What Went Wrong: A Survey of Wildfire UAS Mishaps through Named Entity Recognition

Sequoia R. Andrade
HX5 LLC.
Under Prime Contract No. 80ARC020D0010
with the NASA Ames Research Center,
Moffett Field, United States
sequoia.r.andrade@nasa.gov

Hannah S. Walsh
Intelligent Systems Division
NASA Ames Research Center
Moffett Field, United States
hannah.s.walsh@nasa.gov

***Abstract*—Increasingly, unmanned aircraft systems (UAS) are being applied to wildfire incidents for tasks such as mapping, aerial ignition, and delivery. As a result, aviation incident reporting systems for wildfires are beginning to accumulate data related to UAS mishaps in wildfire response. In this research, we apply state-of-the-art natural language processing (NLP) techniques to develop a custom Named Entity Recognition (NER) model which extracts entities relevant to safety analysts. The custom NER model is built by fine-tuning an existing Bidirectional Encoder Representations from Transformers (BERT) model, resulting in a generalizable NER model that can extract engineering relevant entities including failure modes, causes, effects, control processes, and recommendations from failure-relevant text. This model performs passably, with a weighted average f1 score of 0.33 across entity types, indicating more labeled training data is needed. Extracted entities are used to form a Failure Modes and Effects Analysis (FMEA)-style survey of wildfire UAS mishaps reported using the SAFECOM system. Similar mishaps are manually clustered and reported as single rows within an FMEA. For each cluster, we compute frequency, severity, and overall risk in accordance with FAA standards. This methodology can be applied as part of a broader safety management system to track trends in mishaps (e.g., likelihood, severity) and discover knowledge (e.g., causes, effects) that can be utilized to improve safety outcomes and system performance.**

***Index Terms*—Machine Learning, Named-entity Recognition, FMEA, Natural Language Processing, UAS**

I. Introduction

Wildfire response is an inherently dangerous operation, with hazardous conditions affecting both ground and aerial operations alike. In the United States, 2021 reported the second highest amount of firefighter fatalities in ten years, with twenty-three personnel fatalities [1]. Of those fatalities, three were due to aerial operations with an additional three due to vehicle accidents [1]. Fire fighters also suffered from twenty-six burn incidents, twenty-three “hit by” (i.e., tree, rock, or vehicle) incidents, and eighteen entrapments. With the number and size of wildland fires increasing [2] in part due to climate change [3], first responders are seeking new methods to improve efficacy and safety of wildfire response operations. Unmanned aircraft systems (UAS) show promise for improving wildfire suppression through increased situational awareness, while simultaneously improving safety by replacing manned operations with autonomous vehicles. Despite the promise of UAS enabled operations, the implementation of this new technology to the complex wildfire response system may inadvertently introduce new hazardous scenarios and failure modes.

As UAS are increasingly used in wildfire response, UAS mishap reports are becoming available through the SAFECOM aviation reporting system. These reports are conventionally accessed by operators and manually parsed by analysts to identify notable mishaps. However, modern advancements in natural language processing (NLP) methods provide an opportunity to efficiently extract information from reports to augment traditional safety analysis. Bidirectional Encoder Representations from Transformers (BERT) models are pretrained on a large corpus of text, including Wikipedia pages, and achieve superior performance on a variety of NLP tasks, such as information extraction, language inference, text summarization, question answering, and query systems [4]. BERT models learn on both the left-to-right and right-to-left text using masking, resulting in context-dependent word embeddings [4]. While BERT models can perform some general NLP tasks out-of-the-box, they can also be fine-tuned for highly specialized tasks. Although BERT models have been successful on a variety of specialized tasks, these models have not previously been applied to the complex, domain-specific language used in wildfire UAS mishap reports.

Mishap and safety reports have been identified as a key information class for an In-time Aviation Safety Management System (IASMS) [5]. The proposed IASMS uses services, functions, and capabilities to conduct risk management and safety assurance at scale [6]. However, safety reports are conventionally under utilized in aerospace engineering because they require manual processing and analysis to identify relevant information. In turn, the present-day method of manual safety report analysis is not scalable when large quantities of text data is generated. Instead, the ability to automatically extract failure relevant information from large sets of safety reports is necessary for an IASMS. Results from this process can support the IASMS defined functions of hazard identification and risk analysis to inform safety assurance.

In this paper, we pre-train and fine-tune a state-of-the-art BERT model to create a custom named entity recognition (NER) model that extracts relevant safety information. Mishap reports are manually annotated for entities and used to train the model. From unstructured text, our custom NER model can identify failure modes, causes, effects, control processes, and recommendations. These components may then be used to construct a data-driven failure modes and effects analysis (FMEA). To illustrate this process, the model is applied to wildfire UAS mishap reports to extract a FMEA-style survey of documented failures. The resulting survey identifies and analyzes the failure modes of battery degradation, dislodged casing, hang fire, loss of ground control station (GCS), loss of GPS, loss of line of sight (LOS), loss of control, parachute landing malfunction, propeller arm disconnection, and airspace intrusions.

II. Background

To provide context, in this section we discuss the current state of UAS in wildfire response, including use cases, aircraft models, and barriers. Next, the failure modes and effects analysis (FMEA) method is defined with an emphasis on data-driven uses. Finally, we introduce named-entity recognition (NER) and describe how a custom NER model can be leveraged to produce data-driven FMEAs.

A. *UAS in Wildfire Response*

Unmanned aircraft systems (UAS) are used in the public sector at a growing rate with programs in wildfire response, law enforcement, and emergency medical services. Table I provides examples of UAS models currently in use for wildfire response, along with the mission use-cases. UAS are most commonly used for reconnaissance and infrared imagery missions at this time; however, they may also be used for aerial ignition in controlled burn scenarios. Organizations such as DRONERESPONDERS help first response agencies initiate drone programs through research, training, certification, and sharing information. While there is increased interest in UAS for public use-cases, most drone programs are fairly new, with 64% of participants reporting UAS programs two years old or less from a spring 2020 DRONERESPONDERS survey [7]. Simultaneously, NASA's Scalable Traffic Management for Emergency Response Operations (STEReO) project aims to introduce advanced technology, including autonomy and unmanned traffic management, into UAS-enabled responses [8]. Hence UAS in wildfire operations is still an emerging concept and may benefit from further safety analysis to better inform operators of potential risks.

B. *Failure Modes and Effects Analysis (FMEA)*

Failure modes and effects analysis (FMEA) is a semi-quantitative risk assessment method used for safety assurance during system verification [10]. Conventionally, an FMEA is performed after an expert analyst constructs a block diagram consisting of a system's high-level functions to low level components [11]. Then, the failure modes of each component are considered along with the operational phase the failure occurs during, failure causes, system level effects, methods for failure detection, and with extensions including criticality of the effects (FMECA) [11], [12].

TABLE I
Examples of UAS aircraft currently used in wildfire response from SAFECOM reports [9]

Model	Use Case
Matrice 600	Infrared Imagery, Reconnaissance, Aerial Ignition
Anafi	Reconnaissance
Solo	Reconnaissance, Infrared Imagery
Silent Falcon	Reconnaissance, Infrared Imagery

Because conventional FMEAs require an expert to manually identify relevant failure information for the system, the process may be time consuming and is limited to the expert's knowledge and experience. As a result, there is a growing body of research leveraging existing documents to augment FMEA construction and build knowledge bases. For example, in 2020, Rehman et al automatically generated FMEAs through an ontology developed from existing FMEA worksheets [13]. This work uses manually coded relationship logic to identify the FMEA components (causes, modes, etc.) from worksheets [13]. Similarly, Spreafico and Russo created an assistive tool that semi-automatically produces FMEAs from patent documents using a custom-build semantic model [14]. Here sentence structure is coding using logic and key identifier words, which in turn identifies FMEA components from free text documents [14]. While these syntactic methods are effective, they only detect information specifically defined by the logical rules and identified indicator terms. In contrast, the research provided in this paper uses state-of-the-art transformer models to identify FMEA components through supervised training.

Named-entity recognition has shown promise as a solution to extract failure-related information from free text. Wang, Xhang, and Gao produce a knowledge graph of industrial safety information from existing hazard operability analysis reports (HAZOP) by augmenting existing BERT models [15]. The authors manually annotate data and use NER to extract information from the reports, resulting in a knowledge graph of all existing information in the sets of reports [15]. Kamath successfully used deep learning methods for named-entity recognition and relation detection to construct a knowledge graph of information from FMEAs [16]. However, Kamath's entities are manually labeled, rather than extracted from NER, and the relation detection model is custom built, rather than pre-trained on a large corpus of text like BERT models [16]. Previous work by the authors used topic modeling to extract a failure taxonomy from the NASA Lessons Learned Information System (LLIS) [17]; however, this work relied on the assumption that documents have separate sections that accurately report failure modes, causes, and recommendations [18].

Instead, this paper provides a method to extract failure-related terms from unstructured text and thus generalizes to non-sectioned documents. Further, the custom NER model developed in this work learns additional failure entities with the goal of constructing a data-driven FMEA with more details (control processes, failure effects, failure likelihood, severity, risk) than the initial taxonomy.

C. Named-Entity Recognition

Named-entity recognition (NER) is an information extraction method used to label words (or tokens) and phrases as specific entities, such as "person", "location", or "date". Established in the 1990s, initial NER methods relied on manual rules and lists of entities [19]. During the early 2000s, NER models shifted towards using machine learning by framing entity recognition as a binary classification problem [19]. More recently in 2016, deep learning architectures have been used for NER, with these methods outscoring all other existing methods [20]. In 2018, the Bidirectional Encoder Representations from Transformers (BERT) deep learning language model [4] became the new state-of-the-art for a range of natural language processing tasks, including NER. BERT models are pre-trained on large corpora of text, which allows the model to learn complex semantic structures relevant to tasks such as question answering, text summarizing, and information extraction [4]. These base BERT models can further be fine-tuned for specialized domains. Liu et al fine-tuned a base BERT model for named-entity recognition across a wide range of entities [21]. Results indicated the fine-tuned model outperforms the base model and successfully learns entities from different domains, despite some entities having minimal labeled data [21].

BERT models can be fine-tuned for advanced entity recognition, relationship detection, and domain specific use cases. Given named-entities, relationships between the entities can be extracted through a process known as relationship extraction (RE) or relationship detection (RD) [22]. While not explored in this research, RD is relevant to extracting failure-relevant information. In the case of FMEAs, a named cause entity results in a named failure mode entity, which may cause an effect entity. The primary relationships in this use case are casual, which may be inferred once entities are appropriately named and annotated. A specialized form of relation detection is causality mining (CM), which has been performed using a variety of methods from syntactic analysis to deep learning (including BERT models) [23]. NER and RD have been successfully applied to the highly specialized biomedical domain to detect complex relationships in research for fields such as genetics [22]. In 2020, Lee et al. pre-trained a base BERT model on biomedical texts to produce a biomedical domain-specific BERT model (BioBERT), which outperforms the base model on a variety of tasks [24]. Fine-tuned BERT models have also successfully been used for biomedical NER to detect entities of dosage forms, disease, drug, route of administration, and symptoms [25]. Advanced NER methods have been applied to detecting chemical entities from patents as well with notable improvements in performance from traditional methods [26]. Despite NER algorithms showing high success in custom use cases, state-of-the art natural language processing and named entity recognition methods have not yet been applied in a highly specialized aerospace engineering context.

III. Method

To produce data driven FMEAs, we develop a two-part method and apply it to a data set of wildfire UAS mishap reports (SAFECOM). First, we pre-train a base BERT model on NASA Lessons Learned Information System (LLIS) and SAFECOM documents to tailor the language model for engineering text. Next, we fine-tune the pre-trained BERT model for custom named-entity recognition using annotated reports from the LLIS. We then apply the trained NER model to SAFECOM reports to extract FMEA entities. Using meta data from the SAFECOM reports alongside the identified entities, similar reports are grouped together to form an FMEA. The following sections describe in detail the SAFECOM data set, pre-training the BERT model, the custom NER model, and the process of forming the data-driven FMEA.

A. SAFECOM Data Set

This research analyzes reports from the Aviation Safety Communiqué Database, or SAFECOM, which is a non-punitive safety reporting system intended for documenting aviation mishaps, hazards, and incidents during specialized operations, including wildfire response. Operators or witnesses may file a report about any safety-related issue that may cause a mishap during operations such as wildfire response, search and rescue, aerial mapping, and research missions. The reports consist of a narrative text, corrective action text, and meta data about the operation, including mission type, damages, and injuries. Of the 21,503 reports filed from 1994 to 2021, only 180 reports are related to UAS in wildfire response.

Figure 1 shows the frequency of both overall and wildfire-specific UAS reports from 2014 to 2021. During this time period, overall UAS reports increase steadily with wildfire-related reports accounting for a growing proportion of total UAS reports. UAS mishap reports originate during various fire suppression operations for different aircraft, including water drops, air attacks, reconnaissance, retardant drops, aerial ignitions, infrared imagery, and passenger transport. SAFECOM reports indicate mishap categories, and the distribution of those reported categories are shown in Figure 2. The most common reported category is "intrusion" followed by general "UAS" and "fleet operations". While reported less often, the "loss of link" and "loss of GPS" reports are interesting as they dictate specific failures in the UAS aircraft, rather than the aerial operation as a whole. Despite UAS being used in wildfire operations across the United States, the Pacific Southwest and Intermountain regions report the most mishaps in Figure 3, which may be due to more UAS use in those regions in general.

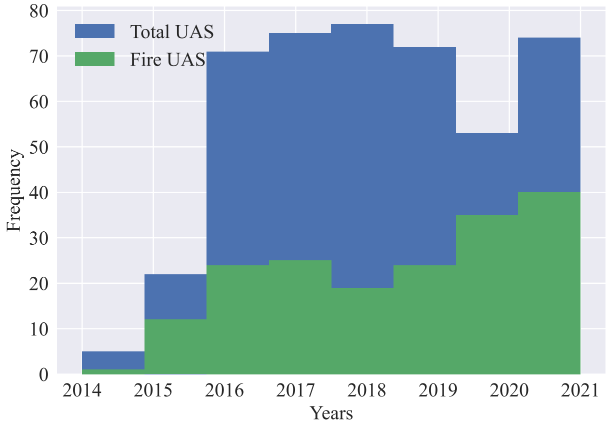

Fig. 1. Histograms of both non-fire and wildfire UAS mishap reports from SAFECOM.

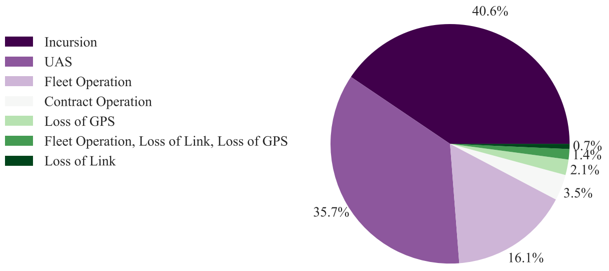

Fig. 2. Distribution of reported mishap categories from SAFECOM reports on UAS in wildfire response.

B. BERT Pre-training

Although BERT models are pre-trained on a large corpus of text, they can be further pre-trained on domain specific text to improve the masked language model. We additionally pre-trained the *'bert-base-uncased'* model on the full LLIS and SAFECOM data sets. The training set consisted of 2,102 LLIS documents from 1985 to 2021 and 21,503 SAFECOM reports from 1995 to 2021. This additional pre-training allows the BERT model to learn the context of specialized words and the unique language style present in engineering documents. Pre-training is technically a supervised process, where the labels used in training are the token (i.e., word or word piece) ids, which are also the model input. The model was pre-trained on the LLIS and SAFECOM data for seven epochs using an NVIDIA GPU, with training time lasting just over forty-eight hours.

C. Custom Named-Entity Recognition

We develop a custom named-entity recognition (NER) model which detects entities for failure analysis through fine-tuning a BERT model. A custom model is needed for this use case because failure-relevant entities are not included in pre-trained models, and thus cannot easily be extracted without custom model development. This custom model is developed according to Figure 4, primarily using the Huggingface Transformers and SpaCy APIs. First, we define the entities we want the model to recognize and manually tag reports to build training, testing, and validation sets. Here the LLIS is used for training and validation, while the SAFECOM UAS reports are used for testing. In accordance with NASA guidelines and handbooks [10]–[12], [27], the custom entities in this research are defined as follows:

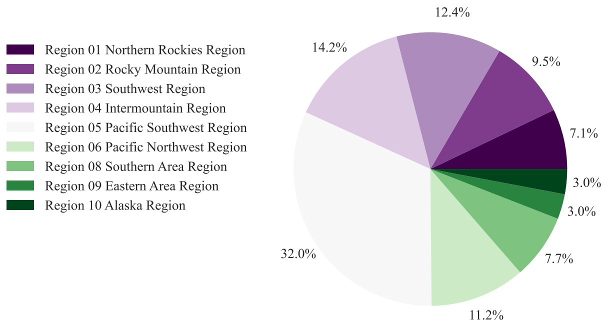

Fig. 3. Distribution of UAS wildfire SAFECOM reports across USFS regions.

1) *Failure Mode (MOD)*: The particular manner in which a component or system fails to perform its intended function.

2) *Failure Cause (CAU)*: Why the failure mode occurs; a condition or defect (a physical defect, a defect in a process or design, an environmental condition, or human error) that initiates a process leading to a failure mode.

3) *Failure Effect (EFF)*: The impact/consequence of the failure mode; an impact can be component level, subsystem level, system level, or mission level.

4) *Control Processes (CON)*: Existing systems or processes that are intended to prevent the occurrence of the failure mode or control the severity of the effect (i.e., a mitigation).

5) *Recommendations (REC)*: Future actions required to prevent the occurrence of the failure mode or its effects; i.e., how should the existing control processes be augmented.

When annotating, we kept tags as short as possible and tagged as many terms as possible for consistency. For model training and validation, we manually tag entities in 160 randomly selected NASA Lesson's Learned Information System (LLIS) reports using the Doccano [28] annotation package. Because the model should generalize to other engineering documents, the LLIS dataset is chosen for model training and validation as it provides detailed information on driving events, lessons learned, and recommendations. The LLIS contains

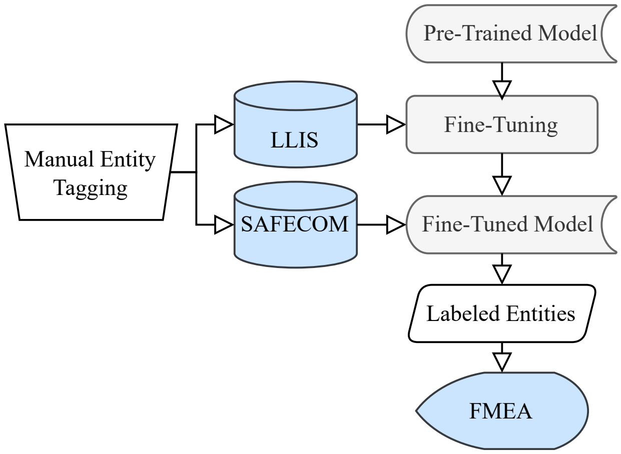

Fig. 4. A high-level flowchart showing the process for developing the custom Named-Entity Recognition (NER) model.

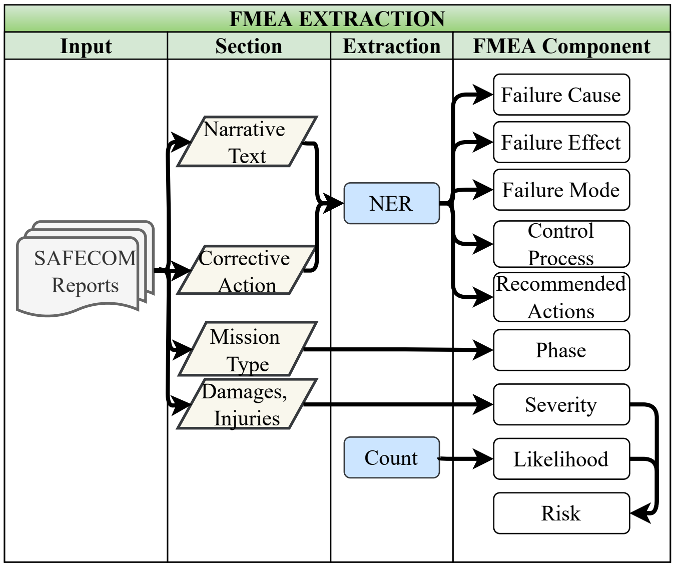

Fig. 5. Displays how each component of the FMEA is extracted from a SAFECOM report.

reports on various engineered systems, from space and aircraft to buildings operations. For example, there are a number of reports on the Challenger space shuttle accident, as well as documentation on best practices for specific systems (e.g., circuits). We assume here that failures are discussed in similar manners across different engineering report repositories. Of the 160 documents, 80% is randomly selected for training and 20% for validation. To test the model performance and provide an example output, we annotate the 180 SAFECOM reports involving UAS in wildfire operations. Raw annotations are transformed into BILOU (Beginning, Inside, Last, Outside or non-entity, Unigram) tags. BILOU style tags tend to result in better model performance [29] and provide detailed information for multi-token entities, such as the start token, middle token, and end token. For example, a failure mode annotation of the tokens “short circuit” would be tagged: “B-MOD” “L-MOD”. Next, the pre-trained *’bert-base-uncased’* model is fine-tuned by training on the annotated LLIS documents. BERT models have limits for the number of tokens in a document, so each document is decomposed to individual sentences to ensure all text is processed in training. The model is trained on an NVIDIA GPU for four epochs, resulting in a total train time of approximately three and a half minutes. The model uses a custom cross entropy loss function that is weighted according to the entity class balances in the training set. The training set is naturally imbalanced with 78,250 non-entity tokens, 6,548 recommendation tokens, 3,679 cause tokens, 2,587 failure mode tokens, 1,840 effect modes, and 1,377 control process tokens. Following training, the fine-tuned model is tested by predicting entity labels on the 180 annotated SAFECOM documents. The fine-tuned model is then saved for reuse on other documents.

D. FMEA Extraction

A Failure Modes and Effects Analysis (FMEA) is extracted from the SAFECOM reports according to Figure 5. The two free text sections of the reports, “Narrative Text” and “Corrective Action”, are combined prior to applying the custom NER model from Section III-C. From these sections, the NER model extracts the failure cause, mode, effect, control processes, and recommended actions for the FMEA. Raw entities extracted using the NER model are post processed to combine sub-word tokens into complete words, remove non-word entities, remove repeat words, and truncate the number of words displayed to a predetermined number for readability. Next, the “Mission Type” field of the reports is used directly for the “Phase” component of the FMEA. Reports with similar failure modes, such as reports on airspace intrusions, are manually clustered together to form a single row of the FMEA. Raw frequency is then calculated for each row, then converted to a likelihood score, L, from 1-5 in accordance to FAA Order 8040.4B [30] in Table II. Here a high likelihood score indicates greater risk. Severity, S, is calculated in Equation 1 and dependent on injuries, damages, and presence of hazardous materials. Similarly, a greater severity score indicates a greater risk. After examining the SAFECOM reports with UAS mishaps, we find none of the reports contain injuries, which is likely due to the unmanned nature of these missions. However, we still include injuries in the definition of severity as it is important to consider when quantifying failure consequences. Thus $S = 0$ indicates a negligible impact, $S = 1$ indicates either minor damage or hazardous materials threat, and $S = 2$ indicates damage with hazardous materials involved. Note that two higher values of S are conventionally considered in cases with: 1) multiple serious injuries, a fatality, and/or major damage; and 2) multiple fatalities and/or complete loss of the aircraft [30]. Risk, R is calculated as the product of likelihood and severity in Equations 2, with greater risk scores indicating higher risk levels.

TABLE II

Likelihood categories from FAA Order 8040.4B.

Level	Description	Rate
5	Frequent	$>$ 100 per year
4	Probable	$>$ 10 per year
3	Remote	$>$ 1 per year
2	Extremely Remote	$>$ 1 per 10 years
1	Extremely Improbable	$\leq$ 1 per 10 years

TABLE III

Inter-Annotator Agreement for LLIS and SAFECOM annotations using F1-score.

Entity	LLIS F1-score	SAFECOM F1-score
CAU	0.507	0.669
CON	0.465	0.538
EFF	0.649	0.691
MOD	0.526	0.540
REC	0.679	0.475
Average	0.589	0.590

$$\begin{aligned} S &= I + D + H \\ I &= \begin{cases} 1 & if \;\; injuries = true \\ 0 & if \;\; injuries = false \end{cases} \\ D &= \begin{cases} 1 & if \;\; damage = true \\ 0 & if \;\; damage = false \end{cases} \\ H &= \begin{cases} 1 & if \;\; hazardous \;\; materials = true \\ 0 & if \;\; hazardous \;\; materials = false \end{cases} \end{aligned} \tag{1}$$

$$R = S * L \tag{2}$$

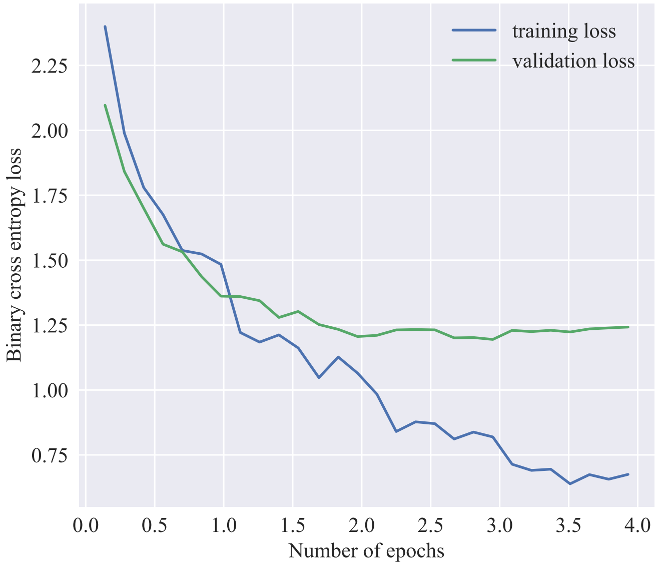

Fig. 6. Learning curves of training and validation loss during custom named-entity recognition model training.

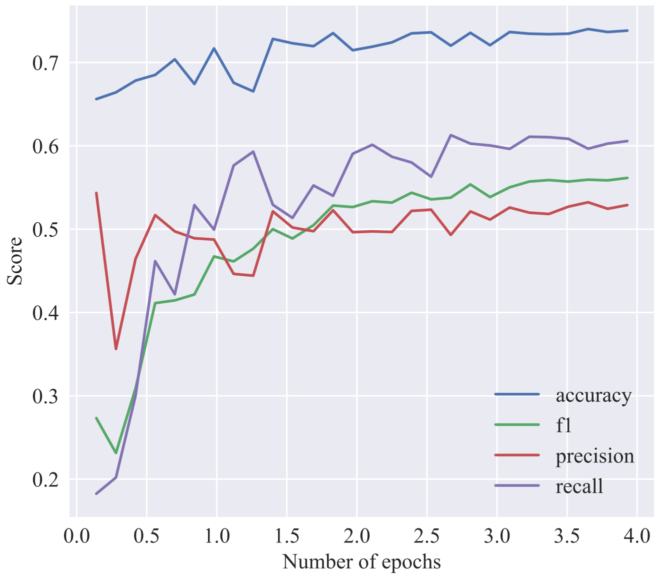

Fig. 7. Performance metrics on the LLIS validation set for the custom named-entity recognition model during training.

IV. Results

To examine the consistency of annotations between the authors, Inter-annotator agreement (IAA) is calculated. Both annotators tag the same twenty-five LLIS documents and eighteen SAFECOM documents. First, we calculate Cohen's Kappa (κ) [31], which is found to be $\kappa = 0.586$, indicating moderate agreement. Additionally, IAA is calculated using f1 [32] shown in Table III. Overall, the authors have similar average IAA score between the LLIS and SAFECOM data sets seen in Table III. Failure effects have the greatest agreement across both data sets, while recommendations and control processes are less consistent. Thus, differentiating between similar entities, such as control processes and recommendations, is difficult even for human annotators, and likely will also be difficult for the NER model as well. The training and validation set for the custom NER model are formed from annotated LLIS reports, and the test set is the 180 SAFECOM UAS reports. Learning curves for the custom NER model are shown in Figure 6, with training loss and evaluation loss graphed over the four epochs. The difference between the training loss and validation loss curves after epoch two increases in Figure 6, which indicates more training data may be needed. Performance on the validation set throughout training is shown in Figure 7. Accuracy is much higher than f1, precision, and recall, due to class imbalances. The scores due increase over time, which implies the model is learning the named entities.

In comparison to the training and validations sets, performance is lower on the unseen test set consisting of 180 SAFECOM reports on UAS mishaps in wildfire response. Metrics on the test set are in Table IV, with metrics on non-entity tokens excluded because they inflate the scores due to class imbalances. In general, the scores in Table IV are not

TABLE IV

Custom Named-Entity Recognition model performance on the test set of SAFECOM reports on UAS mishaps.

Entity	Precision	Recall	F-1	Support
CAU	0.31	0.19	0.23	1634
CON	0.49	0.34	0.40	3859
EFF	0.45	0.20	0.28	1959
MOD	0.19	0.52	0.28	594
REC	0.30	0.59	0.40	954
Average	0.41	0.32	0.33	9000

very high. The model performs best on identifying control processes and recommendations, with F1=0.40 for both. Precision is higher than recall for all entities except failure modes, thus the model tends to correctly identify entities more often than incorrectly identifying non-entities as entities. This is further seen in the confusion matrix in Figure 8, where the proportion of the true labels identified as a predicted label is shown. Correctly predicted labels are on the diagonal of the matrix, with recommendations and failure modes having over 50% of entities accurately predicted. For all entities the majority of false predictions are non-entity labels (“O”); however, failure causes also have a large proportion (27%) of entities incorrectly classified as failure modes. Overall, Figure 8 shows the custom NER model does successful generalize to the SAFECOM reports by correctly identifying entities, but the large amount of tokens mistakenly classified as non-entities is again an indication that more labeled training data is needed. While these quantitative measures of model performance could be improved, the qualitative output from the constructed FMEA exemplifies the usefulness of the custom NER model for failure entity extraction.

Manual clustering of the 180 SAFECOM reports on wildfire response UAS mishaps led to twenty-four distinct failure modes. Due to space limitations, a set of ten clusters is selected and the resulting FMEA generated using the custom NER model and method described in Section III-D is displayed in Table V. The cause, mode, effect, control process, and recommendations columns contain a truncated set of the named entities extracted from each cluster's reports with minimal post-processing. Most clusters have extracted entities that are relevant to the column, such as dislodged casings having cause “winds funneled under” and loss of line of sight (LOS) having an effect of “collided” with “tree”. In contrast, the causes extracted for UAS intrusions seem to be less relevant, where as the effect, control process, and recommendations are more intuitive. Each row of the FMEA also has the failure's likelihood (L), severity (S), and risk (R). Results indicate propeller arm disconnections are of the highest risk ($R = 3.0$), followed by loss of control ($R = 2.0$), and loss of GPS ($R = 1.2$).

V. Discussion

A custom named-entity recognition model trained to detect failure-relevant entities shows promise for extracting FMEA-style results from repositories of mishap reports. We successfully trained a custom model on limited data and applied it to an unseen data set to produce a study of UAS failures in wildfire response operations. The current model has difficulty differentiating failure causes from failure modes, which is also a known problem for human experts who manually construct FMEAs [33]. Regardless, the resulting FMEA provides valuable insight on possible failures of UAS in wildfire response operations with the majority of extracted entities corresponding with expectations.

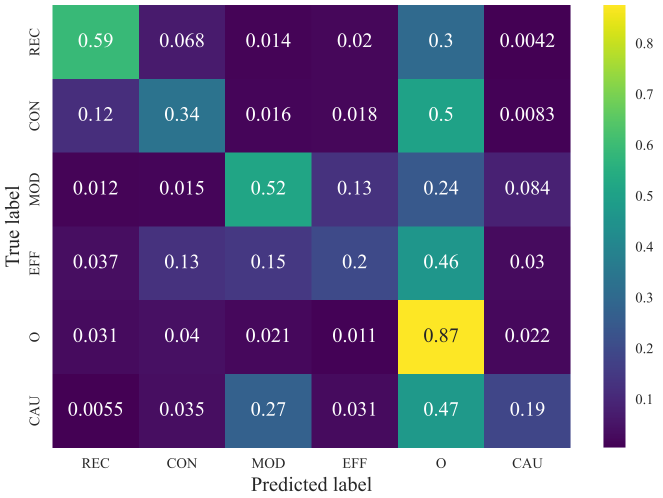

Fig. 8. Confusion matrix for named-entity recognition on the SAFECOM test set. Proportions shown are the amount of the true label that is identified.

While the results of this research show promise in leveraging BERT models for engineering information extraction tasks, like custom named-entity recognition, our custom model's performance is sub-par compared to conventional NER models. The failure-relevant entities our model extracts tend to be “long tailed”, consisting of multiple words and characters more so than conventional entities. In turn, our model would benefit from more extensive training with more labeled text, yet annotation is a costly and inconsistent process. Additionally, there are varying levels of granularity of FMEAs in practice, which may impact the model's performance. Some analyses are focused on the failure mode of an individual component in a mechanical system, where as other failure modes may be from a complex system as a whole. In other cases, there is a chain reaction of one condition causing another condition, which may cause another, eventually resulting in a failure mode with certain effects. Discerning failure cause from failure mode may be difficult for our custom NER model in these cases, as one failure mode may be the cause of another failure mode [33]. Despite the custom NER model successfully extracting the qualitative components of an FMEA discussed in this paper, there are other components of an FMEA that cannot be determined from this process. For example, FMEAs often include a column to discuss the probability of detection for a failure mode, with difficult to detect failures indicating higher risk. This can only be performed by an expert, and trying to approximate probability of detection with the automated

TABLE V

A portion of the FMEA extracted from SAFECOM reports using the custom Named-Entity Recognition model, including example SAFECOM ID numbers for each cluster. Extracted entities are separated by commas and come from the reports for that cluster.

Cluster	Phase	Cause	Mode	Effect	Control Process	Recommendations	L	S	R	ID
Battery	Reconnaissance; Infrared Imagery	button, not, could, issue, battery level status, showing	hard, landing, depleted, battery, 40 percent, battery level, sufficient power	dropping, 10, percent, lost, fell at close to free, fall	assumed manual, control, bringing it down, manually, the, uas	batteries will be, tracked on an individual, level, be, removed	2	0.33	0.67	17-0977
Casing dislodged	Reconnaissance; Infrared Imagery	fuselage, cowling on the aircraft was, missing, winds funneled under	battery hatch cover disengaged from the, motor failsafe, activated, dirt	hit, motors, shut down, the, small, nick, trap the, dirt	propeller and battery hatch cover were, replaced, redesigning the battery	None	2	0.00	0.00	21-0015
Hang Fire	Aerial Ignition	form of, visible, hang fire, functioned, melted, sphere, was still	a, hang, fire, on, aircraft gave a, hatch motion, error	in, flight, fires	vo assisted the pilot, resetting the ignis per, took control	follow, immediately using the, camera, identify any, ensure that you	1	0.00	0.00	20-0872
Loss of GCS	Aerial Ignition; Reconnaissance; Infrared Imagery	error, combination, thermal, signal, controller and, feedback, gcs did not	in flight, failure, gsc, disconnection, error, video, loss, motor, wine	immediately, ignis, warning, crash from, separated, motor, home, not, turned	reset the home, point, noted the gps, location, up, plan	management, pulling flight logs and, video, ensure that, are, done	3	0.33	1.00	21-0172
Loss of GPS on UAS	Other; Reconnaissance; Infrared Imagery	erratic, nose of the aircraft was pointed at, lack of	of, solo made contact with, solo lost, gps, winds, battery	experienced loss, gps, tree, loss of, control, and, crash, shifted	autonomous, regain manual flight, control, initiate ” return to home	should have been, suspended, or, cancelled, having eyes on the	3	0.40	1.20	21-0138
Loss of Line of Sight (LOS)	Aerial Ignition	had, lost, of the aircraft, position, and the, pad, could	with a, broken, broken arm locking, ignis housing was, cracked	aircraft, collided, tree, tilted and, fell about 15’to the, ground	a hand held led, light, spot the, pad, exactly, analysis	having the, visual observer 90, degrees, off of the landing	1	1.00	1.00	20-0949
Loss of control	Reconnaissance; Infrared Imagery	left wing aileron servo was, sticking, aircraft, hit, wall, refused	rapid and uncommanded, descent, roll and, aircraft, to quickly lose	steep, aircraft, dropped and hit the, feed and connection was	no, monitored instruments, programmed flight, terminate, land, troubleshoot, reviewing the	be, inspected and, tested, at, facility, compass and, micro	3	0.67	2.00	20-1042
Parachute Landing Failure	Infrared Imagery	chu, fully, parachute was packed, incorrectly, drogue chute was packed	deploy, partial, opening, the, canopy	hard, fuselage was, damaged, been	checked all parachute, on, confirmed proper	site, packing, use a, buddy, check	1	1.00	1.00	18-0821
Propeller arm disconnect (sheared bolt heads, etc.,)	Aerial Ignition; Aerial Ignition (Prescribed); Infrared Imagery	heads on a propeller bolt had, normal, aircraft, having difficulty	missing, separate bolt head had, sheared, loud, piece of, unknown	snap, descended and, impacted the, ground, 4, propeller, where, flight	photo, was, propeller assembly was, rebuilt, test flight, full, inspections	check propeller, bolts, and, document all bolt, failures, and potential	3	1.00	3.00	19-0298
UAS Intrusion	Water Drop; Aerial Ignition; Passenger Transport; Helitack; Initial Attack; Other; Retardant; Air-Attack; External Load; Leadplane; Reconnaissance	system, heavy, smoke, erratic fire, conditions, recreational, ua, dropping, water	firefighters, a, recreational type unmanned aircraft system	immediately, cease, leave the, catastrophic, flew out of, sight, ceased	tfr was put in place, was, uas, confronted, and, removed	wide circulation of, events, go, taken, maintain visual on, documented	4	0.07	0.27	16-0657

processes in this work may lead to incorrect estimations with potentially costly consequences.

Despite these limitations, the work presented in this paper is a step towards automatic information extraction from safety reports and can be a component of a scalable IASMS [5]. This method provides a means to track risk-relevant trends in mishaps, such as likelihood and severity. The custom NER model discovers knowledge and extracts specific safety relevant information, which can in turn be used to improve safety outcomes. By identifying the set of failure causes and effects, safety analysts can target specific known causes of failures and implement mitigation strategies with the range of effects in mind. Recommendations documented in safety reports by operators are also aggregated and can be leveraged by engineers. As the custom named-entity model improves with more training data from different datasets, it can be applied to a greater range of safety reports with higher quality information extraction.

VI. Conclusion and Future Work

In this research, we built a custom named-entity recognition model to extract failure-relevant entities, including failure cause, mode, effect, control process, and recommendations, from mishap reports. Entities identified from the custom model can be used to automatically construct a data-driven failure modes and effects analysis. This was achieved by fine-tuning a BERT model on annotated NASA Lessons Learned Information System documents. Our custom model performed satisfactorily, given the limited training data, with an average f1-score of 0.56 on training data. To test the model's generalizability, we applied it to a set of 180 SAFECOM mishap reports on UAS in wildfire response operation. The model performed acceptably on the test set, with a weighted average f1-score of 0.33. Finally, the entities extracted from the SAFECOM reports were synthesized into an FMEA detailing UAS mishaps in wildfire response operations. These results indicate airspace intrusions are the most common failure incident. However, other failure modes, such as loss of GPS, loss of control, and sudden battery drainage have an effect on mission outcomes.

Through the processes described in this paper, repositories of mishap reports are successfully leveraged to construct a data-driven FMEA. This process is part of a larger need to optimize the management of historical engineering knowledge. Currently, there is a vast amount of knowledge captured in mishap, near miss, and failure reporting systems, such as SAFECOM and the LLIS. Unfortunately, however, this information exists primarily in text-data and reports must be individually parsed and accessed using simple query methods. Instead, moving towards an intelligent knowledge-management system that can synthesize trends in reports in a digestible way will provide more valuable use to designers and safety analysts.

In the future, we hope to improve our custom named-entity recognition model for FMEA extraction through additional training. There is a large body of research using ontologies to extract failure information, and utilizing the information from these ontologies in training could improve model performance with little additional human annotation required. The model was only applied to a small subset of SAFECOM reports, and we would like to apply it to more SAFECOM reports, as well as other data sets. Previously, topic modeling methods have been used to extract FMEA-style results, and it would be insightful to compare the topic modeling results to the custom NER model results. Eventually, relation detection (RD) [22] should be implemented into the custom model in this research, specifically for detecting causal relationships. In turn, the NER with RD would allow for the construction of a knowledge graph, which may bridge together various failure modes, causes, and effects to understand complex system dynamics.

Acknowledgment

This research is supported by the System-Wide Safety (SWS) project in the Airspace Operations & Safety program (AOSP) in the NASA Aeronautics Research & Mission Directorate (ARMD). Any opinions or findings of this work are the responsibility of the authors and do not necessarily reflect the views of the sponsors or collaborators.

References

[1] Wildland Fire Lessons Learned Center, "2021 Incident Review Summary," 2022.

[2] P. E. Dennison, S. C. Brewer, J. D. Arnold, and M. A. Moritz, "Large wildfire trends in the western united states, 1984–2011," *Geophysical Research Letters*, vol. 41, no. 8, pp. 2928–2933, 2014.

[3] J. T. Abatzoglou and A. P. Williams, "Impact of anthropogenic climate change on wildfire across western us forests," *Proceedings of the National Academy of Sciences*, vol. 113, no. 42, pp. 11 770–11 775, 2016.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[5] K. K. Ellis, P. Krois, J. Koelling, L. J. Prinzel, M. Davies, and R. Mah, *A Concept of Operations (ConOps) of an In-time Aviation Safety Management System (IASMS) for Advanced Air Mobility (AAM)*. AIAA, 2021. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2021-1978

[6] K. K. Ellis, P. Krois, J. H. Koelling, L. J. Prinzel, M. D. Davies, and R. W. Mah, *Defining Services, Functions, and Capabilities for an Advanced Air Mobility (AAM) In-time Aviation Safety Management System (IASMS)*. AIAA, 2021. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2021-2396

[7] DRONERESPONDERS, "Spring 2020 public safety uas survey results," 2020.

[8] L. Martin, Y. Arbab, and J. Mercer, "Initial exploration of stereo (scalable traffic management for emergency response operations) system user requirements for safe integration of small uas," in *40th Digital Avionics System Conference*, October 2021.

[9] G. White, "Evaluation of systems to recognise and address safety issues promptly, effectively and universally and evaluation of systems which promote safe fire fighting behaviours and initiatives." Department of Natural Resources and Environment, Tech. Rep., 2002.

[10] NASA, "NASA systems engineering handbook," National Aeronautics and Space Administration, Tech. Rep. NASA SP-2016-6105, 2007.

[11] NASA, "Procedure for failure mode, effects, and criticality analysis (FMECA)," National Aeronautics and Space Administration, Tech. Rep. NASA-TM-X-65227, 1966.

[12] T. DiVenti, "Failure mode effect analysis (fmea) & critical items list (cil) glast lat anti-coincidence detector (acd) report," Goddard Space Flight Center, Tech. Rep. ACD-RPT-12001, 2001.

[13] Z. Rehman, C. Kifor, F. Jabeen, S. Naz, and M. Waqar, "Automatic acquisition of failure mode and effect analysis ontology for sustainable risk management," *Sustainability*, vol. 12, p. 10208, 12 2020.

[14] C. Spreafico and D. Russo, "A semi-automatic methodology for making fmea surveys," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 6, pp. 79–102, 01 2021.

[15] Z. Wang, B. Zhang, and D. Gao, "A novel knowledge graph development for industry design: A case study on indirect coal liquefaction process," 2021. [Online]. Available: https://arxiv.org/abs/2111.13854

[16] R. Kamath, "Applying bilstm and cnn to assist actionable ontology building from fmea documents," Master's thesis, School of Science, 2019.

[17] NASA, "NASA Public Lessons Learned Information System," https://llis.nasa.gov/, 2020.

[18] S. R. Andrade and H. S. Walsh, "Knowledge discovery for early failure assessment of complex engineered systems using natural language processing," in *ASME 2021 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2021)*, 2021.

[19] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[20] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[21] Z. Liu, F. Jiang, Y. Hu, C. Shi, and P. Fung, "Ner-bert: A pre-trained model for low-resource entity tagging," 2021. [Online]. Available: https://arxiv.org/abs/2112.00405

[22] N. Bach and S. Badaskar, "A review of relation extraction," *Literature review for Language and Statistics II*, vol. 2, pp. 1–15, 2007.

[23] W. Ali, W. Zuo, R. Ali, X. Zuo, and G. Rahman, "Causality mining in natural languages using machine and deep learning techniques: A survey," *Applied Sciences*, 2021.

[24] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[25] R. Ramachandran and K. Arutchelvan, "Named entity recognition on bio-medical literature documents using hybrid based approach," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–10, 2021.

[26] P. Corbett and J. Boyle, "Chemlistem: chemical named entity recognition using recurrent neural networks," *Journal of cheminformatics*, vol. 10, no. 1, pp. 1–9, 2018.

[27] NASA, "Fault management handbook," National Aeronautics and Space Administration, Tech. Rep. NASA-HDBK-1002, 2012.

[28] H. Nakayama, T. Kubo, J. Kamura, Y. Taniguchi, and X. Liang, "doccano: Text annotation tool for human," 2018, software available from https://github.com/doccano/doccano. [Online]. Available: https://github.com/doccano/doccano

[29] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, 2009, pp. 147–155.

[30] FAA, "Safety risk management policy order 8040.4b," 2017.

[31] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.

[32] A. Brandsen, S. Verberne, K. Lambers, M. Wansleeben, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck *et al.*, "Creating a dataset for named entity recognition in the archaeology domain," in *Conference Proceedings LREC 2020*. The European Language Resources Association, 2020, pp. 4573–4577.

[33] H. Yu, G. Zhang, and Y. Ran, "A more reasonable definition of failure mode for mechanical systems using meta-action," *IEEE Access*, vol. PP, pp. 1–1, 12 2018.