Towards Automated Assessment of Crowdsourced Crisis Reporting for Enhanced Crisis Awareness and Response

by

Dylan R. Lewis

B.S. Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 2020

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author .....
Department of Electrical Engineering and Computer Science
May 13, 2022

Certified by .....
Una-May O'Reilly
Principal Research Scientist
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Towards Automated Assessment of Crowdsourced Crisis Reporting for Enhanced Crisis Awareness and Response

by

Dylan R. Lewis

Submitted to the Department of Electrical Engineering and Computer Science
on May 13, 2022, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The availability of information during a climate crisis event is critical for crisis managers to assess and respond to crisis impact. During crisis events, affected residents post real-time crisis updates on platforms such as RiskMap and Twitter. These updates provide localized information, which has the potential to enhance crisis awareness and response. However, with limited resources, crisis managers may endure information overload from the inundation of these updates. Prior work has demonstrated the potential of machine learning (ML) methodologies to mitigate this problem. We have identified limitations in the prior work including the lack of involvement of crisis managers in the development and evaluation of a ML methodology.

To address these limitations, we propose a novel framework and ML methodology which investigate the efficacy of various ML methods in enhancing crisis awareness and response beyond model performance metrics. This framework aims to iteratively embed the information needs and priorities of crisis managers during crisis into the design of the ML methodology. We cooperated with crisis managers in Fukuchiyama City (FC), a city in Japan which is susceptible to flood events, and analyzed crowdsourced crisis image and text data from past FC flood events. We devised the Flood Presence image classification task, constructed Train/Dev/Test splits, and annotated images from FC. We report a weighted F1 score of 92.1% on the test split and 82.5% on the FC images. Using the results of our image analysis ML methodology and the insights we gained from crisis managers, we iterated on the design of our text analysis ML methodology. This led to the creation of the Human Risk text classification task which is tailored to a subset of the identified information needs of the crisis managers. To align with the priorities of crisis managers for this task, we determined the model evaluation metric to be the F2 score. We report an F2 score of 92.8% on an FC crisis text test dataset, which is a significant improvement over the baseline score of 43.4%.

Thesis Supervisor: Una-May O'Reilly
Title: Principal Research Scientist

Acknowledgments

The work conducted in this thesis would not have been possible without the support of some truly remarkable people.

I am deeply grateful for the invaluable guidance, mentorship, support, and time my thesis advisor, Una-May O'Reilly, has generously provided me throughout my MEng. I learned a lot from her expertise, and this thesis would be lacking in technical rigor and quality without her insights. She is an extraordinary mentor.

I am immensely thankful to Miho Mazereeuw for giving me the opportunity and support to investigate an important, challenging, and interdisciplinary problem at the Urban Risk Lab. It has been an honor to work with such passionate teammates, Saeko Baird, Aditya Barve, and Mayank Ojha, who have taught me so much. This thesis would be lacking in the implications it has for the broader crisis informatics community without the focus-group research with crisis managers done by Saeko Baird. Relatedly, we thank Toyoaki Nishida, Richard Serino, Yasuaki Yokoyama, and our partners in Fukuchiyama, whose participation in focus-group research was paramount for the implications derived from this thesis. We acknowledge Kyoko Murayama who assisted in our image annotation efforts. We thank Evan Owens & Abe Quintero who provided insight and feedback into the software we developed in this thesis. I am incredibly thankful to have mentored and worked with the following undergraduates during my research: Ben Gao, Clarise Han, Ibuki Iwasaki, Katie Pelton, Sabrina Queipo, Sandra Tang, Liane Xu, and Will Yang.

I am grateful for my friends and their boundless support, namely: Keis Bejgo, Sara Sime, Fernando Juarez, Jaime Osuna, Amelia Trainer, Alex Canepa, Nick Bonaker, Will Torous, Karina Hinojosa, Madison Hill, Julia Fiksinski, Katharina Gschwind, and Kat Adams. These friendships inspire me everyday.

To my family, I am thankful for your love and support that inspired me to apply to MIT and to see it through to the end of my masters degree. I love you.

This work was supported by a grant from Google.org and the Tides Foundation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The availability of accurate and real-time crisis information during a climate crisis event is critical for crisis awareness and response. Residents seek this information to avoid risk in areas that are severely impacted by the ongoing crisis event [2]. Crisis managers that operate Emergency Operations Centers (EOCs) use this information to analyze the unfolding situation and make decisions to quickly respond to affected areas [3]. To gain awareness about an unfolding crisis, residents will often utilize their most immediate information streams: phone calls or texts with family and friends, government and non-government organization (NGO) communications, radio, television, and various internet sources including social media [4]. With the increasing use of smartphones and social media globally in recent years, social media has emerged as an important source of crowdsourced, real-time, localized, crisis information, where residents ask for help and post crisis updates about their area. This paradigm of residents crowdsourcing crisis updates is sometimes referred to as "civic sensing" or "people as sensors" [5, 6].

There are numerous examples from past crisis events where residents have utilized social media to stay informed and to disperse information during crisis providing information such as flood level, fire line, geolocation information, requests for help, and other crisis-related information [2, 3, 4, 7, 8]. The increased use of social media at all stages of crises has highlighted how social media can operate as a medium through which affected communities, both residents and crisis managers alike, can contribute

and gain awareness of an unfolding climate crisis in real-time. State-of-the-art crisis information systems have typically leveraged remote sensing and satellite imagery. The deployment and utilization of these technologies is costly and time-consuming for extracting useful information about the crisis event. Furthermore, data collected from satellite imagery can be noisy and uninformative due to cloudiness such as during hurricane events [9].

More recently, however, crisis information systems have been developed to leverage civic sensing to enhance situational awareness and response [3]. One such system is the RiskMap web platform developed by the Urban Risk Lab (URL) at MIT that is described in detail in the Background section. RiskMap solicits crowdsourced crisis data from residents during a climate crisis event by leveraging the social media platforms they already use, presenting the crowdsourced, geotagged crisis information on a map for affected communities to see in real-time.

1.1 Motivation

Utilizing crisis data provided through social media or mediums such as RiskMap has the potential to enhance awareness of the unfolding crisis, however in practice, it does present its own major challenges for crisis managers and EOCs. The influx of crisis reports that come into crowdsourced crisis information systems during a crisis can result in information overload for crisis managers; there can be too many reports to assess manually and not enough resources to quickly assess and respond, resulting in incomplete awareness of the situation and delayed response [4]. The inundation of crowdsourced crisis reports to the crisis information system diminishes the informative utility the reports have for enhancing localized situational awareness for EOCs.

This diminished utility highlights the potential of augmenting crowdsourced crisis information systems like RiskMap with scalable, automated report assessment. This augmentation has the potential to yield timely and accurate extraction of useful, localized information for gaining situational awareness. Machine Learning (ML) methods provide the means to scalably automate the assessment of the unwieldy amount of

crowdsourced crisis reports that are received during a crisis event. This is best seen by previous applications of state-of-the-art ML techniques to crisis posts on social media, which we discuss in the Background and Related Work sections.

With this thesis, we aim to expand upon prior work in the area of using ML to mitigate information overload in crisis information systems and address some of the limitations we have identified through our understanding of the literature. Namely, we have determined that there has been large emphasis placed on the performance evaluation of ML algorithms on various, often similar crisis classification tasks on crowdsourced crisis data [1, 5, 9, 10, 11, 12, 13, 14, 15, 16]. While performance is certainly important for determining model efficacy, we argue that is just one piece in understanding the efficacy that ML methods and Artificial Intelligence (AI) systems have in mitigating information overload. There are other aspects of assembling an AI system to assist crisis managers that are not typically discussed in the literature such as assessing the informative utility a specified prediction task has for crisis managers, documenting the reliability of the human-annotated data used to train and evaluate models published in the literature, and incorporating insights and feedback from crisis managers into the design of the system and the models contained within it. We have also understood that there is sparse literature involving natural language processing (NLP) applied to crowdsourced crisis text in languages other than English. Lastly, we note that to the best of our knowledge, there exists a gap in the prior work in regards to cooperating with crisis managers when designing and constructing AI-augmented crisis information systems. As we describe in later sections of this thesis, this engagement is crucial in understanding the information needs and priorities crisis managers have, which should inform the design of the system. This engagement is imperative as it has influence on the information models should aim to capture with their outputs, the metrics used to optimize those models based on the priorities of crisis managers, and how researchers should design and test the interfaces and channels through which crisis managers and models interact in order to yield enhanced crisis awareness and response.

We aim to address these limitations and gaps in an effort to broaden the discussion

within the crisis informatics community. Thus, we investigate various ML techniques that can be applied to mitigate information overload for crisis managers during crisis events while also assessing if those techniques can satisfy the information needs and priorities of crisis managers through qualitative and quantitative evaluation. To meet these aims, we present a novel framework within the crisis informatics community consisting of the following components:

- Classification Task Creation
- Data Annotation Procedure
- Interannotator Agreement/Data Reliability Analysis
- Model Development and Evaluation; Per-Class Performance Analysis
- Qualitative Analysis through Workshops with Crisis Managers

This framework situates *Model Development and Evaluation*, which is commonplace in prior work, as part of a larger, contextualized analysis. On that note, we expand on the notion of *Model Development and Evaluation* beyond what is typically seen in the prior work.

An important aim of this framework is the development of models with performance metrics that are inline with the priorities of crisis managers. Similarly, we aim to build models to predict labels that should have informative utility for crisis managers during a crisis event. Thus, we develop new classification tasks using labels provided to us directly from crisis managers in addition to labels present in open-source datasets. Additionally, the metrics we used to evaluate models are derived either from metrics reported in the literature or, more notably, metrics determined from insights provided by crisis managers directly. We note that besides taking into consideration the priorities of crisis managers as they pertain to the task when deciding on an appropriate performance evaluation metric, we also consider the impacts of class imbalance. In addition to the aggregate metrics typically reported in the literature, we report the per-class performance metrics and confusion matrices for

models for more granular insights into model performance. Lastly, we determine the performance of baseline models we can compare against the models we develop to assess the degree to which our developed models outperform the baseline.

A novel aim of this framework is transparency in the procurement and quality of human-annotated data that is used to train and evaluate ML models. To that end, we conducted a data annotation effort with members of the Urban Risk Lab and provide subsequent analysis of the labeled data prior to using it for ML purposes.

Finally, we note that this framework is designed with the intent of building systems which are iteratively developed, that is, the system's design and aims, such as the specific classification tasks performed by the models, iteratively incorporate the insights and feedback of crisis managers. Therefore, we sought to cooperate with crisis managers.

To investigate the framework we have developed, we focus our study on flood crisis events noting that flood events account for half of all deaths from natural hazards [17, 18] and due to climate change, the intensity and frequency of flooding is expected to increase globally [19]. Japan experiences flooding every year largely from typhoons, which are defined by a period of rainfall of strong intensity for long duration [20]. In this work, we partnered with EOCs in Fukuchiyama, a city located in the Kyoto Prefecture of Japan. Fukuchiyama is located on the Yura River, which makes it particularly vulnerable to flooding. In recent years, Fukuchiyama has endured disastrous flooding crises resulting from typhoons and heavy rainfalls [21]. This partnership enabled us to both evaluate the aims of our research in a context that is becoming increasingly affected by flood crises every year and directly engage with crisis managers in that context.

Our partners provided us with image and text data which were part of crowdsourced crisis reports from past flood events in Fukuchiyama,[1] we thus present our framework through two ML modules which we refer to as the Image Analysis Module and the Text Analysis Module.

[1]By the request of the data providers, we are unable to open-source the datasets containing this data used in our analysis

1.2 Contributions

By pursuing the framework above through the Image Analysis and Text Analysis ML modules and using it to evaluate our ML methodologies in the flood-susceptible context of Fukuchiyama with our partners, we summarize the main contributions of this thesis below.

1.2.1 Novel Framework for Information Overload Mitigation of Crowdsourced Crisis Data

Our main contribution is a novel framework that unifies a variety of machine learning techniques, analysis, and evaluation on images and text present in crowdsourced crisis data, which aims to reduce the information overload of crisis managers, specifically during flood crisis events. This approach leverages classical machine learning models and deep learning models aimed to provide accurate and efficient classifications on individual flood reports to mitigate information overload of crisis managers. This framework addresses the limitations in the prior work discussed above. In this framework, we embed various utilities for conducting analysis such as computing properties and statistics of annotated datasets and model performance. Finally, based on the analysis conducted in this study and the insights gained from the feedback provided in workshops we held with crisis managers, we expand on the implications informed from the results we have attained for the future of this framework and the field of crisis informatics more broadly. Although this framework is our main contribution, we discuss other notable contributions to the field of crisis informatics that came out of this study.

1.2.2 Flood Presence Task Creation, Labeled Image Dataset, and Performance Benchmark

Since we focused on flood crises, we defined the image prediction task of Flood Presence classification. The Flood Presence task is the binary classification task of de-

termining whether or not there is presence of flood in an image. We construct a labeled dataset for the task consisting of $\sim 23.6k$ images by combining various open-source datasets which were labeled with labels useful for this task, although they were originally developed for other adjacent tasks. We contribute this dataset for further research in crisis informatics. In addition, we provide Train/Dev/Test splits and an associated benchmark performance on the test split using a state-of-the-art Convolutional Neural Network (CNN) model, EfficientNet-B1 [22], discussed in the Image Analysis Module section.

1.2.3 Data Annotation Procedure and Analysis of Interannotator Agreement

We developed a procedure for annotating images in order to label the unlabeled image data provided to us by our partners in Fukuchiyama. This procedure included creating an annotation guide to assist annotators in their labeling. This guide included the name of each task, the names of the mutually-exclusive classes associated with each task, and an associated description and example image for each class. We then had annotators from the Urban Risk Lab independently annotate the images using this guide.

After the annotation effort was completed, we were able analyze the interannotator agreement between the annotators and construct ground-truth labels for these images. We computed interannotator reliability statistics to get a sense of how reproducible the annotation procedure was for each task as well as to have transparency of the data quality prior to using it for ML purposes. Finally, we created ground-truth datasets using these labels for the Fukuchiyama images to use in evaluating the image classification ML models we developed.

1.2.4 Classification and Clustering of Crowdsourced Japanese Crisis Text

As noted in the Motivation section, research in crisis informatics on crowdsourced crisis data focuses mostly on English, and thus research on crowdsourced Japanese crisis text is sparse. The Text Analysis Module developed in this work focused exclusively on Japanese text data. We explored various numerical representations of the Japanese crisis text reports provided by our partners and developed a pipeline for preprocessing the raw text and producing the numerical representations. Additionally, our partners provided labels along with the text reports, so we experimented with classifying the text. Lastly, we explored the text data using unsupervised learning, specifically, we employed clustering methods to help inform development of classification tasks in future work.

Pipeline for Japanese Crisis Text Preprocessing, Tokenization, and Featurization

In order to use the text reports as input to the various ML models employed in this work, we represent the raw text string of each report as a numerical vector, or feature vector. This process of transforming the text data into a feature vector is called featurization. Depending on the featurization we choose to use for a text report, we may first preprocess the text. This preprocessing included various steps including tokenization, stopword removal, and lemmatization, which we performed using open-source software (i.e. tokenizer and lemmatizer) and publicly available data (i.e. stopwords list) for the Japanese language.

We provide a pipeline for preprocessing and performing the following featurizations on Japanese text:

- n-gram Bag-of-Words (BOW)
- n-gram Term Frequency-Inverse Document Frequency (TF-IDF)

- Pretrained Japanese Masked Language Modeling (MLM) BERT Model with Classification (CLS) Pooling Embedding

The resultant feature vectors representing the text enabled us to use them as input to ML models. Thus, we can then employ classification and clustering techniques on the text data.

Human Risk Task Creation and Performance Metric Determination

We devised a new text classification task, Human Risk classification. The human risk text classification task determines whether or not a crisis text report indicates if there are people in need of rescue from a crisis. This includes people being unable to evacuate due to physical disability (such as unable to use stairs), surrounding conditions (such as being trapped in a submerged car), and/or being in need of life-saving emergency medical care. This classification task was unique among the classification tasks presented in this work because it was devised using labels that came with the text reports given to us by our crisis management partners. Relatedly, we determine the metric to use in model performance evaluation using the insights we gained from our partners.

Exploratory Analysis of Japanese Crisis Text using Unsupervised Learning

With the intention of finding cohesive groupings within the Japanese crisis text corpus, which can inform the development of future text classification tasks, we devise a pipeline for featurizing Japanese crisis text, reducing the high-dimensional text feature vector to 2 dimensions, and clustering the data. We evaluate this pipeline both quantitatively and qualitatively, experimenting with various text featurizations including unigram TF-IDF features and pretrained Japanese MLM BERT with CLS Pooling embeddings mentioned above, t-Distributed Stochastic Neighbor Embedding (t-SNE) [23] and Principle Component Analysis (PCA) [24] for dimensionality reduction, and finally K-means [25] and K-medoids [26] for the algorithm which clusters the data.

After we determine the optimal combination of text embedding, dimensionality reduction technique, and clustering algorithm, we create brief summaries of each cluster using the unigrams with highest TF-IDF score for each cluster and the closest reports (by euclidean distance) within each cluster to the cluster center to help in the determination of a label for each cluster. Lastly, a member of the Urban Risk Lab at MIT who is fluent in Japanese used these summaries to determine an interpretable label to accompany each cluster found. We thus provide various labels which can be used for classification experiments and analysis in a future work.

1.2.5 Open-source Python Packages

To ensure the reuse of the analysis conducted in this work, we release two open-source Python packages: One for the Image Analysis Module and the other for the Text Analysis Module. These packages include utilities for training, testing, and prediction using the models presented in this work, computing statistics for interannotator agreement, and computing metrics for model performance. We note that there are utilities for plotting such as methods for producing the plots shown throughout this thesis.

In addition to these packages, we release an open-source repository containing Jupyter notebooks [27], relevant documents (e.g. the annotation guide mentioned above), and other code required to reproduce the experiments and reuse the analysis conducted in this work.

The code for this thesis and links to associated packages can be found here.

1.2.6 Quantitative and Qualitative Evaluation in Japanese Flood Crisis Context

Prior work has typically evaluated ML methods using quantitative measures, mainly classification performance metrics, e.g. accuracy, precision, recall, F1, and AUROC (Area Under the Receiver Operating Characteristic Curve) and their macro and micro variants. However, with the framework we present in this thesis, we aimed to expand

the evaluation of the efficacy ML models have in reducing information overload to not only include similar quantitative measures mentioned above, but also qualitative evaluation derived from engaging with our crisis management partners. Beyond having good performance, we hoped to use the qualitative evaluation used in this study to gain a broader understanding of the efficacy a model can provide crisis managers in mitigating information overload and gaining situational awareness.

We held image annotation workshops with various crisis managers and aimed to understand what type of information they seek to gain from a crowdsourced image during a flood crisis event. With their insights, we began to understand how our models can be refined or improved, or how new models can be created in order to better serve the information needs of crisis managers more effectively, such as by tailoring the labels and their associated meanings to the information needs of crisis managers suggested from their annotations. Additionally, we gained more insight into the appropriate metrics to use when evaluating models based on the priorities of crisis managers as it relates to the task. From these workshops, we share key lessons that can influence the design of this framework and AI-augmented crisis information systems of the future. In fact, within this work, we used the lessons learned from the image analysis workshops to assist us in determining the performance metric to use when developing models for the human risk text classification task. This exercise exhibited the principle of iterative development our framework intends to promote.

This thesis is organized by first detailing the background context for which this study is based upon, describing the RiskMap crowdsourced crisis reporting platform and the limitations RiskMap and systems like it have in contributing to the information overload of crisis managers. We follow this by detailing an AI-powered system that aims to tackle the information overload present in RiskMap. This system is called REACT [28]. Additionally, we discuss related work that uses machine learning methods to address the problem of information overload of crowdsourced crisis reporting, detailing how this work differs from and builds on some of the ideas presented in the prior work. We then detail our ML methodology for approaching the mitigation

of information overload and corresponding qualitative and quantitative evaluation, which follow from the framework described above. We discuss the results of the evaluation of our methodology using data from past flood events in Fukuchiyama as well as feedback we received from our crisis management partners from Fukuchiyama and other contexts from manual image annotation workshops. Finally, we conclude with the insights we gained from conducting this study, the implications this work has for the crisis informatics community, and future directions for this framework and the development of AI systems for enhancing crisis awareness and response through automated assessment of crowdsourced crisis reporting more broadly.

Chapter 2

Background

Gaining understanding of an unfolding crisis event by crisis relief and rescue organizations is an arduous endeavor which can take hours, days, or even weeks to gain enough situational awareness of the event to make decisions and engage in relief efforts. The lack of data and situational understanding in the early stages of a crisis event is often referred to as the cold-start problem [3]. In addition to the data collected from on-the-ground crisis responders during and after a crisis event, social media posts by residents during crisis events can provide quick, useful, and localized information pertaining to how they or their community have been impacted by the crisis [2, 3, 4].

Although this crowdsourced data is abundant [3, 4], it is typically not centralized nor presented in a way that is immediately helpful for other residents and crisis managers. Additionally, only a small fraction of crisis social media posts have associated location data [29] eliminating the potential for extracting crisis information that is localized. Furthermore, even if a social media post has a geotag associated with it, the geotag may not correspond to the actual location referenced in the post. A crisis information system which has aimed to centralize, accurately geotag, and display crowdsourced crisis reports collected from various social media platforms is RiskMap.

2.1 RiskMap Overview

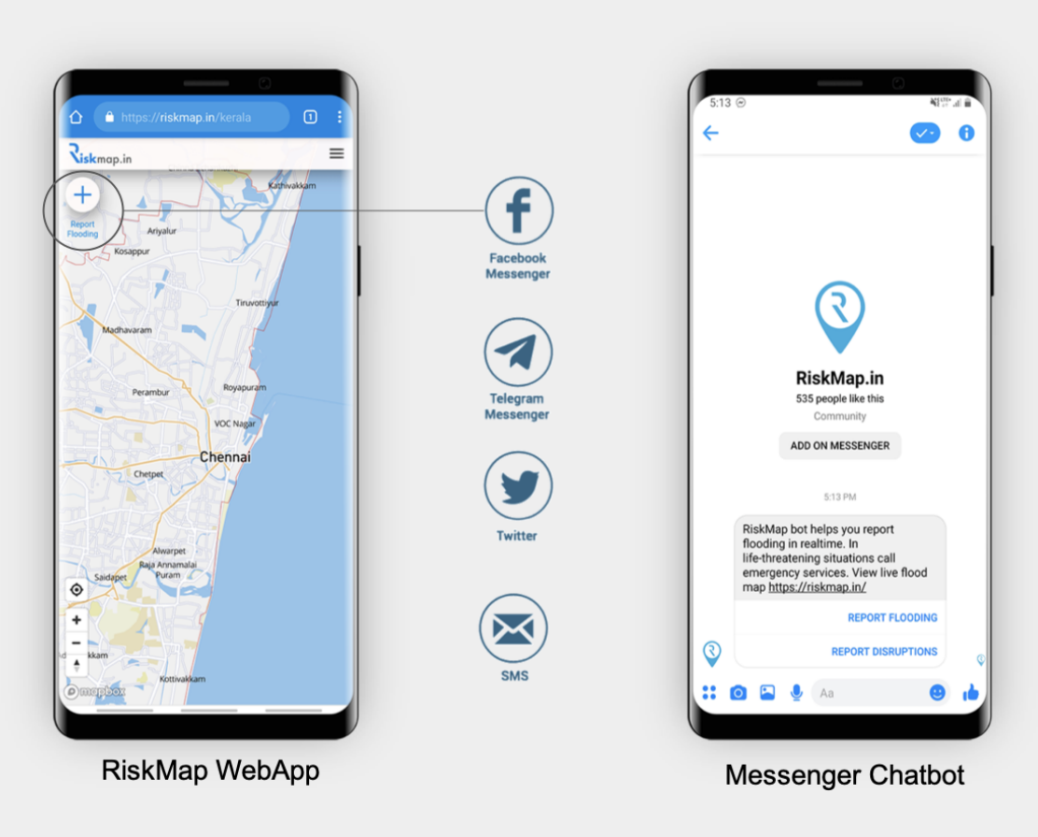
RiskMap[1] is a free, open-source web platform developed by the Urban Risk Lab (URL) at MIT. The RiskMap platform is a comprehensive situational awareness and disaster management application based on civic sensing. The platform was created in 2016 to connect residents, who have the best localized information at all stages of a crisis, to each other and crisis managers through crowdsourced crisis reporting. RiskMap taps into social media platforms residents already use (Twitter, Facebook, LINE) as well as SMS, making it very easy for residents to post informative crisis updates in a matter of seconds, without any installation. RiskMap allows for quick and centralized aggregation of crisis reports across multiple social media platforms to enhance awareness for residents and crisis managers by placing the reports on a map in real-time [29]. We discuss the process by which a user submits a report in RiskMap.

2.1.1 Crisis Reporting

When a user visits RiskMap, they choose a social media platform to serve as a channel through which they fill out a crisis report. Once a social media platform is selected, RiskMap provides the user with a deeplink. This deeplink navigates the user to their selected social media platform, straight into a direct message chat with a RiskMap chatbot. The chatbot prompts the user with options for reporting different types of incidences such as flooding, typhoon damage, and road closure. This interaction is shown in Figure 2-1. When a user selects a report type, a card deck which collects various data modalities (e.g. image, text, estimated flood height, etc.) related to the selected crisis type is generated for the user to create a crisis report. The reporting flow for a flood event is shown with the user interface in Figure 2-2.

In addition to users directly visiting the RiskMap website to initiate the crisis reporting flow, RiskMap utilizes various bots on social media which filter social media streams in order find users who may be reporting about the relevant crisis event

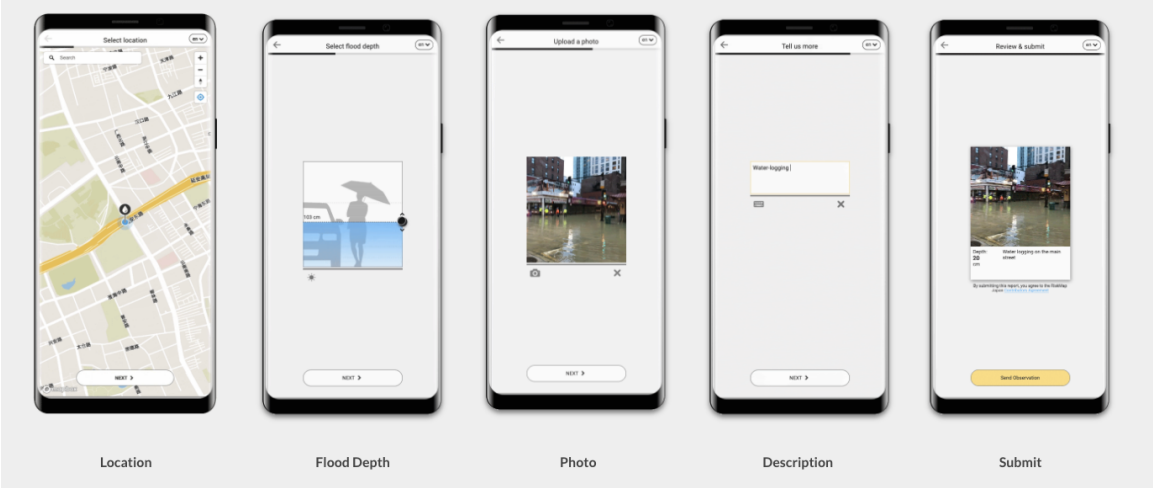
[1] https://riskmap.mit.edu/



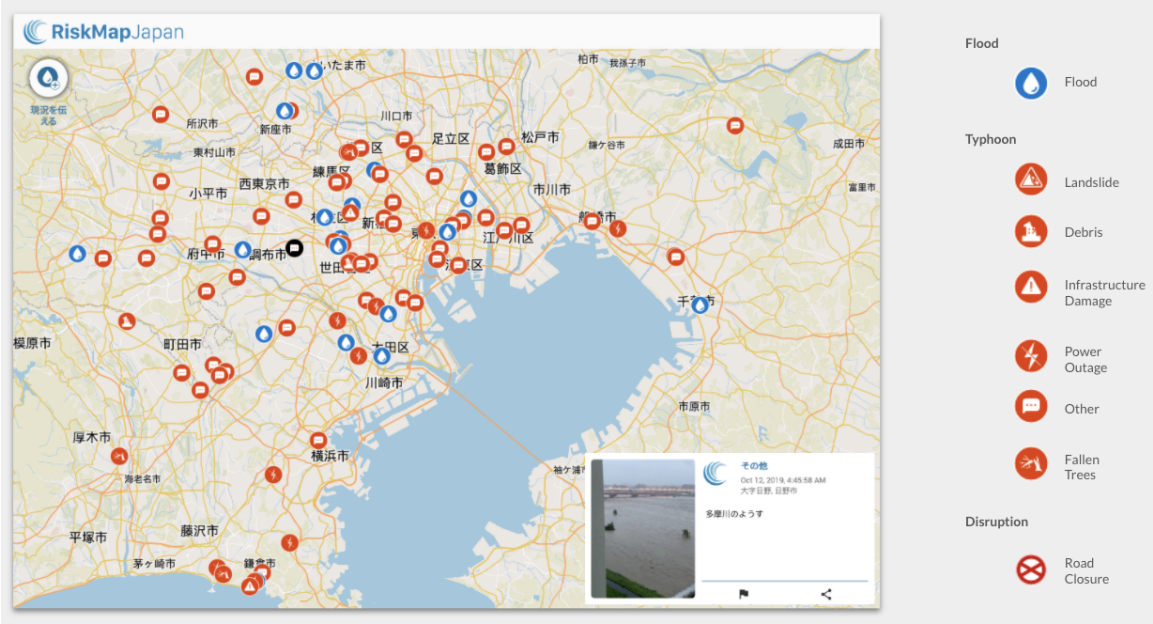
▲ Figure 2-1: RiskMap User Interface and RiskMap Chatbot in Facebook Messenger for reporting a flooding incident in Chennai. Graphic by URL MIT

happening in the affected area. When the bots find such users, the bots then prompt these users to submit a crisis report so that their update can be shared with other residents and crisis managers through RiskMap.

From the flooding card deck in Figure 2-2, we see multiple data modalities are collected on the incident from the user including location coordinates, estimated flood height (in cm), an image, and a text description. Finally, when a user submits the report, the report gets displayed on a map of the area along with all other reports made during the crisis. A map of Tokyo, Japan during Typhoon Hagibis with actual reports of flooding, typhoon damage, and road closures is shown Figure 2-3.



▲ Figure 2-2: RiskMap Crisis Report Flow for Flooding Incident. Graphic by URL MIT



▲ Figure 2-3: RiskMap displaying reports during Typhoon Hagibis in 2019 on a map. Graphic by URL MIT

2.1.2 Crisis Management Dashboard for EOCs

Beyond the public map displaying reports, RiskMap also provides a dashboard for EOCs to monitor reports as they come into RiskMap. This dashboard is referred to as the Risk Evaluation Matrix (REM). The REM permits crisis managers to share alerts

and important information for residents on the map. The REM helps to visualize and assess individual crisis reports and analyze other time critical data.

2.1.3 Information Overload from Crisis Reports

Extracting detailed information useful for attaining situational awareness from an individual crisis report typically requires manual assessment. With a large influx of crisis reports during a crisis event and limited resources in an EOC, this process of manual assessment can lead to information overload, reducing the ability of EOCs to respond in a timely manner during a crisis event, when making decisions quickly is critical.

The RiskMap Evaluation and Coordination Terminal (REACT) is a system developed by the Urban Risk Lab that aims to mitigate information overload of crisis managers using the REM during a crisis event by leveraging various machine learning models to automatically classify whether or not a report is indicative of heavy flooding.

2.2 REACT: The RiskMap Evaluation and Coordination Terminal

REACT is a system developed for use within RiskMap that classifies flooding reports as described in Section 2.1.1 as being indicative of "Heavy Flooding" or "No Heavy Flooding" using an ensemble learning method. REACT is modular and flexible by providing options to users in the creation of data loaders for different data input sources, several featurizations of report data, and various machine learning algorithms used to train and evaluate the classification of RiskMap reports. Various system configurations were trained and tested using RiskMap report data from 2017 flood events in Chennai, India (356 reports) and Jakarta, Indonesia (2229 reports) [28].

2.2.1 Configurable Flood Report Featurizations

For the featurization of text data, REACT provides BOW based on unigrams and BOW based on bigrams feature vectors [30]. The author notes that n-gram BOW is particularly beneficial for RiskMap text data since n-gram BOW is language-agnostic. The author also notes that the text of RiskMap reports is typically short (140 characters or less). We note that we determined the Japanese text reports given to us by our partners in Fukuchiyama to also be short text, about 100 characters or less.

As part of the various image featurizations offered by REACT, the system leverages the highly scalable machine learning as a service offered by cloud computing service providers such as Amazon Web Services (AWS) and Google Cloud Platform (GCP) to provide labels of objects and scenes of report images, such as "Nature", "Human", "Flood", etc. along with an associated confidence score of the predicted label. For the respective cloud service providers, this specific service is referred to as AWS Image Rekognition and GCP Vision AI. Using these labels and confidence scores, REACT provides a Visual Bag-of-Words (VBOW) [31] feature vector of the report images and benefits from the improvements made over time to the models used by the cloud providers to provide these labels and confidence scores.

2.2.2 Ensemble Learning for Multimodal Flood Reports

Finally, the system offers various machine learning algorithms such as Support Vector Machine (SVM), Perceptron, and the pretrained CNN, ResNet18. Using ensemble learning, the author synthesized a single multimodal feature embedding using text, image, and estimated flood height modalities of a flood report. To do this, the author first trained SVM models on the text and image modalities separately and computed the signed distance of the input feature vector for a specific modality to the respective learned separator for that modality. The signed distances are concatenated to form a new embedding of the report, which also concatenates the estimated flood height as a raw scalar feature. If any of the data modalities mentioned are missing from a report, the respective entries in the embedding corresponding to that particular modality

are set to zero. This embedding is passed as input to a neural network, which then classifies the overall report as "Heavy Flooding" or "No Heavy Flooding".

Due to the small size of the datasets, the ensemble model is evaluated by calculating the mean accuracy across stratified 10-fold Cross Validation (CV) on the Chennai and Jakarta datasets. The model achieves mean accuracy scores of 80% and 74% on the Chennai and Jakarta datasets, respectively.

This thesis builds upon some key ideas presented in REACT. We note that REACT analyzed non-English crisis text, namely text in Bahasa Indonesia, thus we were inspired to investigate similar approaches such as using the language-agnostic BOW based on unigrams and BOW based on bigrams featurizations in our analysis of Japanese crisis text. Additionally, our Japanese text dataset was similar in size (716 text reports) as those investigated in REACT, so we also incorporate K-fold cross validation and variations of it as part of our text classification experiments. In the next section, we dive deeper into related work. Having discussed REACT and other prior related work in detail, we then expand on how this study differs.

Chapter 3

Related Work

Early crowdsourced crisis information systems such as the Ushahidi system [3] and CrisisTracker [32] received a large influx of reports during crisis events and relied on the manual assessment of volunteers to accurately classify them all. Though these systems provided accurate classifications of crisis reports, they struggled to scale to the influx of crisis reports even when they had their highest participation of volunteers [33]. The information overload apparent in these systems as well as RiskMap demonstrates the need to investigate methods of scalable and accurate automated crowdsourced crisis report assessment. Automating the assessment of crisis reporting has motivated researchers in crisis informatics to explore and evaluate various machine learning methods to improve the situational awareness of the ongoing crisis for affected residents and crisis managers and better enable the allocation of resources during crisis.

Machine learning techniques have had increasing success in a variety of real-world applications. As we have mentioned, there has also been increased social media use at all stages of a crisis in recent decades. These changes have led researchers to studying the applications of machine learning techniques for addressing the information overload that comes with the big data produced on social media during crisis.

Most of the prior research that investigates applying ML methods on crowdsourced crisis data uses data from Twitter. Similar to RiskMap reports, tweets can consist of image and text data, so the research conducted on crisis tweets utilizes techniques

from the fields of computer vision and NLP.

3.1 Image Analysis on Social Media Crisis Images

It is typically infeasible to have enough labeled data to train and test a supervised learning model during the early hours, days, or potentially weeks of a crisis, however this would be the most ideal setting as the training and testing data would come from the same underlying distribution. This is the cold-start problem mentioned in the Background section.

With enough training data, Deep Learning (DL) architectures, e.g. Fully-Connected Neural Networks (FCNNs), CNNs, Recurrent Neural Networks (RNNs), transformers, etc. have been shown to outperform their classical non-neural counterparts, e.g. Linear Regression, Logistic Regression, Support Vector Machine, Decision Tree, Random Forest, K-Nearest Neighbors, etc. in generalization performance, i.e. performance on unseen data. However, deep learning models tend to have significantly more trainable parameters to estimate than their classical counterparts, thus require significantly more training data in order to outperform. We note that one major advantage deep learning methods have over their classical counterparts is that they eliminate the need for manual feature extraction [5, 34, 35], which proves useful in a variety of settings.

CNNs have quickly emerged as the dominant model for object recognition and image classification tasks, however training these models requires larger datasets than what is typically available in our domain. A popular and successful technique used for computer vision tasks when the dataset for the prediction task is small is Transfer Learning.

3.1.1 Transfer Learning

Deep neural networks require being trained on large datasets representative of the domain of the task to perform well. Training these models from scratch with little data leads to overfitting on the training data and poor generalization to unseen data.

Transfer learning utilizing CNNs pretrained on large image datasets (e.g. ImageNet [36]) with domain-specific finetuning on a smaller dataset has been shown to significantly improve performance on image recognition and classification tasks in domains with small datasets [37], which has historically been the case in the crisis informatics domain [34].

Finetuning

One approach to transfer learning is finetuning. In finetuning, we first train a model on a larger dataset for a general prediction task, which likely has different outputs than the desired task. The output layer is adjusted to have the appropriate output for the desired task and the weights connecting the previous hidden layer to the output layer are randomly initialized. In this paradigm, all previously learned weights remain trainable, that is, they are unfrozen and can change when trained on new data. Typically any training beyond the initial training on the larger dataset uses a smaller learning rate than the one used for the original training as to not dramatically perturb the weights that were learned from training the model on the larger dataset. Lastly, in this paradigm, the model is trained on the smaller, more specific dataset, "finetuning" the model to perform the desired task [38].

Feature Extraction

An alternative transfer learning approach is feature extraction. In feature extraction, after training the model on a larger dataset, the learned weights are frozen, i.e. the weights cannot be adjusted in future training, although additional randomly initialized hidden layers can be added on top as well as an output layer adjusted for the desired task. Then the model is trained on the smaller, more specific dataset, adjusting the weights in the newly added layers of the network for learning the desired task.

These approaches are effective because the early layers of the model (closer to the input) learn basic features of an image (e.g. lines, edges, shapes, etc.), which are transferable across many problem domains, whereas the features learned in the

deeper layers of the network are specific to the desired problem domain [38].

The transfer learning technique forms the basis behind many of the state-of-the-art image classification models used in crowdsourced crisis image classification tasks presented in the literature and the image classification models used in this work.

3.1.2 Classification of Crisis Images

Using images from social media and Google images of infrastructure damage from earthquake, hurricane, and typhoon events, a finetuned VGG16 CNN model was able to reasonably classify images into severity categories of "Little-to-No", "Mild", and "Severe" damage. The finetuned VGG16 model was shown to outperform the VBOW model that had been used in prior work, achieving macro-averaged accuracy, precision, recall, and F1 scores of 0.87, 0.86, 0.87, 0.86, respectively, on a test set for data from the 2016 Ecuador Earthquake event [34]. Furthermore, the authors found that a combination of both in-event and out-of-event, or cross-event crisis images used as a training set for fine-tuning a VGG16 model performed better on the damage classification task for a specific event than with either in-event or out-of-event data alone. This suggests a potential benefit to using the active learning paradigm [39] in crisis information systems with AI augmentation (an example of this is discussed in Section 3.3.3) as it would enable the ability to retrain on newly labeled batches of in-crisis data, potentially improving performance of the model during the crisis event, while also making use of out-of-event data as part of the training data.

More recently, to address the lack of large image datasets in the crisis informatics community, researchers in [1] compiled large, open-source datasets from other, smaller crisis datasets of crowdsourced crisis images for various image classification tasks including Damage Severity ($\sim$ 34.9k images), Humanitarian Categories ($\sim$ 16.8k images), Informativeness ($\sim$ 59.7k images), and Disaster Type ($\sim$ 17.5k images). Additionally, to address the lack of performance benchmarks in the community, and thus the inability to directly compare results, the authors created randomized, non-overlapping Train/Dev/Test splits and reported performance benchmarks on the test splits for each of the tasks. While exploring the finetuning of several pretrained

deep CNN architectures for performing these tasks, the researchers provided performance benchmarks, by weighted F1 score, for each of the test splits. These performance benchmarks enable the crisis informatics community the ability to directly compare results in future research based on the test split performance. We utilize these Train/Dev/Test splits in this work and make use of the most performant and efficient deep learning model determined in [1], the EfficientNet-B1 model [22], for use within our Image Analysis Module. We discuss the implications [1] has for our work in greater detail in the Image Analysis Module chapter.

A similar endeavor to the one described above was undertaken for crisis text data and crisis text classification models by the researchers in [13].

3.2 Natural Language Processing on Social Media Crisis Text

During climate crisis events, there are a large number of social media posts created using hashtags and keywords regarding the crisis event. Many of these posts can be very unrelated to the event and result in noise, which can impair the performance of NLP techniques. When researchers use these posts for conducting text analysis, they employ machine learning models to filter the crowdsourced data for posts that are actually relevant to the crisis event [2, 16, 40]. This filter is often referred to as the "informativeness" or "relevancy" classification task, which is often applied to both image and text data. We note that this classifier is likely more useful in the context of social media data, which can contain posts that cover a wide array of topics outside of the crisis event, as opposed to data collected in systems like RiskMap, which by design have data that is more consistently related to a crisis event. Due to the availability of labeled data for this task as mentioned in Section 3.1.2, we elect to investigate this task in our work by applying it to Fukuchiyama image data.

Various natural language processing techniques have been applied to crisis social media posts to analyze or classify different aspects of the text in posts to extract

insightful or actionable information pertinent to the crisis event being referenced.

3.2.1 Text Classification for Social Media Crisis Text

The authors in [5] sought to investigate the use of CNNs and word embeddings, specifically word2vec [41], either pretrained or finetuned to a large crisis corpus, for binary informativeness text classification and multiclass crisis category text classification.

Prior to performing any model development the authors in [5] computed the interannotator agreement (IAA) percentage for the datasets they obtained, which had the original labels provided by all of the annotators. Computing an IAA statistic for an annotated dataset is important because, as the authors state, "... a computer cannot generally agree with annotators at a rate that is higher than the rate at which the annotators agree with each other.", so the IAA is useful in assessing the difficulty humans have in agreeing on the proper label to prescribe a data point for a classification task. We provide a similar computation in our work using the Fleiss' Kappa statistic, or Fleiss' Kappa coefficient [42], to determine interannotator agreement on the annotated datasets we create as Fleiss' Kappa coefficient accounts for random chance agreement that occurs between annotators.

In evaluating the performance of CNN models, which used various word2vec embeddings, on the binary informativeness classification task, the authors in [5] found that the CNN models outperformed (by AUROC) classical classification algorithms, namely SVM, Logistic Regression, and Random Forest, which used TF-IDF features as input across various test sets corresponding to different crisis events. A similar result was observed for the multiclass classification task in which two variations of a CNN model mostly outperformed an SVM model by accuracy and macro-F1 scores across different data settings and different crisis events. The authors also provided plots showing the performance on each class by some of the models in addition to the aggregate performance metrics mentioned above. We provide similar per-class performance analysis in this work.

Similar to [1], recently a large, consolidated open-source dataset was compiled consisting of eight different datasets of crisis text tweets, collectively known as Cri-

sisBench, for the text classification tasks of Informativeness (∼ 166.1k tweets) and Humanitarian Categories (∼ 141.5k tweets) [13]. Performance benchmarks are provided for each of these tasks using various deep learning architectures including CNN, fastText [43], and pre-trained transformers, specifically BERT [44], DistilBERT [45], and RoBERTa [46]. The researchers conducted most of their experiments on an English-only dataset, which is a subset of a larger multilingual dataset, most of which is English text. They found that on the English-only consolidated dataset, the fine-tuned RoBERTa model achieves the highest weighted F1 scores of 0.883 and 0.872 for the informativeness and humanitarian categories tasks, respectively. On the multilingual dataset, which included Japanese tweets (although few), the researchers evaluated all of the same monolingual models used in the English-only dataset experiments and also tested multilingual variants of the transformers, namely BERT-m, DistilBERT-m, and XLM-RoBERTa. For the multilingual dataset, XLM-RoBERTa, BERT-m, and RoBERTa achieved the highest weighted F1 score of 0.879 for the informativeness task. Lastly, the XLM-RoBERTa model achieved a weighted F1 score of 0.788 on the multilingual dataset for the humanitarian categories task.

In addition to classifying crisis text into various categories, researchers have also investigated the potential to extract information from crisis text that is useful for gaining situational awareness and engaging in response during crisis.

3.2.2 Information Extraction from Social Media Crisis Text

Imran et al. devised a system with the goal of extracting useful information from crisis tweet text. The system first determines whether a tweet is informative or not for the crisis event. Then, the informative tweets are classified into descriptive classes that describe the content of the tweet such as "Caution and Advice", "Donations", "Casualties and Damage", etc. Finally, using Conditional Random Field (CRF) as the sequence labeling model and the concatenation of various text feature representations such as BOW based on unigrams and BOW based on bigrams as the input features to the CRF model, the system outputs words from the tweet that are relevant for the class the tweet was classified into [15].

3.2.3 Unsupervised Methods on Social Media Crisis Text

Clustering on text data (such as by K-means or K-medoids) is utilized on crisis tweets in order to find clusters of the tweets which share high textual similarity and thus potentially belong to the same overarching category. We discuss examples of this in Section 3.3.1 and Section 3.3.4. Latent Dirichlet Allocation (LDA) is a popular topic modeling technique in text analysis used to generate latent topics from a text corpus [47]. LDA is a generative probabilistic model that takes a hyperparameter K, which determines the number of latent topics the model finds across the corpus documents. LDA outputs a distribution over the words or n-grams in a corpus for each topic, where the words with the highest probability are considered the most relevant to the topic. Similarly, after fitting an LDA model to a text corpus, each document in the corpus is represented as a distribution across the K topics found by the model. In other words, a document is a mixture of the topics found by the LDA model, where some topics are more relevant to the document than others. An application of LDA on crisis tweets is discussed in Section 3.3.4.

3.3 Crowdsourced Crisis Information Systems using Artificial Intelligence

To study and evaluate machine learning methods used within crowdsourced crisis information systems, researchers have developed AI-powered systems which aim to utilize a variety of machine learning techniques to mitigate information overload and assist in gaining situational awareness.

3.3.1 Tweet4act

Tweet4act is a system that filters for informative crisis-related tweets by utilizing K-medoids clustering and predicts the phase of a crisis (i.e. pre, during, post crisis periods) an informative tweet is discussing [16].

Tweet Informativeness Clustering

Evaluating their informativeness clustering algorithm on tweets for datasets corresponding to the 2011 Joplin Tornado, 2010 Haiti Earthquake, and the 2011 Nesat Typhoon, the authors report precision scores of 96%, 100%, and 97% and recall scores of 85%, 88%, and 84%, respectively.

Crisis Period Classification

Once clustered, the system then classifies the clustered informative tweets into categories depending on the most likely period of the crisis the tweet content is discussing. The tweets were classified into the crisis period by comparing the words in a tweet against an incident dictionary and by the identification of the verb tenses [16]. The authors compare the performance of the Tweet4act crisis-period classifier to the classification algorithms of SVM, Maximum Entropy, Decision Tree, and Random Forest. They found that the Tweet4act crisis-phase classifier performed better by a large margin on the Joplin Tornado dataset, worse on the Haiti Earthquake dataset by a smaller margin, and tied on the Nesat Typhoon dataset when compared to the other algorithms.

3.3.2 DISAANA and D-SUMM Large-scale Crowdsourced Japanese Crisis Text Analyzers

In the aftermath of the 2011 Great East Japan Earthquake, there was a lot of useful information posted to Twitter, however most people were overwhelmed by the sheer amount of information that was published to the platform about the crisis event. The information overload that resulted led to widespread confusion and an inability to make decisions. To assist crisis victims and crisis responders in the aftermath of large crisis events, DISAANA and D-SUMM NLP text analyzers were developed by researchers in [48]. These systems were built for analyzing Japanese crisis tweets.

DISAANA provides tools for Question-Answering (QA) and problem-listing. For QA, a user provides a question, e.g. "What is in short supply in Kumamoto?", the

system then responds with various answers classified by semantic categories, e.g. Category: "daily necessities" → Answers: "portable toilets", "baby necessities", "dried milk". The system also enables user to map the answers to locations on a map using a large location database that has part-of relations between locations. The problem-listing tool prompts a users to first select a specific area, e.g. Kumamoto Prefecture, then generates a list of problems found in tweets corresponding to that area, e.g. "landslides occur", "aftershocks continue". The DISAANA system was used by the Japanese government during the 2016 Kumamoto Earthquake.

It was observed that DISAANA often provides too many items in the problem-listing tool, thus D-SUMM was developed to summarize the list of problems found for a specific area discussed above [48].

3.3.3 AIDR: Artificial Intelligence for Disaster Response

AIDR is a crisis information system which aims to leverage human and machine intelligence in order to minimize the information overload that comes with the influx of tweet streams during crisis while also improving predictive model performance by applying active learning [33, 39, 49]. At a high-level, the AIDR system consists of three modules: Collector, Trainer, and Tagger. The Collector selectively consumes text tweets during crisis on the basis of user-specified keywords, geographic region via a user-defined bounding box, and/or language using the Twitter API (Application Programming Interface). The system allows a user to define categories for which a Random Forest classifier will classify the consumed tweets into [33, 49]. The text is featurized as BOW based on unigrams or bigrams as input to the classifier. A carefully selected subset of newly attained data is sent to an external crowdsourced labeling platform, e.g. CrowdFlower, or to volunteers using a labeling tool that is internal to the AIDR system, to provide labeled training and testing data for the classifier. The remainder of the data is classified by the trained classifier automatically. After training, predictions can be made by the classifier in real-time via an API to assist in crisis mapping or visualizations [33]. These steps are repeated periodically as to keep classifiers up-to-date and as accurate as possible using in-crisis data to train,

test, and predict.

Active Learning

In an effort to reduce the amount of human labeling required to improve the classifier's performance, AIDR selects the subset of the data which would best improve the model's performance [49]. These data points are typically those which lie close to the decision boundary the model has learned from the training data. These are the data points the model is most uncertain about in its prediction. When labeled, these data points are considered to be the ones which would most improve the model's predictive performance after retraining on them. The paradigm of using an ML model to inform the selection of a subset of unlabeled data to be labeled by humans in order to improve the model's performance when retrained is active learning. Active learning is in contrast to passive learning in which the training data is entirely selected by a practitioner and then given to the learning algorithm to train on all at once [39, 49].

The system was able to achieve an AUROC score of 80% on the informativeness classification task for tweets posted during the 2013 Pakistan Earthquake [33]. The system provided low-latency, high-throughput, scalability, flexibility, cost-effectiveness, and good classification quality by AUROC score [49]. The system provides an interesting framework for integrating human and machine intelligence to amplify the efficacy of both through active learning. The system also allows for the flexibility of user-defined classification tasks while also training a classifier with fewer labeled examples to achieve slightly better classification performance through active learning than in the passive learning setting.

With reports that have multiple data modalities such as text and image data, it is imperative that crowdsourced crisis information systems provide analysis for both data modalities to better improve situational awareness beyond that which could be attained by either one individually.

3.3.4 Visual and Descriptive Summaries of Disaster Events using AI

To utilize the often complementary information derived from having both image and text data in crisis tweets, the researchers in [50] explored a variety of supervised and unsupervised ML techniques applied to the text and image modalities. They constructed a system which provides a unified framework for using these techniques to provide a variety of visualizations which aim to improve situational awareness by simultaneously using image, text, and location (when available) data from tweets. The data used in this study was collected using Twitter's streaming API using keywords and time intervals related to the 2017 Hurricanes Harvey, Irma, and Maria.

Unsupervised Methods on Crisis Tweets

In order to uncover categories pertinent to the ongoing crisis event using unlabeled text data from crisis posts, the authors applied K-means clustering on the potentially relevant tweets of each day of each crisis event to assist human annotators in assigning an overarching, interpretable category to each cluster found. The researchers embedded the tweets into document embeddings using the average of the word embeddings found for a tweet. These word embeddings were finetuned on a large English crisis dataset using a Continuous Bag-of-Words (CBOW) word2vec model. Fixing the cumulative variance to 50%, PCA is then applied to the document embeddings. After applying grid search to find the optimal number of clusters on the basis of maximum silhouette score, a human annotator observes the 10 tweets closest to the cluster center and determines a category to assign to the cluster. We were inspired by this use of unsupervised learning when we pursued the application of dimensionality reduction and clustering algorithms to the Fukuchiyama Japanese crisis text reports dataset.

The system also utilizes LDA and provides the top 30 most relevant words for each of the 10 latent topics ($K = 10$) to assist in understanding the topics found by LDA for a specific day.

Text Classification and Named Entity Recognition

For text data, the system classifies text on the basis of relevancy (identical to the informativeness classification task previously mentioned), and then classifies sentiment and humanitarian category on tweets classified as relevant. For sentiment analysis, the authors used the Stanford sentiment analysis classifier [51], they then study the temporal aspects of sentiment, specifically the proportions of predicted sentiment categories (Negative, Neutral, and Positive) over several days, which coincided with each hurricane event. For the relevancy and humanitarian category classification tasks, a Random Forest model is used as the classifier. Labeled twitter data and corresponding Train/Dev/Test splits provided from prior work are used for training (Train), hyperparameter tuning (Dev), and testing (Test). For the relevancy classification task, the model achieves an F1 score of 0.82 on the test split. For the humanitarian category classification task, the model is reported as having attained a weighted F1 score and weighted accuracy score of 0.64 and 0.66, respectively.

After classifying tweets into humanitarian categories, the system uses the Stanford Named Entity Recognition (NER) toolkit [52] to extract the most frequent entities in the "donation and volunteering" category. By using NER, the system allows for deeper analysis of the tweets by surfacing the most discussed organizations and people within a class of the humanitarian categories task.

Image Relevancy, Deduplication, and Classification

For image data, the researchers used data from prior work to finetune a pretrained CNN for image relevancy/irrelevancy classification. The image relevancy classifier achieves 99% precision and 97% recall on a held-out test set.

Aiming to identify exact- and near-duplicate images, the authors use a perceptual hashing algorithm to determine if images are exact- or near-duplicates using a similarity threshold determined in a prior work, which yielded $\sim 90\%$ precision and recall on a held-out test set.

For the damage assessment task, the classes of the task included "Severe", "Mild",

and "None". The authors finetune a CNN for the task using data from prior work. The authors report the model as having an overall accuracy ranging from 76% to 90% on held-out test sets corresponding to different crisis types.

Using the predicted classifications made on the text and image modalities, classified, geotagged tweets are placed on a map, color-coded based on the predicted class label.

Throughout our discussion of the prior work (including REACT), we have noted the ways in which our work takes inspiration from the applications of various ML techniques and methods of evaluation. We now describe how our work differs from prior work in some novel ways.

Since we focused our study on flood crisis events, we developed the novel binary classification task of Flood Presence and construct a dataset and splits that are of comparable size to those presented in [1]. We create Train/Dev/Test splits and report a performance benchmark on the test split for the crisis informatics community to use in future work.

We devise an image annotation procedure with a corresponding guide for annotators to use in their annotation of crisis images. We conducted an annotation effort to annotate the unlabeled Fukuchiyama crisis images using classification tasks presented in prior work, adapting the corresponding class label definitions from prior work, and creating a new set of definitions for classes of the novel Flood Presence task. After conducting the annotation effort, we compute the Fleiss' Kappa coefficient for each task-specific dataset, since Fleiss' Kappa takes into account random chance agreement between annotators. The use of Fleiss' Kappa coefficient is a novel component of this work, as prior work in crisis informatics has either not reported an IAA statistic for an annotated dataset or reported the percentage of agreement between annotators, which does not take into account random chance agreement and thus is likely an optimistic measure of the agreement between annotators for the task.

In an effort to address the sparse research on Japanese crisis text, the focus of our Text Analysis Module is on Japanese text. We use the language-agnostic text fea-

turizations discussed in the prior work, that is, BOW based on unigrams or bigrams, and TF-IDF based on unigrams featurizations. In addition to using the mentioned language-agnostic featurizations, a novel aspect of our work is that we also use a text featurization which is optimized for Japanese text, namely pretrained Japanese MLM BERT Model with CLS Pooling embeddings. We use these embeddings in experiments involving supervised and unsupervised ML methods.

Finally, with our partners in Fukuchiyama, we form a case-study for mitigating information overload of crisis reports during flood crisis events in Fukuchiyama. A major aim of this work is to both quantitatively and qualitatively evaluate the efficacy of ML techniques in mitigating information overload of crisis managers and meeting their information needs. In gaining insights and feedback from crisis managers through image annotation workshops, we used those insights, in addition to the labels provided to us directly from crisis managers, to develop a classification task which has labels that are more closely tied to the expressed information needs of crisis managers. Additionally, we used these insights to develop models that use a performance metric which is aligned with the priorities of crisis managers as it relates to the task. To the best of our knowledge, this exercise has not been conducted in prior work.

In the next two chapters, we dive into the details of the Image Analysis Module and the Text Analysis Module, which together showcase the components of the framework presented in the Motivation section, the supervised and unsupervised ML methods we employ in our experiments, and the quantitative and qualitative evaluation methods we use to assess the efficacy of our approach in mitigating information overload and automating crisis report assessment for enhanced crisis awareness and response. For each module, after presenting the ML methodology, we report our results and discuss our interpretations of the findings. Additionally, for each module, we include an implementation discussion, noting features that enable the crisis informatics research community to reuse the analysis contained herein for future research.

Chapter 4

Image Analysis Module

The goal of the Image Analysis Module is to utilize pretrained CNN models to yield efficient and accurate predictions from image data in crowdsourced crisis reports. The classification tasks of damage severity, humanitarian categories, informativeness, and flood presence form a diverse suite of labels. In a fraction of a second, the model predictions made for these tasks provide a series of categorizations for an individual report. We leverage state-of-the-art CNNs, which strike a necessary balance between model complexity, memory and storage constraints, and model performance to provide these predictions.

To achieve this aim, we use large, labeled, open-source datasets for training and evaluating models. In addition to using open-source datasets for the prediction tasks of damage severity, humanitarian categories, and informativeness, we formulate a new crisis image classification task for detecting flood presence in an image and construct a new dataset altogether using flood images from [1], [53], and [54], which contained flood-adjacent labels which we relabeled to consolidate these datasets into a larger dataset consisting of images labeled as "Flood" and "Not Flood". We additionally devise an annotation procedure and analyze the results of annotations provided by multiple annotators on crisis images provided to us by crisis managers in Fukuchiyama. We describe the models we employed in this work to perform the image classification tasks described above and the associated experiments we conducted. Finally, we describe the quantitative and qualitative evaluation procedure for the Image Analysis

Module.

4.1 Image Classification Tasks

For each of the image classification tasks mentioned above, we provide the definitions of their corresponding classes. With the exception of the Flood Presence task, the class definitions below are based on existing definitions determined in prior research and we augment some of these definitions with additional text to provide additional clarification that is pertinent to flood crisis events, e.g. damage to terrain or landslides. Wherever we augment these definitions, **we show the added text in bold**. We note that the classes associated with each of the tasks are on a nominal scale, that is, for each task, we treat the classes as if there is no ordering between them. Additionally, the classes associated with each task are mutually-exclusive, that is, only one class can be selected for the classification.

4.1.1 Damage Severity

An issue with the task of predicting the severity of damage to infrastructure is the inherent subjectivity of the task. In other words, what one person may consider severe damage, someone else may consider to be mild [34]. It is important to have well-defined, clear, and distinct class definitions for the damage severity task. The following category descriptions of damage severity were originally developed in [34] and are also used in [1]. The original definitions were constructed for exclusively structural damages such as "broken bridges, collapsed or shattered buildings, destroyed or cracked roads etc.", so we have added additional text to include damage to terrain, and damage from flooding, which is in bold:

- **Little or None**: Images that show damage-free infrastructure **or terrain** (except for wear and tear due to age or disrepair) belong to the no-damage category.

- **Mild**: Damage generally exceeding minor [damage] with up to 50% of a building, for example, in the focus of the image sustaining partial loss of amenity/roof. Maybe only part of the building has to be closed down, but other parts can still be used. In case of a bridge, if the bridge can still be used, but, part of it is unusable and/or needs some amount of repairs. **This also includes terrain which has been somewhat impacted by the disaster event.**

- **Severe**: Images that show substantial destruction of an infrastructure **or terrain** belong to the severe damage category. A non-livable or non-usable building, a non-crossable bridge, or a non-drivable road are all examples of severely damaged infrastructures. **This includes terrain which has been severely affected by landslide or flooding.**

4.1.2 Humanitarian Categories

The humanitarian categories classification task aims to classify images based on four categories depicting different types of information related to humanitarian aid. These class definitions were originally formulated in [11] and are adapted in [1]. The "Not Humanitarian" category below was originally "Not relevant or can't judge" in [11]. In [1], the original definitions of "Affected individuals" and "Injured or dead people" defined in [11] are merged to form the category "Affected, Injured, or Dead People" below. To include characteristics unique to flood crisis events as well as damage to terrain, we added additional text to the "Infrastructure and Utility Damage" category, which is in bold:

- **Not Humanitarian (NH)**: If the image is irrelevant or you can't judge, for example, due to its low-quality.

- **Infrastructure and Utility Damage (IAUD)**: Image reports/shows any built **or land structure** affected or damaged by earthquake, fire, heavy rain, floods, strong winds, gusts, etc. such as damaged houses, roads, buildings; flooded houses, **terrain with landslides**, streets, highways; blocked roads, bridges, pathways; collapsed bridges, power lines, communication poles, etc.

- **Rescue, Volunteering, or Donation Effort (RVDE)**: If the image reports/shows any type of rescue, volunteering, or donation effort such as people being transported to safe places, people being evacuated from the hazardous area, people receiving medical aid or food, people in shelter facilities, donation of money, blood, or services, etc.

- **Affected, Injured, or Dead People (AIDP)**: If the image reports/shows people affected by the disaster event such as people sitting outside; people standing in queues to receive aid; people in need of shelter facilities, injured or dead people.

4.1.3 Informativeness

The informativeness of a report is defined by its usefulness for humanitarian aid purposes [11]. As such, the class definitions for the binary informativeness classification task were originally defined in [11] and are used in [1]. We have added text to include "normal day", or scenes which show an area unaffected by a natural hazard in the "Not Informative" category:

- **Not Informative**: Images showing banners, logos, and cartoons, **or normal day scenes unaffected by disaster** are not considered as "Informative".

- **Informative**: The image is considered "Informative" if it reports/shows one or more of the following: cautions, advice, and warnings, injured, dead, or affected people, rescue, volunteering, or donation request or effort, damaged houses, damaged roads, damaged buildings; flooded houses, flooded streets; blocked roads, blocked bridges, blocked pathways; any built structure affected by earthquake, fire, heavy rain, strong winds, gust, etc., disaster area maps.

4.1.4 Flood Presence

Flood crisis events are the focus of this work, so we devise a new classification task specific to the flood crisis context, that of flood presence, or detecting flood presence

in an image. This task is inspired from the "No Heavy Flooding"/"Heavy Flooding" binary multimodal report classification task in REACT [28].

We construct the following class definitions for this newly-defined task:

- **Not Flood**: Images that show no flooding, including bodies of water which are normal, i.e. not overflowing with water or landscapes or infrastructure which are normal, i.e. not inundated with water.
- **Flood**: Images showing flooding of streets, homes, land, or submerged infrastructure.

4.2 Image Datasets

For training and evaluating the image models discussed in Section 4.4 for the classification tasks described in Section 4.1, we leverage open-sourced labeled datasets, combine various datasets together to form a larger dataset for the flood presence task, annotate images provided by crisis managers in Fukuchiyama and subsequently construct a new dataset for model evaluation.

4.2.1 Open-source Consolidated Crisis Image Datasets

As discussed in Section 3.1.2, the researchers in [1] developed large, open-source datasets for the image classification tasks of disaster type, damage severity, humanitarian categories, and informativeness. The images composing these consolidated datasets were aggregated using images from the Damage Multimodal Dataset (DMD) [10], the CrisisMMD dataset [11], images collected from the AIDR system [33], and the Damage Assessment Dataset (DAD) [34], which together cover various past crisis events across a variety of geographical contexts. For the damage severity, humanitarian categories, and informativeness tasks, we use these datasets directly and their corresponding randomized, non-overlapping Train/Dev/Test splits[1] provided in [1].

[1]The consolidated crisis image datasets and their corresponding splits can be found here: https://crisisnlp.qcri.org/crisis-image-datasets-asonam20

The Train/Dev/Test splits are in respective percentages of 70%/10%/20% of the large, consolidated dataset for the corresponding task. We use these splits directly to train and evaluate models for the damage severity, humanitarian categories, and informativeness tasks.

4.2.2 Consolidation of Flood-Related Image Datasets to Form the Flood Presence Dataset

As mentioned above, [1] also provides labeled data along with Train/Dev/Test splits for the task of disaster-type classification from crisis images. The disaster type classification task has classes for "Earthquake", "Fire", "Flood", "Hurricane", "Landslide", "Other Disaster", and "Not Disaster". Though we do not explore the disaster type classification task in our work as we focused on flood crisis events, we used the labeled data for "Flood" class as the positive class and relabeled all other classes as the negative "Not Flood" class for the binary flood presence classification task discussed above. Since the authors in [1] provided Train/Dev/Test splits, we note that if a specific data point was present in a specific split provided by the authors, we keep that data point in the same split that we create for the flood presence task after consolidating various open-source datasets. For example, if a data point was originally present in the train split for disaster type in [1], we automatically put that data point in the final flood presence train split we create. These images form the base of our Train/Dev/Test splits for the Flood Presence classification task.

The authors in [53] collected 3435 images from Wikimedia Commons for the May and June 2013 Central European floods and 275 additional images containing water pollution which were manually harvested by using image search engines to search for images from major oil spill events from recent years from when the study took place [53]. Thus, there are 3710 images in total from this source. These images were each provided three separate boolean labels by hydrologists. These labels were whether or not an image indicated a certain area was flooded, whether or not the depth of inundation could be estimated from visual cues present in the images such as traffic

signs or structures whose height is known, and finally a label indicating whether or not the image shows pollution substances such as oil. We refer to these labels as the "Flooding", "Depth", and "Pollution" labels, respectively. In this work, we use all images for which the Flooding label was true as the positive "Flood" label, and all images for which this label is false as the negative "Not Flood" label in the flood presence task. We call this resulting dataset the Central European Floods 2013 dataset. We note that 890 of these images contain geographical metadata about the geographical location where the photo was taken. Using geographical metadata, such as latitude and longitude coordinates, will be investigated in a future study. We note that the authors provided two variations of the images in this dataset, in which images are resized such that the smaller side of an image is at most 512 pixels (1.1 GB in total), or alternatively at most 1280 pixels (5.6 GB in total). In this work, we use the images for which the smaller side of an image is at most 512 pixels.

The same boolean labels discussed above were used to provide labels by expert annotators for the Twitter flood image datasets constructed in [54]. These datasets were created by extracting images from tweets which contained keywords related to flood and which were tweeted during the flood crisis events that occurred in the Harz region in Germany in July 2017 and the Rhine River in January 2018. We relabel the images in these datasets to be "Flood" and "Not Flood" as we did for the Central European Floods 2013 dataset. We refer to the resulting datasets as the Harz 2017 and Rhine 2018 datasets.

We combine the labeled images from the Central European Floods 2013, Harz 2017, and Rhine 2018 datasets and randomly split these images into percentages of 70%/10%/20% for the flood presence Train/Dev/Test splits, which already contained the relabeled images from the disaster type Train/Dev/Test splits in [1]. Combining all of these relabeled images from [1], [53], and [54] yielded Train/Dev/Test splits into approximate percentages of 70%/10%/20% for the entire flood presence dataset. These open-source datasets provided us a means to form a new, larger dataset for benchmarking the performance of image classification models on the test set split we constructed for the flood presence task. We refer to the labeled dataset made from

this data as the Flood Presence Dataset.[2] We show the composition of the Flood Presence dataset formed from the open-source datasets discussed above in Table 4.1.

▲ Table 4.1: Number of Labeled Images by class from various Open-source Flood-related Datasets which compose the Flood Presence Dataset

Original Dataset	Not Flood	Flood	Total
Consolidated Disaster Types [1]	14310 (81.7%)	3201 (18.3%)	17511 (100%)
Central European Floods 2013 [53]	559 (15.1%)	3151 (84.9%)	3710 (100%)
Harz 2017 [54]	405 (60.5%)	264 (39.5%)	669 (100%)
Rhine 2018 [54]	1007 (58.0%)	730 (42.0%)	1737 (100%)

We show the number of data points, or the support, associated with each class in the Train/Dev/Test splits provided in [1] for the damage severity, humanitarian categories, and informativeness tasks as well as the support for the classes in the Train/Dev/Test splits for the Flood Presence dataset in Table 4.2.

4.2.3 Annotation of Fukuchiyama Crisis Images

In collaboration with crisis managers in Fukuchiyama, we were provided hundreds of images from previous flood events as well as non-crisis normal days in Fukuchiyama, which were collected on the ground, similar to RiskMap images mentioned in Section 2.1.1. To utilize this data for evaluating the models in our experiments, four members of the Urban Risk Lab[3] independently labeled the images using the definitions for the classes in each task described in Section 4.1 as well as an illustrative example image associated with each class.[4]

In total, there are 658 unique images from Fukuchiyama where each image is independently labeled by 3 different annotators. When an annotator labeled an image, they provided a single label for each of the four classification tasks discussed in Section

[2] Link to Flood Presence Dataset Creation Code

[3] We acknowledge the following members of the Urban Risk Lab at MIT who performed the annotation: Saeko Baird, Aditya Barve, Dylan Lewis, and Kyoko Murayama

[4] Link to Annotation Materials

▲ Table 4.2: Number of Images in randomized, non-overlapping Train/Dev/Test Splits by class for the Image Classification Tasks specified in Section 4.1

Crisis Image Classification Train/Dev/Test Splits				
Task Name	**Class Name**	**Train**	**Dev**	**Test**
Damage Severity	Little or None	11437	1378	2135
	Mild	4072	489	629
	Severe	12810	845	1101
Humanitarian Categories	Not Humanitarian	6076	578	1550
	Infrastructure and Utility Damage	4001	406	821
	Rescue, Volunteering, or Donation Effort	1769	172	391
	Affected, Injured, or Dead People	772	73	160
Informativeness	Not Informative	21700	1622	5063
	Informative	26486	1432	3414
Flood Presence	Not Flood	11900	1398	2983
	Flood	5226	684	1436

4.1. Furthermore, two annotators labeled all 658 images, a third annotator labeled 470 of the images, and finally a fourth annotator labeled the remaining 188 images.

To provide a statistical measure of the level of inter-annotator agreement (IAA) on the labeled data between the annotators, we computed a Fleiss' Kappa coefficient [42] for each of the classification tasks, which is valued in the range [-1, +1]. The motivation for using an IAA statistic such as the Fleiss' Kappa coefficient is that it estimates the extent of agreement on the labels the annotators provided for a task over what would be expected if the annotators labeled the data entirely at random, i.e. the expected level of agreement by random chance. By correcting for the expected level of agreement by random chance in order to provide a normalized measure of the overall level agreement, Fleiss' Kappa coefficient is an important statistic for reporting the

IAA for a classification task dataset with nominal categories (i.e. no ordering between mutually exclusive, exhaustive categories), in addition to the percent of unanimous agreement and percent of plurality agreement discussed below. We use the Fleiss' Kappa coefficient over other statistical measures of IAA, for example Cohen's Kappa coefficient [55] or Scott's pi statistic [56], because it generalizes to more than two annotators on nominal categories for a task, which is applicable to the annotation effort conducted in this work. Additionally, Fleiss' Kappa coefficient provides the flexibility such that as long as there are a fixed number of labels, n, provided to all of the data points by n independent annotators, any n independent annotators can label a particular data point. That is, it is not required for the same n annotators to label all of the data points. Thus, Fleiss' Kappa coefficient is appropriate for the annotation procedure we conducted in this work.

Generally, the closer to $+1$ the Fleiss' Kappa coefficient, the higher the overall level of agreement is among the annotators compared to the expected level of agreement by random chance for the task and the more confident we can be in the reliability of the labels [42]. Fleiss' Kappa coefficient is a reasonable proxy measure of how useful the labeling guide was as a method for assisting independent annotators in providing consistent labels for data points and thus how reproducible the annotation procedure is for a classification task. If there is a low level of agreement on the labeled data, it can lead to forming a dataset of poor quality for training or evaluating a classification model or in the extreme case, the inability to form a dataset at all. Low agreement demonstrates a need for refining the annotation procedure, i.e. the labeling guide, such as replacing a vague definition for a class with checklists containing specific criterion to meet in order for a data point to qualify as being labeled as a specific class. We provide more detail on the Fleiss' Kappa coefficient computation for an annotated dataset in Appendix Section B.1.

Using the labels provided for each task for each image by the annotators, we determine the ground truth label of an image to be the plurality/mode label, i.e. the label that appears most frequently. We use the plurality/mode label as ground-truth in order to diminish the impact of any specific annotator's bias. In the general

case, plurality agreement is achieved when there exists a unique most frequent label amongst the labels provided by all of the annotators for an image. Therefore in the case of 3 annotators, plurality agreement is achieved when at least 2 out of the 3 annotators provided the same label for the image. We note that in the case of three annotators, binary classification tasks such as informativeness and flood presence will have a plurality agreement percentage of 100% regardless of which specific labels are provided by the annotators.

We report the Fleiss' Kappa coefficient, percentage of unanimous agreement (i.e. all three annotators provided the same label for an image), and percentage of plurality agreement for each task mentioned in Section 4.1 in Table 4.3.

▲ Table 4.3: Agreement Measures by Task for Labeled Fukuchiyama Crisis Images

Task	**Fleiss' Kappa Coefficient, κ**	**Unanimous Agreement Percentage**	**Plurality Agreement Percentage**
Damage Severity	0.413	42.9%	97.7%
Humanitarian Categories	0.304	42.4%	96.7%
Informativeness	0.313	74.9%	100%
Flood Presence	0.829	87.2%	100%

From Table 4.3, we observe Fleiss' Kappa coefficients for the tasks of damage severity, humanitarian categories, informativeness, and flood presence to be 0.413, 0.304, 0.313, and 0.829, respectively. Comparatively, the Fleiss' Kappa coefficients for damage severity, humanitarian categories, and informativeness tasks are significantly lower than that for the flood presence task. This suggests that further investigation should be conducted in refining the labels and annotation guide by understanding potential causes for the interannotator disagreement on those tasks in order to improve the interannotator agreement and thus the quality of the dataset.

To the best of our knowledge, the field of crisis informatics has no standard guidelines or methodology for determining an acceptable minimum Fleiss' Kappa coeffi-

cient or similar IAA statistics. We are interested in exploring this further in a future study. In this work, we use this labeled data from Fukuchiyama to form ground-truth datasets.

Fukuchiyama Ground Truth Image Datasets

For each of the image classification tasks, we construct a test dataset including the images from the Fukuchiyama crisis images which achieved plurality agreement for the label of that image for the task. When constructing a dataset for each task, we use the plurality label as the ground-truth label for the image for the task and thus exclude images which do not have a plurality label for that task. The resulting support for each class in each task for the Fukuchiyama Flood Crisis Image Dataset is shown in Table 4.4.

4.3 Image Preprocessing

To convert the raw images in various file formats (e.g. .jpeg, .jpg, .png, etc.) into featurized input for the CNN models, we make several transformations to the images which depend on whether they are used for training, development, testing, or inference by the models. All of the images are represented using red, green, blue color channels and the pixel value in each channel has a value in the range [0, 255].

In order to prevent the neural networks from overfitting to the training data, we employ regularization by applying various data augmentation techniques to improve the model's generalizability [57]. Online data augmentation occurs on the images in the training dataset by applying random transformations to the original training images. The random transformations are applied to images during training at the start of each batch iteration during a training epoch.

More specifically, in each training batch, for each image, at a random location in the image, a portion of the original image is cropped at random using a uniformly selected proportion in the range [0.8, 1.0] of the area of the original image and a uniformly selected aspect ratio for the crop in the range of $\left[\frac{3}{4}, \frac{4}{3}\right]$. This cropped

▲ Table 4.4: Support for each class of the Image Classification Tasks specified in Section 4.1 for Fukuchiyama Crisis Images

Labeled Fukuchiyama Crisis Image Datasets		
Task Name	**Class Name**	**Support**
Damage Severity	Little or None	182 (28.3%)
	Mild	236 (36.7%)
	Severe	225 (35.0%)
Humanitarian Categories	Not Humanitarian	184 (28.9%)
	Infrastructure and Utility Damage	399 (62.7%)
	Rescue, Volunteering, or Donation Effort	40 (6.29%)
	Affected, Injured, or Dead People	13 (2.04%)
Informativeness	Not Informative	69 (10.5%)
	Informative	589 (89.5%)
Flood Presence	Not Flood	358 (54.4%)
	Flood	300 (45.6%)

image is then resized to 256 pixels by 256 pixels representing the height and width of an image, respectively. Then, the resized image is rotated about the center with a random angle selected uniformly in the range of [-15, 15] degrees. The rotated image is then flipped horizontally (left-right) with a probability of 0.5.

The first transformation applied to images used for development, testing, or inference before the transformations mentioned below is the resizing of the original image to 256 pixels by 256 pixels.

The following final image transformations are applied to all images regardless of which split they are a part of. The image is cropped at the center to be of size 224 pixels by 224 pixels. The images are then converted into 3-channel tensors,

representing the red, green, blue color channels with each pixel value in a channel being in the range [0, 1]. Finally, the pixels in each color channel are normalized using mean values of 0.485, 0.456, and 0.406 and standard deviation values of 0.229, 0.224, and 0.225 for the red, green, and blue channels, respectively. This normalization is required for the pretrained PyTorch [58] CNN models used in this work.

4.4 Image Classification Models and Training

We now begin our discussion of the CNN models we use to perform the image classification tasks mentioned in Section 4.1.

4.4.1 EfficientNet-B1 CNN

EfficientNets are a family of image classification CNN models, which were developed to achieve competitive accuracy with other state-of-the-art image classification models, while also being both an order of magnitude smaller in the number of model parameters and faster compared to their state-of-the-art counterparts. For example, the Efficient-B1 model has $\sim$ 7.8 million parameters and achieves a Top-1 accuracy of 79.1% on ImageNet, while the ResNet152 model has $\sim$ 60 million parameters and achieves a Top-1 accuracy of 77.8% on ImageNet. In addition to being an order of magnitude smaller in terms of the number of parameters, the EfficientNet-B1 model was shown to be 5.7 times faster in inferencing on a single core of Intel Xeon CPU E5-2690 than ResNet152 [22]. Finally, since the EfficientNet models are pretrained on the large ImageNet dataset, these pretrained models can be utilized for finetuning or feature extraction for domain-specific image classification tasks with relatively small datasets as discussed in Section 3.1.1.

In [1], the authors finetune a variety of state-of-the-art image classification model architectures pretrained on ImageNet using their train and dev splits for the tasks of classifying damage severity, humanitarian categories, informativeness, and disaster type. These model architectures include ResNet18, ResNet50, ResNet101, AlexNet, VGG16, DenseNet, SqueezeNet, InceptionNet, MobileNet (MobNet (v2)),

and EfficientNet-B1. Using weighted F1 as the performance metric for which to compare model performance on the test splits for each of the tasks, the authors determine that the EfficientNet-B1 model performs best on the tasks of damage severity, informativeness, and disaster type achieving a weighted F1 score of 75.8%, 86.3%, and 81.6%, respectively. On the humanitarian categories task, the VGG16 model performs best with a weighted F1 of 77.3% and the EfficentNet-B1 and ResNet101 models are tied for second best with a weighted F1 score of 76.5%. The authors conclude that due to the lower computational complexity and competitive performance on each of the tasks for the EfficientNet-B1 model compared to the other models, it is the best model to use in real-time applications, such as RiskMap. From this conclusion, we focus on using the EfficientNet-B1 architecture for image classification in this work.

4.4.2 Training on Large, Consolidated Crisis Image Datasets and the Flood Presence Dataset

Similar to [1], we use EfficientNet-B1 models pretrained on ImageNet for the image classification tasks in our work. We train the models to perform their respective classification tasks using the train and dev splits in [1] for the damage severity, humanitarian categories, and informativeness tasks. For the flood presence task, we train the model using the train and dev splits we constructed for the Flood Presence Dataset described in Section 4.2.2. The frequencies of the labels in the splits for each of the tasks can be seen in Table 4.2. We finetune the models, that is, we do not freeze the pretrained weights, and we randomly initialize the weights between the final hidden layer of the pretrained model and construct an output layer to have one output node in the case of binary classification or the same number of nodes as there are classes for the case of multiclass classification, where there are more than two classes for the task. For the binary classification tasks of informativeness and flood presence, we use negative log-likelihood loss as the loss function. For the multiclass tasks of damage severity and humanitarian categories, we use categorical cross entropy loss as the loss function. We note that the training data is reshuffled at every epoch.

Each model is trained using the same set of hyperparameter values. We present these hyperparameters and their values in Table 4.5. We use the Adam [59] optimizer algorithm for updating model weights throughout training, an initial learning rate of 10^{-4}, and train for 25 epochs. We use unweighted accuracy as the metric for monitoring the model's performance on the train and dev sets throughout training. If the model's maximum accuracy on the dev set does not improve for 11 consecutive epochs (i.e. patience of 10), we reduce the learning rate by a factor of 10. If the learning rate is reduced, the count of consecutive no-improvement epochs is reset to zero. We save the weights of the model at the epoch for which the model attains maximum accuracy on the dev set across all 25 epochs. These saved weights are used in the model performance evaluation described in Section 4.5.

We note that the models developed for each of the classification tasks in Section 4.1 have $\sim$ 6.5 million total parameters. The models were trained using the graphics processing unit (GPU) model Tesla P100-PCIE-16GB in order to dramatically speed up the training process. We used a batch size of 128 images for the training data. We note, however, that the GPU model we used for training had insufficient memory for what was required for batches of this size, thus we use gradient accumulation for two steps on batches of size 64, respectively. That is, we sequentially accumulate the gradients for two batches each containing 64 unique images from the original batch of 128 images. After the gradients have been accumulated for these two "mini-batches" formed from the original batch of 128 images, then the weights of the model are updated.

4.5 Evaluation

The focus of this module is to provide automated, accurate, and informative classifications of crowdsourced crisis images. Thus we evaluate the image analysis module in terms of accurately classifying crisis image data, i.e. model performance, and in our assessment of the informative utility the predicted labels have for crisis managers towards enhanced crisis awareness and response.

▲ Table 4.5: Hyperparameters for EfficientNet-B1 CNN Models

Hyperparameter	Value
Architecture	EfficientNet-B1
Optimizer Algorithm	Adam
Initial Learning Rate	10^{-4}
Total Epochs	25
Metric to Optimize on Dev Set	Accuracy
Patience	10
Learning Rate Reduction Factor	10
Batch Size	128

4.5.1 Quantitative Evaluation: Model Performance

For most of the test splits for the consolidated crisis image datasets (as seen in Table 4.2), we observe from their label distributions that there is an imbalance in the data, i.e. one or more classes appears disproportionately higher than some of the other classes. In the Fukuchiyama datasets, we also observe imbalance, so like the authors in [1], we use weighted metrics to assess overall model performance to take into account class imbalance, i.e. by taking the weighted sum across the per class metric scores and weighting the per-class metric scores by the ratio of support for that class to total number of data points in the test set (weighted precision, weighted recall, and weighted F1) or by taking the average of the per-class metric scores (balanced accuracy). We also report the Cohen's Kappa coefficient [55] of the models for each of the tasks on the Fukuchiyama crisis images. We report Cohen's Kappa coefficient to provide a measure of the degree the EfficientNet-B1 models performs better than a baseline classifier which predicts randomly according to the label distributions of the Fukuchiyama image data for the task. The Cohen's Kappa coefficient computation is detailed in Appendix Section B.2.

Finally, we report the per-class metrics of precision, recall, and F1 along with the

confusion matrix for each task to have more granular insight into how the models perform for each specific class for a task and to understand the types of mistakes a model makes when misclassifying the Fukuchiyama data points.

This experiment enabled us to understand how well training the models on the large, consolidated crisis datasets mentioned in Section 4.2, which cover a diverse set geographical locations and a multitude of crisis events, would perform on the unseen data from flood events in Fukuchiyama.

To broaden the discussion of the efficacy the image analysis module has in mitigating information overload and enhancing situational awareness, we also qualitatively examine the informative utility the image models have in assisting crisis managers during a flood crisis event.

4.5.2 Qualitative Evaluation: Image Annotation Workshops

In addition to evaluating the performance of the models on these four classification tasks, the Urban Risk Lab conducted focus group research[5] with crisis managers and experts. This research included a series of 4 virtual workshops held with with various individuals and organizations with expertise and experience in crisis management. We held image annotation workshops in December 2021 with the following partners in Fukuchiyama:

- Yasuaki Yokoyama, Director of the Regional Disaster Management Research Center, Fukuchiyama Public University and Former Crisis Management Supervisor of Fukuchiyama City
- 3 Crisis Managers at an EOC in Fukuchiyama
- 5 Associates (including 1 Firefighter) of the Fire Department in Fukuchiyama

In January 2022, we also held a workshop with Richard Serino, who is a former

[5]We acknowledge Saeko Baird of the Urban Risk Lab at MIT who conducted the qualitative focus-group research in the form of interviews, workshops, and general interfacing with our partners in Fukuchiyama and in the US, and also provided the translations of the results from Japanese to English to enable the analysis of those results as it relates to the work presented in this thesis. We note that Saeko Baird has formal training in Human-Computer Interaction (HCI) research.

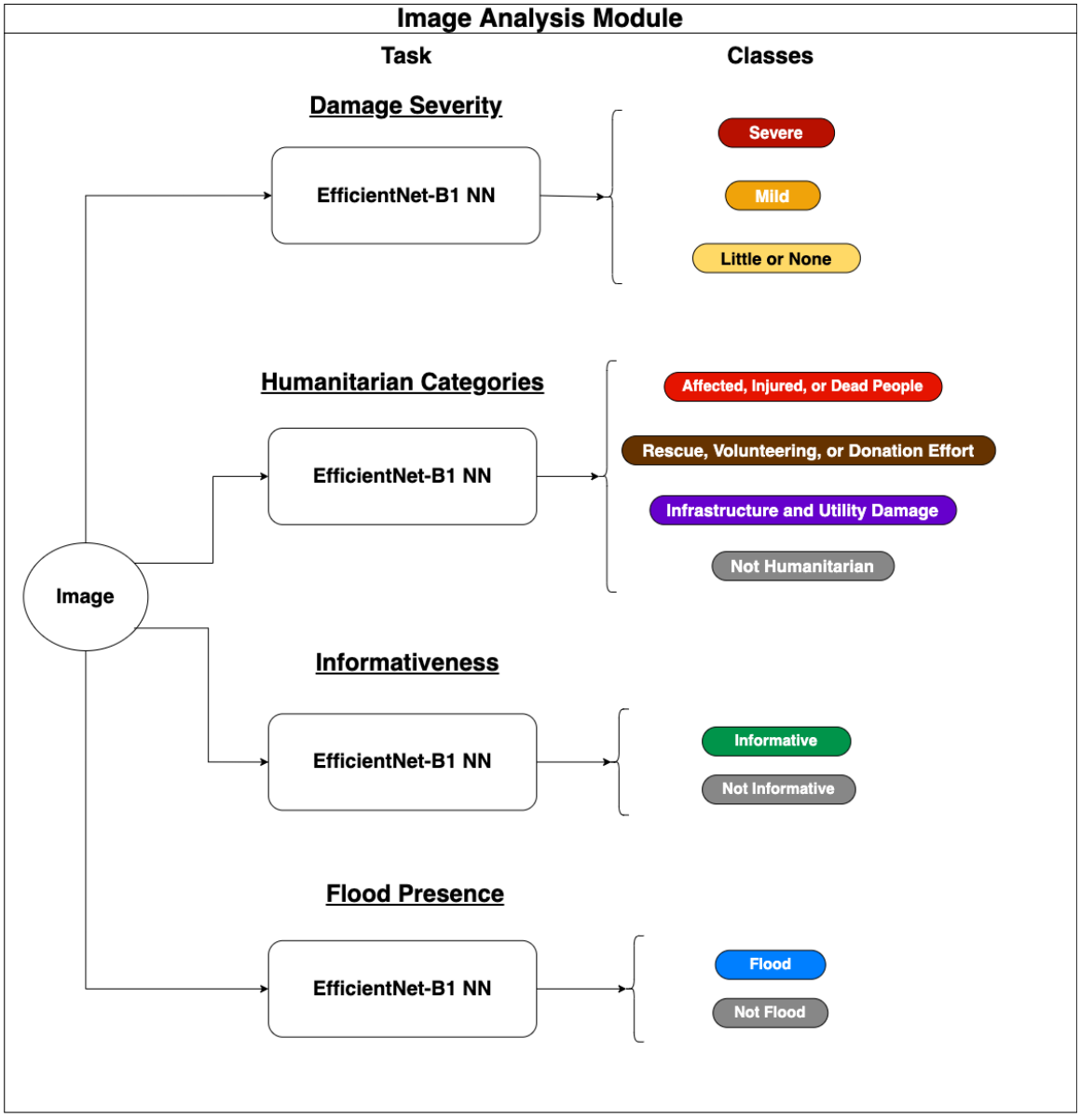
Deputy Administrator of the Federal Emergency Management Agency (FEMA) in the US and Senior Advisor to the Urban Risk Lab at MIT.

In holding workshops with domain experts in two different geographical contexts and across different departments or crisis management organizations, we aimed to investigate the insights gained from manual assessment of crisis reporting by crisis experts which are cross-contextual. These insights derived from domain expert manual assessment could validate the informative utility of the tasks discussed in this work, motivate their refinement, or inform the development of new prediction tasks and associated classes which could be used cross-contextually in a future work.

In these workshops, the domain experts were tasked to identify insights from an image that are useful for decision making and response during a crisis event. We made note of the insights they determined from their manual assessment of a set of 25 flood/sediment crisis images which were selected from the annotated Fukuchiyama images discussed in Section 4.2.3. A subset of the images that were selected were picked because they had disagreement between annotators, but still had a plurality label, from the annotation effort discussed in Section 4.2.3 and wrong predictions given to them by the CNN models discussed in Section 4.4 in contrast to the ground-truth plurality label. Additionally, the selected images contained various types of impact from the crisis event and varying levels of impact severity. The following types of impact are represented by the selected images: river flood, rock-fall, landslide, residential housing damage, blocked roads, agricultural land damage, submerged residential areas, and damaged infrastructure.

In our evaluation, we report qualitative summaries describing the insights derived from manual assessment by domain experts of the crisis report images and their expressed information needs. We then used these summaries to compare how the insights the Image Analysis Module aims to automatically provide to crisis managers compares to the insights derived from manual assessment by crisis experts and their expressed information needs. This comparison allowed us to qualitatively assess the utility of the Image Analysis Module in attaining situational awareness as it was devised in this work, and helped us in determining ways to iterate on the module in

the future to better accommodate the information needs of the crisis managers during a crisis event through new prediction tasks which align with their information needs. We expand on this in our discussion of future work in the Conclusion section. This qualitative evaluation is discussed in Section 4.7.3.



▲ Figure 4-1: Image Analysis Module Diagram

Having discussed the image ML methodology we used to showcase components of the framework presented in Section 1.1, in the next section, we discuss the utilities created from the analysis performed on image data within this study and how they

are open-sourced for reuse in future studies.

4.6 Implementation

We leverage the EfficientNet Python package [22], the PyTorch Python Neural Network library [58], scikit-learn [60], PIL [61], pandas [62], NumPy [63], matplotlib [64], seaborn [65], and statsmodels [66] to construct an open-sourced Python package for the development of our models and general utilities for conducting the various analyses contained in the Image Analysis Module.

4.6.1 URL Image Module Python Package

We created an open-source Python package to enable reproducible experiments such as those conduced in this work as well as the flexibility to be extended to leverage future state-of-the-art image classification models for potentially better performance on the tasks discussed in this work as well as tasks devised in future studies.

The model utilities of this package include image preprocessing, training, testing, and inference of EfficientNets and the VGG16 CNN image classification models pretrained on ImageNet for single-label image classification tasks. We provide utilities to train, test, and infer with these models on host machines using either CPU or GPU hardware. Other utilities of this package include methods for computing and plotting classification performance metrics of models.

This package also includes methods useful for dataset annotation efforts including methods for computing interannotator statistics such as Fleiss' Kappa coefficient, unanimous agreement percentage, and plurality agreement percentage. Relatedly, we provide utilities for constructing ground-truth single label datasets using the plurality label associated with an image or text data point.

Lastly, we note that this package contains utilities for plotting results such as those shown in Section 4.7 and interacting with datasets on a host machine's file system.

Reproducibility of Experiments and Model Metadata

For ensuring reproducibility of our experiments, we store various model metadata, including the random seed used for model training. For conducting inference or testing, we store a PyTorch file containing the trained weights from the training process described above, that can then be loaded into the EfficientNet or VGG16 model architectures. In addition to the trained model weights, during training we store dictionaries containing various model metadata useful for reproducibility as well as general information about the model. These dictionaries include:

- Hyperparameters used for training
- Mapping from string class name to integer index (e.g. "little_or_none" $\rightarrow$ 0) to give readable model output useful for plotting and analysis
- The specific settings of the host machine relevant for training (e.g. training on CPU or GPU, random seed used)
- Other metadata about training including the number of total and trainable model parameters and time to train (in seconds). Additionally, we store the average training loss, average dev loss, and the metric scores achieved on the train set and dev set, respectively, at each epoch.

To learn how to use the URL Image Module Python Package, please see here.

4.7 Results

In this section, we report the results of our experiments with the evaluation procedure for the Image Analysis module discussed in Section 4.5. We then discuss the implications of the results of the finetuned models on both the test splits discussed in Section 4.2.1 and the Fukuchiyama crisis images as well as the insights gained from image annotation workshops held with various crisis management experts.

4.7.1 Performance on Consolidated Crisis Image Datasets & Flood Presence Dataset Test Splits

▲ Table 4.6: Performance of Finetuned EfficientNet-B1 CNN Models on Consolidated Crisis Test Splits in [1] for the Damage Severity, Humanitarian, and Informativeness Tasks and the Flood Presence Test Split for the Flood Presence Task

Task	Balanced Accuracy	Weighted Precision	Weighted Recall	Weighted F1	Weighted F1 in [1]
Damage Severity	66.8%	75.0%	76.0%	75.4%	75.8%
Humanitarian Categories	63.6%	75.9%	76.7%	76.1%	76.5%
Informativeness	84.6%	85.7%	85.7%	85.6%	86.3%
Flood Presence	90.8%	92.1%	92.2%	92.1%	–

We note that the authors in [1] report a weighted F1 score of 75.8%, 76.5%, and 86.3% for the damage severity, humanitarian categories, and informativeness tasks for EfficientNet-B1 models trained using the consolidated crisis image training sets and dev sets (for validation) provided for those tasks in [1]. From Table 4.6, we observe that the EfficentNet-B1 models finetuned in this work perform at a slightly lower weighted F1 on the consolidated crisis image test sets for those tasks, 75.4%, 76.1%, and 85.6%, respectively. Since our finetuned models perform within a 1% difference by weighted F1 compared to the model performances reported in [1], we consider our models to perform comparably to the models reported in [1]. Lastly, the flood presence model achieves a weighted F1 of 92.1%, which is comparatively higher compared to the performance of the EfficientNet-B1 models for the other tasks, which we postulate to be due to the task being a binary classification task as well as being the most clear and objective task among the tasks, thus being a comparatively simpler task for the model to learn. We similarly observe this when assessing the model performances on the Fukuchiyama crisis images. Furthermore, we establish the weighted F1 score of 92.1% as the benchmark performance on the flood presence test set and make the corresponding Train/Dev/Test splits available for the crisis

informatics research community to use in future research.

4.7.2 Performance on Fukuchiyama Flood Crisis Images

▲ Table 4.7: Performance of Finetuned EfficientNet-B1 CNN Models on Unseen Labeled Fukuchiyama Flood Crisis Images

Task	Cohen's Kappa Coefficient	Balanced Accuracy	Weighted Precision	Weighted Recall	Weighted F1
Damage Severity	0.152	43.8%	47.2%	43.2%	43.2%
Humanitarian Categories	0.282	38.6%	64.7%	63.1%	62.4%
Informativeness	0.154	62.2%	84.9%	74.8%	78.7%
Flood Presence	0.647	82.3%	82.5%	82.5%	82.5%

When evaluating the performance of the models on the unseen Fukuchiyama dataset, we provide the aggregate weighted metrics as was done for the consolidated crisis image dataset test splits and the flood presence test split. We also provide the Cohen's Kappa coefficient. To have more granular insight into these aggregate results, we additionally provide confusion matrices showing the model's predicted classes against the ground-truth values. Lastly, we report the per-class performance by precision, recall, and F1 for each class in each task.

Aggregate Performance Metric Scores

The aggregate results, namely Cohen's Kappa coefficient, balanced accuracy, weighted precision, weighted recall, and weighted F1 for the finetuned EfficientNet-B1 models on the Fukuchiyama crisis images are reported in Table 4.7. Since there is a positive Cohen's Kappa coefficient for each of the classification tasks on the Fukuchiyama crisis, albeit to varying degrees, this indicates that the models perform comparatively better than the baseline classifier that predicts randomly according to the distribu-

tions of the ground-truth labels in the Fukuchiyama task-specific datasets. We observe that the Cohen's Kappa coefficient is significantly lower for the damage severity, humanitarian categories, and informativeness tasks at 0.152, 0.282, and 0.154, respectively, compared to the flood presence task at 0.647. This suggests that the models for the damage severity and informativeness tasks provide a relatively small improvement over a random classifier for their corresponding datasets as compared to the humanitarian categories model and far more so for the flood presence model, which provides the most improvement over the random classifier for its dataset.

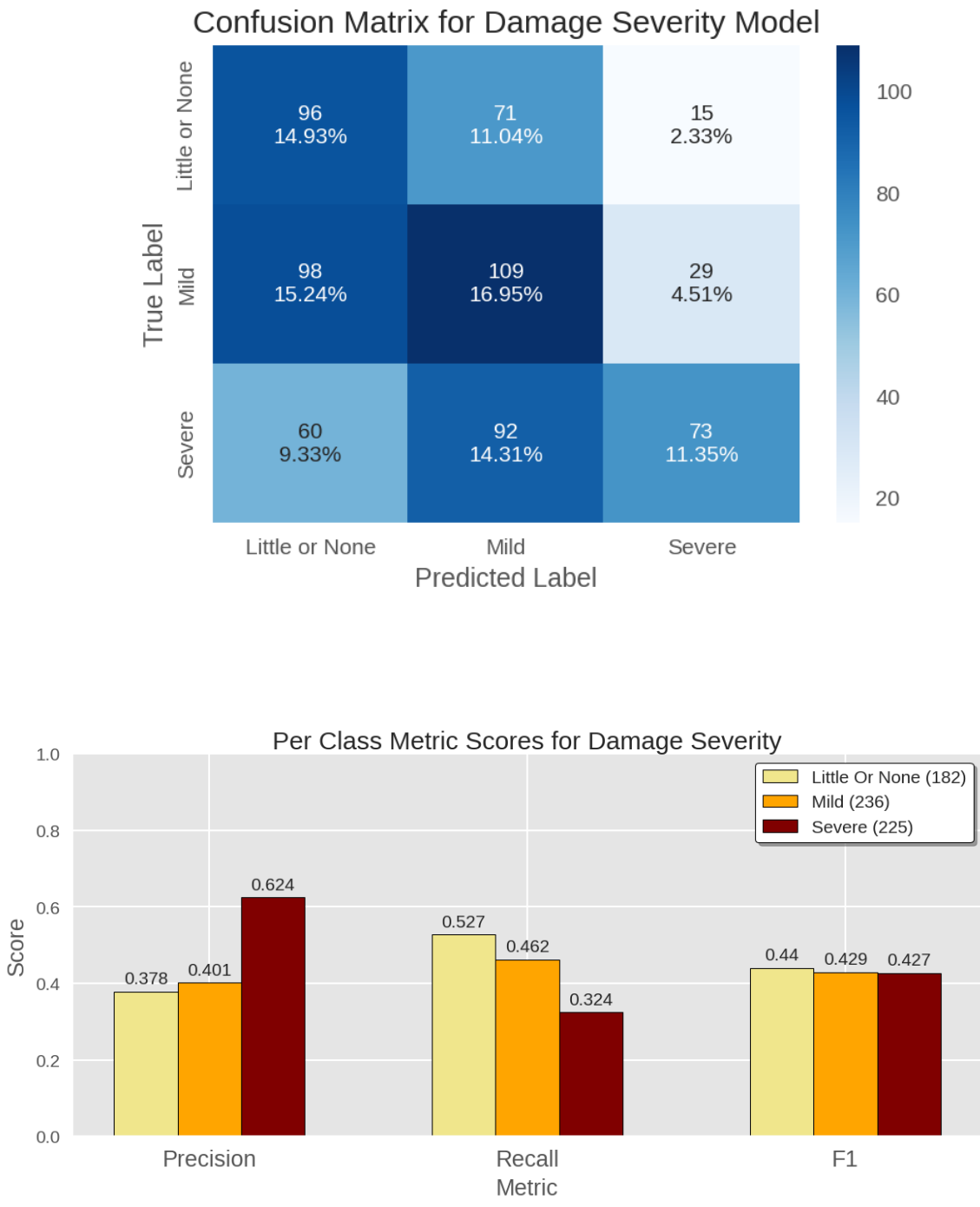
We observe across all of the weighted metrics for all tasks, the performances of the models on the Fukuchiyama data shown in Table 4.7 are less than the performances achieved on the consolidated crisis image test splits and flood presence test set reported in Table 4.6. This is most apparent for the damage severity model, which achieves a weighted F1 score of 43.2% on the Fukuchiyama data. Similar to the performance of the flood presence model on flood presence test split, the flood presence model performs relatively well on the Fukuchiyama flood presence task images achieving a weighted F1 of 82.5%.

Confusion Matrices and Per-Class Metric Scores

We report the confusion matrices and per-class model performance plots for the unseen Fukuchiyama crisis image data in Figures 4-2, 4-3, 4-4, and 4-5, for the damage severity, humanitarian categories, informativeness, and flood presence tasks, respectively. For each task, we report on the confusion matrix and per-class performance by the finetuned EfficientNet-B1 model for that task.

For the damage severity model, in Figure 4-2, we notice that when the model mispredicts the "Little or None" class it predicts "Mild" far more than "Severe". Relatedly, when the damage severity model mispredicts the "Mild" class, it far more often predicts "Little or None" than "Severe". Finally, when the model mispredicts the "Severe" class, it predicts "Mild" more than either "Little or None" or "Severe".

We see that the model achieves a precision of 0.624 on the "Severe" class that is comparatively higher than the precision for the "Mild" and "Little or None" classes

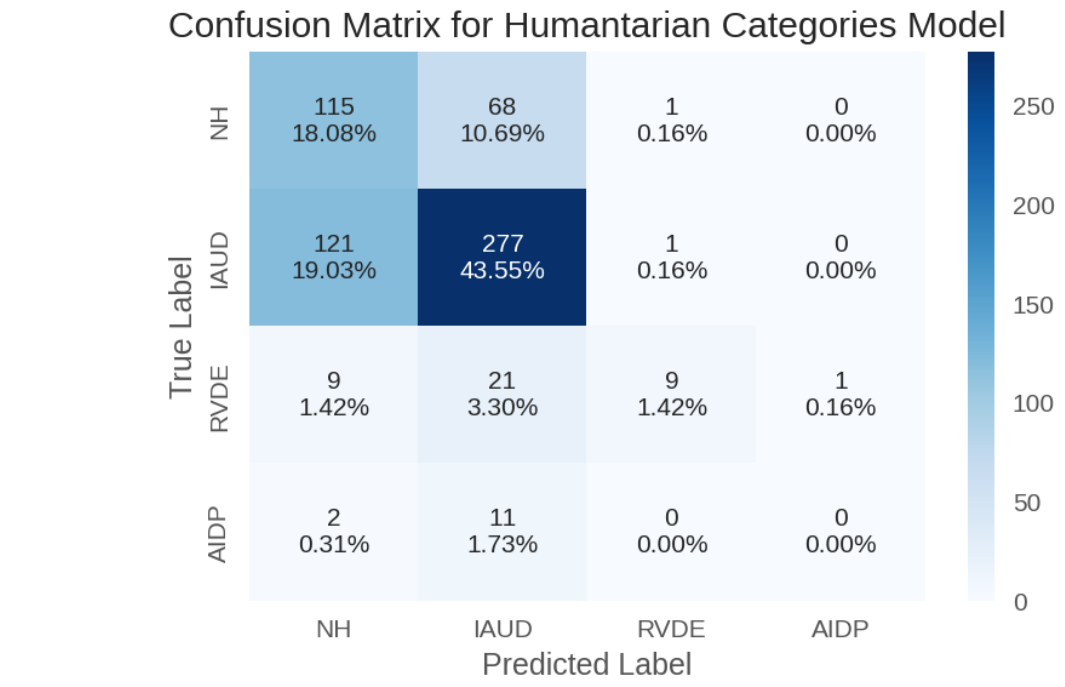


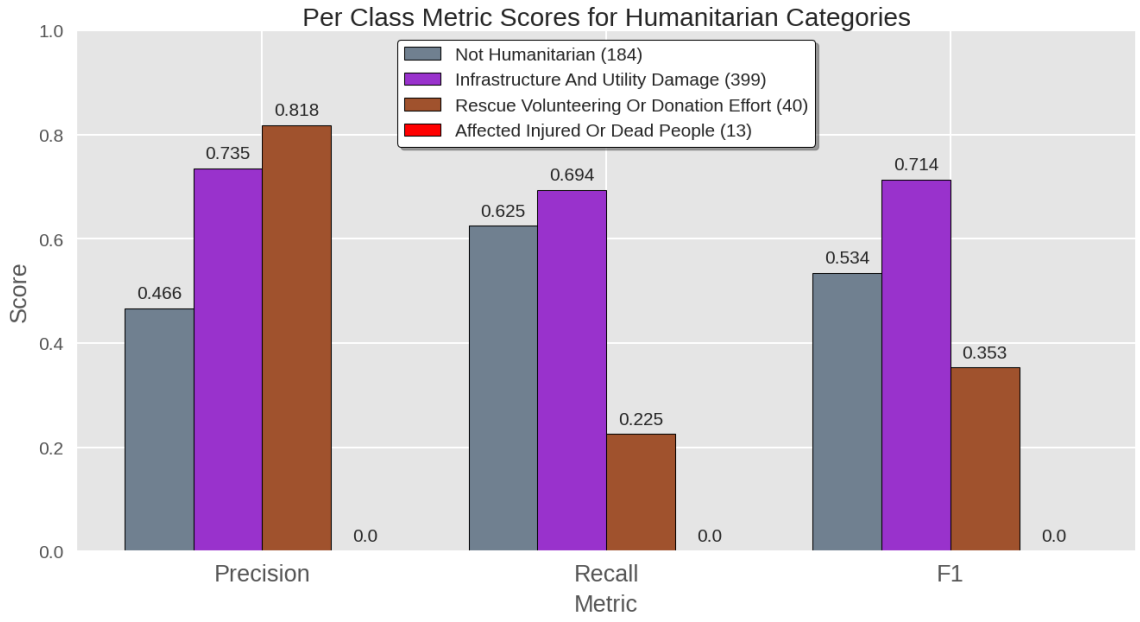
▲ Figure 4-2: Confusion Matrix and Per-Class Performance Metric Scores for **Damage Severity** Model on Fukuchiyama Flood Crisis Images

for which the model attains precision of 0.401 and 0.378, respectively. This indicates that when the model predicts the "Severe" class, 62.4% of those "Severe" predictions are actually labeled as "Severe", which is notably higher than the 40.1% of "Mild" predictions which are actually "Mild" and the 37.8% of "Little or None" predictions which are actually "Little or None". The per-class recall was highest for the "Little or None" class at 0.527, followed by 0.462 for "Mild", and 0.324 for "Severe". This means that of the data points which were actually labeled as "Little or None", 52.7% were correctly classified by the model. The model had the lowest recall for the "Severe" class, correctly classifying only 32.4% of the data points labeled as "Severe". Finally, we note that the F1 score for each class, that is, the harmonic mean of the precision and recall for that class, are very similar, namely 0.44, 0.429, and 0.427 for "Little or None", "Mild", and "Severe", respectively.

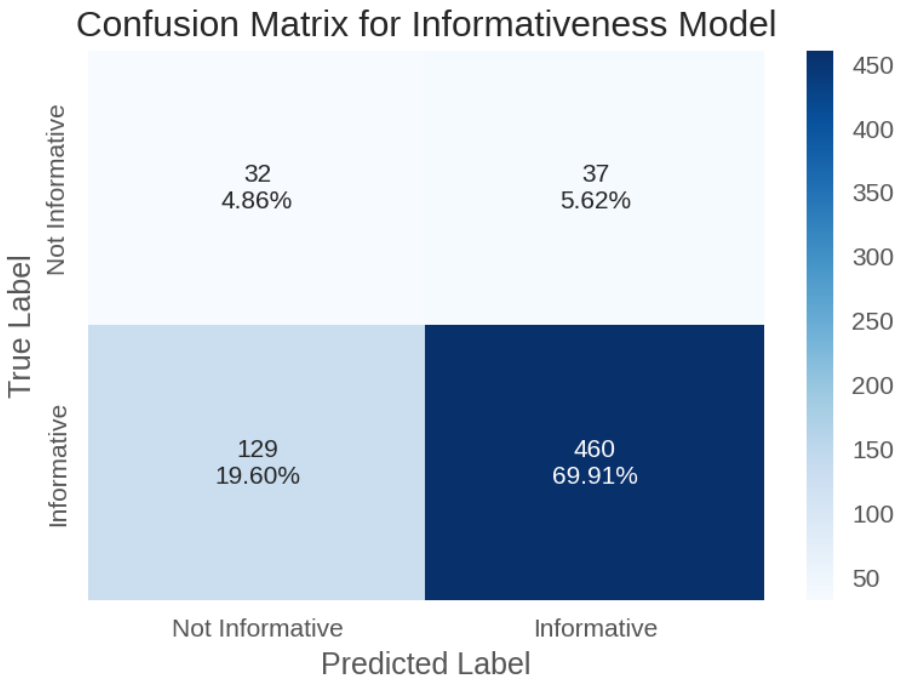
When discussing the humanitarian categories task classes, we refer to the "Not Humanitarian" class as NH, "Infrastructure And Utility Damage" class as IAUD, "Rescue, Volunteering, or Donation Effort" class as RVDE, and the "Affected, Injured, or Dead People" class as AIDP. From the humanitarian categories confusion matrix in Figure 4-3, we observe that for the AIDP, NH, and RVDE classes, when the model mispredicts these classes, it disproportionately predicts the IAUD class. When the model mispredicts the IAUD class, it almost exclusively predicts the NH class. The model attains a value of 0 for precision, recall, and F1 on the AIDP class. The model attains highest precision on the RVDE class at 0.818, followed by 0.735 on IAUD, and 0.466 for NH. For the IAUD and NH classes, the model achieves similar recall values of 0.694 and 0.625, respectively, with significantly lower recall of 0.225 for the RVDE class. Thus, by F1, we see that the model performs relatively best on the IAUD class at 0.714, followed by the NH class at 0.534, then RVDE at 0.353, and lastly AIDP at 0.

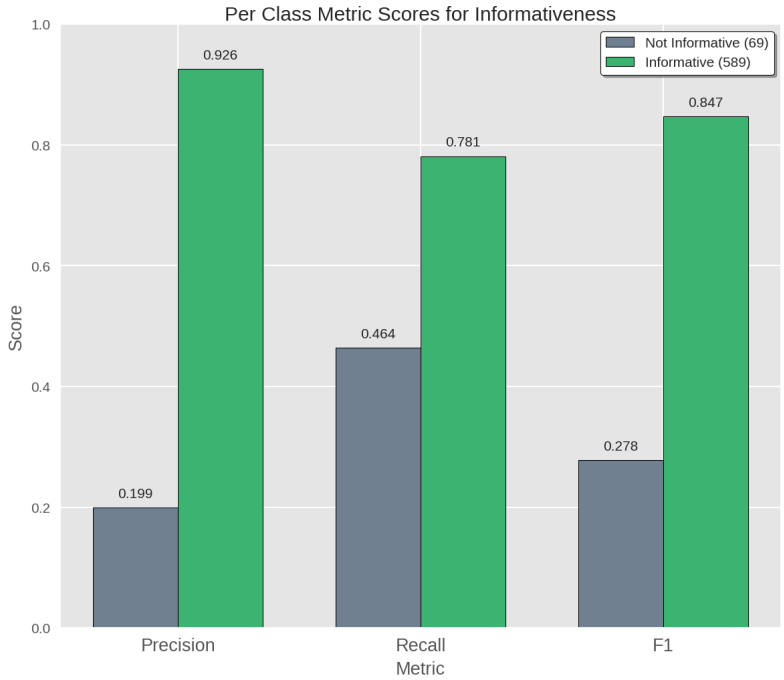
Figure 4-4 shows the confusion matrix and per-class metric scores for the informativeness model. By nature of the labeled Fukuchiyama crisis image data being almost exclusively related to crisis events or "normal day" photos, the "Not Informative" class is only 69 images as opposed to the 589 "Informative" photos. We note that





▲ Figure 4-3: Confusion Matrix and Per-Class Performance Metric Scores for **Humanitarian Categories** Model on Fukuchiyama Flood Crisis Images



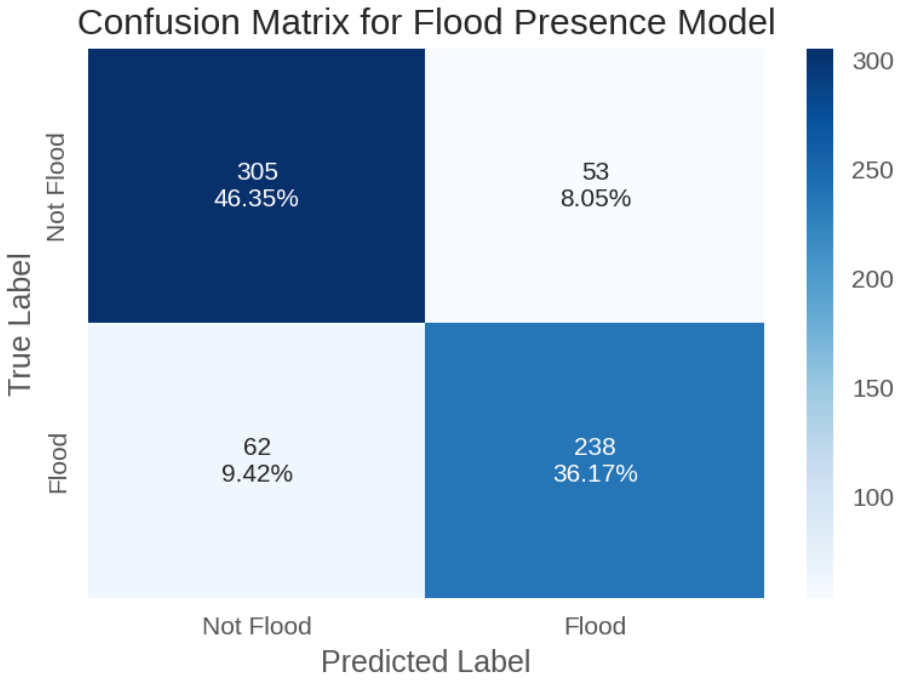


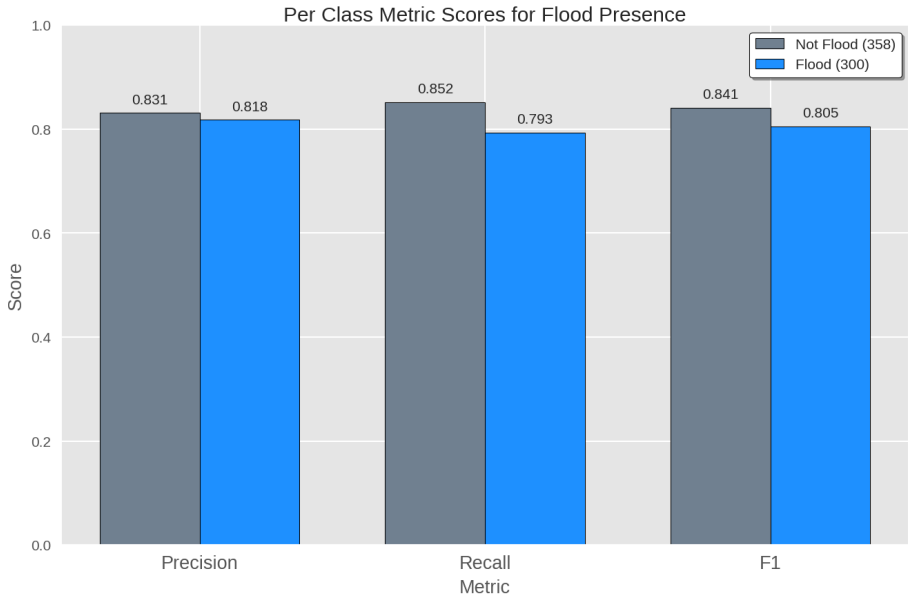
▲ Figure 4-4: Confusion Matrix and Per-Class Performance Metric Scores for **Informativeness** Model on Fukuchiyama Flood Crisis Images

in the original conception of the task in [1], the informativeness classifier is intended to be used for filtering noisy tweets which are completely unrelated to crisis events from relevant tweets, however as we report, it is not an adequate classifier for filtering images indicative of crisis impact and those of "normal-day" scenes, because, as we have learned, that is a different task altogether. We observe that the model correctly classifies most of the images labeled "Informative" with a recall score of 0.781. When classifying the images labeled as "Not Informative", the classifier classifies 53.6% of these images incorrectly as "Informative" and 46.4% of these images correctly as "Not Informative". Across all metrics, the model performs reasonably well on the "Informative" class, but significantly worse on the "Not Informative".

We report on the per metric scores for the flood presence model as shown in Figure 4-5. Unlike the models for damage severity, humanitarian categories, and informativeness, the flood presence model performs consistently well (less variation and higher values) by all metrics across all classes in the task, attaining metric scores above 0.793 across all metrics for both classes. More specifically, we note that the model performs reasonably well on the "Flood" class achieving precision, recall, and F1 scores of 0.818, 0.793, and 0.805, respectively. The model performs slightly better on the "Not Flood" class achieving precision, recall, and F1 scores of 0.831, 0.852, and 0.841, respectively.

Having quantified the performance of the models above with both aggregate and per-class performance metrics, we wished to understand the utility these models could have in assisting crisis managers in gaining situational awareness while also mitigating information overload. We thus held multiple image annotation workshops with various crisis managers. We report our qualitative analysis from those workshops as summaries of what we observed from their annotations and expressed information needs from crisis report images. We note that by incorporating this qualitative analysis into our framework, our goal is to broaden methods of evaluation and development of AI-augmented crisis information systems to iteratively incorporate the insights and feedback gained from engaging with crisis managers.





▲ Figure 4-5: Confusion Matrix and Per-Class Performance Metric Scores for **Flood Presence** Model on Fukuchiyama Flood Crisis Images

4.7.3 Qualitative Analysis from Image Annotation Workshops with Crisis Experts

In order to understand the informative utility of the insights gained from the automated classification by the ML models of the Image Analysis Module in relation to the insights gained from manual assessment by crisis managers when analyzing crisis reports, we held image annotation workshops with various domain experts and investigated the insights they gain from analyzing on-the-ground crisis imagery. The domain experts were asked to analyze an image and articulate various categorizations of the image based on their domain expertise and information needs. We report qualitative summaries of the insights the crisis managers derive from manual assessment in order to embed our findings from this focus-group research into the development of our ML methodology both within this study and in future iterations.

Manual Image Assessment Workshops

Below we summarize the main findings determined from the insights and feedback given by crisis experts in the image assessment workshops:[6]

- **Potential of Human Casualties in Crisis Imagery**: The possibility of human casualty by the crisis event is the primary concern of the crisis experts. When analyzing images, the crisis managers consistently identified markers for potential human casualties including submerged vehicles, collapsed buildings, rockfall and landslides with housing right below/above a cliff as having high priority. Relatedly, from our discussions, we learned that for an EOC, generally the cost associated with not investigating the potential for human casualties when there are human casualties is higher than the cost of investigating potential human casualties when there is none.

- **Presence of People in Crisis Imagery**: According to the crisis experts, deciding if there are people in the images is important. Images with human

[6]We acknowledge Saeko Baird of the Urban Risk Lab at MIT who assisted in the synthesis of these findings from the observations and discourse that occurred during the image annotation workshops she conducted as discussed in Section 4.5.2.

presence should be analyzed with high priority. The analysis of human presence should be specific such as determining if there are people laying down vs. people walking about/standing casually. These specific insights can indicate various levels of severity to trained emergency management personnel.

- **National Standards for Assessing Impact Severity**: The national standard of Japan for flood severity level and housing damage levels are used in Fukuchiyama, as well as other municipalities in Japan. In flood crisis, housing which is considered severely flooded/destroyed is defined as when "water reaches up to the first-floor ceiling," partially flooded/destroyed is defined as "water reaches 1 meter above first-floor level," and minor flooding/destruction is defined as "water reaches below floor level."

- **Insights of Broader Impact derived from Physical Markers in Images**: When analyzing images, the crisis managers noted that specific physical markers suggest potential broader impact to the area beyond what is depicted in the image. This included muddy water, which suggests the possibility of landslide outside of the image. A fallen power pole suggests the possibility of a power outage in the area. Finally, they noted that the road condition, i.e. whether or not a road is passable, indicates the possibility of emergency vehicle use and the possibility of isolated residential areas.

- **Insights Derived from both the Image and Contextual-Knowledge**: When analyzing the images, the crisis experts in Fukuchiyama noted that they tie in their contextual knowledge of the area where the image was taken. For example, if an image depicted flooding but the image was taken in an area that does not typically flood, this would heighten their concern for that area.

4.8 Discussion

There are likely multiple reasons why the model performance is comparatively lower for the Fukuchiyama data as compared to the test splits for the consolidated crisis

image datasets and the flood presence dataset. Although to varying degrees for each of the tasks, this may be in part due to concept drift between the data the models were trained on and the Fukuchiyama data which the models were evaluated on. Relatedly, the labeled Fukuchiyama data may have been of poorer data quality as suggested from the relatively low Fleiss' Kappa coefficients in Table 4.3 for the damage severity, humanitarian categories, and informativeness tasks.

The low Fleiss' Kappa coefficients for the damage severity (0.413), humanitarian categories (0.304), and informativeness (0.313) tasks for the images from Fukuchiyama suggest that components of the annotation process, i.e. the labeling guide and associated class descriptions for those tasks as used in this work should be improved to ensure better quality datasets for training and evaluating. Some ways to potentially improve the consistency in the labeling between annotators include converting the abstract descriptions/definitions of different classes for a task into checklists (reminiscent of the lists presented for the human risk task in Table 5.2 and Table 5.3) and understanding common disagreements between annotators from the annotation conducted in this work to identify points of clarification by resolutions from those discussions and adding those to the checklists to better equip annotators to handle potentially ambiguous data points by related examples. Additionally, the annotation procedure could be further improved by adding a larger representative supply of example images, including ambiguous data points, for each class to the labeling guide.

When analyzing the per-class performance for each of the models on the Fukuchiyama datasets, we observe that the humanitarian categories model performance varies greatly between the classes for the task. Namely, we observed that the AIDP class has scores of 0 across all metrics. From the training set distributions in Table 4.2, we observe that the AIDP class is only 6.12% of the entire training set for the humanitarian categories task. This is the smallest training set class proportion for any of the image classification tasks examined in this work, with the next lowest training set class proportions being those for the RVDE class in the humanitarian categories task and the "Mild" class of damage severity at 14.0% and 14.4%, respectively. This suggests that

the imbalance of the humanitarian categories training set impacts the performance of the humanitarian categories severely on the minority classes, especially the AIDP class, the class with the lowest proportion.

The informativeness model performs reasonably well on the "Informativeness" class and significantly worse on the "Not Informative" class. Most of the "Not Informative" images in the Fukuchiyama dataset are images of "normal day" photos, but from manual inspection of the training set used for the task in [1], we have observed that this dataset contains majorly unrelated content to crisis management such as illustrations, logos, banners, etc. and relatively few "normal day" photos, so the informativeness model would be better suited for the task of filtering tweets based on relevancy to crisis at all rather than applied on data which is mostly related to crisis such as the Fukuchiyama images or RiskMap images. Thus, we have determined that classifying "normal day"/"blue-sky" images vs. images showing impact from a crisis event would be better framed as a new task altogether.

The flood presence model achieved the highest weighted F1 score on both the flood presence test split (92.1%) and the Fukuchiyama crisis images (82.5%) among all of the image classification models across the tasks. Similarly, the flood presence model attains the highest Cohen's Kappa score of 0.647 on the Fukuchiyama images, which is much larger compared to the other models, suggesting that the model performs significantly better than the classifier that predicts at random on the Fukuchiyama flood presence image dataset. Finally, when considering per-class performance, the flood presence model is the only model that performs consistently well for all classes across all metrics. We consider this robust performance to be attributed to the task being binary as opposed to multiclass and to the task having classes which yield higher agreement between annotators as indicated by the relatively high Fleiss' Kappa score of 0.829 for the Fukuchiyama images. Although the flood presence model is able to accurately classify both the unseen data in the flood presence test split and the Fukuchiyama images, we sought to incorporate the domain expertise of crisis experts by understanding if the classes or outputs provided by the Image Analysis Module provide informative utility for attaining crisis awareness as compared to the insights

derived from manual assessment by crisis managers of crisis reports.

From the results of focus group research conducted by the Urban Risk Lab at MIT with various crisis experts, we have greater insight into the informative utility of the classes associated with the image classification tasks presented in this work. More specifically, from their insights and feedback, we have determined that the tasks as they are presented in this work have classes with interpretations that are either too vague and subjective (damage severity, humanitarian categories, and informativeness) or too simplistic (flood presence) to be useful for them in gaining situational awareness about an unfolding crisis event.

Additionally, the task of humanitarian categories has the "Rescue, Volunteering, or Donation Effort" category, which has insights for the recovery phase of a crisis event rather than the emergency phase. Since our ML methodology aims to assist crisis managers during the emergency phase of a crisis event, such classes should be revised or replaced with classes which have insights directly for the emergency phase. Although the flood presence task has classes with interpretations which are too simple for attaining situational awareness, we note that the relatively high performance, high consistency between independent annotators, and clarity in the interpretation of the classes associated with the flood presence task sets precedent for task creation and model performance for the future tasks developed from the insights and feedback received from the workshops discussed in this work and future workshops. The insights and feedback provided by the domain experts enabled us to determine how the Image Analysis Module we have developed in this work is limited in helping to gain insights about the unfolding crisis event. Where our ML methodology falls short in meeting their information needs, their feedback will assist in developing new classification tasks which would be informative enough to assist them during a crisis event and clear enough to yield more consistent labels between annotators, ensuring better quality data to train and evaluate models. The development of new image prediction tasks and associated models will be conducted in a future work. However, we were able to apply some of these insights to inform the ML methodology of the Text Analysis Module. We discuss the Text Analysis Module in the next chapter.

Chapter 5

Text Analysis Module

The Text Analysis Module aims to provide accurate and efficient classifications of crisis reports using the text modality that is often present in the reports. Another aim was to incorporate the insights we gained from the results of our qualitative analysis on the Image Analysis Module as discussed in Section 4.7.3 that were transferable between the data modalities, i.e. the importance of identifying potential for human casualty or risk to humans. We did this to exemplify our framework's intention of producing iteratively developed ML methodologies and AI systems to enhance crisis awareness and response using insights gained from crisis managers.

To incorporate the insights of the crisis managers into the design and development of a new text classification model, we first created a classification task and associated classes that align with the expressed information needs of crisis managers during a crisis event. Then, we selected a performance evaluation metric that aligns with priorities of the crisis managers for that task, finally developing a model that is evaluated using the selected performance metric.

In the process of conducting this exercise, we performed various classification experiments, experimenting with various text featurizations, classical machine learning algorithms, and importantly, we deliberated on the selection of a performance evaluation metric based on our findings from the qualitative analysis of the image annotation workshops.

We note that since this study focused exclusively on Japanese crisis text, we

constructed a preprocessing pipeline that uses open-source Japanese tokenizers, stopwords, and a lemmatizer to preprocess the Japanese text. Additionally, we investigated the use of text embeddings of the Japanese crisis text that are created by applying CLS pooling, a process which creates a contextualized numerical embedding of inputted text, using a pretrained Japanese Masked Language Modeling (MLM) BERT model in both our supervised and unsupervised learning experiments.

Finally, we conclude the development of this ML module on an exploratory note, devising a pipeline that evaluates a combination of text featurizations, dimensionality reduction techniques, and clustering algorithms to provide intuitive groupings of text to help inform the development of text classification tasks in future work.

In our evaluation of our text classification experiments, we perform quantitative evaluation, assessing the performance of the model for the task based on the determined evaluation metric mentioned above in addition to other metrics, e.g. per-class performance metrics. For our unsupervised experiments, we include both quantitative and qualitative evaluation. Using the Within-Cluster Sum of Squares (WCSS) metric, we determine a set of optimal clustering pipeline configurations and their corresponding optimal number of clusters to use for further investigation. We assess qualitatively by investigating the resulting clusters and determining for each cluster, whether or not the representative documents within that cluster have a cohesive, interpretable label, and if they do, what that label is.

5.1 Fukuchiyama Firefighter Flood Text Reports Dataset

Our partners in Fukuchiyama City (FC) compiled a dataset of 716 text transcripts of radio communications from on-the-ground firefighters, which occurred in real-time during past flood events. The data comes from the following past flood events in FC:

- Typhoon Manyi in 2013
- Heavy Rain Event in August 2014
- Typhoon Lan in 2017
- Heavy Rain Event in July 2018

The following procedure was followed in real-time by the Fire Department (FD) in FC for creating these text transcripts during these events:

1. Both city and shobodan (volunteer) firefighters reported updates via radio to the Fire Department Headquarters (FDHQ) during the flood/typhoon event.

2. FDHQ operator(s) which received these transmissions wrote a transcript (in Japanese) of the radioed report.

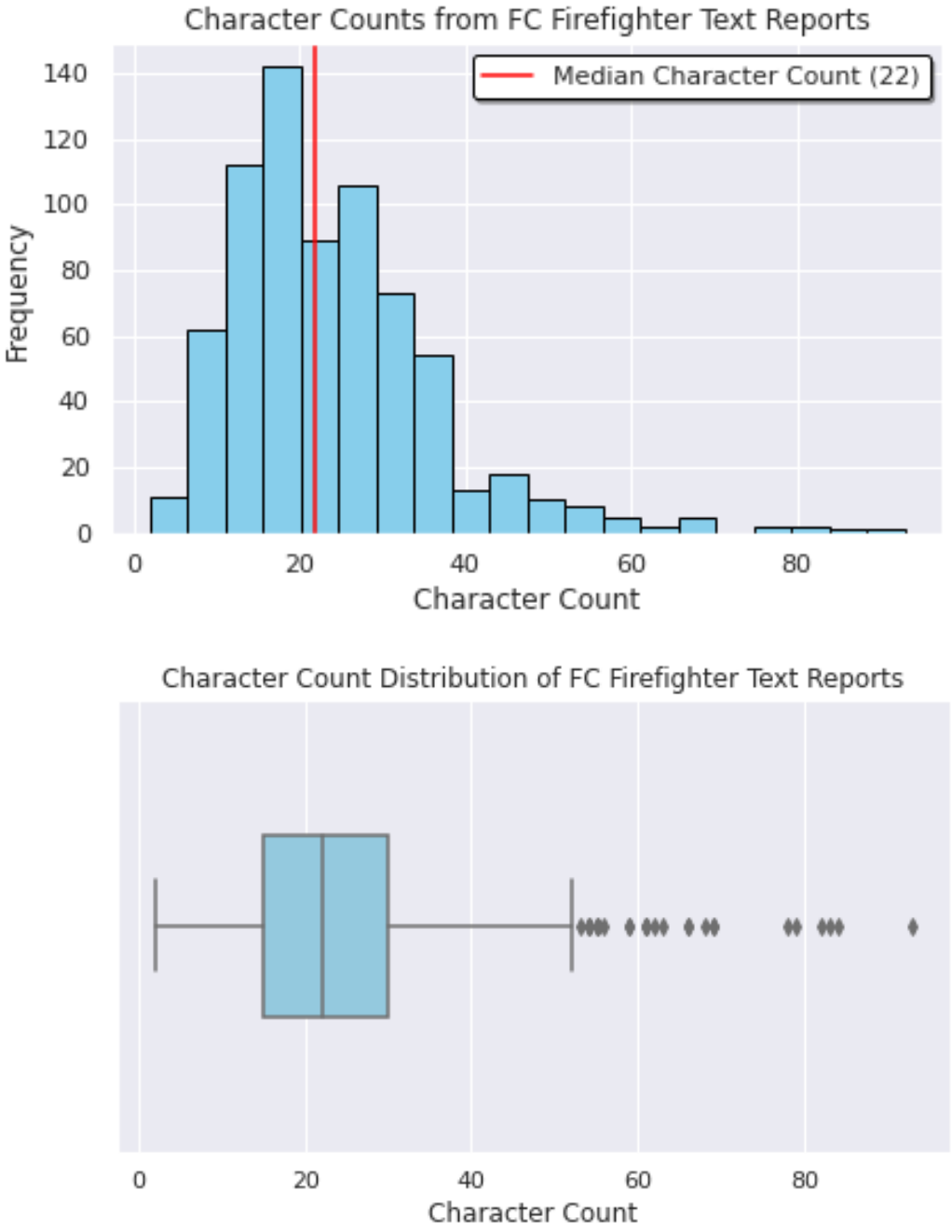
3. The operator(s) would subsequently categorize the report as whether or not the report was indicative of **Human Risk** and sometimes also one category of a variety of humanitarian categories.

From Figure 5-1, we see that most of the Fukuchiyama firefighter text reports are between only a few characters in length to about 50 characters. We note that the distribution has slight right-skewness. The median report character length has 22 characters. There are a few outliers which are more than 50 characters, but less than 100 characters.

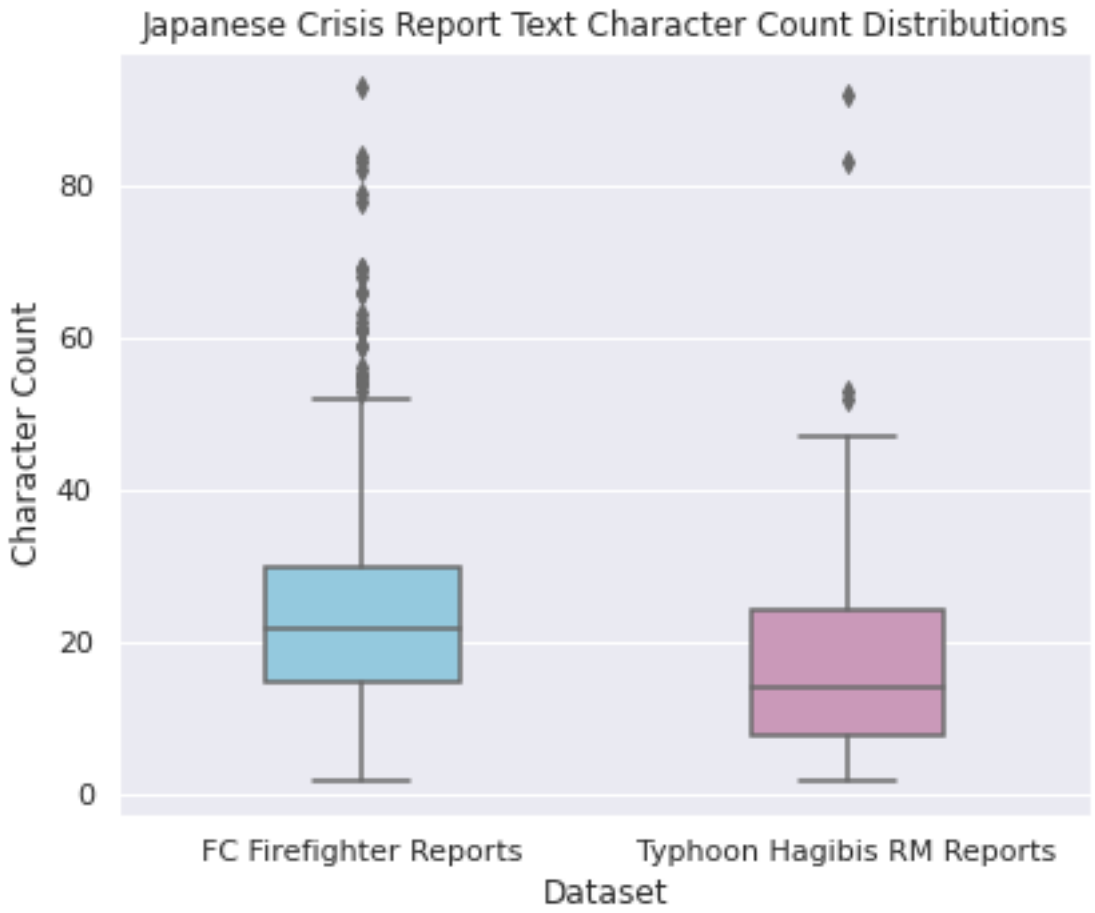
To understand how these reports compare to other Japanese crisis reports, we compare the FC firefighter reports character length distribution against another Japanese crisis text report corpus, the text of Tokyo crisis reports received by RiskMap (RM) during Typhoon Hagibis in 2019. We note that there are 68 reports in total for the Typhoon Hagibis RM reports dataset. We show the character length distribution of the reports from each of these datasets in Figure 5-2.

We observe that there is a difference between the distributions, namely that Typhoon Hagibis reports are more right-skewed and have less outliers than the FC Firefighter reports. Additionally, the Typhoon Hagibis RM reports have a median character length of 14 and most of the RM reports fall between a few characters to about 50 characters, similar to the FC Firefighter reports.

According to Twitter research on Japanese tweets, Japanese tweets have a mode of 15 characters, with the character length distribution of Japanese tweets also exhibiting a right-skew similar to the FC firefighter reports and the Typhoon Hagibis RM reports



▲ Figure 5-1: Character Count Distribution of Fukuchiyama City (FC) Firefighter Text Reports



▲ Figure 5-2: Character Count Distributions of Japanese Crisis Text Reports

[67]. The authors in [67] note that English tweets have a mode of 34 characters, stating that, "This is because in languages like Japanese, Korean, and Chinese you can convey about double the amount of information in one character as you can in many other languages, like English, Spanish, Portuguese, or French.". These comparisons suggest that the Fukuchiyama firefighter crisis text reports we focus on in our text analysis are brief crisis updates of similar length to that of Japanese crisis reports on RiskMap and Japanese tweets.

5.2 Human Risk Text Classification Task

As mentioned in Section 5.1, FDHQ operators would label reports by whether or not they were indicative of human risk as well as one of a variety humanitarian categories. The human risk labels were binary (i.e. Yes/No), and sometimes the reports were also classified into one of 108 unique humanitarian categories. Additionally, the full FC dataset consists of 716 text reports, 715 of which were labeled with a human risk label, while only 584 were labeled with a humanitarian category. These characteristics of the dataset are summarized in Table 5.1.

▲ Table 5.1: Characteristics of the FC Firefighter Flood Text Report Dataset

Total Reports	**Reports Labeled for Human Risk**	**Reports Labeled for EOC Humanitarian Categories**	**Unique EOC Humanitarian Categories**
716	715	584	108

Since all but one of the text reports were given a human risk label and considering the lessons we learned from the image annotation workshops with crisis experts, we have determined the assessment of the potential of human casualty to be an important information need of crisis managers during a crisis event. Therefore we focus on forming the human risk text classification task in this work. We note, however, that the humanitarian category labels can be used together with the categories that are uncovered from the clustering experiments described in Section 5.5 to create

classification tasks in a future work.

5.2.1 Human Risk Task Formulation

We determined the classes associated with the human risk classification task by inspecting the labeled data and observing qualities associated with reports in each category, **Human Risk** and **No Human Risk**, respectively. As stated in Section 1.2.4, the human risk text classification task determines whether or not a crisis text report indicates if there are people in need of rescue from a crisis. This includes people being unable to evacuate due to physical disability (such as unable to use stairs), surrounding conditions (such as being trapped in a submerged car), and/or being in need of life-saving emergency medical care. For clarity, we choose to present the descriptors of each of the classes as lists instead of text descriptions, such that if a report has one or more of the qualities listed for the class, it should be considered as being part of that class.[1] A report should be classified as *Human Risk* if it describes any of the information shown in Table 5.2. Alternatively, a report should be classified as *No Human Risk* if it describes any of the information shown in Table 5.3.

Using the formulated human risk text classification above, we describe our text classification experiments and evaluation procedure for the task in Section 5.4. In order to extract features from the raw Japanese crisis text reports to use as inputs to models used in our classification and clustering experiments, we now describe our Japanese text preprocessing and text featurization pipeline.

5.3 Text Preprocessing and Featurization

In our text analysis, we devise a preprocessing and featurization pipeline which yields four different featurizations of the raw Japanese text, namely BOW based on unigrams, BOW based on bigrams, TF-IDF based on unigrams, and pretrained Japanese MLM BERT with CLS Pooling embeddings. Depending on the featurization, different

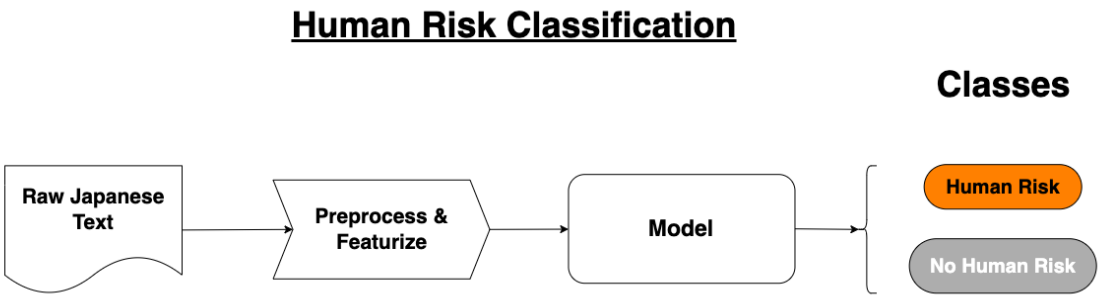
[1] We acknowledge Saeko Baird of the Urban Risk Lab at MIT who determined these class definitions from examining the original Japanese reports.

▲ Table 5.2: "Human Risk" Class Descriptors

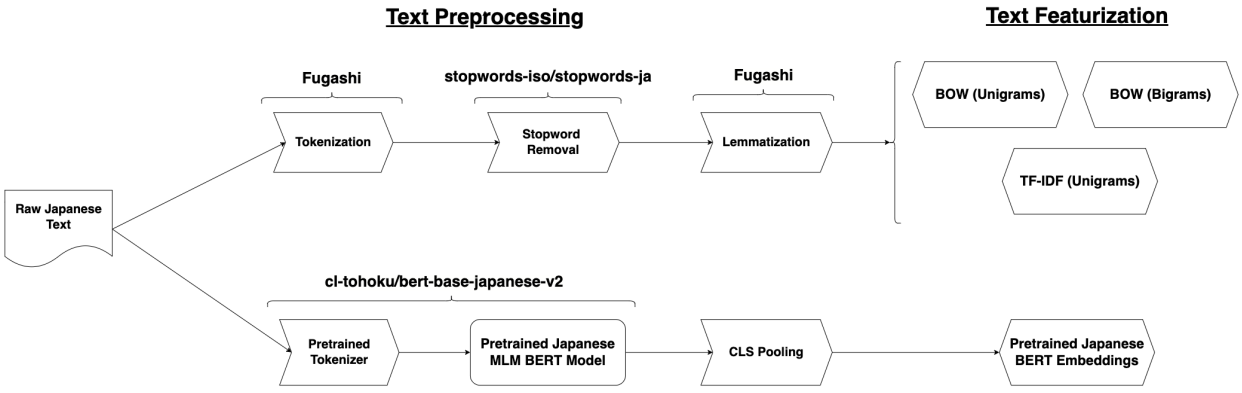
Rescue being requested (to the FD)
Evacuation support being requested (to the FD)
Human missed the chance to evacuate from their own house, at work, shopping center, etc.
Vulnerable population (elderly, disabled, small children) being left in the house in the flooding area
Water rising inside the house above the floor (human inside)
Water current is fast inside the house and hard to move upstairs (human inside)
Sediment flowing into the house (human inside)
Human being trapped in elevator
Human being trapped in a submerged car
Human being trapped in a car which is not submerged yet
Human being washed away in a river
Rescue team dispatched
Rescue team in activity (such as helping evacuation, rescuing, etc.)
Rescue activity completed
Landslide occurrence on the highway - possible vehicle being involved

▲ Table 5.3: "No Human Risk" Class Descriptors

Dam discharge
Meteorological information
River water level information
Weather alert
Road Closure
Road Flood Risk (not flooded yet)
Area Flood Risk (not flooded yet)



▲ Figure 5-3: Human Risk Text Classification



▲ Figure 5-4: Japanese Crisis Text Preprocessing and Featurization Pipeline

preprocessing steps are applied to the text prior to the featurization. After preprocessing and featurization, the text features can be used as input to the ML models we investigate in this module. The preprocessing and featurization pipeline on Japanese crisis text is shown in Figure 5-4.

5.3.1 BOW and TF-IDF Preprocessing

In this section, we discuss the preprocessing and featurization pipeline for producing BOW and TF-IDF text features. First, we provide the raw text string of a report as input to the fugashi[2] [68] tokenizer. fugashi is a Python wrapper of the open-source Japanese text tokenizer and morphological analysis tool, MeCab. fugashi requires a Japanese dictionary; we use Unidic Lite,[3] which is about 250MB of disk space. Provided with the raw text report string, the fugashi tokenizer converts this string into a list of Japanese word tokens.

We then make use of an open-source Japanese stop words list[4] (134 stopwords) to remove any stopword tokens that are present in the report. We remove stopwords as they do not contribute any meaningful information in the report text and would thus add noise to the text features when used as model input.

Finally, we lemmatize the remaining word tokens using fugashi's lemmatization tagger, i.e. the word tokens are converted from their inflected form into their dictionary form, or lemma. We use these processed word tokens to construct BOW and TF-IDF text features, or feature vectors of the text reports.

5.3.2 BOW and TF-IDF Featurization

The BOW and TF-IDF features are extracted using a set of reports which we refer to as the text corpus, and we refer to the reports within the corpus as documents. First, we select the n-gram representation we would like to have for the tokens yielded from the preprocessing. In this work, we look at both unigram and bigram BOW

[2]fugashi Python Package: https://pypi.org/project/fugashi/
[3]Unidic Lite Python Package: https://pypi.org/project/unidic-lite/
[4]Link to Japanese Stopwords List

features and unigram TF-IDF features. Once the n-gram representation is selected, the preprocessed list of word tokens is converted into the n-gram representation for the document, yielding a new list of n-gram tokens. Using this n-gram token list, we can then determine the vocabulary of the corpus, V, as being all unique n-grams across all documents of the corpus.

For both BOW and TF-IDF, we create a feature representation of each document in the corpus, by constructing a vector of size $|V|$, where each entry in the vector corresponds to a unique n-gram in the vocabulary. For BOW, the entries of this vector are simply the frequency of n-grams present in that document, with the rest of the entries being zero for n-grams in V which are not present in the document. The entries in the TF-IDF feature vector, correspond to the TF-IDF scores for the n-grams present in that document, all other entries are zero. The TF-IDF score computation is shown in Appendix Section B.3. The TF-IDF score extracts a score of importance based an n-gram's occurrences in the document (term-frequency) and the inverse of the presence of the word across all documents in the corpus (inverse document frequency) [69]. TF-IDF has the advantage over BOW features for extracting n-gram importance because it weights n-grams which appear in many documents in the corpus as less important and less informative than words which appear in few documents [60].

These featurizations are initially "fitted" using a specific corpus of documents. Thus, if we transform a document that was not in the original corpus using these featurizations, any Out-of-Vocabulary (OOV) words which were not present in the original corpus are ignored. This note is important when considering our discussion of the classification experiments in Section 5.4, in which the featurizations are fitted on a corpus of training documents, and test documents are transformed using those fitted featurizations, as to not have any data leakage, i.e. knowledge of the unseen test data seeping into the training data. Although these n-gram-based featurizations have the benefit of being language-agnostic, we note that they have the limitations of being high-dimensional and sparse in which most entries of the feature vector are zero, an inability to model long-range dependencies between tokens in the context of a

document, and a severely limited ability to capture token similarity and understanding of a language [70]. Therefore, we investigate a featurization strategy that yields dense, contextualized document representations specific to Japanese text documents. This strategy uses a pretrained Japanese MLM BERT model and the CLS pooling technique.

5.3.3 Pretrained Japanese MLM BERT with CLS Pooling Text Embeddings

Researchers at Tohoku University pretrained a Japanese masked-language model[5] which has the same architecture as the original BERT base model [44], that is, 12 layers each containing 12 attention heads, and hidden states of 768 dimensions. To train this model, the researchers used a training corpus of $\sim$ 30 million sentences from the Japanese version of Wikipedia. They tokenize text using MeCab (with fugashi) and the Unidic Lite dictionary. The tokens are split into subword tokens using the WordPiece algorithm [71]. This results in a vocabulary of 32768 tokens, and any OOV tokens are represented using a special "[UNK]" token. The model was trained with the masked-language modeling objective, in which the model is trained to predict randomly masked tokens (replaced with a special "[MASK]" token) in the input. The model learns to predict the most probable token for the masked token by using the information contained in the whole sentence, i.e. the context of the original token. The model represents each token in the input as an embedding vector. The authors specifically employ whole word masking in which all subword tokens that are part of a single word are masked at once.

To create dense, contextualized document-level embeddings for the Japanese crisis reports, we first pass the raw report string through the pretrained tokenizer based on WordPiece discussed above. The tokenized input then gets passed to the pretrained Japanese MLM BERT model. Input sequences to BERT models are prepended with a special "[CLS]" token. The authors in [44] note that the final hidden state corre-

[5]Pretrained Japanese MLM BERT Model: https://huggingface.co/cl-tohoku/bert-base-japanese-v2

sponding to this token is a contextualized embedding, or representation, of the input document for classification tasks, which is why "CLS" is short for classification. Extracting the contextualized embedding for a document with this technique is referred to as CLS pooling. For brevity, we refer to this featurization as BERT embeddings, or simply BERT in the appropriate tables.

Using the pipeline presented in Figure 5-4, we create various featurizations of the text to use as input in our text classification and clustering experiments.

5.4 Text Classification Experiments

As mentioned in Section 5.2, we focus our text classification experiments on the human risk task. We aimed to develop a task that better met the information needs of crisis managers during a crisis event. For this task, we did this by using labels our crisis management partners in Fukuchiyama provided to us directly. In addition, we aimed to develop this model using a performance metric which better aligned with the priorities of the crisis managers as it pertains to this task. In our determination of a performance metric, we also took into consideration the issue of class imbalance.

5.4.1 Determination of the Performance Evaluation Metric

▲ Table 5.4: Distribution of Human Risk Labels across entire FC Text Report Corpus

No Human Risk	Human Risk	Total
620 (86.7%)	95 (13.3%)	715 (100%)

From Table 5.4, we observe that there exists class imbalance across the labels for this task, where there are disproportionately more "No Human Risk" data points than "Human Risk" data points. If we used accuracy as the performance metric, the useless classifier which always predicts the "No Human Risk" class would yield an accuracy of 86.7%. Furthermore, from the results of our image annotation workshops, we have come to understand that in regards to assessing the potential for human casualties,

the cost associated with not investigating the potential for human casualties when there are human casualties (False Negative) is considered higher than the cost of investigating potential human casualties when there are none (False Positive). From this, we consider the performance of the model on the "Human Risk" class to be paramount. Accuracy does not tell us how well the model performs on the "Human Risk" class, so the selected evaluation metric should focus on the performance on this class, the class of interest. Such metrics include precision and recall.

Ideally, we would like to have both high precision (few false positives) and high recall (few false negatives), the F1 score combines these metrics into a single value by using the harmonic mean of precision and recall. The F1 score treats both precision and recall as equally important. However, as mentioned above, we have determined that the cost of a false negative is higher than the cost of a false positive for the human risk task. Thus we elect to use an F-β score (from which the F1 score is derived) as the evaluation metric for the task, noting that the F-β score considers recall as β times as important as precision [72]. The F-β score is computed as follows:

$$\text{F-}\beta = \frac{(1+\beta^2) \cdot \text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}}$$

In this work, we select $\beta = 2$, where recall is considered twice as important as precision for assessing the performance of the classifier. We pick $\beta = 2$, simply because we understand it to be a popular choice when using F-β where recall is more important than precision, but we note that tuning β should be investigated further in a future work, i.e. assessing if $\beta = 2$ is too lenient or too strict when considering the relative cost of a false negative to a false positive. This is the F2 score, which we use in our classification experiments described below.

5.4.2 Data Splits

To conduct the human risk classification experiments, we first split the full human risk dataset of 715 labeled reports into train and test splits in percentages of 80%/20%. We preserve the data imbalance in the splits by using stratified splitting. The resulting

stratified splits and class support can be seen in Table 5.5.

▲ Table 5.5: Human Risk Text Classification randomized, non-overlapping, stratified Train/Test Splits

Human Risk Text Classification Stratified Train/Test Splits			
Class Name	**Train**	**Test**	**Total**
No Human Risk	496 (86.7%)	124 (86.7%)	620 (86.7%)
Human Risk	76 (13.3%)	19 (13.3%)	95 (13.3%)
Total	572 (100%)	143 (100%)	715 (100%)

5.4.3 Nested Cross Validation for Algorithm Selection

When performing model selection and subsequently model evaluation, using random divisions of the full dataset into Train/Dev/Test splits such as those used in Section 4.2 is considered the best approach when you have a large amount of data for the task you are trying to predict [73]. When there is insufficient data to utilize Train/Dev/Test splits for model selection and model evaluation, such as the case for the human risk task, K-fold Cross Validation (CV) is useful for performing model selection or computing a reliable measure of a model's generalization performance. With K-fold CV, we randomly split the data into K non-overlapping folds, fitting (i.e. training) a model on $K-1$ of the folds and predicting on the remaining fold, the test fold, doing this for each of the K folds. K-fold CV can be used for hyperparameter tuning (such as with grid search) or generalization performance estimation by computing the average metric score and standard deviation across test folds [73]. However, if we used the same CV procedure for both performing hyperparameter optimization and estimating a model's generalization performance, the estimated generalization performance may be a biased, overly optimistic estimate [74]. For the human risk classification task, we were interested in investigating multiple ML algorithms, each with their own set of tunable hyperparameters. We aimed to determine which algorithm paired with a corresponding hyperparameter grid, i.e. a set of unique hy-

perparameter combinations, had the best estimated generalization performance. To determine this we used the Nested CV procedure [74].

To mitigate the bias in the estimate that is made when performing hyperparameter optimization and estimating generalization performance in K-fold CV, Nested CV nests hyperparameter optimization (a K-fold CV) within the generalization performance estimation procedure (another K-fold CV, not necessarily the same K used for the other CV). During Nested CV, for a specified algorithm, e.g. Logistic Regression, and a corresponding hyperparameter grid, a K-fold CV procedure is performed to assess the generalization performance of the ML algorithm along with the hyperparameter search procedure (e.g. grid search using the hyperparameter grid). This is the outer loop of Nested CV. On each iteration of the outer loop, one fold will be treated as the test set and the other $K-1$ folds are treated together as the train set. The inner loop of Nested CV uses the train set provided by the outer fold for a separate K-fold CV procedure using grid search to enable hyperparameter optimization. Using the results of grid search on the validation set of each iteration of the inner loop, the hyperparameter combination which yields the highest mean performance score across the validation sets of the inner loop K-fold CV is passed to the outer loop. Using this hyperparameter combination found from the inner loop, the algorithm is fitted on the train set in the current iteration of the outer loop and a performance score is computed on the test set. At the end of this procedure, the performance scores computed on the test sets of the outer loop K-fold CV are used to estimate the generalization performance of the algorithm and hyperparameter search procedure. Pseudocode of the Nested CV algorithm is shown in Algorithm 1. We note that Nested CV has the limitation that it is substantially more computationally expensive to perform compared to K-fold CV, but yields a less biased estimate of generalization performance when using CV to estimate generalization performance and perform hyperparameter tuning [75]. In our low-data setting, we consider Nested CV reasonable to use.

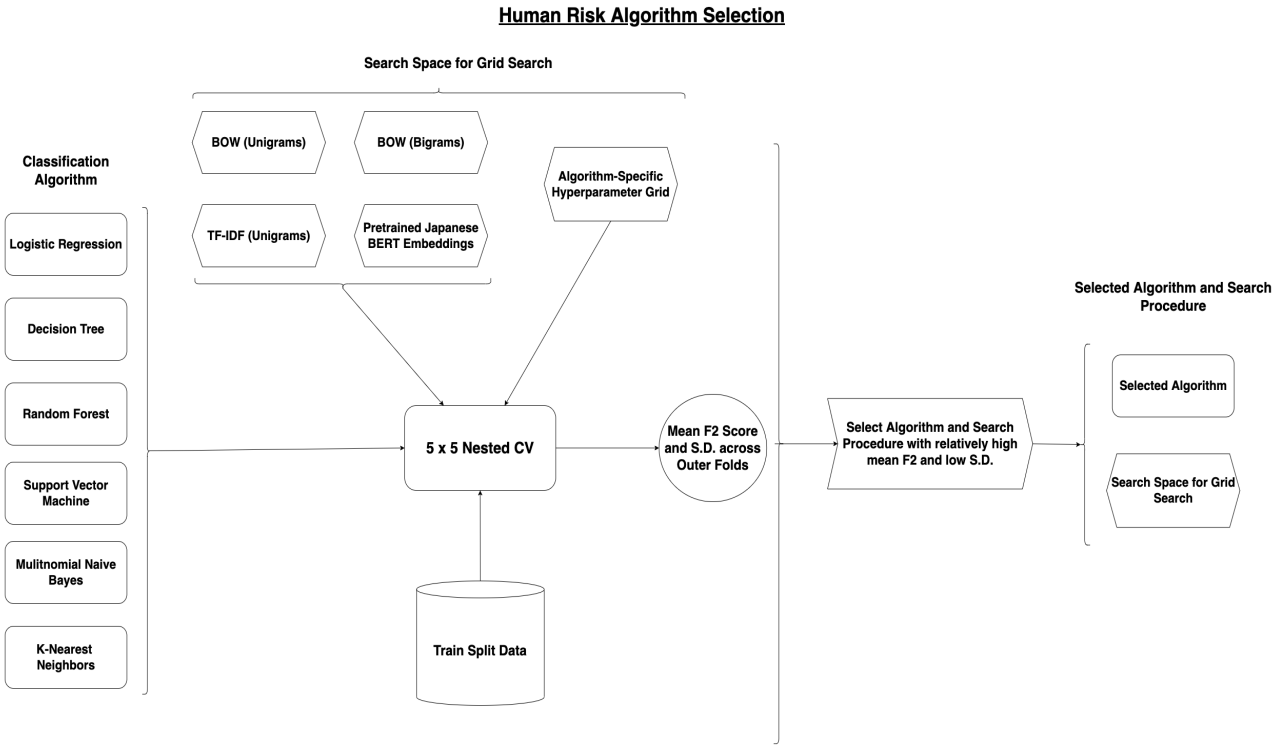
As mentioned, we were interested in investigating various machine learning algorithms for performing the task of human risk classification. These algorithms included

Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), and K-Nearest Neighbors (KNN). Each of the algorithms comes with a unique set of tunable hyperparameters. We publish the hyperparameter grids we used for each algorithm as a JSON.[6] Additionally, the hyperparameter grid used for each algorithm can be found in Appendix Section A.1. In addition to the defined hyperparameter grids, we treat the choice of text featurization as an additional hyperparameter. That is, we investigate BOW (Unigram), BOW (Bigram), TF-IDF (Unigram) featurizations, and BERT embeddings as described in Section 5.3 as another hyperparameter in the hyperparameter grid of each ML algorithm. We note that for the Logistic Regression, Support Vector Machine, and K-Nearest Neighbors algorithms, the text features are scaled using the StandardScaler construct from the scikit-learn Python library [60] to make the text features standard normal, i.e. by subtracting the mean and dividing by the standard deviation prior to being inputted to the model. The StandardScaler instance is "fitted" to a set of text feature vectors, e.g. text features associated with training documents, and once fitted, the instance can then transform, or scale the features of other documents, e.g. test documents. For the other algorithms, the features are used as is, i.e. no standardization or normalization. Lastly, we note that the Multinomial Naive Bayes algorithm is only capable of working with non-negative inputs (since multinomial distributions do not support negative values), so we do not use the BERT embeddings when doing hyperparameter tuning on the MNB algorithm during Nested CV.

For each ML algorithm, we apply a 5 x 5 Nested CV algorithm on the training split shown in Table 5.5, using 5-fold stratified cross validation for both the CV that occurs in the outer loops (generalization performance estimation) and the CV that occurs in the inner loops (hyperparameter optimization using grid search). We use stratified cross validation to ensure that the class imbalance for the task is preserved in the folds. As input to the Nested CV algorithm, the full training split data is provided along with a specified algorithm and its corresponding hyperparameter grid.

[6] Link to Algorithm Hyperparameter Grid JSON

Additionally, the F2 score is specified as the metric that is being optimized for. This algorithm selection procedure is shown in Figure 5-5. We note that the random folds of the outer loop 5-fold CV and the random folds of the inner loop 5-fold CV are set prior to the experiment, such that for each algorithm, the same folds are used by the 5 x 5 Nested CV algorithm as to make the performances of the various algorithms and their search procedure comparable. This also allows for the experiment to be reproducible. Relatedly, the random seed of each model is set to the same value for all of the text analysis experiments (both classification and clustering).



▲ Figure 5-5: Algorithm Selection for Human Risk Text Classification

5.4.4 Model Evaluation

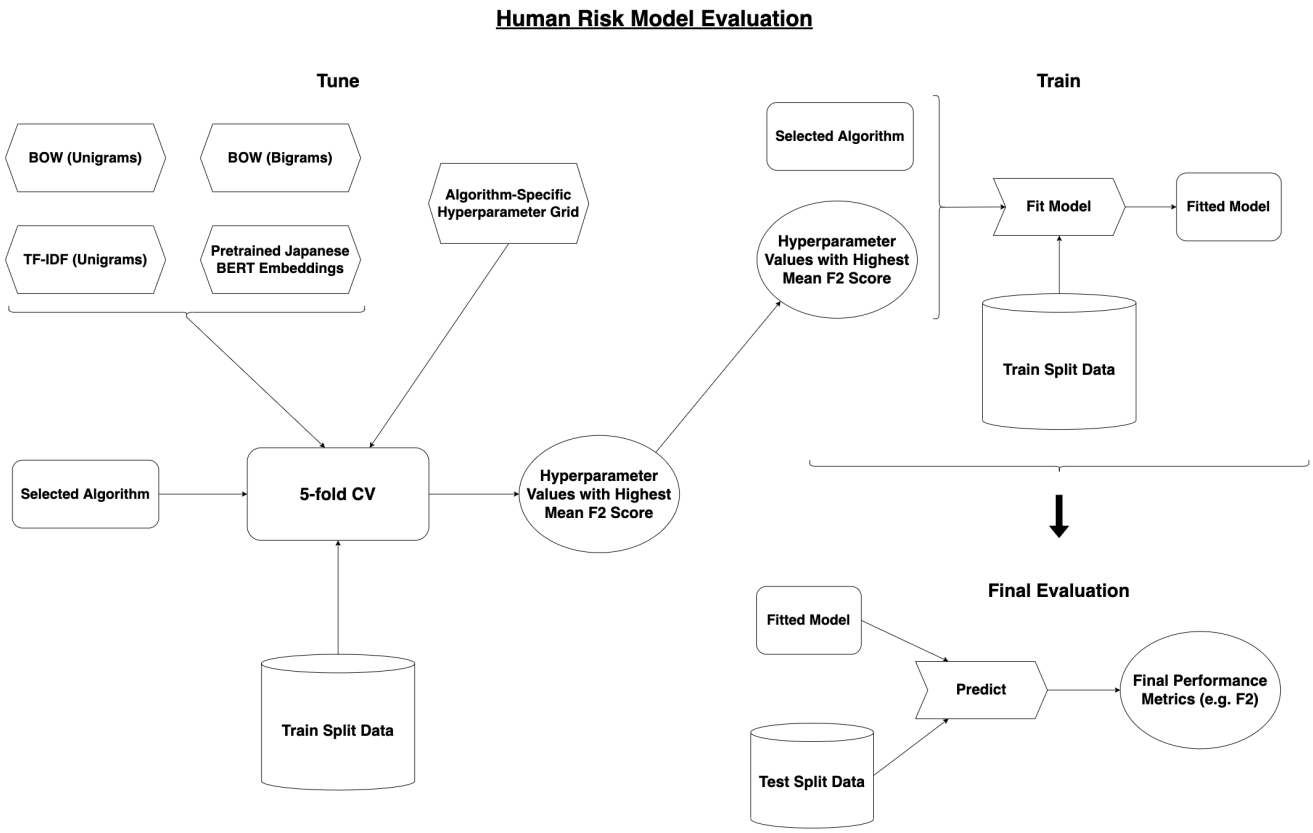
Using the outer folds mean F2 score and corresponding standard deviation (S.D.) found for each ML algorithm from the Nested CV procedure, we select an algorithm which has a relatively high mean F2 score and relatively low variance. Once the algorithm is selected, we perform grid search using 5-fold CV on the entire training

Algorithm 1: P x Q Nested Cross Validation

1. Specify the following parameters:
    - `data`: Full Dataset
    - `P`: Number of folds to use in outer loop cross validation
    - `Q`: Number of folds to use in inner loop cross validation
    - `ml_algo`: ML Algorithm (e.g. Logistic Regression)
    - `hyperparam_grid`: List of hyperparameter combinations to search in the inner loop for algorithm `ml_algo`
    - `metric`: Metric used to evaluate performance
2. `outer_folds` ← Randomly split `data` into list of `P` non-overlapping folds
3. `outer_scores` ← Initialize list of P elements to store `metric` score on each fold of the outer loop
4. For `p` in range(`P`): // *Outer Loop of Nested CV – Estimation of Procedure Generalization Performance*
    (a) `outer_test_data` ← `outer_folds`[`p`]
    (b) `outer_train_data` ← Merge remaining other $P - 1$ folds of `outer_folds`
    (c) `inner_folds` ← Randomly split `outer_train_data` into list of `Q` non-overlapping folds
    (d) `inner_scores` ← Initialize matrix of zeros of shape length(`hyperparam_grid`) by `Q`
    (e) For `q` in range(`Q`): // *Inner Loop of Nested CV – Hyperparameter Optimization (Grid Search)*
        i. `inner_val_data` ← `inner_folds`[`q`]
        ii. `inner_train_data` ← Merge remaining other $Q - 1$ folds of `inner_folds`
        iii. For `i` in range(length(`hyperparam_grid`)):
            A. `hyperparams_i` ← `hyperparam_grid`[`i`]
            B. Fit `ml_algo` on `inner_train_data` with `hyperparams_i`
            C. `metric_score_q` ← `metric` score on `inner_val_data` by fitted `ml_algo`
            D. `inner_scores`[`i`, `q`] ← `metric_score_q`
    (f) `hyperparams_mean_scores` ← Compute vector of row-wise mean of `inner_scores`
    (g) `best_hyperparam_index` ← Get index which corresponds to best value in `hyperparams_mean_scores`
    (h) `best_hyperparam_combo` ← `hyperparam_grid`[`best_hyperparam_index`]
    (i) Fit `ml_algo` on `outer_train_data` with `best_hyperparam_combo`
    (j) `metric_score_p` ← `metric` score on `outer_test_data` by fitted `ml_algo`
    (k) `outer_scores`[`p`] ← `metric_score_p`
5. Return mean and standard deviation of `outer_scores`

set to find the optimal set of hyperparameter values for the algorithm which yields the highest mean F2 score across the folds. Using the best hyperparameter values found, we fit the algorithm to the entire training set, to produce a fitted model.

With this fitted model, we determine a final assessment of the generalization performance (by F2) for the model by predicting on the test split shown in Table 5.5. This model evaluation procedure is shown in Figure 5-6. In addition to the F2 metric, we report other metrics found for this model on the test split including the Area Under the Precision-Recall Curve (AUCPR), the confusion matrix, and per-class metrics (precision, recall, and F1).



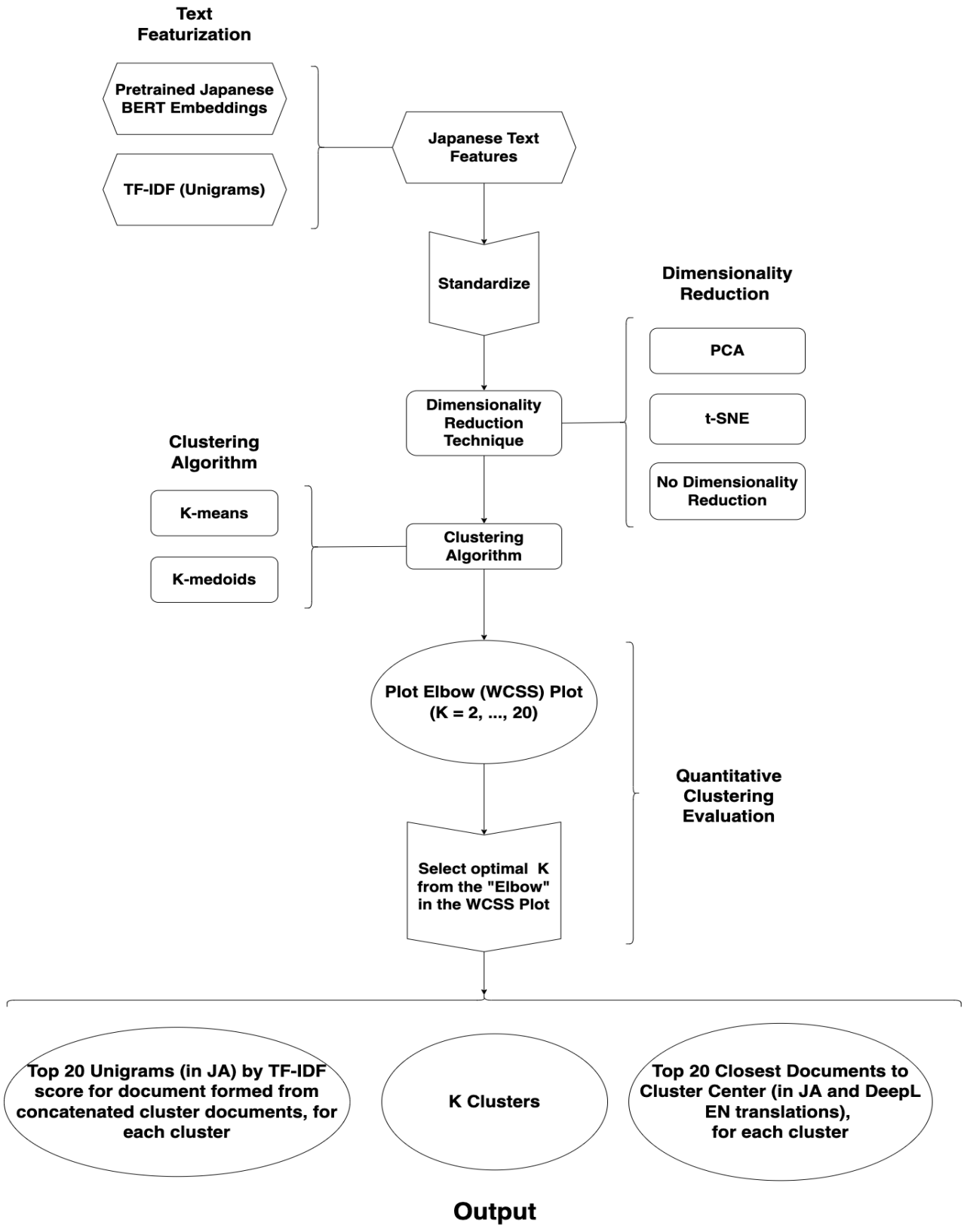
▲ Figure 5-6: Human Risk Model Evaluation

5.5 Clustering of Crowdsourced Japanese Crisis Text Data

Beyond investigating the human risk classification task, we aimed to explore the Fukuchiyama firefighter flood crisis report corpus to see if we could uncover other coherent categories which may exist in the data. These uncovered categories could inform the development of classification tasks in a future work, in addition to any of the humanitarian categories provided by the EOC as mentioned in Section 5.2. This exploration is powered by a series of unsupervised learning techniques including dimensionality reduction and clustering. We developed a pipeline that utilizes these unsupervised techniques to perform the clustering experiments we conducted to uncover coherent categories in the data.

5.5.1 Clustering Experiments

We conducted 12 clustering experiments, each using one of 12 unique combinations of text featurizations, dimensionality reduction techniques, and clustering algorithms, which are applied sequentially as shown in Figure 5-7. For text featurizations, we investigated TF-IDF based on unigrams and BERT embeddings as described in Section 5.3. These featurizations are created for the entire FC firefighter text report corpus of 716 reports. We note that we apply the StandardScaler construct in scikit-learn to the featurizations prior to using them as inputs to the downstream algorithms. Since the feature vectors for both the TF-IDF and BERT embeddings are high-dimensional, 1489 features and 768 features, respectively, we investigate two dimensionality reduction techniques, PCA and t-SNE (as mentioned in Section 1.2.4) with 2 components as well as no dimensionality reduction at all, i.e. using the standardized features as is for the clustering algorithm. Finally, for clustering, we apply either K-means or K-medoids, placing each data point into a single cluster. We compare these two clustering algorithms as K-means is sensitive to outliers present in the data, whereas K-medoids is known to be more robust against outliers [26], which we hypothesize



▲ Figure 5-7: Japanese Crisis Text Clustering Pipeline

improves the capability of producing clusters which have an interpretable overarching category. We summarize the various configuration hyperparameters and corresponding search space of values for the clustering experiments in Table 5.6.

▲ Table 5.6: Clustering Experiment Configuration Hyperparameters and Search Space

Hyperparameter	Search Space
Text Featurization (Standardized)	{BERT, TF-IDF (Unigram)}
Dimensionality Reduction Technique	{None, PCA (2 Comps.), t-SNE (2 Comps.)}
Clustering Algorithm	{K-means, K-medoids}

For each experiment, we plot the WCSS plot, also known as the "Elbow" plot, for values of K clusters between 2 to 20. WCSS is a loss function which captures the extent to which data points within the same cluster are at a close distance to each other, in our case, by euclidean distance. Ideally, we want this value to be low, but not too low as we may fit to noise in the data, or "overfit". Thus, a widely used heuristic is to select an optimal K value at which an "elbow" occurs in the plot, i.e. when the marginal decrease in WCSS is dramatically less after K clusters than it is before K clusters [73].

5.5.2 Clustering Evaluation

From the resulting 12 WCSS plots of the experiments mentioned above, we identify a subset of text featurization, dimensionality reduction technique, and clustering algorithm combinations which yield relatively low WCSS scores across cluster values and which had an "elbow" in their corresponding WCSS plot. For each combination in this subset, we identify an optimal K value (using the "elbow") associated with the combination. With the optimal K value identified, we use the clustering pipeline to produce a 2D visualization of the resultant clustering, if dimensionality reduction is applied. To enable qualitative analysis and assignment of interpretable labels to the clusters, first, for each cluster, we show the top 20 closest documents (by euclidean distance) within a cluster to the cluster center. We show both the original Japanese

(JA) report text and the English (EN) translation of the report provided by DeepL,[7] a neural translation system. We note that the English translations were manually cleaned of inaccuracies by a member of the Urban Risk Lab who is fluent in English and Japanese.[8] Then, for each cluster, we form a cluster-level document by concatenating all unigram tokens of the documents within the cluster, forming a new corpus of cluster documents. From this cluster-level document corpus, we compute the top 20 words (in JA) by TF-IDF score for that cluster. These auxiliary utilities aided in performing the qualitative analysis of the interpretability of the clusters.

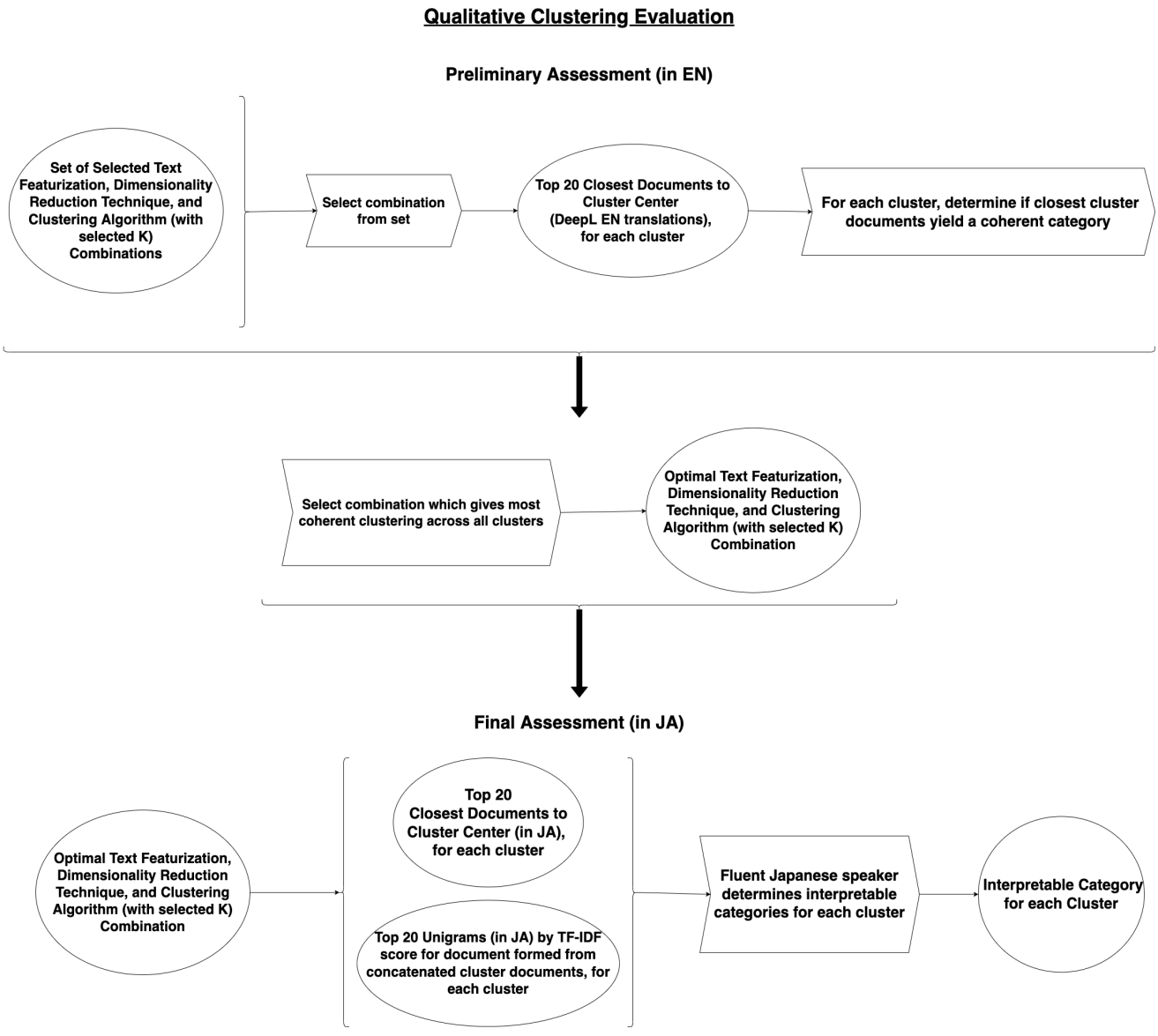
For each combination in the subset mentioned above, we utilize the English translations of the top 20 closest cluster documents to assess the interpretability of the cluster, essentially answering the question: **When looked at together, does the content of the representative documents in a cluster elicit an interpretable label?** We refer to this as the preliminary assessment. We then identified a combination in the subset which yielded the most interpretable labels across combinations, i.e. selected the combination which had the highest number of clusters that had representative documents which elicited an interpretable label.

Using the selected optimal combination, for each resulting cluster found using this combination, we produce auxiliary outputs showing the top 20 closest documents in Japanese within the cluster to the cluster center and the top 20 unigrams (in JA) by TF-IDF score for the cluster-level document corpus. As a final assessment, we were assisted by a fluent English and Japanese speaker from the Urban Risk Lab at MIT[9] who investigated the closest documents and the top TF-IDF unigrams in each cluster, and then used these to assign an interpretable label to each cluster. This qualitative evaluation procedure is shown in Figure 5-8.

[7] DeepL Translation: https://www.deepl.com/en/translator

[8] We acknowledge Saeko Baird of the Urban Risk Lab at MIT who cleaned these translations of their inaccuracies.

[9] We acknowledge Saeko Baird of the Urban Risk Lab at MIT who assigned an interpretable label to each cluster.



▲ Figure 5-8: Qualitative Clustering Analysis Workflow & Evaluation

5.6 Implementation

Similar to the Image Analysis Module, we built an open-sourced Python package for the Text Analysis Module to enable the flexibility to use the utilities we developed for our text classification and clustering experiments described above in future work. To implement this Python package, we leveraged several Python packages including scikit-learn [60], pandas [62], NumPy [63], matplotlib [64], seaborn [65], the Hugging Face library [76], fugashi [68], Unidic Lite,[10] IPAdic,[11] scikit-learn-extra,[12] nltk [77], and PyTorch [58].

5.6.1 URL Text Module Python Package

The URL Text Module Python Package provides utilities for text preprocessing and featurization as well as conducting text classification and clustering experiments as presented above. Specifically, we embed the preprocessing and featurization pipeline and the clustering pipeline we have developed within this Python package. We note that some of the utilities are geared towards the Japanese language, however there are also utilities which are language-agnostic that can be applied on any language, i.e. n-gram-based featurization.

We provide utilities for saving metadata from experiments such as hyperparameters used, intermediate and final results from Nested CV, auxiliary outputs from clustering experiments, and random states used for reproducibility. We also provide methods useful for visualization and analysis of Nested CV results, tuned model performance, and clustering results which are shown in Section 5.7.

To learn how to use the URL Text Module Python Package, please see here.

[10] Unidic Lite Python Package: https://pypi.org/project/unidic-lite/
[11] IPAdic Python Package: https://pypi.org/project/ipadic/
[12] scikit-learn-extra Python Package: https://scikit-learn-extra.readthedocs.io/en/stable/

5.7 Results

In this section, we report our results from the human risk text classification experiments, specifically results from the algorithm selection procedure and subsequent model evaluation. We also report results from the clustering experiments for the full FC firefighter crisis text report corpus, which includes results from quantitative and qualitative analysis.

5.7.1 Algorithm Selection through Nested Cross Validation

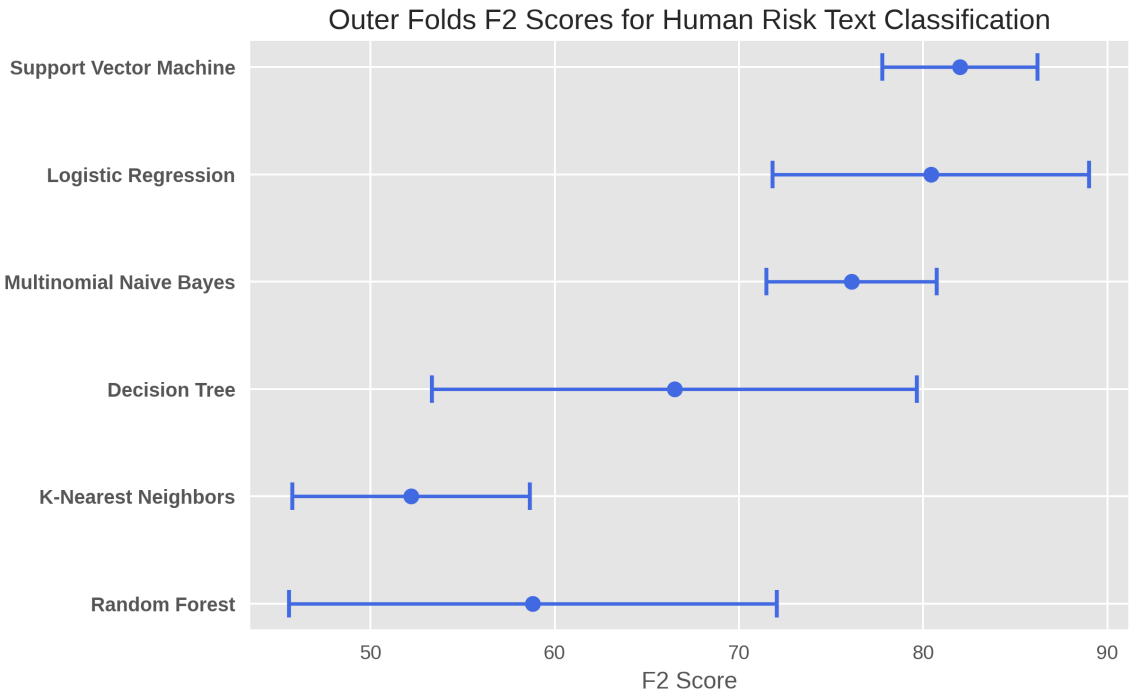
In Table 5.7 and Figure 5-9, we present the final results of the 5 x 5 Nested CV procedure applied to each algorithm and its corresponding hyperparameter grid. The hyperparameter grid of each algorithm can be found in Appendix Section A.1. These results are determined from the generalization performance estimation found from the performance (by F2 score) on the outer loop 5-fold CV in Nested CV. We make available the intermediate and final results of Nested CV for each algorithm and corresponding hyperparameter grid.[13]

We note that some of the algorithms had relatively high standard deviation, i.e. variation, in their performance on the outer folds, namely Decision Tree, Random Forest, and Logistic Regression with 13.2%, 13.2%, and 8.59%, respectively. Support Vector Machine, Logistic Regression, and Multinomial Naive Bayes had the highest mean F2 performances with 82.0%, 80.4%, and 76.1%, respectively. From this, we determine that the performance of the Support Vector Machine algorithm with its corresponding hyperparameter search procedure yielded the highest mean F2 score, 82.0%, and the lowest standard deviation, 4.22%. We therefore select the Support Vector Machine (SVM) algorithm and its corresponding hyperparameter grid for the final human risk model evaluation.

[13] Link to Results of Nested CV

▲ Table 5.7: Mean F2 Score and Standard Deviation for Algorithm and Hyperparameter Search Procedure on Outer Folds of Nested CV. Algorithm and corresponding Search Procedure with best results are **boldfaced**

Algorithm	Mean F2	Standard Deviation
Support Vector Machine	**82.0%**	**4.22%**
Logistic Regression	80.4%	8.59%
Multinomial Naive Bayes	76.1%	4.62%
Decision Tree	66.5%	13.2%
K-Nearest Neighbors	52.2%	6.44%
Random Forest	58.8%	13.2%



▲ Figure 5-9: Mean F2 Score and Standard Deviation for Algorithm and Hyperparameter Search Procedure on Outer Folds of Nested CV

5.7.2 Human Risk Model Performance Evaluation

Prior to performing the final evaluation of the SVM on the test split data, we performed 5-fold CV on the full train split, applying grid search with the hyperparameter grid associated with the SVM algorithm to find optimal hyperparameter values. The search space forming the hyperparameter grid for SVM is shown in Table A.7. The optimal hyperparameter values found for the SVM algorithm from 5-fold CV applied on the full train split are presented in Table 5.8.

▲ Table 5.8: Hyperparameter Values of the Tuned SVM Model

Hyperparameter	Optimal Value
C	0.001
Kernel	Linear
Class Weight	Balanced
Text Featurization (Standardized)	BERT

When fitting to the training set, the hyperparameter C of SVM is inversely proportional to the regularization strength applied when creating the margin of the linear separator, the decision boundary, which separates the data points into the "Human Risk" and "No Human Risk" classes. Specifically, a lower value of C increases the margin of the hyperplane to prevent overfitting, potentially misclassifying some points in the training data, while a higher value decreases the margin misclassifying less training data points, but potentially producing a model which is not as generalizable [60].

SVM aims to find the linear separator which best separates the data. Sometimes the data in the original feature space is not linearly separable. Besides applying soft-margin SVM, which makes of use of hyperparameter C, it sometimes helps to apply a kernel function to the input data. When a kernel function is applied the original data, additional features are added, mapping the data to a higher-dimensional space in which the data can hopefully then be linearly separated [60]. This results in a non-linear separator in the original feature space. Since the optimal value for the

SVM hyperparameter *kernel* is *linear*, the data is separated using a linear separator in the original feature space. For the tuned SVM model, the linear separator was used to separate the standardized BERT embeddings in 768-dimensional space.

When the *Class Weight* hyperparameter of SVM is set to *Balanced*, this makes the model weight misclassification penalties for each class with a weight that is inversely proportional to the class's relative frequency in the training set. During training, this results in weighting the misclassification penalty associated with misclassifying a data point in the minority class, e.g. the "Human Risk" class, as higher than the penalty for misclassifying a data point from the majority class [60].

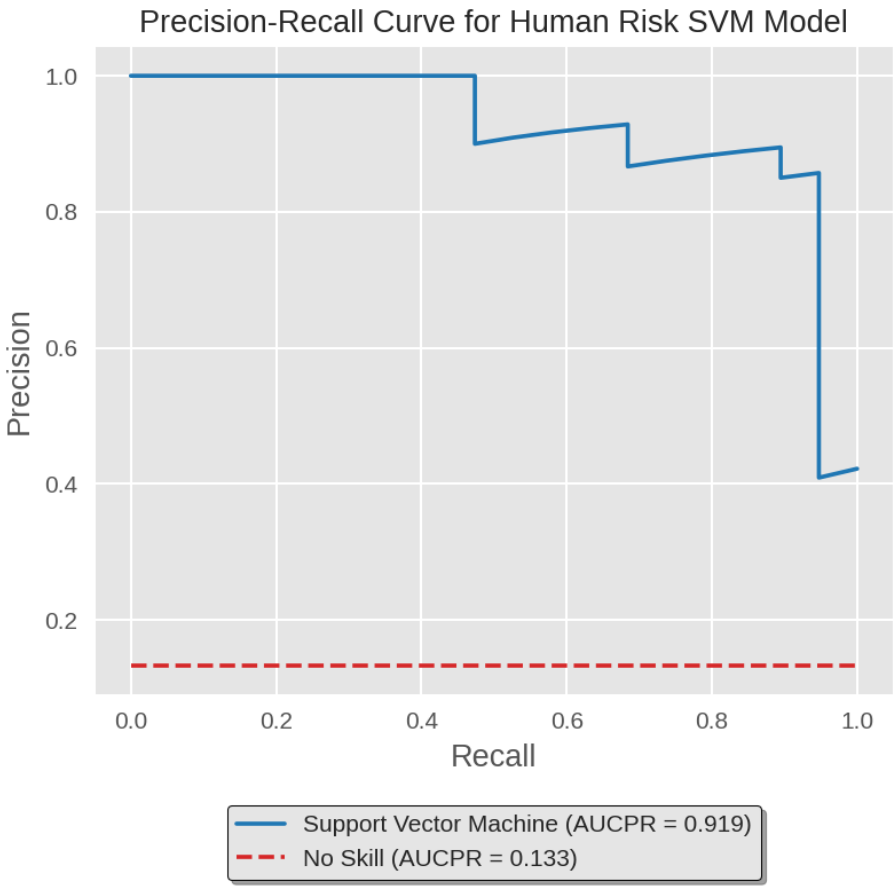
We report the estimated generalization performance of the tuned SVM found from the 5-fold CV described above, noting that this is likely a biased estimate of generalization performance as the 5-fold CV procedure was also used to tune the model. **The tuned SVM model achieves a mean F2 score of 85.0% and has a standard deviation of 7.40%.**

Summary Performance Metrics on Test Split Data

As a final evaluation of the generalization performance of the model, we use the optimal hyperparameters shown in Table 5.8, train the SVM algorithm on the entire train split dataset, and predict on the test split shown in Table 5.5. **We report an F2 score of 92.8% on the test split data.** We note that the baseline model which always predicts the "Human Risk" class, achieves an F2 score of 43.4%. To gain other insights into the model performance, we investigate various other performance metrics of the model on the test split, such as the Precision-Recall plot and the associated Area Under the Precision-Recall Curve (AUCPR) score.

▲ Table 5.9: Performance (by F2) of Tuned Human Risk SVM Model in Different Evaluation Settings

Evaluation Setting	F2
5-Fold CV on Full Training Set (Mean F2 ± S.D.)	85.0% ± 7.40%
Trained on Full Training Set, Evaluated on Test Set	92.8%



▲ Figure 5-10: Precision-Recall Curve for Human Risk SVM Model (AUCPR = 0.919) on Test Split

The Precision-Recall curve (PR curve) visualizes the tradeoff in precision and recall for different classification thresholds used by a classifier when classifying data points [60]. [78] advises using the PR curve over the ROC curve in the case of imbalanced data, as ROC can give an optimistic estimate of the classifier's output quality by considering true negatives in the computation, which in high quantity can dramatically lessen the effect of the false positives, false negatives, and true positives in the performance estimate, giving a misleadingly high estimate of performance. This is especially true when the positive class is the class of interest for the task as is the case for the human risk task. Alternatively, AUCPR is a single-number summary of the quality of a classifier based on the PR curve, which incorporates false positives, false negatives, and true positives into the computation, while ignoring true negatives altogether. The better the classifier (higher recall and higher precision) the closer

the AUCPR is to 1 [60]. For this metric, the performance of a baseline model has an AUCPR score which is given by the proportion of positive samples to total number of samples in the test dataset [78]. Thus, in this case, the baseline AUCPR score is equal to 0.133. For SVM, the confidence level for a prediction is computed from the model's score function, which finds the signed distance between a data point and the learned separator. These confidence levels are used in determining precision-recall values for various decision thresholds. We report an AUCPR of 0.919 for the SVM model on the test split data. The PR curve for the SVM model is shown in Figure 5-10. In addition to F2 score and AUCPR, we investigate the performance of the SVM model using the confusion matrix and per-class metric scores.

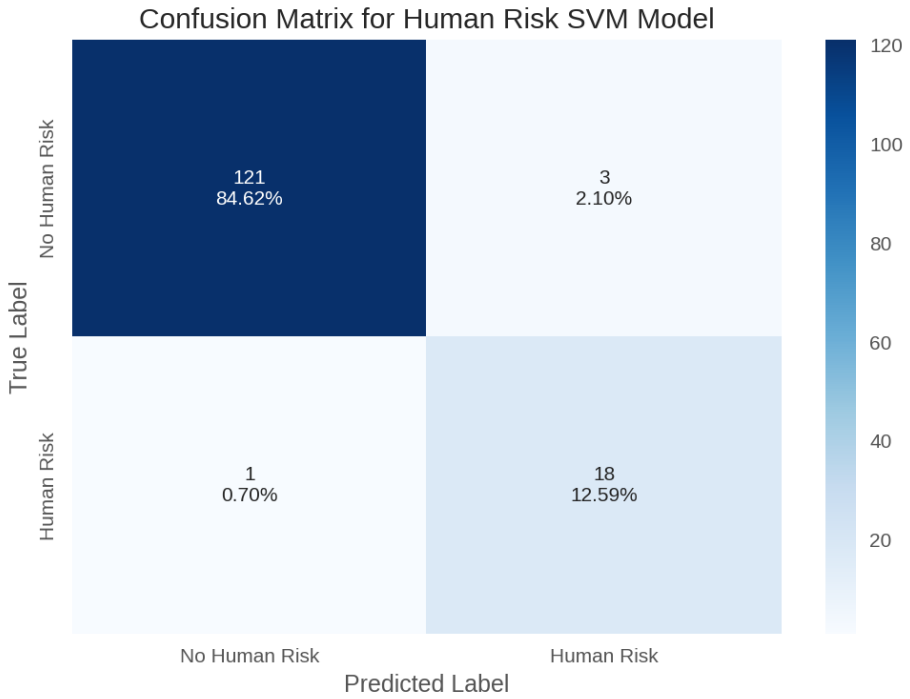
Confusion Matrix and Per-Class Metric Scores

From the confusion matrix in Figure 5-11, we see that the model made very few misclassifications on the test split data. Specifically, the model only misclassified one data point which was labeled as "Human Risk" as "No Human Risk" out of all 19 data points labeled as "Human Risk", thus the model had low false negatives. The model misclassified 3 "No Human Risk" data points as "Human Risk" out of a total of 124 "No Human Risk" data points, thus the model predicted 3 false positives. We note that the model had more false positives than false negatives, but few of each.

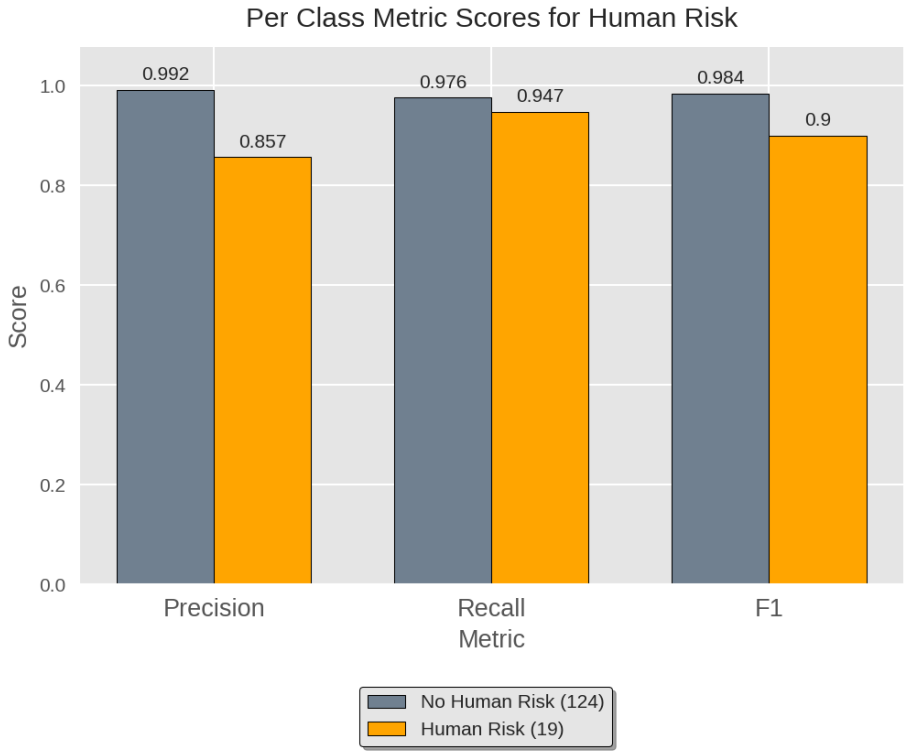
We provide the plot of the per-class metrics of precision, recall, and F1 for each class in Figure 5-12. The model performs well by all per-class metrics on the "No Human Risk" class achieving scores equal to and above 0.976. Comparatively lower performance is observed across all metrics for the "Human Risk" class with precision, recall, and F1 scores of 0.857, 0.947, and 0.9, respectively.

5.7.3 Uncovering Categories by Clustering Fukuchiyama Firefighter Crisis Reports

Using our clustering pipeline shown in Figure 5-7, we present our results for clustering of the FC firefighter crisis text report corpus using various configurations for



▲ Figure 5-11: Confusion Matrix for Human Risk SVM Model (F2 = 92.8%) on Test Split



▲ Figure 5-12: Per-Class Performance for Human Risk SVM Model on Test Split

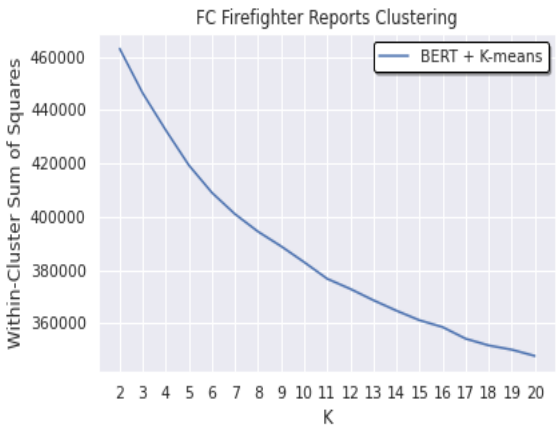
text featurization, dimensionality reduction technique, and clustering algorithm. We used these results to form a query subset of configuration combinations, where we selected specific configuration combinations to further investigate for intuitive clustering quality. After investigating whether the clustering results were intuitive for each configuration combination in the query subset, we select a combination for final assessment by a fluent English and Japanese speaker who assigned an interpretable label to each cluster.

Selecting Subset of Configuration Combinations for Qualitative Analysis

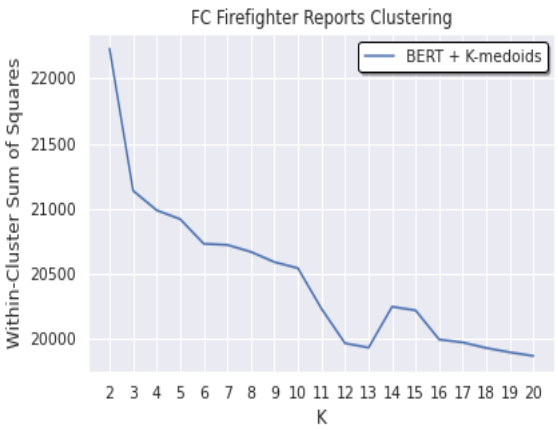
We present the WCSS plots for standardized BERT embeddings in Figure 5-13 and corresponding WCSS values in Table A.10. We observe that the WCSS curves produced using K-means are much smoother than the plots made when using K-medoids across various dimensionality reduction techniques. For the BERT embeddings, across the various dimensionality reduction techniques, it is observed that the WCSS is significantly higher across values of K (2-20) when using K-means than when using K-medoids.

Similarly, we provide WCSS plots using standardized TF-IDF featurizations of the text in Figure 5-14 and corresponding WCSS values in Table A.11. When no dimensionality technique is applied to the standardized TF-IDF features, we see that the WCSS plots have an interesting shape, decreasing significantly between many values of K, even for the higher values of K, i.e. $17, 18, 19, 20$. Similar to what was observed for the BERT embeddings, the values of WCSS are significantly lower when using K-medoids rather than K-means across K values when there is no dimensionality reduction or when t-SNE is applied on the standardized TF-IDF features. However, this does not occur when using PCA, where K-means initially produces a significantly higher WCSS for the first few K values, but then drops below the WCSS values found from applying K-medoids.

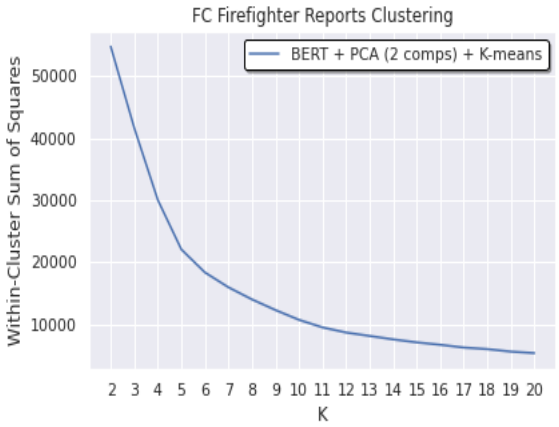
To further investigate clustering quality, we choose to focus on the configuration combinations which used K-medoids for clustering since it is observed that typically the WCSS values found using K-medoids are significantly lower than those found



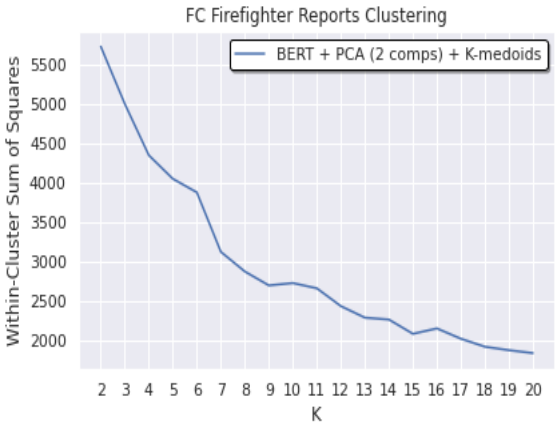
(a) K-means



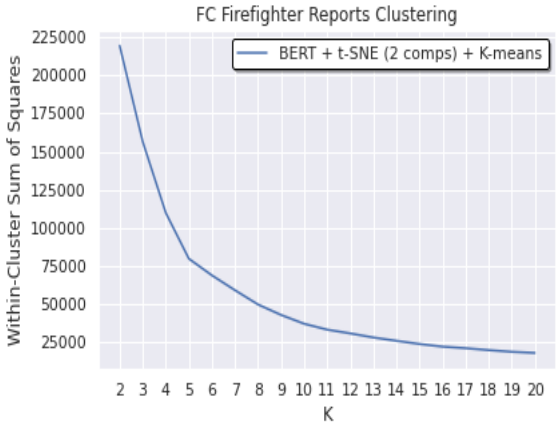
(b) K-medoids



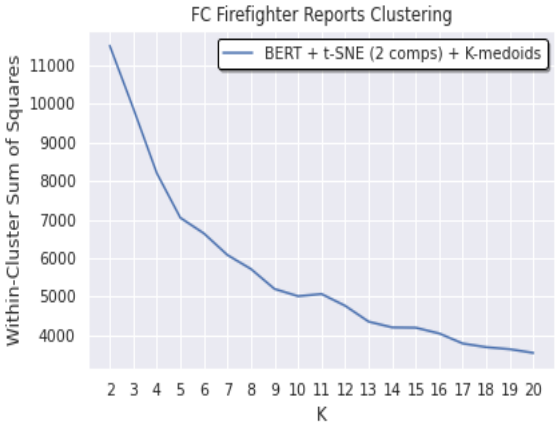
(c) PCA (2 comps) & K-means



(d) PCA (2 comps) & K-medoids

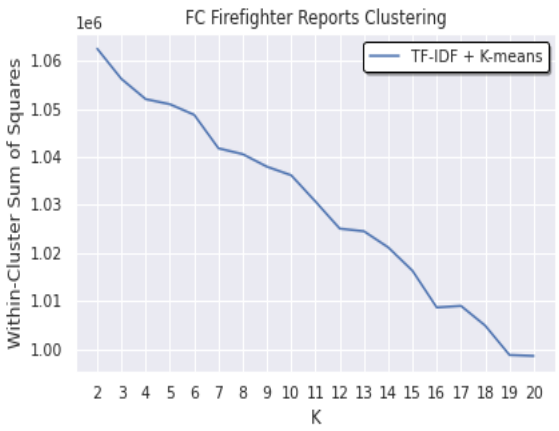


(e) t-SNE (2 comps) & K-means

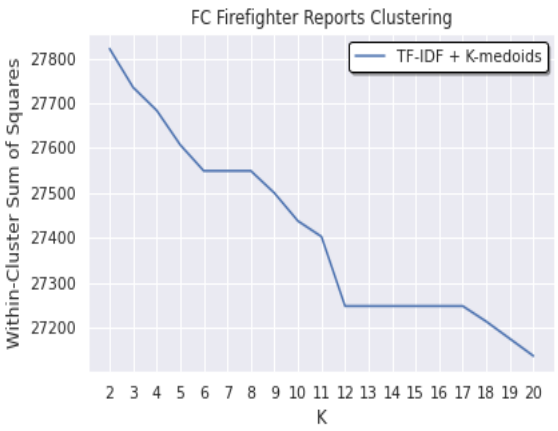


(f) t-SNE (2 comps) & K-medoids

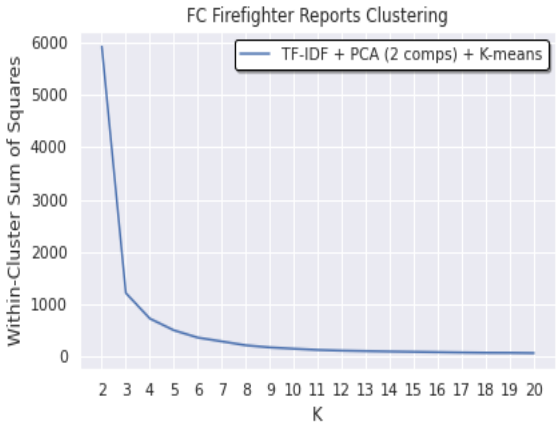
▲ Figure 5-13: WCSS Plots for Standardized Pretrained Japanese MLM BERT + CLS Pooling Embeddings. See Table A.10 for WCSS Values for each K



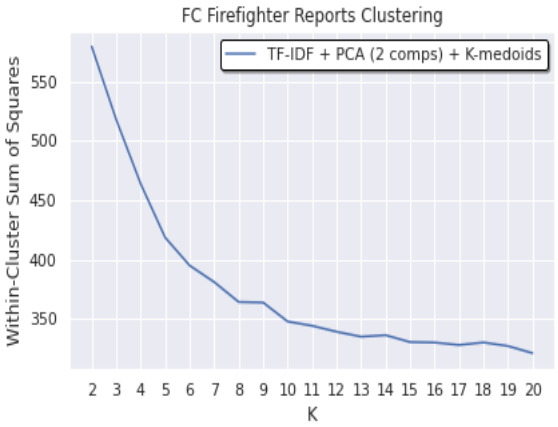
(a) K-means



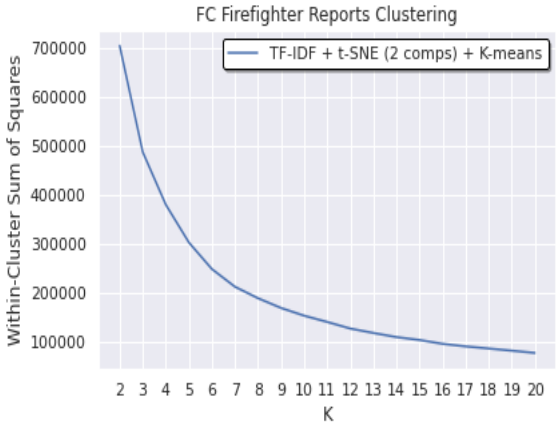
(b) K-medoids



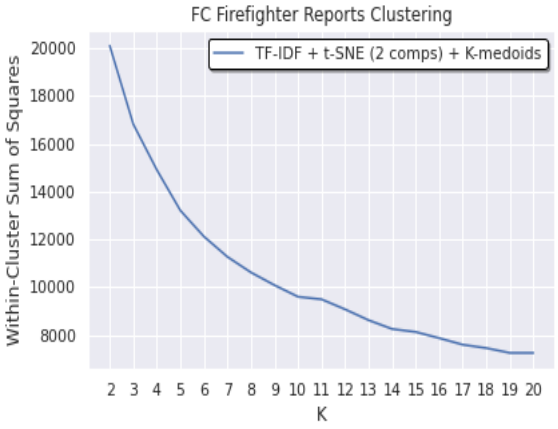
(c) PCA (2 comps) & K-means



(d) PCA (2 comps) & K-medoids



(e) t-SNE (2 comps) & K-means



(f) t-SNE (2 comps) & K-medoids

▲ Figure 5-14: WCSS Plots for Standardized TF-IDF (Unigram) Text Features. See Table A.11 for WCSS Values for each K

from K-means across all K values between 2 to 20. Additionally, we chose not to investigate the configuration combination which uses TF-IDF features, no dimensionality reduction, and K-medoids clustering as the WCSS values decrease dramatically even for higher values of K. To select to the optimal number of clusters, i.e. K, we reference the WCSS plot of each selected configuration combination and choose the K value where the "elbow" occurs. The resulting subset of configuration combinations we query for qualitative analysis along with their corresponding selected K value are shown in Table 5.10. In Table 5.10, we give each selected configuration combination an ID. We use these IDs in our presentation of the qualitative results of the clustering found using each selected configuration in the next section.

▲ Table 5.10: Selected Subset of Configuration Combinations for Preliminary Qualitative Clustering Analysis. Configuration Combination which gave best results is **boldfaced**

Configuration ID	**Featurization (Standardized)**	**Dimensionality Reduction Technique** (2 Components)	**Clustering Algorithm**	**Selected K**
0	BERT	—	K-medoids	12
1	BERT	PCA	K-medoids	9
2	**BERT**	**t-SNE**	**K-medoids**	**9**
3	TF-IDF	PCA	K-medoids	8
4	TF-IDF	t-SNE	K-medoids	14

Investigating Configuration Combinations in the Query Subset

For each configuration combination in the query subset shown in Table 5.10, we investigated the DeepL English translations of the top 20 documents within a cluster which were closest, by euclidean distance, to the cluster center. We report qualitative summaries from our investigation of closest cluster documents in each cluster in Table 5.11.

For some of the selected configuration combinations, we got mixed results, with the configuration giving some intuitive clusters but also producing clusters which did

▲ Table 5.11: Preliminary Qualitative Clustering Analysis. Configuration which gave best results is **boldfaced**. Refer to Table 5.10 for hyperparameter values associated with each Configuration ID

Configuration ID	Possible Cluster Labels and General Qualitative Summary of Clustering
0	Impassable Roads/Closed Roads; Infrastructure Flooding (for 2 separate clusters); Meteorological Advisory or Warning; Impassable Roads due to Landslide or Fallen Tree; Landslide Impact/Power Outages/Rescue Requests; Landslide Impact; Rescue Requests **Summary:** Although there are some intuitive clusters, there are also clusters which are not coherent enough to elicit an overarching label. One cluster was representative of three distinct labels, namely landslide impact, power outages, or rescue requests. There was some redundancy in the cluster labels such as for the infrastructure flooding, landslide impact, and rescue requests labels, which are evident in multiple clusters.
1	Infrastructure Flooding/Landslide Impact (for 3 separate clusters); Residential Flood Impact/Residential Landslide Impact/Rescue Requests/Meteorological Information; Rescue Requests/Residential Flood Impact/Residential Landslide Impact; Impassable Roads/Closed Roads **Summary:** The clustering is somewhat intuitive. Many of the clusters we found to be intuitive actually correspond to multiple cluster labels, e.g. the cluster which has the labels residential flood impact, residential landslide impact, rescue requests, and meteorological information. There is also redundancy in the cluster labels, namely, the infrastructure flooding and landslide impact labels appear together for 3 separate clusters.
2	**Rescue Requests; Closed Roads; Meteorological Advisory or Warning; Impassable Roads; Residential Flood Impact; Landslide Impact/Fallen Tree Impact; Flood Warning; Infrastructure Flooding; Location of Firefighter Activities** **Summary: Of all selected configurations, this resultant clustering was the most intuitive, where each cluster had a clear, intuitive label. Each cluster represents one unique interpretable label.**
3	Closed and/or Flooded Roads (for 2 separate clusters); Rescue Requests; Residential Flood Impact (for 2 separate clusters) **Summary:** Clustering is mostly intuitive, although there is redundancy in the interpreted labels across some clusters such as for the closed and/or flooded roads and residential flood impact labels, which are evident in multiple clusters.
4	Meteorological Advisory or Warning; Closed and/or Flooded Road; Infrastructure Flooding; Rescue Requests/Power Outages; Residential Flood Impact **Summary:** Apart from these cluster labels, many clusters are not coherent enough to produce an interpretable label. One cluster had many reports which indicated either power outages or requests for rescue.

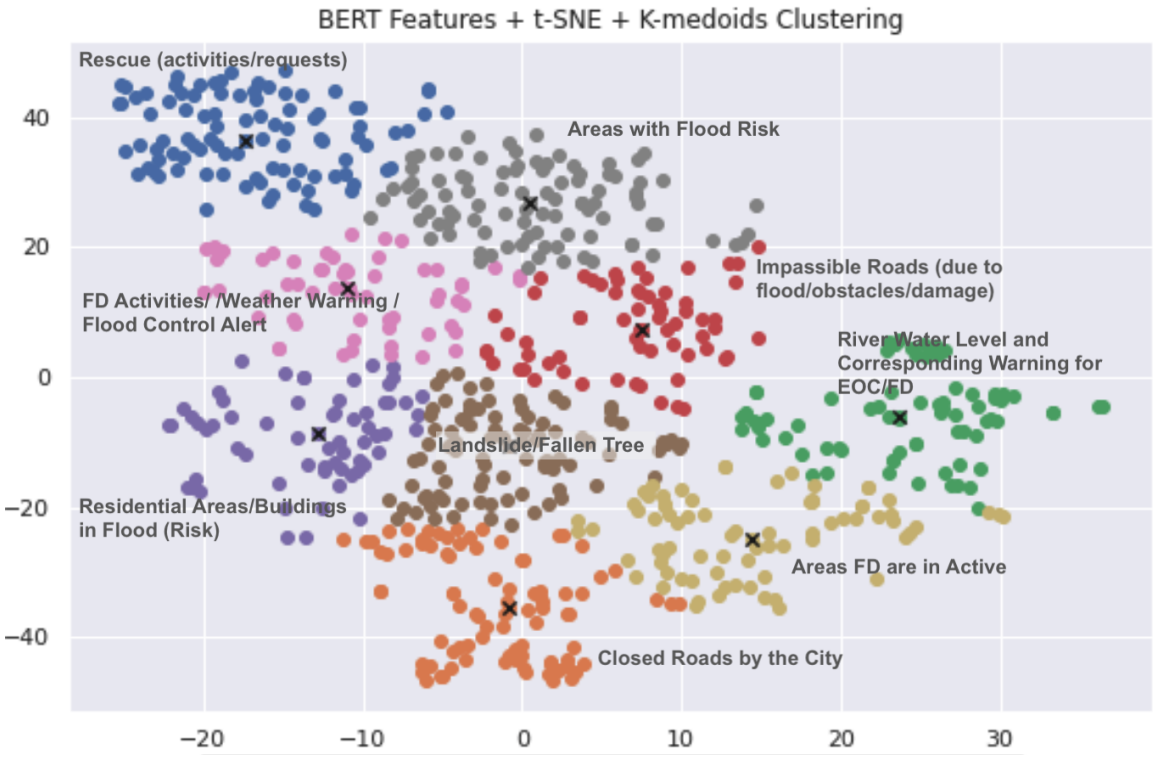
not have an overarching category. We also sometimes observed redundancy in the interpretable labels, i.e. the labels would be evident in multiple clusters. Another interesting observation was that sometimes the interpretable label of a cluster would be representative of multiple labels at once, e.g. both Rescue Requests and Power Outages were evidenced by one cluster for Configuration 4. We note that we also observed a non-negligible amount of data points for which could be considered as indicating multiple of the found interpretable labels at once, not necessarily only the label given to its cluster. We found the best results for Configuration 2, the configuration combination which uses standardized BERT embeddings, t-SNE (2 components) dimensionality reduction, and K-medoids clustering with 9 clusters. This configuration produced the greatest number of unique intuitive cluster labels as indicated from the 20 closest documents within a cluster to the cluster center, yielding a total of 9 clusters which had a unique interpretable label. Thus, we selected this configuration combination for the final qualitative assessment and interpretable label assignment.

Assignment of Interpretable Category to each Cluster

As a final qualitative evaluation, we used the selected configuration combination mentioned above and created a document which shows the 20 closest reports (in JA) within a cluster to the cluster center as well as the top unigrams (in JA) by TF-IDF score using the cluster-level document corpus which identifies the most important unigrams specific to each cluster. We presented this document to a fluent English and Japanese speaker who then assigned an interpretable label to each cluster. We provide the final interpretable cluster labels in Table 5.12. Additionally, since t-SNE was applied to the standardized BERT embeddings, specifically using two dimensions, we are able to visualize the resulting clusters along with their identified interpretable labels in Figure 5-15.

▲ Table 5.12: Interpretable Label given to each Cluster by member of URL MIT who is fluent in English and Japanese

Cluster ID	Interpretable Cluster Label
0	Rescue (Activities/Requests)
1	Closed Roads by the City
2	River Water Level and Corresponding Warning for EOC/FD
3	Impassable Roads (due to Flood/Obstacles/Damage)
4	Residential Areas/Buildings in Flood (Risk)
5	Landslide/Fallen Tree
6	FD Activities/Weather Warning/Flood Control Alert
7	Areas with Flood Risk
8	Areas where FD is active



▲ Figure 5-15: Visualization of Clusters found using BERT Embeddings, t-SNE (2 components), and K-medoids (9 clusters) with labels given by member of URL MIT

5.8 Discussion

Since these text transcripts came from FD operations from past flood events, we note that these clusters highlight some of the information which is deemed important to report by on-the-ground firefighters during a flood crisis event to the Fire Department Headquarters. These clusters suggest potential information needs of the Fire Department in Fukuchiyama, which can be cross-referenced by checking with the other humanitarian category labels provided by the fire department as mentioned in Section 5.2 or by engaging with them directly. The resulting cluster labels revealed new labels for potential classification tasks in addition to other labels provided directly from EOC, to be studied in a future work, which we had not originally considered. It would be interesting to conduct a similar clustering exercise in a future work on Japanese RiskMap data from Typhoon Hagibis, which were submitted by residents on-the-ground, and see if there are any similar or different cluster labels which are revealed from analyzing the resulting clustering. Finally, from our manual inspection of reports in each cluster, we observed that there were a non-negligible amount of reports which mention information that spans multiple of the determined interpretable labels.

When determining the performance metric for the human risk text classification task and evaluating the developed model, we had to answer technical questions including: Does the metric account for imbalance present in the data distribution? Once the metric is determined, what is the performance of the baseline model for the task?

While these technical questions are no doubt important for assessing model efficacy for the task, we underscore that there were other questions we asked which could only be answered by our engagement with crisis managers. These questions played a critical role in determining the performance metric for the task and assessing model efficacy. An important question we asked before we began the development and evaluation of a model for the task was a question about the task itself, namely, do the classes for the task sufficiently capture the expressed information needs of crisis

managers during a crisis? Since these labels were provided directly by crisis managers and given the lesson we learned about the importance of assessing the potential of human casualty in Section 4.7.3, we determined that these classes sufficiently capture an important information need of crisis managers during crisis. For the determining the performance metric for the task, questions which required crisis manager engagement included: Does the metric incorporate the priorities of the crisis managers as it relates to the task, e.g. the cost of a false negative is significantly higher than the cost of a false positive for assessing human risk? Are there multiple metrics that should considered in assessing model efficacy in performing the classification task, e.g. precision and recall, or F2?

Asking these questions allowed us to both consider the technical intricacies for the task (i.e. data imbalance and baseline performance) and directly embed the information needs and priorities of crisis managers into our text ML methodology and evaluation, which are important aims of our framework.

5.9 Summary

Our framework was developed to both highlight the importance of involving crisis managers in the process of developing a ML methodology and contextualize model performance among other measures of efficacy for the ML methodology. Since our framework seeks to be used in the development and iteration of an ML methodology based on the insights gained from crisis managers, we iterated on the Text Analysis Module using insights we had gained from our results on the Image Analysis Module.

Using labels provided directly to us by crisis managers, we created a new text classification task in an effort to better fulfill their information needs during a crisis event. Using insights gained from the results of the qualitative evaluation of the Image Analysis Module in Section 4.7.3, we determined F2 score to be an appropriate performance metric for model performance evaluation as false negatives are more costly than false positives for assessing human risk from text reports. To the best of our knowledge, the exercises of creating a classification task from labels provided

directly by EOC and formulating an appropriate model performance metric informed from crisis expert insights are novel contributions of this work. These exercises follow directly inline with the framework outlined in the Motivation section, using the results from the Image Analysis Module to iteratively design and develop ML models for the Text Analysis Module.

Using F2 as the metric to optimize for during 5 x 5 Nested CV, we were able to identify the SVM algorithm and its corresponding hyperparameter grid as achieving a relatively high mean F2 performance with low variance. To assess the model's ability to perform the human risk classification task, we found the tuned SVM model to achieve an F2 score of 92.8%, which is a substantial improvement over the baseline model's F2 score of 43.4%. Having a baseline is an important aspect of our framework as it enables us the ability to determine if a developed model is performing the task well, i.e. if it does not perform the task better than the baseline, it is not a useful model for the task. This suggests the tuned SVM model is a useful classifier for the task and performs the task reasonably well. This is further evidenced from the Precision-Recall curve, where the tuned SVM model achieved an AUCPR score of 0.919, which is a significant improvement over the typical baseline classifier used for that metric which achieves an AUCPR of 0.133. We note that recall for the "Human Risk" class is higher than precision likely being a result of using F2 as the performance metric to optimize in the classification experiments. Lastly, when looking at the per-class performance metrics for each class, we see that the model performs reasonably well on both classes achieving scores at or above 0.857 for the "Human Risk" class and at or above 0.976 for the "Not Human Risk" class.

From our preliminary clustering assessments, we observed that clustering using K-medoids rather than K-means with all else equal (i.e. text featurization and dimensionality reduction technique), typically yielded lower WCSS scores across all K values between 2-20. This is likely due to the K-medoids algorithm's robustness to outliers and noise, suggesting that there may exist some reports in the corpus which are quite different from the rest.

We note that since the corpus we studied was specific to flood and typhoon cri-

sis events, it is no surprise that many of the identified cluster labels are geared towards flood-related information such as "Areas with Flood Risk", "River Water Level and Corresponding Warning for EOC/FD", "Residential Areas/Buildings in Flood (Risk)", and "Landslide/Fallen Tree". Although some of the categories are quite general such as "Rescue (Activities/Requests)", "Closed Roads by the City", and "Impassable Roads (due to Flood/Obstacles/Damage)", we also see that some of the cluster labels are specific to the fire department such as "Areas where FD is active" and "FD Activities/Weather Warning/Flood Control Alert".

We discuss the broader implications of the findings of our study for future work and conclude this work in the next chapter.

Chapter 6

Conclusion

6.1 Discussion and Implications of Study

Previously developed ML methodologies for mitigating information overload of crisis managers have typically measured the efficacy of their methodology through the use of model performance metrics, such as accuracy, AUROC, F1, precision, or recall. Additionally, these methodologies have been developed and evaluated without the involvement of crisis managers. In this thesis, we have demonstrated that this method of development and evaluation provides a limited perspective on assessing the holistic efficacy a ML methodology has in enhancing crisis awareness and response for crisis managers.

For assessing efficacy towards enhancing crisis awareness and response, we have come to understand that much is lost in the summarization model performance metrics provide in isolation. There are many other points of consideration that come from placing the developed model and its associated performance metric in context. With our framework, we were able consider the following technical questions: Does the model performance metric account for imbalance present in the data distribution for the classification task? What is the performance of the baseline model for the task? More notably, using our framework, we were also able to consider crucial questions which are best answered by engaging with crisis managers including: Do the classes for a task sufficiently capture the expressed information needs of crisis managers dur-

ing a crisis? Does the selected performance metric for a task incorporate the priorities of the crisis managers as it relates to the task? Are there multiple metrics that should considered in assessing model efficacy in performing the task? When developing or iterating on an ML methodology for enhancing crisis awareness and response, these questions can serve as a guide for contextualizing the development and evaluation of ML models so that they can more effectively assist crisis managers during a crisis event.

Beyond the determination of performance metrics, other considerations proved important for assessing the informative utility of the classes associated with a classification task. Interacting with crisis managers enabled us to directly compare the ML methodology we developed for the Image Analysis Module, understanding how the class labels of the classification tasks contained within it and the information they provide compared to the information crisis managers seek to gain from crowdsourced crisis image data. Understanding those similarities and differences can yield classification tasks which have classes with enhanced informative utility for crisis managers in future work.

Lastly, using interannotator agreement measures such as Fleiss' Kappa allowed us to further contextualize our analysis by assessing data quality and reliability in the annotation procedure. These measurements can lead to the identification of tasks which can benefit from the revision of the devised annotation procedure or the reformulation of the classification task itself, i.e. if the classes of a task and the abstract meaning they represent are heavily disputed between humans, we cannot expect a ML model to perform the classification task well. This type of analysis also has the benefit of happening before any modeling takes place, potentially saving time and development resources when building models, which would be unable to perform well if the task is not well-defined.

6.2 Summary of Main Contribution

With this thesis, we have introduced and exemplified a framework which aims to embed all of the mentioned considerations above into the process of designing, developing, and iterating on a ML methodology for enhancing crisis awareness and response. Most notably, this framework situates model development and evaluation, which is commonplace in prior work, as one piece of a broader, contextualized understanding of the efficacy ML methodologies can have for crisis managers in mitigating information overload from crowdsourced crisis reports. Additionally, this framework promotes the iterative development of AI systems and ML methodologies which is informed from the insights gained from engaging with crisis managers, aiming to address this gap in prior work.

This framework is only the beginning for similar work in this domain, as the development of AI systems and ML methodologies for mitigating information overload of crisis managers has many complex intricacies for which we only scratch the surface in our attempt to broaden this discussion within the field. We contribute this framework and the full exhibition of the principles and analyses contained within it to work closer towards a goal worth striving for: enhanced crisis awareness and response from automated assessment of crowdsourced crisis reporting. We now discuss natural extensions of the work presented in this study for future work.

6.3 Future Work

Development of Classification Tasks Informed by Domain Expertise and Clustering

From the results of the image analysis workshop conducted in the Image Analysis Module and the clusters found in the Text Analysis Module, a wide array of labels have been unveiled which we had not originally considered in our devised ML methodology. In addition to the humanitarian categories that were provided with the FC firefighter text reports as mentioned in Section 5.2, these labels can be used to form

the basis of a variety of new classification tasks in a future work which embed the information needs of crisis managers into the class labels. From our clustering analysis, we observed that reports often times contained relevant information pertaining to multiple labels uncovered from the clustering experiments. This highlights the important consideration of forming multilabel classification tasks.

Multilabel Classification Tasks for the Crisis Context

In this work and prior work, many of the classification tasks developed have used nominal, mutually-exclusive classes and thus the models developed for these tasks predict a single label. Data points can sometimes correspond to multiple labels simultaneously for a given task. This suggests the potential for the formulation of multilabel classification tasks in the crisis context, for example assessing types of impact from sediment crisis for which multiple types of impact can be indicated from a report, e.g. landslide, rockfall, and debris flow, etc.

Further Investigation of Information Needs and Priorities of Crisis Managers for Automated Crisis Report Assessment

We note that our qualitative analysis and the insights gained from it were attained by engaging with a sample from the broader population of interest. By continuing to expand the number of crisis managers and EOCs interviewed across geographical contexts who face their own set of unique challenges, ML practitioners and researchers can gain even more insights and use them to iterate on our ML methodologies to embed the information needs and priorities of crisis managers more effectively.

Refinement of Model Performance Metrics Informed by Domain Expertise

In our determination of the β value of the F-β metric used for the human risk task, we noted that we selected the value of $\beta = 2$ because it treated recall as more important than precision and is a popularly used version of F-β. It would be an interesting exercise to further refine this metric by determining a β value which more accurately depicts the priorities of crisis managers as it pertains to this task.

Alternative Methods for Handling Class Imbalance

We aimed to account for class imbalance in part by our determined metric for the human risk task. We also attempted to counteract the negative effects class imbalance can have on model performance by leveraging the class weight hyperparameter of some of the algorithms examined in Section 5.4.3. We note that other techniques could be investigated such as oversampling and undersampling. Additionally, acquiring more data for the minority classes is a potential option for improving performance on the minority classes.

Developing User Interfaces and Performing User Testing

User interfaces with which crisis managers and models interact bring about many interesting questions to consider when designing and developing such interfaces including: What information should be presented? (e.g. the predicted label, the confidence score associated with that prediction) What should be prioritized for viewing? What summaries should be determined from the predicted labels on individual reports during a crisis event? These questions can be investigated by prototyping such interfaces, performing user testing with crisis managers using simulated crisis events, and using the results to iteratively inform how such interfaces should be constructed to be most effective for enhancing crisis awareness and response.

Appendix A

Tables

A.1 Text Classification Hyperparameter Grids

The following tables show the hyperparameter search space or grid for each algorithm as used by grid search in our text classification experiments in Section 5.4. Refer to the Nested CV for Algorithm Selection, Algorithm Selection Results, and Model Evaluation Results sections for more information on how these hyperparameter grids were used in our text classification experiments. We note that the hyperparameter grid for the Logistic Regression algorithm is formed from 4 separate hyperparameter grids shown in Table A.1, Table A.2, Table A.3, and Table A.4. Additionally, we note that all of these classification algorithms as used in our work were imported from the scikit-learn [60] Python library,[1] thus the names of the hyperparameters as presented in these tables were informed from the documentation of the scikit-learn library for these algorithms.

[1] scikit-learn Homepage: https://scikit-learn.org/stable/

▲ Table A.1: **Logistic Regression** Algorithm Hyperparameter Grid I

Hyperparameter	**Search Space**
C	{0.0001, 0.001, 0.01, 0.1, 1, 10, 100}
Solver	{liblinear}
Dual	{True}
Class Weight	{None, Balanced}
Text Featurization (Standardized)	{BOW (Unigram), BOW (Bigram), TF-IDF (Unigram), BERT}

▲ Table A.2: **Logistic Regression** Algorithm Hyperparameter Grid II

Hyperparameter	**Search Space**
C	{0.0001, 0.001, 0.01, 0.1, 1, 10, 100}
Solver	{saga}
Penalty	{Elastic Net}
L1 Ratio	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}
Class Weight	{None, Balanced}
Text Featurization (Standardized)	{BOW (Unigram), BOW (Bigram), TF-IDF (Unigram), BERT}

▲ Table A.3: **Logistic Regression** Algorithm Hyperparameter Grid III

Hyperparameter	**Search Space**
C	{0.0001, 0.001, 0.01, 0.1, 1, 10, 100}
Solver	{newton-cg, lbfgs, liblinear, saga, sag}
Penalty	{L2}
Class Weight	{None, Balanced}
Text Featurization (Standardized)	{BOW (Unigram), BOW (Bigram), TF-IDF (Unigram), BERT}

▲ Table A.4: **Logistic Regression** Algorithm Hyperparameter Grid IV

Hyperparameter	**Search Space**
C	{0.0001, 0.001, 0.01, 0.1, 1, 10, 100}
Solver	{liblinear, saga}
Penalty	{L1}
Class Weight	{None, Balanced}
Text Featurization (Standardized)	{BOW (Unigram), BOW (Bigram), TF-IDF (Unigram), BERT}

▲ Table A.5: **Decision Tree** Algorithm Hyperparameter Grid

Hyperparameter	Search Space
Criterion	{Gini, Entropy}
Max Depth	{5, 10, 20, None}
Min. Number of Samples to Split	{2, 5, 10}
Class Weight	{None, Balanced}
Text Featurization	{BOW (Unigram), BOW (Bigram), TF-IDF (Unigram), BERT}

▲ Table A.6: **Random Forest** Algorithm Hyperparameter Grid

Hyperparameter	Search Space
Number of Trees in the Forest	{10, 100, 250, 500, 1000, 10000}
Max Depth	{5, 10, 20}
Class Weight	{None, Balanced}
Text Featurization	{BOW (Unigram), BOW (Bigram), TF-IDF (Unigram), BERT}

▲ Table A.7: **Support Vector Machine** Algorithm Hyperparameter Grid. Refer to the Model Evaluation Results section

Hyperparameter	Search Space
C	{0.0001, 0.001, 0.01, 0.1, 1, 10, 100}
Kernel	{Linear, RBF}
γ **(For RBF Kernel)**	{0.00001, 0.0001, 0.001, 0.01, 0.1}
Class Weight	{None, Balanced}
Text Featurization (Standardized)	{BOW (Unigram), BOW (Bigram), TF-IDF (Unigram), BERT}

▲ Table A.8: **Multinomial Naive Bayes** Algorithm Hyperparameter Grid

Hyperparameter	Search Space
α	{1, 10, 100}
Text Featurization	{BOW (Unigram), BOW (Bigram), TF-IDF (Unigram)}

▲ Table A.9: **K-Nearest Neighbors** Algorithm Hyperparameter Grid

Hyperparameter	Search Space
Number of Neighbors	{2, ..., 15}
Algorithm for Computing Nearest Neighbors	{Ball Tree}
Leaf Size	{50}
Power Parameter, p	{1, 2}
Text Featurization (Standardized)	{BOW (Unigram), BOW (Bigram), TF-IDF (Unigram), BERT}

A.2 WCSS Scores from FC Firefighter Flood Text Reports Clustering

The following tables correspond to values of the WCSS plots from our clustering experiments on FC Firefighter Flood Text Reports. For the corresponding analysis, please refer to Section 5.7.3.

▲ Table A.10: WCSS Scores for Standardized Pretrained Japanese MLM BERT + CLS Pooling Embeddings for $K = 2, ..., 20$. See Figure 5-13 for corresponding WCSS Plots

	BERT Featurization (Standardized)					
	No Dimensionality Reduction		PCA (2 Components)		t-SNE (2 Components)	
K	**K-means**	**K-medoids**	**K-means**	**K-medoids**	**K-means**	**K-medoids**
2	$4.63 \cdot 10^5$	$2.22 \cdot 10^4$	$5.48 \cdot 10^4$	$5.73 \cdot 10^3$	$2.20 \cdot 10^5$	$1.15 \cdot 10^4$
3	$4.46 \cdot 10^5$	$2.11 \cdot 10^4$	$4.17 \cdot 10^4$	$5.01 \cdot 10^3$	$1.57 \cdot 10^5$	$9.89 \cdot 10^3$
4	$4.32 \cdot 10^5$	$2.10 \cdot 10^4$	$3.01 \cdot 10^4$	$4.35 \cdot 10^3$	$1.10 \cdot 10^5$	$8.21 \cdot 10^3$
5	$4.19 \cdot 10^5$	$2.09 \cdot 10^4$	$2.21 \cdot 10^4$	$4.06 \cdot 10^3$	$7.98 \cdot 10^4$	$7.05 \cdot 10^3$
6	$4.09 \cdot 10^5$	$2.07 \cdot 10^4$	$1.84 \cdot 10^4$	$3.88 \cdot 10^3$	$6.88 \cdot 10^4$	$6.65 \cdot 10^3$
7	$4.01 \cdot 10^5$	$2.07 \cdot 10^4$	$1.60 \cdot 10^4$	$3.13 \cdot 10^3$	$5.91 \cdot 10^4$	$6.09 \cdot 10^3$
8	$3.94 \cdot 10^5$	$2.07 \cdot 10^4$	$1.40 \cdot 10^4$	$2.88 \cdot 10^3$	$4.97 \cdot 10^4$	$5.72 \cdot 10^3$
9	$3.89 \cdot 10^5$	$2.06 \cdot 10^4$	$1.23 \cdot 10^4$	$2.70 \cdot 10^3$	$4.29 \cdot 10^4$	$5.21 \cdot 10^3$
10	$3.83 \cdot 10^5$	$2.05 \cdot 10^4$	$1.07 \cdot 10^4$	$2.73 \cdot 10^3$	$3.71 \cdot 10^4$	$5.02 \cdot 10^3$
11	$3.77 \cdot 10^5$	$2.02 \cdot 10^4$	$9.49 \cdot 10^3$	$2.67 \cdot 10^3$	$3.32 \cdot 10^4$	$5.07 \cdot 10^3$
12	$3.73 \cdot 10^5$	$2.00 \cdot 10^4$	$8.67 \cdot 10^3$	$2.44 \cdot 10^3$	$3.07 \cdot 10^4$	$4.77 \cdot 10^3$
13	$3.69 \cdot 10^5$	$1.99 \cdot 10^4$	$8.12 \cdot 10^3$	$2.29 \cdot 10^3$	$2.81 \cdot 10^4$	$4.36 \cdot 10^3$
14	$3.65 \cdot 10^5$	$2.02 \cdot 10^4$	$7.56 \cdot 10^3$	$2.27 \cdot 10^3$	$2.59 \cdot 10^4$	$4.21 \cdot 10^3$
15	$3.61 \cdot 10^5$	$2.02 \cdot 10^4$	$7.08 \cdot 10^3$	$2.09 \cdot 10^3$	$2.38 \cdot 10^4$	$4.20 \cdot 10^3$
16	$3.59 \cdot 10^5$	$2.00 \cdot 10^4$	$6.69 \cdot 10^3$	$2.16 \cdot 10^3$	$2.21 \cdot 10^4$	$4.05 \cdot 10^3$
17	$3.54 \cdot 10^5$	$2.00 \cdot 10^4$	$6.24 \cdot 10^3$	$2.03 \cdot 10^3$	$2.10 \cdot 10^4$	$3.79 \cdot 10^3$
18	$3.52 \cdot 10^5$	$1.99 \cdot 10^4$	$5.99 \cdot 10^3$	$1.93 \cdot 10^3$	$1.98 \cdot 10^4$	$3.69 \cdot 10^3$
19	$3.50 \cdot 10^5$	$1.99 \cdot 10^4$	$5.59 \cdot 10^3$	$1.88 \cdot 10^3$	$1.87 \cdot 10^4$	$3.64 \cdot 10^3$
20	$3.48 \cdot 10^5$	$1.99 \cdot 10^4$	$5.35 \cdot 10^3$	$1.84 \cdot 10^3$	$1.80 \cdot 10^4$	$3.55 \cdot 10^3$

▲ Table A.11: WCSS Scores for Standardized TF-IDF (Unigram) Text Features for $K = 2, ..., 20$. See Figure 5-14 for corresponding WCSS Plots

	TF-IDF Featurization (Standardized)					
	No Dimensionality Reduction		PCA (2 Components)		t-SNE (2 Components)	
K	**K-means**	**K-medoids**	**K-means**	**K-medoids**	**K-means**	**K-medoids**
2	$1.06 \cdot 10^6$	$2.78 \cdot 10^4$	$5.93 \cdot 10^3$	580	$7.03 \cdot 10^5$	$2.01 \cdot 10^4$
3	$1.06 \cdot 10^6$	$2.77 \cdot 10^4$	$1.21 \cdot 10^3$	518	$4.86 \cdot 10^5$	$1.68 \cdot 10^4$
4	$1.05 \cdot 10^6$	$2.77 \cdot 10^4$	720	464	$3.79 \cdot 10^5$	$1.49 \cdot 10^4$
5	$1.05 \cdot 10^6$	$2.76 \cdot 10^4$	496	419	$3.02 \cdot 10^5$	$1.32 \cdot 10^4$
6	$1.05 \cdot 10^6$	$2.75 \cdot 10^4$	355	395	$2.47 \cdot 10^5$	$1.21 \cdot 10^4$
7	$1.04 \cdot 10^6$	$2.75 \cdot 10^4$	282	381	$2.11 \cdot 10^5$	$1.13 \cdot 10^4$
8	$1.04 \cdot 10^6$	$2.75 \cdot 10^4$	208	364	$1.88 \cdot 10^5$	$1.06 \cdot 10^4$
9	$1.04 \cdot 10^6$	$2.75 \cdot 10^4$	167	364	$1.68 \cdot 10^5$	$1.01 \cdot 10^4$
10	$1.04 \cdot 10^6$	$2.74 \cdot 10^4$	142	348	$1.52 \cdot 10^5$	$9.61 \cdot 10^3$
11	$1.03 \cdot 10^6$	$2.74 \cdot 10^4$	119	344	$1.39 \cdot 10^5$	$9.50 \cdot 10^3$
12	$1.03 \cdot 10^6$	$2.72 \cdot 10^4$	106	339	$1.26 \cdot 10^5$	$9.08 \cdot 10^3$
13	$1.02 \cdot 10^6$	$2.72 \cdot 10^4$	97	335	$1.17 \cdot 10^5$	$8.62 \cdot 10^3$
14	$1.02 \cdot 10^6$	$2.72 \cdot 10^4$	89.5	336	$1.08 \cdot 10^5$	$8.26 \cdot 10^3$
15	$1.02 \cdot 10^6$	$2.72 \cdot 10^4$	83	331	$1.02 \cdot 10^5$	$8.13 \cdot 10^3$
16	$1.01 \cdot 10^6$	$2.72 \cdot 10^4$	76.5	330	$9.44 \cdot 10^4$	$7.87 \cdot 10^3$
17	$1.01 \cdot 10^6$	$2.72 \cdot 10^4$	70	328	$8.92 \cdot 10^4$	$7.60 \cdot 10^3$
18	$1.00 \cdot 10^6$	$2.72 \cdot 10^4$	64.7	330	$8.49 \cdot 10^4$	$7.46 \cdot 10^3$
19	$9.99 \cdot 10^5$	$2.72 \cdot 10^4$	63.4	327	$8.04 \cdot 10^4$	$7.26 \cdot 10^3$
20	$9.99 \cdot 10^5$	$2.71 \cdot 10^4$	58.3	321	$7.59 \cdot 10^4$	$7.26 \cdot 10^3$

Appendix B

Derivations

B.1 Fleiss' Kappa Coefficient (κ)

The Fleiss' Kappa Coefficient, κ, and the associated computation described below were originally introduced in [42]. Refer to Section 4.2.3 to see how we use this computation in our analysis of interannotator agreement on labeled images.

For an annotated data set, where each data point has been given a label by n independent annotators (i.e. each data point has n labels assigned to it), we denote the total number of labeled data points in the data set as N. A label is selected from k categories on a nominal scale (i.e. mutually exclusive and exhaustive categories). We denote the subscript i, where $i = 1, ..., N$, which pinpoints a specific data point, i, across all N data points in the dataset and subscript j, where $j = 1, ..., k$, which refers to a specific category, j, across all k possible categories. n_{ij} is defined as the number of annotators who labeled the ith data point with the jth category. Finally, the proportion of labels for category j across all $N \cdot n$ labels, denoted as p_j, is computed as follows:

$$p_j = \frac{1}{N \cdot n} \sum_{i=1}^{N} n_{ij}$$

We note that:

$$\sum_{j=1}^{k} p_j = \sum_{j=1}^{k} \Big(\frac{1}{N \cdot n} \sum_{i=1}^{N} n_{ij} \Big) = \frac{1}{N \cdot n} \sum_{j=1}^{k} \sum_{i=1}^{N} n_{ij} = \frac{1}{N \cdot n} \sum_{i=1}^{N} \sum_{j=1}^{k} n_{ij}$$

Since $\sum_{j=1}^{k} n_{ij} = n$, we see that:

$$\sum_{j=1}^{k} p_j = \frac{1}{N \cdot n} \sum_{i=1}^{N} \Big(\sum_{j=1}^{k} n_{ij} \Big) = \frac{1}{N \cdot n} \Big(\sum_{i=1}^{N} n \Big) = \frac{1}{N \cdot n} \cdot (N \cdot n) = 1$$

$$\therefore \sum_{j=1}^{k} p_j = 1$$

To measure the level of agreement on a specific data point, i, between the n annotators who provided a label to i, we compute the proportion of observed agreement pairs across all categories $\Big(\sum_{j=1}^{k} \binom{n_{ij}}{2} = \sum_{j=1}^{k} \frac{n_{ij}(n_{ij}-1)}{2} \Big)$ to all possible pairs $\Big(\binom{n}{2} = \frac{n(n-1)}{2} \Big)$, which is denoted as P_i:

$$P_i = \frac{1}{n(n-1)/2} \sum_{j=1}^{k} \frac{n_{ij}(n_{ij}-1)}{2} = \frac{1}{n(n-1)} \sum_{j=1}^{k} n_{ij}(n_{ij}-1) = \frac{1}{n(n-1)} \Big(\sum_{j=1}^{k} n_{ij}^2 - n \Big)$$

To estimate the overall level of observed agreement across all N data points, we compute the mean agreement proportion $\bar{P}$ across the agreement proportions as determined above for each ith data point, P_i, namely:

$$\bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i$$

If the annotators labeled the dataset entirely at random, e.g. without looking at the data at all, they would still have some level of agreement by random chance. The expected level of agreement by random chance, denoted as $\bar{P}_e$, is computed as follows:

$$\bar{P}_e = \sum_{j=1}^{k} p_j^2$$

The quantity $1 - \bar{P}_e$ provides the maximum degree of agreement that can be

attained above the expected level of agreement by random chance. The quantity $\bar{P} - \bar{P}_e$ provides the degree the actual level of agreement is above the expected level of agreement by random chance. Using the ratio of these quantities, namely $\frac{\bar{P}-\bar{P}_e}{1-\bar{P}_e}$, we can compute a normalized measure of the overall level of agreement corrected for the expected level of agreement by random chance, that is, the Fleiss' Kappa coefficient, κ, as follows:

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e}$$

B.2 Cohen's Kappa Coefficient (κ)

Cohen's Kappa coefficient, κ, and the associated computation described below were originally introduced in [55]. Refer to Sections 4.5.1 and 4.7.2 to see how we used this computation in our image classification model performance evaluation.

Cohen's Kappa coefficient assumes the same two annotators label the same N data points, such that for each data point, the annotators independently select a label from k possible classes on a nominal scale. We denote the subscripts i and j, where $i, j = 1, ..., k$, which individually refer to a specific category. n_{ij} is defined as the number of data points where annotator 1 labeled a data point as category i and annotator 2 labeled a data point as category j.

We first compute the proportion of observed agreement between the annotators, which we denote as P_o:

$$P_o = \frac{1}{N} \sum_{s=1}^{k} n_{ss}$$

We note that the proportions the annotators label a data point as belonging to a particular class c is as follows, which we denote as P_{1c} and P_{2c} for annotators 1 and 2, respectively:

$$P_{1c} = \frac{1}{N} \sum_{j=1}^{k} n_{cj}, \quad P_{2c} = \frac{1}{N} \sum_{i=1}^{k} n_{ic}$$

We can compute the probability that the annotators agree that a data point should be labeled as class c by random chance (e.g. if they did not look at the data at all when labeling) by using the proportions depicted above. We denote this probability as P_c. In other words, this is the probability that annotator 1 labels a data point as class c **and** annotator 2 labels a data point as class c. Since the labeling was done independently by the annotators, this probability is simply the product of the annotators' respective probabilities of labeling a data point as class c:

$$P_c = P_{1c} \cdot P_{2c}$$

Thus, we can attain the probability of random chance agreement between the two annotators, which we denote as P_e, by taking the sum of the probabilities they agree by random chance across the k classes, namely:

$$P_e = \sum_{c\,=\,1}^{k} P_c$$

The quantity $1 - P_e$ determines the maximum degree of agreement that can be achieved over random chance agreement. The quantity $P_o - P_e$ determines the actual degree of agreement achieved by the observed agreement over the random chance agreement. Thus, in computing the ratio $\frac{P_o - P_e}{1 - P_e}$, we get a normalized measure of the degree the observed agreement is over and above random chance agreement. This is the Cohen's Kappa coefficient, denoted as κ below:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

Cohen's Kappa coefficient ranges from -1 to +1. Generally, the closer to +1 the coefficient, the larger the degree of agreement is over and above the expected agreement by random chance between the annotators. Cohen's Kappa coefficient can also be used as a classification metric for a predictive classification model, which is readily seen if the annotators mentioned above are replaced by a predictive model and the ground-truth labels for the classification task. Used in this manner, P_o is

interpreted as the observed accuracy of the predictive model and P_e is the expected accuracy for a classifier which predicts randomly according to the distribution of the class labels in the ground-truth dataset. This gives a measure of the degree the predictive model performs better than the expected accuracy for the random classifier baseline. It is a classification metric well-suited for multi-class classification and imbalanced datasets [79]. Values equal to zero or less than zero indicate that the predictive model performs the same or worse than the random classifier baseline, and therefore is not useful for the classification task.

B.3 Term Frequency-Inverse Document Frequency (TF-IDF)

Refer to Section 5.3.2 to see how we used this computation in our text preprocessing and featurization pipeline.

$$\text{Term Frequency} = tf(w_i, d_j) = \frac{n_{w_i,d_j}}{\sum_k n_{w_k,d_j}}$$

$$\text{Inverse Document Frequency} = idf(w_i, N) = \log\left(\frac{N}{df_{w_i}}\right)$$

$$\text{TF-IDF}(w_i, d_j, N) = tf(w_i, d_j) * idf(w_i, N)$$

- w_i is a word in the corpus vocabulary.
- d_j is a document in the corpus.
- n_{w_i,d_j} is the count of the occurrence of w_i in document d_j.
- $\sum_k n_{w_k,d_j}$ is the total number of words in document d_j.
- N is the number of documents in the corpus.
- df_{w_i} is the number of documents in the corpus in which word w_i is present.

Bibliography

[1] F. Alam, F. Ofli, M. Imran, T. Alam, and U. Qazi, "Deep learning benchmarks and datasets for social media image classification for disaster response," in *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, IEEE, 2020. (Cited on pages 13, 17, 38, 39, 40, 48, 51, 52, 53, 54, 55, 56, 57, 58, 64, 65, 67, 73, 80, and 85.) ▲

[2] S. Vieweg, A. Hughes, K. Starbird, and L. Palen, "Microblogging during two natural hazards events: What twitter may contribute to situational awareness," in *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2, pp. 1079–1088, Jan. 2010. (Cited on pages 15, 27, and 39.) ▲

[3] P. Meier, *Digital Humanitarians: How Big Data Is Changing the Face of Humanitarian Response*. USA: CRC Press, Inc., 2015. (Cited on pages 15, 16, 27, and 35.) ▲

[4] C. Castillo, *Big Crisis Data: Social Media in Disasters and Time-Critical Situations*. USA: Cambridge University Press, 1st ed., 2016. (Cited on pages 15, 16, and 27.) ▲

[5] D. T. Nguyen, K. Al-Mannai, S. R. Joty, H. Sajjad, M. Imran, and P. Mitra, "Rapid classification of crisis-related data on social networks using convolutional neural networks," *CoRR*, vol. abs/1608.03902, 2016. (Cited on pages 15, 17, 36, and 40.) ▲

[6] H. Gao, G. Barbier, and R. Goolsby, "Harnessing the crowdsourcing power of social media for disaster relief," *Intelligent Systems, IEEE*, vol. 26, pp. 10–14, July 2011. (Cited on page 15.) ▲

[7] S. Perng, M. Büscher, L. Wood, R. Halvorsrud, M. Stiso, L. Ramirez, and A. Al-Akkad, "Peripheral response: Microblogging during the 22/7/2011 norway attacks," *International Journal of Information Systems for Crisis Response and Management*, vol. 5, pp. 41–57, Jan. 2013. (Cited on page 15.) ▲

[8] F. Alam, F. Ofli, M. Imran, and M. Aupetit, "A twitter tale of three hurricanes: Harvey, irma, and maria," *CoRR*, vol. abs/1805.05144, 2018. (Cited on page 15.) ▲

[9] M. Imran, F. Alam, U. Qazi, S. Peterson, and F. Ofli, "Rapid damage assessment using social media images by combining human and machine intelligence," *CoRR*, vol. abs/2004.06675, 2020. (Cited on pages 16 and 17.) ▲

[10] H. Mouzannar, Y. Rizk, and M. Awad, "Damage identification in social media posts using multimodal deep learning," in *ISCRAM 2018 Conference Proceedings – 15th International Conference on Information Systems for Crisis Response and Management*, pp. 529–543, 2018. (Cited on pages 17 and 55.) ▲

[11] F. Alam, F. Ofli, and M. Imran, "Crisismmd: Multimodal twitter datasets from natural disasters," in *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM)*, June 2018. (Cited on pages 17, 53, 54, and 55.) ▲

[12] D. T. Nguyen, S. R. Joty, M. Imran, H. Sajjad, and P. Mitra, "Applications of online deep learning for crisis response using social media information," *CoRR*, vol. abs/1610.01030, 2016. (Cited on page 17.) ▲

[13] F. Alam, H. Sajjad, M. Imran, and F. Ofli, "Crisisbench: Benchmarking crisis-related social media datasets for humanitarian information processing," in *15th International Conference on Web and Social Media (ICWSM)*, 2021. (Cited on pages 17, 39, and 41.) ▲

[14] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, and P. Meier, "Extracting information nuggets from disaster- related messages in social media," in *ISCRAM 2013 Conference Proceedings – 10th International Conference on Information Systems for Crisis Response and Management*, pp. 791–801, May 2013. (Cited on page 17.) ▲

[15] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, and P. Meier, "Practical extraction of disaster-relevant information from social media," in *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13 Companion, (New York, NY, USA), pp. 1021–1024, Association for Computing Machinery, 2013. (Cited on pages 17 and 41.) ▲

[16] S. R. Chowdhury, M. Imran, M. Asghar, S. Amer-Yahia, and C. Castillo, "Tweet4act: Using incident-specific profiles for classifying crisis-related messages," in *ISCRAM 2013 Conference Proceedings – 10th International Conference on Information Systems for Crisis Response and Management*, pp. 834–839, 2013. (Cited on pages 17, 39, 42, and 43.) ▲

[17] F. Chan, G. Mitchell, O. Adekola, and A. Mcdonald, "Flood risk in asia's urban mega-deltas drivers, impacts and response," *Environment and Urbanization Asia*, vol. 3, pp. 41–61, Feb. 2013. (Cited on page 19.) ▲

[18] C. A. Ohl and S. Tapsell, "Flooding and human health," *BMJ*, vol. 321, pp. 1167–1168, Nov. 2000. (Cited on page 19.) ▲

[19] M. Ahern, R. S. Kovats, P. Wilkinson, R. Few, and F. Matthies, "Global Health Impacts of Floods: Epidemiologic Evidence," *Epidemiologic Reviews*, vol. 27, no. 1, pp. 36–46, July 2005. (Cited on page 19.) ▲

[20] M. Tominaga, "Case study of the social impact of flood," *IFAC Proceedings Volumes*, vol. 31, no. 28, pp. 69–74, 1998. IFAC Workshop on Control in Natural Disasters (CND'98), Tokoyo, Japan, 21-22 Sept. 1998. (Cited on page 19.) ▲

[21] A. C. Yu, "Fukuchiyama city - japanese wiki corpus." https://www.japanese-wiki-corpus.org/geographical/Fukuchiyama%20City.html, Feb. 2021. (Cited on page 19.) ▲

[22] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019. (Cited on pages 21, 39, 64, and 71.) ▲

[23] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. (Cited on page 23.) ▲

[24] G. H. Dunteman, *Principal Components Analysis*. Quantitative Applications in the Social Sciences, Thousand Oaks, CA: SAGE Publications, July 1989. (Cited on page 23.) ▲

[25] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means clustering algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979. (Cited on page 23.) ▲

[26] X. Jin and J. Han, "K-medoids clustering.," in *Encyclopedia of Machine Learning* (C. Sammut and G. I. Webb, eds.), pp. 564–565, Springer, 2010. (Cited on pages 23 and 107.) ▲

[27] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, "Jupyter notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (F. Loizides and B. Schmidt, eds.), pp. 87 – 90, IOS Press, 2016. (Cited on page 24.) ▲

[28] A. Quintero, "React: the riskmap evaluation and coordination terminal," Master's thesis, Massachusetts Institute of Technology, Sept. 2019. (Cited on pages 25, 31, and 55.) ▲

[29] M. P. Dharmapuri Sridhar, "Real-time flood mapping for disaster management decision support in chennai," Master's thesis, Massachusetts Institute of Technology, June 2017. (Cited on pages 27 and 28.) ▲

[30] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *CoRR*, vol. abs/1708.05148, 2017. (Cited on page 32.) ▲

[31] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the International Workshop on Multimedia Information Retrieval*, MIR '07, (New York, NY, USA), pp. 197–206, Association for Computing Machinery, 2007. (Cited on page 32.) ▲

[32] J. Rogstadius, M. Vukovic, C. Teixeira, V. Kostakos, E. Karapanos, and J. Laredo, "Crisistracker: Crowdsourced social media curation for disaster awareness," *Ibm Journal of Research and Development*, vol. 57, Sept. 2013. (Cited on page 35.) ▲

[33] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg, "AIDR," in *Proceedings of the 23rd International Conference on World Wide Web - WWW '14 Companion*, (New York, New York, USA), ACM Press, 2014. (Cited on pages 35, 44, 45, and 55.) ▲

[34] D. T. Nguyen, F. Ofli, M. Imran, and P. Mitra, "Damage assessment from social media imagery data during disasters," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017 - ASONAM '17*, (New York, New York, USA), ACM Press, 2017. (Cited on pages 36, 37, 38, 52, and 55.) ▲

[35] J. Liu, T. Singhal, L. T. Blessing, K. L. Wood, and K. H. Lim, "Crisisbert: A robust transformer for crisis classification and contextual crisis embedding," in *Proceedings of the 32nd ACM Conference on Hypertext and Social Media*, HT '21, (New York, NY, USA), pp. 133–141, Association for Computing Machinery, 2021. (Cited on page 36.) ▲

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, June 2009. (Cited on page 37.) ▲

[37] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31st International Conference on Machine Learning* (E. P. Xing and T. Jebara, eds.), vol. 32 of *Proceedings of Machine Learning Research*, (Bejing, China), pp. 647–655, PMLR, 22–24 June 2014. (Cited on page 37.) ▲

[38] Z. Li and D. Hoiem, "Learning without forgetting," *CoRR*, vol. abs/1606.09282, 2016. (Cited on pages 37 and 38.) ▲

[39] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. (Cited on pages 38, 44, and 45.) ▲

[40] C. Caragea, A. Silvescu, and A. H. Tapia, "Identifying informative messages in disasters using convolutional neural networks," in *ISCRAM 2016 Conference*

Proceedings - 13th International Conference on Information Systems for Crisis Response and Management, 2016. (Cited on page 39.) ▲

[41] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. (Cited on page 40.) ▲

[42] J. Fleiss *et al.*, "Measuring nominal scale agreement among many raters," *Psychological Bulletin*, vol. 76, no. 5, pp. 378–382, 1971. (Cited on pages 40, 59, 60, and 145.) ▲

[43] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *CoRR*, vol. abs/1607.01759, 2016. (Cited on page 41.) ▲

[44] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. (Cited on pages 41 and 98.) ▲

[45] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter," *CoRR*, vol. abs/1910.01108, 2019. (Cited on page 41.) ▲

[46] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. (Cited on page 41.) ▲

[47] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Mar. 2003. (Cited on page 42.) ▲

[48] J. Mizuno, M. Tanaka, K. Ohtake, J.-H. Oh, J. Kloetzer, C. Hashimoto, and K. Torisawa, "WISDOM X, DISAANA and D-SUMM: Large-scale NLP systems for analyzing textual big data," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, (Osaka, Japan), pp. 263–267, The COLING 2016 Organizing Committee, Dec. 2016. (Cited on pages 43 and 44.) ▲

[49] M. Imran, C. Castillo, J. Lucas, P. Meier, and J. Rogstadius, "Coordinating human and machine intelligence to classify microblog communications in crises," in *ISCRAM 2014 Conference Proceedings – 11th International Conference on Information Systems for Crisis Response and Management*, pp. 712–721, May 2014. (Cited on pages 44 and 45.) ▲

[50] F. Alam, F. Ofli, and M. Imran, "Descriptive and visual summaries of disaster events using artificial intelligence techniques: case studies of hurricanes harvey, irma, and maria," *Behaviour & Information Technology*, vol. 39, pp. 1–31, May 2019. (Cited on page 46.) ▲

[51] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, (Seattle, Washington, USA), pp. 1631–1642, Association for Computational Linguistics, Oct. 2013. (Cited on page 47.) ▲

[52] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, (Baltimore, Maryland), pp. 55–60, Association for Computational Linguistics, June 2014. (Cited on page 47.) ▲

[53] B. Barz, K. Schröter, M. Münch, B. Yang, A. Unger, D. Dransch, and J. Denzler, "Enhancing flood impact analysis using interactive retrieval of social media images," *CoRR*, vol. abs/1908.03361, 2019. (Cited on pages 51, 56, 57, and 58.) ▲

[54] B. Barz, K. Schröter, A. Kra, and J. Denzler, "Finding relevant flood images on twitter using content-based filters," *CoRR*, vol. abs/2011.05756, 2020. (Cited on pages 51, 57, and 58.) ▲

[55] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960. (Cited on pages 60, 67, and 147.) ▲

[56] W. A. Scott, "Reliability of content analysis: The case of nominal scale coding," *Public opinion quarterly*, pp. 321–325, 1955. (Cited on page 60.) ▲

[57] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *CoRR*, vol. abs/1712.04621, 2017. (Cited on page 62.) ▲

[58] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS 2017 Workshop on Autodiff*, 2017. (Cited on pages 64, 71, and 112.) ▲

[59] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, Dec. 2014. (Cited on page 66.) ▲

[60] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. (Cited on pages 71, 97, 103, 112, 115, 116, 117, 118, and 137.) ▲

[61] P. Umesh, "Image processing in python," *CSI Communications*, vol. 23, 2012. (Cited on page 71.) ▲

[62] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference* (Stéfan van der Walt and Jarrod Millman, eds.), pp. 56–61, 2010. (Cited on pages 71 and 112.) ▲

[63] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020. (Cited on pages 71 and 112.) ▲

[64] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. (Cited on pages 71 and 112.) ▲

[65] M. L. Waskom, "seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021. (Cited on pages 71 and 112.) ▲

[66] S. Seabold and J. Perktold, "statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010. (Cited on page 71.) ▲

[67] A. Rosen and I. Ihara, "Giving you more characters to express yourself." https://blog.twitter.com/en$_u$s/topics/product/2017/*Giving − you − more − characters − to − express − yourself*, Sept. 2017. (Cited on page 92.) ▲

[68] P. McCann, "fugashi, a tool for tokenizing Japanese in python," in *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, (Online), pp. 44–51, Association for Computational Linguistics, Nov. 2020. (Cited on pages 96 and 112.) ▲

[69] A. Rajaraman and J. D. Ullman, *Data Mining*. Cambridge University Press, 2011. (Cited on page 97.) ▲

[70] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, Mar. 2003. (Cited on page 98.) ▲

[71] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. (Cited on page 98.) ▲

[72] C. J. V. Rijsbergen, *Information Retrieval*. USA: Butterworth-Heinemann, 2nd ed., 1979. (Cited on page 100.) ▲

[73] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, New York, NY, USA: Springer New York Inc., 2001. (Cited on pages 101 and 109.) ▲

[74] G. C. Cawley and N. L. C. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, no. 70, pp. 2079–2107, 2010. (Cited on pages 101 and 102.) ▲

[75] J. Wainer and G. C. Cawley, "Nested cross-validation when selecting classifiers is overzealous for most practical applications," *CoRR*, vol. abs/1809.09446, 2018. (Cited on page 102.) ▲

[76] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, (Online), pp. 38–45, Association for Computational Linguistics, Oct. 2020. (Cited on page 112.) ▲

[77] E. Loper and S. Bird, "Nltk: The natural language toolkit," *CoRR*, vol. cs.CL/0205028, 2002. (Cited on page 112.) ▲

[78] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PLOS ONE*, vol. 10, pp. 1–21, Mar. 2015. (Cited on pages 117 and 118.) ▲

[79] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: An overview." arXiv:2008.05756, Aug. 2020. (Cited on page 149.) ▲