disagreement for even annotation actually could

provides us with a better event description from the perspective of automatic event detection. By factoring in the different viewpoints that annotators can have, the likelihood of identifying events that have been represented with such viewpoints is higher.

In this paper we have contributed a classification framework of the variety of ways in which people can perceive events, with a matrix for the identification of patterns of agreement and disagreement (with the aim to be able later to exploit them in the MR of events), and with a description of the design of the experiment to verify the effect of using the matrix in the annotation task.

6 Acknowledgments

The authors gratefully acknowledge the support of Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0172. Any opinions, findings, conclusion or recommendations ex-pressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the US government. We would like to thank Sid Patwardhan from IBM Research for his contribution to the implementation.

References

1. Higginbotham, J., Pianesi, F., Varzi, A.: Speaking of Events. Oxford University Press, USA (2000)
2. Lewis, D.K.: On the Plurality of Worlds. Blackwell Publishers (1986)
3. van Zwol, R., Garcia, L., Ramirez, G., Sigurbjornsson, B., Labad, M.: Video tag game. In: 17th International World Wide Web Conference (WWW developer track), ACM (April 2008)
4. Baker, K., Bloodgood, M., Diab, M., Dorr, B., Hovy, E., Levin, L., McShane, M., Mitamura, T., Nirenburg, S., Piatko, C., Rambow, O., Richardson, G.: Simt scale 2009 modality annotation guidelines. Technical Report 4, Human Language Technology Center of Excellence (2010)
5. Hovy, E., Mitamura, T., Verdejo, F.: Event coreference annotation manual. Technical report, Information Sciences Institute (ISI) (2012)
6. Ang, J., Dhillon, R., Krupski, A., Shriberg, E., Stolcke, A.: Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In: in Proc. ICSLP 2002. (2002) 2037–2040
7. Litman, D.J.: Annotating student emotional states in spoken tutoring dialogues. In: In Proc. 5th SIGdial Workshop on Discourse and Dialogue. (2004) 144–153
8. Zhou, Z.H., Li, M.: Semi-supervised learning by disagreement. Knowl. Inf. Syst. **24**(3) (2010) 415–439
9. Riloff, E., Jones, R.: Learning dictionaries for information extraction by multi-level bootstrapping. In: AAAI/IAAI. (1999) 474–479
10. Chklovski, T., Mihalcea, R.: Exploiting agreement and disagreement of human annotators for word sense disambiguation. In: UNT Scholarly Works. UNT Digital Library (2003)
11. Gligorov, R., Hildebrand, M., van Ossenbruggen, J., Schreiber, G., Aroyo, L.: On the role of user-generated metadata in audio visual collections. In: K-CAP. (2011) 145–152
12. Leason, T.: Steve: The art museum social tagging project: A report on the tag contributor experience. In: Museums and the Web 2009:Proceedings. (2009)

Using Syntactic Dependencies and WordNet Classes for Noun Event Recognition

Yoonjae Jeong and Sung-Hyon Myaeng

Korea Advanced Institute of Science and Technology
291 Daehak-ro (373-1 Guseong-dong), Yuseong-gu, Daejeon 305-701,
Republic of Korea

{hybris, myaeng}@kaist.ac.kr

Abstract. The goal of this research is to devise a method for recognizing TimeML noun events in a more effective way. TimeML is the most recent annotation scheme for processing the event and temporal expressions in natural language processing fields. In this paper, we argue and demonstrate that the dependencies and the deep-level WordNet classes are useful for recognizing events. We formulate the event recognition problem as a classification task using various features including lexical semantic and dependency-based features. The experimental results show that our proposed method outperforms significantly a state-of-the-art approach. Our analysis of the results demonstrates that the dependencies of direct object and the deep-level WordNet hypernyms play pivotal roles for recognizing noun events.

Keywords: Event Recognition, TimeML, TimeBank, WordNet, Natural Language Processing, Machine Learning

1 Introduction

Automatic event extraction from text is one of the important parts in text mining field. There are two types of definitions for events. In the area of topic detection and tracking (TDT), an event is defined as an instance of a document level topic describing something that has happened (Allan 2002). On the other hand, the information extraction (IE) field uses a more fine-grained definition of an event, which is often expressed by a word or phrase in a document. In TimeML, a recent annotation scheme, events are defined as situations that happen or occur and expressed by verbs, nominalizations, adjectives, predicative clauses or prepositional phrases (Pustejovsky, Castaño, et al. 2003). In this paper, we follow the view of IE, and focus on recognition of TimeML events.

Previous studies have proposed different approaches for automatic recognition of events, most notably adopting machine learning techniques based on lexical semantic classes and morpho-syntactic information around events (Bethard and Martin 2006; Boguraev and Ando 2007; Llorens, Saquete, and Navarro-Colorado 2010; March and Baldwin 2008; Saurí et al. 2005). In recognizing events, some of the past work used top level WordNet classes (Fellbaum 1998) to represent the meanings of events. It

turns out, however, that such WordNet classes used as lexical semantic features are not sufficient. When WordNet hypernyms within the top four levels (Llorens, Saquete, and Navarro-Colorado 2010) or some selected classes (Bethard and Martin 2006) were used, they could not represent events well. For example, the WordNet class *event* is a representative level-4 class expressing events, but just 28.46% of *event* nouns, i.e., hyponyms of WordNet *event* class occurring in the TimeBank 1.2 corpus are annotated as TimeML events. TimeBank is a corpus containing news articles annotated based on the TimeML scheme (Pustejovsky, Hanks, et al. 2003).

Events can be recognized in different part-of-speech. In this paper, we focus on noun event recognition because the previous approaches showed low performances for recognizing noun events although nouns cover about 28% of all the events, according to our data analysis. For the problem of recognizing event nouns, we propose a method of using dependency-based features that exist between an event noun and its syntactically related words. In addition, we chose to use deeper level WordNet classes than those at the top-4 levels as in the previous work. We show that our proposed method outperforms the previous work by running experiments.

The rest of the paper is organized as follows. Section 2 introduces TimeML and TimeBank corpus as a representation and annotation scheme and as a test bed, respectively. It is followed by a discussion of related work for TimeML-based event recognition in Section 3. Section 4 presents our event recognition method using the deep-level WordNet classes and the dependency-based features. We then discuss our experiments and results in Section 5. Finally, the last section presents our conclusions.

2 TimeML and TimeBank Corpus

TimeML is a robust specification language for event and temporal expressions in natural language (Pustejovsky, Castaño, et al. 2003). It was first announced in 2002 in an extended workshop called TERQAS (Time and Event Recognition for Question Answering System)[1]. It addresses four basic problems:

1. Time stamping of events (identifying an event and anchoring it in time)
2. Ordering events with respect to one another (lexical versus discourse properties of ordering)
3. Reasoning with contextually underspecified temporal expressions (temporal functions such as "*last week*" and "*two weeks before*")
4. Reasoning about the persistence of events (how long does an event or the outcome of an event last)

Fig. 1. Four problems in event and temporal expression markup (Hobbs and Pustejovsky 2003)

There are four major data components in TimeML: EVENT, TIMEX3, SIGNAL, and LINK (Pustejovsky et al. 2007). TimeML considers event as a term for situations

[1] http://www.timeml.org/site/terqas/index.html

that happen or occur or elements describing states or circumstances in which something obtains or holds the truth (EVENT). Temporal expressions in TimeML are marked up with the TIMEX3 tags referring to dates, durations, sets of times, etc. The tag SIGNAL is used to annotate function words, which indicates how temporal objects (event and temporal expressions) are to be related to each other. The last component, LINK, describes the temporal (TLINK), subordinate (SLINK), and aspectual relationship (ALINK) between temporal objects.

Fig. 2 shows an example of TimeML annotation. For an event "*teaches*", its type is kept in class attribute, and its tense and aspect information is tagged in MAKEINSTANCE. The normalized value of temporal expressions "*3:00*" and "*November 22, 2004*" are stored in value attribute in TIMEX3 tag. The signal words "*at*" and "*on*" make links between events and temporal expressions through TLINK tags.

```
John
<EVENT eid="e1" class="OCCURRENCE"> teaches </EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PRESENT"
    aspect="NONE" />
<SIGNAL sid="s1"> at </SIGNAL>
<TIMEX3 tid="t1" type="TIME" value="2004-11-22T15:00"
    temporalFunction="TRUE" anchorTimeID="t2"> 3:00
    </TIMEX3>
<SIGNAL sid="s2"> on </SIGNAL>
<TIMEX3 tid="t2" type="DATE value="2004-11-22">
    November 22, 2004 </TIMEX3>.

<TLINK eventInstanceID="ei1" relatedToTime="t1"
    relType="IS_INCLUDED" signalID="s1"/>
<TLINK timeID="t1" relatedToTime="t2"
    reltype="IS_INCLUDED" signalID="s2"/>
```

Fig. 2. An example of TimeML annotation (Pustejovsky et al. 2007)

Among several corpora[2] annotated with TimeML, TimeBank is most well-known as it started as a proof of concept of the TimeML specifications. TimeBank 1.2 is the most recent version of TimeBank, annotated with the TimeML 1.2.1 specification. It contains 183 news articles and more than 61,000 non-punctuation tokens, among which 7,935 are events.

We analyzed the corpus to investigate on the distribution of PoS (Part of Speech)[3] for the tokens annotated as events. As shown in Table 1, most events are expressed in verbs and nouns. Sum of the two PoS types covers about 93% of all the event tokens, which is split into about 65% and 28% for verb and nouns, respectively. The percentages for cardinal numbers and adjectives are relatively small. They usually express quantitative (e.g., "*47 %*") and qualitative (e.g., "*beautiful*") states. Adverbs and

[2] TimeML Corpora, http://timeml.org/site/timebank/timebank.html

[3] By Stanford PoS tagger, http://nlp.stanford.edu/software/tagger.shtml

prepositions indicate events when they appear in predicative phrases (e.g., "*he was here*" or "*he was on board*").

Table 1. PoS distribution of event tokens

PoS tag	# Event	Coverage
VB (Verb)	5,171	65.17 %
NN (Noun)	2,183	27.51 %
CD (Cardinal Number)	279	3.52 %
JJ (Adjective)	223	2.81 %
RB (Adverb)	29	0.37 %
IN (Preposition)	46	0.58 %
Misc.	4	0.05 %
SUM	7,935	100.00 %

In finding verb events automatically from the TimeBank corpus, Llorens et al. (2010)'s work, a state-of-the-art approach, showed high effectiveness in terms of F1 (0.913). We note, however, its performance in recognizing noun events was just 0.584 in F1. This clearly indicates that noun even recognition, which is significant by itself, is a harder problem that needs to draw more attention and research.

3 Related Work

EVITA (Saur íet al. 2005) is the first event recognition tool for TimeML specification. It recognizes events by combining linguistic and statistical techniques. It uses manually encoded rules based on linguistic information as main features to recognize events. It also uses WorldNet classes to those rules for nominal event recognition, and checks whether the head word of noun phrase is included in the WordNet event classes. For sense disambiguation of nouns, it utilizes a Bayesian classifier trained on the SemCor corpus[4].

Boguraev and Ando (2007) analyzed the TimeBank corpus and presented a machine-learning based approach for automatic TimeML events annotation. They set out the task as a classification problem, and used a robust risk minimization (RRM) classifier (Zhang, Damerau, and Johnson 2002) to solve it. They used lexical and morphological attributes and syntactic chunk types in bi- and tri-gram windows as features.

Bethard and Martin (Bethard and Martin 2006) developed a system, STEP, for TimeML event recognition and type classification. They adopted syntactic and semantic features, and formulated the event recognition task as classification in the word-chunking paradigm. They used a rich set of features: textual, morphological, syntactic dependency and some selected WordNet classes. They implemented a Support Vector Machine (SVM) model based on those features.

Lastly, Llorens et al. (2010) presented an evaluation on event recognition and type classification. They added semantic roles to features, and built the Conditional Ran-

[4] http://www.gabormelli.com/RKB/SemCor_Corpus

dom Field (CRF) model to recognize events. They conducted experiments about the contribution of semantic roles and CRF and reported that the CRF model improved the performance but the effects of semantic role features were not significant. The approach achieved 82.4% in F1 in event recognition for the TimeBank 1.2 corpus. It is a state-of-the-art approach in TimeML event recognition and type classification.

4 Event Recognition

The main goal of our research is to devise an effective method for recognition of TimeML noun events. Our proposed method consists of three parts: preprocessing, feature extraction, and classification. The preprocessing part analyzes raw text to do tokenizing, PoS tagging, and syntactic parsing (dependency parsing). It is done by the Stanford CoreNLP package[5], which is a suite of natural language processing tools. Then, the feature extraction part converts the preprocessed data into the feature spaces. We explain the details of our feature extraction methods in Subsection 4.1. Finally, the classification part determines whether the given noun is an event or not using the MaxEnt classification algorithm.

4.1 Feature Sets

The feature sets to recognize events consist of three types: *Basic Features*, *Lexical Semantic Features, and Dependency-based Features*. The *Basic Features* are based on one of the TimeML annotation guidelines – prenominal noun is not annotated as events –, and the *Lexical Semantic Features* are the lemmas and all WordNet hypernyms of target nouns to be classified. Those hypernyms include the deep WordNet classes indicating the specific concept of nouns. The *Dependency-based Features* are adopted because syntactically related words tend to serve as important clues in determining whether or not a noun refers to an event.

Basic Features. *The Basic Features* include named entity (NE) tags and an indication of whether the target noun is prenominal or not. A personal name and a geographical location cannot be an event whereas prenominal nouns are not considered as events according to the TimeWML annotation guideline.

Lexical Semantic Features. The *Lexical Semantic Features* (*LS*) is the set of target nouns' lemmas and their all-depth WordNet semantic classes (i.e., hypernyms). Some nouns have high probabilities of indicating an event when they are included in a very specific WordNet classes. For example, a noun "*drop*" is always an event regardless of its context of a sentence. While the word sense-ambiguity problem arises in mapping a token to a synset in WordNet, we ignore the problem and simply use the WordNet hypernyms of all the senses.

[5] http://nlp.stanford.edu/software/corenlp.shtml

Dependency-based Features. We posit that nouns become events if they occur with a certain surrounding context, namely, syntactic dependencies. We use the words and their semantic classes related to the target noun through dependency relations. Four dependencies we consider are: direct object (OBJ), subject (SUBJ), modifier (MOD), and preposition (PREP).

- ***VB_OBJ type.*** A feature is formed with the governing verb, which has the OBJ relation with the target noun, and its hypernyms. In "*... delayed the game...*", for instance, the verb "*delay*" can describe the temporal state of its object noun, "*game*".
- ***VB_SUBJ type.*** It is the verb that has the SUBJ relation with the target noun and its hypernyms. For example, the verb "*occur*" indicates that the subject of the verb is an event because it actually occurs as in the definition of an event.
- ***MOD type.*** It refers to the dependent words and their hypernyms in MOD relation. This feature type is based on the intuition that some modifiers such as temporal expression reveal the noun it modifies has a temporal state and therefore is likely to be an event.
- ***PREP type.*** This is the preposition of a noun. Some prepositions such as "*before*" may indicate that the noun after them occurs at some specific time.

Sometimes, *Dependency-based Features* need to be combined with *Lexical Semantic Features* because a certain syntactic dependency may not be an absolute clue for an event by itself but only when it co-occurs with a certain lexical or semantic aspect of the target noun. As shown in Table 2, direct objects of "*report*" are not always events (about 32% are not events in the TimeBank corpus). However, then the direct object belongs to the WordNet *process* class, the target noun would be almost always an event. In this case, therefore, we need to use a combined feature.

Table 2. The *process* class as direct objects and its event ratio in TimeBank 1.2 corpus

Verb	Object (Noun)	# of Event (Ratio)
"*report*"	WordNet *process* class	14/14 (100.00%)
*	WordNet *process* class	153/325 (47.08%)
"*report*"	*	30/44 (68.18%)

[*] Indicates the any verbs or nouns

4.2 Classification

While the three different types of features make their own contributions in determining whether a noun is an event, their relative weights are all different. A strict classification algorithm categorizes the target nouns based on the weighted features.

We weight the features with Kullback-Leibler divergence (KL-divergence), which is a non-symmetric measure of the difference between two probability distributions (Kullback and Leibler 1951) and a popular weighting scheme in text mining. For a feature *f*, its weight is calculated using the formula in (1) where E and $\neg E$ are the dis-

tributions of event and non-event term. $P_E(f)$ and $P_{\neg E}(f)$ are the probabilities of f in E and $\neg E$, respectively.

$$W(f) = KL(E|\neg E) = P_E(f)\ln\frac{P_E(f)}{P_{\neg E}(f)} \quad (1)$$

Since we decided to use all the WordNet hypernyms as possible features, which cause the feature space too large to handle, we need to select more valuable ones from the candidate set. We use the weighing method using KL-divergence for this purpose and selected top 104,922 features because the cut-off value empirically showed the best performance in our preliminary experiment. We measured the performance when we applied top-k features, and it was maximized at k = 104,922.

For our classification algorithm, we considered four popular ones in machine learning: Naïve Bayes, Decision Tree (C4.5), MaxEnt, and SVM algorithms. Among them, the MaxEnt showed the best performance for our classification task. The packages we used are Weka (Witten, Frank, and Hall 2011) and Mallet machine learning tools (McCallum 2002).

5 Experiment

5.1 Comparison with Previous Work

We first evaluated the proposed method by comparing the previous work, whose result is shown in Table 3. We chose two baselines (Bethard & Martin 2006; Llorens et al. 2010) that were most recent ones using the TimeBank 1.2 corpus.

The proposed method shows an improvement of about 22% and 9% in terms of precision and recall than the state-of-the-art, respectively, the work of Llorens et al. Overall, the proposed method increased the F1 score by about 18% and 13% compared to the two baselines, respectively. The evaluation was done by 5-fold cross validation.

Our classifier used only 85,518 features within the top-8 WordNet classes among the 104,922 features mentioned in Section 4.2. In Section 5.3, we describe the cumulative level-8 features in detail.

Table 3. Comparison with the proposed method and previous works

Approach	Precision	Recall	F1
Bethard & Martin (2006)	0.729	0.432	0.543
Llorens et al. (2010)	0.727	0.483	0.584
Proposed Method	0.950	0.577	0.718

5.2 Contribution Analysis

We ran additional experiments to understand the roles of the individual feature types. In order to show relative importance of *Lexical Semantic Features* (*LS*), *De-*

pendency-based Features (*VB_OBJ*, *VB_SUBJ*, *MOD*, and *PREP types*), we measured performance changes caused by excluding one feature type at a time.

As shown in Table 4, *VB_OBJ* and *MOD* features are judged to be most important because the performance was decreased most significantly. The effects of the other features were not as great, but cannot be disregarded as they always contribute to the overall performance increase.

Table 4. Contributions of individual feature types

Feature Type	***Precision***	***Recall***	***F1***
ALL	0.950	0.577	0.718
- *LS*	0.958 (+0.8%)	0.561 (-1.6%)	0.708 (-1.0%)
- *VB_OBJ*	0.939 (-1.1%)	0.517 (-6.0%)	0.667 (-5.1%)
- *VB_SUBJ*	0.944 (-0.6%)	0.554 (-2.3%)	0.698 (-2.0%)
- *MOD*	0.941 (-0.9%)	0.524 (-5.3%)	0.673 (-4.5%)
- *PREP*	0.940 (-1.0%)	0.564 (-1.3%)	0.705 (-1.3%)

5.3 The Effect of Deep-level WordNet Classes

To investigate the effect of deep-level WordNet classes, we observed the performance changes incurred by increasing the cumulative WordNet depth within which features were generated. Depth fifteen, for example, means all the hypernyms of the matched word are considered as features. The results are presented in Fig. 3.

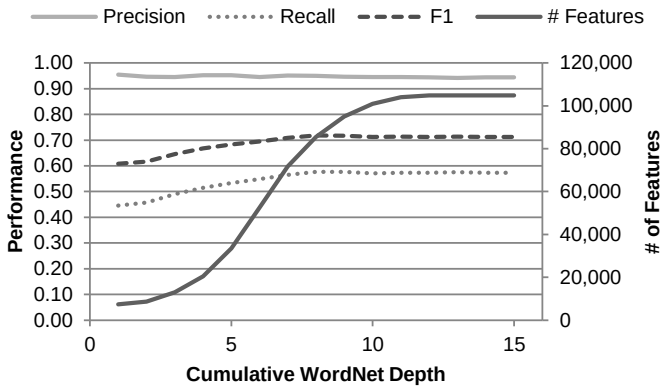

Fig. 3. Performance per cumulative WordNet depth

In this figure, the *y*-axis on the left represents the performance of event recognition in terms of precision, recall, or F1, and the *y*-axis on the right shows the numbers of features that vary when we apply the cumulative WordNet depth, which is represented by the *x*-axis.

Regardless of the depth of WordNet classes, the classifier reached the high precision over 0.9, but the recall varied quite widely. Recall increased with the rise of class depth, and it rose to the peak at top-8 level. The recall and F1-scores were 0.577 and 0.718, respectively.

The number of features increased continuously up to the level 13, but stayed the same beyond that. The number of features was 104,922, but the classifier used only 85,518 features at level 8 (where the performance was the best). From these results, we expect that there is a proper level of ontology to recognize events, which is shown to be level 8 in WordNet classes.

6 Conclusion

In this paper, we propose a TimeML noun event recognition method using syntactic dependency and WordNet classes and show their effect using the TimeBank collection. We chose to focus on noun events because they were recognized poorly in the previous research although they constitute about 28% of the events. The problem of recognizing such events was formulated as a classification task using lexical semantic (lemma and WordNet hypernyms) and dependency-based features.

Our experimental results show that the proposed method is better than the previous approach in recognizing TimeML noun events. The performance increase in terms of F1 measure is from 0.584 to 0.718, which we consider very significant. Through our analysis, we arrive at the conclusion that using dependency-based features and deep-level WordNet classes are important for recognizing events. We also showed that recall was increased significantly by using the hypernym features from lower depth of the WordNet hierarchy. A performance increase in recall for event detection, mainly due to the accurate handling of nouns and to effectiveness of the proposed classification method, would be translated into wider coverage of event-related triples in Semantic Web.

Although the proposed method showed encouraging results compared to the previous approaches, it still has some limitations. One issue is on the level of WordNet or an ontology for expanding the feature set because the current method requires too large feature space. Another one is word sense disambiguation that we ignored entirely in the current work. Although we obtained some performance increase with deeper levels, it's not clear how much more gain we will get with sense disambiguation. We are currently working on these two issues.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0027292).

Reference

1. Allan, James, ed. 2002. *Topic Detection and Tracking: Event-based Information Organization*. Springer.
2. Bethard, Steven, and James H Martin. 2006. “Identification of Event Mentions and Their Semantic Class.” In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 146–154. Association for Computational Linguistics.
3. Boguraev, Branimir, and Rie Ando. 2007. “Effective Use of TimeBank for TimeML Analysis.” In *Annotating, Extracting and Reasoning About Time and Events*, ed. Frank Schilder, Graham Katz, and James Pustejovsky, 4795:41–58. Springer Berlin / Heidelberg. doi:10.1007/978-3-540-75989-8_4.
4. Fellbaum, Christiane, ed. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.
5. Hobbs, Jerry, and James Pustejovsky. 2003. “Annotating and Reasoning About Time and Events.” In *AAAI Technical Report SS-03-05*.
6. Kullback, Solomon, and Richard A. Leibler. 1951. “On Information and Sufficiency.” *The Annals of Statistics* 22 (1): 79–86.
7. Llorens, Hector, Estela Saquete, and Borja Navarro-Colorado. 2010. “TimeML Events Recognition and Classification: Learning CRF Models with Semantic Roles.” In *Proceedings of the 23rd International Conference on Computational Linguistics*, 725–733. Association for Computational Linguistics.
8. March, Olivia, and Timothy Baldwin. 2008. “Automatic Event Reference Identification.” In *Proceedings of the Australasian Language Technology Workshop*, 6:79–87.
9. McCallum, Andrew Kachites. 2002. “MALLET: A Machine Learning for Language Toolkit.” http://mallet.cs.umass.edu/.
10. Pustejovsky, James, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. “TimeML: Robust Specification of Event and Temporal Expressions in Text.” In *Proceedings of the 5th International Workshop on Computational Semantics*.
11. Pustejovsky, James, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, et al. 2003. “The TIMEBANK Corpus.” In *Proceedings of the Corpus Linguistics 2003 Conference*, 647–656.
12. Pustejovsky, James, Robert Knippen, Jessica Littman, and Roser Saurí. 2007. “Temporal and Event Information In Natural Language Text.” In *Computing Meaning*, ed. Harry Bunt, Reinhard Muskens, Lisa Matthewson, Yael Sharvit, and Thomas Ede Zimmerman, 83:301–346. Springer Netherlands. doi:10.1007/978-1-4020-5958-2_13.
13. Saurí, Roser, Robert Knippen, Marc Verhagen, and James Pustejovsky. 2005. “Evita: a Robust Event Recognizer for QA Systems.” In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 700–707. Association for Computational Linguistics. doi:10.3115/1220575.1220663.
14. Witten, Ian H., Eibe Frank, and Mark A. Hall. 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd ed. Morgan Kaufmann.
15. Zhang, Tong, Fred Damerau, and David Johnson. 2002. “Text Chunking Based on a Generalization of Winnow.” *The Journal of Machine Learning Research* 2 (March): 615–637.

Bringing parliamentary debates to the Semantic Web

Damir Juric[1,3], Laura Hollink[2], Geert-Jan Houben[1]

[1] Delft University of Technology, [2] VU University Amsterdam
[3] FER University of Zagreb

Abstract. An analysis of parliamentary debates and media resources that cover them can provide insight into the political climate of a country. Although debates are now regularly published on official government portals, their analysis remains a cumbersome and challenging task for historians and political scientists. One of the main tasks of the PoliMedia project is to allow easy cross-media comparisons and give better insight into choices that different types of media outlets make when covering parliamentary debates. As a first step of that task, Dutch parliamentary debate data available in XML files is being translated into Semantic Web standards, which will allow users to easily query the data. In this paper we discuss design choices we made to build a semantic model that will represent events and topics from the Dutch parliamentary debates.

Keywords: XML, RDF, SEM, Semantic Web, parliamentary debates

1 Introduction

In this paper we discuss ongoing work on the representation of political events on the Semantic Web. We present the design choices of a model in which we capture parliamentary debates, including how they are covered by various media.

Analyzing media coverage across several types of media outlets is a challenging task, especially for people who need deep understanding of the data and its implications, like media historians. Previous research has focused mainly on newspaper articles, because they are generally available in digital, computer-readable format. To make cross-media comparisons between different types of media outlets, links between datasets would need to be produced. For example, to support researchers that want to know how political debates are represented in the media and how the representation of topics and people change over time. We aim to facilitate this kind of analysis by providing links between datasets of political debate events and media data.

The media-historic research questions that guide the project are: "*What choices do different media make in the coverage of people and topics while reporting on debates in the Dutch parliament since the first televised evening news in 1956 until 1995? Does the representation of topics and people change over time and how do the various media types differ?*" These questions specify a number of things that needs to be expressed in the model, like people, topics, time and media types.

To answer these questions we first created a semantic model that is expressive enough to allow us to represent all important information about events from the Dutch parliament, that are recorded in the form of debate transcripts (and later in XML files). After this step, an RDF repository is created in which we instantiate the model with instances of debate events, that allows various interesting information to be extracted from this dataset using SPARQL queries.

This paper is organized as follows: first we describe the PoliMedia project in which this work is carried out. In Section 2 we give a description of our datasets of debate events and media items. In Section 3 we discus the semantic model, and in Section 4 we describe our future work.

1.1 Background: the PoliMedia project

The PoliMedia project[1] is driven by research questions from historians with respect to media coverage across several types of media outlets. Cross-media comparisons will be conducted over a longer period of time, on different topics. The project will focus on the coverage of the debates in the Dutch parliament and give insight on the different choices that different media make while reporting on those debates. Also, when research can be performed with time and media type in mind, another question can be answered: Does the representation of topics and people change over time and how do the various media types differ?

The project will be carried out in three phases: (1) a modeling phase: creating a semantic model (that is the phase described in this paper), (2) a data production phase: creating links between debates and associated media sources, and (3) an application phase: searching and navigating linked datasets.

1.2 Related work

Related work for this project comes from three domains: other projects using parliamentary debate data, event modeling and relatedness discovery.

In [1], the author describes the structure of parliamentary proceedings and sketches a widely applicable DTD. He also describes how proceedings in PDF format can be transformed into deeply nested XML files. The work described is done as part of a project called War In Parliament [2]. In the work described in this paper, we use structured XML files from War In Parliament as a basis for our model. This dataset can be searched on the Political Mashup portal [3]. [4] presents an approach that extends existing metadata enrichment processes with a method to discover historical events. The events are structured in a historical event thesaurus to enrich object metadata. As such, the event thesaurus is used as a bridge between objects in different collections. The results of the approach allows for topic-based and event-centered browsing, searching and navigating in integrated collections. In [5], the authors put events as the central elements in the representation of data from domains such as history, cultural heritage, multimedia and geography. The Simple Event Model (SEM)

[1] PoliMedia project: http://www.polimedia.nl/

is created to model events in these various domains, without making assumptions about the domain-specific vocabularies used. The researchers designed SEM with a minimum of semantic commitment to guarantee maximal interoperability. In [6] the authors describe real life problem using SEM. Some properties of SEM are used in the semantic model described in this paper. We used SEM model as a starting point on which we build our own model that conforms to the events in the parliament. The problem of link discovery is tackled in [7]: there a validation approach is presented of detected alignment links between dialog transcript and discussed documents, in the context of a multimodal document alignment framework of multimedia events (meetings and lectures). The validation approach consists of an entailment process of the detected alignment links. This entailment process exploits several features, from the structural level of aligned documents to the linguistic level of their tokens. In [8] the authors present a function that discovers relatedness between news articles across four aspects: relevance, novelty, connection clarity, and transition smoothness.

2 Description of datasets

The PoliMedia project is aimed at cross-linking four different datasets, each from different media outlets. All datasets, which are textual and audiovisual, are available via the CLARIN infrastructure.

Primary dataset for this project is a collection of Dutch parliamentary debates, the so-called *Handelingen der Staten-Generaal* or the *Dutch Hansard*. Parliamentary debates used in this project, are actual transcripts of speeches that politicians had in the parliament. At the time of writing this article, three sources of Dutch parliamentary debates are available online. On the Officiële Bekendmakingen portal, which is an official source for parliamentary debates from the Dutch parliament, debates can be found in an XML format, using XML schema and permanent identifiers. Existing identifiers point only to the debate as a whole, not specifically to parts of the debate. Also, only debates from 1995 till present are available at this source.

A second source for Dutch parliamentary debates can be found online, on the Staten-Generaal Digitaal portal[2], which contains debates from the parliament from before the year 1995. Data can be accessed publically using the SRU (Search and Retrieval via URL) [9] or OAI (Open Archives Initiative) [10] protocols. Contrary to the previous source, debates from this source posses no further structure (data is provided in txt or pdf formats).

A third source for political debates from the Dutch parliament can be found on Political Mashup [3]. This data is created by the CLARIN project War in Parliament (WIP). The project is still ongoing, and the way debates are published is continuously improving. At the time of writing this article, all debates until the year 1995 are published as XML documents (OCR with satisfactory quality is being used). This data shows a fine-grained structure.

[2] Staten-Generaal Digitaal: http://www.statengeneraaldigitaal.nl/

Secondary datasets contains different media types: newspaper articles, radio bulletins, and newscasts. The newspaper and radio bulletins dataset is available from the National Library of the Netherlands, which allows users to analyze the text of the articles and see in which way they are layered. Metadata of the articles and bulletins are available from the metadata store of the Koninklijke Bibliotheek (KB), the KB-MDO or Koninklijke Bibliotheek metadataopslag [11] as DIDL (Digital Item Declaration Language – an XML dialect [12]). The newscast dataset contains evening news and current affairs programs. Audiovisual content include program level metadata in Dublin Core and CDMI format, enriched with thesaurus terms from the Gemeenschappelijke Thesaurus Audiovisuele Archieven (GTAA). Data can be accessed using the OAI-PMH protocol.

3 Semantic model

The semantic model for the PoliMedia project is built to satisfy the requirements of the project, i.e. the research questions from the users. The model is based on the Simple Event Model [5] developed in the NWO CATCH project Agora. SEM is a model to represent events on the Web, and to explicate complicated semantic relations between people, places, actions and objects: not only who did what, when and where, but also the roles each actor played, the time during which this role is valid and the authority according to whom this role is assigned. Because the PoliMedia project deals with a specific domain, our semantic model is adapted to it so it can express important information associated with the events and actors in political debates.

3.1 Requirements for the semantic model

The goal of the project is to publish the links on the Web, so using open Web formats and standards, a Web query language, and unique identifiers (URI's) is compulsory.

The semantic model of the PoliMedia project is to be expressive in a way that it allows important information regarding parliamentary debates to be easily accessed. Important information for every parliamentary debate is:

- The **time** on which the debate is held
- **What** is being said in the debate (**topics**)
- **Who** is giving the speeches in the debate and in which **role** (**persons**)
- Links to **additional information** about actors involved in the event (names of the politicians, their party, age, etc.)
- **Subparts of the debate** have their own identifiers (part of the debate where only one speaker can be identified as actor)
- Important information about subparts is their **chronological order** (the order in which the subparts where occurring inside the parliament debate,
- **Named entities** apart from politicians (persons, locations, etc.)

Important information for parliamentary debates that are specific to PoliMedia project:

- **Links** between subparts of the debate and news articles, radio bulletins and television newscasts
- Various information about media items linked to the debate

Data from the parliamentary debates is available online, so unique identifiers are created for:

- Debates (as a document as a whole) and for the parts of the debates
- Individual news articles, radio bulletins, and television newscasts
- All political parties of the speakers in the debates as well as the speakers them self

All important information about debates listed here are represented in the semantic model.

3.2 URIs as identifiers

On the Semantic Web, all entities are identified by a URI. In our case, all source datasets already contain URIs. Our preference is to use these existing URIs directly instead of creating our own URIs. For example, we link to the newspapers of the *Koninklijke Bibliotheek* with statements like:

<http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19590000048.1.9>
<http://purl.org/linkedpolitics/nl/polivoc:coveredIn>
"http://resolver.kb.nl/resolve?urn=ddd:**010688440**:mpeg21:a0001:ocr" ;

We have made a different choice for the debate events, as these are the core of our dataset. Also for debates, URIs do already exist: the government website *officielebekendmakingen.nl* provides persistent URIs to debates after 1995, and the project War In Parliament provides URIs for debates as well as parts of debates. Nevertheless, we create our own URIs for each debate and parts of debates, for example:

<http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19590000048.1.9>
a <polivoc:Speech> ;

The reason for this is that we want the URIs to be dereferenceable, i.e. we want to serve informative and descriptive RDF when the event URI is requested. Neither *officielebekendmakingen.nl* nor *War in Parliament* does this. We use so-called PURLs (**P**ersistent **U**niform **R**esource **L**ocators), Web addresses that act as permanent identifiers.

3.3 Provenance

We build on existing data and tools. It is important to preserve this provenance information, both to give credit where credit is due and to provide information about how much the data can be trusted. For every debate in our model we add information about the original source of the debate. For example, the next statement uses the dc:source property to state that the original debate came from Political Mashup:

<http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19590000048>
<http://purl.org/dc/elements/1.1/source>
"http://resolver.politicalmashup.nl/nl.proc.sgd.d.19590000048" ;

Named entities were extracted in the War In Parliament project using the NER tool Folia[14]. We use the *dc:provenance* property to state the source of extracted entity.

3.4 Description of the semantic model

The semantic model, as well as the links between datasets, is expressed in the RDF format, W3C Standard for Semantic Web. Also, the data is made compatible with the ISOCAT standard[3], Dublin Core[4] and SKOS[5].

We created this semantic model to conform to the rules and regulations of the Dutch parliament, although the model can be easily adapted to follow different rules (of parliaments in other countries), because in its core all parliamentary debates consists of the same most important elements like the topics and the speeches.

All debates conform to the same rule, where speakers give speeches in the parliament in some chronological order. First speaker is always the "voorzitter" (the person who is in charge of the actual debate and can be called chairman). The chairman gives usually an introduction to the topic and after his speech he gives the floor to some member of the parliament.

Every debate has three main structural elements:

- The topics – the themes or agenda of the meeting
- The speeches – every word by every speaker is transcribed including the names of the speakers and their affiliation
- Actions – descriptions, lists, etc.

Every transcript contains metadata with important information about the debate as a whole, like the date when the debate actually happened in the parliament, the title of the debate etc. In the PoliMedia semantic model, as can be seen in Fig1., a debate is represented as a resource with its own unique identifier (for example: http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002). This resource serves as a domain for Dublin Core properties like *dc:date*, *dc:title*, *dc:identifier*, *dc:publisher*, *dc:source* and *dc:language*, which points to the literals that contain information about the date when the debate happened, its official title, unique identifier and original source, the publisher and the language on which the debate was published (an RDF example is given in Fig. 2).

The PoliMedia specific property *hasPart* is attached to the resource containing the debate URI and points to the range of possible parts of the debate that the debate as a whole can contain (this element is shown in Fig.3). One specific part of a debate always contains elements called *DebateContext* and *Speech*. Element *DebateContext* contains text that is read by the chairman (*voorzitter*) of the debate and that text represents the short description of subjects that will be addressed in the forthcoming speech.

[3] http://www.isocat.org/
[4] http://dublincore.org/
[5] http://www.w3.org/2004/02/skos/

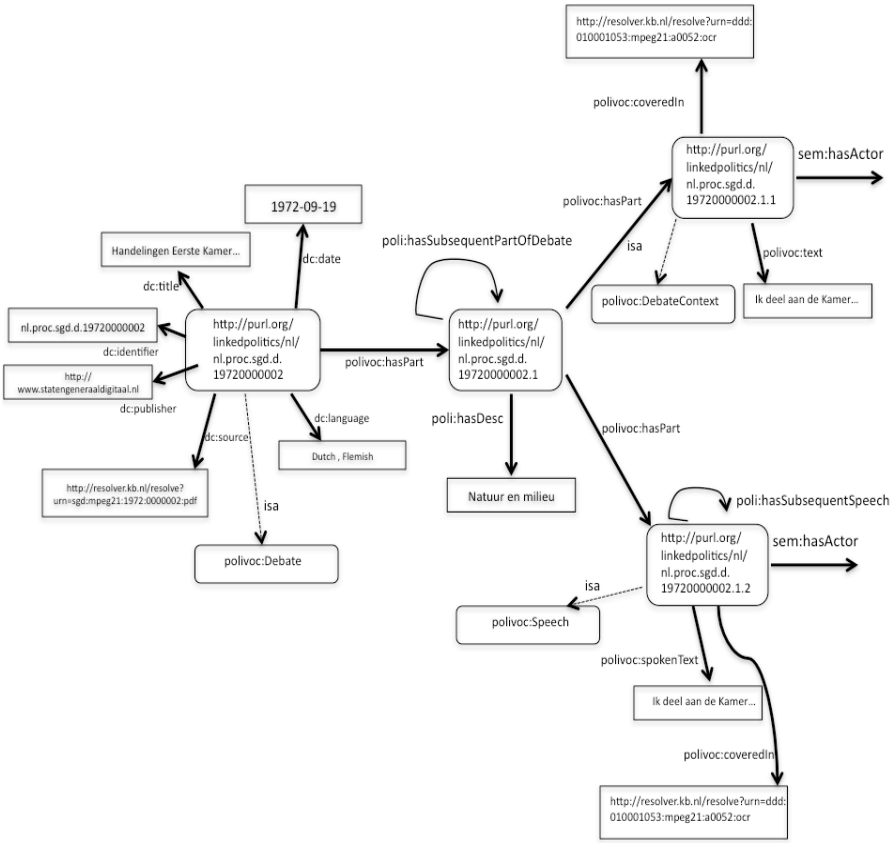

Fig. 1. Part of semantic model representation of the debates dataset (second part on Fig. 4.)

```
<http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002>
      a       <polivoc:Debate> ;
      <http://purl.org/dc/elements/1.1/date>
              "1972-09-19" ;
      <http://purl.org/dc/elements/1.1/identifier>
              "nl.proc.sgd.d.19720000002" ;
      <http://purl.org/dc/elements/1.1/language>
              "Dutch; Flemish" ;
      <http://purl.org/dc/elements/1.1/publisher>
              "http://www.statengeneraaldigitaal.nl" ;
      <http://purl.org/dc/elements/1.1/source>
              "http://resolver.politicalmashup.nl/nl.proc.sgd.d.19720000002" ;
      <http://purl.org/dc/elements/1.1/title>
              "Handelingen Eerste Kamer 1972 19 september 1972, Paginas 3-10." ;
      <http://purl.org/linkedpolitics/nl/polivoc:hasPart>
              <http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002.2> ,
              <http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002.3> ,
              <http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002.4> ,
              <http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002.1> .
```

Fig. 2. Debate represented in RD

```
<http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002.2>
      a       <polivoc:PartOfDebate> ;
      <http://purl.org/dc/elements/1.1/source>
              "http://resolver.politicalmashup.nl/nl.proc.sgd.d.19720000002.2" ;
      <http://purl.org/linkedpolitics/nl/poli:hasDescription>
              "behandeling van het wetsontwerp Gemeentelijke herindeling van het Land van Heusden en Altena ( 11 284 )." ;
      <http://purl.org/linkedpolitics/nl/polivoc:hasPart>
              "http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002.2.4" ,
              "http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002.2.3" ,
              "http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002.2.5" ,
              "http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002.2.6" ,
              "http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002.2.2" ,
              "http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002.2.1" ;
      <http://purl.org/linkedpolitics/nl/polivoc:hasSubsequentPartOfDebate>
              "http://purl.org/linkedpolitics/nl/nl.proc.sgd.d.19720000002.3" .
```

Fig. 3. Part of the debate represented in RDF

The most important element of the PoliMedia semantic model is the element *Speech* that represents the actual speech that a certain member of Parliament has spoken while addressing the issues of the debate topic (Fig.4 and Fig.5). The content of the speech is saved as a *Literal*. Every speech has its speaker and those two resources are connected with the *sem:hasActor* property described in the Simple Event Model[5]. Property *hasActor* points to the blank node with three other properties leaving from the node. Objects of those properties are URIs that lead to the pages of the politician giving the speech, to the party the mentioned politician is member of, and SEM properties denoting the role of the *hasActor* property.

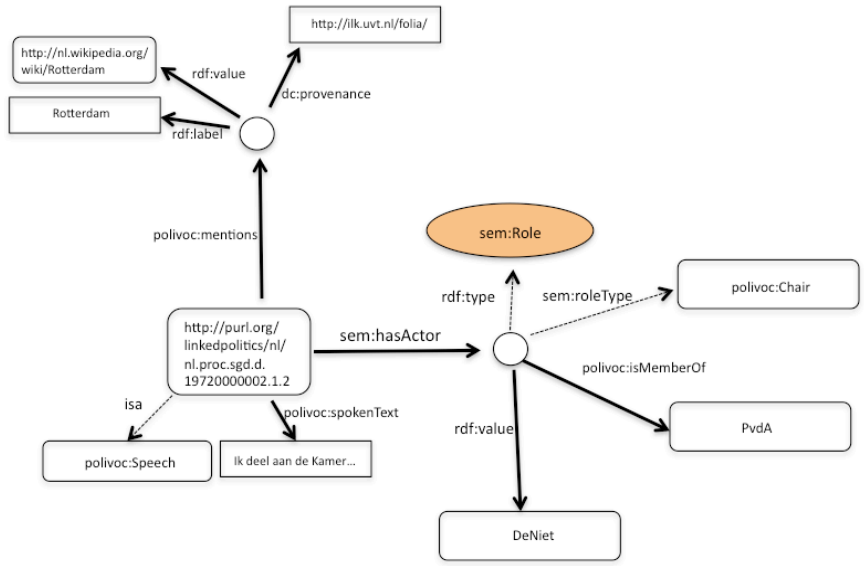

Fig. 4. Semantic model representation of the debates dataset

Fig. 5. Example of one speech in RDF

By nature, speeches in the parliament usually contain a great number of named entities, such as names of politicians or business people, names of different organizations, and geographical locations. Named entities were recognized in parliamentary debates in the project War in Parliament. Names of persons, organizations, locations, and miscellaneous entities were extracted from transcripts using a tool for Linguistic Annotation[13]. Named entities are connected with four different properties where each one points to different objects of the triple (either person, location, organization or miscellaneous entity). A literal is created for every named entity found in the speech together with a URI that leads to the Wikipedia page of the entity, in case that page exists.

The semantic model for secondary datasets is straightforward. Both SRU and OAI-PMH protocols allow the client to submit a search and retrieve request for matching records from the secondary datasets. A response on a query containing the matched keywords contains Dublin core properties such as *dc:identifier*, *dc:type*, *dc:publisher*, *dc:date*, *dc:source* and *dc:title* which are used in our PoliMedia semantic model in case of newspaper articles. The model will contain the instance of a newspaper article with a URI that uses a resolver for accessing the OCR text or pdf document at the National Library. Both radio bulletins and newscast datasets have very similar models. The newscast dataset contains very rich metadata about its resources, so except information about the date, type and publisher, this metadata contains spatial information and names of subject that appears in the videos.

As a final result of the first phase of our project, we created an RDF repository that contains around 38,8 million triples, that came from 10,924 XML files containing information about debates in Dutch parliament. Important elements from XML files were extracted using Java libraries (SAX) and RDF triples were created (JENA). The semantic repository is created using OWLIM[6], a software component for storing and manipulating huge quantities of RDF data. OWLIM is packaged as a Storage and Inference Layer (SAIL) for the Sesame OpenRDF framework.

[6] OWLIM – Semantic repository: http://owlim.ontotext.com/display/OWLIMv51/Home

4 Summary and next steps

In this paper we described the process of creating the semantic model for the purpose of building a semantic repository for the PoliMedia project. The semantic repository is filled with triples that describe events and topics that happened in the Dutch parliament and allows us to use queries to fetch interesting information that was not as easily available before (for example, how many times a particular politician spoke of a particular person in the parliament).
As previously stated, the PoliMedia project will be carried out in three phases. Phases that will be carried in the future are phases (2) and (3), with an automatic detection of the semantic links between primary and secondary datasets and the creation of a demonstrator application.
For the creation of links Named Entities (that appears in primary and secondary datasets) will be used to decide whether the media resource is on some way connected to the events discussed in the debates. Important entities are persons but also locations and time. As debate events consist of smaller sub-events, namely speeches of consecutive speakers (as it is expressed in the semantic model described in this paper), we will search for possible links between those sub-events and media items that cover that particular part of the debate. A virtual research environment will be built that allows the exploration of the debate events and media coverage thereof via search and browsing. Next to the use of standard information retrieval libraries (Lucene), navigation options will be implemented that will allow users to browse through the linked datasets of debates and media.

References

1. Maarten Marx: Advanced Information Access to Parliamentary Debates. J. Digit. Inf. 10(6): (2009)
2. War In Parliament project: http://www.clarin.nl/page/about/projects/162#WIP
3. Political Mashup project: http://politicalmashup.nl/
4. van Erp, Marieke et al, :Automatic Heritage Metadata Enrichment with Historic Events. In J. Trant and D. Bearman (eds). *Museums and the Web 2011: Proceedings*. Toronto: Archives & Museum Informatics. Published March 31, 2011. Consulted March 5, (2012).
5. W. van Hage, V. Malaisé, R. Segers, L. Hollink, and G. Schreiber: Design and use of the Simple Event Model (SEM). J. Web Semantics, 2011.
6. Hage, W.R. van, V. Malaisé, G. de Vries, G. Schreiber and M. van Someren: "Combining Ship Trajectories and Semantics with the Simple Event Model (SEM)". In: Proceedings of the 1 stACM International Workshop on Events in Multimedia 73-80, (2009)
7. Dalila Mekhaldi, Denis Lalanne: Multimodal Document Alignment: Feature-based Validation to Strengthen Thematic Links. JMPT 1(1): 30-46, (2010)
8. Y. Lv, T. Moon, P. Kolari, Z. Zheng, X. Wang, and Y. Chang: Learning to model relatedness for news recommendation. In *WWW*, (2011).
9. SRU: Search/Retrieval via URL: http://www.loc.gov/standards/sru/
10. OAI protocol: http://www.openarchives.org/OAI/openarchivesprotocol.html
11. Koninklijke Bibliotheek metadataopslag:http://research.kb.nl/documenten.html
12. Digital Item Declaration Language: http://xml.coverpages.org/mpeg21-didl.html
13. FoLiA: Format for Linguistic Annotation: http://ilk.uvt.nl/folia/

Making Sense of the Arab Revolution and Occupy: Visual Analytics to Understand Events

Thomas Ploeger[1], Bibiana Armenta[2], Lora Aroyo[1,⋆], Frank de Bakker[2], and Iina Hellsten[2]

[1] Computer Science Department VU University Amsterdam
[2] Organization Sciences Department VU University Amsterdam

Abstract. Knowledge on the Web comes in ever larger amounts and in a wider variety of structure and semantics that ever before. In order to exploit this knowledge in different applications, many researchers investigate techniques for making sense of Web data. Objects that the techniques try to identify and extract are, for example, people, organizations, and locations. Many applications though observe how events play an increasingly more important role. Capturing and extracting events for sense making analysis is what this research is aiming for, and in this paper we present the first results and contributions from our research. We consider how events get extracted, how they get conceptualized, and how visual analytics helps to make sense of the represented events. All of this is illustrated in a representative example where driven by questions from social scientists we apply our pipeline to the domain of activism, e.g. Occupy, Arab Revolution.

1 Introduction

Events play an increasingly important role in representing and organizing knowledge on the Web. We observe how more and more applications are centered around events, specially in the Social Web. That is in line with the important role events play in all our lives: social networks and applications for our personal communication include events as central elements. Events are not only important for ourselves for organizing personal information, but the way we use events is also of interest to third parties, such as commercial stakeholders (i.e. event organizers and providers) or social scientists (to model and explain social phenomena). A first step in the process of dealing with events is their representation, e.g. in terms of formats that allow further processing, application and sense making. This is evidenced by recent research projects on the modeling of events (cf. SEM [8], LODE [6] and EO [5]) and projects such as Agora [1] and Poseidon [2] that capture events from unstructured and semi-structured texts, respectively. Capturing and modeling events is the first step towards answering domain questions and sense making.

There are a number of reasons why capturing events from unstructured or even semi-structured text is a no trivial task. Primarily, this is because of the

⋆ Also affiliated with IBM Watson Research Center, NY.

inherent limitations of current natural language processing technology. Additionally, the source texts that are relevant for event information are often scattered and may present information that is incorrect, out of context, or biased. For a complete overview of a certain event, all the different perspectives on the given event would have to be found and captured.

Many of the aforementioned research initiatives concentrate on different aspects of these challenging problems. In this research we focus on the contribution of visual analytics. This research aims to find out if visualizing events based on their properties (e.g. location, type, involved actors, timestamps) would help overcome the aforementioned problems for making sense of events. In this way, we hope to facilitate better understanding of events and their properties, by both social scientists and the general public.

To validate our work and demonstrate the concrete contributions, we conduct our research within the social sciences domain of activist organizations. We identified this to be an interesting use case as activists have always had an impact on the present and have a significant role in shaping the future: *In 2011 activists occupied the West and revolutionized the Middle East.* From a social science point of view, if we look at this use case, we see the importance of the events that are involved - for both individual people to see what is happening in their own local environment, or for scientists to tell and explain what triggered and caused what effect.

It is important to differentiate this event-oriented research from research into issues (cf. [4]) and sentiments (cf. [3]). An event is more clearly delineated in terms of spatial aspects than an issue – an issue is more vague, it doesn't require a spatial location where it takes place. Sentiment analysis, in turn, does not capture one particular event, but rather reactions to that event.

At the same time, as computer scientists we can also be impressed by the challenges to identify and extract these events from the large amounts of textual (user-generated) information available, e.g. in newspapers, personal or organizational blogs, social networks, and other forms of social media. Next to the obvious technical challenges, we can also easily identify that the way in which people, individual citizens or social scientists, can visually overview and subsequently interpret the massive amounts of event information is limited. Not only is the capability to visually overview and interpret limited compared to the size and volume. Also, the fact that events are often perceived from different angles makes it even more difficult to account for the different perspectives. For visual analytics the support of different perspectives provides another interesting challenge. Of course, also the ambition to warrant objective (unbiased) presentation of events plays an important role in social science as well as in our daily lives.

Thus, in this research, the overall goal is to explore the social sciences domain of activist events and their properties and do so with suitable event visualization techniques. This implies that we are contributing with relevant modeling and analysis techniques for event knowledge. The rest of this paper is structured as follows. In Section 2, we describe the activist use case in more detail. In Section 3,

we present a pipeline for extracting, modeling and visualizing events. In Section 4, we present our concluding remarks and expectations for future work.

2 Activist Events Use Case

Our research focuses on activist events. As stated in the introduction, activists have an influence on the present and play a role in shaping the future. Because even a single activist event might have consequences, we think they are worthy of study, by both the general public and social scientists. In this paper we will use events related to the Arab Spring and the Occupy Movement as examples because of their recency and social relevance.

This can be seen in many real-world examples of activist events. One such event was a particular confrontation between police and 'occupy' protesters in New York, where a police officer sprayed 4 protesters with pepper spray. A video recording of the event was uploaded to YouTube [3] and several news outlets picked up the story as it went viral[4,5]. Protesters argued that the use of pepper spray was uncalled for. Initially, the police department defended the officer, saying that the use of pepper spray was appropriate. The officer in question stated that the event was taken out of context. Later, a more detailed investigation concluded that the officer was at fault and he was reprimanded.

In order to make fair and unbiased judgments during such an investigation, it is important to represent different perspectives on an event and to take into account the larger context. Another example event that demonstrates this is the self-immolation of Mohamed Bouazizi, a Tunisian street vendor who set himself on fire to protest after officials confiscated his wares. This event is seen by many as the 'spark' that ignited the Tunisian Revolution and the Arab Spring. Many people see Bouazizi as a martyr, standing up to a dictatorial regime. Nevertheless, whether Bouazizi's exact motivations were personal, political or economic is the subject of some debate[6]. Placing this event in context could help investigators make sense of Bouazizi's motivations. How was he treated by officials when they confiscated his wares? How did officials respond to his complaints? Are there any earlier encounters between him and officials that might have played a role in the process? Detailed information about the event in question and the relations to earlier events are important when investigating Bouazizi's motivation.

As stated in the introduction, while the relevance and need are obvious, from a computer science perspective, it is challenging to identify, extract and aggregate mentions of these events from large amounts of textual information, such as newspapers, blogs, social networks and other forms of social media. In the next section we describe our pipeline for doing so.

[3] http://www.youtube.com/watch?v=TZ05rWx1pig
[4] http://online.wsj.com/article/SB10000872396390443866404577565341948999820.html
[5] http://cityroom.blogs.nytimes.com/2011/09/28/police-department-to-examine-pepper-spray-incident/
[6] http://www.frumforum.com/how-an-entrepreneur-sparked-the-arab-spring/

3 Main Process & Pipeline

To be able to explore the social sciences domain of activist events and their properties, as illustrated in Section 2, we turn to suitable event modeling and visualization techniques. Before we can consider the modeling, we must overcome the challenge of building a corpus of source material (such as newspapers, blogs, social networks and other forms of social media) from which to extract events. The process of constructing a corpus and extracting events is described in Section 3.1. It involves using natural language processing technology to extract enough information from unstructured text to 'build' event descriptions from. In Section 3.2 we describe the type of information we are looking for by describing our event model. This is a necessary step in order to be able to prepare for our main contribution, which is in visual analytics (Section 3.3), where we report on exploratory investigations into available instruments for visual event analysis.

3.1 Extracting Activist Events

In our use case, the event extraction process starts with creating a corpus from a (Web) source of news articles. Currently, we gather articles from The Guardian[7] via a simple search using keywords related to activism. This is a representative example, also since the extraction itself is not the main focus here in this paper but a necessary prior step in the process, as we have argued before. In future work, multiple sources of news articles will be used to strengthen validity.

Using keyword extraction and concept tagging, we attempt to determine the type of event that is described in the article. Actors, locations and timestamps of the event are identified using a named entity recognizer. Relation extraction is used to determine the type of involvement an actor had in an event. The details of this process are beyond the scope of this paper and will be described in a forthcoming paper.

Thus, for each news article in the corpus, we obtain several properties of the event described in the article: The type of the event, the actors involved in the event, the locations at which the event has taken place, and the time at which it has taken place. Additionally, we attempt to identify the type of involvement or 'role' of an actor in an event. We represent this event information using the Simple Event Model (SEM) as described in the next section. Additionally, we try to identify the authors of event information to represent different perspectives or viewpoints on the same event.

3.2 Conceptualizing the Activist Domain

When modeling events, there are a number of interesting additional challenges that come with the nature of events. In particular, we focus on the implications of different viewpoints: events are perceived from different perspectives, are thus being reported in text from different perspectives, and are also consumed (interpreted) from different perspectives.

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

Modelling: ACTEVE In this section we report on the explorations we performed in the use case to make the challenges concrete and explore the possible solutions. As we followed an exploratory and evolutionary way to capture and model the events, we describe the different revisions of the model. By doing this we illustrate and motivate our modeling decisions and define and explain concepts important to this research. We call our model "ACTEVE" for "ACTivist EVEnts".

Initial Model The first version of the model was based on the Social Science concepts for describing networks of activist organizations and their activities. Essentially, the model represented *Organizations* targeting *Companies* through *Campaigns* that consist of *Tactics* with the aim of reaching a certain *Solution* (to a problem). *Campaigns* and *Solutions* were associated with a *State*, e.g. 'resolved' or 'partial'. *Organizations* were modeled with a Type, e.g. 'radical' or 'reformative'. *Campaigns* and *Tactics* were also associated with *Time* and *Location*, e.g. '12-06-12' or 'Amsterdam'. This version of the model can be seen in Figure 1A.

Revised Model After reviewing the initial model, we observed that the focus was much more on the campaigns and that the events were not yet considered as first class citizens in our model, like the event of the self-immolation of Bouazizi in the Arab Spring. The second version of our model was therefore made more event-centered. We introduced the concepts of an *Event* and an *Issue*. *Organization* was changed into an *Actor*. Essentially, the new model consisted of *Actors* organizing *Events* (as part of *Campaigns*) using a *Tactic* and targeting an *Issue* (related to *Company* responsible or causing the *Issue*). Each *Issue* has a *State* associated and each *Event* had a *Location* and *Time* associated. What follows are the definitions of each concept. This version of the model can be seen in Figure 1B.

1. *Event:* An action undertaken by an actor as part of a campaign with the aim of influencing the state of an issue.
2. *Tactic:* Defines the type of an event.
3. *Actor:* May be a person, group, or organization who performs tactics.
4. *Company:* An organization that triggers an issue.
5. *Issue:* Is a topic or problem important to actors and companies.
6. *Campaign:* consists of a set of events undertaken by an actor aiming to influence the state of an issue.

ACTEVE-SEM Model We then considered how the revised ACTEVE model could be expressed with the Simple Event Model (SEM) (Figure 2, [8]), to profit from the way SEM allows for a minimal modeling of events to facilitate interoperability (similar to Lode [6] and the Event Ontology [5]) and SEM's compatibility with external vocabularies. We observed how SEM allows us to express all of the constructs of the ACTEVE model. As can be seen in Figure 2, SEM models events in terms of who did what with what to whom where and when, modeled as *Actors*, *Events*, *Objects*, *Roles* and *Places*, each of which has a *Timestamp*.

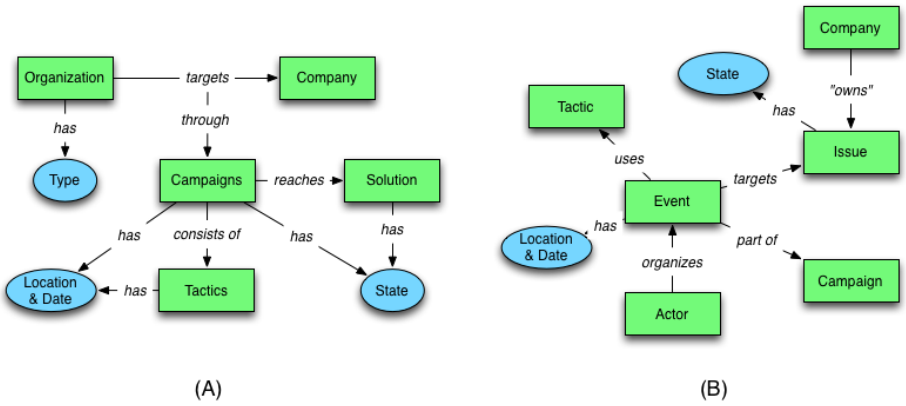

Fig. 1. (A): Initial model: Campaign-centered. (B): Revised model: Event-centered

SEM also allows us to specify certain 'views' on an event. This important concept for ACTEVE is explained in more detail in the next section and is one of the main reasons for choosing SEM over Lode, EO or a custom model.

Modelling Different Viewpoints As stated in the previous section, SEM also allows us to specify certain 'views' on an event, which hold according to a certain authority. This makes it possible to model different perspectives on the same event, which is an important notion as illustrated by the examples in Section 2.

Specifically, SEM allows us to specify three aspects of viewpoints [8]:

1. Event-bounded roles.
2. Time-bounded validity of facts.
3. Attribution of the authoritative source of a statement.

Being able to specify event-bounded roles is an important notion, specifically in our domain, because it allows us to differentiate between role types that hold according to different authoritative sources. In the Occupy event example of Section 2, the police officer might be seen by the protesters as having the role of an 'aggressor', while the officer might be seen by the NYPD as a 'peacekeeper'. This can be expressed in SEM using the 'sem:View' according to some 'sem:Authority' construct.

Similarly, it is useful to be able to specify the time-bounded validity of a certain fact. For example, the role of Mr. Bouazizi in the second example from Section 2 was initially 'street vendor' but changed after a certain point in time to 'martyr', at least according to some people. This can be expressed in SEM using the 'sem:Temporary' construct.

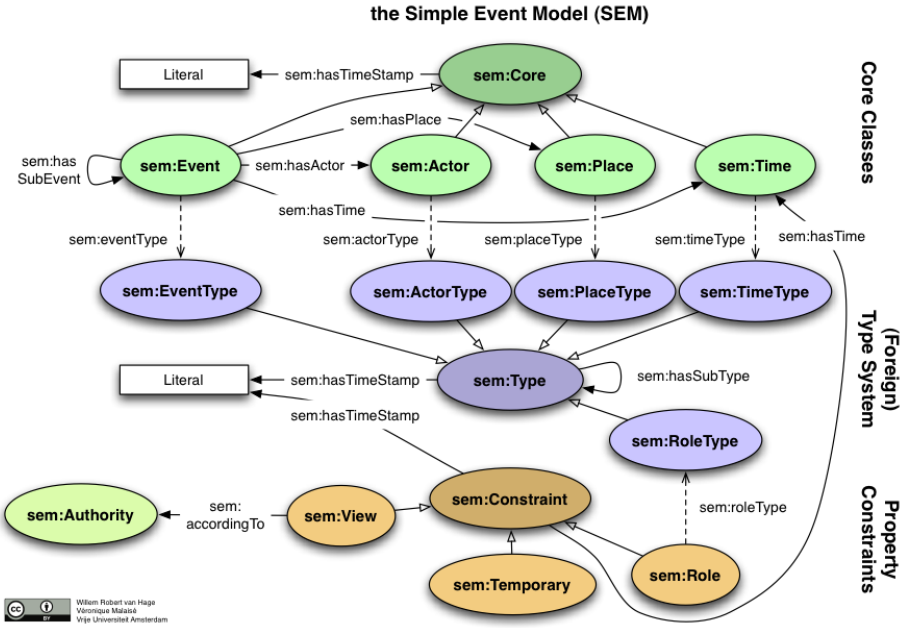

Fig. 2. Simple Event Model constructs

3.3 Visual Analytics for Activist Events

In the previous section we showed how our investigations from the use case learned us how the event information can be represented, including the different viewpoints. In our further use case exploration, we have seen how the representation of event information is not only geared towards a sound and complete storage of the events inside a system: the actual usage of the event information is done as part of the analysis that end-users do. The end-users are the ones that need to make sense of the information. Usually, they are accessing the information with some questions in mind: these questions are the ones that drive them in their explorations of the information. Through interviews with potential end-users (e.g. social scientists and interested lay people) we have identified 3 kinds of questions that could be posed.

1. Simple questions: basic 1-dimensional questions targeting simple statistical analysis, e.g "What is the most common type of event?"
2. Advanced questions: multidimensional questions targeting comparative analysis between different events, e.g. "Which Dutch actors have been involved in Occupy events?" This example covers two dimensions, i.e. actor and event type.
3. Interpretive questions: questions that cannot be answered only through reasoning with our event dataset, but require some interpretation and interaction by the users. For example, for the question "What are the most effective

tactics used by Tunisian actors during the Arab Spring?" we could retrieve statistical information from our dataset, with respect to tactics of Tunisian actors within a certain timespan, as well as issues and their states. However, the answer is in the user's interaction and final interpretation of this information. These questions are the ideal example to motivate the need for visual analytics.

In the context of these three types of questions, we envision the incorporation of different viewpoints to form an essential part of the visual analytics and end-users' exploration of the event information. In the study of our use case examples and domain conceptualization we have already seen several examples that demonstrate this.

To understand the best practices for supporting visual analysis of event information, the first step is to elicit visual analysis requirements for answering the above questions and then map existing tools to them. The ultimate goal that is behind this requirements elicitation of course is to understand what visual analysis support is needed in real-life use cases like the one we are considering here, and how such support could be realized

Visualization Requirements In this section, we present basic requirements for event visualizations. On the basis of these requirements, we will be able to select (or construct) different types of visualization techniques and evaluate their suitability. We distinguish between basic and advanced visualizations, which map to the question types of Section 3.3. For both types of visualizations, we report on how they answer the questions of Section 3.3, what is necessary to represent different viewpoints and the sources according to whom these viewpoints hold (Section 3.2 - Modeling Different Viewpoints).

Basic Visualizations Basic visualizations should present statistics about events and their properties. They use a simple numerical representation (e.g. percentage or ratio) and are typically used for representing statistical information on (parts of) large collections of events. Sorting, ranking and filtering by different criteria should be the most advanced features these types of visualizations have.

Even basic visualizations will have to facilitate representing different versions of the same statistic. This means that there should be a way to show according to whom a certain value is true, to represent different perspectives. Additionally, it should be possible to see to which point in time a certain value belongs, to be able to represent the temporal evolution of a statistic.

Unsurprisingly, these visualizations are best suited for answering the 'simple' questions as defined in Section 3.3. One example of a basic visualization would be a table, which lends itself well to showing one-dimensional information, but is poor for comparing multiple dimensions at the same time. This is where more advanced visualizations are necessary.

Advanced Visualizations Advanced visualizations should allow for links between events and/or links between event properties. They involve comparing multiple

dimensions at the same time, such as both location and time. Advanced visualizations typically map data to visual properties of geometric shapes to reveal trends and patterns in the data. Positioning data points in comparison to each other should be possible for categorical and temporal comparison.

As with basic visualizations, advanced visualizations will have to be able to represent different versions of the same data. Again, this means that there should be functionality to represent and switch between different versions of the same data over time or according to a certain authoritative figure.

Advanced visualizations are intended to answer the multidimensional 'advanced' and 'interpretive' questions of Section 3.3. An example is a scatterplot, where the x- and y-axis are mapped to two dimensions, while a third dimension could be represented by varying size/color of displayed symbols.

Visualization Tools Because of the highly specific nature of the requirements defined in the previous section, it is unlikely that we will be able to find a ready-made visualization technique or tool that meets these requirements. Therefore, it will be necessary to construct new visualization tools or modify existing visualization tools to incorporate the specific functionality that is necessary for event visualizations. In this section, we present a non-exhaustive overview of existing visualization tools that we believe are flexible and extensible enough to be usable in this process. In essence, we report on tools that might be useful when we start constructing visualizations per the requirements described above.

R R[8] is a free software environment for statistical computing and graphics. R allows for both basic one-dimensional statistical analysis as well as more advanced, multi-dimensional visual analytics (cf. [7]). R has a comprehensive library of plugins that extend its base functionality with additional functionality, such as the ability to generate different types of graphs and charts.

Google Chart Tools Google's Chart Tools[9] allow for the creation of various data visualizations, varying in complexity from pie, line or bar charts to maps, timelines or motion charts. The created visualizations have options for interactivity and are easily created on the fly for embedding in websites.

D3.js D3.js[10] is a JavaScript library for creating data visualizations using HTML, SVG and CSS. Like the Google Chart Tools, many types of data visualizations can be created, but D3.js is slightly more flexible as users can create new types of visualizations from scratch instead of having to select from a predefined library.

The study of the requirements and techniques has learned us what is needed to meet the research goal. We have seen how the elicitation of the questions that drive end-users in their analysis have determined the concrete targets for the visualization techniques in terms of features and functionality.

[8] http://www.r-project.org/
[9] http://developers.google.com/chart/
[10] http://d3js.org/

4 Conclusion and Future Work

In this paper we have reported the first results from our work concerning the modeling and analysis of events in the domain of activism. Many applications observe how events play an increasingly more important role. Capturing and extracting events for sense making analysis is what this research is aiming for. Reporting from the concrete context of our activism use case, e.g. Occupy, Arab Revolution, we show how events first get extracted, then how they get conceptualized, and then how visual analytics helps to make sense of the represented events. We emphasized the need to be able to represent different perspectives on events, as well as event properties. We have contributed the first SEM-based model for event modeling in the activism domain and we have identified the objectives and requirements for the visual analysis of these events. In future work we continue mapping the requirements for the visual analysis to the available techniques and tools, to design visual analysis support that can be evaluated with social scientists and lay people in the context of the activism domain.

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 Social Sciences at the VU University Amsterdam. We would like to thank Willem R. van Hage and Jesper Hoeksema from VU University Amsterdam for their help in the NLP tools exploration.

References

1. van den Akker, C., Legêne, S., Van Erp, M., Aroyo, L., Segers, R., Van der Meij, L., van Ossenbruggen, J., Schreiber, G., Wielinga, B., Oomen, J., Jacobs, G.: Digital hermeneutics: Agora and the online understanding of cultural heritage. ACM Web Science Conference (Koblenz, Germany, June 14–17 2011)
2. van Hage, W., Malaisé, V., van Erp, M., Schreiber, G.: Linked open piracy. In: K-CAP. pp. 167–168. ACM (2011)
3. Leetaru, K.H.: Culturomics 2.0: Forecasting large-scale human behavior using global news media tone in time and space. First Monday 16(9), 2 (2011)
4. Marres, N., Rogers, R.: Recipe for tracing the fate of issues and their publics on the web. In: Latour, B., Weibel, P. (eds.) Making Things Public: Atmospheres of Democracy, pp. 922–935. MIT Press, Cambridge (Mass) (2005)
5. Raimond, Y., Abdallah, S.: The event ontology. Tech. rep., Queen Mary University of London (2007)
6. Shaw, R., Troncy, R., Hardman, L.: Lode: Linking open descriptions of events. The Semantic Web 5926(Section 2), 153–167 (2009)
7. Van Hage, W.R.: Sparql package for r - linked open piracy tutorial (2012)
8. Van Hage, W.R., Malaisé, V., Segers, R., Hollink, L., Schreiber, G.: 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)