A Novel Operations-Based Application of Natural Language Processing to Enhance Aircraft System Troubleshooting

Jamie Asbach[1], Daniel Wade[2]

[1,2]*Lockheed Martin, Bethesda, MD, 20817, USA*

jamie.asbach@lmco.com
daniel.r.wade@lmco.com

ABSTRACT

Troubleshooting an aircraft system is difficult. With flights often logging hundreds, or even thousands, of codes, the task of isolating the root cause of an issue is a complex undertaking. By leveraging Natural Language Processing techniques such as Word2Vec, artificial intelligence can be used to extract patterns from the context of these faults. Treating the fault codes issued by the on-board system in an aircraft as the "words" which make up a body of text, a model can be trained to understand the patterns of this language in a similar approach to how natural language is processed by computers to discretize the order and structure of human language. By assessing the cosine similarity of vectorized fault sequences used to train the model, faults occurring in similar sequences can be extracted, resulting in improved troubleshooting. The result of this effort is a tool to aid maintainers in isolating faults by quantifying the relations between the different codes and analyzing the patterns in which they occur. The benefits of such a tool include significant reduction in time and cost in aircraft maintenance by avoiding unnecessary exploratory maintenance. This is an ongoing project by Lockheed Martin and while the details of the concept and approach can be shared in an academic setting, the specifics of the validation data used and detailed model development cannot.

1. KEYWORDS

Large Language Model (LLM), Artificial Intelligence (AI), aircraft maintenance, Natural Language Processing (NLP), fault isolation, sympathetic faults, troubleshooting

2. BACKGROUND

A major cost driver in the life of an aircraft is the cost of maintenance. Parts, labor, hangar space, and many other factors contribute to a significant expense in conducting various repairs, inspections, and refurbishments (Heisey). Although many of these costs are unavoidable, there is a significant interest in the aerospace community to drive down cost of maintenance by using more advanced analytics to avoid unnecessary maintenance.

Traditionally, aircraft maintenance can be separated into two categories: scheduled and unscheduled. Scheduled maintenance includes replacement of life-limited components, inspections, and many other tasks determined to be necessary to ensure an acceptable factor of safety for flight. These requirements are typically defined in a maintainers manual and include a list of tasks and part replacements that are tracked against usage metrics. Typical metrics that initiate these repairs are flight hours, number of landings, or a calendar date. Although there are efforts underway to optimize the frequency of these such tasks (such as Condition Based Maintenance, CBM+), they are rigid and do not leave significant room for improvement. For the scope of this paper, these tasks will not be the focus of the maintenance improvement effort.

The second traditional category of aircraft maintenance is unscheduled events. This refers to reactive maintenance to address a failure that happens unexpectedly. In operation, when a failure occurs, the fault system aboard an aircraft reports a fault code. While these automated reports can have corrective action suggestions, the situation is typically reviewed by a maintainer once the aircraft is grounded to ensure proper action. If action is needed, a maintenance task can be initiated and performed to address this failure. Upon completed repair, the aircraft can be returned for flight usage.

2.1. Problem Overview

The focus of this paper will be the troubleshooting process of assessing unscheduled failures. One of the main challenges in isolating a fault on an aircraft system is the sheer number of fault codes reported. At various points throughout startup, taxi, and flight, both fault and status codes are recorded. These codes are the language of the aircraft, and they communicate valuable information about the status of the system. The function of these codes can differ greatly. Some codes indicate nominal status, while others offer insight to major mechanical problems. This is similar to the dashboard of a car, which is a combination of status (engine RPMs, engine temperature, gas level) and issues that require the driver's immediate attention (check engine light, seat belt unbuckled). With thousands of codes in the library of possible aircraft fault codes, noise becomes a significant problem when trying to isolate a fault. It quickly becomes difficult to read and understand the reported codes and extract required action.

For many aircraft systems in the defense industry, it is common for a single flight to incur thousands of these codes. It is also common for a singular root failure to cause the issuance of multiple fault codes, which introduces the problem of sympathetic faults. Sympathetic faults are fault codes that occur as a downstream failure to a parent system. An illustration of sympathetic faults is a domestic power outage. When experiencing a power outage, all lights in a home would go out. Although the first thought may be that the bulbs or the lamps themselves may have broken, this is merely a symptom of the actual fault. Replacing the lamp or its bulb would not solve the issue and would be an inefficient use of time and resources. A helpful tool in this situation might be a model that has studied past issues and recognizes the connection between power outages and the lights going out. This simple example shows the importance of learning the patterns of past failures to inform our future action and reduce unnecessary maintenance. Although in this example this pattern would be very easy for a human to recognize and would not require a model, more complex failure modes, such as those found in aircraft systems, are often much less obvious.

To compound the importance of this problem, it is common in deployed environments for aircraft maintainers to be young and inexperienced. This lack of expertise makes trouble shooting these systems more difficult. If multiple faults were indicated, each with a different prescribed corrective action, an undesired response would be for the maintainers to progress through list, performing maintenance tasks until the issue is resolved. Because of the significant cost and time associated with exploratory maintenance, any information that can be offered to the maintainer to aid in fault isolation is extremely valuable.

To address this, an effort is being made to look at the historical patterns of faults and establish relationships between them. If certain fault codes commonly occur in groups, this often indicates a common root cause. Thus by quantifying these relationships, with sufficient historical data, a distinction can be made between root cause faults and sympathetic faults.

2.2. Literature Review

It is difficult to fully evaluate the existing solutions to this problem, as many approaches to these types of problems (including the one described in this paper), are propriety information, kept as industry trade secrets. However, there are some approaches documented through publication that warrant mentioning.

Ezhilarasu et al. describe an approach that considers the interactions between sub-systems and their effect on the overall health of an aerospace system (Ezhilarasu, 2019). By using AI to understand the connections between these sub-systems, an IVHM (Integrated Vehicle Health Management) system can be established to monitor the health of various components. This groundwork paves a way toward CBM (Condition Based Maintenance), which eliminates periodic maintenance entirely, only performing maintenance tasks when needed. This approach uses rulesets and an inference model to determine overall system health. While this approach may be effective, one major difficulty with applying such a model is the domain knowledge requested to set it up. Intimate understanding of the systems and nature in which they interact make this approach very laborious to stand up.

Kala et al. document a method in which natural language processing is used in the aerospace domain to organize and understand maintenance log reports (Kala, Analyzing Aircraft Maintenance Findings with Natural Language, 2022). Since many fault reports are written manually by maintainers, it is complex to synthesize these natural language datasets. Using a technique such as natural language processing can help to quantify the meaning of these write-ups and use this information to inform future decisions on corrective action.

2.3. Summary of NLP Methods

In exploring an appropriate AI method to apply to this problem, the nature of the fault codes must be first analyzed. These fault codes, issued automatically by the on-board system, are the language of the system. In many ways they parallel human language, as each code has meaning in itself, but is not valuable to communicate information until the context of its occurrence is seen. A string of these fault codes issued by the system can be thought of as a sentence, made up of many words in a specific order that communicate the state of the system. Because of these similarities, Natural Language Processing (NLP) techniques were investigated to see if they could provide value.

One solution explored was Word2Vec, a pip installable package that leverages the numpy python library (word2vec Tutorial, 2022). This technique was first published in 2013 (word2vec, 2023) and used a neural network model to explore the word associations in a large body of text. Word2Vec is not a single algorithm, but instead a set of model architectures that vectorize the individual words in a body of text by considering their surrounding context and inferred meaning. These types of models are notably used for text prediction. A famous example of this can be seen in smart phone technology, where AI will predict the next word in a sentence based on the previous words typed and personal vocabulary history. These predictions work through a neural network with one or more hidden layers, shown below in **Figure 1** (Introduction to Word Embedding and Word2Vec, 2023).

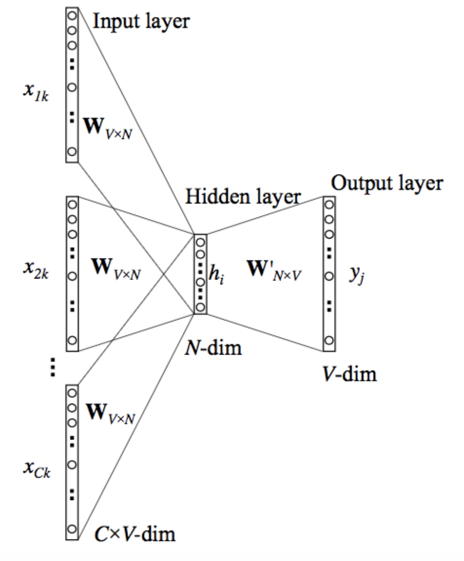

Figure 1: A Multi-Input Neural Network

In this figure, a variable input layer can intake C context words. Using one or many hidden layers, weights are applied to the connections between these matrices in order to make predictions on the target output layer. As this is established technology, further details on neural networks will not be explored in this paper. For more details on the operation of neural networks, please read further in the reference section of this paper.

3. DATA PRE-PROCESSING

To explore this theory, a dataset of historical maintenance data was used. In order to clean and sort the data into a format where it can be used to train a NLP model, the data was first loaded into a Domino workspace. Domino is a data science platform used to aid in the heavy computing of model training (Domino Data Labs, 2023). The data was first arranged into chronological order. It is important for the word context that the faults are time-ordered, as the position of the fault in the list of faults greatly affects the context and the prediction of similarity. The data was next segmented by flight, with each unique flight being a separate file to train the model. The dataset was then reduced to an ordered list of faults, leaving only the fields of fault codes. This reduction drops the timestamps of the faults as well as all other information surrounding them, but retains the chronological order of their occurrence. With the input data prepared, a model can then be trained. A sample of the data format can be seen below in **Table 1**.

Table 1: Example Data for Maintenance Log

Flight	Aircraft #	Fault Codes	Timestamp
F100	1	123-1000	1/1/2023 11:35:12.5
F100	1	123-2000	1/1/2023 11:35:13.1
F200	2	456-5000	1/2/2023 09:55:00.0
F200	2	456-5000	1/2/2023 09:56:20.0
F200	2	456-7000	1/2/2023 09:56:22.5

Note that although the additional fields of the flight number, the aircraft number, and the timestamp, were not used in the model training, but are included in the table for clarity of the data structure.

Once the data is sorted and cleaned into a format conducive for NLP, Word2Vec can be used to train a model on this corpus. The idea behind this approach is to treat a fault code dataset as natural language and perform NLP using existing algorithms. In **Table 2** below the translations between these two domains can be seen.

Table 2: The Bridge Between NLP and Aircraft Faults

Natural Language	**Aircraft System Equivalent**
Word	Single Fault Code
Sentence	Flight of Many Fault Codes
Corpus	Group of Many Flights

In this context, a fault code will be treated as a word, a flight full of codes will be treated as a sentence, and the entire dataset of all flights across all aircraft will be treated as the corpus. By using an approach like this, a problem with numeric data can be approached using NLP methods.

No filtering or windowing of the data was used. It is acknowledged that this could lead to some class imbalance, as the faults likely do not occur in proportional frequencies. Word2Vec is a strong modeling approach for these types of datasets, as the method of training the embeddings handles unbalanced data well (word2vec Tutorial, 2022), but this is an area that will receive additional attention in the future development of this project.

4. METHODOLOGY

With the data prepared, the model can be trained. Domino was used to perform the model training. The result of this effort is a Word2Vec model that contains the vectorized word embeddings of the training dataset. This model can then be used to extract fault code relationships and similarity to aid in troubleshooting. A diagram of this workflow can be seen below in **Figure 2**.

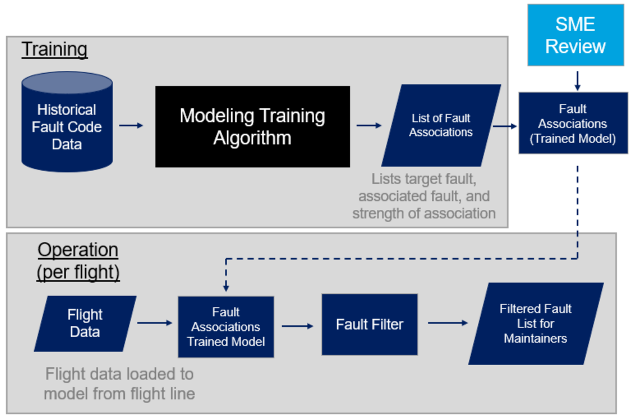
SME Review
Training
Historical Fault Code Data
Modeling Training Algorithm
List of Fault Associations
Fault Associations (Trained Model)
Lists target fault, associated fault, and strength of association
Operation (per flight)
Flight Data
Fault Associations Trained Model
Fault Filter
Filtered Fault List for Maintainers
Flight data loaded to model from flight line

Figure 2: Concept of Approach

Further information on how the word embeddings can be used to improve troubleshooting can be seen in the sections below.

4.1. Training Data Metrics

A large dataset of historical maintenance logs was used to train the model. Due to data privacy, the specifics of the data cannot be disclosed. However, below in **Table 3** are metrics on the dataset used to train this model.

Table 3: Metrics of Training Data

Metric	**Value**
Data Size	~50 GB
Number of Unique Flights	5000+
Data Points	~100 million
Timespan	5 years
Average Codes per Flight	~1100
Number of Unique Codes	5000+
Unique Tails	10+ (same aircraft type)

4.2. Model Cosine Similarity

With the word embeddings trained in the Word2Vec model, the similarities between these embeddings can be useful. This information is particularly useful for troubleshooting, as it gives indication as to which faults occur in similar situations and may have common root causes.

The way this information is extracted from the model is through the *similarity* metric built in to Word2Vec. Similarity computes the cosine similarity between two words used to train the model, as seen in Equation 1 (Introduction to Word Embedding and Word2Vec, 2023).

$$sim(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \tag{1}$$

From this equation, a high similarity score will occur when two words have high cosine similarity, indicating similar trajectory of their vectors. These values, referred from hereon as similarity “strengths”, indicate the level or correlation between two faults. The intention of this feature is to identify synonyms in natural language. However, in our use case, this similarity can be used to relate fault codes to each other. For each fault code in the model, the top 100 strengths were reported, with the only information extracted being the target fault, the associated faults, and the strength of their context similarity. Note that these relationships are symmetric, so the similarity of two faults is equal in both directions.

One important aspect of quantifying these cosine similarity scores is that these vectors are related to each other through context. A high strength in this respect refers to two codes that have similar surrounding context. When others codes and indicators are present from both before and after the fault of interest, it creates a specific context. Our goal in highlighting these similarity strengths is to identify faults that may have common causes or create similar downstream effects. The hypothesis is that these patterns of fault relation will give insight to root cause failures and help reduce noise in troubleshooting.

5. RESULTS

Since the objective of this work is to provide a tool for maintainers to aid in troubleshooting, the results of this experiment are not numeric metrics but instead specific examples of instances where this tool provided insight that could have led to cost-saving action. Due to their proprietary nature, those specific examples will not be shown in this paper. Instead, a general overview of the tool will be shown with acknowledgement of specific applications of the data it provides.

The main method of visualizing these similarity strengths was through the use of a Circos plot. A Circos plot is a visualization tool developed by Krzywinski to display data in a circular layout (Krzywinski, 2009). Although originally created with the field of biology and genomes in mind, the data format has many applications. One advantage of a visualization method like this over a similar word-embedding visual such as t-SNE 2D projection, is clarity. Using a Circos plot, only the top several faults can be displayed, showing clear connections between these faults through the connecting bands. This approach was chosen to quickly communicate the strongest relationships in the dataset and avoid a noisy plot that may be difficult to interpret. Note that all data shown in the following figures has been renamed and sanitized.

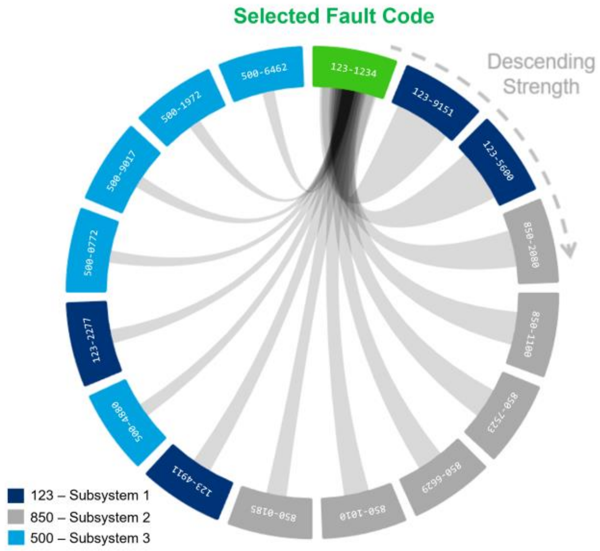

Figure 3: The Circos Plot

Figure 3 shows a Circos plot with a single fault code selected and highlighted in green (dash_bio.Circos Examples and Reference, 2023). The bands emanating from this fault connect to related faults, with these faults being arranged in descending order clockwise from the origin based on their cosine similarity strengths. This means that going around the circle clockwise will display the strongest associated fault code to the target first, followed by the second strongest, etc. The width of the band connecting two faults indicates the strength of connection. The color of the boxes maps to the first 3 digits of their fault codes, which indicate the subsystem from which they originate. This color scheme gives a clear indication to strong relationships across subsystems, a phenomenon that is particularly hard to observe without the aid of pattern recognition tools such as NLP.

The maintainer may further be interested in not only the strength of similarity between the selected fault and its top relatives, but also among those relatives themselves. To address this a feature was added to also display the strength bands between the various fault codes in the plot, as seen below in **Figure 4**.

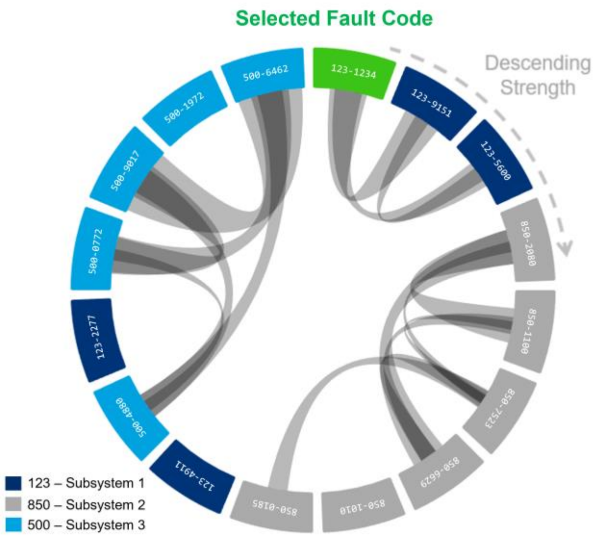

Figure 4: The Circos Plot with Internal Bands

As shown, additional bands are now plotted connecting various faults to each other and showing their strength of similarity by the thickness of this band. Note that some bands in **Figure 4** are hidden to avoid visual clutter. This was done by adjusting a filter to only display strengths between two fault codes that exceed a defined strength. This is a vital feature to avoid cluttering the plot with excessive bands, nullifying the information trying to be communicated. For additional information, the user can hover over a specific band, which displays the source and target, as well as its corresponding strength.

6. CONCLUSION

This project successfully demonstrated that a dataset such as fault codes on an aircraft system can successfully be analyzed using NLP methods. By training a Word2Vec model using only an ordered list of fault codes, the context of their issuance can be observed and quantified. Many examples were found where this tool highlighted a connection between two fault codes that may have proved useful for a maintainer. In these situations, incorrect action was taken that may have been avoided with proper knowledge of the relationships between these faults. Due to ever increasing system complexity, this problem is only becoming more relevant, and an automated AI tool to address this could be extremely valuable. Any information offered by a tool that can help a maintainer isolate the cause of a failure will lead to significant cost reduction. While autonomous fault detection is not currently fully actualized, this advancement is a vital stepping stone towards that capability.

7. CONTINUED WORK

This project is ongoing at Lockheed Martin. While the current implementation serves only as a tool for a user to aid in the troubleshooting and decision making process, future uses of this model may include fully automating a pattern matching system to connect current situations to past corrective actions. If the model could recognize the behavior it is seeing in real time and connect this behavior to a past issue which led to an active resolution, the model could recommend the same action with a given confidence. Although reducing cost is a main objective, there are impressive benefits to technology like this beyond fiscal. The safety of an aircraft could also improve as the chances of unexpected events would reduce due to proper addressing of maintenance issues.

As this work continues in industry, the benefits of such technology offer exciting opportunities for the future of health management in the Aerospace community.

8. REFERENCES

Pypi word2vec. (2023, 5 17). Retrieved from word2vec: https://pypi.org/project/word2vec/

(2023, 05 17). Retrieved from dash_bio.Circos Examples and Reference: https://dash.plotly.com/dash-bio/circos

Ezhilarasu, C. M. (2019). The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities. *Progress in Aerospace Sciences*, 14.

Heisey, R. (n.d.). *Maintenance Costs*. Retrieved from boeing.com: https://www.boeing.com/commercial/aeromagazine/aero_19/717_story.html

Introduction to Word Embedding and Word2Vec. (2023, 05 17). Retrieved from Towards Data Science: https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

Kala, M. (2022). Analyzing Aircraft Maintenance Findings with Natural Language. *11th International Conference on Air Transport – INAIR 2022*, 8.

Krzywinski, M. (2009). Circos: an information aesthetic for comparative geonomics.

word2vec Tutorial. (2022, 12 20). Retrieved from TensorFlow: https://www.tensorflow.org/tutorials/text/word2vec