An Ontology-Based Approach to Social Media Mining for Crisis Management

Vanni Zavarella, Hristo Tanev, Ralf Steinberger, and Erik Van der Goot

Joint Research Center - European Commission

Abstract. We describe an existing multilingual information extraction system that automatically detects event information on disasters, conflicts and health threats in near-real time from a continuous flow of on-line news articles. We illustrate a number of strategies for customizing the system to process social media texts such as Twitter messages, which are currently seen as a crucial source of information for crisis management applications. On one hand, we explore the mapping of our domain model with standard ontologies in the field. On the other hand, we show how the language resources of such a system can be built, starting from an existing domain ontology and a text corpus, by deploying a semi-supervised method for ontology lexicalization. As a result, event detection is turned up into an ontology population process, where crowdsourced content is automatically augmented with a shared structured representation, in accordance with the Linked Open Data principles.

1 Introduction

Monitoring of open source data, such as news media, blogs or micro-blogging services, is being considered worldwide by various security agencies and humanitarian organizations as an effective contribution to early detection and situation tracking of crisis and mass emergencies. In particular, several techniques from text mining, machine learning and computational linguistics are applied to user-generated ("crowdsourced") content, to help intelligence experts to manage the overflow of information transmitted through the Internet, extract valuable, structured and actionable knowledge and increase their situation awareness in disaster management[3].

A massive amount of messages on social media platforms such as Twitter, Facebook, Ushahidi etc. are generated immediately after and during large disasters for exchange of real-time information about situation developments, by people on the affected areas.The main added value of this content is that: a. it is nearly real-time, and typically faster than mainstream news; b. it potentially contains fine-grained, factoid information on the situation on the field, far more densely distributed geographically than official channels from crisis management organizations. Nonetheless, there are several problems that hinder social media content to unfold its full potential for real-world applications.First, content is massive, containing a high rate of information duplication, partly due to several people reporting about the same fact, partly due to platform-specific content-linking practices, such as re-tweeting. Secondly, while social media messages can be generated with a certain amount of platform-specific metadata (such as time,

geocoding and hashtags) a crucial part of their content is still encoded in natural language text[1].

Both issues raise from the same general problem that crowdsourced content is unstructured and lacks interpretation with respect to a shared conceptualization of the domain, so that it cannot be integrated with knowledge bases from humanitarian agency information systems and thus be converted into actionable knowledge. This issue clearly emerged in a survey with disaster management experts active on the field right after the 2010 Haiti earthquake, where Twitter users produced a significant amount of reports, which were though underexploited because of the lack of semantic integration with existing information systems [3].One solution that has been explored consists in providing the users with tools for structuring their content "on the fly", for example by engaging a layer of domain experts in encoding unstructured observations into RDF triples by using the Ushahidi platform ([27]). We propose instead a general architecture for augmenting unstructured user contributions with structured data that are automatically extracted from the same user-generated content. The overall method is top-down.We then apply a semi-supervised method for the lexicalization of the target ontology classes and properties from text [1]. The method learns a mapping from classes of linguistic constructions, such as noun and verb phrases, to semantic classes and event patterns, respectively. Then, with a relatively limited human intervention, such constructions can be linearly combined into finite-state grammars for detection of event reports. Finally, we run the output grammar on crowdsourced content and populate the target ontology with structured information from that content, by deploying an instance of the event detection engine tuned to the social media streams.

As the ontology lexicalization method is language and domain independent, the proposed architecture is highly portable across languages, including for instance the il-formed ones used in social media platforms.

Next section is devoted to related work. Section 3 outlines the architecture of event extraction from news text and describe its implicit domain model. Section 3 proceeds by describing the ontology lexicalization algorithms. Section 5 shows how the resulting ontologies can be populated from user-generated content by information extraction techniques. We conclude by exploring open issues and future developments of the proposed architecture.

2 Related Work

Event detection and tracking in social media has gained increased attention in last years and many approaches have been proposed, to overcome the information overload in emergency management for large scale events, and to increase sensitivity of small case incident monitoring ([20], [17],[29]). Different approaches vary with respect to the level of analysis they perform on crowdsourced content and the amount of structure they extract. This ranges from taking tweets as simple "sensor reading", associated with a

[1] The equally important issue of the confidence and trust of user-generated content, and the specific techniques for content validation which can be deployed to cope with it, are out of the scope of this paper.

time and location, of a target earthquake ([18]), to deriving RDF triples posts for content merging ([19]).

Many studies have explored the use of ontologies for enhancing semantic interoperability in various domains of interest for the crisis management professionals (e.g. spatial data and GIS, [21]). Nonetheless, a comprehensive standard ontology encompassing all those subject areas does not exist, while different sub-ontologies cover critical subject areas (such as Resource, Damage, Disaster) (see [6] for a survey). This poses an issue of ontology mapping and integration for a crisis response information system which would like to make use of Linked Open Data for its data sharing.

Methods for ontology learning and population have been explored in recent years both within the Natural Language Processing community and in the Knowledge Representation field (see [22] and [23] for an overview). Strongly related to our work is the concept learning algorithm described in [24], which finds concepts as sets of semantically similar words by using distributional clustering.

Finally, there are many approaches for text mining from Twitter data ([25]). Relevant to our approach are the methods for automatic event detection from Twitter like the one described in [26]. However, in contrast to the already existing approaches, we perform structured event extraction from the tweets rather than only event detection-based tweet classification.

3 Event Extraction Architecture

We have created a multilingual event extraction engine NEXUS for global news monitoring of security threats, mass emergencies and disease outbreaks [9]. The system is part of the Europe Media Monitor family of applications (EMM) [8], a multilingual news gathering and analysis system which gathers an average of 175,000 online news articles per day in up to 75 languages, taken by a manually selected set of mainstream newspapers and news portals. NEXUS builds upon the output of EMM modules and identifies violent events, man-made and natural disasters and humanitarian crises from news reports. It then fills an event type-specific template like the one depicted in Figure 1, including fields for number and description of dead, injured, kidnapped and displaced people, descriptions of event perpetrators, event time and location, used weapons, etc.

Nexus can work in two alternative modes, cluster-based and on single articles. In the latter the full article text is analysed. In the former, we process only the title and the first three sentences, where the main facts are typically summarized in simple syntax according to the so-called "inverted pyramid" style of news reports. [2].

Figure 2 sketches the entire event extraction processing chain, in cluster-based mode.

First, news article are scanned by EMM modules in order to identify and enrich the text representation with meta-data such as entities and locations, that are typically

[2] This is feasible without significant loss in coverage as the news clusters contain reports from different news sources about the same fact, so that this redundancy mitigates the impact on system performance of linguistic phenomena which are hard to tackle, such as anaphora, ellipsis and long distance dependencies.

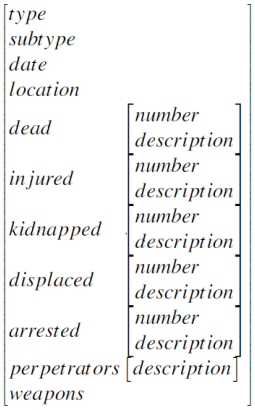

Fig. 1. The output structure of the event extraction system

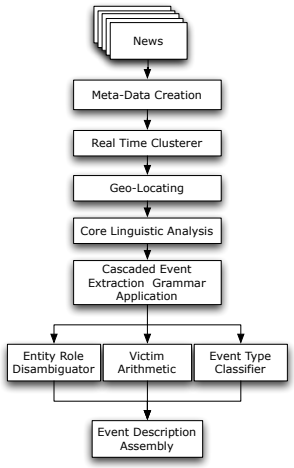

Fig. 2. Event extraction processing chain

separate from the ones deployed in the event extraction process proper. Next the articles are clustered and then geo-located according to extracted meta-data. Each article in the cluster is then linguistically preprocessed by performing fine-grained tokenization, sentence splitting, domain-specific dictionary look-up (i.e. matching of key terms indicating numbers, quantifiers, person titles, unnamed person groups like "civilians", "policemen" and "Shiite"), and finally morphological analysis, simply consisting of lexicon look-up on large domain-independent morphological dictionaries. The aforementioned tasks are accomplished by CORLEONE (Core Linguistic Entity Online Extraction), our in-house core linguistic engine.

Subsequently, a multi-layer cascade of finite-state extraction grammars in the ExPRESS formalism [12] is applied on such more abstract representation of the article text, in order to: a) identify entity referring phrases, such as *persons*, *person groups*, *organizations*, *weapons* etc. b) assign them to event specific roles by linear combination with event triggering surface patterns. For example, in the text "Iraqi policemen shot dead an alleged suicide bomber" the grammar should extract the phrase "Iraqi policemen" and assign to it the semantic role *Perpetrator*, while the phrase "alleged suicide bomber" should be extracted as *Dead*.

We use a lexicon of 1/2-slot surface patterns of the form:

```
<DEAD[Per]> was shot by <PERP>
<KIDNAP[Per]> has been taken hostage
```

where each slot position is assigned an event-specific semantic role and includes a type restriction (e.g. *Person*) on the entity which may fill the slot. ExPRESS grammar rules are pattern-action rules where the left-hand side (LHS) is a regular expression over Flat

Feature Structure (FFS) and the right-hand side (RHS) consists of a list of FFS, which is returned in case the LHS pattern is matched. See Section 5 for a sample rule.

The systems includes an event type classification module consisting of a blend of keyword matching, event role detection and a set of rules controlling their interaction. First, for each event type, we deploy: a) a list of weighted regular expression keyword patterns, allowing multiple tokens and wild cards b) a set of boolean pattern combinations: `OR` pattern lists are combined by the `AND` operator, each pattern is a restricted regular expression and conjunctions are restricted by proximity constraints.

As contradictory information on the same event may occur at the cluster level, a last processing step in the cluster-based mode consists of cross-article cluster-level information fusion: that is, the system aggregates and validate information extracted locally from each single article in the same cluster, such as entity role assignment, victim counts and event type.

The system architecture is highly customizable across languages and domains, by making use of simple local parsing grammars for semantic entities, backed by a number of lexical resources learned by semi-supervised machine learning methods [11]. Currently instances are in place for English, French, Italian, Spanish, Portuguese, Romanian, Bulgarian, Czeck, Turkish, Russian and Arabic language. The live event extraction results are freely accessible online for anybody to use, both in text form (`http://emm.newsbrief.eu/NewsBrief/eventedition/all/latest.html`) and displayed on a map (`http://emm.newsbrief.eu/geo?format=html&type=event&language=all`).

While extracted data are not delivered by the system in any Linked Data standard, a fine-grained categorization of domain entities and relations is implicitly used throughout the extraction process. An ontology representation of the domain model shared by the different instances of Nexus engine is illustrated in Figure 3.

As it can be seen, currently the system is profiled to model crisis event occurrence, rather than emergency tracking and relief operations. However, we experimented with a statistical method for building semi-automatically the core resources of an event extraction engine for a given language, starting from a pre-existing ontology and a text corpus.

4 Ontology Lexicalization Method

The general schema of our method is outlined here (see [1] for more details). Given a set of classes and properties from an event ontology:

1. we learn terms which refer to ontology classes by using a multilingual semantic class learning algorithm. These classes typically represent event-related entities, such as *Building* and *EmergencyCrew*.
2. we learn pre-and post-modifiers for the event-related entities in order to recognize entire phrases describing these entities. We do not give details on this part here (see [1]).
3. we learn surface event-triggering patterns for the properties relating ontology classes. As an example, the pattern *[BUILDING] was destroyed* instantiate the property *involvesBuilding* relating the event class *BuildingDamage* to a *Building* entity.

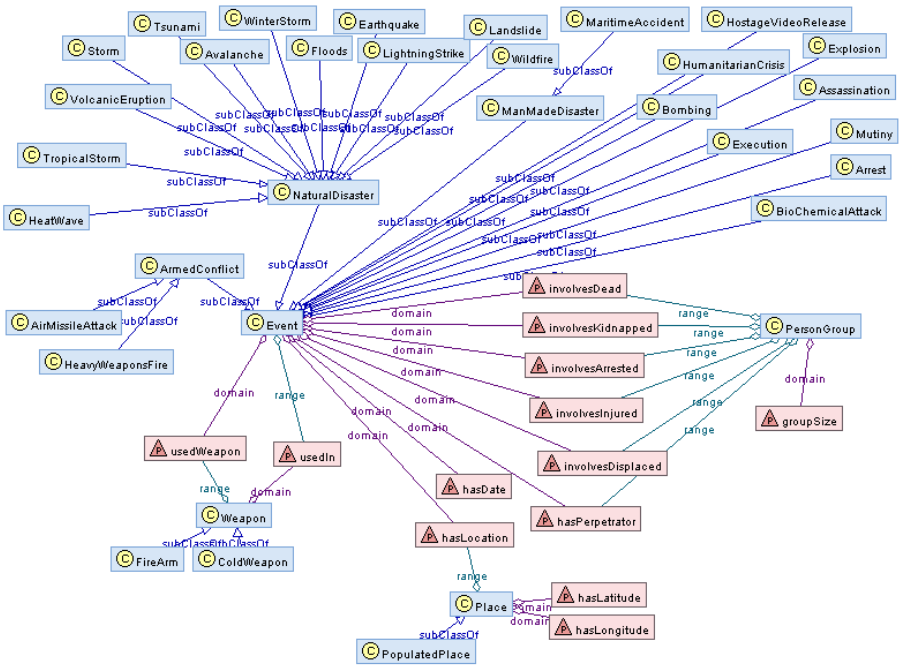

Fig. 3. Ontology representation of the Event Extraction system domain model.

4. Using the lexical classes and patterns learned in steps 1, 2 and 3, we create a finite-state grammar to detect event reports and extract the entities participating in the reported events, using the text processing modules and grammar engine described in Section 3.

4.1 Semantic Class Learning Algorithm

The algorithm we describe here accepts as input a list of small seed sets of terms, one for each semantic class under consideration in addition to an unannotated text corpus. Then, it learns additional terms that are likely to belong to each of the input semantic classes. As an example for English language, starting with two semantic classes *Building* and *Vehicle*, and providing the seed term set *home*, *house*, *houses* and *shop*, and the seed term set *bus*, *train* and *truck*, respectively, the algorithm will return extended classes which contain additional terms like *cottage, mosque, property*, for *Building* and *taxi,*

lorry, minibus, boat for *Vehicle*. The semantic class learning algorithm looks like the following:

input : OriginalSeedSets- a set of seed sets of words for each considered class; Corpus- non-annotated text corpus; NumberIterations- number of the bootstrapping iterations
output : Expanded semantic classes

CurrentSets ← OriginalSeedSets;
for $i \leftarrow 1$ **to** NumberIterations **do**
  CurrentSets ← `SeedSetExpansion(`CurrentSets, Corpus`)` ;
  CurrentSets ← `ClusterBasedTermSelection(`CurrentSets, OriginalSeedSets, Corpus`)` ;
end
return CurrentSets

It makes use of two sub-algorithms for: (a) Seed set expansion; (b) Cluster-based term selection.

The seed set expansion learns new terms which have similar distributional features to the words in the seed set. It consists of two steps: (a)Finding contextual features where, for each semantic class c_i, we consider as a *contextual feature* each uni-gram or bi-gram n that co-occurs at least 3 times in the corpus with any of its seed terms $seed(c_i)$ and that is not composed only of stop words (we have co-occurrence only when n is adjacent to a seed term on the left or on the right); (b) Extracting new terms which co-occur with the extracted contextual features. In other terms, we take for each category the top scored features and merge them into a contextual-feature pool, which constitutes a semantic space where the categories are represented, and represent each term t as a vector $v^{context}(t)$ in the space of contextual features. Then we score the relevance of a candidate term t for a category c by using the projection of the term vector on the category vector.

The presented algorithm is semi-supervised ([16]) and in order to improve its precision we introduce a second term selection procedure: the learned terms and the seed terms are clustered, based on their distributional similarity; then, we consider only the terms which appear in a cluster, where at least one seed term is present. We call these clusters *good clusters*. The increased precision introduced by the cluster-based term selection allows for introducing bootstrapping in our process, which otherwise is typically affected by the problem of error propagation across iterations (so called "semantic drifting").

4.2 Learning of Patterns

The pattern learning algorithm acquires in a weakly supervised manner a list of patterns which describe certain actions or situations. We use these patterns to detect event reports.

Each pattern looks like the ones shown in Section 3. For example, the pattern *damaged a [BUILDING]* will match phrases like *damaged a house* and *damaged a primary school*.

The algorithm accepts as its input: (a) A list of action words, e.g. *damaged*, *damaging*, etc. (b) a representation of the semantic category for the slot as a term list, e.g. *house*, *town hall*, etc. (c) an unannotated text corpus. It then returns a list of additional patterns like *[BUILDING] was destroyed*.

The main idea of the algorithm is to find patterns which are semantically related to the action, specified through the input set of action words and at the same time will co-occur with words which belong to the semantic class of the slot. It consists of three steps (see [1]):

1. find terms similar to the list of action words, e.g. *destroyed*, *inflicted damage*, using the semantic class expansion algorithm above;
2. learn pattern candidates which co-occur with the slot semantic category (e.g. *Building*), using the contextual feature extraction sub-algorithm. Each contextual feature of the slot class is considered a candidate pattern.
3. select only candidate patterns which contain terms similar to the action words (discovered in the first step). In this way, only contextual patterns like *inflicted damage on a [BUILDING]* will be left.

The algorithms presented here make use of no language analysis or domain knowledge, consequently they are applicable across languages and domains. In particular, this makes our method potentially applicable to the ill-formed language used in Twitter and other social media, provided that a relatively large text corpus is available.

5 Populating Disaster Management Ontologies

Once the lexicalization of an ontology has been carried out in a target language, we can apply part of the event extraction infrastructure described in Section 3 to populate that ontology from text streams. In [1] we report about an evaluation on populating a micro-ontology for disaster management with instances of events extracted from a stream of tweets published during several big tropical storms, in English and Spanish language. The ontology structure is shown in Figure 4

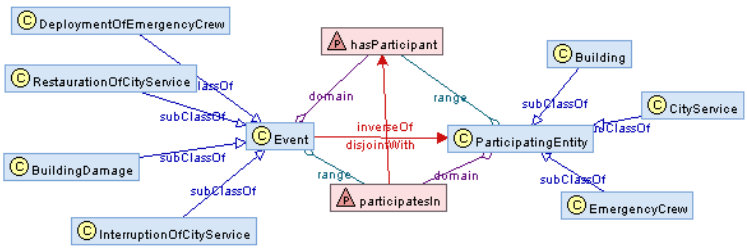

Fig. 4. The micro ontology used for the experiment

Each object from the sub-types of *Event*, namely *BuildingDamage*, *InterruptionOfCityService*, *RestaurationOfCityService* and *DeploymentOfEmergencyCrew*, is related to

exactly one object from a sub-class of *ParticipatingEntity* via *has-a-participant* relation. Consequently, the detection of an event report can be done by detecting an action or situation with one participant. Event detection was performed by devising a simple two-level grammar cascade, whose rules combine semantic classes, patterns and modifiers learned with the previously described algorithms. Namely, it detects at the first level event participant entities: then, it combines them with event patterns to detect event-specific actions or situation, by rules like the one in Figure 5: where the boolean

```
event_rule :> ( leftPattern & [CLASS:"A"]
      participatingEntity & [CLASS:"C", SURFACE:#surf] )
-> event & [CLASS:"A", PARTICIPANT:#surf ]
& PossibleSlotFor(A,C).
```

Fig. 5. Rule schema for single participant event detection.

operator $PossibleSlotFor(A)$ returns a list of all possible sub-classes of *ParticipatingEntity* which may fill the slots of a pattern of class *A*. As an example, when matching an *BuildingDamage* pattern followed by a *Building* entity expression on a tweet like

```
@beerman1991 yeah...i mean, there was thatb  neighborhood
in queens where like, 70 houses burned down during #sandy
```

this rule will populate the micro-ontology with an instance of the *BuildingDamage* event class and an instance of *Building* class, and relate them through an instance of *has-a-participant* property.

On an evaluation performed on a mixed-language English and Spanish corpus of 270,000 tweets, this method proved accurate in extracting events (**87%** and **95%** for English and Spanish, respectively), while recall on a small test set turned up to be very low (**23%** and **15%**). This seems to be due to missing action/situation patterns and to the finite-state grammars not being able to parse pattern-entity combination.

We plan to tackle the current limitation by deploying the ontology-derived grammar within the full-fledged event extraction infrastructure outlined in Section 3, where bag-of-words methods would be put into place, parallel to strict grammar matching, for the instantiation of event type entities and the extraction of participant entities.

To this end, we sketch a general strategy to map the current event extraction engine to the information structure of existing ontologies for Crisis Management. First, we manually map the implicit data model outlined in Section 3 with domain ontologies such as MOAC (`http://observedchange.com/moac/ns#`) or HXL (`http://hxl.humanitarianresponse.info/ns/`) and IDEA (Integrated Data for Events Analysis, see `http://vranet.com/IDEA.aspx`). We believe that, by doing this, we would increase the usefulness of our system output, both for human users working in the field of crisis management or conflict resolution and for services making use of Linked Open Data resources. In some cases, a one-to-one mapping already exists, as for instance between the hxl:PopulationGroup and nexus:PersonGroup classes,

or between moac:Floods and nexus:Floods. In other cases, our ontology event type entities constitute an application of the more generic hxl:Incident entity.

Then, we add the missing, emergency-related sub-ontologies, encompassing classes like moac:InfrastructureDamage, moac:ContaminatedWater and moac:PowerOutag and we customize the system to detect events from them, via the method described in the previous section. Figure 6 illustrates the whole process:

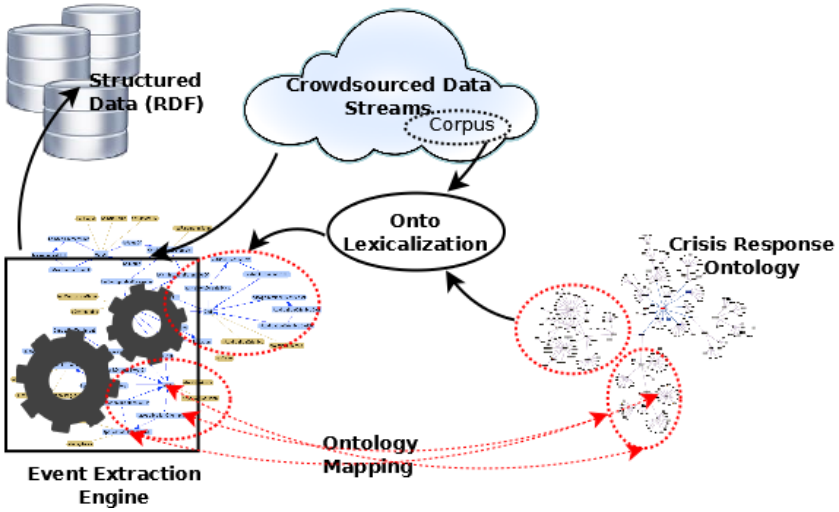

Fig. 6. A general schema for customizing the ontology-based event extraction engine.

6 Summary and Future Work

We outlined a procedure for mapping the domain model of an existing event extraction engine to larger scope ontologies for Crisis Management, in order to perform ontology-based event detection on social media streams. While the core of the procedure is a language-independent ontology lexicalization method that proved promising in supporting text mining from social media, processing of such sources remains challenging because: (a) potentially relevant messages have to be filtered from the large amount of user messages and (b) the usage of ill-formed text in social media messages (lower-casing names, omitting diacritics, doubling letters for stress etc.) reduces the information extraction recall and accuracy of automatic systems. We partially address (a) by generating a keyword-based query to retrieve event-related tweets starting from the text of a news cluster about that event ([28]). This will only be applicable though to large events which make to news clusters, while for the small scale incidents typically targeted for emergency tracking some methods for message duplication detection, clustering and merging appear necessary. (b) is currently addressed by deploying a pre-

processing layer for tweet language normalisation ([13]) but work is still to be done in this direction.

Finally, as a prospective evaluation exercise, we plan to use a test corpus of around 18000 SMS messages (manually translated into English) collected by the Ushahidi platform, in the context of the Mission 4636 relief project, soon after the Haiti 2010 earthquake. We are currently searching for some validation event data from that context for measuring the extractive performance of our system.

References

1. Tanev, H. and Zavarella, V. (under publication) 'Multilingual Lexicalisation and Population of Event Ontologies. A Case Study for Social Media'.
2. Ashish, N., Appelt, D., Freitag, D. and Zelenko, D. (2006) 'Proceedings of the workshop on Event Extraction and Synthesis', held in conjunction with the AAAI 2006 conference, Menlo Park, California, USA.
3. Ortmann, J., Limbu, M., Wang, D., Kauppinen, T. (2011) 'Crowdsourcing Linked Open Data for Disaster Management', Proceedings of the 10th International Semantic Web Conference, Germany, 11-22.
4. Fan, Z., Zlatanova, S. (2011) 'Exploring Ontologies for Semantic Interoperability of Data in Emergency Response', Appl Geomat, 3, 2, 109-122.
5. Gruber, T. (2008) 'Collective knowledge systems: Where the Social Web meets the Semantic Web', Web Semantics: Science, Services and Agents on the World Wide Web, Volume 6 Number 1, pp. 4-13.
6. Shuangyan Liu and Duncan Shaw and Christopher Brewster (2013) 'Ontologies for Crisis Management: A Review of State of the Art in Ontology Design and Usability', Proceedings of the Information Systems for Crisis Response and Management conference (ISCRAM 2013 12-15 May, 2013)
7. Han, Bo, Cook, P. and Baldwin, T. (2012)'Automatically Constructing a Normalisation Dictionary for Microblogs', Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL '12, Jeju Island, Korea, pp. 421–432.
8. Steinberger, R., B. Pouliquen, and E. van der Goot. (2009). 'An introduction to the europe media monitor family of applications'. In Information Access in a Multilingual World - Proceedings of the SIGIR 2009 Workshop, Boston, USA.
9. Piskorski, J. and Tanev, H. and Atkinson, M., Van der Goot, E. and Zavarella, V. (2011) 'Online News Event Extraction for Global Crisis Surveillance', Lecture Notes in Computer Science, volume 6910, pp. 182-212.
10. Tanev, H., Piskorski, J., Atkinson, M. (2008) 'Real-Time News Event Extraction for Global Crisis Monitoring', NLDB '08 Proceedings of the 13th international conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems, pp. 207 - 218.
11. Tanev, H., Zavarella, V. Linge, J. Kabadjov, M., Piskorski, J. Atkinson, M. and Steinberger, R.. (2009). 'Exploiting Machine Learning Techniques to Build an Event Extraction System for Portuguese and Spanish', Linguamatica, 1(2):5566.
12. Piskorski, J. (2007). 'ExPRESSextraction pattern recognition engine and specification suite'. In Proceedings of the International Workshop Finite-State Methods and Natural language Processing.
13. Küçük D., Guillaume J. and Steinberger, R. (2014) 'Named Entity Recognition on Turkish Tweets', Proceedings of the Conference Language Resources and Evaluation' (LREC2014).

14. Xiaohua Liu, Ming Zhou, Furu Wei, Zhongyang Fu, and Xiangyang Zhou. (2012). 'Joint Inference of Named Entity Recognition and Normalization for Tweets'. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pp. 526535.
15. OBrien Sean P. (2002). 'Anticipating the Good, the Bad, and the Ugly: An Early Warning Approach to Conflict and Instability'. Journal of Conflict Resolution 46:791.
16. Tanev,H. and Magnini,B. (2008). 'Weakly Supervised Approaches for Ontology Population'. Ontology learning and population. Bridging the gap between text and knowledge, Berlin: Springer Verlag.
17. Schulz, A. Ristoski, P., Paulheim, H. (2013) 'I See a Car Crash: Real-Time Detection of Small Scale Incidents in Microblogs' Lecture Notes in Computer Science Volume 7955, 2013, pp 22-33.
18. Sakaki, Takeshi and Okazaki, Makoto and Matsuo, Yutaka (2010) 'Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors', Proceedings of the 19th International Conference on World Wide Web, ACM, pp 851–860.
19. Schulz, A. and Ortmann, J. and Probst, F. (2012) 'Getting User-Generated Content Structured: Overcoming Information Overload in Emergency Management', Global Humanitarian Technology Conference (GHTC), 2012 IEEE, pp 143-148.
20. Vieweg, S. and Hughes, A.L. and Starbird, K. and Palen, L., (2010) 'Microblogging During Two Natural Hazards Events: What Twitter May Contribute to Situational Awareness' Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, pp 1079-1088.
21. Guarino, N. (1998) 'Formal Ontology and Information Systems' in FOIS 2004: Proceedings of the Third International Conference. IOS Press pp 315.
22. Drumond,L. and Girardi,G. (2008) 'A survey of ontology learning procedures'. In The 3rd Workshop on Ontologies and their Applications, pp.1325, Salvador, Brazil.
23. Buitelaar, P., and Cimiano, P. (eds.) (2008) 'Ontology learning and population. Bridging the gap between text and knowledge', Berlin: Springer Verlag.
24. Pantel,P. and Lin, D. (2002) 'Discovering Word Senses from Text'. In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 613619. Edmonton.
25. Breslin, J., Ellison, N., Shanahan, J., and Tufekci, Z. (Eds.) (2012) 'Proceedings of the 6th International Conference on Weblogs and Social Media (ICWSM - 12)', Dublin: AAAI Press.
26. Reuter,T. and Cimiano, P. (2002) 'Event-based classification of social media streams'. In Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, Hong Kong.
27. Okolloh, O. (2009) 'Ushahidi, or testimony: Web 2.0 tools for crowdsourcing crisis information', Participatory Learning and Action, vol. 59, no. 1, pp. 6570.
28. Tanev H., Ehrmann M. Piskorski, J. and Zavarella, V. (2012) 'Enhancing Event Descriptions through Twitter Mining'. In: AAAI Publications, Sixth International AAAI Conference on Weblogs and Social Media, pp 587-590.
29. Ritter, Alan and Mausam and Etzioni, Oren and Clark, Sam (2012) 'Open Domain Event Extraction from Twitter'. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,KDD '12, Beijing, China, pp. 1104–1112.