Emergency Event Detection in Twitter Streams Based on Natural Language Processing

Bernhard Klein[1], Federico Castanedo[1], Iñigo Elejalde[1], Diego López-de-Ipiña[1], and Alejandro Prada Nespral[2]

[1] Deusto Institute of Technology (DeustoTech), University of Deusto
Avda. Universidades 24, 48007 Bilbao, Spain
{bernhard.klein,fcastanedo,ielejalde,dipina}@deusto.es
[2] Treelogic, Parque Tecnológico de Asturias, Parcela 30
E33428 Llanera - Asturias, Spain
alejandro.prada@treelogic.com

Abstract. Real-time social media usage is widely adapted today because it encourages quick spreading of news within social networks. New opportunities arise to use social media feeds to detect emergencies and extract crucial information about that event to support rescue operations. A major challenge for the extraction of emergency event information from applications like Twitter is the big mass of data, inaccurate or lacking metadata and the noisy nature of the post text itself. We propose to filter the real-time media stream by analysing posts seriousity, extract facts through natural language processing and group posts using a novel event identification scheme. Based on a manually tagged social media feed corpus we show that false or missed alarms are limited to posts with highly ambiguous information with less value for the rescue units.

Keywords: Emergency detection, social media mining, natural language processing, incremental clustering.

1 Introduction

Online social media applications have become an invaluable tool to gather users feelings and comments and spread them in real time to the rest of the world. Popular social media applications generate a big amount of data during important events. For instance, during last U.S elections Twitter was serving a peak of 15000 tweets per second and currently is sending a billion tweets every two and a half days on average[1]. Several companies are using these social data to perform data analysis and drive marketing decisions. The use of social media applications became very popular in the last disaster events such as hurricane Sandy, the earthquake in Haiti or the tsunami in Fukushima. Observations show that social media is meanwhile also used as an alternative communication tool

[1] see Twitter Blog, http://alturl.com/v4mpe

G. Urzaiz et al. (Eds.): UCAmI 2013, LNCS 8276, pp. 239–246, 2013.

for disasters victims [1]. In fact, people tend to communicate emergency information faster and more effective within their social network rather than using other communication media like phone or email [2]. However, real-time social media communication tools, such as Twitter, are used to communicate and share different type information which is usually not related to emergency detection and management. Thus the signal to noise ratio in these domains is very low and detecting emergency events is like finding a needle in a haystack. In addition, these type of media sources are highly dynamic and make the early detection of an event really complex since there is no historical data about new events.

One of the main goals of the Social Awareness Based Emergency Situation Solver (SABESS) project is the development of an emergency event detection tool for twitter streams to aid the emergency operation and rescue teams in the decision support process. In order to achieve this goal it is necessary to detect an event from a continuous media stream and to provide a good summarization. In this work, we present an approach to the problem of real-time event detection in Twitter streams.

The rest of the article is as follows. The next section present the related works. Then, in Section 3 the problem statement is presented together with our proposed clustering approach. Section 4 describes the research model and the experiment corpus. Experimental results are provided in Section 5. Finally, Section 6 concludes the article.

2 Related Work

Several researchers have worked on similar analysis tools to improve information for rescue teams by exploiting data from social networks: SensePlace [3], the TEDAS system [4] and the Crime Detection Web3 use an iterative crawler which monitors the global Twitter stream to identify emergencies within a given region. Queries are issued as a set of keywords specifying specific time points (July 2010), locations (Houston) and emergency types (car accidents). Their user interface allows rescue organizations to parametrize emergency filters, visualize emergency information on the map and summarize the content of emergency messages through tag clouds. The Twitcident project [5] goes one step further and enriches structured emergency information with data obtained from Twitter streams. They use natural language processing (NLP) techniques, more specifically part-of-speech (POS) tagging and named entity recognition (NER), to tag tweets and enrich tweet contents for incident detection and profiling. Gnip and DataSift are further examples which interface with different social media, provide complex query syntax for more general events and integrate event based information through NLP techniques. Above that, it is important to aggregate tweets which describe the same emergency event. Marcus et al. [6] and Becker et al. [7] describe ways how to cluster tweets based on the inferred topic similarity measured through the keyword distance obtained from an emergency taxonomy. Alarms are automatically issued if the amount of tweets belonging to an event exceeds a certain threshold value. Pohl et al. [8] extend this clustering concept

with the capability of sub-event detection. In case tweet clusters are not strongly coherent, less frequently used keywords in the tweet cluster are used to identify sub clusters which point for instance to different hot spots in the emergency region.

All these clustering approaches use knowledge about a new evolving emergencies e.g. from the 911 hotline to improve the focus of the crawler by specifying more adequate query keywords. This first set of data can then be enhanced through finding similar tweets or better organized by identifying important sub topics. They work, however, is less reliable in identifying new emergencies just from the Twitter stream without a prior knowledge. Within the SABESS project the objective is to identify emergencies completely in a autonomous process.

3 Stream Filter and Event Clustering Approach

For the SABESS project we consider different type of natural and human disasters. These include weather related disasters like hurricanes, flooding and fires but also geological disasters like an earthquake or even health related events like epidemics. Disasters can have varying complexity with respect to scale, spatial distribution and dynamics. Small scale disasters like a fire center around a single hotspot with a coverage range not more than few hundred meters. Large scale events like a hurricane have usually multiple hotspots which can span entire regions and may includes several hundred victims. Of course, large scale events are usually easier to detect as more people would report about them.

From all these messages generated by all the users in the Twitter system we are able to retrieve some messages by using an external API. For this we use a bag of words approach and query the Twitter stream with manually selected keywords frequently appearing in emergency posts like *112, 911, Accident, Affected, Aid, Alarm, Alert, Ambulance, Bodies, Casualties, Collapse, Collateral, Corpses, etc.*. In addition, we limit the potential geographic scope of detected emergencies by specifying observation ranges through the Twitter API. Although the retrieved messages represents approximately a 10% of the complete communication in the system the number of collected tweets may still very large. Given the size of the data, it is important to separate emergency from non-emergency messages in a very fast and effective pre-filtering process. Since a survey [9] shows that the degree of information extracted from tweets strongly correlates with the slang or sentiment degree of a given post, we automatically remove tweets with several letter/punctuation repetitions or other obvious misspellings. Examples for removed tweets are "*Set my life ...ON FIRE!!!!*","*burn baby burn, light a fire*", "*make it pondeeeemmmm whitee boii ya betta runnnnnnnn* and "*Dont be a fire stone bitxhhhhh!!!!*".

The goal is to group posts that share some data about emergency event. Before this can be done, emergency relevant knowledge has to be extracted from the post. Such event data can be extracted indirectly from the metadata of the tweet or directly from the tweet text. Several studies show that the majority of users do not maintain personal profile data nor do they agree on sharing e.g.

spatio-temporal data attached to the tweet. Even if such a metadata is available it is not necessarily secure to use it for the clustering process, as the authoring location may differ significantly from the event location if the user just forwards an emergency message. For this reason we use natural language processing tools to extract emergency facts direct from the post text written by the user. More specifically we apply the Stanford NER library which can extract tags referring to person, organisation, and location knowledge. NLP tools, however, were originally designed for larger texts and may fail not only due to the short size of tweets but also because of their noisy writing style. For this reason we process post texts with adequate slang/text cleaners, and stop word removers prior to the NLP processing. The event grouping process differentiates between two sub processes. First an emergency is classified according to an emergency taxonomy and second a specific emergency event is identified from further clues in the post text. In an abstract view, words in a tweet message can be roughly distinguished in words specifying a given emergency e.g. hurricane sandy, words correlated with emergencies e.g. injured people and relative meaningless stop words providing the kit between the previous word groups.

More formally we define the emergency classification as follows:

Definition (Emergency Classification). Given an emergency taxonomy t we define a message m belonging to the emergency domain if more than n words exist where $\forall x = \{1 \dots n\} w_x \in t \wedge w_x \in m$.

In order to increase the matchmaking probability we apply a tolerant matchmaking approach by comparing the word stem through the startWith() function. In this case abbreviations, plural forms or other word concatenations can still be classified correctly. For the identification of specific emergencies like a fire in Bilbao more complex concatenated expressions have to be considered. Since small-scale emergency events center around a single hotspot (see above paragraph), any location tag found in the tweet text during the preprocessing phase can be used. Because people may refer to locations with varying precision it is important to compare locations along administrative hierarchies e.g. on city or district level. Geocoding services like Google or Geonames provide functions that allow a complete hierarchical specification of a given location. This approach, however, cannot be applied for large scale events where multiple hotspots are usually involved. Here we make use of the fact that humans tend to name bigger emergency events. Disaster names usually follow the emergency category term e.g. hurricane *Sandy*, and can thus be easily extracted from the tweet text. More formally we define an emergency identifier as follows:

Definition (Emergency Identification): An emergency identifier is a concatenated string that is build after following syntax: $\langle emergency\ identification \rangle :=$ $\{\langle emergency\ keyword \rangle + \langle disaster\ name \rangle \vee \langle emergency\ keyword \rangle + \langle location\ hierarchy \rangle\}$ whereas $\langle location\ hierarchy \rangle := \{\langle country \rangle + \langle region \rangle + \langle city \rangle + \langle district \rangle\}$.

4 Research Model and Test Corpus

Clustering algorithms can be evaluated with internal (measuring the similarity) or external assessment approaches (comparing predictions with an 'external' golden standard). As the success of the SABESS project depends finally on the applicability for real-world rescue operations the second approach is the more appropriate.

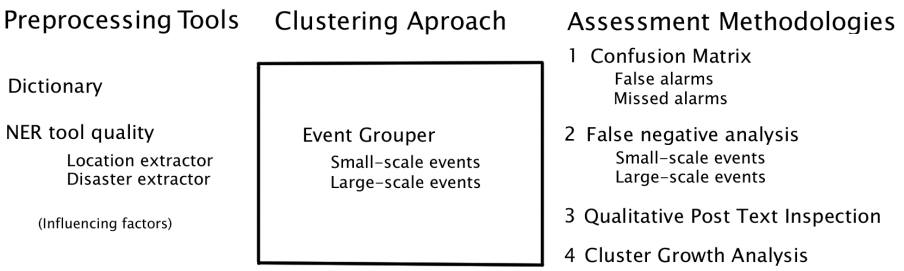

Fig. 1. Research Model

For all experiments we collected a corpus through a Twitter crawler using different emergency keywords. These include events like fires in Tasmania, a cyclone in the Fiji islands, hurricane Sandy and the tsunami in Japan. From this corpus we extracted randomly 1000 tweets for manual tagging. We have developed a tagging tool that enables us to classify posts in emergency and non-emergency messages, and assign each message to an concrete pre-specified emergency event. We further enlarged the tagged corpus up to 10000 tweets with a corpus generator that added new tweets by copying them and randomly replacing a given percentage of the words in the corpus with words from a dictionary. In order to keep the same the emergency classification, words identifying the tweets have been excluded from this process. The analysis of the ground truth (actual emergencies) reveals an almost similar amount of emergency (55%) and non-emergency tweets (45%) in the test corpus.

In the following we describe our research model illustrated in Fig. 1. In a first step we evaluate the overall performance of the event grouper. More specifically, we have been interested in determining the failed detection rate (percentage of incorrect detected emergencies). By representing these results through a confusion matrix, we are able to derive the proportion of missed and false alarms. As rescue operations require a lot of resources and planning both cases need to be considered. As the clustering process generally differentiates between small-scale and large-scale events (see Section 3) it is important to evaluate them separately in more detail. Therefore, the corpus has been separated based on the pre-defined ground truth facts so that one corpus contains only small-scale events and the other large-scale events. For each corpus we perform a false alarm analysis to see the efficiency difference between both approaches. We finalize the evaluation

with an inspection of the incorrect classified posts to gain qualitative impression of the failed cases and ideas on how to improve the event grouper in future.

5 Results and Interpretation

In the following we present the results of the experiments and the corpus we presented above. First we take a look on the confusion matrix. Here we are interested in the missed and false alarms, both are problematic for rescue teams as rescue measures need a non significant amount of time and resources. The majority of events have been correctly classified (see 87% emergencies and 76% of non-emergencies in Table 1), which represents a mandatory prerequisite to build a support tool for rescues. However, still some failures exist. 23% false alarms were generated (tweets classified as emergency although they were not) and 12% of al emergencies have not been detected.

Table 1. Confusion Matrix

Actual \ Predicted	**Positive**	**Negative**
Positive	0.8726236	0.2347826
Negative	0.1273764	0.7652174

Fig. 2 a) shows the true and false positive rate analysis for small and large-scale event clustering techniques. The dotted line represents small scale events (e.g. fire events) whereas the solid line large scale events (e.g. hurricanes). Small scale events show a much higher true positive rate than large scale events. Since large-scale emergencies are identified through a corresponding disaster name, name misspellings (see noisy character of tweets) or incorrect word ordering may lead in some cases to an incorrect event identification. In contrary, the small scale event detection process is less error prone, as missing location references in the text or incorrect identified location tags immediately lead to exclusion of the clustering process by marking them as noise.

Looking at the non detected emergency cases reveals that they have been due to word connections, misinterpreted word order or the mentioning of multiple locations or most often due to unclear message content. An example for unidentified emergency tweets are "*CycloneEvan appeal launched to aid displaced people in Fiji amp Samoa*" or "*Nails hammers tarpaulins blankets arrived from Australia Aid from govt for Western Fiji heading to Lautoka CycloneEvan*".

In order to still assure adequate rescue team support false clusters should not be displayed to the end-user. In the following we want to show how this can be achieved with reasonable effort. Since incorrect identified large-scale clusters are based on individual spelling errors or less frequent occurring word order problems, we can assume that these clusters will (in comparison to correctly identified emergency event clusters) evolve much slower and usually remain small.

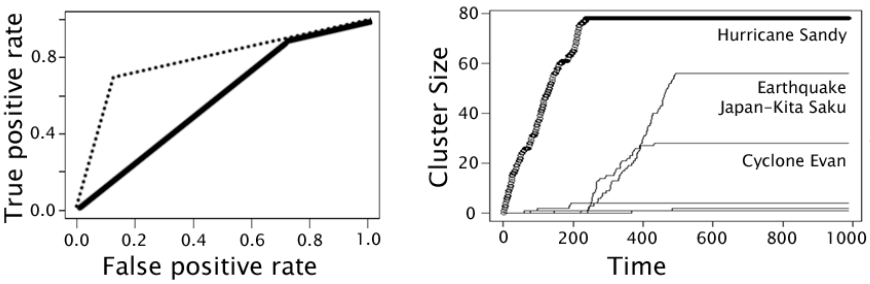

Fig. 2. a) Comparison of small (dotted line) and large-scale (black line) event detection quality and b) evolution of emergency cluster sizes

Figure 2 b) shows exemplary the cluster size increase for the test corpus. The thick line represent the hurricane sandy event (large scale event), and the following curves fires in Australia and Tasmania (small scale events). The curves on the button represent false identified emergency events like *hurricane superstorm*, *hurricane photos*, *hurrican san.* or events which have just emerged e.g. *accident west birkshire*. A threshold value for the cluster size of 10 tweets was good enough in our experiment to remove all meaningless emergency events or incorrect identified emergency events. It is however important to not remove these small clusters from the memory unless the last tweet has not been detected for a very long time ago. This age assumption for a cluster makes sense as tweets usually occur close to the event because the tweet time line for the users is limited.

6 Conclusion

We have presented a novel real-time clustering technique for performing event grouping on public tweet messages. Our approach is based on extracting event information from post texts and therefore outperforms approaches utilizing post metadata. Emergencies are classified based on emergency taxonomies and identified through a widely applied disaster name scheme or alternatively through location information extracted based on natural language processing tools. Posts are finally assigned to specific emergency clusters by a matchmaking approach.

The approach has been evaluated with a tagged emergency corpus containing several disaster events collected during the evaluation period. The results show that the event clustering works quite well and only few emergencies are missed or false alarms created. The application of preprocessing tools such as slang cleaning, word separators and stop word removal generates a positive influence on the results of the event clustering- Whereas small-scale events can be reliably detected by extracting location information through NER tools, large-scale events require an additional post processing step because disaster names can not be safely detected due to spelling errors or word order problems. As these type of problems occur much less frequently than correctly detected events we can do a thresholding step to show only relevant emergency clusters in the user interface.

Acknowledgment. This research was funded by the SABESS project, Innpacto Project Funding of the Spanish government, grant agreement no. IPT-2011-1052-390000.

References

1. Licamele, G.: Web metrics report from Fairfax county (2011), `http://www.fairfaxcounty.gov/emergency/flooding-090811-metrics.pdf` (last visited June 1, 2012)
2. Acar, A., Muraki, Y.: Twitter and natural disasters: Crisis communication lessons from the Japan tsunami. International Journal of Web Based Communities 7(3), 392–402 (2011)
3. MacEachren, A.M., Jaiswal, A.R., Robinson, A.C., Pezanowski, S., Savelyev, A., Mitra, P., Zhang, X., Blanford, J.: SensePlace2: GeoTwitter Analytics for Situational Awareness. In: IEEE Conference on Visual Analytics Science and Technology (VAST 2011), Rhode Island, USA (2011)
4. Li, R., Lei, K., Khadiwala, R., Chang, K.: TEDAS: a Twitter Based Event Detection and Analysis System. In: Proc. of the 28th IEEE International Conference on Data Engineering (ICDE), Washington, USA (2012)
5. Abel, F., Hauff, C., Houben, G.-J., Stronkman, R., Tao, K.: Semantics + Filtering + Search = Twitcident Exploring Information in Social Web Streams. In: 21st International ACM Conference on Hypertext and Hypermedia (HT 2010), Toronto, Canada (2010)
6. Marcus, A., Bernstein, M., Badar, O., Karger, D., Madden, S., Miller, R.: Twitinfo: aggregating and visualizing microblogs for event exploration. In: Proc. of ACM CHI Conference on Human Factors in Computing Systems, pp. 227–236 (2011)
7. Becker, H., Naaman, M., Gravano, L.: Beyond Trending Topics: Real-World Event Identification on Twitter. In: Proc. of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM) (2011)
8. Pohl, D., Bouchachia, A., Hellwagnerr, H.: Automatic Sub-Event Detection in Emergency Management Using Social Media. In: Proc. of the 1st International Workshop on Social Web for Disaster Management (SWDM 2012), pp. 683–686 (2012)
9. Verma, S., Vieweg, S., Corvey, W.J., Palen, L., Martin, J.H., Palmer, M., Schram, A., Anderson, K.M.: Natural Language Processing to the Rescue?: Extracting "Situational Awareness" Tweets During Mass Emergency. In: Proc. of Fifth International AAAI Conference on Weblogs and Social Media (2011)