Towards Safer Operations: An Expert-involved Dataset of High-Pressure Gas Incidents for Preventing Future Failures

Shumpei Inoue[1], **Minh-Tien Nguyen**[1,2,*], **Hiroki Mizokuchi**[1], **Tuan-Anh D. Nguyen**[1], **Huu-Hiep Nguyen**[1], **Dung Tien Le**[1]

[1]Cinnamon AI, 10th floor, Geleximco building, 36 Hoang Cau, Dong Da, Hanoi, Vietnam.
{sinoue, ryan.nguyen, hmizokuchi, tadashi, hubert, nathan}@cinnamon.is

[2]Hung Yen University of Technology and Education, Hung Yen, Vietnam.
tiennm@utehy.edu.vn

Abstract

This paper introduces a new IncidentAI dataset for safety prevention. Different from prior corpora that usually contain a single task, our dataset comprises three tasks: named entity recognition, cause-effect extraction, and information retrieval. The dataset is annotated by domain experts who have at least six years of practical experience as high-pressure gas conservation managers. We validate the contribution of the dataset in the scenario of safety prevention. Preliminary results on the three tasks show that NLP techniques are beneficial for analyzing incident reports to prevent future failures. The dataset facilitates future research in NLP and incident management communities. The access to the dataset is also provided.[1]

1 Introduction

Daily activities usually face incidents that can significantly affect risk management. In specific industries such as manufacturing, an incident can make a significant consequence that not only reduces the reputation of companies but also breaks the product chain and costs a lot of money. It motivates the introduction of the safety-critical area where AI solutions have been proposed to prevent repeated failures from historical samples (Yampolskiy, 2019; McGregor, 2021; Durso et al., 2022; Nor et al., 2022; Chandra et al., 2023; Tikayat Ray et al., 2023; Andrade and Walsh, 2023).

There still exists a gap in the adoption of AI techniques for actual incident management scenarios due to the lack of high-quality annotated datasets. The main challenges arise from two main reasons. First, data annotation of incidents for AI-related tasks is a labor-expensive and time-consuming task that requires domain experts who have a deep understanding and excellent experience in their daily work. Second, the collection of historical incidents is also challenging due to its dependence on the policies of companies. We argue that the growth of the safety-critical area can be leveraged by introducing annotated incident datasets.

To fill the gap, this paper takes the high-pressure gas domain, a sector of the gas industry, as a case study. This is because gas and its products are the major industry in the energy market that play an influential role in the global economy (Mokhatab et al., 2018; Pellegrini et al., 2019). In addition, a gas incident may cost a lot of money with significant consequences. The detection and analysis of past incidents are crucial for improving safety prevention and avoiding future failures. Figure 1 shows the scenario of the detection and analysis. The description of an occurred incident is noted in an incident report. Then, important information (named entity and cause-effect extraction) from the incident report is extracted and stored in an incident database. In operation, given the description of an incident, the manager can search historical incidents for potential risk analyses. The system alerts the worker by showing historically relevant incidents and cause-effect information based on assigned tasks. The worker can use suggested information for failure analysis to avoid future incidents. In practice, given an incident report, workers, managers, or analyzers would like to know: (i) which aspects (entities) are relevant to the incident?, (ii) what is the cause and effect of the incident?, (iii) and which are historically relevant incidents of the current incident for risk and failure analyses?

To address the aforementioned questions, this paper introduces a new Japanese dataset that focuses on high-gas incidents and demonstrates the potential NLP applications in analyzing high-gas incident reports. To do that, we first work closely with business members and domain experts to identify three potential NLP tasks: named entity recognition (NER), cause-effect extraction (CE), and infor-

*Corresponding Author.

[1]The IncidentAI dataset is available at: https://github.com/Cinnamon/incident-ai-dataset

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 509–521
December 6-10, 2023

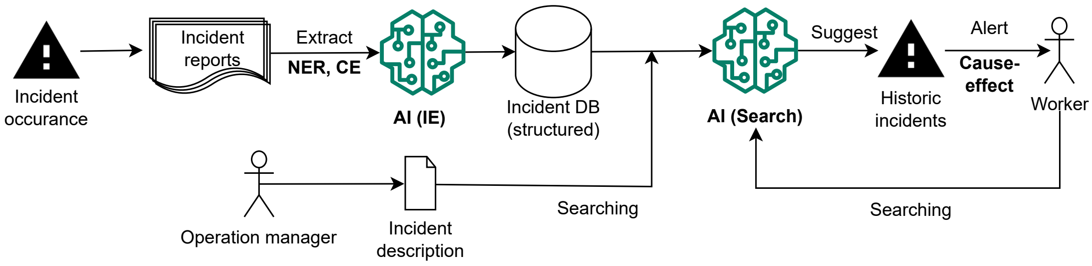


Figure 1: The scenario of IncidentAI in actual business cases. CE stands for cause-effect extraction.

mation retrieval (IR) based on actual scenarios. The NER task allows analyzers to extract fundamental units of an incident in the form of entities, e.g., the product or the process of the product. This information is used to visualize statistics concerning key entities from past incidents retrieved through IR steps. The CE task allows analyzers to extract the cause and effect of an incident. The IR task is typically used to examine historical incidents similar to the current one and to develop countermeasures to prevent the recurrence of such incidents. In business scenarios, information from the three tasks is vital for safety-critical and risk management. This paper makes three main contributions as follows.

- It introduces a new IncidentAI dataset that focuses on high-gas incidents for NER, CE, and IR. To the best of our knowledge, this is the first Japanese dataset that covers all three tasks in the context of high-gas incidents. It is annotated by domain experts to ensure a high-quality dataset that can assist in the efficient analysis of incident reports using AI models.
- It shows a scenario of IncidentAI in actual business cases. The scenario can serve as a reference for AI companies that are also interested in the analysis of incident reports.
- It benchmarks the results of AI models on NER, CE, and IR tasks that facilitate future studies in NLP and safety prevention areas.

2 Related Work

Incident databases There exist industry-specific incident databases in many industries (NTSB, 2017). The databases contain a wide range of incidents such as cyber-security vulnerabilities and exposures (Corporation, 2023), aviation reports that contain accident and incident information of air flights (Administration, 2023), or reports of the pharmaceutical industry, healthcare providers, and consumers (Food and Administration, 2023). Recently, an AIID database (AI Incident Database) was introduced (McGregor, 2021). It indexes more than 1,000 publicly available incident reports. The existing databases allow the storage of incident information, yet simple AI techniques (simple matching or classification) create a gap to analyze the incidents. We leverage safety operations by introducing a new dataset that includes three main tasks: NER, cause-effect extraction, and IR. It facilitates the adoption of AI to prevent future failures in the context of high-pressure gas incidents.

NLP techniques have been applied to analyze incident reports (McGregor, 2021; McGregor et al., 2022; Pittaras and McGregor, 2022; Hong et al., 2021; Nor et al., 2022; Macrae, 2022; Durso et al., 2022; Shrishak, 2023; Nor et al., 2022). The methods range from indexing for incident databases (McGregor, 2021; McGregor et al., 2022; Pittaras and McGregor, 2022) to deeper analyses using machine learning (Durso et al., 2022; Nor et al., 2022; Chandra et al., 2023). Recently, BERT (Devlin et al., 2019) has been adapted to aviation safety (Chandra et al., 2023; Tikayat Ray et al., 2023; Andrade and Walsh, 2023; Jing et al., 2023). The recent survey also shows the role of NLP in aviation safety (Yang and Huang, 2023). We share the direction of using NLP techniques in the analysis of incident reports. However, instead of focusing on single tasks, we are interested in three different tasks: NER, CE, and IR, that provide critical information for safety prevention. In addition, we provide a Japanese dataset to facilitate the creation of AI pipelines in a low-resource language.

3 The HPGIncident Dataset

3.1 Data Collection

The original dataset was collected from publicly available reports of high-gas incidents published in 2022 by the High-Pressure Gas Safety Institute of Japan.[2] The original data contains descriptions of incidents, types of incidents, dates of incidents, industries, etc. From the original 18,171 incident cases, 2,159 cases belonging to three industries: “general chemistry", “petrochemical", and “oil refining" were first extracted. These cases were used for the annotation of IR. Subsequently, we selected 970 cases from that 2,159 cases based on the most recent dates for both the annotation of NER and CE tasks. We used the description of incidents as the input for annotation shown in the next section.

3.2 The Annotation Process

The dataset was created by three Japanese domain experts, each with at least six years of practical experience as high-pressure gas conservation managers. These experts possess qualifications as high-pressure gas production safety managers, a national certification demonstrating a certain level of knowledge and experience necessary to ensure the safety of high-pressure gas manufacturing facilities.

The process was divided into two steps: the creation of the guideline and the annotation of the entire dataset. In the first step, we randomly selected 100 samples from 970 collected samples for NER and CE, and from 2,159 collected samples for IR. Our team collaborated closely with experts to establish criteria for consistent annotations, including identifying the information types (entities) and their definitions for NER and CE, and determining the attributes that characterize incidents for IR. These criteria formed the basis of our guidelines. This initial stage was iterative, conducted in several rounds until a certain agreement score was achieved among the experts. This process played a vital role in training the annotators, ensuring that they shared a uniform understanding of the guidelines. Once a high agreement score had been achieved, the remaining samples were apportioned into three segments, each corresponding to an annotator, who then proceeded to annotate their respective parts. Subsequently, 100 random samples were selected from one annotator's portion. The other two annotators were tasked with annotating these 100 samples. For each task, NER, CE, and IR, an inter-annotator agreement was computed using these 100 samples. Due to space constraints, please refer to Appendix A for a more detailed explanation of annotation.

NER annotation As mentioned, entities provide basic information about an incident. This represents the first tier in the incident report analysis. The initial step of NER annotation involves identifying the set of entities. The identification was carried out through meticulous coordination and several meetings with domain experts. Built on their insights and experiences, six critical entity types for incident analysis were established: **Products**, **Chemicals**, **Storage**, **Incidents**, **Processes**, and **Tests**. Table 1 shows the definition of entities.

Table 1: The definition of entities.

Entity	Definition
Products	A noun phrase that mentions various gases. Gaseous state at normal temperature and pressure. Do not tag items that are not general (things that do not appear even if you search the Web). Examples: mixed gas; flammable gas; refrigerant gas.
Chemicals	A noun phrase mentions chemical substances, reactants, and materials (other than gases) used in gas generation and process management. Items not included in the above Products. Examples: Benzene; Hydrocarbons.
Storage	General equipment where above Products and Chemicals come into contact. (i) Include equipment such as supports and insulators. (ii) Include expressions that indicate the entire plant or facility. (iii) Do not include expressions indicating parts such as entrances and exits if they are placed at the end of a word. Examples: tank; maturation furnace; refining tower; dehumidification tower.
Incidents	A phrase mentions incidents that resulted in or caused an accident, regardless of severity. It includes only incidents that actually occurred, and do not include situations that did not lead to an incident. Examples: seepage; leakage; fire; serious injury; death.
Processes	A phrase mentions handling of gas, and unit operations related to gas. Abnormal processes are included in incidents. Examples: filling; distillation.
Tests	A phrase mention inspection devices and inspection actions outside the production process line. Do not include inspection items such as XX concentration. Examples: inspection; visual inspection; leak test.

The annotation of NER uses the definition of entities in Table 1 and the rules mentioned in the Appendix A.1. We observed two important points. Firstly, the annotation was challenging even with domain experts. In the first round of guideline creation, the agreement score was so low. After several meetings, the agreement score was significantly improved. It provides strong evidence for the adaptation of the whole NER dataset. After annotating the whole NER dataset, the Fleiss' Kappa score of randomly cross-checking 100 other samples was 0.814, showing good agreement. Secondly, entities are nested. It comes from the nature of data, for example, a product can contain a chemical or a process can include storage. An annotated example of NER is shown in Figure 2 in the Appendix A.1. Table 2 summarizes the statistics of the NER dataset that follows the BIO format.

Cause-effect annotation Causes and effects provide critical information about a given incident for the analysis, in which causes contain information about the cause of an incident and effects mention the consequences of the causes. Similar to NER, we engaged in detailed discussions with domain experts to identify cause and effect types. We observed that the cause is quite easy to identify while the effect composes several types such as the leak-

[2] https://shorturl.at/BLWX6

Table 2: Statistics of the NER dataset.

Entity	Train	Test	All
Product	2,189	905	3,094
Chemical	1,440	576	2,016
Storage	5,881	2,251	8,132
Incident	5,274	2,045	7,319
Process	1,138	426	1,564
Test	1,615	572	2,187
#entities in total	17,537	6,775	24,312
#reports in total	700	270	970

age of gas, physical damages (explosion or fire), human injuries caused by the incident, or others (not related to leakage). After consulting, all the types of the effect were considered as the effect of an incident. Table 3 show CE's definition and the examples of cause and effect types.

Table 3: The guideline of CE annotation.

CE types	Definition
Event_Leak (EL)	Tag sentences in which gas leakage can be directly confirmed. However, automatic detection by equipment is not included due to the possibility of malfunction. Human detection is included. The definition of gas follows the NER Product. Example: Hydrogen and aniline leakage.
Damage_Property (DP)	Tag sentences that confirm physical damage to equipment or facilities caused by Event_Leak and Event_others. Physical damage includes burst pipes, destruction of heat exchangers, etc. Example: Container ruptures.
Damage_Human (DH)	Tag sentences that confirm Human casualties caused by Event_Leak and Event_others and Damage_Property. Human casualties include deaths, injuries, and physical illnesses. Example: One employee injured left thigh and left ear.
Event_others (EO)	Tag sentences containing accident events other than gas leakage. For example, explosions, fires, etc. Example: It is estimated that hydrogen, which has a low ignition energy, was ignited by static electricity.
Cause	Tag sentences that confirm the event causing Event_Leak and Event_others. Target not only direct causes but also indirect causes (e.g., Cause's Cause). In case of ignition or explosion, the three elements of combustion (combustibles, oxygen, and heat) shall be noted as a cause. Example: As a result of reduced tightening torque in some of the flange sections cooled by hydrogen

The annotation of causes and effects is on the span level. The annotation was done in two steps (follows Section 3.2), in which the first step was conducted in several rounds to create the annotation guideline to annotate the whole CE dataset. For annotation, the definition of causes and effects in Table 3 and the rules in Appendix A.2 were used. After creating the guideline with a high agreement score, the guideline was adopted to annotate the remaining 870 samples. Fleiss' Kappa score of randomly cross-checking 100 other samples was 0.764. Table 4 shows the information of the CE dataset. 467 samples have cause-effect pairs. Others only contain causes or effects. A sample of cause-effect annotation is shown in Figure 3, Appendix A.2.

IR annotation The objective of the IR annotation task is to realize a use case where users can

Table 4: Statistics of the CE dataset.

Information type	Train	Test	All
Cause	1,073	396	1,469
Effect	1,063	400	1,463
#samples in total	700	270	970
#samples with CE pairs	467	—	—

query incident descriptions to retrieve relevant past incidents. We found that the annotation of IR is challenging to measure the similarity of incidents by using single aspects, e.g., the description of incidents. Therefore, instead of directly assigning a relevance score to predefined levels like "Not Relevant," "Relevant," and "Highly Relevant," we first identified a set of key attributes to each incident report and then evaluated relevance on an attribute-by-attribute basis. The attributes allow us to reflect the nature of similarity among incidents.

We collaborated with domain experts to identify crucial attributes for determining how similar incident reports are. These specific attributes are shown in Table 5. Each incident description was annotated by assigning a relevant label to every identified attribute. The relevance score among incidents was measured by the degree of overlap in their labels. This strategy offers two advantages: (i) it provides a framework for a numeric evaluation of the relevance among incidents and (ii) it allows the flexible generation of relevance scores.

Table 5: The definitions of attributes and their labels.

Attribute	Label	#samples
Type of high pressure gas	(a) Flammable or Flame Retardant Gas	911
	(b) Toxic Gas	78
	(c) Satisfies a & b	563
	(d) Not applicable	607
Cause of incident	(a) Equipment Factor	940
	(b) Human Factor	598
	(c) External Factor	67
	(d) Other Factor	554
Incident result	(a) Leakage	1510
	(b) Fires and explosions	337
	(c) a & property damage	24
	(d) a & human casualties	88
	(e) b & property damage	47
	(f) b & human casualties	78
	(g) Property damage & human casualties	30
	(h) Others	45
Time span from cause to effect	(a) Sudden	364
	(b) Long term	1238
	(c) Unknown	557
Operational status of equipment	(a) Steady-state operation	900
	(b) Non-steady state operation	344
	(c) During maintenance	409
	(d) Other situations	506

In this study, we analyzed 2,159 high-pressure gas incident reports, detailed in Section 3.1. We employed a straightforward approach where each attribute's label overlap was scored as 1, and no overlap received a score of 0. When labels were jointly attributed through the use of 'and'—for instance, label (c) in the *Incident Result type*—the overlap score was increased to 1.5. The final relevance score was computed by summing these individual overlap scores. Sample incident descriptions and their corresponding relevance scores are presented in Figures 4 and 5, respectively. In this schema, the incident description itself is used as a query, and the goal of the retriever model is to identify reports with high relevant scores. To assess inter-annotator reliability, we evaluated the consensus across 100 incident reports annotated by three individuals. The resulting average Fleiss' Kappa score was 0.541, denoting a moderate level of agreement that is good enough for IR.

3.3 Quantitative Observation

This section shows the statistics of recent incident databases and corpora. The databases include CVE (Common Vulnerabilities and Exposures) (Corporation, 2023), FAA[3] (Federal Aviation Administration and National Aeronautics and Space Administration) (Administration, 2023), AIID[4] (McGregor, 2021), and EF (explosion and fire) (NIOSH, 2023). The corpora contain CFDC (high-level causes of flight delays and cancellations) (Miyamoto et al., 2022) and AIR (aviation incident reports) (Jiao et al., 2022). We also note that there are quite a lot of other incident databases and corpora but due to space limitation, we could not show them all.

Table 6: Statistics of incident databases and corpora. Manufac is Manufacturing and Lang is language.

Name	Samples	Label	Problem	Domain	Lang
CVE	141076	No	IR	Security	EN
FAA	—	No	IR	Aviation	EN
AIID	2842	No	IR	Mix	EN
EF	6430	No	IR	Fire	JA
CFDC	4195	No	Clustering	Aviation	EN
AIR	1775	Yes	Classification	Aviation	CN
Ours	**970**	**Yes**	**NER, CE, IR**	**Manufac**	**JA**

As observed, incident databases are usually designed for IR in diverse sectors without clear labels. Annotated corpora, e.g., AIR, are created for target problems with a smaller number of samples. Our dataset contains a quite small number of samples. However, it has the human annotation of three NLP tasks which are beneficial for the analysis of incident reports. In addition, a small number of samples is still helpful in business scenarios for training AI models by using transfer learning (Devlin et al., 2019; Nguyen et al., 2020, 2023).

[3]We could not know exactly the number of samples.

[4]https://incidentdatabase.ai

4 NLP Tasks and Methodology

Once the dataset has been created, NLP tasks were designed to establish the baselines of each task.

4.1 Nested Named Entity Recognition

The NER task was formulated as a sequence labeling problem (Ju et al., 2018; Rojas et al., 2022; Zhang et al., 2022; Yan et al., 2022). Strong nested NER models were selected as follows.

Layered nested NER This model stacks flat NER layers for nested NER (Ju et al., 2018). Each flat layer composes of a BiLSTM layer to capture the sequential context representation of an input sequence and a cascaded CRF layer for labeling.

Multiple BiLSTM-CRF This model uses multiple flat BiLSTM-CRF, one for each entity type (Rojas et al., 2022). The input layer combines character embeddings and token representation from Flair (Akbik et al., 2018) and BERT (Devlin et al., 2019). The combined representation is fed into BiLSTM layers to obtain long-contextual information. Sequence labeling is done with CRF.

BINDER is an optimized bi-encoder model for NER by using contrastive learning (Zhang et al., 2022). It formulates the NER task as a representation learning problem that maximizes the similarity between an entity mention and its type.

CNN-Nested-NER It is a simple but effective model for nested NER (Yan et al., 2022). It uses BERT (Devlin et al., 2019) for mapping input sequences into contextual vectors. The spatial relations among tokens are modeled by an additional CNN layer for prediction with a sigmoid layer.

Preliminary results Table 7 reports the performance of the baseline in terms of micro and macro F-scores. It shows that the CNN-Nested-NER

Table 7: Performance of NER models.

Model	Micro F-1	Macro F-1
Nested NER	78.87	75.63
Multiple BiLSTM-CRF	83.62	79.67
BINDER	86.96	84.02
CNN-Nested-NER	**87.53**	**84.54**

model is the best for recognizing nested incident entities. A possible reason comes from the use of partial relations among entities and contextual representation from BERT. The BINDER model follows with tiny margins. It shows the contribution of contrastive learning. Two nested NER models based on multiple layers do not show the efficiency. It suggests to improve representation learning. We did not use LLMs for NER due to nested entities.

4.2 Cause-Effect Extraction

The CE extraction task was formulated as a span extraction problem (Devlin et al., 2019; Nguyen and Nguyen, 2023). As shown in Table 3, each sample may contain one or more spans annotated as **Cause** or other types. For simplicity, EL, DP, DH, and EO spans were merged as **Effect** and cause spans were kept identically. For span-based extraction models, the question is *"cause"* or *"effect"* and the context is an incident report. This is because the definition of complete questions does not guarantee the semantic relationship between the questions and context documents (Mengge et al., 2020).

BERT-QA We followed BERT-QA (Devlin et al., 2019) to extract cause and effect spans. The question and the context were concatenated before being encoded by BERT. The contextual representations of tokens were put into a feed-forward network followed by a softmax layer. Each candidate span for the answer was extracted based on *start/end* probabilities predicted by the model.

FastQA Apart from BERT-QA, we also tested FastQA (Son et al., 2022). While BERT-QA extracts each cause or effect span independently, FastQA extracts cause and effect simultaneously as a pair. By embedding both *"cause"* and *"effect"* questions in a separate module, FastQA allows the model to encode cause and effect at the same time, which halves the complexity of the encoding.

Guided-QA Guided-QA (Nguyen and Nguyen, 2023) is an extension of BERT-QA that implicitly models the relationship between causes and effects in a sequence manner. It receives a cause question (*"cause"*) and predicts the corresponding cause span. Then the predicted cause span is used as a question for effect extraction. Compared to BERT-QA, Guided-QA takes into account an implicit relationship from effect for cause prediction.

LLMs We tested ChatGPT[5] and Vicuna-13b-4bit[6] to assess the capability of LLMs for CE in two settings: zero-shot and 1-shot. For zero-shot experiments, an incident report was appended to a pre-defined prompt such as *"Find the cause and effect in the following incident. Outputs should be in Japanese."* and feed them directly to LLMs. For 1-shot experiments, we used the completion format <instruction><example><input> as follows.

```
  Outputs should be in Japanese.
Text: <example incident>
Cause/effect: <example cause>
Text: <target incident>
Cause/effect:
```

Preliminary results Table 8 reports the results of CE models in terms of SQUAD F-1 (token match) (Devlin et al., 2019). With full 700 training samples, BERT-QA is competitive followed by ChatGPT 1-shot. It is understandable that BERT-QA was trained with 700 samples and easy for domain adaptation. ChatGPT and Vicuna may need more samples for working well with this CE task.

Table 8: Performance of cause-effect extraction models.

Model	Cause	Effect	Average
The train set of 700 samples			
BERT-QA	65.90	77.81	**71.85**
ChatGPT 1-shot	75.48	49.80	62.64
ChatGPT 0-shot	55.48	37.25	46.36
Vicuna 1-shot	32.00	31.55	31.77
Vicuna 0-shot	19.44	28.16	23.80
The train set of 467 samples			
BERT-QA	**76.34**	**80.98**	**78.66**
FastQA	71.25	79.39	75.32
Guided-QA	75.81	72.72	74.26

As mentioned in Table 4, 467 samples have cause-effect pairs. We did another experiment on this refined set. The F-scores show that span-based extraction models obtain improvement compared to the models trained on the original set. It shows that with more refined training samples, a simple BERT-QA model can achieve promising results. Note that FastQA and Guided-QA can only work with samples that include cause-effect pairs.

4.3 Information Retrieval

The IR task was formulated as a dense text retriever problem using bi-encoder (Zhao et al., 2022). A

[5]https://platform.openai.com/playground
[6]https://huggingface.co/elinas/vicuna-13b-4bit

deep neural network was used to convert incident reports and queries into dense vectors with their nearest neighbors searched in the database. We conduct the evaluation on the annotated IR dataset with several baselines as follows.

BERT-based bi-encoder (public) Bi-encoders are highly efficient retrieval models based on pre-trained transformer backbones (e.g., BERT). We utilized the popular sentence-BERT multilingual model[7] (Reimers and Gurevych, 2019) as the main baseline for our dense retrieval task.

BERT-based bi-encoder (finetuned) To better adapt the model to challenging technical terms and jargon in the incident reports, we further fine-tuned the aforementioned base encoder by using the unsupervised constrastive learning objective (Gao et al., 2021) on the collection of the incident corpus in Section 3.2. Detail of the fine-tuning process can be found in Appendix A.4.3.

Commercial embedding model (OpenAI) We also evaluated the recent commercial solution from OpenAI with the model name `text-embedding-ada-002`.[8] The model is available in form of an API, which we can use to create the embedding vector for a given document.

Preliminary results Table 9 presents the results of all IR models with `nDCG@k`, `mAP@k` and `Recall@k` as evaluation metrics with `k=20`. The evaluation dataset is described in Section 3.2.

Table 9: Performance of information retrieval models.

Model	nDCG@k	mAP@k	R@k
BERT-public	45.27	15.91	30.90
BERT-finetuned	**56.11**	21.72	**42.51**
OpenAI-emb	54.48	**22.25**	38.32

We can observe from the table that the fine-tuned BERT encoder produces significantly better performance than the default base model and achieves the best score on `Recall@k` and `nDCG@k`. The OpenAI embedding model closely follows the fine-tuned model, albeit not directly trained on similar data domains before. This shows the performance of the proprietary model from OpenAI is quite transferable and robust across different domains.

[7]https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2

[8]https://platform.openai.com/docs/guides/embeddings/what-are-embeddings

4.4 Output Observation

Nested NER Figure 6(a) shows the example of a success case, where the model correctly detects all spans of entities, including the nested one between Product and Test. Further observation shows that for the same entities that are close together, the model tends to incorrectly recognize these entities separately, as shown in figure 6(b). This type of error is more common for entities such as Incident and Process due to their complex nature and their lengths. For entities such as Chemical and Product, the common problem is misclassification of entities or incorrect recognition of other noun spans. The observation was done by using CNN-Nested-NER.

Cause-Effect Extraction As we observe the data, cause and effect spans usually appear in phrases that indicate incidents such as leak, insufficient tightening. Because causes and effects share such common patterns, it is harder for our models to make correct predictions. Figures 6(a) and (b) show an example of correct effect prediction and an example of incorrect cause prediction of BERT-QA finetuned on 467 samples (the best model).

Information Retrieval We analyze the success and failure cases of IR model BERT-finetuned. Most success retrieval cases such as Figure 7, with query document at the top most and followed retrieval results, typically mention several common subjects such as flame, gas leak, and substance name (ethylene). However, there are still a lot of failure cases of the model regarding the understanding of substance properties (toxic, flammable, etc) when retrieving similar cases containing different substances, and understanding the effect (e.g: leakage vs explosion) of the incident (Figure 8).

5 Conclusion

This paper introduces a new Japanese dataset for safety prevention by using AI models. The high-quality dataset is annotated by domain experts for NER, CE, and IR tasks. The dataset contributes to IncidentAI in two important points. First, it composes the three NLP tasks in a corpus that facilitates the development of AI pipelines for safety prevention in a low-resource language. Second, it benchmarks the results of the three tasks which are beneficial for the next studies of analyzing incident reports. Future work will adapt the dataset to create AI pipelines for preventing failures of IncidentAI.

Limitations

Although the newly created dataset of incidents is a very high-quality corpus that is composed of three NLP tasks: NER, cause-effect extraction (CE), and IR, the size of the dataset is quite small with 970 annotated samples for NER and CE. The number of annotated samples for IR is also small with 2,159 samples. While collecting raw data is quite easy, data annotation is time-consuming and labor-expensive with the involvement of domain experts. It explains the size of our dataset is quite limited. So, it requires more effort for data augmentation when using the dataset in some cases. For example, LLMs need thousands annotated samples for fine-tuning. In addition, the dataset is in Japanese. On the one hand, it facilitates the introduction of AI models for IncidentAI in a low-resource language. However, the dataset requires translation to more popular languages, e.g., English for wider use.

For evaluation, some models are quite straightforward because the purpose is to provide preliminary results of the dataset. We believe the performance of the three tasks can be still improved with stronger models, especially in the case of cause-effect extraction with BERT-QA and LLMs.

Ethics Statement

The dataset and models experimented in this work have no unethical applications or risky broader impacts. The dataset was crawled from publicly available reports of high-pressure gas incidents published in 2022 by the High Pressure Gas Safety Institute of Japan. Raw data contains information such as descriptions of incidents at high-pressure gas plants, types of incidents, dates of incidents, industries, ignition sources, etc. It does not include any confidential or personal information of workers or companies. Three annotators are domain experts who have at least six years of experience in the high-pressure gas incident domain. They knew the purpose of data creation and agreed to join the annotation process with their responsibilities. Their personal information is kept for data publication.

The models used for evaluation can be publicly accessed with GitHub links. There is no bias for the re-implementation that can affect the final results.

References

Federal Aviation Administration. 2023. Accident and incident data. https://www.faa.gov/data_research/accident_incident. Accessed: 2023-07-17.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649.

Sequoia R Andrade and Hannah S Walsh. 2023. Safeaerobert: Towards a safety-informed aerospace-specific language model. In *AIAA AVIATION 2023 Forum*, page 3437.

Chetan Chandra, Xiao Jing, Mayank V Bendarkar, Kshitij Sawant, Lidya Elias, Michelle Kirby, and Dimitri N Mavris. 2023. Aviation-bert: A preliminary aviation-specific natural language model. In *AIAA AVIATION 2023 Forum*, page 3436.

The MITRE Corporation. 2023. Cve - common vulnerabilities and exposures. https://cve.mitre.org/cve/search_cve_list.html. Accessed: 2023-07-17.

Jacob Devlin, Changm Ming-Wei, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Francis Durso, MS Raunak, Rick Kuhn, and Raghu Kacker. 2022. Analyzing failures in artificial intelligent learning systems (fails). In *2022 IEEE 29th Annual Software Technology Conference (STC)*, pages 7–8. IEEE.

United States Food and Drug Administration. 2023. Accident and incident data. https://www.fda.gov/drugs/questions-and-answers-fdas-adverse-event-reporting-system-faers/fda-adverse-event-reporting-system-faers-public-dashboard. Accessed: 2023-07-17.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. volume abs/2104.08821.

Matthew K Hong, Adam Fourney, Derek DeBellis, and Saleema Amershi. 2021. Planning for natural language failures with the ai playbook. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–11.

Yang Jiao, Jintao Dong, Jingru Han, and Huabo Sun. 2022. Classification and causes identification of chinese civil aviation incident reports. *Applied Sciences*, 12(21):10765.

Xiao Jing, Akul Chennakesavan, Chetan Chandra, Mayank V Bendarkar, Michelle Kirby, and Dimitri N Mavris. 2023. Bert for aviation text classification. In *AIAA AVIATION 2023 Forum*, page 3438.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of* *the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459.

Carl Macrae. 2022. Learning from the failure of autonomous and intelligent systems: Accidents, safety, and sociotechnical sources of risk. *Risk analysis*, 42(9):1999–2025.

Sean McGregor. 2021. Preventing repeated real world ai failures by cataloging incidents: The ai incident database. In *Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 17, pp. 15458-15463*.

Sean McGregor, Kevin Paeth, and Khoa Lam. 2022. Indexing ai risks with incidents, issues, and variants. In *arXiv preprint arXiv:2211.10384*.

Xue Mengge, Bowen Yu, Zhenyu Zhang, Tingwen Liu, Yue Zhang, and Bin Wang. 2020. Coarse-to-fine pre-training for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6345–6354.

Ayaka Miyamoto, Mayank V Bendarkar, and Dimitri N Mavris. 2022. Natural language processing of aviation safety reports to identify inefficient operational patterns. *Aerospace*, 9(8):450.

Saeid Mokhatab, William A Poe, and John Y Mak. 2018. *Handbook of natural gas transmission and processing: principles and practices*. Gulf professional publishing.

Huu-Hiep Nguyen and Minh-Tien Nguyen. 2023. Emotion-cause pair extraction as question answering. In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence - Volume 3: ICAART*, pages 988–995.

Minh-Tien Nguyen, Viet-Anh Phan, Le Thai Linh, Nguyen Hong Son, Le Tien Dung, Miku Hirano, and Hajime Hotta. 2020. Transfer learning for information extraction with limited data. In *Computational Linguistics: 16th International Conference of the Pacific Association for Computational Linguistics, PACLING 2019, Hanoi, Vietnam, October 11–13, 2019, Revised Selected Papers 16*, pages 469–482. Springer.

Minh-Tien Nguyen, Nguyen Hong Son, et al. 2023. Gain more with less: Extracting information from business documents with small data. *Expert Systems with Applications*, 215:119274.

NIOSH. 2023. National institute of occupational safety and health, japan. https://www.jniosh.johas.go.jp/publication/houkoku/houkoku_2020_05.html. Accessed: 2023-10-17.

Ahmad Kamal Mohd Nor, Srinivasa Rao Pedapati, Masdi Muhammad, and Víctor Leiva. 2022. Abnormality detection and failure prediction using explainable bayesian deep learning: Methodology and case study with industrial data. *Mathematics*, 10(4):554.

NTSB. 2017. Collision between a car operating with automated vehicle control systems and a tractor-semitrailer truck. *Technical Report No. NTSB/HAR-17/02*.

Laura A Pellegrini, Giorgia DE GUIDO, Stefano Lange, et al. 2019. Handbook of natural gas transmission and processing-principles and practices. In *Handbook of Natural Gas Transmission and Processing: Principles and Practices*, pages 669–739. Elsevier-Gulf Professional Publishing.

Nikiforos Pittaras and Sean McGregor. 2022. A taxonomic system for failure cause analysis of open source ai incidents. In *arXiv preprint arXiv:2211.07280*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. volume abs/1908.10084.

Matías Rojas, Felipe Bravo-Marquez, and Jocelyn Dunstan. 2022. Simple yet powerful: An overlooked architecture for nested named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2108–2117.

Kris Shrishak. 2023. How to deal with an ai near-miss: Look to the skies. *Bulletin of the Atomic Scientists*, 79(3):166–169.

Nguyen Hong Son, M Yu Hieu, Tuan-Anh D Nguyen, and Minh-Tien Nguyen. 2022. Jointly learning span extraction and sequence labeling for information extraction from business documents. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Archana Tikayat Ray, Bjorn F Cole, Olivia J Pinon Fischer, Ryan T White, and Dimitri N Mavris. 2023. aerobert-classifier: Classification of aerospace requirements using bert. *Aerospace*, 10(3):279.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Roman V Yampolskiy. 2019. Predicting future ai failures from historic examples. *foresight*, 21(1):138–152.

Hang Yan, Yu Sun, Xiaonan Li, and Xipeng Qiu. 2022. An embarrassingly easy but strong baseline for nested named entity recognition. *arXiv preprint arXiv:2208.04534*.

Chuyang Yang and Chenyu Huang. 2023. Natural language processing (nlp) in aviation safety: Systematic review of research and outlook into the future. *Aerospace*, 10(7):600.

Sheng Zhang, Hao Cheng, Jianfeng Gao, and Hoifung Poon. 2022. Optimizing bi-encoder for named entity recognition via contrastive learning. In *The Eleventh International Conference on Learning Representations*.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense text retrieval based on pretrained language models: A survey.

A Appendix

A.1 Annotation rules and examples of NER

The following rules must be observed when tagging entities from domain experts.

- Do not tag words indicating parts such as entrances, exits, and connections if they are at the end of a word. For example, tag "inlet pipe" as "inlet pipe", but for "heat exchanger outlet", only tag "heat exchanger".

- For tagging corresponding parts that indicate a range like "4-6", tag the entire "4-6".

- If the same word is used in different meanings, tag only the relevant entity.

- For words like "XX gas generation equipment," tag both Storage and Products (nested). For example, tag "XX gas generation equipment" as Storage and tag “XX gas”" as Products.

- If there is a modifier Process within Products, Chemicals, or Storage, do not tag Process. For example, do not tag "recycle" as Process in "recycle gas."

- Do not tag phrases containing particles like "of" in "XX of YY" (Tag only "XX" or "YY" separately).

- Tag abbreviations as well. However, do not tag specific abbreviations such as equipment or model numbers.

- Do not tag the state of individuals. For example: "Lack of perspective".

- Do not tag words within legal names or standards, such as the names of laws or regulations. For example: "High-Pressure Gas Safety Act," do not tag "High-Pressure Gas."

- Only tag Pc when it pertains to gas handling. For example: Do not tag "tightening further".

A.2 Annotation rules and examples of CE

The following rules must be strictly followed when tagging CE from domain experts.

- Include in one sentence to be tagged: Who, When, Where, What. Also include endings up to verb phrases (e.g. Tag up to "leaked").

- Do not include in a sentence to be tagged: (i) punctuation at the end of tagging ", and.", (ii) such as "due to...", and (iii) conjunctions at the beginning of a sentence (e.g. And," "And then," "And then," etc.).

- How to separate each tagging: (i) do not separate with "," but separate with ".", (ii) in the case of "broken and leaked," "broken" and "leaked" are two different tags, so separate them.

- No nesting.

- Do not tag the trigger for accident discovery (unless it is a causal factor in the accident). For example: "I noticed a strange odor,"

A.3 Annotation rules and examples of IR

Definitions of each attribute are shown in Table 10. The annotated example of IR is shown in Figure 4

A.4 Implementation

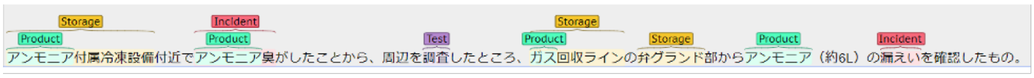
A.4.1 NER models

Except for the neural layered model for nested NER (Ju et al., 2018), whose word representation is based on the concatenation of character and word embeddings, other models use pre-trained BERT-based encoder, *TurkuNLP/wikibert-base-ja-cased*.[9] Other hyperparameters are set as follows.

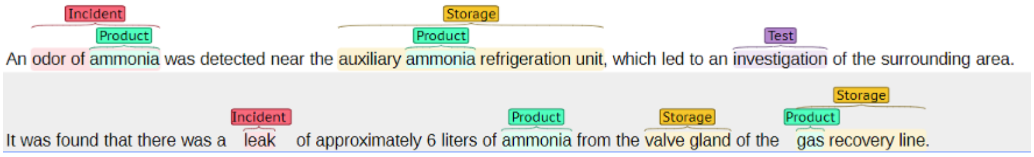
Layered nested NER The number of training epochs is set at 100, with the learning rate and decay rate of $1e-4$ and the batch size of 32. The dimension of word embedding and character embedding are 200 and 25, respectively.

Multiple BiLSTM-CRF The number of training epochs is 10 for each entity type.

[9] https://huggingface.co/TurkuNLP/wikibert-base-ja-cased

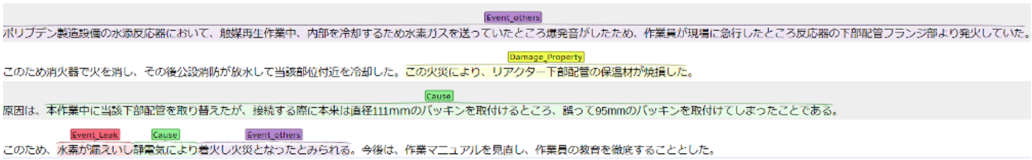


(a) Japanese (original).

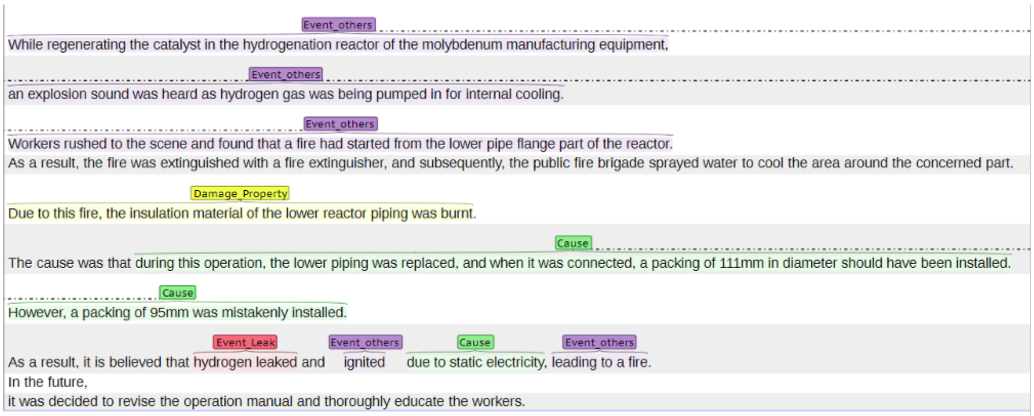


(b) English (translated version).

Figure 2: An annotated sample of NER.



(a) Japanese (original).



(b) English (translated version).

Figure 3: An annotated sample of CE.

Sample ID	Incident Desctiption
#1	Japanese (Original): 平成25年10月18日にフルオロカーボン検知器で調査したところ、付属冷凍機の吸入圧力計の導圧管継手部からフルオロカーボン22が漏えいしていることが確認されたため、すぐに増し締めを行い、漏えいが止まった。事業所では、9月18日からフルオロカーボンの液面変動に不審を感じており、発泡試験を行っていたが、当該漏えい箇所には行っていなかった。発生日以前から漏えいがあったものと考えられる。原因は、冷凍機の振動により、配管の締結部が緩んでフルオロカーボン22が漏えいしたと推定される。今後は、サポートを付けて配管の振れ止めを行うとともにフルオロカーボン検知器を導入する。 English (Translated): On October 18, 2013, an investigation with a fluorocarbon detector revealed that fluorocarbon 22 was leaking from the joint section of the pressure gauge suction pipe of the attached freezer. Immediate tightening was conducted, and the leak was stopped. The facility had been suspicious of fluctuations in the liquid level of the fluorocarbon since September 18 and had been conducting foam tests, but they had not been done at the leak location. It is believed that there had been a leak since before the date of occurrence. The cause is presumed to be that the pipe joint loosened due to vibrations of the freezer, resulting in the leakage of fluorocarbon 22. In the future, support will be added to prevent the piping from swinging, and a fluorocarbon detector will be introduced.
#2	Japanese (Original): 塩素出荷場でタンクローリに液化塩素を充てん中、流量計から約50㌔の液体塩素が噴出した。直ちに緊急遮断弁を操作し漏れを止めたが、風下にいた作業員8名がガスを吸い病院へ運ばれた。調べによると、同工場ではこれまで台ばかりを使って液体塩素の計量を行っていたが、事故当日、新品の流量計を取り付け充てん作業を行っていた。この流量計の材質はステンレス製で溶けていた。 English (Translated): At a chlorine shipping site, during the filling of a tank lorry with liquefied chlorine, approximately 50 kilograms of liquid chlorine spurted from a flow meter. The emergency shut-off valve was immediately operated to stop the leak, but eight workers downwind were exposed to the gas and transported to the hospital. Upon investigation, it was found that the factory had been using scales for measuring liquid chlorine until now, but on the day of the accident, they were filling with a newly installed flow meter. This flow meter was made of stainless steel and had melted.

Figure 4: The samples of incident descriptions for IR.

Table 10: The guideline of IR annotation.

Attribute	Definition
Type of high pressure gas	The high-pressure gas that caused the reported accident was classified from the perspective of danger in the event of an accident. Cases where the gas could not be identified were included under "d. Not applicable". The definition of flammable gas and toxic gas shall conform to the High Pressure Gas Safety Act in Japan.
Cause of incident	The events that caused or triggered the accident were classified. Equipment factors refer to those caused by initial defects in parts built into the equipment. Human factors refer to errors made in operation or judgment by people on site. External factors indicate those caused by events from outside the equipment, such as falling objects.
Incident result	The events that occurred as a result of the accident were classified. Physical and human damage were only considered if they occurred as secondary events, such as gas leaks or fires. Property damage: Accidents resulting in damage to equipment or facilities due to fire or explosion. Do not include damage to equipment or other items that caused the accident. Human casualties: Accidents resulting in health hazards to humans due to leakage, fire, or explosion
Time span from cause to effect	The classification was made based on the time from when the cause or trigger of the accident occurred until the accident event took place. Sudden: Accidents where the results are caused generally within a few minutes to several tens of minutes from the occurrence of the cause.
Operational status of equipment	The classification was made based on the operational status of the equipment at the time of the accident. Non-steady state operation refers to operating conditions that differ from normal operation, such as immediately after the equipment starts running or during test operation

Attribute	**Label for Sample #1**	**Label for sample #2**	**Overlap Score**
Type of High Pressure Gas	(c) Satisfies a & b	(c) Satisfies a & b	1.5
Cause of Incident	(a) Equipment Factor	(a) Human Factor	1
Incident Result	(a) Leakage	(d) a & Human Causalities	1
Time Span from Cause to Effect	(b) Long Term	(a) Sudden	0
Operational Status of Equipment	(d) Other Situations	(a) Steady-state Operation	0

Figure 5: The scoring method for the samples in Figure 4 is as follows: each single overlap is assigned a score of 1. If a label partially overlaps, as seen with the factor *"a"* under the *"Incident Result"* attribute, it still receives a score of 1. With instances where a label overlaps with two factors, such as with 'Type of High Pressure Gas', the score is 1.5. The final correlation score is the sum of each individual overlap score, totaling **3.5** in this case.

BINDER The number of training epochs is 10, with the learning rate of $3e-5$ and the batch size of 8 due to the heavy model. In addition, the model requires a text description written in Japanese for each entity type, which describes what the entity is and how it is labeled.

CNN-nested-NER The number of training epochs is 10, with the learning rate of $3e-5$ and the batch size of 8. The depth of CNN layers is 3, with a dimension of 120 for each.

A.4.2 Cause-effect extraction models

The BERT-QA models were implemented using BERT classes provided by Huggingface (Wolf et al., 2020). The model was trained in 5 epochs, with the learning rate of $5e-5$, and the batch size of 16. FastQA (Son et al., 2022) and Guided-QA (Nguyen and Nguyen, 2023) were trained using the source code from each paper. Again, FastQA and Guided-QA were trained in 5 epochs with the learning rate of $5e-5$ and the batch size of 16.

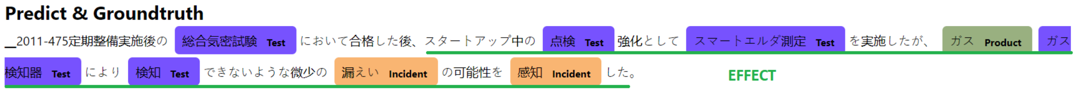
The pre-trained model *TurkuNLP/wikibert-base-ja-cased* was also used for all CE models.

A.4.3 IR models

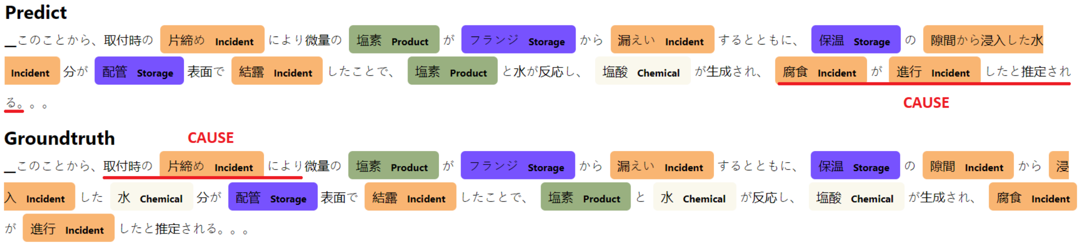
We fine-tuned the base model `distiluse-base-multilingual-cased-v2` from sentence-BERT [10] on the small subset of HPGIncident dataset described in Section 3.2 containing 3000 samples (not overlapped with the annotated IR dataset).

We utilize the unsupervised training objective from SimCSE (Gao et al., 2021), which takes an input sentence and predicts itself in a contrastive

[10]https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2



(a) Correct Nested NER and CE example. Consecutive tokens in green denote an effect.



(b) Incorrect Nested NER and CE example. Consecutive tokens in red denote a cause.

Figure 6: The figure shows a success and failure case for Nested NER and CE.

Successs	
Case ID	**Content**
1,566	ガソリン水添脱硫装置の熱交換器フランジ部から炎が出ているのをパトロール中のオペレーターが発見し通報した。直ちに装置を緊急停止し調査したところ、フランジ部のボルト締め付けが偏っておりガスが漏えいし着火したもの。 Translated An operator on patrol discovered flames coming from the heat exchanger flange of a gasoline hydrodesulfurization unit and reported the incident. When the equipment was immediately shut down and investigated, it was discovered that the bolts on the flange were not tightened unevenly, allowing gas to leak and ignite.
777	高圧ポリエチレンから残留エチレンを回収する低圧分離器からの回収エチレンが通るパイプに取り付けられた破裂板が破裂し着火、約13分後に鎮火した。（原因）点検の際、スパナを装置内に置き忘れ、このためバルブが詰ったこと等が原因である。 Translated A rupture disc attached to a pipe through which recovered ethylene from a low-pressure separator, which recovers residual ethylene from high-pressure polyethylene, ruptured and ignited, but the fire was extinguished approximately 13 minutes later. (Cause) The cause was that a spanner was left in the equipment during inspection, which caused the valve to become clogged.

Figure 7: A sample of success case for IR model.

Failure		
Type	**Case ID**	**Content**
Query	17,340	液化酸素貯槽は、加圧蒸発器にて一定圧力にコントロールしながら、ローリ車に液化酸素を充てんしている。9月24日14:30頃、加圧蒸発器の現地確認を実施していた際、冷気の流れが通常と異なることに気づき、詳細点検を実施したところ、15:00頃に発泡液にて漏れ箇所および微量漏れを確認した。そこで、当該加圧蒸発器をブロック、液抜きなどを実施し、遮断板を挿入した。 Translated The liquefied oxygen storage tank uses a pressurized evaporator to control the pressure at a constant level and fills the lorry with liquefied oxygen. At around 14:30 on September 24th, when we were conducting an on-site inspection of the pressurized evaporator, we noticed that the flow of cold air was different from normal.We conducted a detailed inspection and found that around 15:00, the foaming liquid Leak points and trace leaks were confirmed. Therefore, the pressurized evaporator was blocked, liquid was drained, and a blocking plate was inserted.
Retrieval Result	928	定期自主検査中に窒素ガスがなくなったので溶断用の酸素ボンベを使用して配管を昇圧し、逆止弁のテストを実施した後そのままの状態で緊急遮断弁の作動テスト実施中に爆発した。酸素ガスを使用したため爆発限界内のガスが発生した。 Translated During a periodic self-inspection, nitrogen gas ran out, so an oxygen cylinder was used to pressurize the piping, and a check valve test was conducted.The pipe was then left in the same state, and an explosion occurred during an emergency shutoff valve operation test. Because oxygen gas was used, gas within the explosive limit was generated.

Figure 8: A failure case of IR. The retrieved sample is different in the results (leakage vs explosion).

objective, with standard dropout used as noise. The model is fine-tuned using 3 epochs with the learning rate of $3e-5$. The encoder model uses mean pooling to aggregate contextual information from all tokens. We trained the model with the sequence length of 512 tokens.

A.5 Output observation

The output of nested NER and CE is shown in Figure 6 and that of IR is shown in Figure 7.