In: 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), Portsmouth, Virginia, 18-22 September 2022
DOI: 10.1109/DASC55683.2022.9925822

An NLP framework for extracting causes, consequences, and hazards from occurrence reports to validate a HAZOP study

Jon Ricketts
Cranfield University
United Kingdom
j.ricketts@cranfield.ac.uk

Jonathan Pelham
Cranfield University
United Kingdom
j.g.pelham@cranfield.ac.uk

David Barry
Cranfield University
United Kingdom
d.j.barry@cranfield.ac.uk

Weisi Guo
Cranfield University
United Kingdom
weisi.guo@cranfield.ac.uk

***Abstract*— A substantial amount of effort and resource is applied to the design of aircraft systems to reduce risk to life and improve safety. This is often applied through a variety of safety assessment methods, one of which being Hazard and Operability (HAZOP) Studies. Once an air system is in-service, it is common for flight data to be collected and analysed to validate the original safety assessment. However, the operator of the air system generates and stores a substantial amount of safety knowledge within free-text occurrence reports. These allow maintainers and aircrew to report occurrences, often describing hazards and associated detail revealing consequences and causes. A lack of resource means it is difficult for safety professionals to manually review these occurrences and although occurrences are classified against a set taxonomy (e.g., birdstrike, technical failure) this lacks the granularity to apply to a specific safety analysis. To resolve this, the paper presents the development of a novel Natural Language Processing (NLP) framework for extracting causes, consequences, and hazards from free-text occurrence reports in order to validate and inform an aircraft sub-system HAZOP study. Specifically using a combination of rule-based phrase matching with a spaCy Named Entity Recognition (NER) model. It is suggested that the framework could form a continual improvement process whereby the findings drive updates to the HAZOP, in turn updating the rules and model, therefore improving accuracy and hazard identification over time.**

Keywords*— *hazard analysis, safety, assurance, safety assessment, natural language processing

I. INTRODUCTION

A substantial amount of effort and resource is applied to the design of aircraft systems to reduce risk to life and improve safety. This is often applied through a variety of safety assessment methods such as Failure Mode Effect Analysis (FMEA), Fault Tree Analysis (FTA) and Event Tree Analysis (ETA). These methods are enshrined into engineering standards such as ARP4761 (Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment) ensuring their widespread, standardized application. Ultimately the analysis and results are used to justify the given safety of the system.

Once an air system is in-service, data is collected and analysed to validate the original results of the selected methods. This can be performed in a variety of ways, for example; it is common for manual review and analysis of numerical, technical data (e.g. structural health monitoring). However, the operator of the air system generates and stores a substantial amount of safety knowledge within natural language (free-text). This is not always readily accessible to the organization that performed the original safety analysis and could represent a source of knowledge that remains under ultilised.

A. Problem

UK military aviation use the Air Safety Information Management System (ASIMS) which allows maintainers and aircrew to record occurrences, often describing hazards and associated detail revealing consequences and causes. A lack of resource means it is difficult for safety professionals to manually review all these occurrences and although classified against a set taxonomy (e.g., birdstrike, technical failure) this lacks the granularity of a specific safety analysis. A further complication is the disconnect between the various organizations involved in an air system. It is not uncommon for the design organization to receive limited feedback on in-service issues. Although the previously mentioned safety assessment methods are well established, Dallat et al [1] argues that the majority of risk assessment methods in use today are not consistent with currently accepted models of accident causation. The methods are primarily for assessing the design and functionality of hardware/software, failing to account for the encompassing socio-technical system that would influence safety. This is an area that Natural Language Processing (NLP) can exploit by providing insight to both technical issues and the encompassing safety management system.

This paper presents an NLP framework for extracting causes, consequences, and hazards from free-text occurrence reports matching them to a Hazard and Operability (HAZOP) study. This validates the HAZOP against real-world events, assisting safety professionals with updating the HAZOP or instigating design modifications.

II. BACKGROUND & RELATED WORK

This paper brings together three areas of research; NLP, safety occurrence reporting and safety analysis tools (in this case a HAZOP).

A. Safety occurrence reports

Occurrence reports are completed in response to incidents, near-misses and accidents. In terms of aviation safety, ‘an occurrence means any safety-related event which endangers or which, if not corrected or addressed, could

endanger an aircraft, its occupants or any other person' [2]. They will typically describe the event and other pertinent information such as hazards, causes and consequences, all of which is important for understanding the safety of the system.

Not only are there regulatory requirements that stipulate the need and processes for occurrence reporting but the following benefits can be realized from our novel research (derived from Johnson [3]):

- Ultimately, it helps us to understand why accidents do not occur. A dangerous occurrence may reveal whether existing controls were effective or ineffective.
- High frequency incident reporting provides insights into human error, system failures, and regulatory weakness. This allows a statistically reliable quantitative insight into the occurrence of factors or combinations of factors giving rise to incidents or accidents [4].
- Maintains a certain level of alertness to danger, especially when the rates of actual injuries and other accidents are already low within an organization [4]. To put it simply, it serves to remind us of the hazards.
- Encourage staff participation in safety improvement.
- Lessons learnt can be shared with other areas of the organization or external bodies.
- Ultimately an occurrence is cheaper than the cost of an accident, best put via the well-used phrase; 'If you think safety is expensive, try an accident' [5].
- Measure of safety culture. The quality of occurrence reports, how they are dealt with, and quantity can indicate the type of safety culture within an organization.

B. *Safety analysis tool - HAZOP*

A number of safety analysis tools could have been selected for this paper, however, HAZOP studies are commonly used across multiple industries, while featuring similarities to other popular methods such as Bowties and FMEA. A HAZOP is a structured method for identifying hazards and problems that would prevent efficient operation [6]. It generally involves a team who review the operation of a system using guidewords to prompt thinking of 'what could go wrong?' during operation, in turn identifying hazards and possible mitigation.

Upon completion of the HAZOP, a table of hazards, causes, consequences and mitigations is generated (example shown in Table I). This data is based upon the team's knowledge and underlying assumptions. Therefore, it can be subject to biases or fail to foresee all hazards applicable to the system, thus validation of the HAZOP is required which is where NLP provides an opportunity, potentially revealing missed events or revealing a realistic likelihood/severity value. It is good practice to regularly review a HAZOP, this ensures that the original study was accurate and also captures new issues as systems, people and intended usage evolve over time. A further benefit to adopting NLP for performing this review is that it can monitor the system in real-time, providing instant feedback and negating the need for resource deployed to such an activity.

Table I – Example hazard taken from HAZOP study

Hazard	Cause	Consequence	Mitigation
No breathing gas from main regulator	Failure of main regulator	Hypoxia	Readily identified by pilot. Emergency Procedures call for switch to 100% regulator and immediate descent.

C. *Application of NLP*

NLP is a field concerned with the ability of a computer to understand, analyse and manipulate human language [7]. NLP overlaps fields such as artificial intelligence, linguistics, formal languages, and compilers.

Natural language (as spoken, written or typed) is complex, comprising of a catalogue of words (lexicon) alongside structural rules (grammar) allowing meaning when combining the words into sentences [8]. Occurrence reports are often presented in structured data form which then contains unstructured data (e.g. free text fields). This can be difficult for machines to interpret due to the use of grammar, specialist terms, and acronyms; relevant only to the industry in question.

A key problem with technical text is the terse language, polysemy, and expansive use of acronyms which creates a real challenge for NLP tasks. Solutions to this problem have been demonstrated by Butters et al [9] who developed a support capability for technical documentation by suggesting and standardizing technical terms therefore improving information retrieval. However, such a solution is complex and expensive to implement. Alternatively, modified tokenizers and part of speech taggers can be used, as demonstrated by Bokinsky et al [10] to assess the viability of extracting information from helicopter maintenance records with a view of improving technical documentation. Ultimately, resource and time is required to create an accurate standardization method. However, once setup, accurate standardization is key to the proceeding steps or machine learning model and should enhance the results.

In order to achieve the aim of this paper, the NLP method needs to successfully identify and label terms within the text as hazard, cause or consequence. Phrase matching methodologies and Named Entity Recognition (NER) are strong contenders as a solution being that it seeks to semantically recognize and identify the occurrences of a given, predefined phrase in an annotated text [11].

A number of off-the-shelf packages exist for NER such as Gensim, spaCy and Natural Language Toolkit (NLTK). Safety occurrence reporting is a specialized topic, therefore these packages cannot just be deployed to recognize specialized safety-related terms with the expectation of usable results. Instead, data must be analysed, cleansed, and typically annotated before an NER model is constructed.

Previous applications of NER to safety incident and accident reports include research into using a link grammar parser and basilisk bootstrapping algorithm to recognize entities in health and safety reports [11]. While NER was used by Razavi et al [12] to identify features such as time and date from text in order to determine risk within the maritime domain. Achieving a similar aim to NER, phrase

matching work has been undertaken to classify railway hazard reports against elements of a bowtie [13] which featured an element of n-gram extraction. Tixier [14] developed an automatic content analysis tool, using programmatical rules and dictionaries to scan text from construction accident reports, returning attributes such as injury type, injured body part and energy. Following trial and refinement, this tool reached an overall accuracy of 95% which demonstrates that the optimum method is not always the most complex.

Thompson et al [15] used the APLenty web-based annotation system, producing a model capable of labelling harmful consequences and hazards, among others. Trained on 600 annotated sentences, this system reached an average F-score of 0.79. It was suggested that this could be improved with the annotation of a larger corpus and incorporation of further ontology/terminological resources.

III. METHOD

This paper focuses on occurrence reports generated from the operation of the Royal Air Force Tornado aircraft. Rather than attempt to label and examine all hazards present within the air system (which would be a time-consuming task), a sub-system of the aircraft was selected as a proof of concept; The Tornado Life Support System (LSS), of which there were 437 occurrence reports recorded from 2009 - 2019.

A. Occurrence data standardisation

Occurrence reports from ASIMS were made available for this study. ASIMS data includes five descriptive fields that are of interest for this study (Table II).

Table II – ASIMS text fields used within the study

Field	Description
Description	Firsthand report of the occurrence, describing the incident/accident and any additional information. Typically where the 'consequence' is described.
Investigation and Rectification work	Describes the occurrence from a technical perspective highlighting any preliminary investigation findings, associated on-going work and mitigation strategies.
Outcome Narrative	Describes the solution to the occurrence (if any).
Cause Narrative	Textual description as to the cause(s) of the occurrence.
Causal Fact 1 Narrative	One causal factor must be identified to progress the occurrence. Therefore, this field will often be a duplicate of the 'Cause Narrative'.

The investigative and cause narratives are key to providing a 'cause' while a 'consequence' can often be obtained from the description narrative as this is what the aircrew/maintainers experienced.

The Python Pandas Numpy and Regular Expression libraries were used to merge the data of the fields into a continuous single string for each occurrence, with 'Causal Fact 1 Narrative' and 'Outcome Narrative' ignored if it was a repeat of the 'Cause Narrative' and 'Description' respectively. ASIMS was found to be used inconsistently over time and by the various organizations, resulting in duplication of mandatory fields. The use of a continuous string helped overcome any misuse of ASIMS by framing each occurrence into a simpler 'consequence – investigative work – cause' layout.

Although it is tempting to apply spelling correction to the text, this was deemed too risky with the sheer number of terms which would not have been encountered by an 'off the shelf' spelling corrector. Therefore, it was decided that dictionaries would be developed to correct any spelling errors and standardize terms.

In order to construct a dictionary for standardizing the text, another Python library; pyspellchecker was used to scan the new strings and create a list of any miss-spelt words. This returned a list of 32,000 words, which naturally contained many acronyms, technical terms and textual nuances that the python library had never encountered before. Unfortunately, neither the time nor resource was available to assess each potential spelling error, so Pandas was again used to find the frequency of potential miss-spelt words across all occurrences. This allowed for the most frequent miss-spellings to be assessed and compiled into a dictionary. For example, the most common miss-spelt English word was 'occurred', spelt 'occured', while variations of acronyms and contractions were common place, e.g. 'left hand' appears as 'lh', 'l/h', 'left-hand', 'lhs'. The dictionary would take the latter example to standardize all left hand references to simply 'left hand'. Particular focus could also be given to terms associated with the LSS to further improve accuracy.

Two dictionaries were compiled using terms from the miss-spelt words and lists of technical and operational acronyms where each miss-spelling/acronym formed the key and the associated value containing the standardized term. The second dictionary was required to contain escape keys such as slashes and dots, applied with slightly different code to prevent Python regular expressions (regex) producing errors or stopping. These two dictionaries were then applied to every newly merged occurrence returning lower case, standardized text with no punctuation.

A vital element of the study is to have a labelled dataset which the results can be assessed against, providing accuracy and precision measures. For the phrase matching method, the HAZOP study forms the starting point where each occurrence needs to be read and (where applicable) matched to a cause, consequence and hazard from the HAZOP. This was completed by several Safety Engineers who were provided with the dataset and the LSS HAZOP, classifying 200 occurrences.

B. Rule based phrase matching

The general steps for developing and deploying the rule-based phrase matching are shown in Figure 1. This human guided approach was selected due to the uniqueness of the data and author expertise.

Following text standardization, further text cleaning and processing was required for the rule based phrase matching methodology, comprising of:

- Tokenization: Representing the narrative as a list.

- Stop word removal: Simplifies the narrative by removing words that add little value such as ‘the’, ‘and’, ‘it’, etc.
- Lemmatization: Reduces words to their base form. Unlike a stemmer, lemmatization is more lenient, reducing the risk of incorrectly trimming certain aviation terms.

This returned the text for each occurrence as a list that could be parsed by the Python code.

The rules are written as simple IF and ELIF operands where each occurrence is scanned for particular keywords. If a keyword is found then the surrounding window of words is scanned for terms that support the keyword. This aims to provide confidence that a given cause or consequence took place and is not merely being mentioned alongside the actual cause/consequence.

To develop these rules, the key terms and supporting terms need to be identified. The HAZOP study was used as the starting point for developing these. For example, ‘hypoxia’ was a consequence to several hazards. This is a specific term, where aircrew are trained to recognize the symptoms and positively identify hypoxia as a consequence within the occurrence. This can then be affirmed with supporting terms such as ‘felt’, ‘onset’, ‘pilot’. If these terms are found in a given window then it is likely that hypoxia was the consequence within the occurrence.

Other consequences present a greater difficultly where terms may relate to other aircraft systems or events. An example is the cockpit depressurisation where the key term ‘depressurisation’ could relate to several systems such as hydraulics or pneumatics, therefore more reliance is placed on the supporting terms to positively identify this as a consequence within the bounds of the LSS. In this instance, the rule was amended to ignore the ‘depressurisation’ key term if it appeared with terms associated with hydraulic and pneumatic systems.

A key element in creating the rules was to trial and review them during their design to ensure that they functioned as anticipated and the selected terms were appropriate. Not only did this activity reveal new terms to use but also influenced the construction of the code. Once the rules were refined, they could be deployed across the occurrence reports.

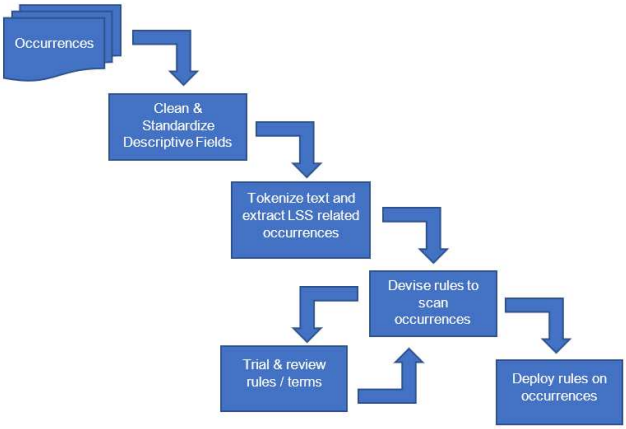

Fig. 1. Process steps for developing rule-based phrase matching code

Functions were created to identify causes and consequences, returning these pre-matched to the HAZOP study. In turn, a further function determined the pre-set hazard based upon the cause and consequence.

C. NER – causes & consequences

Unlike the rule based method described previously, training and implementing a bespoke spaCy NER model is somewhat easier. The occurrences reports must still be standardized, however further processing steps and extensive code writing is not necessary. Rather, a batch of occurrences must be annotated with causes and consequences to form training data. The critical and time consuming element is to correctly label entities within the records.

Prior to annotating the text with labels, an annotation scheme was developed in order to ensure consistency when labelling entities and enable future iterations to follow the same process. The annotation is relatively simple with only two entities; ‘cause’ and ‘consequence’ to be labelled in each occurrence. The following definitions were used to guide the annotator:

Cause: Factor(s) which directly led to the occurrence (usually revealed within the technical investigation element of the occurrence).

Consequence: The outcome of the occurrence, or more precisely the ‘outcome of an event affecting objectives’ [16].

A difficulty lies with the typical knowledge of the reporter, often the occurrence will describe the incident and consequence. This means that the cause or consequence may not be readily identifiable within the text, there may even be several present within the occurrence.

A further consideration is the span of each entity, or range of words that form the entity, as discussed by Thompson [15]. For example, adverbs may be included as these provide more detail while verbs may not need inclusion as they do not provide any additional value. Affirmative statements such as ‘shut off valve identified as the cause’ are clearly identifiable as the cause, while in other occurrences it may not be so clear, and subjective, e.g. ‘suspect shut off valve’.

Due to restrictions in resource, the corresponding author (with an aviation safety engineering background) was the sole annotator for the occurrences with a sample of 258 occurrences annotated.

A production-ready framework, ‘Rubrix’ was used to annotate the occurrences [17]. This offered several advantages, one of which is that it provides a user-friendly, easily sharable interface where the annotator(s) can simply highlight the text and select which entity applies. Negating the need for the annotator to have knowledge of Python. Secondly, the framework has been designed to work with a number of main-stream libraries such as spaCy, Hugging Face and FlairNLP which altogether allows for increased functionality and future analysis options.

The natural language open source tool ‘spaCy version 2.2.3’ was used to identify causes and consequences. The operation of spaCy can be broken down into a four step process (see figure 2); initially terms are embedded via a bloom filter into a continuous vector space [18]. Next, a convolutional neural network is used to encode the terms into

a sentence matrix [19] therefore taking context into account. The third step takes an input query vector providing a problem specific representation, deciding which parts are more informative. The final step is to predict the cause/consequence labels, which is achieved through a multi-layer perceptron process.

The annotated training data was loaded into a blank English language model (en_core_web_sm), an NER pipeline was then prepared complete with an entity recognizer. A dropout rate of 0.5 was selected to make it harder for the model to memorise training data and avoid overfitting. It is then a matter of looping over each occurrence where a prediction is made and checked against the annotations. If the model is incorrect, the weights are adjusted to improve the result.

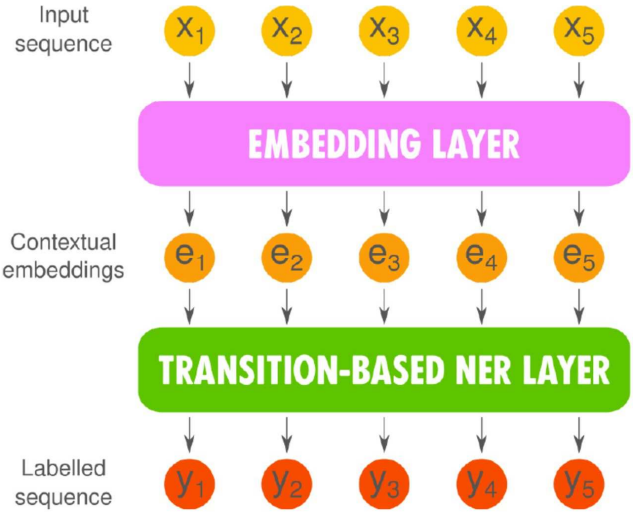

Fig. 2. General neural architecture for spaCy NER [20]

Once the model was trained, two functions were created to iterate over each occurrence within a csv file and create two columns with the predicted cause and consequence. This could then be easily used for onward review and analysis.

IV. RESULTS

A. *Phrase matching results*

The results of the phrase matching rules were validated against the aforementioned dataset which contained 200 assessed occurrences, producing a precision score of 0.73, recall score of 1 and F1 score of 0.84. The high recall score provides reassurance that the rules are not discounting occurrences, however, a high number of false positives are produced representing disagreements between the rules and human assessors.

Out of 437 occurrence reports available for this study, the rule based phrase matching revealed a count of hazards. Over a quarter of the occurrences were 'not applicable' and do not relate to the LSS HAZOP. While the most common hazards relate to temperature control valve and bleed air shut off valve failures. 51 occurrences or 'new hazards' are of interest to safety engineers, being that these represent cause – consequence combinations that are not in the HAZOP. It is these that can be assessed, and if accurate, added into future iterations of the HAZOP and code.

A difficulty lies with the hazards recorded where only a sole cause or consequence was identified. Ultimately these require further assessment to;

- Determine they actually relate to the LSS.
- Identify the missing cause or consequence – which is where the NER method is deployed.

The fact that a large proportion of occurrences cannot be linked to specific HAZOP hazards through the use of rules is not surprising. Especially when we consider the number of possibilities and variations apparent in the operation of an air system, coupled with HAZOP studies not being an exact science. However, it is here the method assists safety professionals by breaking down free-text occurrences to simple causes/consequences allowing for more expedient analysis. Further options include the ability to track a given hazard(s) over time. Alongside operational data, we can ask questions such as does operating low-level or in hot and humid conditions increase the likelihood of a given hazard occurring?

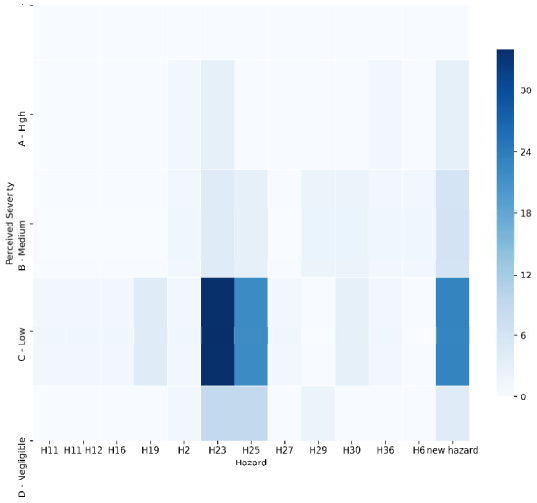

Fig. 3. Hazard vs perceived severity heatmap

Additional data analysis can easily be performed such as using the Python Seaborn library to produce a heatmap (Figure 3) for the quantity of extracted hazards vs perceived severity (Note – perceived severity is recorded by the reporter and not derived from the NLP analysis). This can be reviewed by safety professionals as a live risk matrix and provide justification as to where time and effort should be spent to reduce hazards.

B. *NER results*

SpaCy evaluation metrics were used to evaluate the developed NER model against the annotated set (Table III).

Table III – NER model evaluation metrics

	Precision	**Recall**	**F1 score**
Overall	80.16	66.67	72.79
'Consequence' entity	85.71	85.47	85.59
'Cause' entity	66.20	38.84	48.96

The NER model was able to predict consequences relatively well, this is assisted by the fact that the majority of occurrences describe the consequence as it is easier for the reporter to state what happened, i.e. the effect. Causes were predicted with much less accuracy, hampered by not always being readily stated within occurrence and ambiguous language (e.g. 'it was suspected…'). Plus, the cause is not always identified due to no fault found events. A sample of the entities from the NER model are shown in Table IV.

Table IV – Sample of consequence and cause entities

Consequence	Cause
'environmental control system temperature caption illuminated'	'main temperature control valve stuck'
'environmental control system temperature caption illuminated' 'cabin pressurisation and flow'	'circuit breaker 235 was found tripped'
'cabin pressure was lost'	'circuit breaker 235 tripped' 'bleed air shut off valve failure'
'burning smell' 'environmental control system t caption illuminated'	'cold air unit were removed on bae advice'
'blue haze between the back' 'acrid smell' 'stinging of the eyes' 'electrical smell'	'oil leak within the engine control unit would have produced'

In order to understand if the evaluation metrics (Table III) could be improved, a further 50 occurrences were processed, annotated and added to the training data. Unlike the original annotated batch, these occurrences did not solely relate to the LSS but featured an array of aircraft systems.

The evaluation metrics from the updated model are shown in Table V. Although some improvement was shown within the 'cause' entity, the 'consequence' entity suffered and has such the overall scores were lower than the initial model. This demonstrates that it could be beneficial to have one NER model per aircraft system which can process the language and typical text related to the given system.

Table V – NER model evaluation metrics with additional non-LSS related training data

	Precision	Recall	F1 score
Overall	74.87 (-6.6%)	65.16 (-2.3%)	69.68 (-4.3%)
'Consequence' entity	79.29 (-7.5%)	78.89 (-7.7%)	79.1 (-7.6%)
'Cause' entity	66.16 (none)	46.18 (18.9%)	54.39 (11.1%)

V. Discussion

The framework has several features that may be of use to safety professionals, especially those responsible for processing large amounts of textual occurrence/incident reports. Additionally, safety engineers who create and maintain safety analysis artefacts might find the method of validation a useful insight to determine the accuracy of the given artefact, highlight improvements to technical publications or event indicate where design modifications are required.

For creation of the phrase matching rules, the time taken to decide terms and tune the ruleset should not be underestimated. A proportion of occurrences were manually reviewed to create the labelled dataset prior to several rule iterations being run to refine and increase the accuracy of the rules. However, once the results are of an acceptable standard, the method instantly shows the regular occurring hazards and what mitigations are effective. This provides granularity in reporting rather than reliance on classification against a set taxonomy. The majority of occurrences used within this study would have traditionally been logged against 'Technical – Environmental Control System' category, which of course does not reveal the actual issues found by aircrew and maintainers. An element of further work could be to enhance the phrase matching rules with a neural model. This method was explored by Magnolini et al [21] and produced promising results, outperforming more conventional approaches. An advantage of programmatical rules (especially from a safety perspective) is that it is easy to understand how they work and therefore verify the code.

The two approaches work well in combination where a first review is conducted by the rules, providing causes/consequences that link directly to the HAZOP. A second pass is then completed by the NER model, this assists in identifying causes/consequences which are not detailed in the HAZOP. For example, several occurrences contained the cockpit not pressurising has a consequence, which was identified by the rules. However, no cause was returned. The NER model identified the cause for these occurrences to be due to static maintenance blanks being left attached to the system. This is one example where the framework has identified a new issue not recorded by the HAZOP.

Overall, there is scope for this framework to form a continual improvement, iterative process (Figure 4). Whereby the HAZOP is created alongside phrase matching rules. The framework described in this paper is then deployed where the results from the entity recognition can generate updates to the HAZOP and rules. The updated framework can be deployed again in the future, hopefully becoming more accurate and capturing more unique hazards.

Several of the occurrences are ambiguous, and as such have been interpreted differently by both the human assessor and the rules. One such example related to a 'fumes' consequence (of which both the human assessor and rules

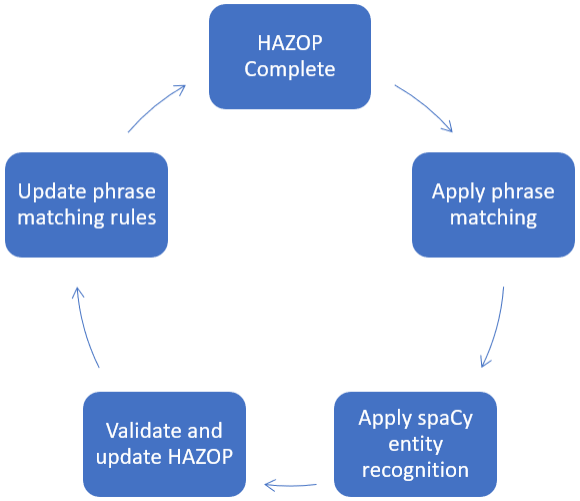

Fig. 4. Iterative process of maintaining rules, NER model and HAZOP study

matched). However, the cause of the fumes was due to contamination, from an oil leak, originating from a ruptured pipe. The human assessor recorded the cause as

'contamination' while the rules selected 'failed pipe'. In this example both causes are correct, although it could be argued the rules more appropriately selected the root cause.

The adoption of spaCy to train a bespoke NER model is quicker to setup, although time is required to review the results and link these to the HAZOP. However, the NER Model is not rigidly programmed like the rule-based approach and will therefore attempt to identify new entities and overall provide more diverse results.

An option is to use this method to help complete HAZOP studies from the outset, where the historical analysis and results can be reviewed alongside the guidance to indicate where flaws in the design or operation exist.

Further insight could be gained by focusing upon 'what went well?', a topic also identified by Baker et al [22]. Occurrence reports, by their nature capture the outcome of a negative event, i.e. something has gone wrong or did not work. There are likely to be multiple occurrences where a mitigation has performed well however there is no requirement to report this and if there was, it would add a further time and cost burden into the safety management system. A potential solution may lie with incorporating maintenance records and other non-safety textual reporting into the occurrence dataset, attempting to access tacit knowledge held by aircrew and maintainers.

VI. CONCLUSION

This paper has introduced a framework for classifying occurrences against a HAZOP study through a combination of a rule based phrase matching approach and bespoke NER model. The authors are not aware of any NLP systems that seek to directly match occurrence reports to safety analysis artefacts, therefore this framework forms a novel and useful tool for safety professionals to understand the accuracy of the safety analysis, observe trends and identify new causes, consequences and hazards which are only revealed through operation of the air system.

The granularity that can be provided through identifying specific hazards rather than relying upon existing classification taxonomies is of real use to safety practitioners as it negates the need to trawl through occurrence report text and manually cross-refer to a safety analysis. Once the framework is setup it can quickly produce results as to when hazards are occurring, the effectivity of existing mitigations and identify unforeseen issues.

A limitation of the framework is that it does require human oversight and could not be fully automated. This is predominantly due to a high false positive rate for the rule based phrase matching and low F1 score for the NER model identifying causes within occurrence reports. It is envisaged the NER model evaluation metrics could be improved if additional data was available and thus, more training data provided to the model. Unfortunately, this was not possible for the selected air system, a data science solution to this could be the deployment of a Generative Adversarial Network to artificially increase the size of the dataset [23]. Where a further limitation of this study was the sparse occurrence dataset, an area of future work is to enrich the occurrence data with maintenance records and flying hour records, with the aim to build a more detailed risk picture.

Although this paper focuses solely on one sub-system of an aircraft, it is expected that the framework could be repeated across the remaining sub-systems alongside unique rule sets and NER models reviewing incoming occurrence data. An additional classification step could be introduced to divide the mass of occurrence reports against the individual systems. Furthermore, this framework need not be restricted to aviation but could be deployed for any safety critical industry.

ACKNOWLEDGMENT

J Ricketts thanks the contribution of the IMechE Whitworth Senior Scholarship Award, QinetiQ and the Royal Air Force in supporting this research.

REFERENCES

[1] C. Dallat, P. M. Salmon, and N. Goode, "Risky systems versus risky people: To what extent do risk assessment methods consider the systems approach to accident causation? A review of the literature," Saf. Sci., vol. 119, pp. 266–279, 2019, doi: 10.1016/j.ssci.2017.03.012.

[2] EASA, "Aviation Safety Reporting," 2020. [Online]. Available: https://www.easa.europa.eu/domains/safety-management/aviation-safety-reporting. [Accessed: 15-Nov-2020].

[3] C. W. Johnson, "A Handbook of Incident and Accident Reporting," Fail. Safety-Critical Syst., vol. 1, pp. 1–1000, 2003.

[4] T. Van der Schaff, Near Miss Reporting as a Safety Tool. Butterworth Heinemann, 1991.

[5] I. Damnjanovic and W. Røed, "Risk management in operations of petrochemical plants: Can better planning prevent major accidents and save money at the same time?," J. Loss Prev. Process Ind., vol. 44, pp. 223–231, 2016, doi: 10.1016/j.jlp.2016.09.012.

[6] T. Kletz, HAZOP and HAZAN, Fourth. Institution of Chemical Engineers, 1999.

[7] R. Singh et al., "A Framework for Early Detection of Antisocial Behavior on Twitter Using Natural Language Processing," in Complex, Intelligent, and Software Intensive Systems Proceedings of the 13th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2019), 2020, pp. 484–495.

[8] C. Manning and H. Schutze, Foundations of Statistical Natural Language Processing. MIT Press, 1999.

[9] J. Butters and F. Ciravegna, "Authoring technical documents for effective retrieval," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6317 LNAI, pp. 287–300, 2010, doi: 10.1007/978-3-642-16438-5_20.

[10] H. Bokinsky et al., "Application of natural language processing techniques to marine V-22 maintenance data for populating a CBM-oriented database," Am. Helicopter Soc. Int. - Airworthiness, CBM HUMS Spec. Meet. 2013, pp. 463–472, 2013.

[11] Y. Sari, M. F. Hassan, and N. Zamin, "A hybrid approach to semi-supervised named entity recognition in health, safety and environment reports," Proc. - 2009 Int. Conf. Futur. Comput. Commun. ICFCC 2009, pp. 599–602, 2009, doi: 10.1109/ICFCC.2009.52.

[12] A. H. Razavi, D. Inkpen, R. Falcon, and R. Abielmona, "Textual risk mining for maritime situational awareness," 2014 IEEE Int. Inter-Disciplinary Conf. Cogn. Methods Situat. Aware. Decis. Support. CogSIMA 2014, pp. 167–173, 2014, doi: 10.1109/CogSIMA.2014.6816558.

[13] P. Hughes, D. Shipp, M. Figueres-Esteban, and C. van Gulijk, "From free-text to structured safety management: Introduction of a semi-automated classification method of railway hazard reports to elements on a bow-tie diagram," Saf. Sci., vol. 110, no. June 2017, pp. 11–19, 2018, doi: 10.1016/j.ssci.2018.03.011.

[14] A. J. P. Tixier, M. R. Hallowell, B. Rajagopalan, and D. Bowman, "Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports," Autom. Constr., vol. 62, no. 2016, pp. 45–56, 2016, doi: 10.1016/j.autcon.2015.11.001.

[15] P. Thompson, T. Yates, E. Inan, and S. Ananiadou, "Semantic annotation for improved safety in construction work," Lr. 2020 - 12th

Int. Conf. Lang. Resour. Eval. Conf. Proc., no. May, pp. 1990–1999, 2020.

[16] BSI, "BS EN 61882:2016 Hazard and operability studies (HAZOP studies) - Application guide," 2016.

[17] Recognai, "Rubrix," 2022. [Online]. Available: https://github.com/recognai/rubrix. [Accessed: 22-Feb-2022].

[18] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL HLT 2016 - Proc. Conf., pp. 260–270, 2016, doi: 10.18653/v1/n16-1030.

[19] M. Honnibal, "spaCy's NER model," 2021. [Online]. Available: https://spacy.io/universe/project/video-spacys-ner-model. [Accessed: 25-Oct-2021].

[20] N. Le Guillarme and W. Thuiller, "TaxoNERD: Deep neural models for the recognition of taxonomic entities in the ecological and evolutionary literature," Methods Ecol. Evol., vol. 13, no. 3, pp. 625–641, 2022, doi: 10.1111/2041-210X.13778.

[21] S. Magnolini, V. Piccioni, V. Balaraman, M. Guerini, and B. Magnini, "How to Use Gazetteers for Entity Recognition with Neural Models," Proc. 5th Work. Semant. Deep Learn., pp. 40–49, 2019.

[22] H. Baker, M. R. Hallowell, and A. J. P. Tixier, "AI-based prediction of independent construction safety outcomes from universal attributes," Autom. Constr., vol. 118, no. February, p. 103146, 2020, doi: 10.1016/j.autcon.2020.103146.

[23] K. Lata, M. Dave, and N. K.N., "Data Augmentation Using Generative Adversarial Network," SSRN Electron. J., pp. 1–14, 2019, doi: 10.2139/ssrn.3349576.