FLARE: Federated Active Learning Assisted by Naming for Responding to Emergencies

Viyom Mittal
University of California, Riverside
vmitt003@ucr.edu

Mohammad Jahanian
University of California, Riverside
mjaha001@ucr.edu

K. K. Ramakrishnan
University of California, Riverside
kk@cs.ucr.edu

ABSTRACT

During disasters, it is critical to deliver emergency information to appropriate first responders. Name-based information delivery provides efficient, timely dissemination of relevant content to first responder teams assigned to different incident response roles. People increasingly depend on social media for communicating vital information, using free-form text. Thus, a method that delivers these social media posts to the right first responders can significantly improve outcomes. In this paper, we propose FLARE, a framework using 'Social Media Engines' (SMEs) to map social media posts (SMPs), such as tweets, to the right names. SMEs perform natural language processing-based classification and exploit several machine learning capabilities, in an online real-time manner. To reduce the manual labeling effort required for learning during the disaster, we leverage active learning, complemented by dispatchers with specific domain-knowledge performing limited labeling. We also leverage federated learning across various public-safety departments with specialized knowledge to handle notifications related to their roles in a cooperative manner. We implement three different classifiers: for incident relevance, organization, and fine-grained role prediction. Each class is associated with a specific subset of the namespace graph. The novelty of our system is the integration of the namespace with federated active learning and inference procedures to identify and deliver vital SMPs to the right first responders in a distributed multi-organization environment, in real-time. Our experiments using real-world data, including tweets generated by citizens during the wildfires in California in 2018, show our approach outperforming both a simple keyword-based classification and several existing NLP-based classification techniques.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; • **Networks** → *Network protocols.*

ACM Reference Format:
Viyom Mittal, Mohammad Jahanian, and K. K. Ramakrishnan. 2021. FLARE: Federated Active Learning Assisted by Naming for Responding to Emergencies. In *8th ACM Conference on Information-Centric Networking (ICN '21), September 22–24, 2021, Paris, France.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3460417.3482978

This work is licensed under a Creative Commons Attribution International 4.0 License.

ICN '21, September 22–24, 2021, Paris, France
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8460-5/21/09.
https://doi.org/10.1145/3460417.3482978

1 INTRODUCTION

In managing disasters, we have seen that it is important to have timely and relevant information delivered to the right people, especially first responders. Social media platforms are increasingly being depended on, both by authorities and the general public to report and exchange disaster-related issues [26]. This has been especially helpful when traditional means of disaster management, *e.g.*, emergency calling (911 in the U.S.), get overloaded during the disaster. However, such social-media interactions are typically adhoc and unorganized [30]. To address this, we seek to develop a framework that allows for meaningful and accurate communication with first responders through social media.

An effective and efficient way to provide such structured dissemination of content is in conjunction with naming [33], which guides information-centric and name-based delivery (either request/response or pub/sub [9, 20]) to large groups of people. A namespace, often a hierarchical structure, provides a robust interface for all participants. We use an incident-specific namespace, which captures the different incident-related roles and their relations, to be used to indicate interests and publication IDs [8]. Most people, *i.e.*, social media users, are likely to have little or no knowledge of the incident namespace required to create a named publication or 'interest' (as in Named-data networking [33]). They would likely prefer to use social media in their common forms, *i.e.*, with free-form text. Therefore, it would be helpful to have those social media posts (SMPs) be mapped to the right subset of the incident namespace, leading to the right first responders receiving those SMPs so they can deal with the specific task related to the incident. There have been many works that classify text in Tweets (*e.g.* on Twitter) from disasters [14, 16, 19]. Apart from these, Twitter has proprietary machine learning methods for classification of tweets into various Topics[1]. Twitter also has an option of using hashtags (#), provided by the user based on their own knowledge of what would be appropriate tags. If first responders follow these tagged tweets, they would receive them. Also, since the users are free to post any information under these hashtags, there is still the need to filter out the irrelevant tweets with minimal burden on the first responders. Our work aims specifically at disaster management and focuses on filtering out the relevant tweets, classifying them in accordance with the incident namespace and delivering them to the right first responders, without requiring the users to have any knowledge of the incident namespace. We seek to leverage various state-of-the-art classification techniques so that the tweets/social media posts can be disseminated in real-time using the name-based pub/sub framework to deliver them to the right first responders.

[1]https://help.twitter.com/en/using-twitter/follow-and-unfollow-topics

In this paper, we propose FLARE (Federated active Learning Assisted by naming for Responding to Emergencies), a framework to coordinate disaster response among the many different participants using social media, and a name-based information dissemination architecture. A key notion in FLARE is a Social Media Engine (SME), which maps free-form social media posts to the right part of the namespace, using a combination of different Machine Learning (ML) and natural language processing (NLP) techniques to address various issues that typically arise during the classification of SMPs in the event of a disaster. To reduce the manual labeling effort and to deal with the scarcity of training data in the beginning, we use stream-based active learning [28], helped by dispatchers with domain-knowledge for selective manual labeling. To allow for load-sharing without requiring the sharing of datasets among all nodes, while exploiting the expertise of individual first-responder departments, we employ multiple SMEs and allow for their learning procedure to be performed in federated manner. In our framework, each SME is associated with a particular department, employing its specialized dispatchers, and crawls its own set of keyword-based SMPs. In addition to regular federated learning messaging (*i.e.*, worker-to-master and vice versa), we add support for messaging between SMEs, using name-based forwarding, which allows us to pass free-form text messages (*i.e.*, SMPs) to be processed and potentially labeled, by the SME specialized in the category most relevant to the SMP. To provide the most complete names, with all the different name levels of the namespace hierarchy, an SMP goes through a set of 3 classifiers: C1 (incident relevance predictor); C2 (organization predictor); and C3 (fine-grained role predictor). The labels in these classifiers directly correspond to the levels in the namespace. Through various experiments, we demonstrate how our approach achieves good accuracy, while not inducing too much messaging overhead and labeling effort. During a disaster, the incident namespace involving the various participating organizations may evolve as the incident evolves. We demonstrate the ability of FLARE's C3 classifier to adapt to the namespace changes without a significant additional labeling or loss in accuracy.

A key novelty of FLARE is its exploitation of naming both for dissemination of the publication as well as for its federated classification/learning procedures. A namespace (or a set of inter-related namespaces) works as a common interface across FLARE's procedures and components, yielding significant benefits. It enables an SME to hand an SMP, based on its initial classification result (class label is a "name"), to the right SME, which has the specialization to process and potentially label it (recipient SME's department is a "name") achieving holistic cooperative active learning among SMEs. Another aspect of FLARE is the cascade of classifiers, C1, C2, and C3, each corresponding to levels of the namespace. As an SMP goes through different classifiers one by one, it gets mapped to increasingly finer granularity nodes (*i.e.*, towards the leaves) of the namespace. This structure allows for inaccurately-mapped SMPs to still have a chance to be delivered to first responders that are "somewhat relevant" (*i.e.*, within the same department, or namespace subset), even if not the precise first responder, thus greatly benefiting the goal of emergency management: saving lives. In this paper we show how FLARE, our name-assisted approach assists and improves the accurate, real-time delivery of SMPs.

The primary job of FLARE is to create and assign the right name to a piece of content (*i.e.*, text or social media post). The only requirement for FLARE is that there should be an underlying network that recognizes those names, and the users interested in those names will receive those contents. It can run on various NDN-based information dissemination mechanisms (without any change to the underlying forwarding mechanisms), such as group-based dataset synchronization solutions (e.g., ChronoSync [38] and PSync [34]), periodic query/response solutions (as in NDN [33]), or pub/sub solutions with long-standing subscription interests ([9] and [8]). We leverage CNS[8] because it enables dynamic modifications of the namespace to reflect the dynamic nature and evolution of incident management (and thereby the roles of individuals) during a disaster.

The contributions of this paper are: 1) a system integrating the main actors in disaster response, *i.e.*, first responders and the general public, using social media, 2) a distributed set of social media engines, as intelligent systems, that cooperatively map free-form social media posts to the right names for publication, 3) an incident naming framework that integrates dissemination (whether request/response or pub/sub) with classification/learning procedures, 4) a federated and stream-based active learning approach for classification that enables our SMEs to be load-shared while not sharing crawled raw data, and be effective in an online manner even when we have little or no labeled data to begin with, and 5) evaluation of the effectiveness of our approach with actual, publicly available, disaster-related Tweets, such as those collected from the California Wildfires of 2018.

2 BACKGROUND AND RELATED WORK

Many studies have pointed out the need for a common means of communication between first responders and citizens (and for that matter communication between first responders from different organizations), for a structured coordination of disaster response [12, 30]. In the recent years, a significant interest has gone into information-centric (ICN) [33] paradigms which provides important benefits of location-independence, and using it for disaster management related communication. It enables timely delivery of content and seamless mobility [27], [3]. In name-based delivery, each publication message needs to have the correct name, in order to reach all the relevant/intended recipients. This can be challenging in disaster scenarios, where many victims (civilians) may not know or have access to the namespace. Fuzzy Interest Forwarding (FIF) [7] addresses the issue of lack of access to the namespace by proposing an NLP-based mapping of requests to names for Interest forwarding, for request/response exchanges. In this work, we explore further by analyzing the whole message content to assign the right name to a message. As observed in many recent disasters [26], social media gets widely used to generate and disseminate information regarding a variety of issues related to incident response. While [15] addresses this with a centralized NLP/ML-based mapping of social media posts to the right names, it requires manually labeled fixed training sets. In FLARE, we allow for a federated, and importantly an active learning method. We also utilize multi-level classification and namespace integration for much more accurate and targeted delivery to the right name.

Several works have studied the classification of disaster-related social media posts (SMPs) using NLP techniques. [4] provided a

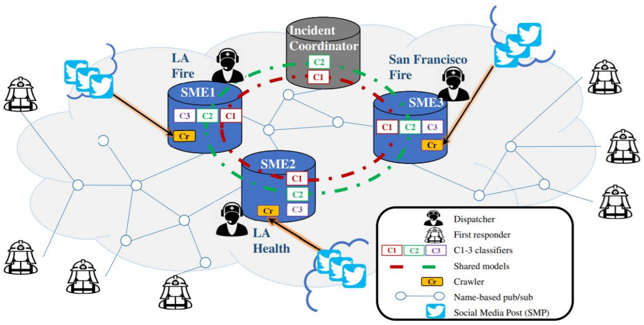


Figure 1: Architecture of FLARE

comparison of various learning-based methods like Naive Bayes, SVM, KNN, Logistic Regression, Decision Tree and Random Forest for identifying disaster related tweets. [32] uses a Random Forest classifier to extract firsthand and credible information from the disaster related tweets. [14] provides human-annotated datasets for 19 different disasters and classifies the SMPs using Naive Bayes, SVM and Random Forest. Work in [19] improves the performance of such classifications by developing deep neural network (DNN) based models. In [16], the authors propose a DNN model with attention layer and auxiliary features to further improve the model. All the approaches discussed above are primarily focused on classification and are agnostic on how to efficiently label data as they come in, as well as fallback labeling and dissemination mechanisms to improve the automatic classification. FLARE addresses those issues by integrating online learning and classification with a supporting framework of dispatchers as knowledgeable human labelers and a name-based pub/sub framework. We seek to classify SMPs in a distributed manner, taking advantage of each organization's specialized knowledge, for real-time delivery. For classification, previous works have examined the use of word embedding techniques for encoding incoming tweets into word vectors [16, 19]. However, the state of the art sentence embedding techniques like InferSent [10], Universal Sentence Encoder (USE) [6] and SBERT [24] outperform word embedding techniques like BERT [11] and GloVe [22] due to the ability of the former to capture semantic relationships among the sentences more accurately [24]. In FLARE, we utilize USE for its ease of use and ability to process the tweets directly without substantial preprocessing.

Active learning methods can reduce manual labeling effort, starting with an initial seed of labeled data, and expanding to a larger labeled training set until a stopping criteria is reached. They require only a subset of manually labeled data. Active learning can be pool-based or stream-based. Pool-based active learning [13, 35] iteratively picks the most informative sample set from the whole dataset, and uses a human labeler to assign a class to them, and then be subject to re-training. The challenge with pool-based active learning is that it requires the whole corpus to be available. This can be a difficult in a real-world disaster where initial data may not be available. Stream-based active learning methods [23, 28] enable data instances, *e.g.*, tweets, to be processed one at a time (or in a batch) as they come in. We adopt stream-based active learning for online training and real-time processing.

Federated learning (FL) [18] allows the learning of a model across multiple clients or entities. FL is typically used for the purpose of preserving the privacy of user data, as sub-data sets stay local to users who generate it. In our scenario, data, *i.e.*, social media posts, are likely to be public and privacy is not a primary concern. However, FLARE employs FL because of its other important feature: enabling the assimilation and use of the specialized knowledge of different entities (e.g., departments in the incident management) and train ing the classifiers collectively. Work in [2] proposes active learning along with FL to classify images into different types of disasters. FLARE uses a combination of active and federated learning for SMP classification.

We also mitigate accuracy losses that may result from an initial simple, but inaccurate partitioning of input SMPs, possibly during a triage of disaster reports, by automatically detecting and transferring the mis-routed SMPs to the relevant learner/organization to leverage their organizational expertise using a message passing scheme.

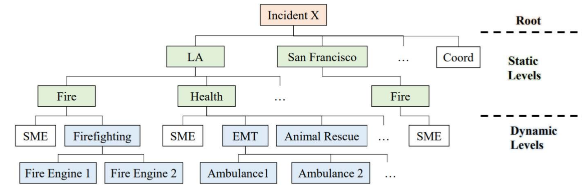


Figure 2: Incident namespace

3 ARCHITECTURE AND DESIGN

An overview of the architecture of FLARE and its primary actors and components are shown in Fig. 1. FLARE's goal is to effectively disseminate social media posts among first responders. It relies heavily on integrating name-based dissemination and machine learning algorithms and classifiers.

3.1 Name-based Pub/Sub in FLARE

The communication between entities in FLARE is name-based (to enable information centricity [33]), used both for disseminating instructions and reports between first responders, and also learning/inference-related information sharing among SMEs. A unified namespace, such as the one in Fig. 2, guides this information-centric framework, and is a common interface that various primary actors and components (*i.e.*, first responders, SMEs, and Incident Coordinator *but not* all the other social media users) have a local copy of, and can use to indicate which subset of the incident namespace they are interested in (*i.e.*, subscribed to), and want to deliver to.

First responders, based on their assigned task (either based on the incident or as part of their organizational hierarchy), subscribe to a *prefix* in the namespace. For example, a firefighter *FF1* assigned with tasks related to *"Fire Engine 1"* will subscribe to "/IncidentX/LA/Fire/Firefighting/FireEngine1". The subscription can be at any desired granularity: *e.g.*, another firefighter, let us call him/her *FF2*, who manages *all fire engines in LA*, will subscribe to "/IncidentX/LA/Fire/Firefighting". Any publication $P1$ with a prefix "/IncidentX/LA/Fire/Firefighting" will be delivered to *FF2*. Additionally, FLARE expands the delivery according to the name hierarchy to reach all relevant recipients.

To provide that capability in a flexible and efficient manner, here we use an NDN architecture that is enhanced with recipient-based pub/sub logic (as proposed in [8]). This forwarding logic matches an incoming packet with *all* entries that contain the packet's name as a prefix. Thus, $P1$ will be delivered to the subscribers of names with the prefix containing "/IncidentX/LA/Fire/Firefighting", including *FF1* as well. In addition to the prefix, the publication names in FLARE can include other attributes as well, such as *tags* and an arbitrary number of input *parameters*. The named-SMP (NSMPs) containing instructions for first responders will be published (typically by SMEs) with the following name format: "/[*prefix*]/tag=instr/[*params*]". The *prefix* (*e.g.*, "/IncidentX/LA/Fire/Firefighting") will be used to forward the publication to the right recipients. The *tag* is used to indicate the type of the message payload (*e.g.*, *instr*). The *params* are input parameters used by the recipients to process the received information, as in "$.../param1 = value1/param2 = value2/...$".

FLARE also uses name-based delivery for message exchange among SMEs in its Federated Learning-based procedures, integrating the namespace with different classifiers. Each SME is organizationally in charge of processing and manually labeling SMPs associated with certain prefixes, which they subscribe to. *E.g.*, $SME1$ in Fig. 1 subscribes to "/IncidentX/LA/Fire/SME". The dispatcher with $SME1$ has specialized domain knowledge to label SMPs regarding "Fire" and its sub-categories, *e.g.*, "SurvivalSearch", *etc.* There are three types of SME-related messages:

1) SME-to-SME (S2S) messaging: An SME asking any SME or a particular SME with the prefix p to process an SMP (for calculating, labeling, *etc.*): "/[p]/tag=proc/[$params$]", where $params$ can include SMP ID, confidence values, *etc.* The payload of this message type is SMP content (*e.g.*, a tweet json).

2) SME-to-Incident Coordinator (S2I) messaging: An SME sends its processing result from training to incident coordinator for aggregation: "/IncidentX/Coord/ tag=result/[$params$]", where $params$ can include which classifier it includes the result it is associated with (whether C1 or C2). The payload of this message type is the calculation result, *e.g.*, weights.

3) I2S messaging: The incident coordinator distributing fully trained models to a set of SMEs under prefix p for synchronization: "/[p]/tag=model/[$params$]", where $params$ can include the classifier, version number, *etc.* The payload of this message is the most recent, fully trained model, after the incident coordinator's aggregation procedure is completed.

The namespace has a hierarchical structure (as a prefix tree). The hierarchical levels are divided into three levels, namely root, static, and dynamic levels to capture both how the namespace nodes are managed, and their relationship with the SME classification procedures. The root identifies the name of the namespace. The static levels follow the organizational/incident command structure, based on a template, created at the beginning, when the response to the incident is initiated and organized. They remain static during the disaster's management. The dynamic levels represent more fine-grained roles under the static levels. Name nodes within the dynamic levels can be created, modified or removed as the disaster unfolds, based on command decisions.

3.2 Learning and Inference in FLARE

The role of SMEs in FLARE is to map incoming SMPs to the right names. To this end, each SMP goes through a pipeline of three levels of increasingly detailed classifiers in order, namely: C1 (Incident relevance predictor); C2 (Organization predictor); and C3 (Fine-grained role predictor). The summary of the characteristics of the three classifiers are provided in Table 1. They are preceded by a crawler component, and are followed by a NSMP generator. All three classifiers are text classifiers comprising natural language pipelines, including word embedding techniques. Each classifier is associated with respective levels of the namespace, with distinct learning methods. C1 is a binary classifier, and predicts if an SMP is relevant to the incident. C2 and C3 predict categories associated with static and dynamic-level names respectively. C1 and C2 are shared and the SMEs cooperate in a federated manner and their classes are assumed to be static during the disaster, in contrast to C3. All the classifiers, C1–3, use active learning to reduce labeling effort with the help of dispatchers who are human labelers. Manual labeling for C1 can be done by any dispatcher, while C2 and C3 must be labeled by the specific dispatchers with SME-specific organizational expertise. We now go into these in greater detail.

3.2.1 Crawler and C1 Classifier (Relevance Predictor).

Crawler: Each SME is equipped with a crawler component which collects or crawls SMPs in real-time during the disaster. Each SME has its own specialization; thus, it may want to collect SMPs that can potentially match its geographic/topic expertise. For example, *SME1* in Fig. 1 ("LA Fire") will look for SMPs limited to the LA area, and contain potentially fire-related keywords, *e.g.*, "fire", "flame", "burn", *etc.* Current social media platforms, such as Twitter, provide APIs to facilitate such crawling. However, with the integration of social media platforms with SMEs, SMEs can be notified of every new SMP through a push notification. Note that our crawler is a generic module, and could be a combination of API-specific modules implemented for multiple participating social media platforms. The combination of all of the crawled SMPs, across all SMEs, constitute the data set for learning and inference in FLARE. A benefit of FLARE's FL-based framework is that the crawled data (which can be a very large number of tweets/messages) need only stay locally at the SME it was originally received at for the most part, for learning purposes. An SMP only needs to be shared when it needs to be forwarded to another SME which has a better specialization to handle it (explained in §3.2.2). This decentralized crawling also allows each SME to change its crawling criteria throughout the disaster, which is helpful as new terms may grow in usage and also need more attention from first responders. Crawler uses keywords to filter out SMPs that are clearly not disaster-related, so as to not burden first responders with "all" generated SMPs in an area. Each of these collected SMPs will be then processed through various classifiers. The use of federated learning in FLARE spreads the processing workload of SMP-to-name mapping across multiple organizations and SMEs, which can potentially make the system more scalable, as the number of generated SMPs grow.

C1 Classifier: The C1 module at the SME determines if the SMP belongs to any prefix under the incident namespace root (*e.g.*, "/IncidentX" in Fig. 2) or not; in other words, whether the SMP is *disaster-relevant* or not. The primary aim of the C1 classification module is to reduce the number of false-negatives, to make sure that the maximum number of relevant SMPs get classified correctly. Fig. 3 shows the C1 and C2 classifiers and their interaction with other components. First, in the FL procedure for C1, each SME trains its C1 model locally using its local training pool. In FLARE, we use

Table 1: Summary of FLARE classifiers and their attributes

Classifier	Prediction	# of classes	Class Dynamicity	Namespace Part for classes	Cooperative/ Local	Federated Learning	Active Learning	Labeler Dispatchers
C1	Incident relevance	Binary	Static	Root	Cooperative	Yes	Yes	Any
C2	Organization	Multi-class	Static	Static Levels	Cooperative	Yes	Yes	SME-specialized
C3	Fine-grained role	Multi-class	Dynamic	Dynamic Levels	Local	No	Yes	SME-specialized

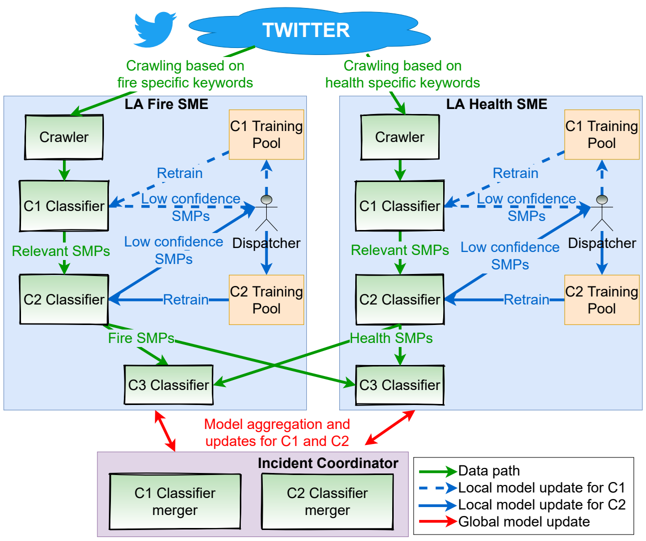


Figure 3: Federated learning for C1 and C2 classifiers

dispatcher-assisted active learning approach for the local training of the model. For this purpose, we check the confidence of the initial classifications: the SMPs with low confidence are passed to the dispatcher for manual labeling (more details in §4). We assume all dispatchers, regardless of department, have enough knowledge and expertise to determine whether an SMP is disaster-relevant or not. The FL aggregation procedure (as in [18]) occurs periodically, where each SME shares its training result with the Incident Coordinator by using the *result* tag in messages. The role of the Incident Coordinator is to aggregate these local calculations based on iterative model averaging to create a global model. This is shared back with all SMEs using the *model* tag in the messages, for synchronization. The role of Incident Coordinator can be performed by one of the organizational SMEs, or can be a separate physical server.

Whenever the C1 classifier receives a batch of SMPs, it classifies them as either *Relevant* or *Irrelevant* and forwards all the *Relevant* tweets to the C2 unit for further classification. It also picks *all* the low confidence SMPs, both Relevant and Irrelevant ones, and passes them to the dispatcher on the same SME, for manual labeling. From these low confidence SMPs, those labeled by the dispatcher as being *Relevant* SMPs "will get a second chance": they will be passed to the C2 classifier. This ensures that SMPs that might have been initially mis-classified as *Irrelevant* by C1 will also reach C2, as a result of the manual labeling by the dispatcher. Additionally, all the SMPs labeled by the dispatcher are added to the C1 training pool. This process ensures that SMPs are not erroneously mis-classified as *Irrelevant* by C1's *initial classification*, *i.e.*, before the dispatcher's help. Having said that, a small portion of relevant SMPs may not be passed on to the next stages if they are initially deemed Irrelevant with high confidence. However, as we show in our evaluation results in §5, those SMPs are a very small percentage, and typically include 'borderline' relevant SMPs, *i.e.*, they express emotions and opinions, without actionable tasks for first responders. For all the classifiers, we use a simple variant of a deep neural network model, which has proven to be effective in the past for text classification for disaster related data [16, 19].

3.2.2 *C2 (Organization predictor).*

The C2 classifier receives the SMPs classified as *Relevant* from the C1 classifier and the dispatcher. It uses *smpID* to avoid redundant classification for the SMPs that have been marked as *Relevant* by both C1 and the dispatcher. The C2 classifier then maps disaster-relevant tweets to the right department or organization name. Thus, the output of C2 is a prefix in the static parts of the namespace, which is then passed to the C3 classifier, at the SME that is responsible for (and specializes in) that prefix. Fig. 3 illustrates the C2 classifier along with its learning mechanisms and interaction with other components. Similar to C1, the C2 classifier uses dispatcher-assisted active learning for making local updates to the model. The global aggregation is done by the Incident Coordinator using the *model* and *result* tagged messages. Unlike C1, the dispatcher might not be as well-informed about all the other departments for C2 and thus not able to correctly label the SMP belonging to other departments. In FLARE, we have specialized dispatchers and thus SMEs with expertise in their respective departments. In such a case, FLARE uses an SME-to-SME (S2S) message passing technique to forward an SMP to the SME/dispatcher of the correct department using the *proc* tag.

In this message passing technique, if a dispatcher of a source SME cannot classify a low confidence SMP, the model prepares a priority list-based on the prediction probability for all the other departments, *i.e.*, names in the static level of the namespace. This list is used to forward the SMP to an SME that it most likely belongs to. If the dispatcher of that SME is able to classify the SMP accurately, the classification result is sent back to the C2 training pool of the source SME. After successively forwarding the SMP through the list of SMEs, and all of them are unable to classify the SMP, the last SME marks the SMP as *Irrelevant* and forwards it back to the C1 classification pool of the source SME. This scheme is illustrated in Fig. 4 where the Fire SME dispatcher receives a low-confidence SMP which he or she is not able to classify. The model prepares the list of most likely SMEs based on its prediction probabilities. As the list in the figure suggests, the SMP has the highest likelihood of belonging to the Health SME, followed by Police SME. The SMP is forwarded to these SMEs in order. If any of them is able to classify the SMP, the classification is added to the C2 training pool of the Fire SME. If the last SME in the list, *i.e.*, Police SME is not able to classify it as well, then it is added to C1 training pool of the Fire SME as an *Irrelevant* SMP. In either case we update the training pools of the source SME. Since SMPs are fetched based on keywords, the source SME has the highest likelihood of encountering a similar SMP in

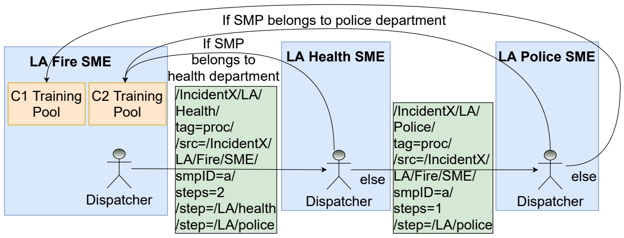


Figure 4: S2S messaging for C2 classification

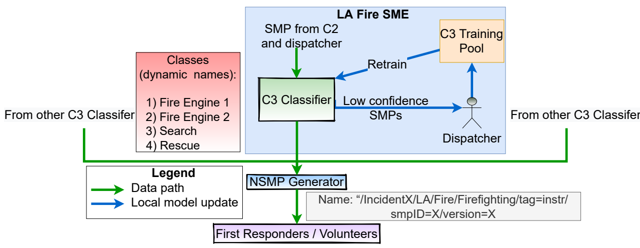


Figure 5: C3 classifier and NSMP generator

the future, and adding the SMP to the training pool of the source SME is likely to be most beneficial.

3.2.3 C3 (Fine-grained role predictor) and NSMP Generator.

C3 Classifier: The C3 classifier aims at mapping SMPs to fine-grained, *i.e.*, longest possible, prefixes, thus also covering the dynamic levels of the namespace. The architecture of C3 and NSMP generator and their detailed interaction with other components is shown in Fig. 5 as well as an example of the finer C3 classes for the Fire department. Such finer granularity classes are defined for each department. We propose a distinct, customizable, C3 classifier based on the detailed classes and requirements for each department. Also, since its classification corresponds to 'dynamic names' which may change during the course of disaster, we only use Active Learning-based training for the classifier, and do not use Federated Learning. The C3 classifier can adapt to the dynamic changes in the incident namespace. Whenever a new name is added to the namespace and the dispatcher receives new SMPs related to that name, he/she can re-instantiate the classifier with an increased number of classes. Active Learning ensures the rapid learning of the classifier, without adding significant labeling load on the dispatcher. The C3 classifier forwards the classification results to the NSMP generator, which uses this information to generate a *NamedSMP* which can be forwarded to the appropriate first responders/volunteers.

NSMP Generator: The Named SMP (NSMP) Generator at each specialized SME uses the output of its C3 classifier to construct the right name to associate with the SMP, as an *instr* in the format described in §3.1. It carries several important parameters, including *smpID* and *version* number. An example name would look like "/IncidentX/LA/Fire/Firefighting/tag=instr/smpID=0/version=0". These NSMP parameters enable it to get transmitted over an ICN (e.g., NDN) to the right first responders subscribed to the corresponding role in the namespace. The *smpID* can be any value uniquely identifying an SMP (*e.g.*, tweet ID) for duplicate detection purposes at the first responder end-device. The *version* is a counter value (starting from 0), showing the version of this SMP's publication. This is important, as we may end up with multiple versions of the

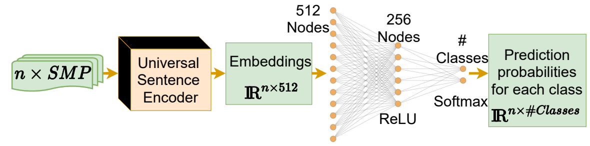


Figure 6: DNN model with USE

same SMP being sent to different first responders, after correcting the errors (hopefully a very small percentage) of past inference results. A version value of n means that this is the n-th duplicate of this SMP's name assignment and publication under this particular static level, *e.g.*, "/IncidentX/LA/Fire" in the example above. The NSMP generator maintains the version values of recently published SMPs in its history. Upon a new name assignment for the same SMP that is different from the previous assignment in the same SME, it publishes the SMP with the version value being incremented.

4 IMPLEMENTATION

We present the implementation details on the major components of the FLARE framework.

Text Embedding. We use the Universal Sentence Encoder (USE) with Transformer architecture [6] for generating the sentence embedding. One of the advantages of USE is its ability to accept the SMPs or the tweets directly without pre-processing. FLARE, however, still does the basic pre-processing involving special character, URL and emoji removal, before feeding the SMPs to the sentence encoder. The motivation behind incorporating such pre-processing in FLARE is that we aim to generate the sentence embedding before feeding it into the DNN model, so that it reduces the model size, and the cost of performing model exchange and aggregation during federated learning.

Classification Model. For classification, we use a NN-based model as these have been shown to perform better for both binary and multi-class classification [19]. We create a DNN model using Tensorflow[1] which is depicted in Fig. 6. We chose this model for its simplicity, while being efficient in classifying SMPs. We avoid models mentioned in [16, 19] due to their complexity and because they have the embedding layer inside the model which makes it compute intensive. Additionally, it increases the cost of model exchange in a FL environment where there is frequent model exchange between the SMEs.

Active Learning. In our model, we use a threshold-based active learning approach for all the classifiers. A batch of SMPs is passed to the classifier. For each SMP, the classifier provides *prediction probabilities* for each of the classes. The class with highest prediction probability is taken as the predicted class for that SMP. In our approach, we use this highest value of prediction probability as a measure of confidence and compare it against a *Threshold*. If the prediction probability value is below the threshold, we say that the classifier has a low-confidence in classifying that SMP. These low-confidence SMPs are selected from the batch and forwarded to the dispatcher for manual labeling. After the labeling is complete, these SMPs are added to the training pool. Once the training pool is updated, it trains the classifier on the new SMPs in an incremental fashion and flushes the data after training is complete.

Federated Learning. For FL, we use the vanilla Federated Averaging algorithm proposed in [18]. We set our system on top of

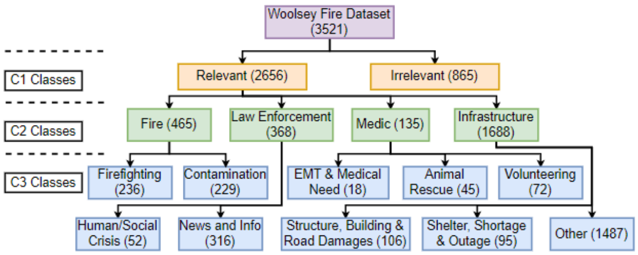


Figure 7: Woolsey dataset C1–3 classes & distributions

Table 2: Hyperparameter values for the DNN model

Hyperparameter	Value/Description
Epochs and batch	Epochs=20-25; batch size = 32
loss model	binary_crossentropy for C1 classification; categorical_crossentropy for C2 classification
Adam optimizer	learning rate = 0.0001; metrics = accuracy
Early Stopping	monitor = Validation accuracy; patience = 3-7; min_delta=0.0001

Flower [5], a framework for FL. We further add S2S message passing in FL environment in FLARE.

5 EVALUATION

5.1 Datasets and Metrics

We use two different sets of data in our experiments:

CrisisNLP/CrisisLex. This dataset is a combination of the CrisisNLP [14] and CrisisLex [21] datasets. It includes tweets collected from the Nepal Earthquake, California Earthquake, Typhoon Hagupit, and Cyclone PAM [19]. We primarily use this data set for the evaluation of our C1 and C2 classifiers and to validate our DNN model by comparing it with the existing state-of-art models. We use the dataset's original labels as provided in [14, 21], for our C1 and C2 classes.

California Wildfires. The second dataset consists of tweets we collected during the relatively recent major California Wildfires, especially the Woolsey Fire (in Southern California) [29] in November 2018. In our experiments, we mainly focus on the peak day of the fire, *i.e.*, Nov 10th. For evaluation purposes, all the tweets in the data set were annotated by students and faculty. The C1–3 classes and the per-class distributions are displayed in Fig. 7.

Evaluation Metrics

The major metrics we focus on are the following:

AUC. The Area under the ROC curve (AUROC or AUC) is a good measure to represent the degree of separability of different classes. We use AUC to compare our DNN model against existing models, primarily for validation purposes.

Accuracy, Recall and F1-score. Throughout our evaluations, we explore accuracy for our various classifiers. In addition to accuracy, we measure recall and F1-score as well. For the binary classifier of C1, we pay special attention to recall, since we aim to minimize the false negatives, or type-2 error, to make sure that a vast majority of disaster-relevant SMPs get predicted as disaster relevant. For the multi-class classifiers C2 and C3, we focus on F1-score to assess how well we are able to minimize false positives and false negatives.

Number of SMPs labeled. To show the effectiveness and efficiency of active learning, we measure the number of SMPs that are picked for manual labeling by the dispatchers. A good active learning technique must enable the classifier to classify the SMPs with reasonable accuracy, while reducing the manual labeling efforts imposed on the dispatchers.

Number of S2S messages. While there are various message types in federated learning, we specifically measure the number of SME-to-SME (S2S) messages needed for handing SMPs to the correct dispatchers/organizations. Considering Fig. 4, let us assume the SMP belonged to 'LA Police'. In that case, overall 3 S2S messages will be required ('LA Fire' to 'LA Health', then to 'LA Police', and eventually back to 'LA Fire'). We aim to reduce this messaging, with reasonable accuracy.

5.2 Analyzing FLARE's Learning Elements

We investigate the performance of the major elements of FLARE using an 80-20 split of the dataset (80% training (with 1/8 of that as our validation set), and 20% testing). We first use the CrisisNLP/CrisisLex data sets to demonstrate the efficacy of our overall approach, and then examine FLARE's performance with the Woolsey Fire data.

5.2.1 Classifier Model: DNN-based with USE.

As described in §4, we have a DNN model at the core for our classifiers. Table 2 provides the hyperparameters determined to be reasonable experimentally. We use a batch size of 32 and 20-25 epochs in our experiments. To avoid overfitting, we use early stopping based on the accuracy of the validation set with 'patience' of 3 to 7, and a minimum delta of 0.0001. We also use the Adam optimizer [17]. We get approximately the same performance with other optimizers as well. For the loss model, we use binary cross-entropy for C1 classifier, and categorical cross-entropy for the C2 classifier. For the C3 classifier, we can use either of them, based on the number of classes. We use the categorical cross-entropy if there are more than two classes because it assumes that all the classes are mutually exclusive, and an SMP must belong to only one of the classes at a time. Thus, it provides prediction probabilities for all the classes that sum up to 1. To avoid over-fitting, we use 'early stopping' based on the accuracy of validation set with patience of 3 and a minimum delta of 0.0001.

We compare our USE-based classifier model with the existing state-of-the-art models discussed in §2. Using the CrisisNLP/CrisisLex data set, previously explored in [14, 21], we evaluate our model by comparing it against models investigated in [16, 19]. For a fair and thorough comparison, we use the same train/test data split, and use the same metrics for evaluating the performance. Table 3 shows the AUC score for the binary classifier, *i.e.*, C1 in FLARE, which predicts if a tweet is relevant or not. As the table shows, using a dense layer model with USE (which we use in FLARE) is able to outperform the non-neural network models, *i.e.*, linear regression (LR) and SVM, and performs on par with the CNN-based model of [19].

We next evaluate the performance of the multi-class classifier, C2, in FLARE, using the metrics of accuracy and F1-score, on the same labeled data. As the Table 3 shows, just as in the case of C1, our DNN-based model with USE outperforms SVM, and is on par with CNN_I for C2 as well. These results show that our DNN-based model performs reasonably well.

5.2.2 Active Learning.

We now add the active learning (AL) element to our core DNN-based model, and investigate its impact. We use the Woolsey Fire

Table 3: Performance metrics for C1 (Binary) and C2 (Multi-class) classifiers

Disaster Name	C1 - AUC					C2 - Accuracy			C2 - Macro F1		
	LR [19]	SVM [19]	CNN_I [19]	CNN_{AAf} [16]	DNN+USE (FLARE)	SVM [19]	CNN_I [19]	DNN+USE (FLARE)	SVM [19]	CNN_I [19]	DNN+USE (FLARE)
California Earthquake	0.755	0.747	0.783	0.836	0.772	0.7563	0.7752	0.7846	0.7	0.71	0.66
Cyclone PAM	0.906	0.9074	0.926	0.926	0.959	0.6788	0.6901	0.7459	0.63	0.65	0.71
Nepal Earthquake	0.826	0.836	0.848	0.875	0.844	0.6961	0.708	0.7381	0.55	0.55	0.59
Typhoon Hagupit	0.759	0.7764	0.858	0.883	0.894	0.711	0.8151	0.744	0.68	0.79	0.68

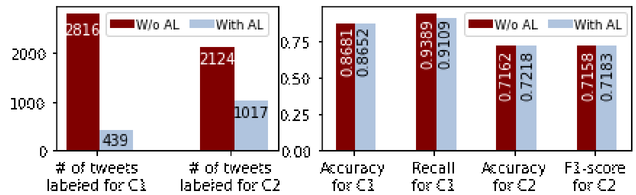


Figure 8: Effectiveness of active learning for C1 & C2

data throughout this sub-section, with an 80-20 training-test division: in the case of AL, sample tweets get 'picked' from the 80% training set for manual labeling and training, rather than the whole training set. In both scenarios (with and without AL), the same 20% testing set is used to measure the classifiers' performance. We train the model for a constant 20 epochs and do not use early stopping, as the training is incremental and we do not have sufficient labeled data to prepare a validation set. All the other hyperparameter configuration is the same as described in Table 2. These hyperparameters can be re-configured while instantiating the classifier. Fig. 8 shows the comparison of the active learning-based approach of FLARE *vs.* the same DNN-based model, but without active learning (a more traditional learning approach), for C1 (among all tweets) and C2 (among disaster-relevant tweets). FLARE's active learning approach reduces the manual labeling effort (~85% reduction for C1 and ~50% reduction for C2), while achieving roughly the same accuracy and recall/F1-score.

5.2.3 Federated Active Learning.

We then make our approach of active learning with DNN-based models federated, resulting in a federated active learning (FAL) framework. We have 4 distinct SMEs participating in the federated learning procedure, each SME being responsible for a distinct C2 class/name of Fig. 7, and its subsidiary C3 classes/names. We investigate our federated approach and its design choices on the same 80-20 division of the tweet dataset for training-testing, with the data distributed among the 4 SMEs.

An important aspect of FLARE is keyword-based data distribution, which is achieved with decentralized crawlers at each SME with customized keywords. An alternate approach, would be a *uniform distribution*, where each SME owns disjoint tweet pools of the equal size, distributed randomly. While there may be multiple ways in which the latter uniform distribution may be realized (the help of a central distributor, a purely budget-based crawling criteria in a decentralized way, *etc.*), we study this alternative assuming theoretically that this distribution is achieved. As Table 4 shows, for C1, our keyword-based distribution does not diminish the performance of federated learning compared to uniform distribution; in fact, it improves it slightly. The advantage of keyword-based distribution becomes more evident for the C2 classifier, as shown in Table 5: it leads to similar (even slightly better) accuracy and F1-score, but importantly it reduces the manual labeling effort. This shows the benefit of our decentralized keyword-based distribution of crawling of tweets in FLARE.

Another important aspect of the federated learning in FLARE is SME-to-SME (S2S) messaging, aimed at delivering each SMP to the SME specialized and best suited to process and label it. This type of messaging is only applicable for the C2 classification (and not C1). As Table 5 shows, whether with uniform or keyword-based distribution, having the S2S messaging capability improves the accuracy and F1-score quite significantly. Note that when S2S messaging is disabled (*i.e.*, a traditional FAL approach), all manual labeling (of low-confidence tweets) is purely *local* (*i.e.*, locally labeled tweets): each SME's dispatcher can manually label only those tweets crawled by its own SME *and* belonging to its corresponding C2 class/name (hence, it has no way of passing it to the 'right' SME). As the table also shows, with keyword-based approach, we cause fewer S2S messages to be passed on (*i.e.*, reduce communication overhead) compared to a uniform distribution approach. We take advantage of name-based pub/sub [9] for this S2S messaging, which delivers content with minimum communication overhead, compared to request/response, polling, or broadcast. Note that C3 does not use federated learning, thus we do not report on it here. As Tables 4 and 5 show, our keyword-based distribution and S2S messaging capability achieves the best performance (in terms of accuracy/F1-score) with the lowest overhead (in terms of manual labeling effort and S2S messaging, when applicable), which are critical in disaster management.

5.3 Fully Streaming Data with Pipelining

We now evaluate FLARE in a practical application scenario, which has two significant features. First, we consider fully streaming data; *i.e.*, our system (at all three classification stages) starts with no initial training data (which can happen when a disaster hits), and has to process them one by one in an online manner. For this purpose, we use all 100% of the Woolsey Fire data as test data (*i.e.*, tweets we want to map and deliver to first responders). This is in contrast to our experiments in §5.2, where we had a 80-20 division of data for training and testing. Second, input tweet data is processed by FLARE's three classifiers (C1–C3) in a pipeline (the stages described in §3.2), taking the inter-dependency between the various classifiers into account. This is in contrast to experiments in §5.2, where we analyzed independent classifiers.

Table 6 shows the summary of results in this setting, for C1–C3. For accuracy and recall/F1-score, we report on both *initial* and *dispatcher-assisted* classification. For example, for accuracy, the initial accuracy refers to accuracy before any active learning procedure, while dispatcher-assisted accuracy accounts for the manual labeling performed by dispatchers on selected tweets as well. Low confidence tweets with potentially incorrect initial classification are corrected by the dispatchers' intervention, and then fed to the next stage in the pipeline. The *overall accuracy* for each classifier

Table 4: Alternatives for FAL in C1: uniform *vs.* keyword-based

	Uniform	Keyword-based (FLARE)
Accuracy	0.8738	0.8766
Recall	0.9425	0.9499
# of tweets labeled	872	831

Table 5: C2 FAL: uniform *vs.* keyword-based, with and w/o S2S messaging

	Uniform w/o S2S messaging	Uniform with S2S messaging	Keyword-based w/o S2S messaging	Keyword-based with S2S messaging (FLARE)
Accuracy	0.6786	0.7237	0.1767	0.7368
F1-score	0.5804	0.6999	0.0669	0.7130
# of low confidence tweets	2124	1434	1312	1172
# of tweets labeled	542	1434	376	1172
# of tweets labeled locally	542	357	376	393
# of S2S messages	Disabled	2661	Disabled	1866

Table 6: Fully streaming and pipelined data on C1–3

	C1	C2	C3 (avg)
Accuracy (initial)	0.8262	0.6847	0.8553
Accuracy (dispatcher-assisted)	0.9091	0.8963	0.9291
Recall/F1 (initial)	0.9462	0.6183	0.8238
Recall/F1 (dispatcher-assisted)	0.9838	0.8589	0.9034
# of input tweets	3521 (of 3521)	2613 (of 2656)	2342 (of 2656)
# of correctly classified tweets	3201	2342	2176
# of tweets labeled	908	1223	441
Overall accuracy	0.9091	0.8818	0.8193

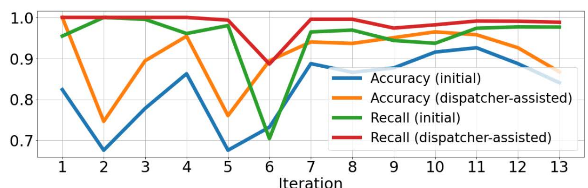


Figure 9: C1 performance across FL iterations

refers to the dispatcher-assisted accuracy when compared to the ground truth in the complete data set, rather than being based on the ground truth in the subset of the data that the respective classifier received as input. As Table 6 shows, overall accuracy degrades going through C1-C3, as each classifier introduces inaccuracies carried over to the next classifier in the pipeline. Note that C1 and C2 each have a single shared model (in the FAL setting), while C3 has four separate individual models across all four SMEs running locally. The C3 results in Table 6 are presented as weighted averages across those four SMEs. Also note that for C1, we focus on recall (for the 'relevant' class), while for C2 and C3 we focus on F1-score.

FLARE's C1 achieves very high accuracy and recall for the task of predicting if the tweet is relevant to disaster or not. This is achieved with only 25% of the tweets (908 out of 3521) being manually labeled, without the need to have any initial trained data. The high dispatcher-assisted recall value (0.9838) is significant since it means that of 2656 total relevant tweets in the data set, 2613 get fed to C2 for further processing. Only 43 relevant tweets (1.62%) are lost going from C1 to C2. But even this number is undesirable if they report critical life-threatening situations needing immediate attention. Having said that, upon examination of these 43 tweets, we observed that they are 'borderline' relevant tweets: they appeared to express emotions and opinions, including political ones. None of them explicitly indicated an actionable task for first responders. Thus, our C1 does not toss out truly relevant SMPs which include an actionable item. Fig. 9 provides the accuracy and recall for relevant class values before and after dispatcher assistance throughout the course of classification. Each iteration represents an instance when the aggregated model was created with FL by the Incident Coordinator and sent back to all SMEs. As new information with new terms arrive, we see a drop in accuracy, but we maintain recall, ensuring no relevant information is lost.

FLARE achieves good initial accuracy and F1-score for C2, and much higher accuracy when dispatcher-assisted, with reasonable amount of manual labeling effort. Considering the total ground truth of 2656 disaster-relevant tweets in the data set (which also includes those 43 tweets C1 mistakenly discarded as irrelevant), C2 achieves 88.18% overall accuracy; 2342 and 271 tweets get passed to the correct and incorrect organizational SMEs respectively, for further processing for C3.

FLARE also achieves high accuracy and F1-score in C3, as shown in Table 6, averaged across the four SME's C3's. This is achieved with very little manual labeling requiring only 18% of C3's input to be labeled (441 out of 2176). Out of all 2656 disaster-relevant tweets, 2176 of them (81.93%) get mapped to the correct prefix (at NSMP generator), which demonstrates a very good performance.

Summarizing the overall accuracy and interpreting them from a first responder point of view, the three classifiers and the three corresponding levels of the namespace achieve:

1) 98.38% of all disaster-relevant tweets get published in the network and will be delivered to "some" first responder(s), whether or not to the right organization/role. 1.62% of tweets deemed disaster-irrelevant by C1 are not re-examined. But as noted, these are borderline, non-actionable tweets.

2) 88.18% of all disaster-relevant tweets are published to first responder(s) in the right organization, whether or not it is the right fine-grained role. 10.2% of disaster-relevant tweets are delivered to the incorrect organization, *e.g.*, tweets with topic under 'Fire' getting delivered to first responders with roles under 'Medic'. We can recover from this inaccuracy in FLARE, as first responders can provide feedback on them to dispatchers post-delivery, and potentially forwarding inaccurately classified tweets to the right department, since they are equipped with the incident namespace.

3) 81.93% of all disaster-relevant tweets get published to the correct organization's first responders having the right role, at the finest granularity possible. 6.45% of disaster-relevant tweets get delivered to first responder(s) in the right organization, but not with the correct fine-grained role, *e.g.*, a tweet with topic '/Medic/AnimalRescue' getting delivered to first responders in the role of '/Medic/EMTandMedicalNeed'. This type of inaccuracy (*i.e.*, the 6.45%) is a more tolerable inaccuracy, as first responders within an organization can easily pass a tweet to a more suitable team or individual in the same organization. The post-delivery feedback from first responders can also help recover from this inaccuracy.

5.4 Comparison to Keyword Matching

Classification based on NLP triumphs over naive methods like keyword matching due to ability to capture semantic information from the text. For disaster response, capturing semantics is even more critical. First, it enables in retrieving tweets which are actually relevant to the disaster and need emergency response. Second, it helps in mapping the disaster relevant tweets to the correct response team. To show that our NLP-based classification is superior, we

Table 7: Keyword matching results

		C2	C3
Including tweets belonging to "Other" Class	**Total Tweets**	2656	2656
	Tweets mapped	3441	4189
	Tweets mapped correctly	963	483
	Accuracy	0.3626	0.1819
Excluding tweets belonging to "Other" Class	**Total Tweets**	1169	1169
	Tweets mapped	1541	1887
	Tweets mapped correctly	654	483
	Accuracy	0.5595	0.4132

also evaluated the performance of keyword matching for mapping tweets to their C2 and C3 classes. The keywords used for matching are the same as those for crawling and partitioning in FLARE.

To achieve an apples-to-apples comparison with FLARE, we ignored 'irrelevant' tweets, which comprises 25% of the whole dataset. Further since the "*Other*" tweets, although relevant, do not provide an actionable request to first responders. Our comparison examines both with and without the *Other* class. Finally, since a single tweet may match multiple keywords, even if one mapping is correct, we say that the tweet can be mapped to the right department/role. We measure the accuracy of keyword matching, based on the tweets mapped correctly divided by the number of unique tweets.

Table 7 shows the accuracy with keyword matching for C2 and C3 classes. Even after excluding the *Other* class and enabling the mapping to multiple classes, we achieve only a 56% and 41% accuracy for C2 and C3 respectively. This is significantly lower than what we achieve using the NLP-based classification, as shown in Table 6.

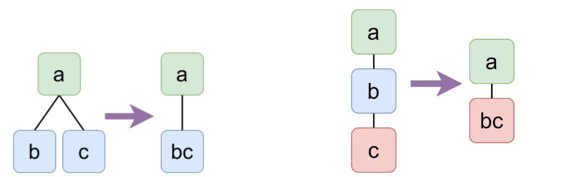
6 SUPPORTING DYNAMIC NAMESPACES

6.1 Namespace Updates

During the course of a disaster, the incident namespace may have to be updated as the situation evolves. These updates may be because of addition or deletion of roles, as well as changes in the command chain, according to the real-time needs of the emergency response tasks. These updates can be applied at various levels of the namespace. Existing name-based pub/sub methods, such as [8] provide mechanisms for making changes to the namespace, and updating subscription tables at ICN routers and rendezvous points, to adapt the downstream paths to the new namespace. In FLARE, this is more challenging, as the namespace is integrated with the classifiers: names are also classifier classes. The result of a change may be that the set of classes of a classifier would also have to be changed, *i.e.*, re-trained. This is important to explore, as making changes to working classifiers can be burdensome, especially if it means a new classifier has to be trained fully from scratch. We describe how FLARE gracefully handles such updates.

Different types of namespace updates may have different impacts on the updating of the classifiers. Here, we focus on sub-namespace at the level of fine-grained roles, *i.e.*, C3 (Fig. 2), where the leaf name nodes are classes in the classifier. Thus, it is important to distinguish between two types of changes in the namespace.

6.1.1 Namespace updates that do not *require classifier update:* Generally, it may be desirable to not to have to change the classifier, as much as possible. This helps avoid the overhead of training the classifier. To this end, we identify cases of namespace changes that can be solely handled through the name-based pub/sub delivery mechanism. As notable cases, changes such as name deletion and name merging do not require changes to the classifier. In *name merging*, two or more names in the incident namespace are merged. As illustrated in the (sub-)namespace updates in Fig. 10, this merging can happen between the names situated at the same level or at different levels. In Fig. 10(a), let us assume names 'b' and 'c' (and not *'a'*) are existing classes in C3. At some point, the department in charge of 'a' decides to merge 'b' and 'c', instead creating a new name 'bc'. The goal is for all SMPs formerly classified as either 'b' (published to "/a/b") or 'c' (published to "/a/c"), to now be classified as "bc" (published to "/a/bc"). It is easy to see that this namespace change does not need a change in classifier: the classifier can still continue working un-interrupted, using 'b' and 'c' as distinct classes. The NSMP Generator at the SME handles the task of merging post-classifications, generating the NSMP with name "/a/bc". Another example for name merging is shown in Fig. 10(b), where name 'c' is initially the only name corresponding to a class ('b' is an intermediate name). While this merging makes changes in the hierarchical structure, it still does not require changes in the classifier, using logic similar to the case in Fig. 10(a).

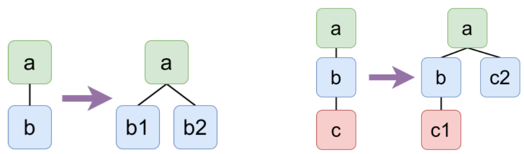


(a) Name merging at same level (b) Name merging across different levels

Figure 10: Namespace updates that do not require classifier update

6.1.2 Namespace updates that do *require classifier update:* In contrast to the changes described above, changes such as addition of a new leaf name node, or splitting an existing leaf name into two new names, require updating the classifier. In the case of name splitting, a single name in the incident namespace is split into two or more names as shown in the examples in Fig. 11. In Fig. 11(a), we assume name 'b' (and not *'a'*) is a class in C3 . At some point, the department in charge of 'a' decides to split 'b' and instead creates two new names 'b1' and 'b2'. The goal is for some SMPs formerly classified as 'b' (published to "/a/b"), to now be classified as either 'b1' (published to "/a/b1") or 'b2' (published to "/a/b2"). This namespace change will require the classifier to change and be re-trained. This is because the classifier initially does not recognize the difference between 'b1' and 'b2', but it now needs to. FLARE allows for such a change in the classifier, which can happen online and incrementally, thanks to FLARE's active learning with the help of dispatcher labellers. The NSMP generator also will be updated (as in the previous case) to now include 'b1' and 'b2'. Another example for name splitting is shown in Fig. 11(b), where name 'c' is split to leaf names 'c1' and 'c2', where the two get placed at two different levels of the hierarchy. Same as in Fig. 11(a), the classifier has to adapt itself to classify some of 'c'-class SMPs into 'c1' and some into 'c2'.

It is important to note that the above namespace updates are confined to C3 classifiers. C3 is fully local and the overhead of updating it is reasonable. This justifies our design choice on the guideline for static and dynamic levels (Fig. 2). Updates to C2 is more complicated as C2 is federated across multiple SMEs belonging to different departments. C2 name changes are organization-level changes, and needs to be coordinated across the departments. Dynamics at the C2-level is part of work we will explore in the near future.



(a) Name splitting at same level (b) Name splitting across different levels

Figure 11: Namespace updates that require classifier update

6.2 Evaluation of Namespace Updates

We evaluate the instantiation of a new classifier with increased number of classes due to a change in the namespace that requires a change in the classifier. We compare it against the case of a static classifier which has a fixed number of classes throughout. From the Woolsey fire dataset, we pick 1K tweets belonging to the C3 classes under 'Infrastructure' in Fig. 7. Let us assume that in the beginning, during the first 400 tweets, we have a name 'Structure and Shelter', that combines tweets belonging to the 'Structure, Building & Road Damages' and 'Shelter, Shortage & Outage' classes. For the last 500 tweets, we split the class 'Structure and Shelter' into their actual categories,*i.e.*, 'Structure, Building & Road Damages' and 'Shelter, Shortage & Outage'. The tweets are passed to the classifier in batches with size of 10 tweets per batch. Fig. 12 and Fig. 13 show the accuracy and manual labeling effort for this dynamic classifier case, alongside the fixed classifier case, *i.e.*, one that keeps going without any splitting. As Fig. 12 shows, after 400 tweets, the accuracy of the static classifier (in red) drops and remains low throughout, as the classifier is unable to distinguish between 'Structure, Building & Road Damages' and 'Shelter, Shortage & Outage'. The dynamic classifier, on the other hand is able to rectify the drop in accuracy. There is a reasonable increase in the load on the dispatcher, but that is alleviated over time as the new classifier gets trained. This ability of FLARE to instantiate a new classifier based on namespace changes makes it adaptive to namespace changes at C3 level. In the future, we will utilize the knowledge of the previous classifier having fewer classes by using a continual learning system [25] to further reduce the manual labeling effort.

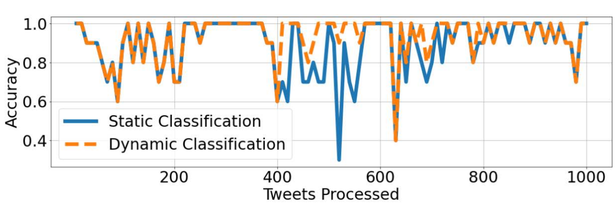


Figure 12: C3 accuracy while performing class split (split at 400 tweets)

7 CONCLUSION

We presented FLARE, a framework for disaster response that synergistically combines Federated Active Learning with name-based information dissemination using ICN. FLARE assigns the right name to content, such as a social media post (SMP), so that all relevant content is delivered to the right recipients in a timely manner. We focused on the important application of SMPs being used for

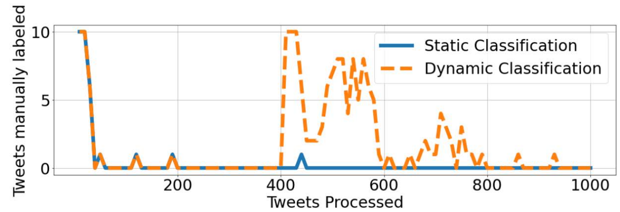


Figure 13: C3 tweets labelled while performing class split (split at 400 tweets)

updates and reports during disasters which have to be delivered to the right first responders.

FLARE performs the mapping of SMPs to the right names in an online, real-time manner using NLP/ML-based Social Media Engines (SMEs). FLARE uses active learning to reduce what would otherwise be a very challenging human manual labeling effort. FLARE further uses federated learning so that various organizations and their SMEs and dispatchers can cooperate to disseminate critical information towards the appropriate first responders. By coupling the classifiers in FLARE with the incident management namespace, FLARE allows SMPs to be delivered to the correct SMEs and thereby to the right first responders, leveraging the expertise of organization-specific dispatchers for labeling, based on an initial classification. We showed that by using keyword-based decentralized crawling and messaging among SMEs, FLARE both reduces the labeling overhead (because of keyword-based crawling) and improves the classification accuracy (because of message passing). Processing streaming real-world Twitter trace data from the California Wildfires, we showed that FLARE maps and delivers 81.93% of all disaster-relevant tweets to the right first responders, at the finest level of granularity of the namespace. This can significantly help in disaster response and improve outcomes. FLARE's namespace-driven multi-classifier pipeline allows even a subset of inaccurately labeled tweets to reach first responders who are "somewhat relevant", *e.g.*, the right department, and leverage their knowledge and expertise to then get them to the right first responder. The framework allows dynamic namespace changes by modifying the fine-grained classifier with minimal labeling and loss in accuracy.

While we did not explore security concerns in this paper, FLARE's name-based pub/sub framework can be integrated with traditional content-based security for integrity and authentication [31], and NDN's encryption methods for confidentiality [37]. The use of a dynamic namespace may additionally raise security issues, such as the need to update a first responders' access privileges as their roles change due to namespace change, which can be addressed using content-based security methods such as NDN's attribute-based access control method [36] with proper attribute/key update mechanisms. Handling the full spectrum of other security issues will be explored in future work.

8 ACKNOWLEDGEMENTS

This work was supported by the US National Science Foundation grant CNS-1818971 and the US Dept. of Commerce, NIST PSIAP award 70NANB17H188. We thank Prof. Amr Magdy of UCR for providing us with the dataset of tweets. We thank the reviewers for their valuable feedback, and the excellent guidance of Dr. Erick Lavoie during the shepherding process of the paper.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*. 265–283.

[2] Lulwa Ahmed, Kashif Ahmad, Naina Said, Basheer Qolomany, Junaid Qadir, and Ala Al-Fuqaha. 2020. Active Learning Based Federated Learning for Waste and Natural Disaster Image Classification. *IEEE Access* 8 (2020), 208518–208531.

[3] M. Arumaithurai, A. Tagami, K. K. Ramakrishnan, and N. Blefari Melazzi. 2020. Research Directions for Using Information-Centric Networking (ICN) in Disaster Scenarios. RFC 8884.

[4] Nasser Assery, Yuan Xiaohong, Sultan Almalki, Roy Kaushik, and Qu Xiuli. 2019. Comparing learning-based methods for identifying disaster-related tweets. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 1829–1836.

[5] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, and Nicholas D Lane. 2020. Flower: A Friendly Federated Learning Research Framework. *arXiv preprint arXiv:2007.14390* (2020).

[6] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175* (2018).

[7] Kevin Chan, Bongjun Ko, Spyridon Mastorakis, Alexander Afanasyev, and Lixia Zhang. 2017. Fuzzy interest forwarding. In *Proceedings of the Asian Internet Engineering Conference*. 31–37.

[8] Jiachen Chen, Mayutan Arumaithurai, Xiaoming Fu, and KK Ramakrishnan. 2016. CNS: content-oriented notification service for managing disasters. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. 122–131.

[9] Jiachen Chen, Mayutan Arumaithurai, Lei Jiao, Xiaoming Fu, and K. K. Ramakrishnan. 2011. COPSS: An efficient content oriented publish/subscribe system. In *2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems*. IEEE, 99–110.

[10] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364* (2017).

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[12] Zayan EL Khaled and Hamid Mcheick. 2019. Case studies of communications systems during harsh environments: A review of approaches, weaknesses, and limitations to improve quality of service. *IJDSN* (2019).

[13] Mohamed Goudjil, Mouloud Koudil, Mouldi Bedda, and Noureddine Ghoggali. 2018. A novel active learning method using SVM for text classification. *International Journal of Automation and Computing* 15, 3 (2018).

[14] Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. 2016. Twitter as a lifeline: Human-annotated twitter corpora for NLP of crisis-related messages. *arXiv preprint arXiv:1605.05894* (2016).

[15] Mohammad Jahanian, Toru Hasegawa, Yoshinobu Kawabe, Yuki Koizumi, Amr Magdy, Masakatsu Nishigaki, Tetsushi Ohki, and K. K. Ramakrishnan. 2019. DiReCT: Disaster Response Coordination with Trusted Volunteers. In *2019 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*. IEEE, 1–8.

[16] Md Yasin Kabir and Sanjay Madria. 2019. A deep learning approach for tweet classification and rescue scheduling for effective disaster management. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 269–278.

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR, 1273–1282.

[19] Dat Tien Nguyen, Kamela Ali Al Mannai, Shafiq Joty, Hassan Sajjad, Muhammad Imran, and Prasenjit Mitra. 2016. Rapid classification of crisis-related data on social networks using convolutional neural networks. *arXiv preprint arXiv:1608.03902* (2016).

[20] Kathleen Nichols. 2019. Lessons learned building a secure network measurement framework using basic NDN. In *Proceedings of the 6th ACM Conference on Information-Centric Networking*. 112–122.

[21] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2014. Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 8.

[22] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[23] Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. 2018. Batch-based active learning: Application to social media data for crisis management. *Expert Systems with Applications* 93 (2018).

[24] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[25] Mark B Ring. 1998. CHILD: A first step towards continual learning. In *Learning to learn*. Springer, 261–292.

[26] Mark Saunders. 2018. Social media: California wildfires force thousands to evacuate. https://tinyurl.com/y2pfolyd.

[27] Jan Seedorf, Atsushi Tagami, Mayutan Arumaithurai, Yuki Koizumi, Nicola Blefari Melazzi, Dirk Kutscher, Kohei Sugiyama, Toru Hasegawa, Tohru Asami, K. K. Ramakrishnan, et al. 2015. The benefit of information centric networking for enabling communications in disaster scenarios. In *2015 IEEE Globecom Workshops (GC Wkshps)*.

[28] Jasmina Smailović, Miha Grčar, Nada Lavrač, and Martin Žnidaršič. 2014. Stream-based active learning for sentiment analysis in the financial domain. *Information sciences* 285 (2014).

[29] Wikipedia. [n.d.]. Woolsey Fire. https://en.wikipedia.org/wiki/Woolsey_Fire.

[30] L Yan. 2011. A survey on communication networks in emergency warning systems. *Sci. Comput* (2011).

[31] Yingdi Yu, Alexander Afanasyev, David Clark, Van Jacobson, Lixia Zhang, et al. 2015. Schematizing trust in named data networking. In *ICN*.

[32] Kiran Zahra, Muhammad Imran, and Frank O Ostermann. 2020. Automatic identification of eyewitness messages on twitter during disasters. *Information processing & management* 57, 1 (2020), 102107.

[33] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. 2014. Named data networking. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 66–73.

[34] Minsheng Zhang, Vince Lehman, and Lan Wang. 2017. Scalable name-based data synchronization for named data networking. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 1–9.

[35] Ye Zhang, Matthew Lease, and Byron Wallace. 2017. Active discriminative text representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[36] Zhiyi Zhang, Yingdi Yu, Sanjeev Kaushik Ramani, Alex Afanasyev, and Lixia Zhang. 2018. NAC: Automating access control via Named Data. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 626–633.

[37] Zhiyi Zhang, Yingdi Yu, Haitao Zhang, Eric Newberry, Spyridon Mastorakis, Yanbiao Li, Alexander Afanasyev, and Lixia Zhang. 2018. An overview of security support in named data networking. *IEEE Communications Magazine* 56, 11 (2018), 62–68.

[38] Zhenkai Zhu and Alexander Afanasyev. 2013. Let's chronosync: Decentralized dataset state synchronization in named data networking. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*. 1–10.