Kaona: Deep Searching and Curating Data from Aviation Safety Reporting Systems

Carlos Paradis*
KBR, Inc., California, USA

Christopher Hong†
Oregon State University, Corvallis, OR 97331, USA

Bryan Matthews‡
KBR, Inc., California, USA

Misty D. Davies§ and Becky L. Hooey¶
NASA Ames Research Center, Moffett Field, CA, 94035, USA

Context: Several works in the literature have examined how safety narrative databases can be leveraged to share lessons learned. However, less attention has been given to augmenting existing processes for mining these safety reporting system databases.

Aim: In this work, we introduce Kaona: An interface that weaves machine learning in existing aviation safety database mining activities.

Method: We provide a use case of search, curation and newsletter writing to showcase how Kaona features build on existing processes and on its own to enhance information retrieval, curation and synthesis of narratives.

Results: We created two instances of Kaona internally for evaluation, one using publicly available NASA's ASRS narratives and another using publicly available C3RS narratives. Data ranged from 1998 to 2024.

Conclusion: Our tool provides a new way to explore safety narratives, serving to re-imagine how text databases can benefit of novel information retrieval mechanisms in the era of large language models.

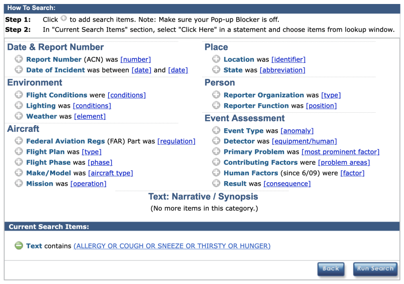
I. Introduction

There has been growing interest in utilizing natural language processing (NLP) algorithms in aviation safety [1]. This interest has extended to leveraging the decades of records publicly available in the Aviation Safety Reporting System (ASRS)* [2–5]. While related literature has given more emphasis in lessons learned from the narratives, our prior work [2] has focused on using NLP to support narrative search in the ASRS database. Specifically, we evaluated if the use of alternative search mechanisms to keyword search, such as the retrieval of related narratives even without matching keywords (e.g. [6]), could support narrative discovery.

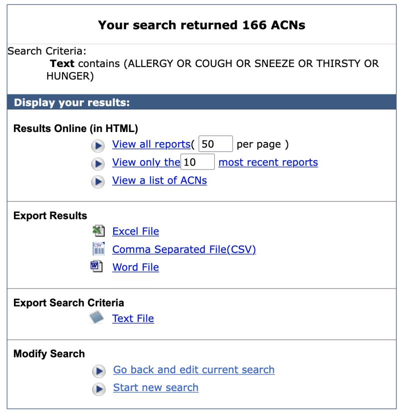
In this work, we further build on the idea of weaving machine learning into the task of mining aviation safety reporting system databases by introducing Kaona. Kaona is an interface which supports search, curation and synthesis of narratives using deep learning. We demonstrate in this work a use case of Kaona using publicly available data of NASA's Aviation Safety Reporting System, in particular the steps of search, curation and writing of newsletters. We emphasize that Kaona, although functional, is not integrated or in-use by NASA ASRS, but rather Kaona is an experiment in how machine learning could be used to support aviation safety reporting system data-mining activities.

Conceptually, Kaona was built with three users in mind: 1) Users interested in discovering and curating data for their own studies, 2) Machine learning researchers interested in easily integrating alternative search methods to an existing

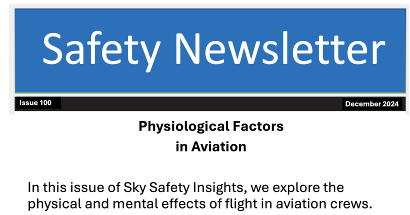
*Data Science Researcher.
†Intern, 1500 SW Jefferson Way
‡Senior Data Science Research Engineer
§Research Computer Engineer, Intelligent Systems Division, Mail Stop 269-1, AIAA Fellow.
¶Director, NASA Aviation Safety Reporting System, Mail Stop 262-4.
*`https://asrs.arc.nasa.gov/`



(a) Search Interface.

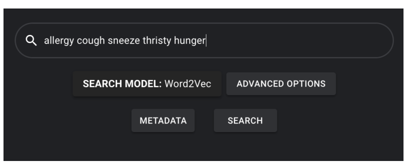


(b) Download Interface.

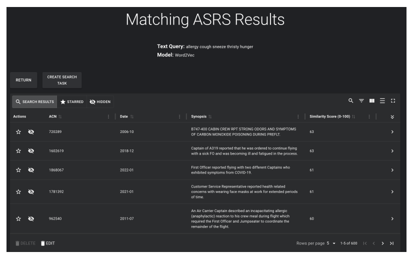


(c) Example Newsletter.

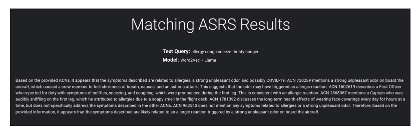
Fig. 1 ASRS Search, Download and Newsletter Use Case.



(a) Search Interface.



(b) Curate Interface.



(c) Newsletter and Synthesis.

Fig. 2 Kaona equivalent of Search, Download and Newsletter ASRS Use Case.

interface to obtain subject matter expert feedback, and 3) Large text dataset administrators, who could benefit from both users and researchers deriving value from their dataset. Kaona, in this sense, serves as a mediator to create a feedback loop between the three types of users, improving overall system wide safety.

II. Newsletter Search, Curation and Synthesis Use Case

To contextualize how deep learning can be used to extract lessons-learned from an aviation safety reporting database, we define a simplified three step process use case of search, curate, and synthesis to create a newsletter about physiological factors, as shown in figure 1. Kaona equivalent of the three interfaces is shown in figure 2. While the actual newsletter exists†. Our goal in this use case is to demonstrate how Kaona could be integrated to parts of the newsletter creation process, rather than exhaustively document the actual process performed in operation. In the following sub-sections, we expand on each one of the three use case steps in greater detail, introducing additional interfaces and features in Kaona.

III. Method

A. Narrative Search

In our use case, to begin preparing a newsletter, the relevant reports must be searched in ASRS. At the time of writing, ASRS offers both keyword search and metadata search capabilities, as shown in figure 1a. More complex queries using keywords can also be performed‡. Metadata search is possible due to ASRS analysts manually annotating labels in each

†https://asrs.arc.nasa.gov/publications/callback/cb_535.html
‡https://asrs.arc.nasa.gov/search/dbol/strategies.html

report using the ASRS Coding Taxonomy[§]. Users can use a combination of one or both search mechanisms to construct the search query.

Since our goal is to weave machine learning into the data mining process, Kaona also offers both keyword and metadata search. Users can in addition choose any different machine learning models available, by selecting advanced options in figure 2a. In this manner, ASRS users now can retrieve narratives using a combination of metadata and any other search algorithm available. Meanwhile, machine learning algorithms can seamlessly be added or removed for operational evaluation by subject matter experts. The choice of the algorithms displayed to the users is specified in a configuration file for each instance of Kaona running in a different dataset. This is because different models may have been pre-trained in the dataset or perform better overall. This further accounts for our third user type, which administrates the dataset, in this use case ASRS.

1. Text Embeddings

In the Kaona ASRS instance, we used the Word2Vec [7] model which we trained using all publicly available public ASRS data. The intuition behind offering this alternative method to keyword search, is that narratives which are semantically similar to the keywords used in the query can be retrieved, even if the actual keywords used are not present in the narratives. Specifically, text embedding can be used to translate words from the natural language into an embedding vector space that can represent relationships between words with similar meanings. Once text is translated into this space numerical methods can be used to quantify similarities between words or groups of words.

Word2Vec is an unsupervised neural network algorithm that learns the relationships between words in a multi-dimensional embedding space. The algorithm learns word associations between a target word and surrounding words on either side within a user defined window. Word2Vec does not require word similarity labels or require subject matter expert context to train the model. Acronyms are treated as any other word and the context of their meanings are inferred by the surrounding words.

During the training phase the model attempts to predict the surrounding words based on the target word. This training methodology is referred to as the skip-gram method. A single fully connected neural network architecture is used to map the entire input word corpus into a latent space dimension that is user defined. The final layer of the neural network attempts to map the latent space to the surrounding words with a single fully connected layer. After training, each word is directly mapped into the latent embedding space.

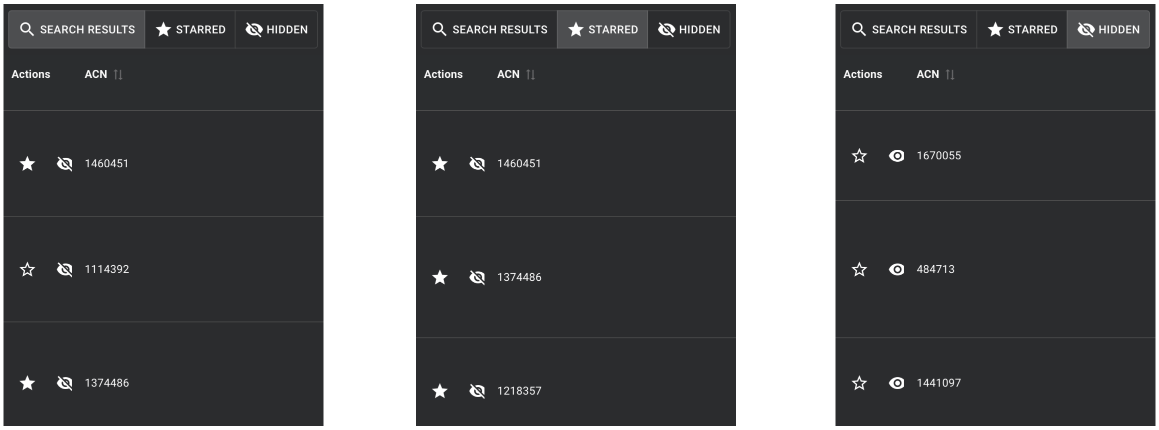
Since the algorithm is trained to predict surrounding words, the embedding space is organized in a way where words with similar meaning are collocated in this vector space. Using a distance metric such as cosine similarity, words that share similar meanings can be quantify numerically. An embedding dimension of 300 was used for this implementation. The decision was based on Mikolov's prior work [7].

After words are mapped to an embedding vector a document can be summarized as the average word embeddings across all the words in the document. However, some words that occur often across all documents can be uninformative and can skew a document's average embedding vector. To prevent these highly occurring uninformative terms from dominating the average word embedding of the document, a technique referred to as Term Frequency-Inverse Document Frequency (TF-IDF) [8] can be used to weigh each of the word's embedding vector. TF-IDF weighting de-emphasizes high frequency words that appear across many of the reports and assigns higher weights to terms that occur within a smaller set of reports. For each document, words are mapped to their Word2Vec embedding vector and weighted by TF-IDF before computing the average across the document. This yields a representative embedding vector for each document. The TF-IDF weighted average embedding vector can be performed offline for each document and stored in a database. This allows all documents to be quickly queried with a cosine similarity when a query is submitted. The query just needs to be mapped to the embedding space at the time of submission.

B. Narrative Curation

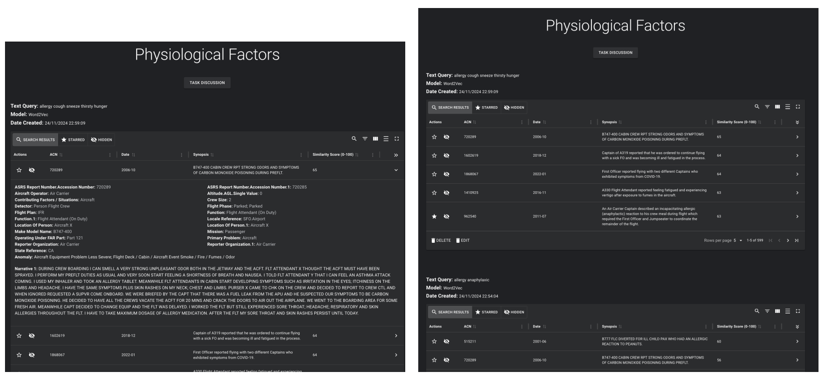
After selecting a search method, the next step in our use case is to curate the results retrieved. Figures 1b and 2b showcase how ASRS and Kaona differs on the display of results. In ASRS, the narratives retrieved can be downloaded in different formats, including MS Word, XLS, and PDF. In Kaona, the retrieved narratives are part of the interface. We emphasize a few features in this page, as they will be important in the subsequent sections to explain the explicit, implicit and replay curation features in Kaona.

[§]https://asrs.arc.nasa.gov/search/dbol/databasecoding.html



(a) Result Tab. **(b) Starred Tab.** **(c) Hidden Tab.**

Fig. 3 Kaona Narrative Display Tabs.



(a) Narrative and Metadata Inspection. **(b) Search Task.**

Fig. 4 Search Task and Narrative Inspection in Kaona.

1. Explicit Curation and Reusable Search Tasks

Figure 3 displays how explicit narrative curation is performed in Kaona. For each narrative retrieved, users can inspect, star or hide the narrative. Narratives are presented as rows in a table of synopses, as they are readily available for each narrative. The intuition for starring and hiding narratives is that, for some narratives, it may be easier to specify they are relevant to the user's search task, or conversely, that it is not.

When a search result narrative is starred for the first time, the user is prompted to a new dialogue, where the entire search, which includes the narrative starred, is saved to a search task. Figure 4 displays a narrative and metadata inspection, which can be done by expanding a narrative row, within an example search task for physiological factors. As we can see in figure 4b, a search task can be used to store multiple searches. This is consistent with ASRS recommended best results practices, which encourages an iterative process where the search strategy is continuously refined[¶].

We build on Kaona's built-in search task feature for narrative curation, and the previous search mechanisms to define the *reusable search task* feature. The intuition of this feature is shown in figure 5. When performing a new search in Kaona, the usual "user query" step is performed. The search can then be saved, as previously described. However, when subsequent searches are performed, users can manually specify they are part of an *existing* search task. Now, subsequent searches can leverage previously curated searches in the search task, to improve information retrieval by performing a *curated query*. For instance, when performing multiple searches within a search task, in our example the physiological factors newsletter, users are bound to observe some overlap among the narratives retrieved and that have already been marked as relevant or not relevant. In this case, Kaona will indicate a half-starred icon or hidden narrative, if it resurfaces. Subsequent machine learning searches can also now leverage knowledge of previous narratives marked

[¶]`https://asrs.arc.nasa.gov/search/dbol/strategies.html` as relevant or not within the context of the task to improve subsequent retrievals.

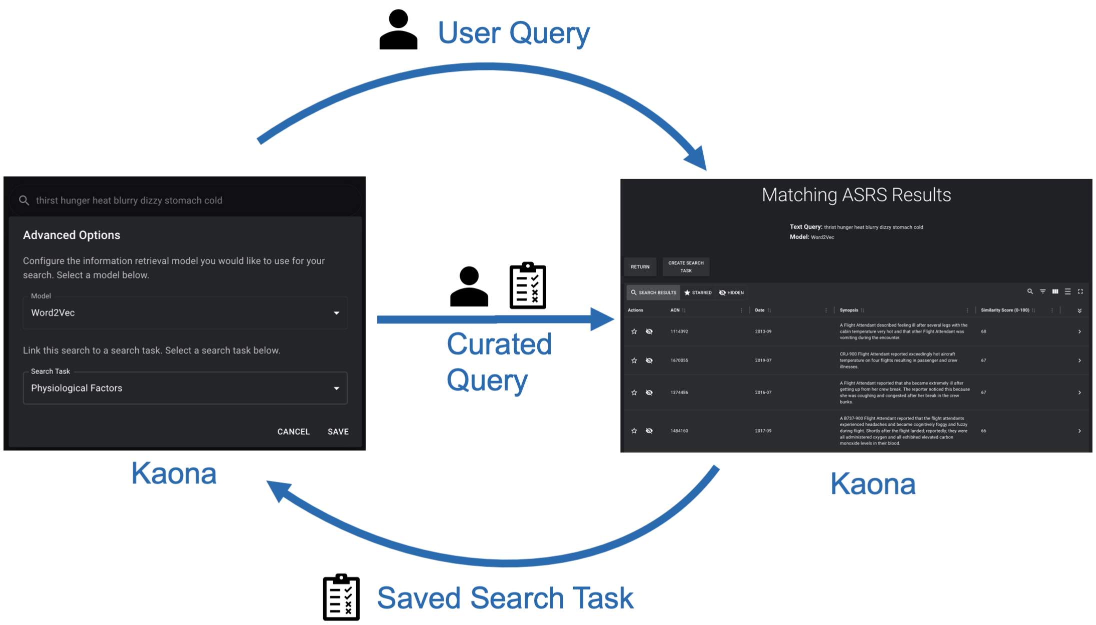


Fig. 5 Search and Curate Loop

2. Implicit Curation, Tracker API and Replay

It is reasonable to assume users will not star and hide every single narrative retrieved in Kaona. Indeed, when presented with irrelevant results, users may prefer to not save a query result altogether. That is, when the results of a search are least useful, Kaona is less likely to have any explicit curation data available to improve results in a subsequent search query. To alleviate this issue, we introduce the notion of implicit curation. Before using Kaona, users are provided with a dialogue explaining the interactions with the interface buttons are registered to the user. We can use this search-specific information to construct useful heuristics which capture poor search performance. Intuitively, if a user continuously browses through pages without inspecting or starring/saving a single narrative, this may suggest the results are not relevant even if any narrative is not manually hidden. Several search attempts without inspection or saving, can also be an indicator of poor performance. By combining explicit and implicit curation within a search task scope, more context-specific data can be used to improve the next user search attempt.

Kaona stores search specific information through a tracker event API. For example, when a user performs a search, a “search event” is logged, with additional information including the search keywords used, and the metadata fields chosen. If the user expands a given narrative in a search result, a “narrative inspection event” is logged, with the additional information of what narrative id was inspected. Altogether, Kaona Tracker API can be seen as a replay feature: Given any usage of the interface by a given user, it is possible to replay the set of actions through the event log, including the timestamps. The replay feature can, in addition of providing implicit curation context for subsequent searches, serve to evaluate search models offline, or better understand successful and unsuccessful search processes (i.e. as a substitute for a think-aloud protocol on a user-human computer interaction evaluation). While Kaona warns users of what information is logged before using the interface, we emphasize again that Kaona is not used in operation in ASRS, especially as some of these features introduce privacy concerns which require careful consideration.

C. Search Task Synthesis and Newsletter

Once users are satisfied with search and curating narratives for a specific task, the remaining step is to synthesize the needed information in a newsletter, as shown in figure 1c. In Kaona, we consider this as the synthesis step, as shown in figure 2c. The intuition is now on combining relevant information across selected narratives in a cohesive story. To leverage deep learning, we use retrieval-augmented generation (RAG) [9]. RAG is a method that combines large language models (LLMs) with additional context provided by an auxiliary search mechanism.

1. Retrieval-Augmented Generation

In the scope of our use case, consider a large language model chatbot that can converse with a user, but is unaware of ASRS narratives. Additionally, because re-training large language models every time a new narrative is added to ASRS would be time and cost prohibitive, even if the model were originally trained in ASRS, therefore enabling to ask questions about the dataset, it would soon become unable to answer questions about more recently added narratives. RAG allows us to provide said Chatbot with ASRS narratives "on the fly", depending on the question. In essence, Kaona users can leverage any of the search mechanisms of section III.A to ask a question to our Chatbot. This can be performed in the search screen shown in figure 2a, by selecting a search model that includes both a search and a chatbot pair from advanced options, resulting in the screen shown in figure 2c. Behind the scenes, the narratives shown in Figure 2b are used as context to the chatbot.

2. Curated Retrieval-Augmented Generation

In addition to the conventional path used by RAG, we build on Kaona's Search and Curate Loop of figure 5 to introduce an alternative path to converse to a Chatbot. Specifically, users can directly select from a search task to initiate a "task discussion", as shown below the "physiological factors" search task title in figure 4. Following this approach, the Chatbot can then use the curated narratives to discuss with the user about the search task, in this case the newsletter. Intuitively, after performing the search and curation steps, the user can leverage large language models to initiate discussion about the narratives to aid in the construction of the newsletter or request additional suggestions to performing additional searches based on the already identified narratives at any time. The two synthesis processes, which extend the method insofar presented, is summarized in figure 6.

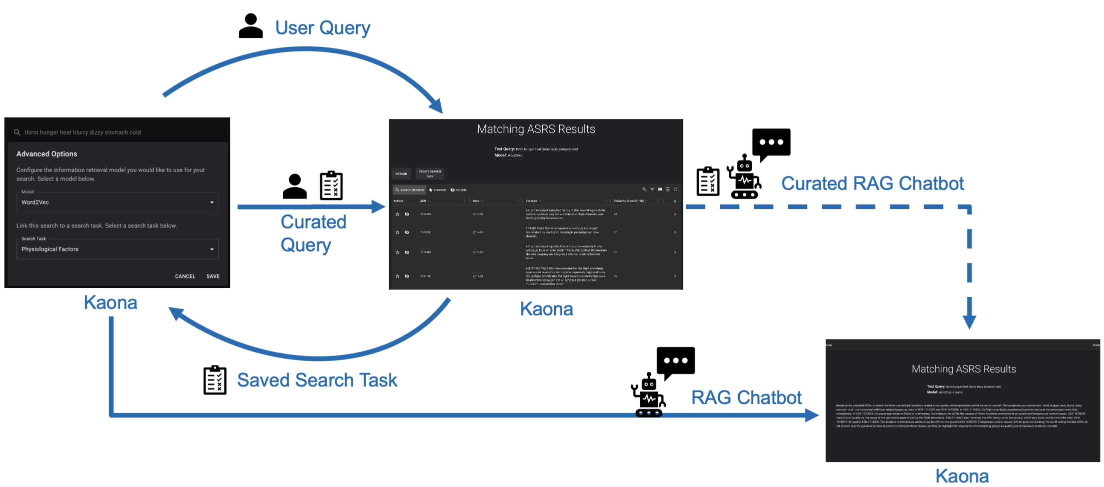


Fig. 6 Search, Curation and Synthesis using RAG.

D. Other Datasets

While we initially built Kaona using NASA's ASRS as a use case, we also performed a second use case using data from the NASA Confidential Close Call Reporting System deployed in C^3RS[‖] which is a similar reporting program for the Rail Industry. In its most simple deployment version, Kaona requires only a .csv table containing textual data which is uniquely identified to provide all the capabilities described in this work. If metadata are available, such as ASRS, C^3RS, or other related data source such as the National Transportation Safety Board (NTSB)[**], then Kaona can also be deployed making the dataset searchable. Therefore, Kaona can virtually be applicable to any text dataset, minimally

[‖] `https://c3rs.arc.nasa.gov/`

[**] `https://www.ntsb.gov/Pages/home.aspx`

formatted as a .csv table, making it a lightweight and generalizable tool for any text dataset. Untrained search models can be provided as a starting point, and later empirically evaluated and configured on a per dataset per Kaona instance.

IV. Conclusion and Future Work

In this work, we introduce Kaona, an interface that weaves machine learning into existing aviation safety reporting system data mining processes and builds on them to provide new mechanisms to search, curate and synthesize safety data to assist in extracting and sharing lessons learned. Kaona replicates and extends existing interfaces, when possible, to decrease the users learning curve. Kaona features builds on one another, to provide novel mechanisms to explore large collections of text.

In future work, we plan to conduct empirical evaluations on the various steps Kaona introduces machine learning, which will allow us to contrast various performance measures often reported in the literature, versus user perception in practice. We also believe Kaona's flexibility to be reused on any text dataset and tracker API to offer opportunities in better understanding search processes beyond the boundaries of a single system and community surrounding it. Finally, we plan on extending Kaona extending on prior research in our group, including the detection of emerging safety threats [10], and metadata networks building on ASRS taxonomy.

Acknowledgments

This research was conducted at NASA Ames Research Center. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government.

References

[1] Yang, C., and Huang, C., "Natural Language Processing (NLP) in Aviation Safety: Systematic Review of Research and Outlook into the Future," Aerospace, Vol. 10, No. 7, 2023. https://doi.org/10.3390/aerospace10070600, URL https://www.mdpi.com/2226-4310/10/7/600.

[2] Paradis, C. V., Davies, M., and Hooey, B., Assessing the use of UAS-related terms in ASRS using Seed Topic Modeling, 2023. https://doi.org/10.2514/6.2023-1127, URL https://arc.aiaa.org/doi/abs/10.2514/6.2023-1127.

[3] Matthews, B., Barshi, I., and Feldman, J., "An Approach to Identifying Aspects of Positive Pilot Behavior within the Aviation Safety Reporting System," 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), 2023, pp. 1–6. https://doi.org/10.1109/DASC58513.2023.10311185.

[4] Mbaye, S., Walsh, H. S., Davies, M., Infeld, S. I., and Jones, G., From BERTopic to SysML: Informing Model-Based Failure Analysis with Natural Language Processing for Complex Aerospace Systems, 2024. https://doi.org/10.2514/6.2024-2700, URL https://arc.aiaa.org/doi/abs/10.2514/6.2024-2700.

[5] Andrade, S. R., and Walsh, H. S., SafeAeroBERT: Towards a Safety-Informed Aerospace-Specific Language Model, 2023. https://doi.org/10.2514/6.2023-3437, URL https://arc.aiaa.org/doi/abs/10.2514/6.2023-3437.

[6] Gallagher, R. J., Reing, K., Kale, D., and Ver Steeg, G., "Anchored Correlation Explanation: Topic Modeling with Minimal Domain Knowledge," Transactions of the Association for Computational Linguistics, Vol. 5, 2017, pp. 529–542. https://doi.org/10.1162/tacl_a_00078, URL https://aclanthology.org/Q17-1037.

[7] Mikolov, T., Chen, K., Corrado, G., and Dean, J., "Efficient Estimation of Word Representations in Vector Space," 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings, edited by Y. Bengio and Y. LeCun, 2013. URL http://arxiv.org/abs/1301.3781.

[8] Leskovec, J., Rajaraman, A., and Ullman, J. D., Mining of Massive Datasets, 2nd ed., Cambridge University Press, 2014. URL http://mmds.org.

[9] Dubey, A., and et al., "The Llama 3 Herd of Models," , 2024. URL https://arxiv.org/abs/2407.21783.

[10] Paradis, C., Kazman, R., Davies, M. D., and Hooey, B. L., "Identifying Emerging Safety Threats through Topic Modeling in the Aviation Safety Reporting System: A COVID-19 Study," 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), 2021, pp. 1–8. https://doi.org/10.1109/DASC52595.2021.9594369.