Active Hazard Observation via Human in the Loop Social Media Analytics System

Zhenyu Wen, Jedsada Phengsuwan, Nipun Balan Thekkummal, Rui Sun,
Pooja jamathi-Chidananda, Tejal Shah, Philip James and Rajiv Ranjan
Newcastle University, UK

ABSTRACT

We demonstrate AHOM, a system that can Actively Observe Hazards via Monitoring Social Media Streams. AHOM proposes an *active* way to include the human in the loop of hazard information acquisition for social media. Different from state of the art, it supports bi-directional interaction between social media data processing system and social media users, which leads to the establishment of deeper and more accurate situational awareness of hazard events. We demonstrate how AHOM utilizes Twitter streams and bi-directional information exchange with social media users for enhanced hazard observation.

ACM Reference Format:

Zhenyu Wen, Jedsada Phengsuwan, Nipun Balan Thekkummal, Rui Sun,, Pooja jamathi-Chidananda, Tejal Shah, Philip James and Rajiv Ranjan . 2020. Active Hazard Observation via Human in the Loop Social Media Analytics System. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340531.3417430

1 INTRODUCTION

Effective response [1, 3] to crises and hazard events such as landslides, floods, fires, hurricanes, tsunamis, and man-made hazards is dependent on the availability of historical data as well as on the effective real-time integration and utilization of data streaming from social media feeds (such as Facebook, Twitter, and Weibo). However, the existing social media data processing and/or acquisition methods are solely based on Machine Learning (ML) and Natural Language Processing (NLP) classifiers, while lacking the capability to include the human experts who can contribute to the data collection and processing loop in the real-time (i.e., while hazard event is unfolding).

This leads to following drawback: the information extracted by pre-defined ML and NLP classifiers may miss the information about antecedent hazards that leads to full-fledged disaster. To illustrate this shortcoming, let us consider the following real-world example from our research project (http://www.landslip.org/) where a twitter user posts a message to report hazards such as leaning utility poles, trees cracking, or collapsed road beds in their village. Given these events, the social media data analysis system can potentially predict the likelihood of the occurrence of a more serious hazard, such as landslide, if some further contextual information can be collected from the Twitter user (human in the loop) such as the location of the village and whether there have been rainfall and flooding events in the past few hours/days.

In order to study how the social media users and human experts can more *actively* contribute to data analysis (collection and processing) loop for natural hazard response and planning, we develop the AHOM system, which has the following unique features that differentiate it from previous social media data analysis systems.

Human in the loop AI system. To enable bi-directional interaction with the social media users, AHOM uses a novel ontology [7], namely the Landslip Ontology (*LO*), that abstracts the landslide experts' knowledge showing the relations among landslip, landslip warning signs, and other potential occurrence of hazards. This enables a new generation of interaction between data processing systems and social media users, based on various "what if" scenarios modeled by the ontology-based data integration and querying engine. This way the exhaust of social media is used to develop more deep situational awareness of disaster events.

Integrative data management pipeline. As a proof of concept, we develop a social media data processing pipeline (systems) which comprises of a Stream processing engine (Kafka), NoSQL database (Elastic search), Natural Language Processing (NLP) engine (spaCy[1]), and a novel Landslip Ontology for data integration and querying (an Ontology-based data integration and querying (Triplestore) engine). While Kafka and Elastic search are capable of accommodating real-time and historical social-media feeds respectively. spaCy is used and interacted with Kafka stream processing APIs, passively extracting information from social media platforms in real-time. *LO*, which is hosted in an ontology database, Triplestore, enables the generation of automatic and interactive follow-up questions based on various "what if" scenarios modeled by the ontology. A running version of our system is available at GitHub[2], and the live demo is available at here[3].

2 SYSTEM OVERVIEW

2.1 System architecture

Essentially, AHOM is a loosely-coupled run-time system that allows the human experts to participate in the information acquisition. As shown in Figure 1, there are four main components in AHOM: **Storage system**, **Stream processing**, **Human machine interaction** and **AHOM API**. First, social media data streams from various

[1] https://spacy.io/
[2] https://github.com/ncl-iot-team/active-hazard-monitoring
[3] https://bit.ly/2V9MkG4

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3417430

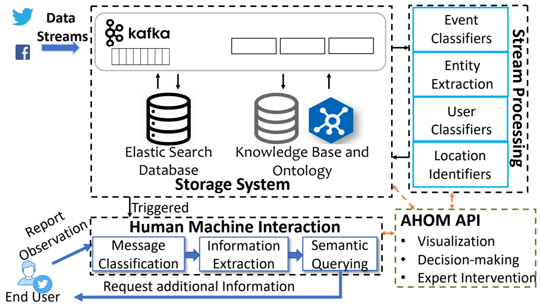

Figure 1: AHOM architecture

sources are injected into the **Storage system** where Apache Kafka is used to

consume the input streams. Next, the **Stream processing** component applies a set of machine learning models (mainly NLP models) to process the injected streams and the outputs of the models are stored in an Elastic Search Database for further queries. The machine learning models are containerized as microservices that are easy to plug-in and plug-off to AHOM via the Kafka publish/subscribe message system. The details of each machine learning model are illustrated in §2.2. When the pre-defined antecedent hazard events are detected and the number of the detected events exceeds a threshold, the **Human machine interaction** framework utilizes the *LO* that we developed in [7] to strengthen the awareness of the situation that domain experts and/or the hazard managers may seek further information from the social media users based on automated follow-up questions generated by our system (see §2.3). Thus, the higher resolution information particularly interested by human experts can be collected from users' replies with a few iterations. Finally, **AHOM API** provides a dashboard to visualize the information of the detected events, extracted from massive social media data.

2.2 Data processing pipeline

In this section, we discuss the details of each machine learning model and its execution workflow. First, the data streams are fed to *User Classifiers* that identify who is posting this message. In this demo, we only consider the classification of two type of users: one is an official account such as Meteorological Office; the other is for normal users. In general, the information provided by official accounts is more reliable compared with normal users, but the normal users can provide richer information. Since Twitter and Facebook already provide the user information, we use the Twitter handle to classify the users based on a dictionary. Next, the *Event Classifiers* are used to identify whether a message relates to landslide hazard and antecedent hazard events (i.e., warning signs). These classifiers are developed by using spaCy a NLP framework. To this end, we collect a small amount of labeled data and then use the dataset to retrain spaCy's convolutional neural network (CNN) models [5, 8] to improve the accuracy of detecting events (these models will also be used to support the *Human machine interaction* in §2.3). Obtaining the geolocation information from the mentions in the message is also essential for analyzing landslides. This task consists of two steps: *Information Extraction* and *Location Identification*. The *Information Extraction* component aims to extract the named entities of a message via `en_core_web_lg` model (available on spaCy), a CNN model trained on OntoNotes[4]. These extracted named entities are the inputs of the *Location Identification* that classifies the named entities as including geolocation information or not. If yes, these geo-names are converted to geo-coordinates by OpenStreetMap (OSM) datasets [4] using geocoding method[5]. A single geo-name may have multiple entries in the OSM dataset. For this demo purpose we take the first entry from the OSM dataset. As future work we will seed the module with the location of interest. Finally, all the outputs are published to Kafka and stored in Elastic Search Database for further analysis.

Scalability. Data processing pipeline runs on Apache Kafka[6] which provides parallelism using data partitioning and consumer groups. In the demonstration system, for the Twitter topic, we used 3 partitions with the Twitter handle as the partition key. The system can be scaled easily by increasing the number of partitions. The number of workers for each processing step in the data processing pipeline can be scaled up to the number of partitions for the given topic. Kafka cluster consists of multiple brokers and a degree of replication ensuring scalability and resilience.

NLP model accuracy evaluation. In this demo, we trained our models with a small dataset of 5000 tweets. The data is collected using Twitter streaming API using keywords landslide and flood. We manually labeled the dataset in two regards: 1) we labeled tweets related to landslide and its antecedent hazard events; 2) for each tweet, we extracted the geolocationnamed entities e.g., country, state, street etc. The trained models have very good accuracy, where the *Event detection* model only for landslide hazard achieves 92% accuracy and the *Geolocation extraction* model extracts geolocation named entities with 87% accuracy.

2.3 Human machine interaction

Key idea. Based on input obtained from domain experts, *LO* is a knowledge model representing information about natural hazards, including hazards such as landslide, flood, rockfall, and earthquake; hazard warning signs such as rainfall and leaning utility poles; and observers such as social media users. The actual events detected or collected from social media can be mapped to the *LO*. *LO*, together with the actual events, acts as a knowledge base that can be queried and used for decision making. Based on the mapping, the "hot zone" of the knowledge base can be identified, which represents information that is of special interested to the decision makers and human experts. Next, the missing knowledge in this "hot zone" can be used to generate the follow-up questions for social media users to obtain further contextual information vital for decision making.

To this end, we develop a framework of human-machine interaction based on *LO* and the knowledge base discussed above and shown in Figure 1. This framework utilizes machine learning techniques implemented in the stream processing component to understand and extract landslide-related information from a large number of social media users and then proactively acquire the missing landslide related information from specific users. The framework consists of three main steps.

[4] https://wiki.openstreetmap.org/wiki/Downloading_data
[5] https://nominatim.org/

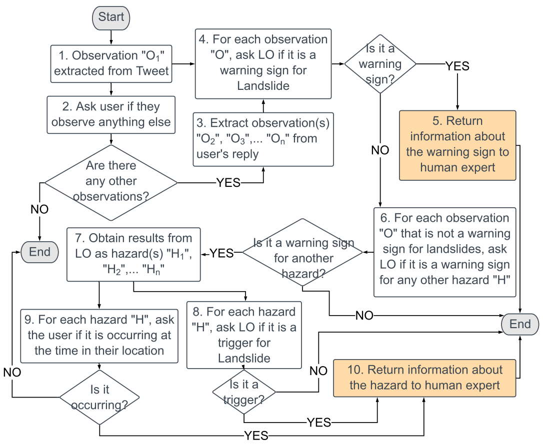

Figure 2: Human machine interaction framework and its execution flow

Message Classification — event classifiers developed in the stream processing component are utilized to collect the tweets related to the context of landslide hazard and landslide warning signs.
Information Extraction — allows the machine to extract actual contents from a tweet using the NLP techniques. For example, we use the location identifier to extract the geolocation information. The essential information for landslide events extracted from the social media includes location, time and warning signs. This information will be used in the next process to investigate missing pieces of landslide related information.
Semantic Querying — provides more comprehensive information about events for decision making. In AHOM, we present three types of information using NLP techniques and ontology based querying: i) essential information about an event (such as location, time, and observation of events), ii) warning signs to indicate potential occurrence of landslides, and iii) potential occurrence of other hazards that could act as triggers for landslide. The aim is to provide as much relevant information as possible to support human experts in making informed decisions. AHOM uses *LO*, to provide information about warning signs and other related hazards to enable human experts to predict the likelihood of the occurrence of landslides with increasing certainty. As can be seen from Figure 2, vital information about warning signs and related hazards is obtained from *LO*.

We use the following example to explain the process depicted in Figure 2. Using NLP techniques, it can be determined that a user has tweeted about a *leaning pole* (step 1 in Figure 2). The system then seeks further clarification from the user by asking if they have noticed or observed anything else such as *rainfall* (step 2). The observations from the user's replies are extracted (step 3) and fed into *LO* to determine the relation of these observations (*leaning pole* and *rainfall*) with landslide and also to extract further relevant information. For each observation extracted from the tweet and the user's subsequent replies, SPARQL query [2] is used to ask the *LO* whether it is an indicator (warning sign) for landslide (natural hazard) (step 4). The shortened query is presented below:

```
ASK {?observation:isWarningSignFor?landslide}
```

If the answer to the query is 'true' (yes), then the information about the warning sign is returned to the human expert (step 5). For example, in this case, the *leaning pole* is a warning sign for landslide as modeled in the *LO* and hence the answer would be true. If, however, the answer obtained is 'false' (no) then for each observation in question, *LO* is further queried to determine if the observation is an indicator for any other hazard(s). For example, since *rainfall* is not by itself an indicator of landslides according to the relationship modeling in *LO*, the answer obtained for the above query for *rainfall* would be 'false'. In this case, the following question is asked of *LO* to determine if rainfall is an indicator of any other hazard (step 6):

```
SELECT ?hazard
WHERE {:Rainfall:isWarningSignFor?hazard}
```

If the query yields an answer (step 7), which in this case would be *flood* and *rockfall*, then follow-on questions are asked of the *LO* about the hazards identified. In this example, the questions would be whether *flood* and/or *rockfall* can trigger landslides (step 8):

```
ASK {:Flood:triggers?landslide} and

ASK {:Rockfall:triggers?landslide}
```

If the answer is 'true', then the information about the type of hazard is returned to the human expert (step 10). In parallel with step 8, the user is also asked whether the hazard(s) (obtained as answer to the query in step 7, which in this example are *flood* and *rockfall*) are occurring or have occurred in their location (step 9). If the answer is 'true', then this information is presented to the human expert (step 10). Thus, in this example where the hazard of interest is landslide, in addition to the location and time of each observation, the following critical information is presented to the human expert with the help of *LO*: i) *Leaning pole* has been observed, which is a warning sign for the occurrence of landslides and ii) *rainfall* has been observed, which is a warning sign for *flood*, which in turn can trigger landslides. Further, the user has confirmed there is *flooding* in their location (see §3.2).

The human expert thus receives information about possible indicators or warning signs for a natural hazard as well as information about other hazards that may eventually lead to the hazard in question. This example demonstrates how rich semantic querying with *LO* can help to identify further relevant information, which may not have been otherwise directly available, thereby providing more comprehensive knowledge that is essential for informed decision making.

3 DEMONSTRATION

This section gives a demonstration of AHOM, with experiments conducted with Twitter via its commercial APIs.

3.1 Experiment setup

The core parts of AHOM (i.e., *Stream processing* and *Semantic query*) were implemented in the Python language, and deployed on an Ubuntu server with 20 cores (Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz) and 64 GB memory. The *Storage System* was deployed on another server with the same configuration. The Kafka cluster was set up using Apache Kafka 2.12, the *Elastic Search Database*

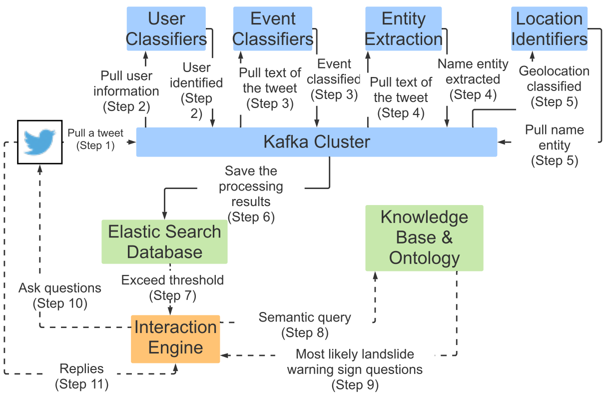

Figure 3: Execution pipeline

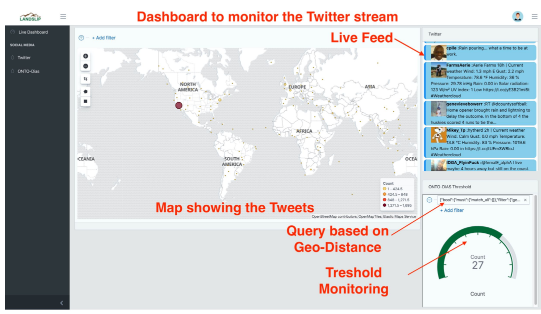

Figure 4: Screenshot of stream monitoring

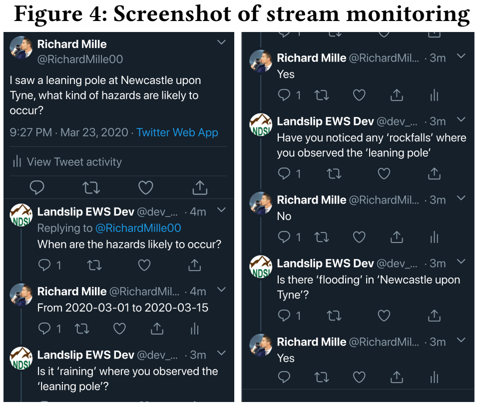

Figure 5: Screenshot of Twitter interaction

and *Knowledge Base* were developed by using Elastic search 7.6 and Stardog respectively.

3.2 Execution sequence

This subsection illustrates the execution sequence of our AHOM to demonstrate its ease-of-use and support for disaster management (see Figure 3).

Stream monitoring. The Twitter stream APIs are used to pull tweets, using a set of subscribed keywords generated by landslide experts, e.g., landslide, landslip and land movement etc [9] (see Step 1). Then, the user accounts are identified in Step 2. Step 3 leverages the trained NLP models to classify landslide related events. If the tweet is related, in Step 4 and 5, the geolocation name entities are extracted (if they exist), via the models discussed in §2.2. Finally, all processing results are stored in *Elastic Search Database* (Step 6) and visualized in real-time as shown in Figure 4, where the size of cycle represents the number of tweets i.e., the bigger size the higher the number of tweets. Moreover, our system can monitor and visualize over a defined geographical area. For example, the right side of Figure 4) shows the number of tweets related to landslide within a 200KM radius of Newcastle upon Tyne. If the number of tweets exceeds the predefined threshold (i.e., 50 in this demo), the system will randomly select some tweets and ask some questions (see Step 7).

Semantic query. Figure 5 is a snapshot of the human-machine interaction system in action. The series of questions posed here for the user and the reasoning behind them is explained in *Semantic query* in §2.3 with reference to Figure 2. The process described in Figure 2 corresponds to steps 8 - 11 in Figure 3, which gives the big picture view of the execution pipeline. These steps in Figure 3 represent the bidirectional interaction of the interaction engine with *LO* (steps 8 and 9) and of the engine with the social media user (steps 10 and 11).

4 CONCLUSION

Social media and other unstructured data are increasingly important for natural hazard management by augmenting traditional data sources used by, for example, landslide scientists. This demo paper shows that our AHOM is able to process massive amounts of data from social media to provide meaningful content to emergency responders, planners and local and national decision makers. Additional benefits accrue by enhancing the completeness of a dataset through automated question-based information gathering which in turn improves perceived trust and reliability in the data collected.

ACKNOWLEDGMENTS

This work is supported by Landslide project (NERC– NE/P000681/1).

REFERENCES

[1] Marco Avvenuti, Stefano Cresci, Andrea Marchetti, Carlo Meletti, and Maurizio Tesconi. 2014. Ears (earthquake alert and report system) a real time decision support system for earthquake crisis management. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*. 1749–1758.

[2] Bob DuCharme. 2013. *Learning SPARQL: querying and updating with SPARQL 1.1*. O'Reilly Media, Inc.

[3] J Brian Houston and Joshua etc. Hawthorne. 2015. Social media and disasters: a functional framework for social media use in disaster planning, response, and research. *Disasters* 39, 1 (2015), 1–22.

[4] Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*. 57–60.

[5] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* (2014).

[6] Jay Kreps, Neha Narkhede, Jun Rao, et al. 2011. Kafka: A distributed messaging system for log processing. In *Proceedings of the NetDB*, Vol. 11. 1–7.

[7] Jedsada Phengsuwan, Tejal Shah, et al. 2019. Ontology-based discovery of time-series data sources for landslide early warning system. *Computing* (2019), 1–19.

[8] Bin Qian, Jie Su, Zhenyu Wen, et al. 2020. Orchestrating the Development Lifecycle of Machine Learning-based IoT Applications: A Taxonomy and Survey. *ACM Computing Surveys (CSUR)* 53, 4 (2020), 1–47.

[9] Faith E Taylor, Bruce D Malamud, Katy Freeborough, and David Demeritt. 2015. Enriching Great Britain's national landslide database by searching newspaper archives. *Geomorphology* 249 (2015), 52–68.