CrisisDPS: Crisis Data Processing Services

Firoj Alam*
Qatar Computing Research Institute
Hamad Bin Khalifa University
Doha, Qatar
fialam@hbku.edu.qa

Muhammad Imran
Qatar Computing Research Institute
Hamad Bin Khalifa University
Doha, Qatar
mimran@hbku.edu.qa

Ferda Ofli
Qatar Computing Research Institute
Hamad Bin Khalifa University
Doha, Qatar
fofli@hbku.edu.qa

ABSTRACT

Over the last few years, extensive research has been conducted to develop technologies to support humanitarian aid tasks. However, many technologies are still limited as they require both manual and automatic approaches, and more importantly, are not ready to be integrated into the disaster response workflows. To tackle this limitation, we develop automatic data processing services that are freely and publicly available, and made to be simple, efficient, and accessible to non-experts. Our services take textual messages (e.g., tweets, Facebook posts, SMS) as input to determine (i) which *disaster type* the message belongs to, (ii) whether it is *informative* or not, and (iii) what *type of humanitarian information* it conveys. We built our services upon machine learning classifiers that are obtained from large-scale comparative experiments utilizing both classical and deep learning algorithms. Our services outperform state-of-the-art publicly available tools in terms of classification accuracy.

Keywords

Social media, humanitarian data processing, text classification, application programming interfaces, data processing services.

INTRODUCTION

In the last decade, we have witnessed a significant increase in the use of Information and Communications Technologies (ICT), mobile devices, and sensors during time-critical events such as natural or human-induced disasters (Wattegama 2014; Imran, Castillo, Diaz, et al. 2015). This increase in the number of data sources results in producing more and more data each day. These data sources include social media platforms (e.g., Facebook and Twitter), WhatsApp groups, SMS communications, web blogs, news articles, RSS feeds, and many more. The combination of easy-to-use technologies and data sources generate vast amounts of data at high velocity, i.e., thousands of documents appear each second.

Extracting useful information from these sources is highly important for humanitarian organizations, government agencies, and public administrative authorities to make timely decisions and to launch relief efforts during emergency situations (Starbird et al. 2010; Vieweg, Hughes, et al. 2010). The information needs of these stakeholders vary depending on their role, responsibilities, and the situation they are dealing with (Vieweg, Castillo, et al. 2014). Often extracting such information becomes difficult for non-technical domain experts from these high volume data sources. It is reported in the literature that manual analysis of such high-volume and high-velocity social media data streams is impossible (Hiltz et al. 2014; Ludwig et al. 2015).

*Corresponding author

Dealing with such large data sources and extracting useful information from them involve many challenges such as parsing unstructured and brief content, filtering out irrelevant and noisy content, handling information overload, among others. Over the last few years, many Artificial Intelligence (AI) techniques and computational methods have been proposed to process social media data for disaster response and management. These techniques aim to solve various challenges ranging from information filtering, overload, and categorization to summarization (Castillo 2016; Rudra et al. 2018; Imran, Castillo, Diaz, et al. 2015). In addition, there have been publicly available tools that collect social media data and automatically categorize them during disaster events (Imran, Castillo, Lucas, et al. 2014; Burel, Saif, Fernandez, et al. 2017; Meier 2012; Okolloh 2009). The common features of these systems include classifying tweets, grouping them into clusters, visualizing them into timelines, geotagging them onto maps, and visualizing topics, sentiment, and concepts over time. We elaborate further on these systems in the Related Work section.

In this paper, as an attempt to shed more light in the direction motivated by the aforementioned studies, we develop automatic data processing services to assist both technical and non-technical end-users including crisis managers, officials from formal humanitarian organizations, first responders, volunteering organizations, non-governmental organizations (NGOs), and researchers in the crisis informatics community in their humanitarian tasks. Specifically, we provide the following services, which are also the contributions of this paper:

- **Disaster type classification:** This service aims to determine whether a text message (e.g., a tweet or a Facebook post or an SMS) is about one of the six disaster types, namely *earthquake*, *fire*, *flood*, *hurricane*, *bombing*, and *shooting*.
- **Informativeness classification:** This service determines whether a text message is *informative* for humanitarian aid purposes or not. Informative messages contain useful information for response organizations to plan and launch relief efforts.
- **Humanitarian information type classification:** The aim of this service is to classify whether a text message is about one of the following ten humanitarian information types: *affected individual*, *caution and advice*, *displaced and evacuations*, *donation and volunteering*, *infrastructure and utilities damage*, *injured or dead people*, *missing and found people*, *response efforts*, and *sympathy and support*. More details about these humanitarian information types are given in the later sections.
- **Crisis Data Processing Services (CrisisDPS):** To provide the aforementioned three classification services to end-users, we develop an end-to-end system that offers various RESTful Application Program Interfaces (APIs). Specifically, we provide separate APIs for (i) single item processing and (ii) batch processing in addition to (iii) a file processing service.
- **State-of-the-art classification models:** We develop state-of-the-art supervised machine learning models to support the aforementioned three classification services.

To develop the underlying machine learning models, we have conducted extensive experiments using both classical and deep learning algorithms. The classical algorithms include Naïve Bayes (NB) (McCallum, Nigam, et al. 1998), Support Vector Machines (SVM) (Hearst et al. 1998) and Random Forests (RF) (Liaw, Wiener, et al. 2002) whereas the deep learning algorithms include Convolutional Neural Networks (CNN) (LeCun et al. 1989), and Long Short-Term Memory Neural Networks (LSTM) (Hochreiter and Schmidhuber 1997). We have also collected and combined many publicly available datasets such as CrisisLex[1], CrisisNLP[2], among others. While consolidating these datasets, one of the challenges was the mismatch between class labels across datasets. We manually mapped the mismatches and selected the class labels that can effectively serve humanitarian purposes. More details can be found in the Datasets section. We make all of our services publicly available to potential end-users at no cost.

The rest of the paper is organized as follows. In the next section, we present a literature review. Next, we describe our data processing services. After that, we present the details of the datasets that we used to develop our machine learning models. Then, we present our experiments, results and discussion. Finally, we conclude the paper in the last section.

[1] http://crisislex.org/

[2] http://crisisnlp.qcri.org/

RELATED WORK

There has been significant progress in crisis computing research with a particular focus on mainly facilitating humanitarian organizations in their relief efforts and supporting their decision-making processes. One of the key research activities has been analyzing social media content. There are many studies that leverage social media such as Twitter, Facebook and YouTube for curating, analyzing, and summarizing crisis-related information in order to extract situational awareness and actionable information (Zade et al. 2018; Imran, Castillo, Lucas, et al. 2014; Vieweg, Hughes, et al. 2010; Terpstra et al. 2012; Tsou et al. 2017). Among different social media platforms, Twitter has been used extensively due to the fact that information appears timely and can be accessed and processed effectively, which are the required key features for the humanitarian organizations. Apart from research studies, many systems have been developed over time, some of which are publicly and freely available. We briefly cover some of these research studies and tools in this section.

Below we provide a list of available tools and systems that utilize social media data to extract crisis-related information.

- Ushahidi (Meier 2012; Okolloh 2009)[3]: Ushahidi is a platform that started its journey back in 2008 as a free and open-source platform that allowed interested individuals and groups to create live, interactive maps. Initially, it aimed to visualize post-election messages on a map. Its current functionality includes mapping and visualization tools to create real-time, dynamic, and multifaceted crisis maps. It allows for visualizing how a crisis situation is evolving over time. A notable deployment is the 2010 Haiti earthquake (Goolsby 2010). It is currently providing one-month-free and premium supports to its users.
- TweetTracker (Kumar et al. 2011)[4]: TweetTracker is another application aimed at supporting Humanitarian Aid and Disaster Relief (HADR) responders to monitor and analyze social media content during crisis situations. The functionality of this tool includes analyzing location, real-time trending, data filtering, and historical analysis.
- Tweedr[5]: Tweedr is another tool that aimed to extract actionable information during natural disasters from Twitter. The functionality of the tool includes clustering and classification.
- Artificial Intelligence for Disaster Response (AIDR) (Imran, Castillo, Lucas, et al. 2014)[6]: AIDR is an active system that facilitates users (e.g., humanitarian organizations) to collect tweets during disaster events and create classifiers on-the-fly by annotating incoming Twitter data according to the categories of their own choices. It is freely available with minimal usability effort.
- Twitris (Purohit and Sheth 2013)[7]: Twitris application is also targeted for Twitter content analysis. The functionality of this system includes collecting, aggregating, and analyzing tweets to give deeper insights as well as facilitate the research and development process.
- SensePlace2 (MacEachren et al. 2011)[8]: SensePlace2 application is focused more on extracting spatiotemporal information with analytic capabilities to deal with large volumes of tweets.
- Enhanced Messaging for the Emergency Response Sector (EMERSE) (Caragea et al. 2011): This system has been developed to analyze tweets and SMS messages. The major functionality of this system includes Twitter crawler, machine translation system, and classification modules while also supporting users with an iPhone application.
- Emergency Situation Awareness (ESA) (Yin et al. 2012)[9]: ESA is aimed at enhancing situational awareness with respect to crises induced by natural hazards, particularly earthquakes. They focus on spatio-temporal information and functionalities include event detection, text classification, online clustering, and geotagging.
- Twicident (Abel et al. 2012)[10]: Twicident is a framework and system that automatically track and filter information, which is relevant for real-world incidents or crises.

[3] https://www.ushahidi.com/
[4] http://tweettracker.fulton.asu.edu/
[5] https://github.com/dssg/tweedr/
[6] http://aidr.qcri.org/
[7] http://twitris.knoesis.org/
[8] https://www.geovista.psu.edu/SensePlace2/
[9] https://esa.csiro.au/aus/index.html
[10] http://www.wis.ewi.tudelft.nl/twitcident/

- Pipeline for Post-Crisis Twitter Data Acquisition (Kejriwal and Gu 2018): This is an ongoing effort that aims to develop a system that requires minimal supervision in an active learning framework.
- Crisis Event Extraction Service (CREES) (Burel and Alani 2018)[11]: CREES is an open-source web API to automatically classify social media posts during crisis situations. The system is developed for three different tasks, where the classification models are based on deep learning algorithms. One can deploy the system on their own server to use the API functionality. This system is currently integrated into the Ushahidi platform as part of the COMRADES project[12].
- Botometer (Varol et al. 2017)[13]: Botometer is an application aimed to detect accounts that are controlled by automated tools, termed as social bots. Such an application is important to detect and filter fake or manipulated content on social media.

As for the crisis-related tweet classification research, current literature explores both classical algorithms (e.g., Maximum Entropy, Logistic Regression (LR), Random Forest (RF), Naïve Bayes (NB) classifier, Support Vector Machines (SVM), and Conditional Random Fields (CRFs)) and deep learning techniques (e.g., Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-term Memory Neural Networks (LSTM)). In (Imran, S. Elbassuoni, et al. 2013; Imran, S. M. Elbassuoni, et al. 2013), authors report different datasets and NB- and CRF-based classifiers to extract actionable information during disaster events. The feature sets used for developing the classifiers include word unigrams, bigrams, Part-of-Speech (POS) tags, among others. In (Nguyen et al. 2017), authors report a comparative study using classical and deep learning-based algorithms, including SVM, LR, RF, and CNN. Their experimental setting also consists of in-domain and out-domain configurations. Across all experimental configurations, CNN outperforms other algorithms. A similar comparative study has been conducted in (Burel, Saif, Fernandez, et al. 2017; Burel and Alani 2018), where the authors explore different types of features. Their study includes three classification tasks such as (i) related *vs.* not-related, (ii) event types, and (iii) information types. Their findings suggest that SVM and CNN provide very competitive results. The study of Aipe et al. 2018 explores CNN with other tweet-centric (e.g., user mentions, hashtags, and their combinations) features. They report results that outperform state-of-the-art results, which used a similar dataset such as CrisisNLP[14]. However, different than our work, they map class labels of CrisisNLP with a set of their own classes. In another study, Neppalli et al. 2018 compare NB, CNN and Recurrent Neural Network (RNN) with different feature combinations and suggest that CNN outperforms the other models. For a more extensive literature review on this subject, we refer the reader to the study conducted by Imran, Castillo, Diaz, et al. 2015.

SYSTEM AND DATA PROCESSING SERVICES

In this work, we present CrisisDPS, a system to process crisis-related data to support multiple humanitarian tasks. The system is developed to facilitate both technical and non-technical users in the domains of crisis informatics and humanitarian aid working at various levels in an organizational hierarchy, e.g., in an Incident-Command System (ICS)[15], Public Information Officers (PIOs), Safety officers, logistics sections, etc. As most humanitarian organizations already have their information systems, our goal is to provide them with ready-to-use data processing services that can be easily integrated into their existing systems, for example, through a new widget backed by one of the CrisisDPS services.

Figure 1 depicts various functionalities of the CrisisDPS system at a high-level. The system mainly provides three types of data processing services, namely (i) disaster type classification, (ii) informativeness classification, and (iii) humanitarian information type classification. We describe each of these services in detail later in this section. Each classification service offers three types of functionality: (i) single-item processing API, (ii) batch processing API, and (iii) file processing service. Furthermore, we also offer a web demo service to test the system online.

Data Processing Services

The three types of data processing services provided by the CrisisDPS system are described below.

1. Disaster type classification: This service aims to classify a given text message into one of the six disaster types, namely, *earthquake*, *fire*, *flood*, *hurricane*, *bombing*, and *shooting*. We also offer a general category called *none*,

[11] https://evhart.github.io/crees/
[12] https://www.comrades-project.eu/
[13] https://botometer.iuni.iu.edu/#!/api
[14] http://crisisnlp.qcri.org/
[15] https://en.wikipedia.org/wiki/Incident$_C$ommand$_S$ystem

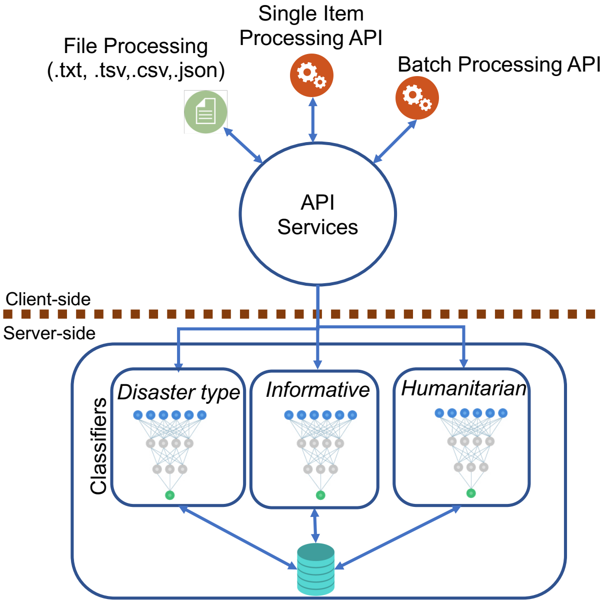


Figure 1. CrisisDPS: Crisis Data Processing Services

which represents everything else that does not belong to any of the six disaster types. The motivation behind this service is to determine whether a message belongs to a particular disaster event or not before dispatching it for further processing. Moreover, this service could be useful for decision-makers in an emergency department who are dealing with multiple disasters at a time helping them to distinguish disaster-specific messages.

2. Informativeness classification: To determine whether a text message contains some useful information for disaster managers or not, we provide the "informativeness" service. Given a text message, the service classifies it either as *informative* or *not informative*. We consider a message as *informative* if it contains some useful information for humanitarian aid; such as disaster-related warnings, reports about injured, dead, or affected people, rescue requests, volunteering or donation offers, reports of damaged houses, roads, etc.

3. Humanitarian information type classification: Given a message is *informative*, which means it contains some useful information for humanitarian aid, the next task is to determine what kind of useful information it conveys. For this purpose, we provide humanitarian information type services. These services correspond to a set of automatic classification services to classify a given text message into one of the ten humanitarian information types described below:

1. *Affected individual:* messages that contain information about affected people due to a disaster event;
2. *Caution and advice:* messages that report warnings, cautions, and give advice to people in the disaster area;
3. *Displaced and evacuations:* messages that report about displaced people or evacuations due to the disaster event;
4. *Donation and volunteering:* messages that request for donations, e.g., food, water, shelter, or offer help or volunteering services;
5. *Infrastructure and utilities damage:* messages that report damages to built structures such as buildings, roads, and bridges;
6. *Injured or dead people:* messages that report injured or dead people due to the disaster event;
7. *Missing and found people:* messages that report missing or found people due to the disaster event;
8. *Requests or needs:* messages that contain requests or urgent needs of affected people;
9. *Response efforts:* messages that report ongoing response efforts by humanitarian organizations, NGOs, and volunteers;
10. *Sympathy and support:* messages that convey thoughts, prayers, sympathy, and support to the victims of the disaster;

11. *Not informative:* messages that do not belong to any of the previous categories or do not contain any useful information.

All three types of classification services are backed by state-of-the-art machine learning classifiers. Specifically, we used supervised machine learning approaches to train the classifiers. More details about these classifiers and the datasets used for training them are given in the next section.

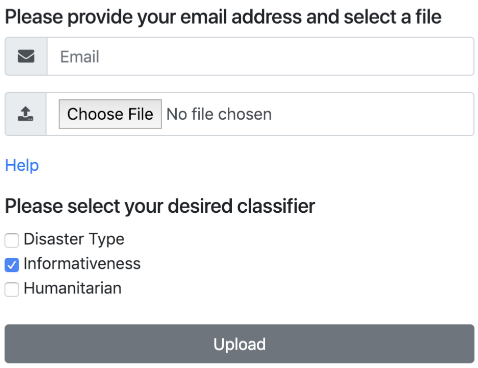
CrisisDPS APIs

The CrisisDPS system provides three types of RESTful APIs for each one of the three classification services. Next, we describe these APIs:

1. Single-item processing API: This programmatic API provides support to classify a single item (e.g., one tweet, or a Facebook message) at a time. Given an item, one can specify one or more classification services to classify it. For example, given a tweet, we want to determine whether it is an *earthquake* or a *hurricane* tweet and whether it contains some useful information or not. Moreover, if it contains useful information, what type of information it contains.

2. Batch processing API: To process more than one items in a batch, the system provides a batch processing programmatic API. For example, given a set of 10,000 tweets, the task is to classify each tweet to determine their informativeness or what humanitarian information they report.

3. File processing service: For non-technical end-users who cannot use our programmatic APIs, we provide a file processing service through a web interface. Figure 2 shows the file uploading interface. Given a file containing a list of messages, tweets, or Facebook posts, this API can classify each message using one or more of the classification services that the end-user selects. Currently, we support four file types: TXT (text files), JSON (JavaScript Object Notation), TSV (Tab-separated values), and CSV (Comma-separated values). Non-technical end-users can easily use the file processing interface to facilitate their humanitarian tasks.



Please provide your email address and select a file

Email

Choose File No file chosen

Help

Please select your desired classifier

□ Disaster Type
☑ Informativeness
□ Humanitarian

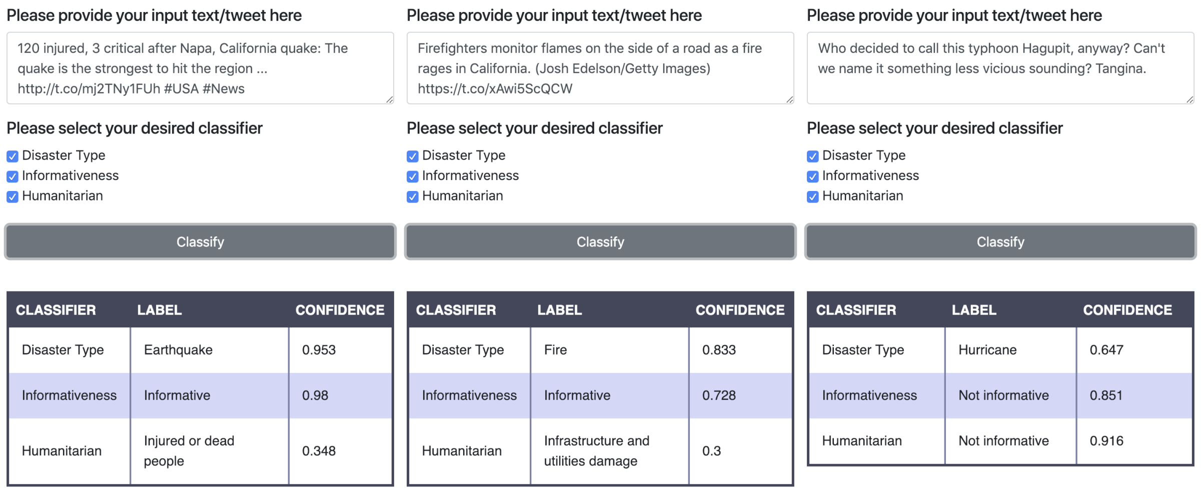
Figure 2. Web interface of the file processing service.

The APIs are implemented in the Java language using the Jersey framework[16], which is an open source and production quality framework for developing RESTful web services in Java. It is combined with Spring[17] and Hibernate[18] frameworks to communicate with databases. The single-item and batch processing APIs only offer programmatic access to use the classification services. However, the file processing API offers a web interface. All the services, APIs, and technical details regarding how to use these APIs with examples are provided on the CrisisDPS website: `http://crisisdps.qcri.org/`. Figure 3 shows a screenshot of the demo page with example tweets and outputs of the selected classification services.

DATASETS

In this work, we use several publicly available datasets including CrisisLex, CrisisNLP, among others. One of the difficulties that arise while combining different datasets is the discrepancies in the class labels. For example, in one dataset, the negative class is labeled as *not-related* while in another one as *not-informative*. Even though these two labels are semantically the same in theory, they are still two different labels in practice. Therefore, for our task

[16] `https://jersey.github.io`. Currently, we use Jersey, in future we might consider to use Spring REST.
[17] `https://spring.io/`
[18] `http://hibernate.org/`



Please provide your input text/tweet here

120 injured, 3 critical after Napa, California quake: The quake is the strongest to hit the region ... http://t.co/mj2TNy1FUh #USA #News

Please select your desired classifier

☑ Disaster Type
☑ Informativeness
☑ Humanitarian

CLASSIFIER	LABEL	CONFIDENCE
Disaster Type	Earthquake	0.953
Informativeness	Informative	0.98
Humanitarian	Injured or dead people	0.348

Please provide your input text/tweet here

Firefighters monitor flames on the side of a road as a fire rages in California. (Josh Edelson/Getty Images) https://t.co/xAwi5ScQCW

Please select your desired classifier

☑ Disaster Type
☑ Informativeness
☑ Humanitarian

CLASSIFIER	LABEL	CONFIDENCE
Disaster Type	Fire	0.833
Informativeness	Informative	0.728
Humanitarian	Infrastructure and utilities damage	0.3

Please provide your input text/tweet here

Who decided to call this typhoon Hagupit, anyway? Can't we name it something less vicious sounding? Tangina.

Please select your desired classifier

☑ Disaster Type
☑ Informativeness
☑ Humanitarian

CLASSIFIER	LABEL	CONFIDENCE
Disaster Type	Hurricane	0.647
Informativeness	Not informative	0.851
Humanitarian	Not informative	0.916

Figure 3. Example usages of the demo page.

we mapped the semantically similar categories to have a unique class label. In addition, we have also filtered out instances with low frequencies and less relevance to the humanitarian tasks at hand. For example, we removed the instances with class labels such as *animal management*, *non-government*, *traditional media*, *terrorism-related*, and *terrorism-related information*. We also removed non-English tweets.

For this study, we aim to develop three different classifiers to serve three different classification tasks: (i) disaster type classification, (ii) informativeness classification, and (iii) humanitarian information type classification. To prepare the data for the informativeness task, we selected all the tweets with *informative/related* class labels and the tweets with humanitarian categories as *informative* and considered the rest as *not-informative*. For the humanitarian information type classification task, we selected data that are labeled with humanitarian categories, which resulted in ten categories as listed in Table 3. In this dataset we have also included instances with *not-informative* label, however, we have only selected a random sample from the whole set of *not-informative* instances. For the disaster type classification task, we selected tweets with the following six disaster types: *bombing*, *earthquake*, *fire*, *flood*, *hurricane*, and *shooting*. In addition, we have also included *none* category as the "background" class. That is, the tweets that do not belong to any of the mentioned six disaster types are considered as belonging to the *none* category.

For the sake of completeness, we provide below brief descriptions of the datasets consolidated for the experiments.

- AIDR[19] is a publicly available system that has been developed to support humanitarian organizations and research community. The system has a functionality that allows users to annotate data while collecting tweets. Over time, AIDR has collected data from many events both labeled and unlabeled. We only selected manually labeled data that are relevant to this study.
- CrisisLex[20] comprises two different datasets, i.e., CrisisLexT26 and CrisisLexT6, that have been used and reported in the literature for crisis informatics research. The CrisisLexT26 is one of the largest datasets consisting of 26 different crisis events that took place in 2012 and 2013 (Olteanu et al. 2014). This dataset was prepared to explore two different crisis dimensions. First dimension is the disaster type (natural *vs.* human-induced), their sub-type (e.g. meteorological, hydrological, etc.), temporal characteristics (instantaneous *vs.* progressive), and geographic spread (focalized *vs.* diffused). The second dimension is the content type in which several type of categories and sub-categories were identified, including informativeness (informative *vs.* not-informative), information type (six different subcategories), and source of information (i.e., an eyewitness, government, NGOs, Business, Media, and Outsiders). The second dataset, CrisisLexT6, consists of six disasters that took place between October 2012 and July 2013 in USA, Canada, and Australia. Crisis keywords and locations have been used during the data collection process (Olteanu et al. 2014). Approximately 10*K* tweets have been annotated using Figure Eight[21] (also known as CrowdFlower) from each crisis event. Annotation of this dataset includes *related vs. not-related*.

[19] http://aidr.qcri.org/
[20] https://crisislex.org
[21] https://www.figure-eight.com/

- CrisisMMD[22] is a multimodal dataset consisting of tweets and associated images (Alam, Ofli, et al. 2018). Tweets have been collected from seven natural disasters that took place in 2017. The annotations include three tasks: (i) *informative vs. not-informative*, (ii) humanitarian categories (eight classes), and (iii) damage severity (three classes). The third annotation task, i.e., damage severity, was applied only on images. The annotation for text and images was run independently for each event using Figure Eight.
- CrisisNLP[23] consists of ~50K tweets, which have been collected from 19 different disasters that took place between 2013 and 2015. The dataset was annotated in two steps. The first set of annotations was obtained using Stand-By-Task-Force (SBTF)[24] volunteers. The second set of annotations was curated using Figure Eight.
- Disaster Response Data[25] contain 30K tweets collected during disasters such as an earthquake in Haiti, 2010; an earthquake in Chile, 2010; floods in Pakistan, 2010; Hurricane Sandy in USA, 2012, and news articles. The annotations include 36 different categories.
- Disasters on Social Media[26] dataset consists of 10K tweets collected and annotated with labels *related* vs. *not-related* to the disasters.
- SWDM[27] consists of two data collections. The Joplin collection contains tweets collected during the tornado that struck Joplin, Missouri on May 22, 2011. The Sandy collection contains tweets collected during Hurricane Sandy, that hit Northeastern US on Oct 29, 2012. The Joplin dataset consists of 4,400 labeled instances and the Sandy dataset consists of 2,000 labeled instances (Imran, S. Elbassuoni, et al. 2013).

After consolidating data from all these collections, we prepared three datasets for three different classification tasks. Tables 2, 3 and 1 present the class distributions of our datasets for disaster type, informativeness, and humanitarian information type, respectively. We split each dataset into three subsets using 70%, 10% and 20% as training, development and test, respectively. The training set is used to learn the model, the development (dev) set is used for parameter tuning, and the test set is used for the model evaluation.

Table 1. Data split for disaster type classification.

Classes	**Train (70%)**	**Dev (10%)**	**Test (20%)**	**Total**	**Class Dist.**
Bombing	378	53	109	540	0.007
Earthquake	9,088	1,285	2,611	12,984	0.162
Fire	2,185	309	628	3,122	0.039
Flood	10,156	1,436	2,917	14,509	0.181
Hurricane	23,491	3,322	6,746	33,559	0.418
Shooting	527	74	153	754	0.009
None	10,306	1,457	2,961	14,724	0.184
Total	56,131	7,936	16,125	80,192	

Table 2. Data split for informativeness classification.

Classes	**Train (70%)**	**Dev (10%)**	**Test (20%)**	**Total**	**Class Dist.**
Informative	131,028	18,531	37,624	187,183	0.639
Not-informative	74,104	10,480	21,280	105,864	0.361
Total	205,132	29,011	58,904	293,047	

EXPERIMENTS, RESULTS, AND DISCUSSION

In this section, we describe the details of our extensive experiments, report our results, and provide a discussion.

[22] http://crisisnlp.qcri.org/
[23] http://crisisnlp.qcri.org/
[24] www.standbytaskforce.org
[25] https://www.figure-eight.com/dataset/combined-disaster-response-data/
[26] https://data.world/crowdflower/disasters-on-social-media
[27] http://crisisnlp.qcri.org/

Table 3. Data split for humanitarian information type classification.

Classes	Train (70%)	Dev (10%)	Test (20%)	Total	Class Dist.
Affected individual	2,536	358	729	3,623	0.073
Caution and advice	2,644	374	760	3,778	0.076
Displaced and evacuations	519	73	150	742	0.015
Donation and volunteering	5,677	802	1,631	8,110	0.163
Infrastructure and utilities damage	4,619	653	1,327	6,599	0.133
Injured or dead people	2,771	392	796	3,959	0.080
Missing and found people	460	65	133	658	0.013
Requests or needs	5,088	719	1,462	7,269	0.146
Response efforts	1,857	263	534	2,654	0.053
Sympathy and support	4,438	627	1,275	6,340	0.127
Not-informative	4,200	594	1,206	6,000	0.121
Total	34,809	4,920	10,003	49,732	

Preprocessing

The tweet texts are noisy, usually consisting of many symbols, emoticons, and invisible characters. Therefore, we preprocessed them to use in classification experiments. The preprocessing part includes removal of stop words, non-ASCII characters, punctuations (replaced with whitespace), numbers, URLs, and hashtags.

Classification Experiments

For this study, we have conducted classification experiments using both classical and deep learning algorithms. As for the classical models, we used the three most popular machine learning algorithms, i.e., (i) Naïve Bayes (NB), (ii) Random Forest (RF), and (iii) Support Vector Machines (SVM). The usefulness of these algorithms have been reported in several studies in crisis computing literature (Burel, Saif, and Alani 2017; Neppalli et al. 2018; Imran, Mitra, et al. 2016). As for the deep learning algorithms, we used a Convolutional Neural Network (CNN), which has also been reported as successful in many studies for disaster tweets classification (Nguyen et al. 2017; Burel and Alani 2018).

- The NB algorithm (McCallum, Nigam, et al. 1998) is a simple probabilistic method that calculates a set of probabilities by counting the frequency and combinations of values in a given dataset. The algorithm uses Bayes theorem and stands on the assumption that all attributes are independent given the value of the class variable, which is rarely true in real applications, but it still performs well and learns rapidly in many supervised tasks (Neppalli et al. 2018; Imran, Mitra, et al. 2016).
- The SVM algorithm (Platt 1998) is based on the Structural Risk Minimization principle from computational learning theory. The algorithm is established as universal learners and well-known for its ability to learn independently of the dimensionality of the feature space. These properties of the algorithm make it one of the most popular supervised classification methods.
- Unlike the previous two classical algorithms, the RF algorithm (Liaw, Wiener, et al. 2002) is an ensemble learning algorithm, comprising multiple decision trees where each tree contributes to the classification decision with a single vote. A final class label is assigned based on majority voting. RF reduces variances in the classification by randomizing features and training instances.
- In Figure 4, we present the architecture of our CNN model. The input to the network is a tweet $\mathbf{t} = (w_1, \ldots, w_n)$ containing words each coming from a finite vocabulary $\mathcal{V}$ that we obtained from our training dataset. The first layer of the network maps each of these words into a distributed representation $\mathbb{R}^d$ by looking up a shared embedding matrix $E \in \mathbb{R}^{|\mathcal{V}|\times d}$. We initialized E using a publicly available pre-trained embedding model designed using disaster related tweets[28] (Alam, Joty, et al. 2018). The output of the look-up layer is a matrix $X \in \mathbb{R}^{n\times d}$, which is passed through a number of convolutional and pooling layers to learn higher-level feature representations. A convolution operation applies a *filter* $\mathbf{u} \in \mathbb{R}^{k.d}$ to a window of k vectors to produce a new feature $h_t = f(\mathbf{u}.X_{t:t+k-1})$, where $X_{t:t+k-1}$ is the concatenation of k look-up vectors, and f is a nonlinear

[28] http://crisisnlp.qcri.org/data/lrec2016/crisisNLP_word2vec_model_v1.2.zip

activation. We used rectified linear units (ReLU) as a nonlinear activation function. We apply this filter to each possible k-length windows in X to generate a *feature map*, $\mathbf{h}^j = [h_1, \ldots, h_{n+k-1}]$. This process is repeated N times with N different filters to get N different feature maps. We then apply a *max-pooling* operation, $\mathbf{m} = [\mu_p(\mathbf{h}^1), \cdots, \mu_p(\mathbf{h}^N)]$, where $\mu_p(\mathbf{h}^j)$ refers to the max operation applied to each window of p features in the feature map $\mathbf{h}^i$. Intuitively, the convolution operation composes local features into higher-level representations in the feature maps, and max-pooling extracts the most important aspects of each feature map while reducing the output dimensionality. Since each convolution-pooling operation is performed independently, the features extracted become invariant in order. In order to incorporate order information between the pooled features, we use a fully-connected (dense) layer $\mathbf{z} = f(V\mathbf{m})$, where V is the weight matrix. From which we have another dense layer $\mathbf{z}_c = f(V_c\mathbf{z})$ where V_c is the corresponding weight matrix. The activations $\mathbf{z}_c$ are used for classification. Formally, the classification layer defines a Softmax

$$p(y = k|\mathbf{t}, \theta) = \frac{\exp\left(W_k^T \mathbf{z}_c\right)}{\sum_{k'} \exp\left(W_{k'}^T \mathbf{z}_c\right)} \tag{1}$$

where W_k are the class weights and $\theta = \{V, V_c, W\}$ defines the relevant parameters.

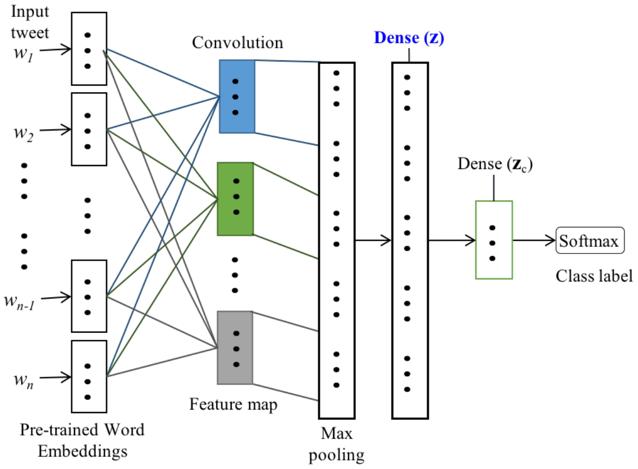


Figure 4. The architecture of the CNN.

To train the classifiers using these classical algorithms, we converted the preprocessed tweets into bag-of-ngrams vectors weighted with logarithmic term frequencies (tf) multiplied with inverse document frequencies (idf), as shown in Equation 2.

$$tf \times idf = log(1 + f_{ij}) \times log\left(\frac{\text{number of tweets}}{\text{number of tweets that include word i}}\right) \tag{2}$$

where f_{ij} is the frequency for word i in tweet j. To take advantage of the contextual benefits of n-grams, we extracted tri-gram features. Because this resulted in a large dictionary and we filtered out lower frequency features by preserving $40K$ most frequent n-grams. For the experiments with SVM and its linear kernel we tune the c parameter, and with RF we tune the number of trees on the development set.

We trained the CNN models using the Adadelta optimizer (Zeiler 2012). The learning rate was set to 0.01 when optimizing on the classification loss. The maximum number of epochs was set to 1,000, and dropout (Srivastava et al. 2014) rate of 0.02 was used to avoid overfitting. We did *early stopping* based on the accuracy on the development set with a patience of 50. We used 100, 150, 200, and 250 filters with the corresponding window size of 2, 3, 4, and 5 respectively. We used a pooling length of 2. We did not tune any hyperparameter (e.g., the size of hidden layers, filter size, dropout rate) in the experimental setting. We also applied batch normalization due to its success reported in the literature (Ioffe and Szegedy 2015).

Table 4 summarizes the results obtained by different classification algorithms described above. We provide weighted average precision (P), recall (R) and F1-measure (F1) for each task. The rationale behind choosing the weighted metric is that it takes into account the class imbalance problem. The performance of the NB model is lower than other models. Whereas RF and SVM model provides competitive results across different tasks. For the informativeness task RF outperforms SVM, whereas for humanitarian and disaster type tasks SVM outperforms RF. We obtain higher results consistently across all the tasks using CNN.

Table 4. Classification performance comparison of all the classifiers on three different tasks.

	Disaster Type				Informativeness				Humanitarian			
	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
NB	0.73	0.82	0.73	0.71	0.85	0.85	0.85	0.85	0.65	0.77	0.65	0.69
SVM	0.93	0.93	0.93	**0.93**	0.90	0.90	0.90	0.90	0.78	0.78	0.78	**0.78**
RF	0.91	0.92	0.91	0.91	0.93	0.93	0.93	**0.93**	0.76	0.77	0.76	0.76
CNN	0.93	0.93	0.93	**0.93**	0.93	0.93	0.93	**0.93**	0.78	0.78	0.78	**0.78**

Table 5. Mapping of the class labels between CREES and CrisisDPS.

Disaster Type	
CREES	**CrisisDPS**
Bombings	*Bombing*
Collapse	*None*
Crash	*None*
Derailment	*None*
Earthquake	*Earthquake*
Explosion	*None*
Fire	*Fire*
Floods	*Flood*
Haze	*None*
Meteorite	*None*
None	*None*
Shootings	*Shooting*
Typhoon	*Hurricane*
Wildfire	*Fire*

Informativeness	
CREES	**CrisisDPS**
Related	*Informative*
Not-related	*Not-informative*

Humanitarian Information Type	
CREES	**CrisisDPS**
Affected individuals	*Affected individual*
Caution and advice	*Caution and advice*
Donations and volunteering	*Donation and volunteering*
Infrastructure and utilities	*Infrastructure and utilities damage*
Not applicable	*Not-informative*
Not labeled	X
Other useful information	X
Sympathy and support	*Sympathy and support*

X indicates the CREES class labels that were not considered in this work.

Baseline Comparisons

To compute the baseline results we used the publicly available deep learning models[29] reported in (Burel and Alani 2018). The CREES API consists of three models for three different tasks such as (i) related *vs.* not-related, (ii) event type (11 class labels), and (iii) information type (i.e., humanitarian categories). To evaluate the performance of the CREES models and have the baseline results for our study, we used our test set and mapped the class labels as shown in Table 5.

Table 6. Comparative results for disaster type classification.

	CREES			CrisisDPS		
Classes	**P**	**R**	**F1**	**P**	**R**	**F1**
Bombing	0.54	0.83	0.66	0.94	0.93	**0.94**
Earthquake	0.96	0.66	0.78	0.92	0.94	**0.93**
Fire	0.85	0.88	0.86	0.95	0.93	**0.94**
Flood	0.35	0.94	0.51	0.94	0.93	**0.93**
Hurricane	0.98	0.1	0.18	0.93	0.96	**0.95**
Shooting	0.97	0.96	0.97	1.00	0.99	**0.99**
None	0.45	0.73	0.55	0.94	0.87	**0.91**
W/Avg	0.76	0.50	0.45	0.93	0.93	**0.93**

In Tables 6, 7, and 8, we provide class-wise precision (P), recall (R) and F1-measure (F1) as well as their weighted average both for the baseline CREES models and the proposed CrisisDPS models. We observed that there is a significant performance difference across all classification tasks compared to the results reported in the study of CREES (Burel and Alani 2018). For the humanitarian categories, the CREES model is designed using eight class

[29] https://github.com/evhart/crees

labels, whereas our model is designed with eleven class labels. Hence, in Table 8, we do not have results for some class labels using the CREES model. In Table 8, the weighted average results for CrisisDPS are computed using only the subset of categories available in CREES for a fair comparison.

Table 7. Comparative results for informativeness classification.

	CREES			CrisisDPS		
Classes	**P**	**R**	**F1**	**P**	**R**	**F1**
Informative	0.88	0.86	0.87	0.95	0.95	0.95
Not-informative	0.76	0.79	0.78	0.91	0.90	0.91
W/Avg	0.84	0.84	**0.84**	0.93	0.93	**0.93**

Table 8. Comparative results for humanitarian information type classification.

	CREES			CrisisDPS		
Classes	**P**	**R**	**F1**	**P**	**R**	**F1**
Affected individual	0.28	0.70	0.40	0.72	0.69	**0.70**
Caution and advice	0.51	0.58	0.54	0.72	0.74	**0.73**
Displaced and evacuations	-	-	-	0.65	0.54	0.59
Donation and volunteering	0.54	0.70	0.61	0.77	0.80	**0.79**
Infrastructure and utilities damage	0.78	0.32	0.45	0.77	0.75	**0.76**
Injured or dead people	-	-	-	0.86	0.86	0.86
Missing and found people	-	-	-	0.67	0.44	0.53
Requests or needs	-	-	-	0.86	0.89	0.88
Response efforts	-	-	-	0.75	0.77	0.76
Sympathy and support	0.47	0.84	0.60	0.80	0.79	**0.80**
Not-informative	0.87	0.02	0.04	0.74	0.74	**0.74**
W/Avg	0.41	0.36	0.31	0.76*	0.76*	**0.76***

* computed using only the subset of categories available in CREES.

Discussion and Future Directions

We have provided comparative results between classical and deep learning models. We have also compared the performance of our best model with publicly available models. Even though we have only provided an end-to-end comparison with the CREES models in this study, our models also outperform some other existing works such as (Nguyen et al. 2017; Neppalli et al. 2018).

To estimate the load of our system, we plan to have an extensive study in the future and plan to deploy the system accordingly. We also plan to make the system more scalable. Other future work includes integrating image processing modules such as finding an informative image, identifying their humanitarian information type, and assessing the damage content.

Even though we developed the classification models using only social media data such as tweets, the CrisisDPS system can easily be extended by integrating other models. The benefit of CrisisDPS is that it provides services with RESTful APIs by taking the burden of running them on a local server. Such facilities can be helpful for humanitarian organizations and crisis computing research community to develop their own applications and systems using our services. Our goal is to make the CrisisDPS freely available, however, an authentication system will be implemented to control the rate limit (e.g., data processing limit per day per user) and to prevent potential abuse of the system so that every user can get equal benefit from the system.

CONCLUSIONS

In this study, we introduced CrisisDPS, a system for automatic data processing services, which comprises state-of-the-art information classification models that enable humanitarian organizations and crisis informatics research community to develop their own applications and systems. Currently, CrisisDPS provide services that can facilitate three important humanitarian tasks: (i) determine which disaster type a text message belongs to, (ii) determine whether the message is informative for disaster response, and (iii) determine whether it manifests about any humanitarian information type. Such information is highly relevant for humanitarian organizations if processed timely and effectively. The classification models that we developed for these three tasks outperform existing publicly available models. We aim to provide as much flexibility as possible to the end-users who are willing to use our system. For example, our current implementation supports single item and batch processing through different APIs, file processing for different formats, and an online demo page for quick testing purposes. We plan to improve our services based on a thorough user study in the future. We also plan to make the system more scalable and extend it with automatic image processing capabilities.

REFERENCES

Abel, F., Hauff, C., Houben, G.-J., Stronkman, R., and Tao, K. (2012). "Semantics+ filtering+ search= twitcident. exploring information in social web streams". In: *Proc. of the 23rd ACM HT conference*. ACM, pp. 285–294.

Aipe, A., Mukuntha, N., Ekbal, A., and Kurohashi, S. (2018). "Deep Learning Approach towards Multi-label Classification of Crisis Related Tweets". In: *Proc. of the 15th ISCRAM, 2018*.

Alam, F., Joty, S. R., and Imran, M. (2018). "Graph Based Semi-supervised Learning with Convolution Neural Networks to Classify Crisis Related Tweets". In: *Proc. of the 12th ICWSM, 2018*. AAAI press.

Alam, F., Ofli, F., and Imran, M. (2018). "CrisisMMD: Multimodal twitter datasets from natural disasters". English. In: *Proc. of the 12th ICWSM, 2018*. AAAI press, pp. 465–473.

Burel, G. and Alani, H. (2018). "Crisis Event Extraction Service (CREES)-Automatic Detection and Classification of Crisis-related Content on Social Media". In: *Proc. of the 15th ISCRAM, 2018*.

Burel, G., Saif, H., and Alani, H. (2017). "Semantic wide and deep learning for detecting crisis-information categories on social media". In: *International Semantic Web Conference*. Springer, pp. 138–155.

Burel, G., Saif, H., Fernandez, M., and Alani, H. (2017). "On Semantics and Deep Learning for Event Detection in Crisis Situations". In: *Workshop on Semantic Deep Learning (SemDeep), at ESWC 2017*.

Caragea, C., McNeese, N., Jaiswal, A., Traylor, G., Kim, H.-W., Mitra, P., Wu, D., Tapia, A. H., Giles, L., Jansen, B. J., et al. (2011). "Classifying text messages for the haiti earthquake". In: *Proc. of the 8th ISCRAM, 2011*.

Castillo, C. (2016). *Big Crisis Data*. Cambridge, UK: Cambridge University Press.

Goolsby, R. (2010). "Social media as crisis platform: The future of community maps/crisis maps". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 1.1, p. 7.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. (1998). "Support vector machines". In: *IEEE Intelligent Systems and their applications* 13.4, pp. 18–28.

Hiltz, S. R., Kushma, J., and Plotnick, L. (2014). "Use of Social Media by U.S. Public Sector Emergency Managers: Barriers and Wish Lists". In: University Park, Pennsylvania, USA.

Hochreiter, S. and Schmidhuber, J. (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Imran, M., Castillo, C., Diaz, F., and Vieweg, S. (2015). "Processing social media messages in mass emergency: A survey". In: *ACM Computing Surveys* 47.4, p. 67.

Imran, M., Castillo, C., Lucas, J., Meier, P., and Vieweg, S. (2014). "AIDR: Artificial intelligence for disaster response". In: *Proc. of the ACM Conference on WWW*, pp. 159–162.

Imran, M., Elbassuoni, S. M., Castillo, C., Diaz, F., and Meier, P. (2013). "Extracting information nuggets from disaster-related messages in social media". In: *Proc. of the 12th ISCRAM*.

Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., and Meier, P. (2013). "Practical extraction of disaster-relevant information from social media". In: *Proc. of the 22nd WWW*. ACM, pp. 1021–1024.

Imran, M., Mitra, P., and Castillo, C. (2016). "Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages". In: *Proc. of the LREC, 2016*. Paris, France: ELRA.

Ioffe, S. and Szegedy, C. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167*.

Kejriwal, M. and Gu, Y. (2018). "A Pipeline for Post-Crisis Twitter Data Acquisition". In: *arXiv preprint arXiv:1801.05881*.

Kumar, S., Barbier, G., Abbasi, M. A., and Liu, H. (2011). "TweetTracker: An Analysis Tool for Humanitarian and Disaster Relief." In: *Proc. of the 5th ICWSM, 2011*. AAAI press.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4, pp. 541–551.

Liaw, A., Wiener, M., et al. (2002). "Classification and regression by Random Forest". In: *R news* 2.3, pp. 18–22.

Ludwig, T., Reuter, C., Siebigteroth, T., and Pipek, V. (2015). "CrowdMonitor: Mobile crowd sensing for assessing physical and digital activities of citizens during emergencies". In: *Proc. of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, pp. 4083–4092.

MacEachren, A. M., Jaiswal, A., Robinson, A. C., Pezanowski, S., Savelyev, A., Mitra, P., Zhang, X., and Blanford, J. (2011). "Senseplace2: Geotwitter analytics support for situational awareness". In: *Proc. of the IEEE conference on VAST, 2011*. IEEE, pp. 181–190.

McCallum, A., Nigam, K., et al. (1998). "A comparison of event models for naive bayes text classification". In: *AAAI-98 workshop on learning for text categorization*. Vol. 752. 1. Citeseer, pp. 41–48.

Meier, P. (2012). "Crisis mapping in action: How open source software and global volunteer networks are changing the world, one map at a time". In: *Journal of Map & Geography Libraries* 8.2, pp. 89–100.

Neppalli, V. K., Caragea, C., and Caragea, D. (2018). "Deep Neural Networks versus Naïve Bayes Classifiers for Identifying Informative Tweets during Disasters". In: *Proc. of the 15th ISCRAM, 2018*.

Nguyen, D. T., Al-Mannai, K., Joty, S. R., Sajjad, H., Imran, M., and Mitra, P. (2017). "Robust Classification of Crisis-Related Data on Social Networks Using Convolutional Neural Networks." In: *Proc. of the 11th ICWSM, 2017*. AAAI press, pp. 632–635.

Okolloh, O. (2009). "Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information". In: *Participatory learning and action* 59.1, pp. 65–70.

Olteanu, A., Castillo, C., Diaz, F., and Vieweg, S. (2014). "CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises." In: *Proc. of the 8th ICWSM, 2014*. AAAI press.

Platt, J. (1998). *Sequential minimal optimization: A fast algorithm for training support vector machines*. Tech. rep. Redmond, USA.: Microsoft.

Purohit, H. and Sheth, A. P. (2013). "Twitris v3: From Citizen Sensing to Analysis, Coordination and Action." In: *Proc. of the 7th ICWSM, 2013*. AAAI press.

Rudra, K., Goyal, P., Ganguly, N., Mitra, P., and Imran, M. (2018). "Identifying sub-events and summarizing disaster-related information from microblogs". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, pp. 265–274.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). "Dropout: a simple way to prevent neural networks from overfitting." In: *Journal of MLR* 15.1, pp. 1929–1958.

Starbird, K., Palen, L., Hughes, A. L., and Vieweg, S. (2010). "Chatter on the red: what hazards threat reveals about the social life of microblogged information". In: *ACM Conference on Computer Supported Cooperative Work*, pp. 241–250.

Terpstra, T., De Vries, A., Stronkman, R., and Paradies, G. (2012). *Towards a realtime Twitter analysis during crises for operational crisis management*. British Columbia, Canada: Simon Fraser University Burnaby.

Tsou, M.-H., Jung, C.-T., Allen, C., Yang, J.-A., Han, S. Y., Spitzberg, B. H., and Dozier, J. (2017). "Building a Real-Time Geo-Targeted Event Observation (Geo) Viewer for Disaster Management and Situation Awareness". In: *Proc. of the International Cartographic Conference*. Springer, pp. 85–98.

Varol, O., Ferrara, E., Davis, C., Menczer, F., and Flammini, A. (2017). "Online Human-Bot Interactions: Detection, Estimation, and Characterization". In: *Proc. of the 11th ICWSM, 2018*. AAAI press.

Vieweg, S., Castillo, C., and Imran, M. (2014). "Integrating social media communications into the rapid assessment of sudden onset disasters". In: *International Conference on Social Informatics*. Springer, pp. 444–461.

Vieweg, S., Hughes, A. L., Starbird, K., and Palen, L. (2010). "Microblogging during two natural hazards events: what twitter may contribute to situational awareness". In: *SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 1079–1088.

Wattegama, C. (2014). "ICT for disaster management". In: *UNDP-APDIP*.

Yin, J., Lampert, A., Cameron, M., Robinson, B., and Power, R. (2012). "Using social media to enhance emergency situation awareness". In: *IEEE Intelligent Systems* 27.6, pp. 52–59.

Zade, H., Shah, K., Rangarajan, V., Kshirsagar, P., Imran, M., and Starbird, K. (2018). "From Situational Awareness to Actionability: Towards Improving the Utility of Social Media Data for Crisis Response". In: *Proceedings of the ACM on Human-Computer Interaction* 2.CSCW, p. 195.

Zeiler, M. D. (2012). "ADADELTA: an adaptive learning rate method". In: *arXiv preprint arXiv:1212.5701*.