2024 34th International Conference on Collaborative Advances in Software and COmputiNg (CASCON) | 979-8-3315-0483-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/CASCON62161.2024.10837802

# Towards Enhancing Aviation Safety through Advanced Incident Analysis using Large Language Models

Vaishali Siddeshwar, Akramul Azim, Sanaa Alwidian, Masoud Makrehchi
Department of Electrical, Computer and Software Engineering
University of Ontario Institute of Technology
{first name.last name}@ontariotechu.ca

***Abstract*—Aerospace incident reports, crucial for identifying abnormal aircraft occurrences, are submitted to regulatory authorities like FAA or EASA. These reports, often in natural language, offer clarity and context. Existing literature proposes incident-analysis tools utilizing manual rules and NLP techniques like Part-of-Speech Tagging and dependency parsers to extract entities from logs. However, these approaches may miss nuances, leading to incomplete or inaccurate extractions and potential misinterpretations. This study aims to automate initial extraction using Large Language Models, extracting Aircraft Number/Model, contributing and human factors, primary root cause, and incident summaries. Automating this phase can expedite investigations and improve expert efficiency during review processes.**

***Index Terms*—Safety Incident Analysis, Text Summarization, Large Language Models (LLMs), Llama.**

## I. Introduction

Aerospace incident reports document abnormal occurrences involving aircraft, submitted by aviation's frontline personnel, including pilots, controllers, mechanics, flight attendants, and dispatchers to aviation authorities like FAA or EASA. These reports aid safety improvements and regulatory decisions. Companies also use them internally for operational enhancements. They are typically written in natural language for clarity and accessibility, allowing for detailed descriptions and contextual information inclusion.

Incident reports are often written in natural language for several reasons: 1) Clarity and Understandability: Natural language is more accessible and easier to understand for a wide audience. Using concise language helps ensure that the content of the incident report is comprehensible to stakeholders. 2) Provides context: Natural language provides the flexibility to include detailed descriptions and contextual information about the incident. Regulatory bodies currently analyze aviation incident reports in two phases: information extraction and investigation. In the information extraction phase, analysts manually extract incident aspects to understand its nature, including human factors and technical failures [1]. The investigation phase involves subject matter experts scrutinizing extracted information, conducting failure mode analysis, and evaluating technical system performance [1]. This study aims to automate the initial phase of aviation incident analysis, addressing the reliance on manual processes. By employing LLMs, our goal is to extract incident-related information such as aircraft details, contributing factors, primary root cause, and a brief summary. Automating this phase offers benefits including shortened investigation time and increased efficiency by assisting analysts during review, despite challenges in obtaining stakeholder approval due to the regulated nature of the aviation industry.

The study in [2] also investigates the application of LLMs for aviation safety, specifically using ChatGPT to extract incident reports. The authors employ BERT-based embeddings to compare the extracted incident summaries with those written by humans. However, we argue that real-world incident reports often contain sensitive information that cannot be shared with remote APIs like ChatGPT. Moreover, their method is validated solely on a subset of incident reports where human factors are the primary cause of the problem.

The main contributions of this project are mentioned below.

- We propose a technique to automate the initial assessment of the safety incident reports. In particular, the paper aims to provide a solution to extract the aircraft details, contributing factors including human factors related to the incident, along with briefly summarizing the detailed narrative of the incident.
- We evaluate the approach on a real-world publicly and available database from civil aviation domain- ASRS (Aviation Safety Reporting System).

The remainder of the paper is structured as follows: We start by illustrating a motivating example for the problem at hand in Section II. Following that, an outline of our methodology is detailed in Section IV, while the experimental setup is elaborated in Section V. In Section III, we delve into a comprehensive review of existing literature pertaining to the analysis of incident reports across diverse domains. The outcomes of our proposed approach are discussed in Section VI. Finally, Section VIII wraps up the study with a summary and concluding remarks.

## II. Motivational Example

To provide context for our research, we present an example incident report submitted to ASRS - Aviation Safety Reporting System [1] in Figure 1. This report includes a narrative detailing the event, alongside key information such as the ACN (Unique Incident ID), and the time and location of the occurrence. Notably, the majority of pertinent details are contained within the 'Narrative' section of the report. Furthermore, some incident reports feature multiple narratives submitted by various individuals involved in the incident. Additionally, certain narratives span over a thousand words, posing challenges for manual extraction of incident aspects. To initiate the investigation of the incident, the first step is to extract the following relevant information: details of the aircraft involved, the contributing factors that led to the event, human factors involved, primary problem, and finally, a brief synopsis that summarizes the incident in one or two sentences. Our approach accepts an incident report, similar to the one shown in Figure 1 and extracts incident-related information. Subsequently this information can be used by human experts to continue their investigation.

## III. Related Work

This section will present a general review of literature in the field of analysing incident reports published since in the last decade. In the past, automated incident analysis has been employed across various domains, including cybersecurity, network management, and operating system logs. Additionally, automation has been applied to analyze incident reports, facilitating the creation of traceability between functional requirements and fault logs. These studies depend on the utilization of manually crafted rules as well as Natural Language Processing (NLP) techniques, such as Part-of-Speech (POS) Tagging and dependency parsers to parse through the textual data, and extract significant entities from incident logs. While these techniques have been valuable in incident analysis, they do come with certain drawbacks. The most important drawback is its limited coverage. Hand-crafted rules may not cover all possible nuances present in incident logs, leading to incomplete or inaccurate extractions. This limitation can result in missed critical information or misinterpretation of incidents.

In [3], a methodology called 'Process mining' is proposed to analyze healthcare system logs for process improvement. Named Entity Recognition (NER) and Deep Learning (DL) techniques were used in [4] to assess Maritime risks by scrapping and analyzing various sources. Phishing, a cybersecurity term, is addressed in [5] with machine learning for email classification. In [6], a fault diagnosis knowledge graph for industrial robots is constructed using named entity recognition. Cloud industry leader Salesforce's Root Cause Analysis (RCA) is discussed in [7], proposing an Incident Causation Analysis (ICA) engine using neural NLP techniques. Classification of cybersecurity documents is tackled in [8]. Authors in [9] propose KESRI, an approach using machine learning-based key phrase extraction to mitigate fault propagation in software requirements. Securing drones is addressed in [10] with NLP-based named entity recognition on drone forensic images. [11] introduces an NLP-based event identification and management framework for network management systems. [12] applies NLP to honeypot logs for detecting attacker patterns. Log analysis of operating systems is discussed in [13], [14], utilizing BERT for entity extraction and sentiment analysis. The study by [2] also explores the use of LLMs in aviation safety, focusing on utilizing ChatGPT to extract incident reports. The authors use BERT-based embeddings to computing the cosine similarity score between the summaries generated by ChatGPT and human-written summaries.

LLMs offer significant advantages over traditional techniques in tasks like text summarization and key phrase extraction. They automatically learn patterns and relationships from data, reducing the need for manual rule creation. With superior performance on diverse datasets, LLMs produce high-quality summaries while generalizing well to unseen domains. Leveraging their deep language understanding, our study aims to assess LLMs' effectiveness in extracting event-related information from aviation incident logs using the LLM-Llama framework. This study contributes to advancements in aviation safety analysis and incident management by accurately identifying pertinent details from incident records.

## IV. Methodology

This section provides an overview of our proposed approach. The approach is illustrated in Figure 2 which constitutes of three modules: HTML Parser, Tokenization, and Executor.

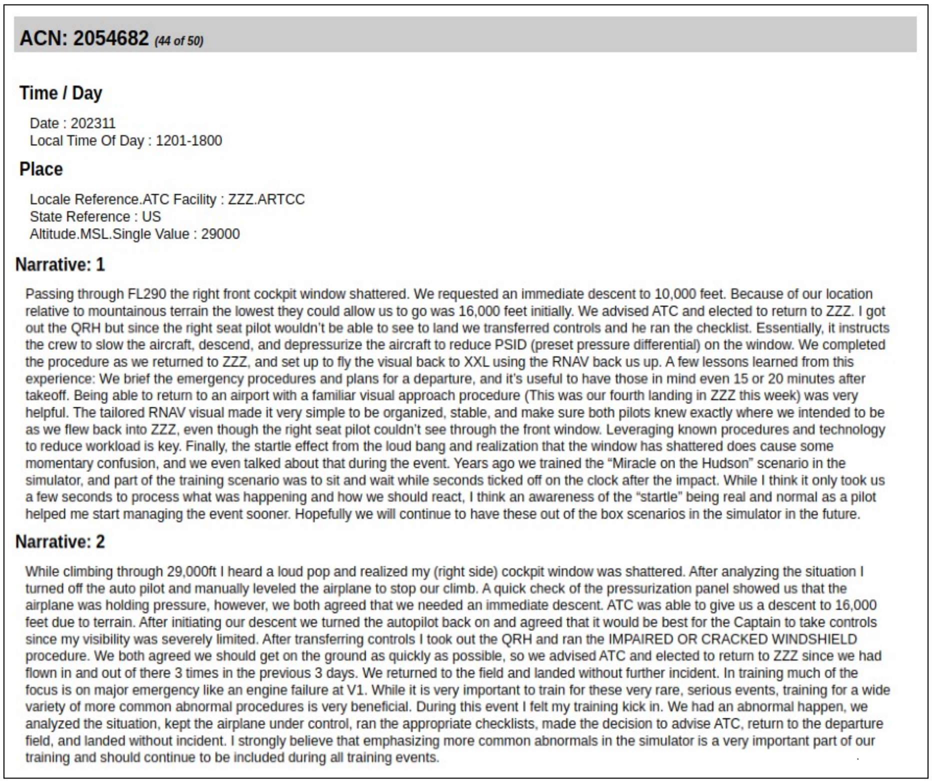
**HTML Parser:** This module interprets the structure of incident reports that are in HTML. It also extracts the synopsis portion of the report that will be summarized by the LLM.

**Tokenization** is a text preprocessing step. The Tokenizer module splits the textual incident summary into smaller units called 'tokens' that are the basic building blocks of language that the LLM operates on.

**Prompt Engineering** is a technique used to instruct a Large Language Model (LLM) to perform a task. In the context of LLMs like GPT, a 'LLM prompt' is constructed to guide a Large Language Model (LLM) to perform a specific task. In the context of automated-report analysis task, the prompt consists of a set of concise instructions along with the contents of the report.

**Execution:** There are predominantly two ways to execute of a LLM. The method used depends on the specific application for which the LLM is used for. The most well-known way of interacting with LLM is through ChatBots, like OpenAI's ChatGPT or Microsoft's Bing. While this approach reduces coding effort and expedites LLM-based application development, there is no assurance of chat history security. Furthermore, HTTP requests sent to OpenAI's ChatGPT server, hosted on OpenAI's servers, are vulnerable to potential hacking or interception by third parties, risking privacy breaches. This holds true even for systems that transmit HTTP requests to an API endpoint provided by vendors of LLMs. The method outlined in this paper does not involve sending HTTP requests over the internet nor does it require internet connectivity to analyze the reports. Instead, the pre-trained LLM is downloaded and utilized locally for inference purpose.

The LLM selected for this investigation is the 'Llama-2-7b-hf'. Llama 2 encompasses a series of finely-tuned generative text models. This study specifically opted for the LLM model



**ACN: 2054682** *(44 of 50)*

**Time / Day**

Date : 202311
Local Time Of Day : 1201-1800

**Place**

Locale Reference.ATC Facility : ZZZ.ARTCC
State Reference : US
Altitude.MSL.Single Value : 29000

**Narrative: 1**

Passing through FL290 the right front cockpit window shattered. We requested an immediate descent to 10,000 feet. Because of our location relative to mountainous terrain the lowest they could allow us to go was 16,000 feet initially. We advised ATC and elected to return to ZZZ. I got out the QRH but since the right seat pilot wouldn't be able to see to land we transferred controls and he ran the checklist. Essentially, it instructs the crew to slow the aircraft, descend, and depressurize the aircraft to reduce PSID (preset pressure differential) on the window. We completed the procedure as we returned to ZZZ, and set up to fly the visual back to XXL using the RNAV back us up. A few lessons learned from this experience: We brief the emergency procedures and plans for a departure, and it's useful to have those in mind even 15 or 20 minutes after takeoff. Being able to return to an airport with a familiar visual approach procedure (This was our fourth landing in ZZZ this week) was very helpful. The tailored RNAV visual made it very simple to be organized, stable, and make sure both pilots knew exactly where we intended to be as we flew back into ZZZ, even though the right seat pilot couldn't see through the front window. Leveraging known procedures and technology to reduce workload is key. Finally, the startle effect from the loud bang and realization that the window has shattered does cause some momentary confusion, and we even talked about that during the event. Years ago we trained the "Miracle on the Hudson" scenario in the simulator, and part of the training scenario was to sit and wait while seconds ticked off on the clock after the impact. While I think it only took us a few seconds to process what was happening and how we should react, I think an awareness of the "startle" being real and normal as a pilot helped me start managing the event sooner. Hopefully we will continue to have these out of the box scenarios in the simulator in the future.

**Narrative: 2**

While climbing through 29,000ft I heard a loud pop and realized my (right side) cockpit window was shattered. After analyzing the situation I turned off the auto pilot and manually leveled the airplane to stop our climb. A quick check of the pressurization panel showed us that the airplane was holding pressure, however, we both agreed that we needed an immediate descent. ATC was able to give us a descent to 16,000 feet due to terrain. After initiating our descent we turned the autopilot back on and agreed that it would be best for the Captain to take controls since my visibility was severely limited. After transferring controls I took out the QRH and ran the IMPAIRED OR CRACKED WINDSHIELD procedure. We both agreed we should get on the ground as quickly as possible, so we advised ATC and elected to return to ZZZ since we had flown in and out of there 3 times in the previous 3 days. We returned to the field and landed without further incident. In training much of the focus is on major emergency like an engine failure at V1. While it is very important to train for these very rare, serious events, training for a wide variety of more common abnormal procedures is very beneficial. During this event I felt my training kick in. We had an abnormal happen, we analyzed the situation, kept the airplane under control, ran the appropriate checklists, made the decision to advise ATC, return to the departure field, and landed without incident. I strongly believe that emphasizing more common abnormals in the simulator is a very important part of our training and should continue to be included during all training events.

Fig. 1. Example Incident Report

with 7 billion parameters, adapted to the Hugging Face Transformers format. Llama2-7B stands out as the most compact model capable of running on the majority of machines for inference tasks.

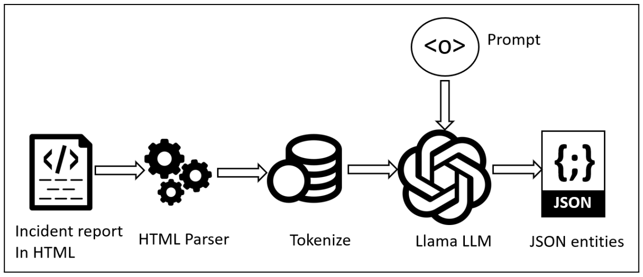


Fig. 2. Proposed Approach

## V. Experimental Setup

### A. About Data

This section outlines the structure of incident reports, each of which is distinguished by an 'ACN' serving as a unique identifier within the ASRS database. These reports encompass an analysis of Human Factors, aiming to elucidate the influence of human performance, cognition, and behavior on aviation incidents. Additionally, 'Contributing Factors' are identified within the report, pinpointing the circumstances that contributed to the incident's unfolding as observed by the reporter. Moreover, the report features a section labeled as 'Primary Problem,' signifying the primary cause underlying the incident. Furthermore, the report includes a 'Narrative' section containing a detailed description of the incident as provided by the reporter.

Table I highlights key elements of the incident report, although this list is not exhaustive.

### B. Dataset Preparation

The dataset to evaluate the proposed approach is collected from ASRS - Aviation Safety Reporting System [1]- a public repository of anonymized aviation incident reports. Incident reports contain a detailed narrative of the incident along with information related to the incident such as, the involved aircraft, contributing factors precipitating the incident, human factors implicated, primary issues encountered, and a summary of the incident which are generated by human experts. It must be noted that only the narrative portion of the report is passed to the LLM. The rest of the fields, excluding the narrative are used as a benchmark against which the performance of LLMs in generating comparable information can be evaluated, providing insights into the model's effectiveness in understanding and summarizing aviation incident data in comparison to domain experts' assessments.

### C. Setup

This section discusses the software and hardware prerequisites essential for executing the methodology elucidated in Section IV. Initially, as explained in the Methodology section, the pretrained Llama model 'Llama-2-7b-chat-hf' is

TABLE I

A list of components in the Aviation incident report, along with additional accompanying information. This list is not exhaustive.

| Name | Description |
|---|---|
| ASRS Record Number (ACN) | Unique identifier assigned to an incident record in the ASRS database |
| Human Factors | These refer to the discipline that examines the impact of human performance, cognition, and behavior on aviation incidents, with the aim of understanding and mitigating factors such as human error, fatigue, communication breakdowns, and inadequate training that contribute to accidents or near misses in the aviation industry.<br>Following categories are listed in [1]: Communication Breakdown, Confusion, Distraction, Fatigue, Human-Machine Interface, Situational Awareness, Time Pressure, Workload. |
| Contributing Factors | The circumstances that played a role in the incident's occurrence as identified by the human analyst.<br>Following categories are listed in [1]: Human Factors, Environment - Non-Weather Related, Procedure, and Airspace Structure are some examples. Each incident can have multiple contributing factors. |
| Primary Problem | The main cause that led to the incident as identified by the human expert.<br>Following categories are listed in [1]: Human Factors, Environment - Non Weather Related, Procedure, Airspace Structure. However, each incident can have only one primary problem that led to the incident. |
| Narrative | The description of the incident submitted by the reporter. This includes information related to the chain of events, how the problem arose, and various human factors such as perceptions, judgments, decisions. |
| Synopsis | The summary of the incident submitted by a human expert. |

downloaded from the HuggingFace[1] website for inference purposes.

The subsequent phase involves generating responses from the LLM. This necessitates the submission of a 'prompt' which serves as the directive for the LLM to generate a response. The prompt is pivotal in steering the model's comprehension and facilitating the production of relevant outputs. In this instance, a prompt along with the textual content of the report is utilized to instruct the LLM to extract specific information. To accomplish this, the prompt detailed in II is employed. This prompt is structured into three distinct sections. The initial section, labeled as 'Context,' sets the stage for understanding the broader context of the task at hand. Following this, the second portion of the prompt, termed as 'Task', deals with the specific objectives to be achieved by the LLM. In this scenario, the LLM is tasked with responding to four questions by extracting relevant information from the report's content. The initial question pertains to identifying the type of aircraft involved in the incident, while the subsequent three questions are presented as multiple-choice queries, each exploring different factors contributing to the event. These factors encompass primary factors, contributing factors, human factors, and personnel responsible. The LLM can select one or more answers for each of these questions. Lastly, the fourth question prompts the LLM to succinctly summarize the report's content.

In concluding section of the prompt, the 'Output' section instructs the LLM to exclusively generate a JSON response, devoid of any accompanying textual explanation.

## VI. Evaluation

This section presents the evaluation strategies employed to assess the quality of the responses generated by the LLM. Initially, a thorough discussion on the selection of evaluation metrics is provided, accompanied by an explanation of the rationale behind choosing a specific metric over others. Subsequently, the results of the evaluation conducted on the chosen dataset are presented, shedding light on the efficacy of the methodology in generating responses of high quality.

### A. Evaluation Criteria

This section presents the evaluation criteria that will be used to examine the efficacy of using LLM for information retrieval task. To measure the task of identifying the aircraft name correctly from the narrative provided, Recall is used. Recall is a metric commonly used in information retrieval and classification tasks, including natural language processing. It measures the ability of a system to retrieve all relevant items or instances from a dataset, without missing any. In this case, Recall metric assesses if the aircraft is correctly extracted from the report. Recall metric is also used to assess the quality of 'Primary Problem' extracted from the incident report. The formula used to calculate Recall is mentioned below, where $TP$ refers to True Positives and $FN$ refers to False Negatives:

$$Recall_{aircraft} = \frac{TP}{TP + FN}$$

$$Recall_{PrimaryProblem} = \frac{TP}{TP + FN}$$

The Recall metric explained in the equation above is computed for every incident report. However, the overall quality of the approach in extracting the aircraft involved, is calculated by computing the Mean Average Recall (mAR), which is the average of Recall metric per incident report. The formula used

[1] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

TABLE II
STRUCTURE OF PROMPT

| Component | Description |
|---|---|
| Context | You are a incident-report analyzer working under the 'Transportation Safety Board Regulations'." Your goal is to extract entities from aircraft-related incident reports. Using the text provided between triple back ticks, extract the following entities: |
| Task | 1. Extract the Flight Name from the given text.<br>2. Choose one or more primary factors leading to the event from the below list: Aircraft, Airport, Airspace Structure, ATC Equipment Nav Facility Buildings, Chart or Publication, Company Policy, Equipment Tooling, Environment – Non-Weather Related, Human Factors, Incorrect Not Installed Unavailable Part, Logbook Entry, Manuals, MEL, Procedure, Software and Automation, Staffing, Weather, Ambiguous.<br>3. Choose one or more contributing factors causing the incident from the below list: Aircraft, Airport, Airspace Structure, ATC Equip Nav Facility Buildings, Chart or Publication, Company Policy,Equipment Tooling, Environment – Non Weather Related, Human Factors, Incorrect Not Installed Unavailable Part, Logbook Entry, Manuals, MEL, Procedure (including Airspace Authorization), Software and Automation, Staffing, Weather.<br>4. Choose one or more personnel responsible for the event: Communication Breakdown, Confusion, Distraction, Fatigue, Human-Machine Interface, Physiological – Other, Situational Awareness, Time Pressure, Training/Qualification, Troubleshooting, Workload, Other Unknown<br>5. Summarize the text in three sentences. |
| Output | It is important you do not include verbose explanation and do not include the markdown syntax anywhere.. |

to calculate *mARecall* is mentioned below, where $Q$ indicates the number of reports in the dataset.

$$mAR_{aircraft} = \frac{\sum_{q=1}^{Q} Recall(q)}{Q}$$

To measure the Contributing Factors, and Human Factors Jaccard Score is used. Jaccard similarity coefficient, measures the similarity between two sets by comparing their intersection to their union. In this study, the Contributing Factors, and Human Factors are sets comprising of one or more factors related to the incident. Therefore, using Jaccard Score instead of Recall is pertinent. Jaccard Score for 2 sets, $A$ and $B$ is calculated as follows:

$$Jaccard = \frac{\mid A \cap B \mid}{\mid A \cup B \mid}$$

The Jaccard Score metric explained in the equation above is computed for every incident report. However, the robustness of the approach in extracting the aircraft involved, is calculated by computing the Mean Average Jaccard (mAJ), which is the average of Jaccard metric per incident report. The formula used to calculate *mAJaccard* is mentioned below:

$$maJaccard_{aircraft} = \frac{\sum_{q=1}^{Q} Jaccard(q)}{Q}$$

BertScore [15] is used for evaluating the summary of the incident. The name 'BertScore' is derived from BERT (Bidirectional Encoder Representations from Transformers), a popular deep learning model for natural language understanding, which forms the backbone of the metric. BertScore computes a similarity score for each token in the candidate sentence with each token in the reference sentence. However, instead of exact matches, it computes token similarity using contextual embeddings. BertScore computes the similarity between two texts by first encoding them into high-dimensional representations using pre-trained BERT models and then comparing these representations using cosine similarity.

## VII. DISCUSSION

In this section, we assess the effectiveness of our approach using a dataset comprising 50 incident records sourced from the ASRS website. To contextualize our evaluation, it is important to acknowledge the inherent complexity of incident reports. The plot displayed in Figure3 illustrates the distribution of word counts within the 'Narrative' section of these reports, which serves as the primary source from which information needs to be extracted. Notably, the word counts in the incident narratives span a considerable range, varying from approximately 200 words to around 1500 words. This variability underscores the diverse nature of incident descriptions, posing a challenge for information extraction methods to effectively process and analyze these narratives across different lengths and complexities.

Given this context, this section discusses the performance of the model on the five information extraction tasks, particularly, in extracting the aircraft, Primary problem, contributing factors, human factors, and the summary of the narrative. The plot shown in Fig.4 depicts the performance of the approach on the five tasks.

The graphical representation in Fig. 4 illustrates the effectiveness of the proposed method in extracting the 'Aircraft Name and Model' from the reports. The Recall scores for this extraction task range between 0.6 to 0.9, with an average Recall score of 0.799. Furthermore, for the extraction of Contributing Factors and Human-related factors, the approach attained scores of 0.811 and 0.797, respectively. Regarding the generation of summaries, the method achieved average Bert score of 0.8018 when compared to the ground-truth summaries. Additionally, it demonstrated an accuracy of 0.801 in retrieving the Primary Problem. The graph further illustrates the score ranges across the four retrieval tasks applied

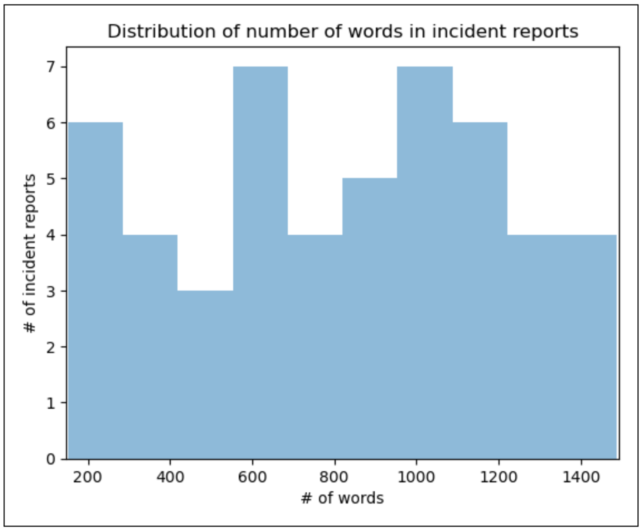


Fig. 3. Plot showing the Distribution of number of words in each incident report

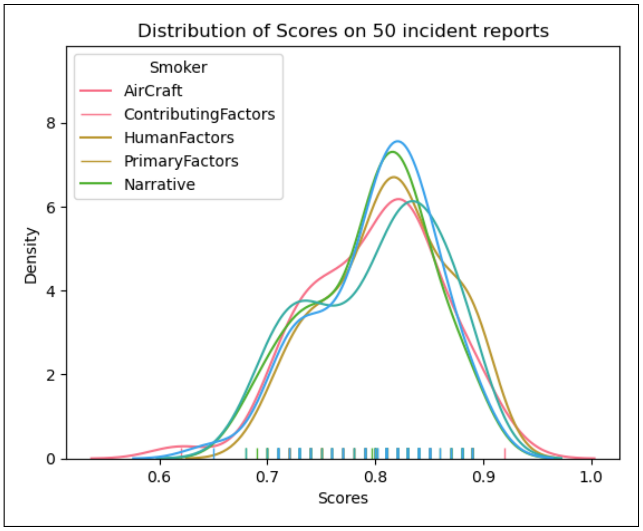


Fig. 4. Plot showing the Distribution of number of words in each incident report

to all processed reports. The distribution of scores appears close to normal, with minimal variance. Notably, the scores consistently fall within the range of 0.6 to 0.9, suggesting good performance of the proposed approach across all incident reports.

## VIII. Conclusion

Aerospace incident reports are formal documents detailing abnormal occurrences involving aircraft, submitted by aviation personnel to regulatory authorities like the FAA or EASA. These reports, often written in natural language, offer clarity, accessibility, and contextual details.

Several automated incident analysis tools have been proposed in the existing literature. However, these approaches analyze fault logs by employing manually crafted rules and NLP techniques, such as Part-of-Speech Tagging and dependency parsers to extract significant entities from incident logs. However, these techniques have limitations. Hand-crafted rules may overlook nuances in incident logs, leading to incomplete or inaccurate extractions and potentially missing critical information or misinterpreting incidents. This study aims to automate the initial extraction phase using Large Language Models to extract crucial incident-related information such as Aircraft Number/Model, contributing and human factors, primary root cause, and incident summaries. Automating this phase can expedite investigations and enhance expert efficiency during review processes.

## References

[1] ASRS. [Online]. Available: https://asrs.arc.nasa.gov/search/database.html

[2] A. Tikayat Ray, A. P. Bhat, R. T. White, V. M. Nguyen, O. J. Pinon Fischer, and D. N. Mavris, “Examining the potential of generative language models for aviation safety analysis: Case study and insights using the aviation safety reporting system (asrs),” *Aerospace*, vol. 10, no. 9, 2023. [Online]. Available: https://www.mdpi.com/2226-4310/10/9/770

[3] H. Yeo, E. Khorasani, V. Sheinin, I. Manotas, N. P. An Vo, O. Popescu, and P. Zerfos, “Natural language interface for process mining queries in healthcare,” in *2022 IEEE International Conference on Big Data (Big Data)*, 2022, pp. 4443–4452.

[4] V. Jidkov, R. Abielmona, A. Teske, and E. Petriu, “Enabling maritime risk assessment using natural language processing-based deep learning techniques,” in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 2469–2476.

[5] P. Saraswat and M. Singh Solanki, “Phishing detection in e-mails using machine learning,” in *2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, 2022, pp. 420–424.

[6] J. ZHOU, T. WANG, and J. DENG, “Corpus construction and entity recognition for the field of industrial robot fault diagnosis,” in *Proceedings of the 2021 13th International Conference on Machine Learning and Computing*, ser. ICMLC ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 410–416. [Online]. Available: https://doi.org/10.1145/3457682.3457745

[7] A. Saha and S. C. H. Hoi, “Mining root cause knowledge from cloud service incident investigations for aiops,” in *Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice*, ser. ICSE-SEIP ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 197–206. [Online]. Available: https://doi.org/10.1145/3510457.3513030

[8] M. Ishii, K. Mori, R. Kuwana, and S. Matsuura, “Multi-label classification of cybersecurity text with distant supervision,” in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, ser. ARES ’22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3538969.3543795

[9] M. Singh and G. Walia, “Automating key phrase extraction from fault logs to support post-inspection repair of software requirements,” in *14th Innovations in Software Engineering Conference (Formerly Known as India Software Engineering Conference)*, ser. ISEC 2021. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3452383.3452386

[10] S. Silalahi, T. Ahmad, and H. Studiawan, “Transformer-based named entity recognition on drone flight logs to support forensic investigation,” *IEEE Access*, vol. 11, pp. 3257–3274, 2023.

[11] A. Dwaraki, S. Kumary, and T. Wolf, “Automated event identification from system logs using natural language processing,” in *2020 International Conference on Computing, Networking and Communications (ICNC)*, 2020, pp. 209–215.

[12] M. Boffa, G. Milan, L. Vassio, I. Drago, M. Mellia, and Z. Ben Houidi, “Towards nlp-based processing of honeypot logs,” in *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2022, pp. 314–321.

[13] N. Abbasli and M. C. Ganiz, “Log and execution trace analytics system,” in *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, 2021, pp. 1–7.

[14] P. Dusane and G. Sujatha, “Logea: Log extraction and analysis tool to support forensic investigation of linux-based system,” in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2021, pp. 909–916.

[15] T. Zhang*, V. Kishore*, F. Wu*, K. Q. Weinberger, and Y. Artzi, “Bertscore: Evaluating text generation with bert,” in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr