

# I See a Car Crash: Real-time Detection of Small Scale Incidents in Microblogs

Axel Schulz<sup>1,2</sup>, Petar Ristoski<sup>1</sup>, and Heiko Paulheim<sup>3</sup>

<sup>1</sup> SAP Research

<sup>2</sup> Technische Universität Darmstadt  
Telecooperation Lab

{axel.schulz, petar.ristoski}@sap.com

<sup>3</sup> University of Mannheim

Data and Web Science Group  
heiko@informatik.uni-mannheim.de

**Abstract.** Microblogs are increasingly gaining attention as an important information source in emergency management. Nevertheless, it is still difficult to reuse this information source during emergency situations, because of the sheer amount of unstructured data. Especially for detecting small scale events like car crashes, there are only small bits of information, thus complicating the detection of relevant information.

We present a solution for a real-time identification of small scale incidents using microblogs, thereby allowing to increase the situational awareness by harvesting additional information about incidents. Our approach is a machine learning algorithm combining text classification and semantic enrichment of microblogs. An evaluation based shows that our solution enables the identification of small scale incidents with an accuracy of 89% as well as the detection of all incidents published in real-time Linked Open Government Data.

## 1 Introduction

Social media platforms are widely used for sharing information about incidents. Ushahidi, a social platform used for crowd-based filtering of information [12], was heavily used during the Haitian earthquake for labeling crisis related information, as well as incidents such as the Oklahoma grass fires and the Red River floods in April 2009 [19] or the terrorist attacks on Mumbai [3]. All these examples show that citizens already act as observers in crisis situations and provide potentially valuable information on different social media platforms.

Current approaches for using social media in emergency management largely focus on large scale incidents like earthquakes. Large-scale incidents are characterized by a large number of social media messages as well as a wide geographic and/or temporal coverage. In contrast, small-scale incidents, such as car crashes or fires, usually have a small number of social media messages and only narrow geographic and temporal coverage. This imposes certain challenges on approaches for detecting small scale incidents: large incidents, like earthquakes,

with thousands of social media postings, are much easier to detect than smaller incidents with only a dozen of postings, and further processing steps, such as the extraction of factual information, can work on a larger set of texts. For large scale incidents, one can optimize systems for precision, whereas recall is not problematic due to the sheer amount of information on the incident. In contrast, detecting small scale incidents imposes much stricter demands on *both* precision and recall.

The approach discussed in this paper combines information from the social and the semantic web. While the social web consists of text, images, and videos, all of which cannot be processed by intelligent agents easily, the Linked Open Data (LOD) cloud contains semantically annotated, formally captured information. Linked Open Data can be used to enhance Web 2.0 contents by semantically enriching it, as shown, e.g., in [4] and [6]. Such enrichments may also be used as features for machine learning [5]. In our approach, we leverage machine learning and semantic web technologies for detecting user-generated content that is related to a small scale incident with high accuracy. Furthermore, we refined current approaches for spatial and temporal filtering allowing us to detect space and time information of a small scale incident more precisely. We further show how Linked Open Government Data can be used to evaluate classifications.

This paper contributes an approach that leverages information provided in the social web for detection of small scale incidents. The proposed method consists of two steps: (1) automatic classification of user-generated content related to small scale incidents, and (2) prefiltering of irrelevant content, based on spatial, temporal, and thematic aspects.

The rest of this paper is structured as follows: In section 2, we review related approaches. Section 3 provides a detailed description of our process, followed by an evaluation in section 4. In section 5, we show an example application. We conclude in Section 6 with a short summary and an outlook on future work.

## 2 Related Work

Event and incident detection in social media gained increased attention the last years. In this case, various machine learning approaches have been proposed for detecting large scale incidents in microblogs, e.g., in [7],[10],[14].

All those approaches focus on large scale incidents. On the other side, only few state-of-the-art approaches focus on the detection of small scale incidents in microblogs. Agarwal et al. [2] focus on detecting events related to a fire in a factory. They rely on standard NLP-based features like Named Entity Recognition (NER) and part-of-speech tagging. Furthermore, they employ a spatial dictionary to geolocalize tweets on city level. They report a precision of 80% using Naïve Bayes classifier.

Twitcident [1] is a mashup for filtering, searching, and analyzing social media information about small scale incidents. The system uses information about incidents published in an emergency network in the Netherlands for constructing an initial query to crawl relevant tweets. The collected messages are further

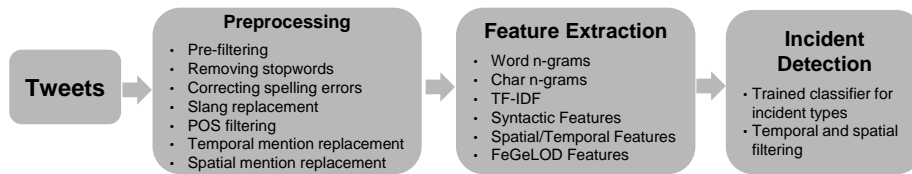


Fig. 1. Pipeline

processed by the semantic enrichment module which includes NER using *DBpedia Spotlight* [11]. The extracted concepts are used as attribute-value pairs, e.g., '(location, `dbpedia:Austin.Texas`)'. In this case, those pairs are used to create a weighting of important concepts for different types of incidents. Furthermore, referenced web pages in the tweet message are provided as external information. A classification of tweets is done using manually created rules based on the attribute-value pairs and keywords. Though they show an advantage of using semantic features, they do not provide any evaluation results for detecting small scale incidents.

Wanichayapong et al. [20] focus on extracting traffic information in microblogs from Thailand. Compared to other approaches, they use an approach which detects tweets that contain place mentions as well as traffic related information. The evaluation of the approach was made on 1249 manually labeled tweets and showed an accuracy of 91.7%, precision of 91.39%, and recall of 87.53%. Though the results are quite promising, they restricted their initial test set to tweets containing manually defined traffic related keywords, thus, the number of relevant tweets is significantly higher than in a random stream.

Li et al. [8] introduce a system for searching and visualization of tweets related to small scale incidents, based on keyword, spatial, and temporal filtering. Compared to other approaches, they iteratively refine a keyword-based search for retrieving a higher number of incident related tweets. Based on these tweets a classifier is built upon Twitter specific features, such as hashtags, @-mentions, URLs, and spatial and temporal characteristics. Furthermore, events are geolocalized on city scale. They report an accuracy of 80% for detecting incident related tweets, although they do not provide any information about their evaluation approach.

In summary, only few of the mentioned approaches make use of semantic web technologies for detecting small scale incidents. Furthermore, all of the mentioned approaches do not compare with events published in governmental data sources, thus the detection of incidents in real-time remains unevaluated.

### 3 Approach

Figure 1 shows our pipeline for detecting incident related tweets. The pipeline is divided into two phases. First, a classifier for detecting incident related tweets is trained. In this case, we crawl and preprocess all incoming tweets. Based on

the preprocessed tweets, several features are extracted and used for classification. Second, for real-time incident detection, new tweets are first filtered based on spatial and temporal filtering. Then the classifier is applied to detect incident related information. For optimizing our classifier, we evaluate the results on incident reports published in Linked Open Government Data. As a result, the optimized pipeline can be used to present valuable information for decision makers in emergency management systems.

### 3.1 Classification

**Crawling and Preprocessing** We continuously collect tweets using the Twitter Search API<sup>1</sup>. In this case, we restrict our search to English tweets of certain cities. Every collected tweet is then preprocessed. First, we remove all retweets as these are just duplicates of other tweets and do not provide additional information. Second, @-mentions in the tweet message are removed as we assume that they are not relevant for detection of incident related tweets. Third, very frequent words like stop words are removed as they are not valuable as features for a machine learning algorithm. Fourth, abbreviations are resolved using a dictionary compiled from [www.noslang.com](http://www.noslang.com). Furthermore, as tweets contain spelling errors, we apply the Google Spellchecking API<sup>2</sup> to identify and replace them if possible.

We use our temporal detection approach (see below) to detect temporal expressions and replace them with two annotations @DATE and @TIME, so we prevent overfitting the classification model for temporal values from the training dataset. Likewise, we use our spatial detection approach (see below) to detect place and location mentions and replace them with two annotations @LOC and @PLC.

Before extracting features, we normalize the words using the Stanford lemmatization function<sup>3</sup>. Furthermore, we apply the Stanford POS tagger<sup>4</sup>. This enables us to filter out some word categories, which are not useful for our approach. E.g., during our evaluation we found out that using only nouns and proper nouns for classification improve the accuracy of the classification (cf. Section 4.3)

**Feature Extraction** After finishing the initial preprocessing steps, we extract several features from the tweets that will be used for training a classifier:

Word unigram extraction: A tweet is represented as a set of words. We use two approaches: a vector with the frequency of words, and a vector with the occurrence of words (as binary values).

Character n-grams: A string of three respective four consecutive characters in a tweet message is used as a feature. For example, if a tweet is: "Today is so hot. I feel tired" then the following trigrams are extracted: "tod", "oda",

<sup>1</sup> <https://dev.twitter.com/docs/api/1.1/get/search/tweets>

<sup>2</sup> <https://code.google.com/p/google-api-spelling-java/>

<sup>3</sup> <http://nlp.stanford.edu/software/corenlp.shtml>

<sup>4</sup> <http://nlp.stanford.edu/software/tagger.shtml>

“day”, “ay ”, “y I”, etc.. To construct the trigram respective fourgram list, all the special characters, which are not a letter, a space character, or a number, are removed.

**TF-IDF:** For every document we calculate an accumulated tf-idf score [9]. It measures the overall deviation of a tweet from all the positive tweets in the training set by summing the tf-idf scores of that tweet, where the idf is calculated based on all the positive examples in the training set.

**Syntactic features:** Along with the features directly extracted from the tweet, several syntactic features are expected to improve the performances of our approach. People might tend to use a lot of punctuations, such as explanation mark and question mark, or a lot of capitalized letter when they are reporting some incident. In this case, we extract the following features: the number of “!” and “?” in a tweet and the number of capitalized characters.

**Spatial and Temporal unigram features:** As spatial and temporal mentions are replaced with corresponding annotations, they appear as word unigrams or character n-grams in our model and can therefore be regarded as additional features.

**Linked Open Data features:** *FeGeLOD* [13] is a framework which extracts features for an entity from Linked Open Data. For example, for the instance `dbpedia:Ford_Mustang`, features that can be extracted include the instance’s direct types (such as `Automobile`), as well as the categories of the corresponding Wikipedia page (such as `Road_transport`). We use the transitive closures of types and categories with respect to `rdfs:subClassOf` and `skos:broader`, respectively. The rationale is that those features can be extracted on a training set and on a test set, even if the actual instance has never been seen in the test set: the types and categories mentioned above could also be generated for a tweet talking about a `dbpedia:Volkswagen_Passat`, for example. This allows for a semantic abstraction from the concrete instances a tweet talks about. In order to generate features from tweets with FeGeLOD, we first preprocess the tweets using *DBpedia Spotlight* in order to identify the instances a tweet talks about. Unlike the other feature extractions, the extraction of Linked Open Data features is performed on the original tweet, not the preprocessed one.

**Classification** The different features are combined and evaluated using three classifiers. For classification, the machine learning library Weka [21] is used. We compare a Naïve Bayes Binary Model (NBB), the Ripper rule learner (JRip), and a classifier based on a Support Vector Machine (SVM).

### 3.2 Real-time incident detection

For real-time incident detection, we first apply spatial and temporal filtering before applying the classifier.

**Temporal Extraction** Using the creation date of a tweet is not always sufficient for detecting events, as people also report on incidents that occurred in the

past or events that will happen in the future. During our evaluations we found out, that around 18% of all incident related tweets contain temporal information. Thus, a mechanism can be applied to filter out tweets that are not temporally related to a specific event.

For identifying what the temporal relation of a tweet is, we adapted the HeidelbergTime [18] framework for temporal extraction. HeidelbergTime is a rule-based approach mainly using regular expressions for the extraction of temporal expressions in texts. As the system was developed for large text documents, we adapted it to work on microblogs. Based on our adaptation, we are able to extract time mentions in microblogs like 'yesterday' and use them to calculate the time of the event to which a tweet refers to. E.g., the tweet 'I still remember that car accident from Tuesday', created on FR 15.02.2013 14:33, can now be readjusted to reference an accident on TU 12.02.2013 14:33.

As discussed above, we also use our adaptation to replace time mentions in microblogs with the annotations @DATE and @TIME to use temporal mentions as additional features. Furthermore, compared to other approaches we are able to retrieve precise temporal information about a small scale incident, which enables the real-time detection of current incidents.

**Spatial Extraction** Besides a temporal filtering, a spatial filtering is also applied. As only 1% of all tweets retrieved from the Twitter Search API are geo-tagged, location mentions in tweet messages or the user's profile information have to be identified.

For location extraction, we use a threefold approach. First, location mentions are identified using Stanford NER<sup>5</sup>. As we are only interested in recognizing location mentions, which includes streets, highways, landmarks, blocks, or zones, we retrained the Stanford NER model on a set of tweets, using a set of 1250 manually labeled tweets. We use only two classes for named entities: LOCATION and PLACE. All location mentions, for which we can extract accurate geo coordinates, such as cities, streets and landmarks, are labeled with LOCATION. The words of the tweets that are used as to abstractly describe where the event took place, such as "home", "office", "school" etc., are labeled with PLACE. The customized NER model has precision of 95.5% and 91.29% recall. As discussed above, we also use our adaptation to annotate the text so that a spatial mention can be used as an additional feature. The following tweet may be the result of this approach: "Several people are injured in car crash on <LOC>5th Ave</LOC> <LOC>Seattle</LOC>"

Second, to relate the location mention to a point where the event happened, we geocode the location strings. In this case, we create a set of word unigrams, bigrams, and trigrams. These are sent to the geographical database GeoNames<sup>6</sup> to identify city names in each of the n-grams and to extract geocoordinates. As city names are ambiguous around the world, we choose the longest n-gram as the most probable city. If there is no city mention in the tweet, we try to extract the city from the location field in the user profile (see [15] for details).

<sup>5</sup> <http://nlp.stanford.edu/software/CRF-NER.shtml>

<sup>6</sup> <http://www.geonames.org>

Third, for fine-grained geolocalization, on street or building level, we are using the MapQuest Nominatim API<sup>7</sup>, which is based on OpenStreetMap data. Using this approach we are able to extract precise location information for 87% of the tweets. Compared to other approaches, which rely on city level precision, we are able to precisely geolocalize small scale events on street level.

**Classification** After temporal and spatial filtering, we apply our classifier to identify incident related tweets. For further refinement and the verification that our approach enables the real-time detection of tweets, we compare our results with the official incident information published in Linked Open Government Data like the data that can be retrieved from [data.seattle.gov](http://data.seattle.gov). Finally, the detected relevant information can be presented to decision makers in incident management systems (cf. Section 5).

## 4 Evaluation

We conduct an evaluation of our method on the publicly available Twitter feed. First, we evaluate the performance of the FeGeLOD features. Second, we measure the performance using all features and third, we compare our results to real-time incident reports.

### 4.1 Datasets

**Training Dataset** For building a training dataset, we collected 6 million public tweets using the Twitter Search API from November 19th, 2012 to December 19th, 2012 in a 15km radius around the city centers of Seattle, WA and Memphis, TN. For labeling the tweets, we first extracted tweets containing incident related keywords. We retrieved all incident types using the “Seattle Real Time Fire 911 Calls” dataset from [seattle.data.gov](http://seattle.data.gov) and defined one general keyword set with keywords that are used in all types of incidents like “incident”, “injury”, “police” etc. For each incident type we further identified specific keywords, e.g., for the incident type “Motor Vehicle Accident Freeway” we use the keywords “vehicle”, “accident”, and “road”. Based on these words, we use WordNet<sup>8</sup> to extend this set by adding the direct hyponyms. For instance, the keyword “accident” was extended with “collision”, “crash”, “wreck”, “injury”, “fatal accident”, and “casualty”.

For building our training set for identifying car crashes, we used the general and the specific keyword set for the incident types “Motor Vehicle Accident”, “Motor Vehicle Accident Freeway”, “Car Fire”, and “Car Fire Freeway” to extract tweets that might be related to car crashes. We randomly selected 10k tweets from this set to manually label the tweets in two classes “car accident related” and “not car incident related”. The tweets were labeled by scientific members of our departments. The final training set consists of 993 car accident related and 993 not car accident related tweets.

<sup>7</sup> <http://developer.mapquest.com/web/products/open/nominatim>

<sup>8</sup> <http://wordnet.princeton.edu>

**Test Dataset** To show that the resulting model using the training dataset is not overfitted to the events in the period when the training data was collected, we collected an additional 1.5 million tweets in the period from February 1st, 2013 to February 7th, 2013 from the same cities. We also used the keyword extraction approach and manually labeled test dataset, resulting in 320 car accident related tweets and 320 not car accident related tweets.

**Socrata Dataset** For evaluating our approach with real-time Linked Open Government Data, we use the “Seattle Real Time Fire 911 Calls” dataset from [seattle.data.gov](http://seattle.data.gov). In the period from February 5th, 2013 to February 7th, 2013, we collected information about 15 incidents related to car crashes and 830 related to other incident types.

## 4.2 Metrics

Our classification results are calculated using stratified 10-fold cross validation on the training set, and evaluating a model trained on the training dataset on the test dataset. To measure the performance of the classification approaches, we report the following metrics:

- Accuracy (Acc): Number of the correctly classified tweets divided by total number of tweets.
- Averaged Precision (Prec): Calculated based on the Precision of each class (how many of our predictions for a class are correct).
- Averaged Recall (Rec): Calculated based on the Recall of each class (how many tweets of a class are correctly classified as this class).
- F-Measure (F): Weighted average of the precision and recall.

## 4.3 Results

**Evaluation of FeGeLOD Features** We used DBpedia Spotlight to detect named entities in the tweet messages<sup>9</sup>. These annotations were used to generate additional features with FeGeLOD, as discussed above. Table 1 shows the classification accuracy achieved using only features generated with FeGeLOD from the training dataset. We used FeGeLOD to create three models with different features. The first one contains only types of the extracted entities, the second one contains only Wikipedia categories of the extracted entities, and the third one contains both categories and types of the extracted entities from the tweets. The best results with accuracy of 67.1% are achieved when using categories and types. Furthermore, analyzing the results from JRip, we get rules using categories as “Accidents”, “Injuries”, or “Road\_infrastructure” and Types as “Road104096066” or “AdministrativeArea”. This shows that the features generated by FeGeLOD are actually meaningful.

<sup>9</sup> The parameters used were: Confidence=0.2; Contextual score= 0.9; Support = 20; Disambiguator = Document; Spotter=LingPipeSpotter; Only best candidate



**Table 1.** Classification results for FeGeLOD features.

| Features           | Method | Training Set |              |              |              |
|--------------------|--------|--------------|--------------|--------------|--------------|
|                    |        | Acc          | Prec         | Rec          | F            |
| Types + categories | SVM    | <b>67.1%</b> | <b>71.3%</b> | <b>67.1%</b> | <b>65.4%</b> |
|                    | NB     | 64.78%       | 68.7%        | 64.8%        | 62.8%        |
|                    | JRip   | 62.42%       | 62.7%        | 62.4%        | 62.2%        |
| Categories         | SVM    | 64.53%       | 72.3%        | 64.5%        | 61.1%        |
|                    | NB     | 63.63%       | 71.9%        | 63.6%        | 59.8%        |
|                    | JRip   | 61.87%       | 72.6%        | 61.9%        | 65.7%        |
| Types              | SVM    | 61.92%       | 63.6%        | 61.9%        | 60.7%        |
|                    | NB     | 58.85%       | 62.6%        | 58.9%        | 55.5%        |
|                    | JRip   | 60.51%       | 60.6%        | 60.5%        | 60.4%        |

**Table 2.** Classification results for all features.

| Features   | Method | Training Set  |              |              |              | Test Set      |              |              |              |
|--|--------|---------------|--------------|--------------|--------------|---------------|--------------|--------------|--------------|
|  |        | Acc           | Prec         | Rec          | F            | Acc           | Prec         | Rec          | F            |
| Word-n-grams+POS filtering, TF-IDF, syntactic, FeGeLOD | SVM    | 88.43%        | 88.5%        | 88.4%        | 88.4%        | <b>89.06%</b> | <b>89.1%</b> | <b>89.1%</b> | <b>89.1%</b> |
|  | JRip   | 86.46%        | 87.0%        | 86.5%        | 86.4%        | 84.21%        | 84.3%        | 84.2%        | 84.2%        |
|  | NB     | 85.81%        | 86.8%        | 85.8%        | 85.7%        | 79.21%        | 85.1%        | 79.2%        | 78.3%        |
| Word-n-grams w/o POS filtering, TF-IDF, syntactic      | SVM    | <b>90.24%</b> | <b>90.4%</b> | <b>90.2%</b> | <b>90.2%</b> | 85.93%        | 86.0%        | 85.9%        | 85.9%        |
|  | JRip   | 84.75%        | 85.5%        | 84.8%        | 84.7%        | 85.93%        | 86.8%        | 85.9%        | 85.9%        |
|  | NB     | 85.86%        | 86.9%        | 85.9%        | 85.8%        | 86.25%        | 87.3%        | 86.3%        | 86.2%        |

**Evaluation of all features** In order to evaluate which machine learning features contribute the most to the accuracy of the classifier, we built different classification models for all the combinations of the features that we described in Section 3.1. We first evaluated the models on the training dataset, then we reevaluated the models on the test dataset. Table 2 shows the best classification results achieved using different combinations of machine learning features.

The tests showed that using word n-grams without POS filtering, TF-IDF accumulate score, and syntactic features provide the best classification results for the training dataset. But re-evaluating the same model on the test set that contains data from a different time period, the results dropped significantly. That leads to conclusion that the model is overfitted to the training dataset. Furthermore, the model contains a large number of features and requires a lot of processing performance to be used for predictions in real-time.

Adding the features generated by FeGeLOD did not affect the classification accuracy on the training dataset, but improved the classification accuracy on the test dataset. This shows that using semantic features generated from LOD helps to prevent overfitting. Additionally, we used POS filtering to filter out some word categories from the tweet to see which word categories contribute to the classification performance. The tests showed that using only nouns and proper nouns when generating the word n-grams improve the results on the test set. This approach significantly reduced the number of attributes in the model, making it applicable for real-time predictions.

**Evaluation on real-world incident reports** To evaluate how many incidents our system can detect compared to governmental emergency systems, we evalu-



Fig. 2. Integration of classified microblogs into the Incident Classifier application.

ated our predictions with the data from the Socrata test set. We correlated each of the 15 car accidents with the incidents from our system, if a spatial (150m) and temporal (+/-20min) matching applies. Using this approach we were able to detect all of the Socrata incidents with our approach. As the average number of tweets identified by our system for each car accident was around ten (with a minimum of only three tweets for one accident), this shows that our approach is capable of detecting incidents with only very few social media posts.

**Performance** For our experiments, we have crawled the Twitter API as discussed above. In the scenario using data from Seattle and Memphis, there were around 100 Tweets per minute. Processing and classifying a bulk of 500 tweets using our trained SVM takes around seven seconds, which is about 14 milliseconds per Tweet.

## 5 Example Application

In [17], we introduced the idea of an information cockpit called Incident Classifier as a central access point for a decision maker to use different types of user-generated content for increasing the understanding of the situation at hand. Based on an aggregation algorithm we introduced in [16], we integrate the classified microblogs in the Incident Classifier. In this case, we apply spatio-temporal filtering and aggregation based on the incident type, e.g., the “car crash” type.

In Figure 2, the aggregation of different incident related information to a small scale incident, including images extracted from referenced web pages in tweets, is shown. E.g., in this case, a picture of the incident as well as the number of involved cars are shown, thus, enabling a decision maker to get additional information about an incident that might be relevant.

## 6 Conclusion and Future Work

This paper contributes an approach that leverages information provided in microblogs for detection of small scale incidents. We showed how machine learning and semantic web technologies can be combined to identify incident related microblogs. With 89% detection accuracy, we outperform state-of-the-art approaches. Furthermore, our approach is able to precisely localize microblogs in space and time, thus, enabling the real-time detection of incidents.

With the presented approach, we are able to detect valuable information during crisis situations in the huge amount of information published in microblogs. In this case, additional and previously unknown information can be retrieved that could contribute to enhance situational awareness for decision making in daily crisis management. In the future, we aim at refining our approach, e.g., to use more sophisticated NLP techniques, exploring the capability of our approach to detect other types of events, as well as including larger sets of open government data in the evaluation.

### Acknowledgements

This work has been partly funded by the German Federal Ministry for Education and Research (BMBF, 13N10712, 01jS12024), by the German Science Foundation (DFG, FU 580/2), and by EIT ICT Labs under the activities 12113 Emergent Social Mobility and 13064 CityCrowdSource of the Business Plan 2012.

## References

1. Abel, F., Hauff, C., Houben, G.J., Stronkman, R., Tao, K.: Twitcident: Fighting Fire with Information from Social Web Stream. In: International Conference on Hypertext and Social Media, Milwaukee, USA, ACM (2012)
2. Agarwal, P., Vaithyanathan, R., Sharma, S., Shroff, G.: Catching the long-tail: Extracting local news events from twitter. In: Proceedings of the Sixth International Conference on Weblogs and Social Media ICWSM 2012, Dublin, Ireland. (2012)
3. Goolsby, R.: Lifting Elephants: Twitter and Blogging in Global Perspective. In: Social Computing and Behavioral Modeling. Springer, Berlin, Heidelberg (2009) 1–6
4. Heim, P., Thom, D.: SemSor: Combining Social and Semantic Web to Support the Analysis of Emergency Situations. In: Proceedings of the 2nd Workshop on Semantic Models for Adaptive Interactive Systems SEMAIS, Springer, Berlin, Heidelberg (2011)

5. Hienert, D., Wegener, D., Paulheim, H.: Automatic classification and relationship extraction for multi-lingual and multi-granular events from wikipedia. In: Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2012). Volume 902 of CEUR-WS. (2012) 1–10
6. Jadhav, A., Wang, W., Mutharaju, R., Anantharam, P.: Twitris: Socially Influenced Browsing. In: Semantic Web Challenge 2009, demo at 8th International Semantic Web Conference, Washington, DC, USA (2009)
7. Krstajic, M., Rohrdantz, C., Hund, M., Weiler, A.: Getting There First: Real-Time Detection of Real-World Incidents on Twitter. In: Published at the 2nd IEEE Workshop on Interactive Visual Text Analytics, Seattle, WA, USA. (2012)
8. Li, R., Lei, K.H., Khadiwala, R., Chang, K.C.C.: Tedas: A twitter-based event detection and analysis system. In: 2011 11th International Conference on ITS Telecommunications (ITST), IEEE Computer Society (2012) 1273–1276
9. Manning, C.D., Raghavan, P., Schütze, H. In: An Introduction to Information Retrieval. Cambridge University Press (2009) 117–120
10. Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., Miller, R.C.: Twitinfo: aggregating and visualizing microblogs for event exploration. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '11, New York, NY, USA, ACM (2011) 227–236
11. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia Spotlight: Shedding Light on the Web of Documents. In: Proceedings of the 7th International Conference on Semantic Systems (I-Semantics), Graz, Austria, ACM (2011)
12. Okolloh, O.: Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information. Participatory Learning and Action **59**(January) (2008) 65–70
13. Paulheim, H., Fürnkranz, J.: Unsupervised Feature Generation from Linked Open Data. In: International Conference on Web Intelligence, Mining, and Semantics (WIMS'12). (2012)
14. Sakaki, T., Okazaki, M.: Earthquake shakes Twitter users: real-time event detection by social sensors. In: WWW '10 Proceedings of the 19th international conference on World Wide Web. (2010) 851–860
15. Schulz, A., Hadjakos, A., Paulheim, H., Nachtwey, J., Mühlhäuser, M.: A multi-indicator approach for geolocalization of tweets. In: Proceedings of the Seventh International Conference on Weblogs and Social Media (ICWSM). (2013)
16. Schulz, A., Ortmann, J., Probst, F.: Getting user-generated content structured: Overcoming information overload in emergency management. In: Proceedings of 2012 IEEE Global Humanitarian Technology Conference (GHTC 2012). (2012) 1–10
17. Schulz, A., Paulheim, H., Probst, F.: Crisis Information Management in the Web 3.0 Age. In: Proceedings of the Information Systems for Crisis Response and Management Conference (ISCRAM 2012). (2012) 1–6
18. Strötgen, J., Gertz, M.: Multilingual and cross-domain temporal tagging. Language Resources and Evaluation (2012)
19. Vieweg, S., Hughes, A.L., Starbird, K., Palen, L.: Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '10, New York, NY, USA, ACM (2010) 1079–1088
20. Wanichayapong, N., Pruthipunyaskul, W., Pattara-Atikom, W., Chaovalit, P.: Social-based traffic information extraction and classification. In: 11th International Conference on ITS Telecommunications (ITST). (2011) 107–112
21. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques. Elsevier, Morgan Kaufman, Amsterdam, Netherlands (2005)