Detection and Extracting of Emergency Knowledge from Twitter Streams

Bernhard Klein[1], Xabier Laiseca[1], Diego Casado-Mansilla[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,xabier.laiseca,dcasado,dipina}@deusto.es
[2] Treelogic, Parque Tecnológico de Asturias, Parcela 30, E33428 Llanera - Asturias, Spain
alejandro.prada@treelogic.com

Abstract. Increasingly, more important information is being shared through Twitter. New opportunities arise to use this tool to detect emergencies and extract crucial information about the scope and nature of that event. A major challenge for the extraction of emergency event information from Twitter is represented by the unstructured and noisy nature of tweets. Within the SABESS project we propose a combined structural and content based analysis approach. We use social network analysis to identify reliable tweets and content analysis techniques to summarize key emergency facts.

Keywords: Emergency detection, social network analysis, natural language processing.

1 Introduction

Since Twitter is widely adopted today and permanently accessible through the mobile phone it is very well suited for emergency reporting. A Survey [1] conducted in Fairfax, USA showed that the benefit of the informal communication through Twitter lies in the early diffusion of emergency information and the potential to organize mutual help within neighborhoods. The majority of Twitter analysis tools today focus on *trend spotting*, a process where popular hashtags/keywords are extracted from tweets based on their retweet statistics. This approach does not go far enough to reasonably support emergency response systems. An adequate analysis tool should not only identify reliably emergencies (e.g. flooding, storms, earthquakes, tsunamis and epidemics) in a given region like Bilbao in Spain, but also summarize that emergency data from the Twitter community. Experienced Twitter users keep tweets short, mention only important facts, provide adequate links and hashtags for tweet discovery. Good information sources in emergency situations are often public media accounts and increasingly rescue organizations. A good example for a tweet emergency report is the following: *"A 2.5 magnitude earthquake occurred 3.11 mi E of Brea,*

J. Bravo, D. López-de-Ipiña, and F. Moya (Eds.): UCAmI 2012, LNCS 7656, pp. 462–469, 2012.

CA. Details: http://t.co/kqF7Xy8t". Experiences [2] from the tsunami in Japan revealed that analyzing tweets suffers often from hashtag misusage (difficulty to find relevant tweets) and inaccurate emergency information e.g. *"We had a pretty strong earthquake a short time ago :("*. Other tweets may report from historic events e.g. *"After the Haitian earthquake struck in January 2010, the IFRC/Voila app sent 1+million SMS messages daily http://t.co/YwN8"*, describe abstract worries e.g. *"I have a strange feeling that there is going to be a massive earthquake in Mexico City today:(I wanna come home"*, unconfirmed facts e.g. *"So is natural gas drilling to blame for these East Texas earthquakes? What do you think? http://t.co/5rWdwDjn"*. Besides such high level tweet interpretation problems, computer analysis often delivers bad results because of the short length of tweets (140 signs per tweet) and their noisy contents (varying capitalization, frequently used abbreviations, frequent character deletion and other misspellings, word merging and dropping for phrase shortening).

2 Related Work

Several researchers have worked on similar analysis tools to improve information for rescue teams by exploiting data from social networks: SensePlace2 [3], the TEDAS system [4] and the Crime Detection Web[1] 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[2] and DataSift[3] 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. The Twitter reporting process can roughly be divided into two main

[1] `http://canary.cs.illinois.edu/crimedetection/web/`
[2] `http://gnip.com/`
[3] `http://datasift.net/`

phases (see Figure 1). First, several witnesses may report independently from an observed emergency event and in a follow-up step their dependent followers may again spread this information in the Twitter community. Previously described research work focuses on knowledge extraction but falls short to identify *reliable* and *informative* tweets. The examples given in Section 1, however, make clear that filtering misleading tweets is crucial to success. A solution may be found by learning more about the dissemination process an emergency tweet triggers. This is because followers not only spread tweets, but also may confirm/deny, enrich or aggregate emergency facts reported from their friends. In other words, the content summarization can certainly benefit from analyzing the social network structure behind a tweet conversation and using this information to filter superior tweets from the crawler.

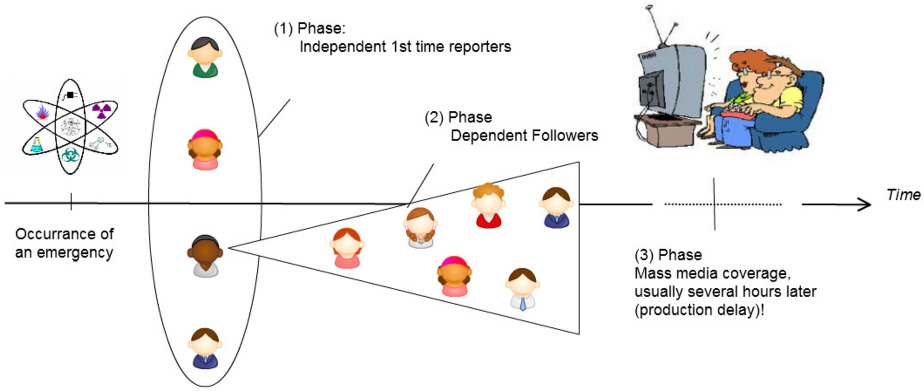

Fig. 1. Emergency reporting timeline

3 SABESS Framework

The goal of this framework is to provide models and tools to analyze the Twitter stream in a more sustainable manner with a combined social and content-aware analysis approach. The framework consists of a Twitter crawler, diverse analysis tools, tweet aggregators and a content summarization component. An interactive crawler monitors the global Twitter stream and extracts relevant tweets by querying the Twitter Streaming API with keywords specifying emergency types and regions. In a follow up process this first Twitter corpus is extended by fetching tweets belonging to ongoing conversations defined by *retweet*, *reply*, *mention* relationships. This type of query data is encoded as metadata in the tweet and thus easy accessible. In order to detect emergencies close to real-time it is important that computer systems analyze a big mass of tweets effectively. For the SABESS project this is achieved with a *pipeline* based architecture which allows to plugin different analysis tools (see Figure 2). This design approach has two main advantages: first, the capabilities of the Tweet analysis can be enhanced just by adding new types of analysis tools and secondly, scalability is managed by replicating computer hardware in dependence of the current analysis demand. ActiveMQ has been selected as our core messaging system. The crawler initially stores the original tweet in a non-sql repository and feeds the messaging system with this tweet. Tweets are formatted in the JSON format and encode various user and tweet related metadata. Analysis tools asynchronously consume tweets from incoming queues, parse their content and enhance the tweet with additional metadata descriptions before they are stored in the outgoing queue. Social network analysis is used to obtain important data from the conversation structure and use this information to filter superior tweets and cluster tweets belonging to the same emergency event. During this process network graphs representing tweet conversations are stored in a graph database like Neo4j. With the help of content analysis additional facts about the emergency are obtained from the tweet text. User data from Twitter, tweet metadata, *credibility* information form the social network analysis and emergency information obtained from the tweet content are used to construct emergency summaries through a matchmaking process. These alert messages are formatted following the Common Alert Protocol (CAP) proposed by the OASIS standard organization as it represents the de-facto reporting standard for todays emergency management systems. For an initial framework evaluation we obtained more than 40000 tweets with "Earthquake" or "Fire" and "Bilbao" as keywords, directly downloaded from Twitter. With this temporary tweet corpus, we perform individual case studies to gain a first impression about real-time capability and the analysis quality of the SABESS framework. In the following subsections we present the analysis processes and our initial experiences in more detail.

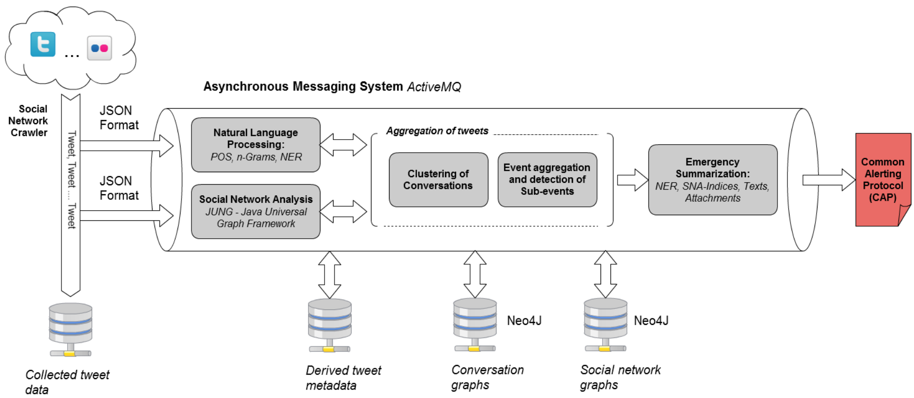

Fig. 2. Tweet analysis realized through a pipeline architecture

3.1 Social Network Analysis

Social network analysis is performed in multiple stages, going into more detail in every phase. Initially, information about tweet authors is retrieved from the Twitter API in order to gain more information about their Twitter experience and influence in the community. Since each tweet represents a communication relation between users, we inspect tweet metadata to learn if it is a single tweet or a tweet in a response to another. Based on this *author* and *relationship* analysis, it is possible to construct social network graphs representing Twitter conversations. Assuming that these graphs are dynamically constructed we are able to identify important witness communities and leading community members close to real-time. The Java Universal Network/Graph (JUNG) Framework has been chosen as a basis for the social network analyzer. JUNG [9] is an open source graph modeling, network analysis and visualization framework written in Java. It supports a variety of graph representations and allows to examine the relations between nodes. We build a social network graph from all collected tweet conversations by obtaining their *user id* as unique node identifier and the *interaction type* e.g. retweet to specify the relationship between the user nodes. JUNG provides a mechanism for annotating nodes and relations with metadata. By adding Twitter specific metadata to the social network graph the analysis can later benefit from their rich expressiveness. Node information is enhanced with the *user name*, the *status count* and *friends/followers count* whenever tweets are parsed by the social network analyzer. By this, the twitter experience of the user and the characteristics of the subscription network can be considered for the network analysis as an additional parameter. Similarly, relationships between the nodes are enhanced with information about the accumulated interaction frequency observed during the analysis period. Figure 3 (right part) shows as an example of an graph metadata annotation for a randomly selected group of Twitter authors. JUNG is able to calculate from the interaction frequency the network centrality (used as indicator for author credibility) for each node and determine social coherent sub communities in the network graph. Various graph metrics are calculated individually for each relationship type. Whereas retweets are considered as a *weak* relation, replies and mentions are considered as a *strong* relationship. The result of this network analysis is shown in Figure 3 (left part), where all Twitter authors are displayed in circular graph (KKLayout) in which leading members are positioned in the center and ordinary members in the edge. The JUNG framework provides filtering mechanisms which allow to filter tweets from the center according to these centrality values, given a manually defined centrality threshold. Executing tests with the tweet data obtained from the crawler described in Section 3 we observed slight delays (40000 tweets render in approximately 5 minutes) during updating the graph display. This delay can be explained with the complete recalculation of the graph whenever a new node/relation is added. Since the graph layout algorithm is very time consuming switching off the visualization component leads to a much better performance. Another solution is to move from an incremental to an interval based recalculation of the graph as the graph evolves only partially.

3.2 Semantic Content Analysis

Following the social network analysis the content of the remaining tweets is analyzed. First, words and separators are identified by the string *tokenizer*. The

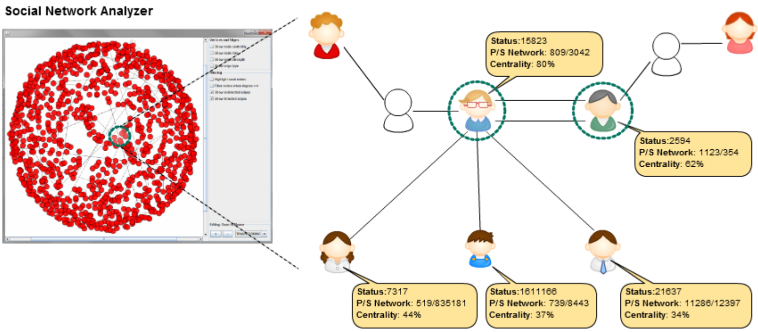

Fig. 3. Example social network graph constructed from collected tweets

result is a list of connected words and text separators like spaces and punctuations. In a subsequent step a grammatical analysis is performed which classifies words in nouns, verbs adjectives etc.. This process is called *part of speech* tagging and utilizes the word context (relationship with other adjacent words) in the tweet text. Based on this knowledge a word *chunker* can group single words in meaningful units that belong together e.g. a combination of family and surnames. This word clustering process leads to specific *n-gram* (sequence of n items) patterns that we use to identify the text language. In order to extract knowledge from texts it is important to associate these chunks with the semantics. This can be achieved through *named entity recognition* algorithms. In the case of an emergency analysis we correlate specific chunks with temporal information, information about mentioned location, involved persons and objects. Figure 4 shows the word tokens, part of speech tags and the named entities discovered for one example tweet. For the language detection we use a library implemented by Shuyo Nakatani. According to his webblog [10] this library detects tweets in 17 languages with 99.1% accuracy. The parser is limited to the Latin alphabet and includes languages like German, English, Spanish, French, Italian and other European languages. Since it is specialized on noisy short text (more than 3 n-grams) it suitable for Twitter. A possible disadvanatage of that library is that the detection quality may suffer from the short text length of tweets, especially if multiple languages are used in one tweet. The results can be improved by repeating the language analysis with aggregated tweet contents belonging to the same conversation or even the same emergency event. As a word tokenizer and POS tagger we currently use the Washington's CoreNLP library[4]. Preliminary tests with different NLP tools have shown that Washingtons word tokenizer and POS tagger work sufficiently fast and accurate enough for the analysis of Twitter streams in real-time. As for the named entity recognition component we choose Stanfords Twitter NLP Library[5], as it demonstrated an improved performance for Twitter. In some cases, however, locations and persons were not properly

[4] https://github.com/aritter/twitter_nlp
[5] http://nlp.stanford.edu/software/corenlp.shtml

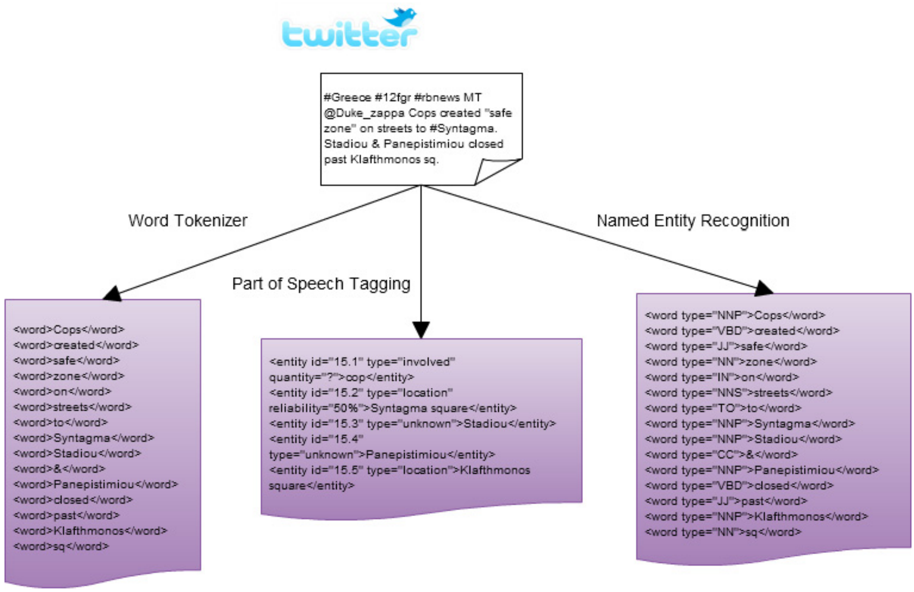

Fig. 4. Example text analysis with derived word tokens, POS and NER tags

detected. The NER tool from the University of Illinois's NLP tools[6] produced the best results in our tests, however, they require a licensing condition which were not appropriate for the SABESS project. Altogether these tests proofed that this combination of text analysis tools works fast and accurate enough to sufficiently detect emergency information.

4 Conclusion

We have presented a novel Twitter analysis framework for performing social network and text analysis tasks on public tweet messages. Our framework is designed for fast and accurate processing in a distributed environment. We demonstrated tools adapted to the specifics of Twitter data and showed that processing of massive tweet streams in real-time is general achievable. Since new modules can easily be added to the messaging pipeline the analysis framework can be dynamically scaled for online processing of streaming data. However, further surveys are needed. Since the filtering functionality is not yet fully implemented tests whether the individual Twitter experience, embedded social networks or the author centrality is a better predictor for the selection of high quality tweets are still outstanding. Moreover we observed that the named entity recognition worked not always well on individual tweets. Therefore we plan to evolve the conversation clustering to an emergency event aggregation. By this larger tweet texts are available which in return will very likely improve these results. A future

[6] `http://cogcomp.cs.illinois.edu/page/software`

version of our system will extend the graphical tool so that users can inspect the data and the outcomes of the analysis.

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), `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. Madadhain, J., Fisher, D., Smyth, P., White, S., Boey, Y.B.: Analysis and visualization of network data using JUNG. Journal of Statistical Software 55(2), 1–25 (2005)
10. Shuyos Weblog, `http://shuyo.wordpress.com/2012/02/21/language-detection-for-twitter-with-99-1-accuracy/` (last visited: June 02, 2012)